



Distributed control strategies for wind farm power production optimization

Nicolo Gionfra

► To cite this version:

Nicolo Gionfra. Distributed control strategies for wind farm power production optimization. Automatic. Université Paris Saclay, 2018. English. NNT: . tel-01812531v1

HAL Id: tel-01812531

<https://centralesupelec.hal.science/tel-01812531v1>

Submitted on 11 Jun 2018 (v1), last revised 11 Jun 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stratégies de commande distribuée pour l'optimisation de la production des fermes éoliennes

Thèse de doctorat de l'Université Paris-Saclay
préparée à CentraleSupélec

École doctorale n°580 Science et technologies de l'information et de
la communication (STIC)
Spécialité de doctorat: Automatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 15 mars 2018, par

M. Nicolò Gionfra

Composition du Jury :

M. Jan-Willem Van Wingerden Professeur, TU Delft (DCSC)	Président
M. Luigi Glielmo Professeur, Università degli Studi del Sannio (GRACE)	Rapporteur
M. Dimitri Lefebvre Professeur, Université du Havre (GREAH)	Rapporteur
M. Mohammed M'Saad Professeur, ENSICAEN (GREYC)	Examineur
Mme Elena Panteley Directrice de recherche, CentraleSupélec (L2S)	Examinatrice
M. Guillaume Sandou Professeur, CentraleSupélec (L2S)	Directeur de thèse
M. Salvatore Monaco Professeur, La Sapienza (DIAG)	Co-Directeur de thèse
Mme Houria Siguerdidjane Professeure, CentraleSupélec (L2S)	Co-Directrice de thèse
M. Damien Faille Ingénieur chercheur, EDF R&D (PRISME)	Co-Encadrant
M. Philippe Loevenbruck Ingénieur chercheur, EDF R&D (EFESE)	Co-Encadrant

*A mio nonno,
ingegnere d'altri tempi*

*"How many roads must a man walk down
Before they can call him a man ?
How many seas must a white dove sail
Before she sleeps in the sand ?
How many times must the cannon balls fly
Before they're forever banned ?
The answer, my friend, is blowin' in the wind
The answer is blowin' in the wind"*

Bob Dylan – 1962

Acknowledgments

The presented research work was carried out in the context of a double degree program between the L2S (Laboratoire des Signaux et Systèmes) laboratory of CentraleSupélec and the DIAG (Dipartimento di ingegneria Informatica Automatica e Gestionale Antonio Ruberti) department of La Sapienza, University of Rome, and in collaboration with EDF (Electricité de France).

The results shown in this PhD work were achieved thanks to the support I received from the great people I met and with whom I shared both an enriching professional and human experience.

This is why, first of all, I would like to express my gratitude to my supervisors. On CentraleSupélec side my warm thanks go to Professors Guillaume Sandou and Houria Siguerdidjane who, not only showed a constant availability and willingness of sharing opinions, but also always ensured a smooth progress of the PhD work in a familiar environment.

On the side of La Sapienza, I would like to thank Professor Salvatore Monaco who made the double degree program as well as the collaboration between Centrale-Supélec and La Sapienza possible.

Beyond such collaboration, the added value of this research work was undoubtedly given by the direct participation of EDF. For this reason I would like to extend my gratitude to research engineers Damien Faille and Philippe Loevenbruck, as their relevant advice throughout the PhD work consolidated the importance of a joint effort between industry and academia.

I would like to express my appreciation to Professor Jan-Willem Van Wingerden, from Delft University of Technology, as it has been an honor for me to have him as president of this PhD jury. His valuable work on wind energy inspired me throughout my research work.

I would also like to convey my thanks to Professors Dimitri Lefebvre, from Université du Havre, and Luigi Glielmo, from Università degli Studi del Sannio, for the time and patience they dedicated to review this work.

My acknowledgments also go to Elena Pantely, research director in the L2S laboratory, and to Mohammed M'Saad, Professor at ENSICAEN, who kindly accepted to be examiners of my work and members of the jury.

I would like to express my gratitude to Professor Didier Dumur and Madame Pascale Lepeltier who welcomed me into the automatic department as well as to all

the professors and researchers I had the chance to learn from and with whom I always had interesting and stimulating discussions. They all contribute to alimentering the friendly atmosphere that makes the department a beautiful place to work.

The past three years would have not been this valuable if had not the possibility to share countless precious moments with my travel companions. Thus, my special thoughts go to Guillaume, Miassa, Iris, Mert, Sophie, Rajesh, Tahar, Geoffray, Nollwenn, Marjorie, Andreaa, Juliette, Ion, Nathan, Thomas, Fetra, Mattia, Giovanni and Daniele for all the good memories that I will jealousy keep with me.

Last but not least, my special thanks go to my mother, my father, my sister and my grandparents who always supported me in my decisions and in overcoming the hardest moments, and to Giulia without whom this work would have not been possible.

Résumé de la Thèse en Français

DURANT la dernière décennie, l'énergie éolienne, en tant que moyen innovant de génération d'électricité, a eu un fort développement dans le monde et a atteint ainsi un haut niveau de capacité installée. La Chine représentait 42.8% de cette capacité en 2016. En particulier, l'évolution croissante de cette capacité au cours des années, mise en avant par le Global Wind Energy Council (GWEC), a atteint 468,8 GW en 2016, dont 153,7 GW installés dans l'Union Européenne. De plus, selon WindEurope, l'énergie éolienne représente aujourd'hui le deuxième moyen de production d'énergie en termes de puissance installée, en ayant dépassé la génération à base de carbone en 2016.

Il est aussi important de mentionner que dans plusieurs pays, la consommation d'énergie est aujourd'hui couverte en grande partie par celle de l'éolien. Le Danemark par exemple, selon le GWEC, affichait en 2016 une pénétration de puissance éolienne qui a atteint un taux de 37,6% de la consommation annuelle totale. De plus, le gouvernement Danois vise un taux de 50% d'ici 2020.

Ces tendances de croissance du marché de l'éolien peuvent être trouvées de manière similaire aussi dans d'autres pays européens, car elles reflètent l'intérêt de suivre les objectifs posés par la COP21. Ce dernier point est confirmé par exemple par le fait qu'en 2016, selon le GWEC, l'utilisation de l'électricité de nature éolienne a permis d'éviter l'émission de plus de 637 millions de tonnes de CO₂ dans le monde entier.

Contexte de la thèse

La forte croissance du marché de production d'énergie éolienne, comme mentionnée ci-dessus, génère en parallèle la croissance de l'innovation technologique nécessaire pour soutenir un haut niveau d'exploitation de la ressource du vent. Cela fait en même temps avancer les travaux de recherche consacrés aux systèmes d'énergie éolienne. Dans cette thèse les fermes éoliennes représentent les systèmes à étudier. Ces dernières peuvent être de différentes tailles et capacités selon le lieu d'installation et les besoins de génération d'énergie. Les fermes éoliennes «offshore» sont typiquement composées d'un nombre élevé de turbines éoliennes, et elles ont des hauts niveaux de capacité installée. On peut par exemple citer le cas de la ferme éolienne «London Array» en Angleterre, qui est composée de 175 éoliennes et qui a une capacité correspondante de 630 MW.

Dans ces travaux de thèse nous avons fait le choix de ne pas focaliser a priori l'attention sur des cas spécifiques de puissance installée, qui définiraient le cahier des charges et les objectifs principaux. En revanche, nous préférons atteindre un plus haut niveau d'abstraction en considérant les problématiques d'intérêt concernant le pilotage des fermes éoliennes de manière générale, pour ensuite les appliquer le cas échéant sur une ferme particulière. La conversion d'énergie dans les fermes éoliennes concerne deux étapes principales :

- la conversion de la puissance du vent en puissance mécanique ;
- la conversion de la puissance mécanique en puissance électrique.

La première étape est obtenue par le système mécanique des turbines éoliennes. Nous considérerons dans ces travaux la technologie à axe horizontal, [Burton 2011]. D'autres types de technologie, comme celle à axe vertical, sont typiquement employées dans le cas de production d'énergie de basse capacité et notamment à l'échelle du consommateur, [Bhutta 2012].

Le rotor est la partie mécanique fondamentale des éoliennes car il capture l'énergie cinétique du vent. Si auparavant, la technologie du rotor était à vitesse fixe, aujourd'hui la plus récente et sans doute la plus utilisée est à vitesse variable. Le rotor peut varier sa vitesse dans une plage de valeurs données, [Carlin 2003]. C'est grâce à cette variation que l'on peut adapter le fonctionnement des éoliennes selon la variabilité du vent pour accomplir des nouveaux objectifs pour les fermes éoliennes, ce qui constitue l'objet de ces travaux.

La conversion de la puissance mécanique en puissance électrique est obtenue par un générateur électrique. Pour la technologie à vitesse variable du rotor, cela repose sur deux solutions possibles. Ce sont soit le générateur d'induction à double alimentation (DFIG), soit le générateur à aimants permanents (PMSG). Dans le premier cas, le bobinage du stator est directement couplé avec le réseau électrique, alors que le bobinage du rotor du générateur est connecté par un convertisseur d'électronique de puissance. Cela permet au rotor de l'éolienne de varier sa vitesse jusqu'à environ 30% de sa valeur nominale. L'avantage principal de cette solution est que le convertisseur de puissance doit être dimensionné par rapport à la seule puissance nominale qui afflue par le bobinage du rotor, et cela réduit les coûts.

Dans le cas des éoliennes fondées sur une machine PMSG, la puissance afflue par le bobinage du stator qui est connecté au réseau par un convertisseur de puissance. Si d'un côté cela augmente le coût du convertisseur lui-même, de l'autre cela permet la variation complète de la vitesse du rotor de l'éolienne, (voir par exemple [Baroudi 2007, Liserre 2011]).

Le pilotage d'une ferme éolienne est typiquement implémenté à travers une structure hiérarchisée fondée sur deux niveaux principaux, (voir par exemple [Hansen 2006]). Au bas niveau de la structure du contrôle-commande, les éoliennes sont typiquement exploitées afin d'extraire le maximum de puissance disponible du vent et d'en limiter l'extraction lorsque la puissance du vent atteint des valeurs plus grandes que celle de la puissance nominale des turbines. Les modes de fonctionnement cités représentent l'état de l'art de la technologie implémentée dans le pilotage des éoliennes et ils peuvent être considérés comme les conditions d'exploitation classiques.

Le pilotage de haut niveau d'une ferme éolienne est réalisé de manière centralisée par des systèmes de type SCADA (Supervisory Control and Data Acquisition), [Crabtree 2014]. Un ordinateur central exploite les données reçues des éoliennes de la ferme afin d'assurer la supervision. Par exemple, les données historiques et en temps réel peuvent être analysées pour la gestion de l'aménagement, etc. Les objectifs du contrôle-commande de haut niveau concernent plutôt la gestion du démarrage et de

l'arrêt des éoliennes.

Plus récemment, la proportion de la production d'énergie renouvelable dans le réseau électrique a fortement changé le rôle que les parcs éoliens doivent avoir par rapport à la satisfaction des contraintes du réseau ainsi qu'à l'équilibre entre production et consommation. S'il leur était précédemment demandé de simplement injecter le maximum de puissance extractible du vent dans le réseau électrique, on attend aujourd'hui d'eux qu'ils participent activement aux opérations du réseau. Celles-ci sont liées à la capacité du système électrique de garder son fonctionnement nominal en présence de perturbations. Précisément cela concerne l'exploitation du réseau électrique tout en garantissant une fréquence du système constante ainsi que des valeurs de tension acceptables autour de la valeur nominale. Les perturbations du réseau peuvent être de différente nature, par exemple pertes de production de puissance, changement des charges, etc.

Il est bien connu que la fréquence du réseau électrique est directement liée à l'équilibre entre la puissance active produite et celle consommée, et que toute perturbation de cet équilibre se reflète à travers une variation de fréquence. D'autre part, le réglage de puissance réactive est strictement lié à la stabilité de tension et aux pertes de puissance dans les lignes de transmission, [Kundur 2004]. C'est pour cela qu'afin de garantir une exploitation correcte du système, les centrales électriques classiques doivent participer à la régulation de tension et de fréquence en agissant sur les valeurs de puissance active et réactive injectées dans le réseau.

Dans le passé, les fermes éoliennes n'étaient pas sujettes à respecter ces conditions. De plus, il leur était imposé de se déconnecter du réseau pendant les conditions de fonctionnement anormales. Dans le contexte actuel où l'on voit une forte croissance de la pénétration d'énergie de nature éolienne dans le réseau, les fermes éoliennes doivent respecter les conditions définies par un code du réseau adapté. Les problématiques principales concernant le branchement des fermes éoliennes au réseau sont la régulation de fréquence, de tension et de puissance réactive, et la capacité à rester connectées pendant les défauts du système électrique, [de Alegría 2007].

Dans ces travaux, nous focalisons l'attention sur la régulation de puissance active. C'est pour cela que dans la suite nous allons rappeler pourquoi la participation des fermes éoliennes au réglage de fréquence est important dans le cas de haut niveau de pénétration dans le réseau. Tout d'abord il faut noter que dans le système électrique classique, où les générateurs synchrones sont directement couplés au réseau, l'inertie globale du système est élevée et cela permet de maintenir des valeurs de déviation de fréquence limitées face à un déséquilibre entre production et consommation de puissance. Autrement dit, dans ce cas, les générateurs contribuent naturellement à l'inertie du système, [Ulbig 2015]. Si, au contraire, un pourcentage significatif de puissance est injecté par les fermes éoliennes, l'inertie globale du système diminue car ces dernières sont découplées du réseau par les convertisseurs de puissance. Cela pose un premier problème intéressant de pilotage concernant le réglage de fréquence par les fermes éoliennes. En particulier, l'inertie du réseau pourrait être produite artificiellement par des techniques de contrôle-commande, [Morren 2006, Mauricio 2009]. Si d'un côté l'inertie du système doit être émulée en permanence par des moyens de

réglage, d'un autre côté, si des perturbations de fréquence plus graves se produisent, alors les fermes éoliennes peuvent être amenées à fournir de la réserve primaire de puissance pour participer aussi à ce réglage primaire de fréquence. Plus précisément, elles doivent pouvoir moduler la valeur de puissance injectée dans le réseau selon les variations de fréquence. Cela implique a priori la nécessité d'avoir de la réserve de puissance à la fois à la baisse et à la hausse.

De manière générale, la participation au réglage de fréquence à la hausse peut encore être considérée économiquement désavantageuse car cela nécessite de gaspiller en permanence une portion de la puissance disponible du vent. C'est pour cela que dans la suite de nos travaux nous ne nous intéressons qu'au réglage de fréquence à la baisse.

Les problématiques liées à la connexion des fermes éoliennes au réseau, que nous avons brièvement exposées, alimentent les travaux de recherche à mener sur ce sujet. En même temps, l'effort de recherche consacré aux systèmes d'énergie éolienne permet en retour de mieux comprendre le fonctionnement de ces systèmes, et cela peut amener à de nouvelles opportunités. C'est le cas de l'effet de sillage parmi les éoliennes qui entraîne leur couplage aérodynamique. Ce phénomène peut être décrit par le fait qu'une turbine, extrayant de l'énergie cinétique du vent, induit une réduction de la vitesse du vent dans le sillage en aval. Par conséquent, toute éolienne placée dans le sillage d'une turbine perçoit une réduction de la puissance disponible du vent, [Park 2015a]. Puisque les conditions de vent de toutes les turbines sont influencées par les turbines situées en amont, le fonctionnement d'une ferme éolienne ne peut pas être décrit comme une simple composition d'unités indépendantes. Cela pose de nouveaux défis pour le problème du pilotage des fermes éoliennes afin de garantir l'exploitation optimale de la ressource de vent. En effet, quand l'effet de sillage n'est pas négligeable, sa prise en considération dans le problème d'optimisation de la ferme éolienne peut amener à des gains intéressants en termes de puissance extraite du vent par rapport au cas où la ferme est exploitée dans le mode plus classique où le couplage entre les turbines n'est pas pris en compte.

Les défis à relever se déclinent donc sur deux axes principaux. L'un concerne la modélisation du phénomène d'effet de sillage, qui est par nature assez complexe, l'autre regarde la méthode et l'architecture de pilotage à choisir pour faire fonctionner la ferme éolienne de manière optimale en considérant ce phénomène.

Contributions

Nous avons indiqué quelques problématiques et défis d'intérêt central pour le développement des systèmes d'énergie éolienne dans un futur proche. D'une part, les fermes éoliennes doivent être en mesure de respecter les contraintes imposées par un nouveau code du réseau. D'autre part, le développement avancé des techniques de contrôle-commande et d'optimisation pour ce type de système pousse les objectifs du pilotage des fermes éoliennes vers une exploitation améliorée de la ressource de vent.

Du point de vue du pilotage d'une ferme éolienne, ces objectifs et opportunités peuvent être traités comme étant des contraintes et des spécifications concernant la puissance active et réactive injectées dans le réseau. En outre, comme les éoliennes sont typiquement basées sur la technologie PMSG ou DFIG, elles sont découplées du réseau électrique, et le réglage de puissance active et réactive peut-être mené par des contrôleurs indépendants, [Arifujjaman 2010]. Le pilotage de la puissance réactive est effectué en agissant sur les convertisseurs de puissance, alors que le pilotage de puissance active concerne à la fois le système électrique et le système mécanique des éoliennes. Dans la suite nous faisons le choix de focaliser l'attention sur le seul pilotage de puissance active. De plus, nous consacrons une attention particulière aux étapes de conversion de puissance qui concernent les phénomènes aérodynamiques et mécaniques. Cela est validé par le fait que, typiquement, les pilotages des systèmes mécaniques et électrique agissent sur des échelles temporelles différentes et ils peuvent être considérés comme découplés, [Boukhezzer 2006a]. Pour ce qui concerne la conception du pilotage des fermes éoliennes, dans ces travaux de thèse, nous allons considérer des spécifications assez générales sur la puissance active injectée dans le réseau et sur la puissance extraite du vent. Cela nous permet d'atteindre un niveau d'abstraction nécessaire pour traiter de manière générale plusieurs problématiques liées au pilotage de ces systèmes.

Comme déjà mentionné, nous allons nous focaliser sur le contrôle de la seule puissance active des fermes éoliennes. Nous pouvons donc dresser la liste des applications principales qui la concernent. Celles-ci sont :

- le réglage de fréquence ;
- la maximisation de puissance extraite du vent ;
- l'effacement de puissance injectée dans le réseau.

On peut noter que nous avons ajouté l'effacement de puissance à la liste des objectifs déjà abordés. Cela représente une condition que les fermes éoliennes doivent être en mesure de satisfaire pour être raccordées au réseau. En particulier cela consiste à limiter la puissance maximale qui peut être injectée dans le réseau pour certaines périodes de temps. Cela est typiquement imposé par le gestionnaire de réseau pour résoudre des problèmes de congestion, [Delille 2013].

L'architecture de contrôle-commande choisie est fondée sur une structure hiérarchisée à deux niveaux. D'une façon générale, le niveau de contrôle plus élevé fournit les références de puissance à suivre par chaque éolienne, lesquelles sont logiquement placées au bas niveau de la structure hiérarchisée. Nous adressons tout d'abord le problème de pilotage des turbines éoliennes, où nous décrivons notre première contribution. De plus, nous focalisons l'attention sur des techniques de contrôle-commande de type non linéaire car, comme montré par [Boukhezzer 2010], celles-ci permettent d'atteindre de meilleures performances comparativement aux contrôleurs de type linéaire, tout particulièrement quand le profil du vent est très turbulent.

Pour qu'une turbine puisse satisfaire les objectifs au niveau du parc, l'éolienne doit

pouvoir fonctionner en mode de production réduite. Autrement dit, l'éolienne doit suivre des signaux de référence de puissance autres que ceux d'extraction maximale du vent. Néanmoins, la plupart des contrôleurs non linéaires proposés en littérature sont conçus pour des zones de fonctionnement bien définies, plus précisément soit à vent faible pour l'extraction maximale de puissance (par exemple [Boukhezzar 2006b]), soit à vent fort pour la limitation à la valeur nominale de la puissance extraite de l'éolienne (par exemple [Boukhezzar 2005, Boukhezzar 2007]). Nous proposons une nouvelle approche de contrôle-commande, d'abord pour l'application au cas d'une éolienne, permettant le suivi d'une trajectoire de puissance réduite et générale en agissant à la fois sur la vitesse angulaire du rotor et sur l'angle de calage de la turbine. De plus le contrôleur n'est pas confiné à l'utilisation dans une zone de contrôle spécifique. Au contraire il est actif pour toute condition de fonctionnement.

Parmi les techniques disponibles en littérature, (par exemple [Yingcheng 2011]), sur les traces des travaux de [Zertek 2012b] nous considérons une approche qui permet de faire fonctionner une éolienne en mode de production réduite tout en maximisant l'énergie cinétique stockée dans ses masses tournantes. Une fois la synthèse du contrôleur de l'éolienne déterminé, il est intégré au niveau de la ferme. Nous nous intéressons alors à la conception du contrôleur au plus haut niveau de la structure de contrôle-commande hiérarchisée de la ferme. Ceci permet le respect des contraintes de connexion de la ferme au réseau tout en optimisant son fonctionnement en intégrant la connaissance de l'effet de sillage. Nous choisissons d'analyser le problème du contrôle au niveau de la ferme dans le contexte des systèmes multi agents, dans le but d'atteindre une architecture évolutive, modulaire et résiliente, et qui permet une exploitation en temps réel. Ceci nous conduit à envisager une architecture distribuée.

Dans ces travaux de recherche, les objectifs de pilotage d'une ferme sont reformulés sous la forme d'un problème d'optimisation non convexe. Ce genre de problèmes peut être traité avec une structure distribuée et nous focalisons ainsi notre recherche sur des versions distribuées de techniques bien connues d'optimisation méta-heuristique. L'utilisation de ces méthodes d'optimisation dans le contexte de systèmes multi agents représente une nouveauté par rapport aux travaux disponibles en littérature et cela mérite une attention toute particulière. Dans les travaux de [Wakasa 2015a, Wakasa 2015b] par exemple, même si les agents coopèrent afin de résoudre un problème commun, ceux-ci ne partagent pas la même variable d'optimisation. Cela exclut donc leur utilisation pour le cas du problème d'optimisation de la ferme éolienne où les agents, c.-à-d. ici les éoliennes, sont couplés par l'effet de sillage. Dans d'autres travaux, comme ceux de [Gazi 2014, Navarro 2015] par exemple, l'algorithme d'optimisation méta-heuristique est distribué parmi les agents qui, à tour de rôle, partagent une variable d'optimisation commune. Malheureusement ces derniers algorithmes sont conçus pour des problèmes très spécifiques et ils ne peuvent pas être directement appliqués au cas de la ferme éolienne. C'est ce qui nous a motivé pour introduire deux nouveaux algorithmes d'optimisation distribuée, avant de traiter le problème de la ferme. De plus, nous les présentons dans

une formulation générale qui permet leur utilisation pour une classe de problèmes d'optimisation distribuée plus large.

Dans une première étape, à titre d'exemple, nous les appliquons à la résolution du problème du flux optimal de puissance qui est un autre problème bien connu d'optimisation non convexe dans le domaine de l'ingénierie électrique. Dans la littérature, les approches distribuées proposées à ce sujet sont typiquement fondées soit sur l'approximation du problème pour être traité par des techniques basées sur le calcul du gradient (par exemple [Sun 2013, Magnússon 2015]), soit par des méthodes de relaxation convexe exacte pour l'application au cas de réseaux définis par des topologies particulières, (par exemple [Peng 2014, Dall'Anese 2013]). Les deux algorithmes proposés permettent en revanche de traiter le problème du flux optimal sans la nécessité de considérer des hypothèses trop contraignantes.

Dans une deuxième étape nous adressons le problème spécifique de la ferme éolienne. En particulier, nous considérons la maximisation de la puissance extraite du vent sous les contraintes de puissance injectée dans le réseau électrique. Dans la littérature, ce problème d'optimisation est typiquement traité soit dans le contexte distribué par des approches sans modèle (par exemple [Gebraad 2016, Marden 2013, Park 2016]), soit dans le contexte centralisé par des méthodes qui exploitent la connaissance a priori du modèle de l'effet de sillage, (par exemple [Park 2015a, Herp 2015]). Néanmoins, les approches sans modèle reposent sur l'apprentissage en temps réel sur la base des données du système ce qui typiquement augmente le temps de convergence des algorithmes. Grâce aux algorithmes d'optimisation méta-heuristique proposés nous sommes en mesure de considérer à la fois le modèle de l'effet de sillage et une architecture distribuée, ce qui nous permet d'atteindre des bonnes propriétés de convergence des algorithmes.

Enfin, l'architecture globale de pilotage de la ferme est évaluée. Cela nous permet de faire le lien entre les deux niveaux de contrôle considérés, en même temps qu'estimer le gain de puissance obtenu quand la dynamique du système est prise en compte. En effet, hormis quelques exceptions (par exemple [Heer 2014]), peu de travaux de la littérature existante ont traité l'évaluation du système de pilotage dans sa totalité, car il est souvent supposé que les contrôleurs au niveau des éoliennes stabilisent parfaitement les turbines pour toute consigne de puissance donnée, [Marden 2013]. Jusqu'ici, l'architecture de pilotage considérée est décentralisée pour les contrôleurs locaux des éoliennes car leur rôle est de permettre le suivi des trajectoires de puissance imposées par l'optimisation de haut niveau. Dans la troisième partie de ces travaux, nous proposons d'ajouter une dernière étape de coopération parmi les éoliennes car cela peut, dans certains cas, améliorer les performances du système global. De manière générale, nous montrons que par des moyens du contrôle par consensus, on peut améliorer le respect des consignes de puissance calculées par l'étape d'optimisation en dépit des dynamiques du système et des perturbations du vent. Cela est obtenu par la formulation du problème de contrôle en termes de partage de puissance disponible du vent parmi les éoliennes, sur lequel ces dernières doivent trouver un accord. Dans la littérature, il existe des problèmes similaires de partage de puissance dans le contexte des fermes éoliennes, (par exemple [Biegel 2013, Spudić 2015, Baros 2017]),

mais à la fois la problématique spécifique adressée dans ces travaux de recherche et la technique de contrôle-commande proposée représentent une nouvelle contribution.

Organisation de la thèse

Le manuscrit est composé de trois parties. Chaque partie est consacrée à un problème spécifique de pilotage des fermes éoliennes et est divisée en deux chapitres. Le premier introduit les outils mathématiques utilisés dans le deuxième chapitre, où les problématiques de l'application au cas des fermes éoliennes sont traitées. Par le choix de cette organisation de thèse nous avons aussi le but de créer un document autoporteur. Les résultats qui s'avèrent plus classiques sont quant à eux reportés en annexe. L'état de l'art relatif à tout argument traité est analysé dans le détail chaque fois que nous proposons une nouvelle contribution.

Partie I

Cette partie est consacrée au problème du contrôle-commande d'une éolienne, et elle comporte les Chapitres 1 et 2. Les problématiques principales traitent du pilotage d'une turbine dans les zones de fonctionnement classiques et du suivi de trajectoires de puissance réduite. L'approche considérée est basée sur la combinaison de deux techniques de contrôle bien connues : la linéarisation par feedback (FL) et la commande prédictive par modèle (MPC). Dans le Chapitre 1, nous rappelons les éléments théoriques de cette commande référencée FL+MPC, et nous analysons des problématiques complémentaires qui portent sur le traitement de points singuliers et les contraintes non convexes du système. Par souci de clarté pour le lecteur, les détails sur les techniques de FL et MPC sont reportées en Annexe A.

Dans le Chapitre 2, le modèle dynamique et le contrôleur d'une turbine sont présentés. Nous focalisons l'attention sur le système mécanique de l'éolienne car il a un rôle central dans la synthèse du contrôleur. Les modes de fonctionnement de l'éolienne y sont décrits. Ceux-ci concernent à la fois l'extraction maximale de puissance du vent et le suivi de trajectoires générales de puissance réduite selon la nécessité au niveau du parc. De plus, quand l'éolienne fonctionne en mode de production réduite, le schéma de pilotage proposé permet de maximiser le stockage de l'énergie cinétique dans les masses tournantes de la turbine. Le contrôleur est complété par l'utilisation d'un filtre de Kalman pour l'estimation du vent. Les performances sont évaluées en simulation en utilisant les paramètres de l'éolienne CART.

Partie II

La deuxième partie est composée des Chapitres 3 et 4. Nous analysons les objectifs de pilotage au niveau du parc éolien. Comme il a été anticipé, le problème de contrôle de haut niveau est formulé par un problème d'optimisation qui est résolu de manière distribuée parmi les éoliennes. Ainsi, dans le Chapitre 3, nous proposons deux nouveaux algorithmes d'optimisation distribuée qui peuvent être utilisés dans

le cas d'application de parcs. Ceux-ci sont basés sur l'algorithme d'optimisation par essais particuliers, dont les caractéristiques principales sont rappelées. Comme décrit dans la littérature en ce qui concerne l'optimisation distribuée, nous fondons le premier algorithme sur l'utilisation de techniques de contrôle par consensus. En particulier, nous nous appuyons sur la technique de consensus moyen en temps fini proposé par [Dung 2013]. Les deux algorithmes sont présentés de manière générale et ils peuvent être utilisés dans le cas d'application d'une classe de problèmes d'optimisation distribuée. Par exemple, nous montrons l'application de ces algorithmes au cas du problème bien connu de flux de puissance optimal.

Dans le Chapitre 4, la structure hiérarchisée du pilotage de la ferme est présentée. Elle est fondée sur le contrôleur analysé au Chapitre 1 et sur un contrôleur distribué de plus haut niveau qui coordonne les éoliennes de la ferme sur la base de la solution d'un problème d'optimisation. Ainsi, une partie importante de ce chapitre est consacrée à la définition et à la solution du problème d'optimisation d'une ferme éolienne. Le modèle de l'effet de sillage et les contraintes de puissance à injecter dans le réseau sont pris en considération. Tout d'abord, nous présentons un modèle de sillage disponible dans la littérature et fondé sur les travaux de [Park 2015a, Gebraad 2016]. Ensuite nous posons le problème d'optimisation. Celui-ci appartient à la classe de problèmes qui peuvent être résolus par les deux algorithmes proposés au Chapitre 3. Leurs performances sont donc évaluées et comparées sur plusieurs exemples de fermes éoliennes. Si d'un côté les deux algorithmes montrent de bonnes performances pour le cas de l'optimisation sans contraintes, de l'autre, le deuxième algorithme est plus performant en termes d'optimalité de la solution dans les cas de contraintes actives. De plus, il peut facilement s'adapter aux différentes tailles de fermes éoliennes considérées, et aussi préserver de bonnes propriétés de vitesse de convergence.

L'étape d'optimisation se base sur des hypothèses de pilotage parfait de la ferme éolienne. En particulier, la dynamique du système est négligée. C'est pour cela que, comme dernière étape, nous proposons d'évaluer l'architecture globale de pilotage de la ferme en considérant les dynamiques contrôlées des éoliennes. Les simulations montrent la présence d'un écart entre les gains en puissance atteignables par l'architecture proposée et ceux donnés par la seule solution du problème d'optimisation. Néanmoins, ces gains peuvent être significatifs et cela confirme l'intérêt pour les méthodes de pilotage coopératif des fermes éoliennes quand l'effet de sillage n'est pas négligeable.

Partie III

Cette dernière partie est composée des Chapitres 5 and 6, et elle est consacrée à l'introduction d'un autre niveau de contrôle dans la structure hiérarchisée du Chapitre 4. L'idée principale est celle de permettre une communication complémentaire entre les éoliennes au niveau des contrôleurs locaux. Cela définit un problème de contrôle distribué qui est résolu par l'utilisation de techniques de consensus. Ainsi, au Chapitre 5, nous introduisons une nouvelle technique de contrôle-commande par

consensus fondée sur une structure PID, et qui peut être appliquée aux systèmes dynamiques LTI MIMO en temps discret. De plus elle peut être utilisée à la fois pour des problèmes sans leader et dans le contexte «leader-follower».

Les outils cités auparavant sont ensuite appliqués au cas spécifique du pilotage distribué d'une ferme éolienne. Néanmoins, puisqu'ils sont conçus pour les systèmes linéaires, au Chapitre 6 nous procédons tout d'abord à la synthèse d'un contrôleur local pour l'éolienne basée sur la technique de poursuite asymptotique de sortie. Cela permet par ailleurs de linéariser par feedback la dynamique de l'éolienne. Les résultats concernant le pilotage d'une turbine introduits au Chapitre 2 ne sont pas utilisés ici, car la structure de FL+MPC ne simplifierait pas la synthèse d'un contrôleur par consensus à cause de l'étape de MPC.

En somme, le contrôle par consensus permet de modifier les références de puissance à suivre, données par l'étape d'optimisation, pour prendre en considération les dynamiques du système et les perturbations du vent afin de garantir des meilleures performances. En particulier, d'une part, la technique de consensus sans leader permet de réduire l'effet de perturbations à moyenne nulle sur le partage optimal de puissance du vent parmi les turbines. D'autre part, le consensus par «leader-follower» a le rôle de restaurer ce partage optimal dans le cas d'erreurs dans les références de puissance.

Le manuscrit se termine par une conclusion générale et des perspectives, ainsi que des annexes et une bibliographie.

Contents

Acknowledgments	iii
Résumé de la Thèse en Français	v
List of Figures	xxi
List of Tables	xxv
List of Main Symbols and Acronyms	xxvii
List of Publications	xxxix
GENERAL INTRODUCTION	1
Part I WIND TURBINE CONTROL	15
1 Combined Feedback Linearization and MPC	17
1.1 Constrained Feedback-linearizable Systems	18
1.1.1 Exact-linearizable Systems	18
1.1.2 Input-output Linearizable Systems	19
1.2 Avoiding Singular Points	20
1.3 Conclusion	26
2 Wind Turbine Controller Design	27
2.1 Introduction	28
2.1.1 Wind Turbine Control Objectives	28
2.1.2 Related Works	29
2.1.3 Contribution	30
2.2 Wind Turbine Model	31
2.2.1 Wind Turbine Aerodynamics	32
2.2.2 Wind Turbine Mechanics	33
2.2.3 Wind Turbine Electrical Conversion	36
2.3 Classic Mode of Operation	38
2.3.1 Zones of Functioning	38
2.3.2 MPPT Control	39
2.3.3 Power Limiting Control	40
2.4 Deloaded Mode of Operation	41
2.4.1 Motivations for Deloading a Wind Turbine	41
2.4.2 Deloaded Control Strategies	44

2.5	Combined FL and MPC for Power Tracking	45
2.5.1	Control Problem Formulation	45
2.5.2	Feedback Linearization Step	46
2.5.3	Singular Points Analysis	49
2.5.4	Model Predictive Control Step	50
2.6	Wind Estimation	54
2.6.1	Aerodynamic Torque Estimation via Kalman Filter	55
2.6.2	Indirect Wind Estimation	55
2.7	Simulation Results	56
2.7.1	PI Control Comparison	57
2.7.2	Montecarlo Simulation	60
2.7.3	Deloaded Mode Scenarios	62
2.8	Conclusion and Future Perspectives	64
2.8.1	Conclusion	64
2.8.2	Future Perspectives	65
Part II WIND FARM HIERARCHICAL CONTROL		67
3	Novel Distributed Optimization Algorithms	69
3.1	Finite-Time Average Consensus	70
3.1.1	Problem Formulation	70
3.1.2	Self-configuration	72
3.2	Particle Swarm Optimization	73
3.2.1	Basic Algorithm	73
3.2.2	Avoiding Premature Convergence	77
3.2.3	Constraints Handling	78
3.3	Gradient-based Distributed Optimization	81
3.4	Distributed Cooperative PSO: DPSO1	84
3.4.1	Related Works and Contribution	84
3.4.2	Problem Statement	85
3.4.3	Algorithm Description	86
3.4.4	Enhancement of Self-configuration Step	92
3.4.5	Conclusion	94
3.5	Distributed Cooperative PSO with reduced communication: DPSO2	94
3.5.1	Related Works and Contribution	94
3.5.2	Problem Statement	96
3.5.3	Dynamic Behavioral Analysis	97
3.5.4	Algorithm Description	103
3.5.5	Benchmark Test	106
3.5.6	Constrained Optimization Extension	111
3.5.7	Conclusion	112
3.6	Optimal Power Flow Application	113
3.6.1	Related Works and Contribution	113

3.6.2	Power System Model	114
3.6.3	Distributed Optimal Power Flow	117
3.6.4	Conclusion	121
4	Wind Farm Control	125
4.1	Introduction	126
4.1.1	Problem Motivation	126
4.1.2	Related Works	127
4.1.3	Contribution	131
4.2	Wake Model	132
4.2.1	Modified Park Wake Model	133
4.2.2	Single Wake Model	134
4.2.3	Wind Farm Power Function	138
4.3	Hierarchical Control	139
4.3.1	High Level Optimization Problem	139
4.3.2	Yaw Angle Influence on the Optimization Problem	142
4.4	Distributed Optimization	146
4.4.1	Application of DPSO1	146
4.4.2	Application of DPSO2	152
4.5	Hierarchical Control Evaluation	162
4.5.1	Combining High Level Optimization and WT Controllers	162
4.5.2	A Wind Farm Example Test	163
4.6	Conclusion and Future Perspectives	165
4.6.1	Conclusion	165
4.6.2	Future Perspectives	166
Part III	WIND FARM DISTRIBUTED CONTROL	167
5	PID-like Consensus Protocol	169
5.1	Preliminaries on Consensus Control	169
5.1.1	Consensus Concept Introduction	169
5.1.2	Fundamental Consensus Algorithms	170
5.2	Discrete-time PID Consensus Control for LTI Systems	171
5.2.1	Related Works and Contribution	171
5.2.2	Leaderless Consensus Under the Presence of Disturbances	173
5.2.3	Leader-follower Consensus with Time-varying Reference State	180
5.2.4	Simulation Example	183
5.3	Conclusion	188
6	Wind Farm Consensus Control	189
6.1	Introduction	189
6.1.1	Problem Motivation	189
6.1.2	Related Works	190
6.1.3	Contribution	191

6.2	Problem Formulation	192
6.2.1	Wind Turbine Model	192
6.2.2	Problem Statement	193
6.3	Control Design	195
6.3.1	Wind Turbine Control for Deloaded Mode	195
6.3.2	Disturbance Effect and Additional Local Control Settings	198
6.3.3	Wind Farm Distributed Protocol	200
6.4	Simulations Results	202
6.5	Conclusion and Future Perspectives	209
6.5.1	Conclusion	209
6.5.2	Future Perspectives	209
GENERAL CONCLUSION AND PERSPECTIVES		211
APPENDICES		215
A	Basic Mathematical Notions and Definitions	217
A.1	Positive Definite Matrix	217
A.2	Lie Derivative	217
A.3	Global and Local Diffeomorphism	217
A.4	Feedback Linearization	218
A.4.1	Single-Input Single-Output Systems	218
A.4.2	Multi-Input Multi-Output Systems	221
A.4.3	Alternative Change of Coordinates	223
A.5	Positive Invariant Set and Local Stability	224
A.6	Model Predictive Control	224
A.6.1	Concept and Implementation	224
A.6.2	Nominal Stability Results	226
A.7	Mathematical Optimization	228
A.7.1	Convex Optimization	228
A.7.2	Nonlinear Optimization	229
A.8	Matrix Lemmas	229
A.9	Gershgorin's Disc Theorem	230
B	Additional Elements for Wind Turbines	231
B.1	dq Transformation	231
B.2	DFIG Working Principle and Electrical Model	232
B.3	Decoupling Active and Reactive Power	233
B.4	Residual due to Wind Disturbance	234
C	Graph Theory Preliminaries	237
C.1	Basic Notions and Definitions	237
C.2	Lemmas	237

D LMI-based Design	239
D.1 \mathcal{H}_∞ Norm of a System	239
D.2 Bounded Real Lemma	239
D.3 Robust Control for Parametric Polytopic Uncertainties	240
D.4 PID for MIMO LTI Discrete-time Systems	241
Bibliography	245

List of Figures

1	Worldwide cumulative installed wind capacity through the years. Source: GWEC.	4
2	Shared installed power capacity in EU in 2005 and 2016. Source: WindEurope.	4
3	Downward frequency regulation, where f_n is the nominal grid frequency. WFs may be required to deload active power P injection for frequency regulation.	7
1.1	State space region where relative degree of system (1.12) is well-defined, in both original and new coordinates.	25
1.2	Convex sets for MPC control, and closed-loop trajectory in both the original and new coordinates.	25
2.1	Wind turbine main components.	31
2.2	CART power coefficient.	32
2.3	Wind flow and wind speed at rotor disc $v_r = v(1 - \alpha)$	33
2.4	Two-mass model of the wind turbine mechanics.	34
2.5	DFIG-based wind turbine.	36
2.6	Sub-synchronous and super-synchronous modes of a DFIG. P_{mec} represents the mechanical power available at the generator. P_e is the electric power delivered to the grid. If DFIG losses are neglected, then $P_e = P_g$	37
2.7	Wind turbine zones of functioning and power coefficient function. . .	38
2.8	Power-rotor speed MPPT characteristic.	40
2.9	A PI-based classic control loop for MPPT/power limiting.	40
2.10	Power-rotor speed deloaded characteristic (solid-line) obtained via <i>overspeeding</i> the MPPT one (dash-dotted line).	43
2.11	Modified classic control scheme for deloaded mode of operation. . . .	44
2.12	Deloading strategy via pitch control.	44
2.13	$\Lambda_t \triangleq \{(\lambda, \vartheta) \in \mathbb{R}^2 \beta(\lambda, \vartheta) < 0\}$, <i>white</i> area, defines the set of points in which the WT relative degree is well-defined.	49
2.14	FL+MPC control scheme.	57
2.15	PI-based control for WT general power tracking.	58
2.16	Effective wind speed.	58
2.17	MPPT/power limiting and deloaded.	58
2.18	FL+MPC vs PI controlled power.	59
2.19	FL+MPC vs PI controlled rotor speed.	59
2.20	FL+MPC vs PI system inputs.	59
2.21	FL+MPC trajectories (<i>blue dots</i>) during the simulation. Thanks to the MPC constraints, these stay within Λ_t	60

2.22	Montecarlo simulation with respect to parameter uncertainties on the error on power and rotor speed reference. These have 99% of probability to lie within the grey area.	61
2.23	Zoom on Montecarlo simulation.	61
2.24	Real and effective estimated wind speed.	62
2.25	Power for constant and varying deloaded mode of functioning.	63
2.26	Rotor angular speed signals for constant and varying deloaded mode of functioning.	63
2.27	Controlled system inputs for constant and varying deloaded mode of functioning.	64
2.28	Surplus of stored kinetic energy for constant and varying deloaded mode of functioning.	64
2.29	System trajectories (blue dots). Thanks to the MPC constraints, these stay within Λ_t	65
3.1	An illustration of PSO classic dynamics of system (3.10).	76
3.2	Singly-linked list definition for $N_m = 2$. Each particle p gives access to its information to particles $p + 1$, and $p - 2$	78
3.3	Variables associated to the N agents of the network for DPSO1.	88
3.4	Physical graph \mathcal{G}_p (dash line), and communication graph \mathcal{G}_c (solid line).	89
3.5	Bucky communication graph.	93
3.6	Combined DPSO1 and backpropagation versus backpropagation.	93
3.7	CC-PSO with parallel update timing dynamic behavior.	100
3.8	Global bests trajectories for different values of damping factor β	103
3.9	Local best based context vectors (<i>dots</i>) for the x_1 component.	105
3.10	Plot of function (3.40).	106
3.11	Benchmark functions.	108
3.12	Mean convergence plots for the selected benchmark functions.	111
3.13	Circuits for power system modeling.	115
3.14	5-bus electric grid example.	119
3.15	Global best trajectories for the OPF problem during the run of DPSO1.	120
3.16	Global best trajectories for the OPF problem during the run of DPSO2.	121
3.17	Particles common cost function, and sum of constraints during iterations of DPSO1.	122
4.1	Wind flow and wind speed at rotor disc, $v_r = v(\cos o - \alpha)$, [Park 2015a].	133
4.2	Modified Park wake expansion model, [Park 2015a].	135
4.3	Wind farm reference frame.	135
4.4	Axial and radial wake distance, [Park 2015a].	136
4.5	Local polar coordinates, [Park 2015a].	137
4.6	Wind deficit as a function of d_{ij} and r_{ij} caused by WT j on WT i , when $o_j = 0, \alpha = \alpha_{betz}$	138
4.7	Hierarchical architectures.	140
4.8	6-WT wind farm example for yaw effect analysis.	142

4.9	Global best trajectories for $\theta^W = 0$ and low wind speed, and for both $\beta_o = 0$, and $\beta_o > 0$: convergence to p_1^*	143
4.10	Global best trajectories for $\theta^W = 0$ and high wind speed.	143
4.11	Global best trajectories for $\theta^W = \pi/2$ and low wind speed.	144
4.12	Global best trajectories for $\theta^W = \pi/2$ and high wind speed.	145
4.13	Horns Rev 1 wind farm layout.	148
4.14	Mean and standard values out of 20 trials of the number of suboptimal WTs, for $\theta^W = \frac{\pi}{4}$, and $v_\infty = 15$ m/s.	150
4.15	Generic WT i reduced communication graph \mathcal{G}_c for DPSO1, based on wake model approximation.	151
4.16	WT i neighborhood for DPSO2: WT_i^{up} in light gray area, WT_i^{down} in light blue area.	154
4.17	Horns Rev 1 communication graphs for $(v_\infty, \theta^W) = (7, \frac{\pi}{2})$	159
4.18	Global bests trajectories during DPSO2 iterations for Horns Rev 1, and $(v_\infty, \theta^W) = (7, \frac{\pi}{2})$	160
4.19	Global bests trajectories during DPSO2 iterations for Horns Rev 1, and $(v_\infty, \theta^W) = (15, \frac{\pi}{3})$	161
4.20	Wind farm layout and wind values.	163
4.21	Wind farm greedy vs cooperative performance.	164
4.22	Greedy vs cooperative power signals.	164
5.1	\mathcal{F}_D region in the complex plane defined via parameters (a, b)	179
5.2	5-agent communication graphs.	184
5.3	System output trajectories for each agent and network average for $\gamma = 0.9$, when $\omega_i(k) = 4\nu_i(k) + c_i$, $i = 1, \dots, 5$	184
5.4	\mathcal{H}_∞ tuning: system output trajectories for each agent and network average for $\gamma = 0.63$, when $\omega_i(k) = c_i$, $i = 1, \dots, 5$	184
5.5	Zoom on first system output component for $\gamma = 0.9$ and $\gamma = 0.63$	185
5.6	Fast consensus tuning: system output trajectories for each follower and leader for step response when $u_0 = 20$	186
5.7	\mathcal{H}_∞ tuning: system output trajectories for each follower and leader for step response when $u_0 = 20$	186
5.8	Fast consensus tuning: system output trajectories for each follower and leader for step response when u_0 is time-varying.	187
6.1	Singular points with and without β approximation.	197
6.2	WT control scheme: the local control is composed of an AOT step and of a PI; the distributed control has a PID structure.	201
6.3	Effective wind speed.	203
6.4	Wind turbine power and rotor angular speed for deloading mode.	203
6.5	Wind turbine inputs for deloading mode.	203
6.6	WF example communication graph.	204
6.7	Wind farm control with no distributed protocol.	204
6.8	Wind farm control with distributed PID.	205

6.9	Wind farm control with distributed multirate PID.	205
6.10	Wind disturbance.	207
6.11	Wind farm control with distributed PID.	207
6.12	Zoom on \mathcal{LDP} and DP during WF consensus control.	207
6.13	Wind farm leader-follower control.	208

List of Tables

2.1	CART Turbine Parameters	35
2.2	Simulation on FL+MPC vs PI	60
3.1	Algorithm Parameters	109
3.2	Test Results	110
3.3	DPSO1 Optimal Variables and Equality Constraints	122
3.4	DPSO2 Optimal Variables and Equality Constraints	122
4.1	Horns Rev 1 wind farm parameters	138
4.2	Yaw Influence Test Results	146
4.3	Horns Rev 1: cooperative vs greedy control	149
4.4	Horns Rev 1: original vs approximate wake model control	152
4.5	Minimum and maximum number of direct communications.	158
4.6	Optimal solution and power gain for different communication settings.	160
4.7	(C_{min}, C_{max}) for WFs of different size and $v_{\infty} = 15$ m/s, i.e. with active physical constraints.	161

List of Main Symbols and Acronyms

SYMBOLS

r	Relative degree	ϑ_r	Pitch actuator input
T_s	Sampling time	γ	Yaw angle
ρ	Air density	θ^W	Free stream wind direction
R	Rotor radius	v_∞	Free stream wind speed
v	Effective wind speed	o	Wind direction/WT orientation phase shift
$v_{cut_{in}}$	Minimum allowed wind speed	C_p	Power coefficient
$v_{cut_{out}}$	Maximum allowed wind speed	C_p^o	Maximum power coefficient
$ \dot{v} _{max}$	Maximum wind speed variation	C_q	Torque coefficient
ω_r	Rotor angular speed	α	Axial induction factor
ω_r^o	MPPT rotor angular speed	α_{betz}	Maximum axial induction factor
$\omega_{r,n}$	Nominal rotor angular speed	k_w	Wake expansion coefficient
$\omega_{r,min}$	Minimum rotor angular speed	δ_v	Wind deficit
$\omega_{r,max}$	Maximum rotor angular speed	T_r	Aerodynamic torque
ω_g	Generator angular speed	T_{ls}	Low-speed shaft torque
ω_{eg}	Generator electrical angular speed	T_{hs}	High-speed shaft torque
s	Generator slip	T_g	Generator torque
λ	Tip speed ratio	$T_{g,r}$	Torque actuator input
λ^o	MPPT tip speed ratio	T_{em}	Electromagnetic torque
ϑ	Pitch angle	P_r	Aerodynamic power
ϑ_{min}	Minimum pitch angle	P_g	Generator power
ϑ_{max}	Maximum pitch angle	P_e	Electric power
$ \dot{\vartheta} _{max}$	Maximum pitch variation	$P_{e,n}$	Nominal electric power
ϑ^o	MPPT pitch angle	P^o	MPPT power

P_{wf}	Wind farm power	C_{max}	Maximum direct communications
P_{wf}^{max}	Maximum wind farm power	Ψ_d	Wind direction distance parameter
δ	Shaft twist	Ψ_t	Transversal wind direction distance parameter
n_g	Gear box ratio	x_p	Particle p position
K_s	Spring constant	s_p	Particle p speed
D_s	Damping coefficient	b_p	Particle p personal best position
K_r	Rotor friction coefficient	l_p	Particle p local best position
K_g	Generator friction coefficient	g	Global best position
J_r	Rotor inertia	l_p	Particle p local best position
J_g	Generator inertia	N_p	Number of particles
J_t	One-mass low-shaft inertia	\mathcal{S}_p	Set of particle p neighbors
K_t	One-mass low-shaft friction	N_m	Number of neighbor particles
τ_T	Torque actuator time constant	ω	PSO inertia factor
τ_ϑ	Pitch actuator time constant	$\phi_{1,p}, \phi_{2,p}$	PSO aleatory variables
W_k	Kinetic energy	β	DPSO damping factor
\mathcal{G}	Graph	$\min\{\mathbf{x}\}$	Minimum value in \mathbf{x}
\mathcal{A}	Adjacency matrix	$\max\{\mathbf{x}\}$	Maximum value in \mathbf{x}
\mathcal{L}	Laplacian matrix	$\mathcal{R}(P)$	Range of matrix P
\mathcal{N}_i	Agent i set of neighbors	$\sigma(P)$	Spectrum of matrix P
h_i	Agent i state for average consensus	$ \mathcal{X} $	Cardinality of set \mathcal{X}
C_{min}	Minimum direct communications		

ACRONYMS

ADMM Alternating Direction Method of Multipliers

AEP Annual Energy Production

AOT Asymptotic Output Tracking

BMI Bilinear Matrix Inequality

CART Control Advanced Research Turbine

CC	Cooperative Co-evolution
CC-PSO	Cooperative Co-evolution Particle Swarm Optimization
CFD	Computational Fluid Dynamics
DFIG	Doubly Fed Induction Generator
DPSO	Distributed Particle Swarm Optimization
FL	Feedback Linearization
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MIMO	Multi Input Multi Output
MPC	Model Predictive Control
MPPT	Maximum Power Point Tracking
OPF	Optimal Power Flow
PID	Proportional Integral Derivative
PMSG	Permanent Magnets Synchronous Generator
PSF	Power Signal Feedback
PSO	Particle Swarm Optimization
RRT	Rapidly-exploring Random Tree
SCADA	Supervisory Control And Data Acquisition
SISO	Single Input Single Output
SOWFA	Simulation for Onshore/offshore Wind Farm Applications
TSR	Tip Speed Ratio
WAsP	Wind Atlas Analysis and Application Program
WF	Wind Farm
WT	Wind Turbine

List of Publications

Publications in Proceedings

N. Gionfra, H. Siguerdidjane, G. Sandou, D. Faille, and P. Loevenbruck, *A Distributed Consensus Control Under Disturbances for Wind Farm Power Maximization*, in Decision and Control (CDC), 2017 IEEE 56th Annual Conference on.

N. Gionfra, H. Siguerdidjane, G. Sandou, P. Loevenbruck, and D. Faille, *A Novel Distributed Particle Swarm Optimization Algorithm for the Optimal Power Flow problem*, Control Technology and Applications (CCTA), 2017 IEEE Conference on.

N. Gionfra, H. Siguerdidjane, G. Sandou, and D. Faille, *A Distributed PID-like Consensus Control for Discrete-time Multi-agent Systems*, in 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2017.

N. Gionfra, H. Siguerdidjane, G. Sandou, and D. Faille, *Hierarchical Control of a Wind Farm for Wake Interaction Minimization*, IFAC PapersOnLine, vol. 49, no. 27, pp. 330-335, 2016.

N. Gionfra, H. Siguerdidjane, G. Sandou, D. Faille, and P. Loevenbruck, *Combined Feedback Linearization and MPC for Wind Turbine Power Tracking*, in Control Applications (CCA), 2016 IEEE Conference on. IEEE, 2016, pp. 52-57.

Poster

N. Gionfra, H. Siguerdidjane, G. Sandou, D. Faille, and P. Loevenbruck, *Optimisation du Contrôle d'un Parc Éolien*, Vers la Transition Énergétique, WTE 2016, EDF, Université Paris-Saclay, Oct. 2016, poster. [Online]. Available: www.universite-paris-saclay.fr/sites/default/files/poster_nicolo_gionfra.pdf

Patent

N. Gionfra, D. Faille, P. Loevenbruck, G. Sandou, and H. Siguerdidjane, *Pilotage d'un Parc Éolien*, filed on the 31st of July 2017 by EDF (Electricité de France).

Communications in Conferences without Proceedings

N. Gionfra, G. Sandou, H. Siguerdidjane, D. Faille, and P. Loevenbruck, *Distributed Control and Optimization of a Wind Farm*, in 4ième Journée Scientifique de l'Institut RISEGrid, Gif-sur-Yvette, France, Nov. 2017.

N. Gionfra, G. Sandou, H. Siguerdidjane, D. Faille, and P. Loevenbruck, *Optimisation du Pilotage d'un Parc d'Énergies Renouvelables*, in 3ième Journée Scientifique de l'Institut RISEGrid, Palaiseau, France, Sep. 2016.

N. Gionfra, H. Siguerdidjane, G. Sandou, D. Faille, and P. Loevenbruck, *Advanced Control for Wind Turbine Grid Connection Requirements*, in World Congress and Exhibition on Wind & Renewable Energy, Berlin, Germany, Jul. 2016. [Online]. Available: <https://hal-centralesupelec.archives-ouvertes.fr/hal-01381643>

Submitted Chapter and Journal Article

N. Gionfra, G. Sandou, H. Siguerdidjane, D. Faille, and P. Loevenbruck, *A Discrete-Time PID-like Consensus Control: Application to the Wind Farm Distributed Control Problem*, Lectures Notes in Electrical Engineering (LNEE), Springer.

N. Gionfra, G. Sandou, H. Siguerdidjane, D. Faille, and P. Loevenbruck, *Wind Farm Distributed PSO-based Control for Constrained Power Generation Maximization*, Energy Conversion and Management.

GENERAL INTRODUCTION

IN recent years wind energy as an innovative electricity generation mean has had a fast-growing worldwide development, and reached high levels of installed capacity, led by China holding 42.8% of global wind power capacity in 2016. This growing trend is evidenced in Fig. 1 showing the evolution of wind power capacity per year, whose total amount reached 486.8 *GW* in 2016, 153.7 *GW* of which being installed in EU. Moreover, according to WindEurope in the EU, as shown in Fig. 2, wind energy represents today one of the leading means of energy generation, as it overtook coal-based energy generation becoming the second largest form of installed power capacity. It is worth mentioning that in many countries the electricity consumption is covered by wind energy in important percentage. An example is given by the case of Denmark where, according to the Global Wind Energy Council (GWEC), in 2016 the wind power penetration reached 37.6% of the total annual consumption. Moreover, the Danish government aims to get 50% of its electricity from wind source by 2020. Similar trends are followed by other European countries in order to meet the wind energy scenario objectives posed by Europe2020. Finally, wind energy exploitation falls in with the objectives recently set by the COP21. Indeed, according to GWEC, in 2016 wind power allowed avoiding over 637 million tons of CO₂ emissions globally.

Thesis Context

The aforementioned trends in the energy market push forward the development of adequate and innovative wind generation systems. This comes along with an extensive research work devoted to wind energy systems. In this work, wind farms (WFs) are the systems under analysis. According to the location and the electricity generation needs, wind farms can be systems of different sizes and corresponding capacity. For instance, offshore wind farms are usually composed of a high number of wind turbines (WTs), and they usually reach high levels of installed capacity. As an example, London Array wind farm in the United Kingdom is composed by 175 WTs with a corresponding total capacity of 630 *MW*. In this work we make the choice not to restrict our attention to a particular installed capacity, which would define the main system specifications and objectives. We rather proceed conversely, i.e. we aim at generality by sweeping the major issues concerning wind farms, which in turns can apply or not according to the specific case. Generally speaking, in a wind farm, two main stages of wind power conversion need to be considered. These are

- Wind to mechanical power conversion.

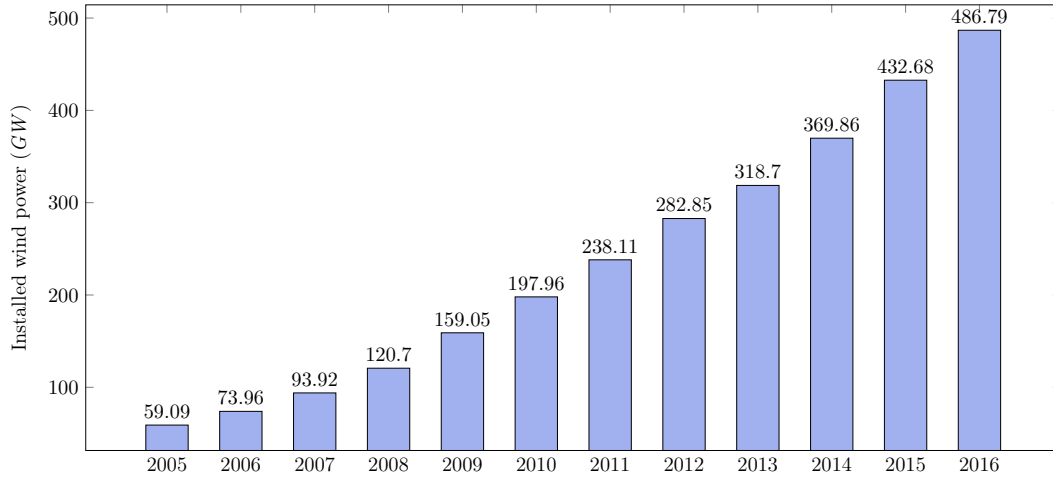


Figure 1: Worldwide cumulative installed wind capacity through the years. Source: GWEC.

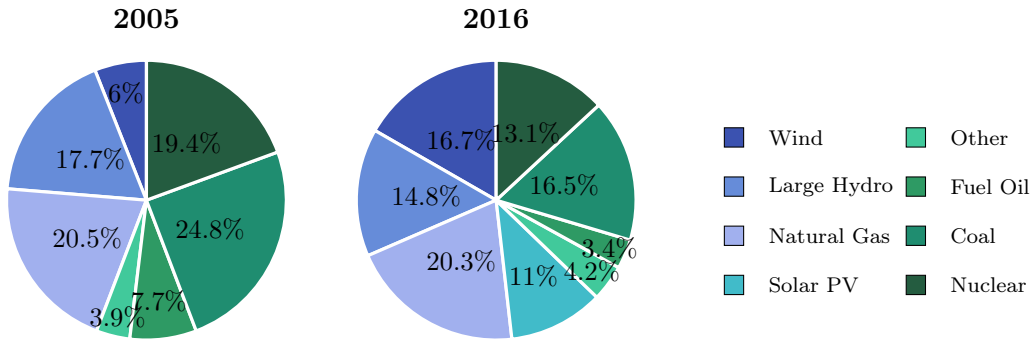


Figure 2: Shared installed power capacity in EU in 2005 and 2016. Source: WindEurope.

- Mechanical to electric power conversion.

Power conversion from wind to mechanical one is obtained via the WT's mechanical system. This is based on *horizontal axis* technology, [Burton 2011], while other WT types, such as vertical axis ones, are usually employed for small wind energy production at the consumer scale, [Bhutta 2012]. The rotor is the WT mechanical part responsible for capturing the wind kinetic energy. While there still exist wind farms based on fixed-speed WTs, the most recent and by far the most employed technology is concerned with *variable-speed* WTs, where the rotor can assume angular speed values within a given range, [Carlin 2003]. While allowing a much better wind source exploitation, more in general this leads to better versatility in accomplishing interesting new WT objectives which are the purpose of this work.

Mechanical to electric power conversion is obtained via the use of an electric generator. For the case of variable-speed technology, this is usually based on two solutions,

namely either on permanent magnets synchronous generators (PMSG) or doubly fed induction generators (DFIG), and this fully characterizes the mechanical to electric power conversion. In the DFIG-based WTs, the stator winding is directly connected to the grid, while the rotor ones are connected via a full power electronic converter. This solution allows a variation of the turbine rotor angular speed up to $\sim 30\%$ of its nominal value. Its main advantage is that the power converter only needs to be sized according to the nominal power flowing through the rotor winding and this reduces the costs. In the PMSG-based WTs, the power flows through the stator winding, which is connected to the electric grid via a full power electronic converter. While raising the cost of the power converter, this solution allows a full rotor angular speed variation, (e.g. [Baroudi 2007, Liserre 2011]).

At the wind farm level, its control is usually implemented on a two-layer *hierarchical* structure, e.g. [Hansen 2006]. At the lower level, wind turbine controllers are typically employed to let the WTs extract the *maximum power* available from the wind, and to switch to *power limiting* control whenever the latter exceeds the WT nominal one. The mentioned operating modes represent the state-of-the-art of implemented technology, and we will refer to them as *classic* operating mode throughout the whole work. Control at the higher level is implemented via SCADA (Supervisory Control and Data Acquisition) system, in a *centralized* fashion, [Crabtree 2014]. A central computer can communicate and receive information from the WF wind turbines. Data from all connected turbines are mainly employed for supervision duties. In particular, historical and real-time data can be used for data analytics, maintenance management, etc. Control duties are mainly concerned with starting and stopping selected WTs.

If in the past years, the wind farms were only required to inject the maximum power they could extract from the wind into the grid, nowadays the strong penetration of wind energy production has changed the role they have in taking into account power system stability issues. Power system stability refers to the ability of the electric grid to maintain nominal operating conditions under disturbances. In particular, stability is preserved by ensuring the balance between power production and demand, as well as maintaining a constant system frequency and voltage within acceptable limits. Disturbances can be of several nature, e.g. losses of power production, load changes, etc. It is well-known that the frequency of the electric grid is dependent on the active power balance and that a change in active power demand or production is reflected throughout the system by a change of frequency. It is also known how the control of reactive power is strongly related to the voltage stability and power losses minimization in transmission networks, (see e.g. [Kundur 2004]). Thus, for a correct and optimized power system operation, classic power plants connected to the grid are required to participate to voltage and frequency regulation by acting on the injected active power and on the injected or absorbed reactive power.

In the past years, these requirements were not expected to be fulfilled by wind energy plants. Moreover, they were required to disconnect under abnormal operating conditions. It is thus clear that under the new previously described context in which wind power penetration is reaching high percentage level of the electricity consump-

tion, in many countries wind farms have nowadays to meet new grid connection requirements defined in an adapted grid code. The main technical issues concerning wind farms connection to the grid are *frequency control*, *voltage and reactive power control*, and *fault ride-through capabilities*, [de Alegría 2007]. As in this work we are interested in drawing attention to the active power control, in the following we shall stress why frequency control is important in the case of high penetration of wind energy grid. First of all notice that in a system composed of synchronous generators directly connected to the grid, the cumulative system stiffness is high, and differences between produced and consumed active power generally have a small impact in the frequency deviation. In this case we say that the generators naturally contribute to the system inertia, [Ulbig 2015]. This is no longer the case if significant percentage of active power is provided by WFs, as they are electrically decoupled from the grid via power converters. This sets a first interesting new problem concerning frequency control in WFs. In particular, system inertia could be *artificially* produced by means of control, (e.g. [Morren 2006, Mauricio 2009]).

While system inertia effect should be continuously provided to the power system, if more severe frequency disturbances occur, then WFs may be required to also participate to primary control for frequency regulation. This involves the capability of WFs to either limit the active power injection when grid frequency is greater than its nominal value or, vice versa, to increase it when grid frequency is lower than its nominal value. Moreover such operation should be guaranteed for a lapse of time of 30 s. While *downward* frequency regulation, i.e. limiting the power injection, is generally always possible, *upward* frequency regulation requires additional power reserve. Thus, if WFs are required to fully participate to primary control, i.e. both downward and upward frequency regulation, then the WTs should be operated in full-time *deloaded* mode. In general, this is economically disadvantageous as it would require to permanently waste a portion of renewable power which could be instead delivered to the grid. As a result, typically upward frequency regulation is assigned to more classic power plants, and only downward frequency regulation is considered of interest for WFs. Thus, if required, WFs have to respect an active power injection limiting curve of the form of Fig. 3.

While the new grid code is pushing extensive research work in the WF control framework, this in turns lets a better understanding of the functioning of these systems, and it opens up to new interesting opportunities. This is the case of the *wake effect* causing the aerodynamic coupling among the WTs of a large WF or, more simply, groups of relatively closed WTs. The wake interaction can be described as the fact that when extracting kinetic energy from the wind, a WT causes a reduction of the wind speed in the downstream wake. As a result a turbine, standing in the wake of an upstream one, experiences a reduction of available wind power, [Park 2015a]. Since the functioning of a WT is affected by the one of the upstream turbines, in such situation a WF should no longer be considered as a simple composition of independent units. This sets interesting new challenges in WF control for a better exploitation of the wind source. As a matter of fact, when the wake effect is not negligible, considering it in the power maximization problem can lead

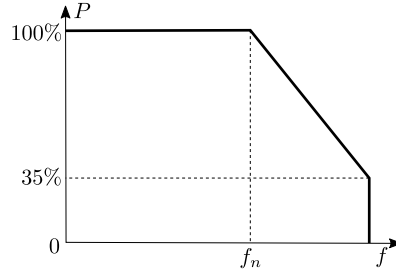


Figure 3: Downward frequency regulation, where f_n is the nominal grid frequency. WFs may be required to deload active power P injection for frequency regulation.

to important power gains with respect to the case in which each WT is operated in classic maximum power extraction mode. The main challenges are thus concerned with modeling the wake effect, which happens to be a non-trivial task due to its complexity, and to exploit this knowledge in efficient ways for optimized wind farm operations.

Contribution

In the previous discussion we mentioned some of the major issues and challenges concerning wind generation that are, and will be, of central interest in the next years. On the one hand, the new grid code sets wind farm control objectives that have to be fulfilled in order to ensure proper power system operations. On the other hand, the development of advanced control and optimization techniques has pushed the WF control targets further ahead towards a better exploitation of the renewable source.

From a WF control perspective the aforesaid objectives and opportunities can be treated as constraints and control specifications on the active and reactive power injected into the grid. Moreover, because PMSG and DFIG-based WT connection to the grid is obtained via power converters, under mild assumptions WTs are decoupled from the grid and, active and reactive power control can be performed independently one from the other, [Arifujjaman 2010]. While reactive power can be controlled by acting on the WTs power converters, active power control involves both the mechanical and electrical WT system. We make the choice to focus our research interest on what concerns the only active power control. Moreover, under the commonly employed assumption that the WT mechanical and electrical system can be treated by two decoupled loops of control, as they act on different time scales (e.g. [Boukhezzar 2006a]), we further target the WF mechanical system with a particular stress on the aerodynamics involved in the wind-to-mechanical power conversion. Throughout the whole work the considered approach is the one of translating specific control objectives in more general control specifications on the active power injection and wind power extraction. This is done in order to reach some

abstraction from the final application, and to provide general control techniques which could be applied to the addressed problem according to the situation. Having restricted the attention on the active power control, we can list the main WF applications in which the tools developed in this work can be used. These are

- Frequency control.
- Power maximization.
- Power curtailment.

Notice that we added *power curtailment* to the list of discussed WF objectives. This represents another important feature that WFs are more and more required to fulfill. In particular it falls within the connection requirement duties, and it consists in limiting the maximum power injection into the grid during limited lapses of time. Such constraint is usually imposed by the grid system operator in order to solve problems such as congestion, [Delille 2013].

The chosen WF control architecture has a two-layer *hierarchical* structure. Roughly speaking the higher control step is responsible for providing power references to the local WT controllers, which hold position at the lower control step. We tackle the WF control problem by firstly analyzing its single units, i.e. the wind turbines, and this is where we propose our first contribution. Moreover, we focus on nonlinear control techniques since, as evidenced by [Boukhezzar 2010], they outperform linear controllers, especially for high-turbulence wind profiles. Bearing in mind the control objectives at the WF level, in order to be accomplished, a WT needs to be able to operate in *deloaded* mode. This basically means that, whenever required, a WT has to track different power references from the classic one of maximum extractable power from the wind. However, most of the available nonlinear control approaches for wind turbine are conceived for either extracting the maximum power from the wind at low wind speed, (e.g. [Boukhezzar 2006b]), or for power limiting at high wind speed, (e.g. [Boukhezzar 2005, Boukhezzar 2007]). Moreover, with a few exceptions (e.g. [Burkart 2011]), the controllers are usually designed to work in a specific region of functioning defined by wind speed ranges. This is why we consider a particular control approach, which to the author's knowledge was never applied for WT control purposes, and that enables a WT to track a general deloaded power reference by means of both rotor angular speed and pitch angle control. Moreover it allows the WT to work in its whole operating envelope, i.e. it is not confined to specific WT zones of functioning. When the WT deloading mode has to be performed, among the available approaches (e.g. [Yingcheng 2011]), inspired by the work of [Zertek 2012b], our proposed control scheme allows to carry out a deloading technique which lets an optimized kinetic storage in the rotating masses.

Once performed the WT control design, the tools developed at the WT level have to be integrated in the more general framework of WF control. Thus we proceed by designing the higher control level of the considered hierarchical structure. Far from playing a mere supervisory role, this is responsible for letting an optimized

system operation as well as for the satisfaction of the considered grid connection requirements. Because a WF can easily reach a high number of units, and the high control level has to be reliable while being possibly required to work in near to real-time operations, we decide to analyze the control problem from a multi-agent system perspective. By doing so we aim at a *distributed* architecture thus enabling important features such as system scalability and modularity, as well as achieving a reduced communication and computational burden.

In this work, the WF control objectives are reformulated as a nonconvex optimization problem. Thus, in order to achieve the mentioned distributed characteristics, as well as to treat nonconvex optimization problems, we capitalize on distributed versions of well-known metaheuristic optimization techniques. The use of such optimization methods in the framework of multi-agent systems represents a fairly new application, and it has not been yet treated extensively. For instance, in the works of [Wakasa 2015a, Wakasa 2015b], even if the agents cooperate to solve a common optimization problem, these do not share a common optimization variable. As a consequence they cannot be applied to the WF optimization problem, in which the agents, i.e. the WTs, are coupled via the wake effect. Other works, such as [Gazi 2014, Navarro 2015], are concerned with distributed metaheuristic optimization techniques where the agents do share a common optimization variable, but they are designed for the specific problem addressed. Thus, before treating the WF problem, we first introduce two novel distributed optimization algorithms. Moreover, we present them in a general formulation allowing their application to a class of optimization problems. Indeed, as a supporting example, these are shown to be applicable to solve another interesting well-known nonconvex optimization problem, namely the optimal power flow one. In the literature, the main distributed approaches to solve it are either based on approximations to treat it via gradient-based techniques, as in [Sun 2013, Magnússon 2015], or on exact convex relaxation methods, as in [Peng 2014, Dall’Anese 2013], in which case their application is restricted to particular network topologies. By employing the proposed algorithms, the optimal power flow problem can be treated without the need for any restrictive assumption. Once discussed the considered optimization approach, we tackle the high level WF optimization problem. Here, we focus on the specific problem of maximizing the wind energy production, while respecting grid constraints on the injected active power in the grid. When it comes to maximize the WF power under wake effect in a distributed way, the works in the available literature capitalize on model-free approaches, such as [Gebraad 2016, Marden 2013, Park 2016], while model-based ones are usually treated via centralized methods, as in [Park 2015a, Herp 2015]. However, data-driven approaches may suffer from low speed of convergence as at each algorithm iteration the effect of the chosen point of functioning has to be tested on the real plant in order to be evaluated. Being interested in a distributed approach, as well as in speeding up the algorithm convergence to a solution, by using the developed optimization tools we are able to consider the wake effect within the WF model, and reach good convergence properties. Finally, the overall WF architecture is evaluated based on the developed tools for the WT level control. This allows

us to define a first important connection between the two layers of control, as well as showing the WF power gain obtained when the system dynamics is considered. Indeed, despite some exceptions, as in [Heer 2014], few works have evaluated the WF power gains by considering the whole WF control architecture, as it is often assumed that there exist local WT controllers capable of stabilizing the WT around any optimal set point, [Marden 2013].

The WF control architecture described so far is decentralized in the local WT controllers. Indeed their role is to, as their role is to let the WTs track the references provided by the higher control level, without any additional information exchange. In the last part of this report we claim that allowing additional cooperation at the WT level can lead to interesting benefits. Generally speaking we show that by means of consensus control techniques we can enhance the respect of the imposed higher level power gains despite the system dynamics and wind disturbances. This can be done by formulating the WF control problem as a power sharing one among the WTs, and on which they have to reach an agreement. Although similar power sharing problems in the WF framework have been proposed, as in [Biegel 2013, Spudić 2015, Baros 2017], both the problem addressed in this work and the control techniques to tackle it are fairly new, and to the author's knowledge they do not have an equivalent research work in the literature.

Thesis Organization

The document is divided into three parts. Each of these addresses a specific problem in the WF control framework and it is in turns divided in two chapters. The first chapter in each part is always related to the mathematical tools employed or developed in order to tackle the WF problem analyzed in the second chapter. By choosing this structure we also aim at providing a self-contained document as much as possible. This is why, especially in the mathematical tools chapters, for the reader convenience we sometimes recall well-known control and optimization techniques. More classic results are instead reported in the according appendix. The literature review is presented in details, locally, any time a new contribution is proposed.

Part I

This part is devoted to the wind turbine control problem, and it is composed of Chapters 1 and 2. The main issues addressed are the one of letting a WT operate in classic zones of functioning as well as track general deloaded power references. The considered control approach is based on a combined action of the well-known techniques of *feedback linearization* (FL) and *model predictive control* (MPC). Thus in Chapter 1 we recall some theoretical elements concerning the combined FL and MPC control technique, sometimes indicated as FL+MPC, while analyzing some additional issues concerning the treatment of system singular points, and nonconvex hard constraints. For the reader convenience, classic results concerning FL and MPC are reported in Appendix A.

In Chapter 2, the WT dynamic model and its controller are presented. Particular attention is devoted to its mechanical dynamics, being of central interest for the controller design. Both the classic and deloading modes of functioning are described. These are concerned with letting the WT extract the maximum available power as well as track more general power references according to the WF control problem. Moreover, when the WT has to be deloaded, the proposed control scheme enables a maximized kinetic energy storage in the WT rotating masses. In order to face well-known problems of wind estimation, we additionally make use of a Kalman filter. Results are presented on simulations based on NREL's CART¹ wind turbine parameters, and they show a good behavior in robustness and performance. The considered controller also proves to outperform a more classic WT control scheme.

Part II

In this second part, composed of Chapters 3 and 4, we analyze the control objectives at the WF level. As previously anticipated, the high level WF control problem is formulated as an optimization problem, which is chosen to be solved in a distributed way among the WTs. In Chapter 3 we thus aim at proposing two novel distributed optimization algorithms, which can be in turns used for the WF optimization problem. Both are based on the well-known *particle swarm optimization* (PSO) algorithm, whose main features are thus reported. As it often happens in the framework of distributed optimization, where the algorithms are usually coupled with consensus control techniques, the first of the two proposed algorithms is based on an existing finite time average consensus technique, proposed by [Dung 2013], whose main technical background is reported for the reader's convenience. The two introduced distributed PSO algorithms are described in a general way and they can be addressed to solve a class of optimization problems. Many engineering problems fall within the latter. As an example, we show that the proposed algorithms can be applied to the well-known nonconvex optimization problem of optimal power flow, without the need for particular problem approximations or simplifications.

In Chapter 4, the WF hierarchical control structure is presented. This is based on the WT control discussed in Chapter 1, and on a higher distributed controller coordinating the WF operations based on the solution of an optimization problem. Thus an important part of this chapter is dedicated to the definition and solution of the WF optimization problem. This is based on taking into account the wake effect into the WF model in order to aim at improved power gains, while satisfying grid constraints on injected active power. First, an existing wake model is presented, based on the work of [Park 2015a, Gebraad 2016], allowing to pose the optimization problem. As a matter of fact this belongs to the class of optimization problems which can be solved via the two proposed optimization algorithms of Chapter 3. Their performance is thus evaluated and compared on different wind farms examples. While both algorithms perform well in the unconstrained WF optimization problem, the

¹CART (Controls Advanced Research Turbine) is located at NREL National Wind Technology Center in Colorado, USA, and it is used for state-of-the-art test bed for controls research.

second proposed algorithm outperforms the first one in optimality of the solution for the constrained case. Moreover it presents good scalability properties when applied to large size WFs, while maintaining a competitive speed of convergence.

The optimization step is based on some simplified assumptions of perfect WF control operation, and in particular, the system dynamics are neglected. This is why, as a final step, we propose to evaluate the overall WF control performance by taking into account the WTs controlled dynamics. Simulations show that there exist a gap between the attainable power gains via the overall proposed WF architecture and the theoretical ones set by the only solution of the WF optimization problem. However still the obtained power gains can be significant and this confirms the interest for cooperative methods in controlling a WF when the wake effect is not negligible.

Part III

This last part is composed of Chapters 5 and 6, and it is dedicated to the introduction of a new control level in the hierarchical structure of Chapter 4 between the WF optimization and the WTs controllers. The main idea is to let further cooperation among the WT local controllers by letting them exchange information on a communication graph, which, for instance, can be set as the one used by the distributed optimization algorithms described in Chapter 4. Such control is based on the employment of consensus techniques. Thus, in Chapter 5 we introduce a new proportional-integral-derivative (PID)-like consensus control for multi-input-multi-output (MIMO) linear-time-invariant (LTI) agents. This is presented in a general formulation and can be applied for both leaderless and leader-follower consensus problems.

These tools are meant to be applied for the specific case of the WF distributed control problem in Chapter 6. However, since they are conceived for the linear dynamic systems framework, in Chapter 6 we first proceed by proposing an *asymptotic output tracking* (AOT)-based control scheme to let a WT track a general deloaded power reference, and make the WT closed-loop system linear. Thus the results concerning the WT control of Chapter 2 are not considered here, as the FL+MPC scheme would not allow an easy introduction of the controlled WT in the chosen WF consensus control framework, because of the presence of the MPC control step. Differently from the FL+MPC approach, the AOT WT controller allows a simpler deloading technique, which though does not let an optimized kinetic energy storage in the WT rotating masses. All in all, the proposed distributed controller can modify the high level optimal power references in order to take into account system dynamics and possible wind disturbances, and to enhance the overall system performance. In particular, on the one hand, via a leaderless approach the consensus controller scope is to even out zero-mean wind disturbances on the optimal WF power sharing. On the other hand, a leader-follower consensus controller enables to restore the optimal WF power sharing in the presence of power reference errors.

The document ends with some general conclusions and further perspectives, as well

as appendices and referenced bibliography.

Collaboration

This PhD work was carried out in collaboration with EDF (Electricité de France) R&D, and it was co-founded by CentraleSupélec and EDF.

It was prepared in a joint doctoral between the *Laboratoire des Signaux et Systèmes* (L2S) in CentraleSupélec, for the doctoral degree in *Sciences et Technologies de l'Information et de la Communication* (STIC), and the *Dipartimento di ingegneria Informatica Automatica e Gestionale Antonio Ruberti* (DIAG) in La Sapienza, Università di Roma, for the doctoral degree in *Automatica, Bioingegneria e Ricerca Operativa* (ABRO).

The EDF R&D departments involved in this research work are PRISME (*Performance, Risque Industriel, Surveillance pour la Maintenance et l'Exploitation*) and EFESE (*Economie Fonctionnement Etudes des Systèmes Energétiques*). Moreover, the PhD work is attached to the EDF project '*Eolienne terrestre*'.

The PhD ascribes to the RISEGrid Institute (Research Institute for Smarter Electric Grids) which was launched jointly by CentraleSupélec and EDF in December 2012 and is dedicated to the study, modelling and simulation of smart electric distribution grids and their interactions with the whole electric power system.

Part I

WIND TURBINE CONTROL

Combined Feedback Linearization and MPC

Contents

1.1	Constrained Feedback-linearizable Systems	18
1.1.1	Exact-linearizable Systems	18
1.1.2	Input-output Linearizable Systems	19
1.2	Avoiding Singular Points	20
1.3	Conclusion	26

THE aim of this chapter is to present the control approach that will be applied for the sake of WT control in Chapter 2. This is concerned with combining two well-known control techniques in cascade, namely *feedback linearization*, and *model predictive control*. For the reader's convenience we report the basic concepts and notions concerning FL, and MPC in Appendices A.4 and A.6 respectively. In the following we focus on two control problems that can be addressed by their combined action, namely the control of feedback linearizable systems with state and input constraints, and the control of feedback linearizable systems whose relative degree is only well-defined in subsets of the state space.

Generally speaking, the combination of FL and MPC is motivated by the following. When a nonlinear system presents state or input constraints, employment of the only FL technique cannot guarantee their satisfaction. This is why, thanks of its capability to handle system constraints, MPC can present a natural choice for a controller in cascade with the feedback linearizing one. In the same way, if MPC is employed to control a nonlinear system, then, in general, nonlinear optimization techniques should be considered in order to solve the receding horizon optimization problem, and this commonly leads to a substantial increase in the computation burden, [Nevistic 1995]. Thus FL enables solving a simplified, possibly convex optimization problem.

Remark 1.1 *In this work the considered FL step is conceived to control continuous-time dynamic systems. The feedback linearized system is then discretized in order to apply MPC technique. Another possible approach can be represented by conceiving the whole control design in the discrete-time framework. For a generalized definition of zero dynamics and the application of FL techniques to sampled-data systems, one may refer to the work of [Monaco 1988].*

1.1 Constrained Feedback-linearizable Systems

The following notes are based on [Nevistic 1995], who firstly proposed the FL+MPC control scheme. For the sake of simplicity, let us consider a feedback-linearizable SISO system affine in the input, i.e. of the form

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (1.1)$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, f is a smooth vector field, and g, h smooth functions. Results hold true for the MIMO case too. Moreover the system input has to satisfy the hard constraint

$$u_{min} \leq u \leq u_{max}$$

1.1.1 Exact-linearizable Systems

Let us first consider the system to have relative degree $r = n$, i.e. there is no zero dynamics. Thus, employing results of FL, in a neighborhood of a given point x_0 , there exists a local diffeomorphism $z = \Phi(x)$ (see Appendix A.3), and a static feedback control law

$$u = \frac{1}{a(z)}(-b(z) + v)$$

where $a(z)$, $b(z)$ are defined in Appendix A.4, yielding closed-loop system (1.1) in the new coordinates

$$\begin{cases} \dot{z} = Az + Bv \\ y = Cz \end{cases} \quad (1.2)$$

Let us consider x_0 to be an equilibrium point for system (1.1) and, without loss of generality $h(x_0) = 0$, and $z = 0$ at x_0 . Thus, stabilizing (1.2) in the origin implies the original coordinates system stabilization in x_0 . Instead of controlling (1.2) with a classic pole-placement controller, we employ MPC. For this purpose, system (1.2) is firstly discretized, yielding

$$\begin{cases} z^+ = A_d z + B_d v \\ y = C_d z \end{cases} \quad (1.3)$$

where $z \triangleq z(i)$ is the new coordinates state at step i , $v \triangleq v(i)$, $y \triangleq y(i)$ the input and output at current step i respectively, $z^+ \triangleq z(i+1)$, and A_d, B_d, C_d the discretized matrices of system (1.2). The optimization problem $\mathbb{P}_N(z)$ that MPC has to solve at each step i is thus of the form

$$\min_v \sum_{k=0}^{N-1} l(z(k), v(k)) + V_f(z(N)) \quad (1.4)$$

subject to $z^+ = A_d z + B_d v$

$$\begin{cases} a(z)u_{min} + b(z) \leq v \leq a(z)u_{max} + b(z) & \text{if } a(z) > 0 \\ a(z)u_{max} + b(z) \leq v \leq a(z)u_{min} + b(z) & \text{if } a(z) < 0 \end{cases} \quad (1.5)$$

where $\mathbf{v} \triangleq \{v(0), \dots, v(N-1)\}$. As shown in Appendix A.6, $\mathbb{P}_N(z)$ is a problem of parametric optimization, where z is the parameter. By choosing l to be *quadratic*, then $\mathbb{P}_N(z)$ generally describes an optimization problem with a quadratic cost function and a constraint that is nonlinear due to (1.5), which constitutes a *nonlinear state dependent constraint*. This is why in order to solve (1.4), (1.5), one would need *nonlinear programming* tools, while one of the motivations for considering the FL+MPC scheme in first place is to simplify the optimization problem, possibly avoiding the use of nonlinear optimization techniques. Moreover, if the functions appearing in (1.5) are nonconvex, then the optimization problem is generally nonconvex too. Nonetheless, it has to be stressed that problem $\mathbb{P}_N(z)$ substantially differs from the one that should be solved if MPC was applied to the original nonlinear control problem. Indeed, thanks to the FL stage, nonlinearities are *moved* from the cost function to the hard constraints of the optimization problem. In order to overcome the need for nonlinear programming though, authors of [Nevistic 1995] identify different implementation possibilities. These are grouped in *iterative procedures*, and *approximation procedures*. The former are concerned with dealing with nonlinear constraints exactly, and they are typically too computationally demanding. As far as the latter are concerned, in this work we only consider procedures of constraint function approximations. In particular we focus on *linear* approximations as, by consequence, the resulting receding horizon optimization problem is *quadratic*, and it can be solved with high computational performance.

Remark 1.2 *Note that if nonlinear constraints (1.5) are linearized at each time step, then the optimization problem is time-varying. Thus it is not straightforward to apply stability results of MPC, shown in Appendix A.6, to this particular case.*

Remark 1.3 *A first extension to the addressed problem is the case of systems with state constraints. If these are of polytopic form then, in the new coordinates, they are mapped into nonlinear, generally nonconvex constraints, via the employed diffeomorphism. As a result, they can be treated via approximation procedures as for the input state dependent ones.*

1.1.2 Input-output Linearizable Systems

When the system is only input-output linearizable, i.e. $r < n$, and the zero dynamics is stable, constraints (1.5) become more difficult to handle. This is due to the fact that in this case, system (1.1) is only *partially* feedback linearizable, yielding a system of the form (see Appendix A.4)

$$\begin{cases} \dot{\xi} &= A\xi + Bv \\ \dot{\eta} &= q(\xi, \eta) + \frac{p(\xi, \eta)}{a(\xi, \eta)}(-b(\xi, \eta) + v) \\ y &= C\xi \end{cases} \quad (1.6)$$

Thus functions a , and b depend on both ξ , and η variables, and the input constraints are

$$\begin{cases} a(\xi, \eta)u_{\min} + b(\xi, \eta) \leq v \leq a(\xi, \eta)u_{\max} + b(\xi, \eta) & \text{if } a(\xi, \eta) > 0 \\ a(\xi, \eta)u_{\max} + b(\xi, \eta) \leq v \leq a(\xi, \eta)u_{\min} + b(\xi, \eta) & \text{if } a(\xi, \eta) < 0 \end{cases} \quad (1.7)$$

where η evolves according to its nonlinear differential equation in (1.7). This can be discretized in order to be treated in the MPC framework, yielding a difference equation of the form

$$\eta^+ = f_\eta(\xi, \eta, v) \quad (1.8)$$

where $\eta^+ \triangleq \eta(i+1)$. As usual, in order to satisfy the constraints in the receding horizon, the system trajectories have to be *predicted* via the model equations, i.e. at each time step k , the state can be expressed as a function of the input sequence $\mathbf{v} = \{v(0), \dots, v(k-1)\}$ and the initial state, via $\phi(k, (\xi, \eta), \mathbf{v})$, where (ξ, η) is the initial state for problem (1.4), (1.7), and where $\phi(k, (\xi, \eta), \mathbf{v})$ is the system solution at time step k , given the initial condition (ξ, η) and the input sequence \mathbf{v} . In this case though, this involves nonlinear dynamics. Constraints as a function of the only input sequence \mathbf{v} are obtained via composition of the constraint functions with nonlinear functions $\phi(k, (z, \eta), \mathbf{v})$, $k = 1, \dots, N$. The overall constraints can be again handled via approximation procedures. Eventually, another possible way to approximate the optimization problem constraints is to *linearize* (1.8) in order to make the prediction $\phi(k, (z, \eta), \mathbf{v})$ linear in the input sequence, thus enabling less computation burden. Similar conclusions can be drawn for the case of state constraints.

1.2 Avoiding Singular Points

In this subsection, we propose to use the FL+MPC control scheme to treat those systems whose relative degree is not well-defined in the whole state space. Moreover, differently from the previous subsection, the systems under analysis are here assumed to be unconstrained. As it is known, the relative degree of a system is defined locally, in a neighborhood of a given point x_0 . In particular, if the relative degree at x_0 is r , then $L_g L_f^{r-1} h(x) \neq 0$ in a neighborhood of this point, where we made use of Lie derivative, defined in Appendix A.2. However, there might be points of the state space in which such condition is lost. In such points the relative degree is not well-defined and classic FL cannot be applied, [Zhang 2006]. These are the *singular points*. While in literature there exists approaches based on system approximation and switching to control a system near its singularities (e.g. [Tomlin 1998], [Zhang 2006]), in this subsection, by means of the combined FL and MPC technique, we aim at *confining* the system trajectories in the subspace where the relative degree is well-defined. This approach can be particularly useful when the functions appearing in the feedback linearizing law are too complex to be treated analytically in order to find a suitable system approximation.

We again address SISO systems for the sake of simplicity. Results hold true for the

MIMO case. Before getting insight into the proposed solution, it is useful to provide the following proposition.

Proposition 1.1 *Consider a SISO system of the form*

$$\begin{cases} \dot{x} = f(x) + g(x)u, & x(0) = x_0 \\ y = h(x) \end{cases} \quad (1.9)$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$. Then $L_g L_f^{r-1} h(x(t)) \neq 0 \quad \forall t \geq 0$, i.e. the state trajectory does not pass through singularities, if and only if the following conditions are verified.

(i) The system relative degree in x_0 is well-defined and equal to $r \leq n$.

(ii) $\text{sign}(L_g L_f^{r-1} h(x(t))) = \text{sign}(L_g L_f^{r-1} h(x_0)) \quad \forall t \geq 0$.

Proof: (Necessity) If for some $t^* > 0$, $\text{sign}(L_g L_f^{r-1} h(x(t^*))) \neq \text{sign}(L_g L_f^{r-1} h(x_0))$, then $\exists t^{**} : 0 < t^{**} \leq t^*$ such that $L_g L_f^{r-1} h(x(t^{**})) = 0$, i.e. $x(t^{**})$ is a singular point. (Sufficiency) If for some $t^* > 0$, $L_g L_f^{r-1} h(x(t^*)) = 0$ then, by condition (ii), $L_g L_f^{r-1} h(x_0) = 0$ and this contradicts condition (i). ■

Note that if the system trajectory is the solution of the closed-loop system where u is chosen as the FL control, i.e. $u = (L_g L_f^{r-1} h(x))^{-1}(-L_f^r h(x) + v)$ (see Appendix A.4), then, if the conditions of Proposition 1.1 are verified, u is finite. Proposition 1.1 gives simple conditions under which a given trajectory of the state does not pass through singular points. A natural question is under which conditions such trajectory exists. In particular we are interested in finding a trajectory connecting two given points while satisfying the conditions of Proposition 1.1.

Proposition 1.2 *Consider system (1.9), where x_0 respects condition (i) of Proposition 1.1. Consider a given output y_e and a corresponding equilibrium pair (x_e, u_e) , i.e. such that $0 = f(x_e) + g(x_e)u_e$, $h(x_e) = y_e$. If the following conditions are verified*

(i) The system relative degree in x_e is well-defined.

(ii) $\text{sign}(L_g L_f^{r-1} h(x_e)) = \text{sign}(L_g L_f^{r-1} h(x_0))$.

(iii) x_0, x_e belong to a connected subset of Λ : Λ_c , where Λ is defined as the set $\Lambda \triangleq \{x \in \mathcal{X} : \text{sign}(L_g L_f^{r-1} h(x)) = \text{sign}(L_g L_f^{r-1} h(x_0))\}$.

Then, by definition of connected set, it always exists a curve γ such that $x_0, x_e \in \gamma$ and $\gamma \in \Lambda_c$, which implies $L_g L_f^{r-1} h(x) \neq 0 \quad \forall x \in \gamma$.

Proposition 1.2 states that, under particular conditions, among all the possible curves connecting two given points, there always exists one which does not pass through singular points. Note that the conditions of Proposition 1.2 are *necessary* for the existence of a trajectory connecting two given points while avoiding singular points, i.e. a curve satisfying the conditions of Proposition 1.2 and system equations (1.9).

In the sequel, we provide stability results for the restricted case of *output regulation* of exactly linearizable systems, i.e. $r = n$. Thus, in the new coordinates, the discretized system is described by (1.3). Given an equilibrium pair (x_e, u_e) for system (1.9) corresponding to a given constant output y_e , without loss of generality, we consider that the equilibrium pair in the new coordinates satisfies $(z_e, v_e) = (0, 0)$ at (x_e, u_e) . Thus the problem of regulating the output at $y_e = Cz_e$ can be handled in the framework of the stabilization problem. In the more general case, the control input v of (1.3) is of the form

$$v = v_e + v_{mpc}$$

i.e. composed by the feedforward constant value v_e , which can be different from 0, corresponding to the steady state of the linearized system, and by the feedback control v_{mpc} stabilizing error $z - z_e$, and which in this framework is given by an MPC controller. Such practice is common in the output regulation problem, (see [Rawlings 2009]). By employing classic results of MPC stability, (see Appendix A.6), we can state the overall stability result in the following.

Theorem 1.1 *Given a system of the form of (1.9) with relative degree $r = n$; given an output target y_e , and an initial condition x_0 satisfying condition of Proposition 1.2; given the feedback linearizing change of coordinate $z = \Phi(x)$, such that $z_e = \Phi(x_e) = 0$, $z_0 = \Phi(x_0)$; if the following conditions are satisfied*

(i) *It exists a convex set $\tilde{\Lambda}_{con} \subseteq \tilde{\Lambda}_c$, where $\tilde{\Lambda}_c$ is a connected subset of*

$$\tilde{\Lambda} \triangleq \{z \in \mathbb{R}^n | \text{sign}(a(z)) = \text{sign}(a(0))\}$$

where $a(z) = L_g L_f^{n-1} h(\Phi^{-1}(z))$, and $z_0, z_e \in \tilde{\Lambda}_{con}$.

(ii) *$\tilde{\Lambda}_{con}$ is compact.*

(iii) *The diffeomorphism $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined in a region including $\tilde{\Lambda}_{con}$.*

(iv) *$l : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ is chosen to be a convex definite positive continuous function, and $l(z, u) \geq \alpha_1(|z|)$, where α_1 is a function of class \mathcal{K}_∞ .*

(v) *It exists a set $\mathbb{Z}_f \subseteq \tilde{\Lambda}_{con}$ such that*

$$\min_v \{V_f(A_d z + B_d v) + l(z, u) | A_d z + B_d v \in \mathbb{Z}_f\} \leq V_f(z) \quad \forall z \in \mathbb{Z}_f$$

where $V_f(z) \triangleq z^\top P z$, and where $P \succ 0$ is the unique solution to the discrete time Lyapunov equation $A_d^\top P A_d - A_d = -Q$, for a chosen $Q \succ 0$.

Then, the implicit control law $v_{mpc} \triangleq \kappa_N(z) \triangleq v_{mpc}^o(0, z)$, where $v_{mpc}^o(0, z)$ is the first value of the optimal sequence $\mathbf{v}_{mpc}^o(z) = \{v_{mpc}^o(0, z), \dots, v_{mpc}^o(N-1, z(N-1))\}$

solving the optimization problem $\mathbb{P}_N(z)$

$$\begin{aligned} V_N^o(z) &= \min_{\mathbf{v}} \sum_{k=0}^{N-1} l(z(k), v(k)) + V_f(z(N)) \\ \text{subject to } z^+ &= A_d z + B_d v \\ z(k) &\in \tilde{\Lambda}_{con}, \quad k = 0, \dots, N-1 \\ z(N) &\in \mathbb{Z}_f \end{aligned} \quad (1.10)$$

stabilizes system (1.3), if $z_0 \in \mathcal{Z}_N$, where \mathcal{Z}_N is the set of states z such that (z, \mathbf{v}) belongs to $\mathbb{Z}_N \triangleq \{(z, \mathbf{v}) | \phi(k, z, \mathbf{v}) \in \tilde{\Lambda}_{con}, k = 0, \dots, N-1, \phi(N, z, \mathbf{v}) \in \mathbb{Z}_f\}$. Thus,

$$u = \frac{1}{L_g L_f^{n-1} h(x)} (-L_f^n h(x) + \kappa_N(\Phi(x))) \quad (1.11)$$

stabilizes system (1.9) in x_e .

Proof: The proof is based on nominal MPC stability results, thus only the main sketches are presented. First of all the system is feedback linearized yielding discretized linear system (1.3). Under conditions (ii), (iv), (v), this is stabilized via the receding horizon control obtained by solving (1.10). This can be shown by employing $V_N^o(z)$ as a Lyapunov function for the closed-loop system $z^+ = A_d z + B_d v_{mpc}$, and prove Lyapunov conditions (A.20) of Theorem A.1 in Appendix A.5. In particular, as far as condition (i) of the Theorem A.1 is concerned, this is attained via condition (ii). Concerning condition (ii) of Theorem A.1, $V_f(z) \leq \bar{\lambda}(P)|z|^2$, $\forall z \in \mathbb{R}^n$, where $\bar{\lambda}(P)$ is the greatest eigenvalue of P . Because of condition (ii), then it exists a function α_2 of class \mathcal{K}_∞ such that $V_f(z) \leq \alpha_2(|z|)$, $\forall z \in \mathcal{Z}_N$, [Rawlings 2009]. Finally, condition (v) implies the descent property of $V_N^o(z)$. If $z_0 \in \mathcal{Z}_N$, then, for Weierstrass theorem, problem $\mathbb{P}_N(z_0)$ is solvable, and since \mathbb{Z}_f is control invariant, the problem is recursively feasible. Moreover, since the underlying dynamic system is linear, and both the constraints and the cost function are convex, then the sequence $\mathbf{v}_{mpc}^o(z)$ is globally optimal.

The overall control input is recovered in the original coordinates by composing the MPC control with the feedback linearizing one, thus yielding (1.11). Since the trajectories are confined in $\tilde{\Lambda}_{con}$, thanks to the MPC control and conditions of Proposition 1.2, for all time they belong to the subspace where the relative degree is well-defined. Eventually condition (iii) is required to have a well-defined change of coordinates in the region of interest. ■

Under the conditions of Theorem 1.1, it exists a control able to bring the system state from x_0 to x_e , and whose trajectories *avoid* the singular points. Moreover the theorem provides a feasible control law to solve the control problem. Note that, as far as the terminal cost function is concerned, matrix P in V_f does not have to be necessarily chosen as the solution of the Lyapunov discrete time equation. Often, P is chosen as the solution of the Riccati equation associated to the infinite horizon linear quadratic regulator (LQR) problem to control system (1.3).

Remark 1.4 *It has to be noted that if l is quadratic and, $\tilde{\Lambda}_{con}$ and \mathbb{Z}_f of Theorem 1.1 are polytopic, then (1.10) is a convex quadratic optimization problem. Thus, efficient quadratic programming algorithms can be employed to solve it with low computation burden.*

Remark 1.5 *The main idea of Theorem 1.1 is the one of finding a time-invariant convex set that allows classic MPC results to be employed to conclude on the closed-loop system stability. Indeed, as stated in Remark 1.2, if the nonlinear constraints are linearized at each time step, then the optimization problem is time-varying. This is why the key point of the theorem lies in the ability of constructing the convex set $\tilde{\Lambda}_{con}$. However such task is typically difficult, and there is no general method to perform it. Nonetheless, there are cases, especially when the system dimension is low, for which $\tilde{\Lambda}_{con}$ can be found. This is shown in the following example.*

Example 1.1 Consider the system

$$\begin{cases} \dot{x}_1 = f_1(x) \\ \dot{x}_2 = f_2(x) + g(x)u \\ y = x_1 \end{cases} \quad (1.12)$$

where $f_1(x) \triangleq 3 \sin x_1 + x_2$, $f_2(x) \triangleq x_1 x_2^2$, and

$$g(x) \triangleq \left(3(1 - x_1)^2 e^{-(x_1^2 + (1+x_2)^2)} - 10 \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) e^{-(x_1^2 + x_2^2)} \right)$$

Let us consider the problem of regulating the output at $y_e = 1$, starting from $x_0 = \text{col}(1/2, -5)$. System (1.12) has relative degree $r = 2$, thus it can be exactly linearized by employing the global diffeomorphism

$$z = \Phi(x) = \begin{pmatrix} x_1 \\ 3 \sin x_1 + x_2 \end{pmatrix}, \quad x = \Phi^{-1}(z) = \begin{pmatrix} z_1 \\ z_2 - 3 \sin z_1 \end{pmatrix}$$

The system in new coordinates is given by

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = 3z_2 \cos z_1 + z_1(z_2 - 3 \sin z_1)^2 + g(\Phi^{-1}(z))u \\ y = z_1 \end{cases}$$

Thus the feedback linearizing input is

$$u = \frac{1}{g(\Phi^{-1}(z))} \left(-3z_2 \cos z_1 - z_1(z_2 - 3 \sin z_1)^2 + v \right)$$

where v is left as the control to be set by the MPC controller.

The equilibrium pair (x_e, u_e) corresponding to $y_e = 1$ is $x_e = \text{col}(y_e, -3 \sin y_e)$, $u_e = -f_2(x_e)/g(x_e)$, where $g(x_e) < 0$. Moreover $g(x_0) < 0$, thus x_e, x_0 satisfy conditions (i), (ii) of Proposition 1.2, and we can define $\Lambda \triangleq \{x \in \mathbb{R}^2 | g(x) < 0\}$, which is shown in

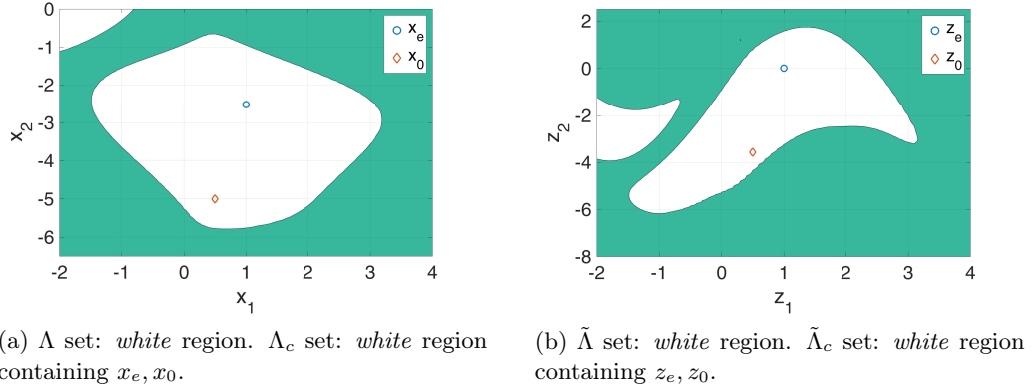


Figure 1.1: State space region where relative degree of system (1.12) is well-defined, in both original and new coordinates.

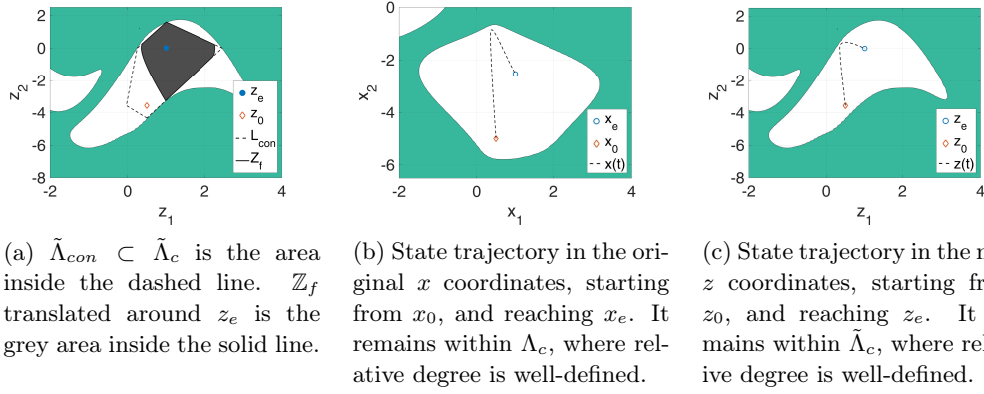


Figure 1.2: Convex sets for MPC control, and closed-loop trajectory in both the original and new coordinates.

Fig. 1.1a. Here we are also able to see that x_e, x_0 belong to a connected set $\Lambda_c \subset \Lambda$, thus satisfying condition (iii) of Proposition 1.2 too. In the new coordinates z , the set Λ is transformed in $\tilde{\Lambda} \triangleq \{z \in \mathbb{R}^2 | \tilde{g}(z) < 0\}$, where $\tilde{g}(z) \triangleq g(\Phi^{-1}(z))$. Similar transformation holds for Λ_c , which is named $\tilde{\Lambda}_c$ in the z coordinates. $\tilde{\Lambda}, \tilde{\Lambda}_c$ together with $z_e = \Phi(x_e)$, and $z_0 = \Phi(x_0)$ are shown in Fig. 1.1b. From $\tilde{\Lambda}_c$ we construct a polytope $\tilde{\Lambda}_{con} \subset \tilde{\Lambda}_c$, and such that $z_e, z_0 \in \tilde{\Lambda}_{con}$. The feedback linearized system is discretized via Euler method using a sampling time $T_s = 10$ ms. The system is thus governed by the linear difference equations $z^+ = A_d z + B_d v$, $y = C_d z$, where $A_d = I_2 + T_s A$, $B_d = T_s B$, being I_2 the identity matrix of dimension 2, A, B matrices in the continuous-time canonical form of controllability, and $C_d = [1 \ 0]$. The equilibrium pair (z_e, v_e) of such system, corresponding to $y_e = 1$, is $z_e = \text{col}(y_e, 0)$, and $v_e = 0$. Thus, by defining the change of coordinates $\tilde{z} \triangleq z - z_e$, and $\tilde{v} \triangleq v - v_e$, we can treat the output regulation problem in the stabilization framework. Concerning the receding horizon optimization problem yielding the MPC implicit control law,

we choose $l(\tilde{z}, \tilde{v}) = \tilde{z}^\top Q \tilde{z} + \tilde{v}^\top R \tilde{v}$, where $Q = 10I_2$, and $R = 1$, the terminal cost function $V_f(\tilde{z}) = \tilde{z}^\top P \tilde{z}$, where P is the solution of the Riccati equation associated to the infinite horizon LQR problem, and \mathbb{Z}_f is computed, using the toolbox provided in [Herceg 2013]¹, as the maximum control invariant set in $\tilde{\Lambda}_{con}$. The prediction horizon is set as $N = 5$. \mathbb{Z}_f translated around z_e , and $\tilde{\Lambda}_{con}$ are shown in Fig. 1.2a. The controlled trajectory using the described FL+MPC technique is shown in the original coordinate in Fig. 1.2b, and in the z variables in Fig. 1.2c. We are thus able to verify that it stays within Λ_c , ($\tilde{\Lambda}_c$) for all time.

1.3 Conclusion

We presented a brief introduction to the control technique that we aim to apply in the WT control framework. In particular we recalled how, on the one hand FL can be useful to treat nonlinear systems in the LTI framework and, on the other hand, how MPC technique can be employed to treat system constraints as well as system singular points. These two latter issues have been stressed as they appear in the WT application, object of Chapter 2.

Results of Theorem 1.1 have been presented mainly to make the reader aware of the particular singular points problem and to suggest MPC as a possible technique to handle it. In fact, these results are pretty conservative and may not find easy application to general real world control problems, as it will be shown in Chapter 2. However, in the author's opinion, in future research work less restrictive results could be obtained by using rapidly-exploring random tree (RRT) algorithms, see e.g. [Weiss 2017]. Indeed the two major limits of Theorem 1.1 are concerned with the existence of a convex set, in which the relative degree is well-defined, to which the system initial state x_0 and final state x_e belong, and on the computation of such set. If, more in general, x_0 , and x_e belong to a connected nonconvex set, then RRT-based algorithms can be used to compute a sequence of convex sets connecting x_0 to x_e , rather than requiring the existence of one convex set. The additional theoretical development concerning FL+MPC technique is considered beyond the scope of this work and will not be treated further.

¹Multi-Parametric Toolbox 3 (MPT3) is an open source, Matlab-based toolbox for parametric optimization, computational geometry and model predictive control.

Wind Turbine Controller Design

Contents

2.1	Introduction	28
2.1.1	Wind Turbine Control Objectives	28
2.1.2	Related Works	29
2.1.3	Contribution	30
2.2	Wind Turbine Model	31
2.2.1	Wind Turbine Aerodynamics	32
2.2.2	Wind Turbine Mechanics	33
2.2.2.1	Two-mass Model	33
2.2.2.2	One-mass Model	35
2.2.3	Wind Turbine Electrical Conversion	36
2.3	Classic Mode of Operation	38
2.3.1	Zones of Functioning	38
2.3.2	MPPT Control	39
2.3.3	Power Limiting Control	40
2.4	Deloaded Mode of Operation	41
2.4.1	Motivations for Deloading a Wind Turbine	41
2.4.1.1	Wind Farm Power Maximization	41
2.4.1.2	Grid Objectives	41
2.4.2	Deloaded Control Strategies	44
2.5	Combined FL and MPC for Power Tracking	45
2.5.1	Control Problem Formulation	45
2.5.2	Feedback Linearization Step	46
2.5.3	Singular Points Analysis	49
2.5.4	Model Predictive Control Step	50
2.6	Wind Estimation	54
2.6.1	Aerodynamic Torque Estimation via Kalman Filter	55
2.6.2	Indirect Wind Estimation	55
2.7	Simulation Results	56
2.7.1	PI Control Comparison	57
2.7.2	Montecarlo Simulation	60
2.7.3	Deloaded Mode Scenarios	62
2.8	Conclusion and Future Perspectives	64

2.8.1	Conclusion	64
2.8.2	Future Perspectives	65

2.1 Introduction

2.1.1 Wind Turbine Control Objectives

IN the field of control of wind turbines, two main regions corresponding to different classic operating modes are distinguished (see for instance [Ackermann 2005, Abdullah 2012]). The first one, at low wind speed, consists in the *maximum power point tracking* (MPPT) algorithm, while the second one, at high wind speed, is concerned with stabilizing the power at its nominal value. Generally speaking, the mentioned wind turbine control aims at converting the maximum extractable power from the upstream wind in electric power whenever the latter does not reach the WT nominal one, in which case the WT control is switched to *power limiting* mode. However, the strong penetration of wind energy production in the distribution and transmission electric grid has recently changed the role that wind farms have in taking into account the grid constraints as well as in ensuring the balance between production and demand. As a result, an adapted grid code for wind power generation establishes a set of grid connection technical requirements that wind farms have to meet. These are typically related to the voltage and reactive power control, frequency control, and fault ride-through capabilities (one may see [de Alegría 2007, ENTSO-E 2013]). If in the past years, wind turbines were not expected to actively participate to the grid operations, they are nowadays required to be able to work in nonconventional operating modes. On the one hand, this implies certain restrictions on the maximum power delivered to the grid, on the other hand, it opens a range of new possible services to which wind farms can participate, bringing an economical gain at the collectivity scale. This is evidenced by [Minaud 2013], and [Delille 2013], who, for instance, consider the economic advantage of power curtailment as a possible alternative solution to the grid reinforcement for the integration of renewable energies. Thus, it follows a growing interest in employing new technologies, such as energy storage, and new methods of control in order to allow the wind turbines to function out of the classic MPPT mode of operation when needed. From a technical point of view the aforementioned requirements can be expressed as constraints on active and reactive power that wind turbines have to respect while maximizing the energy production. Furthermore, recently, research studies on the aerodynamics involved in wind farms have shown that the optimal power reference for a wind turbine does not necessarily coincide with the MPPT one if coupling effects between turbines cannot be neglected. In particular, WTs would in general track *deloaded* power references with respect to MPPT in order to maximize the WF power production. This is treated in details in Part II, and Part III. Since a great deal of applications, especially concerning frequency response, active power

constraints, and power optimization, can be in practice treated as a general problem of *active* power control, we decided to focus the work of this part on the control of the aerodynamic power extraction of variable-speed variable-pitch turbines, i.e. the problem of tracking a desired power reference signal. Their technology is typically based either on doubly-fed induction generators or permanent magnet synchronous generators, and this characterizes the main electrical parameters of the considered WTs. Thanks to the power electronic converters employed in the connection of DFIG and PMSG WTs to the grid, the choice of focusing on the only active power control is validated by the possibility to control active and reactive power independently (see for instance [Poller 2003, Arifujjaman 2010, Perdana 2008, Hamon 2010]).

2.1.2 Related Works

The WT mechanical system is a nonlinear dynamic systems, mainly due to the aerodynamic relation describing the conversion from wind to mechanical power. For such system to be controlled in the classic operating mode, i.e. MPPT and power limiting, several linear control techniques have been proposed in the literature. However, as pointed out by [Boukhezzar 2010], nonlinear controllers can outperform linear ones, especially for high-turbulence wind speed. One can cite [Thomsen 2006, Boukhezzar 2005, Boukhezzar 2007, Boukhezzar 2011]. They all propose *feedback linearization* and *asymptotic output tracking* techniques to control a WT that differ one from another according to the tackled control problem. Authors of [Thomsen 2006] additionally provide a *sliding mode* application to WT control. In [Boukhezzar 2005], AOT is first used for monovariable control of a WT with fixed pitch angle, then, in [Boukhezzar 2007], the technique is coupled with a separate *proportional controller* to control the pitch angle and reach better performance. In both the latter references, the control is only applied at high wind speed values. In [Boukhezzar 2011], the authors capitalize on AOT with wind estimation to perform MPPT, in which case the pitch angle is fixed to its optimal constant value. The extension of these nonlinear control techniques to the more general active power control framework is not a trivial task. Moreover, these approaches are conceived for well-defined *modus operandi*, again either MPPT at low wind speed or power limiting at high wind speed. Few works have treated the problem of nonlinear control for the entire operating envelope. Authors of [Burkart 2011] present an approach based on *switched linear systems* and FL which enables the turbine to be controlled in all the regions of interest. Nevertheless, when operating at low wind speed, the pitch angle is kept constant to its optimal value. This basically limits its employment for different power references than the MPPT one. In [Gros 2013], the author employs an *economic nonlinear model predictive control* techniques to let the WT extract the maximum power from the wind without providing any pre-computed optimal steady state reference, and while satisfying the WT constraints, among which there is the rated power one. Thus, the controller can be applied in all classic operating modes. However, it was validated on only constant wind speed signals.

When controlling a WT in the classic mode of operation, typically, references for the turbine rotor angular speed and for the pitch angle need to be provided to the local controller. For a given wind speed value, these are usually obtained via the static aerodynamic relation between the mentioned variables and the aerodynamic power. When the desired aerodynamic power is lower than its maximum value, different set points for the rotor angular speed and the pitch angle must be provided. Even though different strategies have been proposed in the literature for the choice of these set points (e.g. [Yingcheng 2011, Žertek 2012a]), in the most cases the control architecture is based on standard linear controllers such as PID (e.g. [Loukarakis 2009, Ramtharan 2007, Tarnowski 2009]) and gain scheduling approaches (e.g. [Wang 2014, Khezami 2010, Camblong 2012]). Eventually, authors of [Penarrocha 2014] propose a nonlinear control to treat the specific problem of transient power increase with respect to the maximum aerodynamic power by means of machine deceleration to sustain the grid. The solution though applies for the case of low wind speed and fixed pitch angle.

2.1.3 Contribution

The problem of WT control is typically divided in two cascaded control loops. The first inner one is concerned with the WT electrical components, while the second outer one with the mechanical part of a WT. As they act on different time scales, their design can be performed independently [Boukhezzar 2010]. Thus, without loss of generality, in the sequel, we only address the problem of controlling the WT mechanical system. It follows that the results are applicable to both DFIG-based and PMSG-based WTs with variable pitch. As far as the electrical part control is concerned, for instance, conventional design for DFIG-based WTs is performed via rotor current vector control. The reader may refer to [Li 2009] for a comparative study of stator-voltage and stator-flux oriented control for DFIG-based WTs. In the following we present a nonlinear control for active power tracking which is not confined to work in a particular region of interest, i.e. no assumptions are made concerning the wind speed. Thus, it enables the WT to be controlled in its whole operating envelope. The employed control approach is based on a combination of FL and MPC, [Gionfra 2016b]. While MPC allows dealing with state and inputs constraints explicitly, FL enables solving an optimal control problem with nonlinear constraints and whose underlying dynamic system is made *linear* by the FL itself. In addition, as seen in the previous chapter, under some approximations, the optimal problem can be made convex, or even quadratic (see [Nevistic 1995]). This control approach applies to all the *deloading* techniques proposed by the references mentioned in the previous subsection. In order to guarantee the satisfaction of the chosen one, both pitch angle and rotor angular speed control is active in every operating zone to attain general power references. For this reason, the addressed problem falls within the framework of nonlinear MIMO systems. Eventually, we apply some of the results proposed in Chapter 1 to treat the *singular points* in the FL framework. As discussed, a singular point is a state of the system in which the

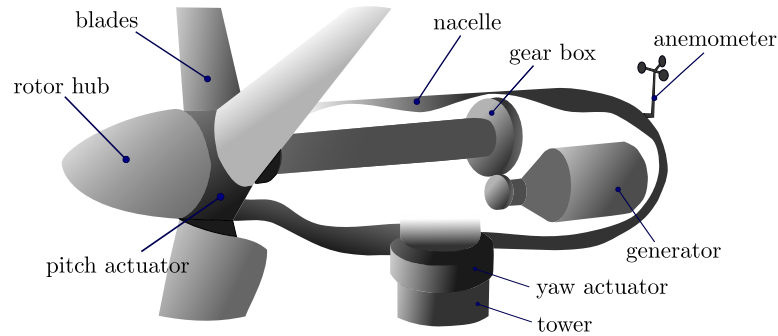


Figure 2.1: Wind turbine main components.

relative degree is not well-defined. If FL technique is employed for, say, a tracking problem, and the system presents singular points, then the classic formulation of FL may not be applied. While in the literature, approximate FL techniques, based on the work of [Hauser 1992], exist to overcome this problem, we propose to treat the wind turbine control problem via the approach based on *avoiding* the singular points rather than performing an approximate FL.

2.2 Wind Turbine Model

In this section we derive the mathematical model of a wind turbine, describing the conversion from wind power to electric power. We restrict our attention to variable-speed variable-pitch WTs, as nowadays they represent by far the most employed technology. In particular, as mentioned in the previous section, variable-speed turbines are based on electrical connection to the grid via either DFIG or PMSG. A WT is mainly composed by

- a rotor, in turns composed by a rotor hub, and rotor blades with their blade pitch actuator.
- a nacelle, in turns composed by a mechanical transmission, generally equipped with a gear box, by a generator responsible for the mechanical to electrical power conversion, and typically by an anemometer for wind measurement.
- a tower on which the nacelle is placed. Variable-yaw WTs are also equipped with a yaw drive.

The overall WT components scheme is shown in Fig. 2.1. The WT model can be expressed as the cascade of three main modules, each of which describing

- the *aerodynamics*, i.e. the conversion from wind power to mechanical power.
- the *mechanics*, i.e. the transmission and adaptation of the mechanical power to be converted in electrical one.

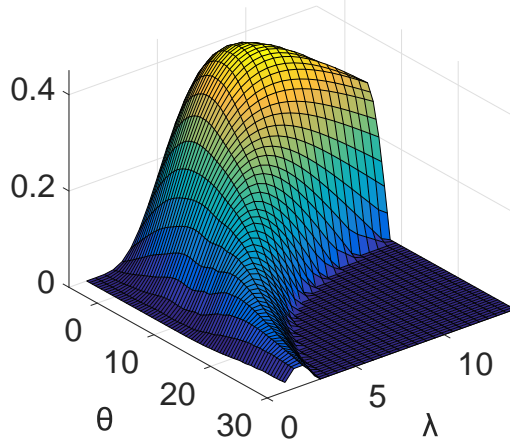


Figure 2.2: CART power coefficient.

- the *electrical conversion* of the mechanical power.

2.2.1 Wind Turbine Aerodynamics

The wind kinetic energy captured by the turbine is turned into mechanical energy of the turbine rotor, turning at an angular speed ω_r and subject to a torque T_r . In order to model the power extracted from the wind, it has been shown that considering the wind speed component in the only longitudinal (axial) direction guarantees sufficient model reliability. Moreover, from the actual wind speed w , the equivalent wind speed v representing the wind field impact on the turbine can be obtained by filtering the time series of wind data w as described by [Petru 2002]. The extracted power can be described by the nonlinear function

$$P_r = \omega_r T_r = \frac{1}{2} \rho \pi R^2 v^3 C_p(\lambda, \vartheta) \quad (2.1)$$

where ρ is the air density, R is the radius of the rotor blades, ϑ is the pitch angle, λ is the *tip speed ratio* given by

$$\lambda = \frac{\omega_r R}{v}$$

C_p is named *power coefficient*, and it describes the ratio between the extracted and the available wind power as a function of the tip speed ratio and pitch angle. This is typically provided in the turbine specifications as a lookup table. As far as the turbine parameters are concerned, in this work we make use of the CART power coefficient, shown in Fig. 2.2. However, often C_p is approximated by an analytic function in order to be treated for control synthesis purpose. Typically it can be described either by a polynomial function of the form

$$C_p(\lambda, \vartheta) \simeq \sum_{i=0}^{n_{c,1}} \sum_{j=0}^{n_{c,2}} a_{ij} \lambda^i \vartheta^j$$

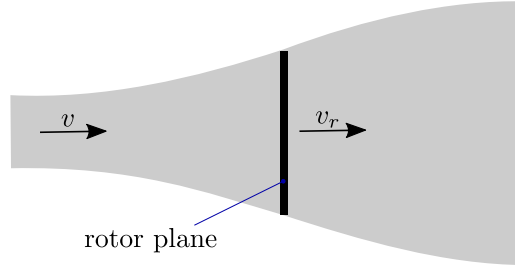


Figure 2.3: Wind flow and wind speed at rotor disc $v_r = v(1 - \alpha)$.

where $n_{c,1} + n_{c,2}$ is the polynomial degree, and a_{ij} its coefficients, or by the following (see [Heier 2014])

$$C_p(\lambda, \vartheta) \simeq 0.5 \left(\frac{116}{\lambda_i} - 0.4\vartheta - 5 \right) e^{\left(\frac{-21}{\lambda_i} \right)} + 0.0068\lambda \quad (2.2)$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\vartheta} - \frac{0.035}{\vartheta^3 + 1}$$

Moreover, C_p can be differently described by employing Betz theory. In particular, we first define the *axial induction factor* α as

$$\alpha \triangleq \frac{v - v_r}{v}$$

where v is the free stream wind blowing in front of the WT, and v_r the wind speed at the rotor plane, see Fig. 2.3. Thus α is the fractional decrease in wind velocity between the free stream and the wind turbine, and it is an index of the amount of kinetic energy captured by the WT. It can be proved (see e.g. [Burton 2011]) that theoretically

$$C_p = 4\alpha(1 - \alpha)^2 \quad (2.3)$$

From (2.3), it is clear that the maximum power coefficient has a theoretical limit equal to $16/27 = \max_{\alpha} C_p(\alpha)$, attained at $\alpha = 1/3$, which we rename here α_{betz} .

2.2.2 Wind Turbine Mechanics

2.2.2.1 Two-mass Model

A drive train turns the slow rotor speed into high speed on the generator side, ω_g . Because of its huge dimension with respect to the other components, the rotor can be model as one mass, [Boukhezzer 2006a]. As a result, the drive train can be naturally described by a *two-mass model* approximation (e.g. [Boukhezzer 2006a, Thomsen 2006, Boukhezzer 2011, Burkart 2011]). This is shown in Fig. 2.4, where the rotor, the gear box, and the slow speed shaft inertia are integrated in J_r . The high speed shaft inertia associated to the generator is represented by J_g . All the flexible modes of the WT can be considered to be localized in the low speed shaft, [Boukhezzer 2006a]. Thus, the twisting of the rotor, gear box, and low speed shaft

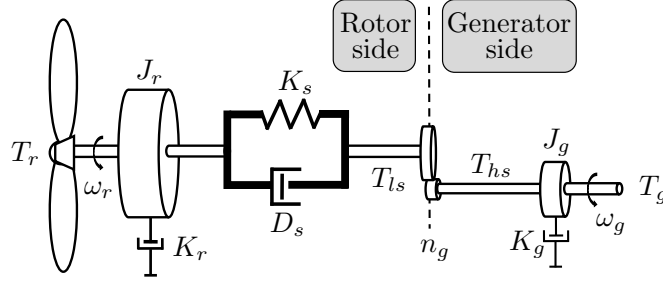


Figure 2.4: Two-mass model of the wind turbine mechanics.

are integrated in the spring constant K_s , while D_s is the associated damping coefficient. The gear box ratio is n_g . Eventually the two masses frictions are considered via coefficients K_r , and K_g .

The rotor dynamics is given by

$$J_r \dot{\omega}_r = T_r - T_{ls} - K_r \omega_r \quad (2.4)$$

where the expression for T_r can be easily obtained from (2.1). The friction and twisting effects caused by the difference between the rotor angular position ϑ_r , and the low speed shaft one ϑ_g/n_g , and between the corresponding angular speed ω_r , and ω_g/n_g , result in the low speed shaft torque T_{ls} . Thus, by naming $\delta \triangleq \vartheta_r - \vartheta_g/n_g$, T_{ls} is given by

$$T_{ls} = K_s \delta + D_s \dot{\delta}$$

The corresponding high speed shaft torque is obtained via n_g as

$$T_{hs} = \frac{T_{ls}}{n_g} \quad (2.5)$$

As a result, the generator side dynamics is given by

$$J_g \dot{\omega}_g = T_{hs} - K_g \omega_g - T_g \quad (2.6)$$

where T_g is the electromagnetic torque of the generator. The overall WT dynamics is given by

$$\begin{cases} J_r \dot{\omega}_r &= T_r - D_s \left(\omega_r - \frac{\omega_g}{n_g} \right) - K_r \omega_r - K_s \delta \\ J_g \dot{\omega}_g &= \frac{D_s}{n_g} \left(\omega_r - \frac{\omega_g}{n_g} \right) - K_g \omega_g + \frac{K_s}{n_g} \delta - T_g \\ \dot{\delta} &= \omega_r - \frac{\omega_g}{n_g} \\ P_g &= T_g \omega_g \end{cases} \quad (2.7)$$

where P_g is the electric power delivered to the grid, if the electrical components losses are neglected. WT equations (2.7) describe a MIMO system, which can be controlled by acting on its inputs, namely the electromagnetic torque T_g , and the pitch angle ϑ , which appears in the expression of T_r via (2.1). Eventually, as mentioned, in

Table 2.1: CART Turbine Parameters

Parameter	Value	Units
R	21.65	m
n_g	43.165	—
K_s	269.1	kN · m/rad
D_s	9500	N · m/rad/s
K_s	269.1	kN · m/rad
K_r	27.36	kN · m/rad
K_g	0.2	kN · m/rad
J_r	$3.25 \cdot 10^5$	kg · m ²
J_g	34.4	kg · m ²
$\omega_{r,n}$	4.3982	rad/s
$\omega_{r,max}$	5.5501	rad/s
ϑ_{max}	30	deg
ϑ_{min}	−5	deg
$ \dot{\vartheta} _{max}$	19	deg/s
$T_{r,max}$	162	kN · m
$P_{e,n}$	600	kW

this chapter we make use of CART turbine parameters. This turbine has a nominal power $P_{e,n}$ equal to 600 kW. The other turbine parameters are reported in Table 2.1, [Boukhezzar 2006a].

2.2.2.2 One-mass Model

When modeling wind turbines of big size, the low speed shaft of the drive train can be considered as perfectly rigid, i.e. its flexible modes can be neglected, yielding $\omega_r = \omega_g/n_g$, [Boukhezzar 2006a]. In this case, the WT dynamics can be described via a *one-mass model*. Indeed, we can rewrite the two-mass dynamics from (2.4) and (2.6) as

$$J_r \dot{\omega}_r = T_r - n_g T_{hs} - K_r \omega_r \quad (2.8)$$

$$n_g J_g \dot{\omega}_r = T_{hs} - n_g K_g \omega_r - T_g \quad (2.9)$$

where we employed (2.5) in (2.8). If we now multiply (2.9) by n_g , and sum (2.8) and (2.9), we obtain the one-mass model

$$\begin{cases} J_t \dot{\omega}_r &= T_r - K_t \omega_r - T_{em} \\ P_g &= T_{em} \omega_r \end{cases} \quad (2.10)$$

where

$$\begin{aligned} J_t &\triangleq J_r + n_g^2 J_g \\ K_t &\triangleq K_r + n_g^2 K_g \\ T_{em} &\triangleq n_g T_g \end{aligned}$$

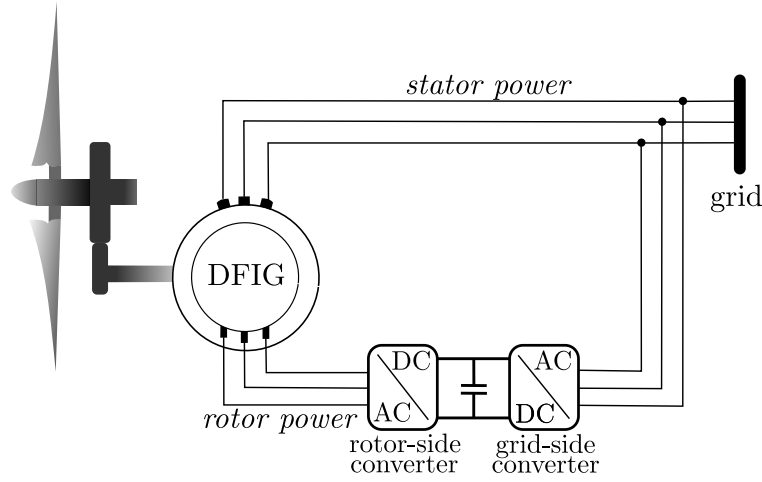


Figure 2.5: DFIG-based wind turbine.

System (2.10) describes WT dynamics whose variables and parameters are *brought* on the low speed shaft side. Similar computation has to be carried out if the system dynamics is to be considered on the high speed shaft side.

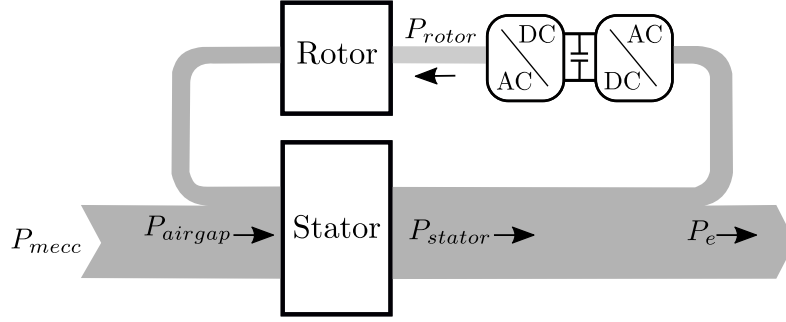
2.2.3 Wind Turbine Electrical Conversion

The main purpose of this subsection is to provide some basic notions regarding the electrical power conversion of a wind turbine. We limit our analysis to the only DFIG-based WTs, as the CART turbine considered in this work belongs to this WT technology. The following notes are based on [Poller 2003, Arifujjaman 2010, Hamon 2010].

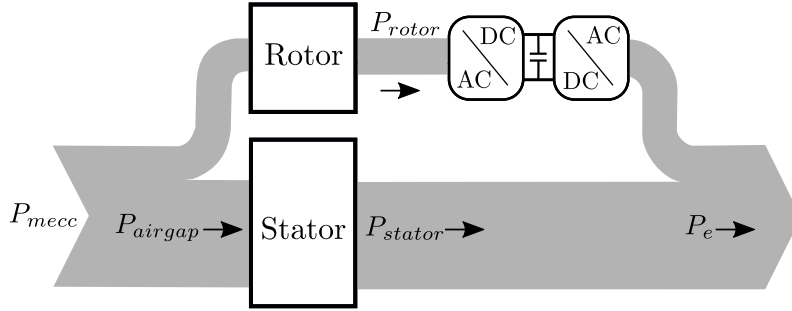
DFIG is based on asynchronous electric machine. In the DFIG configuration, the stator windings are directly connected to the grid, which thus imposes its currents frequency f_s , while the rotor ones are connected through two power electronic converters, namely the *rotor-side*, and *grid-side* converter. A schematic of the above description is given in Fig. 2.5. The stator currents create a rotating magnetic field in the air gap between the stator and the rotor. Being an asynchronous machine, according to Faraday's law, in order to induce currents in the rotor windings, it is required to have an electrical angular speed ω_{eg} *different* from $\omega_s = 2\pi f_s$. ω_{eg} is linked to the generator mechanical one ω_g via the machine number of poles p , $\omega_{eg} = p\omega_g$. By defining the slip s as

$$s = \frac{\omega_s - \omega_{eg}}{\omega_s}$$

the DFIG operating modes can be divided in two, namely the *sub-synchronous* mode, for *positive* s values, and the *super-synchronous* one, for *negative* s values.



(a) Sub-synchronous operating mode.



(b) Super-synchronous operating mode.

Figure 2.6: Sub-synchronous and super-synchronous modes of a DFIG. P_{mec} represents the mechanical power available at the generator. P_e is the electric power delivered to the grid. If DFIG losses are neglected, then $P_e = P_g$.

When the slip is negative, an asynchronous machine works as a *generator*, and as a *motor* when the slip is positive. In the DFIG configuration, to enable generation also when s is positive, power has to be injected in the rotor via the power converters. Such power is collected from the grid and re-introduced in the stator through the air gap, thus creating a closed-loop power flow between the stator and the rotor. Sub-synchronous and super-synchronous modes are illustrated in Fig. 2.6. All in all the electric power P_e delivered to the grid is given by $P_e = P_{stator} + P_{rotor}$, i.e. the sum of the power flowing through the stator windings, P_{stator} , and through the rotor one, P_{rotor} . As a matter of fact this can also be expressed as

$$P_e = (1 - s)P_{stator}$$

This means that the transferred power portion flowing through the rotor can be expressed as function of the stator one via $P_{rotor} = -sP_{stator}$, and thus, it can be controlled acting on s . For the reader's convenience we show this fact, together with the differential equations and the electrical variables involved in the DFIG-based electric machine in Appendix B.2.

In this work we only address the problem of controlling the WT active power, under the assumption that active and reactive power control can be performed independ-

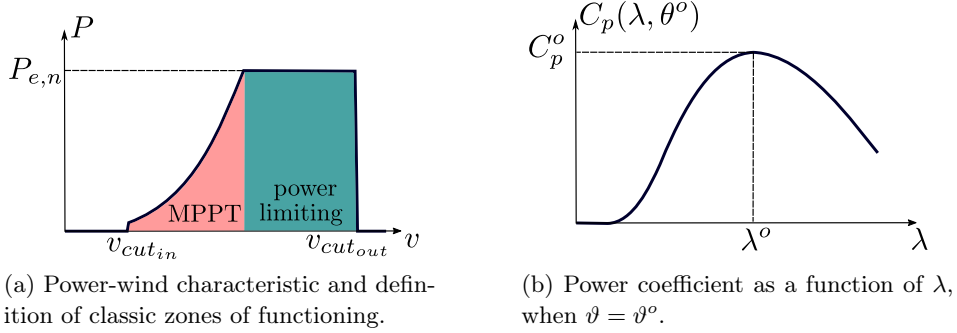


Figure 2.7: Wind turbine zones of functioning and power coefficient function.

ently, [Arifujjaman 2010]. For the reader's convenience some notes supporting this assumption are reported in Appendix B.3. Moreover, in general, active and reactive control is performed via inner control loops that are fast with respect to the mechanic system dynamics. Being interested in the active power control, and focusing on the only WT mechanical system, in the sequel we consider a first order system to model the DFIG system, actuating the electromagnetic torque T_g of system (2.7). This takes the form

$$\tau_T \dot{T}_g = T_g + T_{g,r} \quad (2.11)$$

where the time constant τ_T depicts the speed of the inner control loop, (see e.g. [Boukhezzer 2009]).

2.3 Classic Mode of Operation

2.3.1 Zones of Functioning

In *classic* WT control, wind turbines are conceived to work in two main zones of functioning. These can be described as a function of wind speed v blowing in the axial direction of the WT. At low wind speed and above a minimum value $v_{cut_{in}}$, a WT is operated in MPPT mode, i.e. it is required to track the maximum power extractable from the wind

$$P^o(v) = \frac{1}{2} \rho \pi R^2 v^3 C_p^o$$

where C_p^o is the maximum value of the C_p function shown in Fig. 2.2, and for which, as mentioned, there exists a theoretical limit equal to $16/27$. MPPT typically occurs in the range of wind speed values $\sim [5, 12]$ m/s. At high wind speed and below a safe maximum wind speed value $v_{cut_{out}}$, when the extractable wind power is above the WT rated one, $P_{e,n}$, a WT is operated in *power limiting* mode, i.e. the extracted power is regulated at $P_{e,n}$. $v_{cut_{out}}$ has a typical value equal to ~ 22 m/s. The described zones of functioning are depicted in Fig. 2.7a, where it is shown the extracted power as a function of the wind speed. In the sequel we show some classic control schemes associated to the MPPT and power limiting mode.

2.3.2 MPPT Control

In the literature there exist several classic control schemes to perform MPPT control, (see e.g. [Barakati 2009, Abdullah 2012]). A first well-known technique is the *Tip Speed Ratio* (TSR) control, and its main concept is described as follows. The wind turbine power coefficient $C_p(\lambda, \vartheta)$ has its maximum at constant ϑ for any value of λ . This optimal pitch value ϑ^o is typically equal to 0° . In Fig. 2.7b the C_p function of λ when $\vartheta = \vartheta^o$ is illustrated. From this, it is easy to identify

$$\lambda^o = \arg \max_{\lambda} C_p(\lambda, \vartheta^o)$$

Thus, MPPT mode is concerned with controlling the WT at constant tip speed ratio equal to λ^o . In order to do so, a local control could be conceived to let the WT track the rotor angular speed, function of v

$$\omega_r^o(v) = \frac{\lambda^o}{R} v \quad (2.12)$$

while keeping the pitch angle at its constant optimal value ϑ^o . Because (2.12) requires knowledge of the wind speed, and because it is often preferable to have direct control on the power injected into the grid, typically other control schemes are considered. Among these, a well-known technique is the *Power Signal Feedback* (PSF) control method. This is based on direct employment of a lookup table relating the MPPT power P^o with ω_r^o in order to generate a power reference for a local controller. Such lookup table is obtained via (2.1), and the expression of C_p . Indeed, for each value of wind speed, it is possible to trace a curve of the extracted wind power as function of the rotor angular speed via (2.1), for $\vartheta = \vartheta^o$. Connecting the maximum points of such curves yields the optimal power as function of ω_r^o . This is shown in Fig. 2.8. The MPPT control loop is thus obtained as illustrated in Fig. 2.9a. From the measure of ω_r , an optimal power reference P^o is obtained via the aforementioned lookup table. Then, a controller, typically PI-based, is used to control the power error $P^o - P_g$ to zero, by acting on the system input $T_{g,r}$. Recall that $T_{g,r}$ has been introduced in the previous section to model the DFIG inner control response. Thus, in this control configuration, the electromagnetic torque is used to control the power, while in the TSR scheme, the torque is devoted to the rotor angular speed control. Sometimes, a similar MPPT control is employed by defining a lookup table relating the optimal torque with ω_r , i.e. the torque guaranteeing the maximum power extraction from the wind. As a result, in a similar way to what is shown above, a torque reference is provided to a local controller.

Remark 2.1 *MPPT modus operandi corresponds to a specific optimal choice concerning the rotor angular speed and the pitch angle, namely ω_r^o and ϑ^o . However, it can be sometimes convenient to define MPPT with respect to the axial induction factor variable α . Since the value α_{betz} , defined in Section 2.2, corresponds to the maximum value of C_p , controlling the WT axial induction factor at α_{betz} implies performing the MPPT mode of functioning. This definition turns out to be useful when treating the WF control problem, and it will be shown in Part II.*

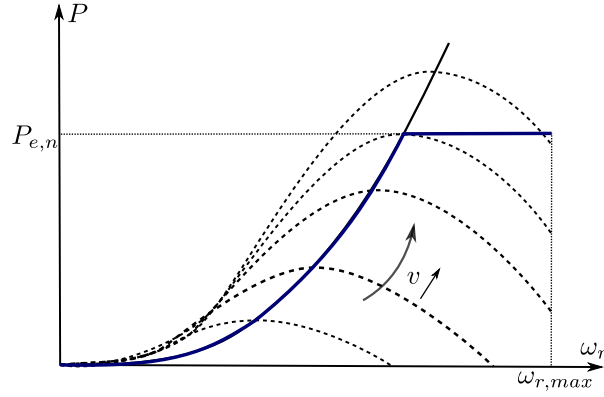


Figure 2.8: Power-rotor speed MPPT characteristic.

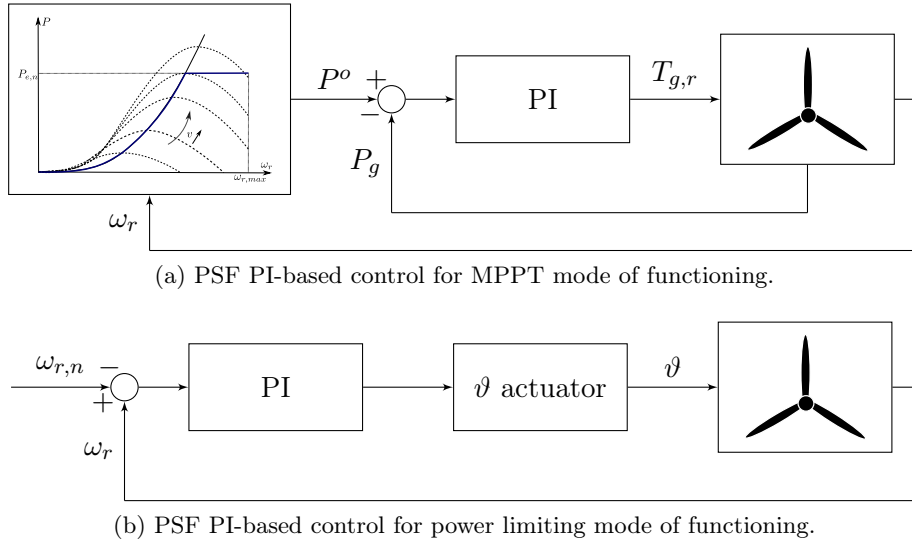


Figure 2.9: A PI-based classic control loop for MPPT/power limiting.

2.3.3 Power Limiting Control

When the extractable wind power reaches the WT nominal one, the wind turbine is controlled in power limiting mode. This is achieved by enabling the pitch control. Typically, if in the MPPT control scheme the torque is employed to regulate the power, then pitch control is used to keep the rotor angular speed at its nominal value $\omega_{r,n}$, [Venne 2009]. This is the case of PSF control, for which power limiting at high wind speed is usually obtained by activating a PI-based control scheme as shown in Fig. 2.9b, while torque control continues to regulate the power at its nominal value. Vice versa, if MPPT is performed by regulating the rotor angular speed via electromagnetic torque, then at high wind speed the pitch angle is used to regulate the power at its nominal value.

Remark 2.2 *Although the WT is a MIMO system, classic WT control is concerned*

with two separate control loops. Moreover, during MPPT mode, the system is treated as a SISO system, as the pitch angle is kept constant. Only when the power has to be limited, the additional pitch control is activated. As it is shown in next sections, this control approach may be no longer applied for a more general power tracking problem.

2.4 Deloaded Mode of Operation

2.4.1 Motivations for Deloading a Wind Turbine

The main objective of MPPT and power limiting is to let a WT extract the maximum available power from the wind while satisfying the nominal WT power constraint. There exist though some important scenarios for which it would be either preferable or even compulsory to track other power references than the classic ones, and for which control approaches of Section 2.2 cannot be considered. The scenarios contemplated in this work can be grouped in two main control objectives

- Wind farm power maximization.
- Satisfaction of grid objectives.

2.4.1.1 Wind Farm Power Maximization

When power maximization concerns a wind farm whose WTs are aerodynamically coupled, the optimal power production is no longer obtained via the MPPT algorithm. Instead, optimal set points need to be computed in a *cooperative* manner. This problem is exhaustively treated in Part II, and III, and we do not provide here further details. However, at WT level, this results in the need for making the WTs track general *deloaded* power references.

2.4.1.2 Grid Objectives

As anticipated in the General Introduction, WFs can be demanded to meet new grid code specifications. Some of these result in particular requirements concerning the *active power* injected into the grid. Among these, one can list the following

- Power curtailment and power smoothness requirements.
- Frequency support requirements.

Power Curtailment and Power Smoothness

By *power curtailment* we identify a general power constraint to be satisfied by a wind farm, or wind turbine. This is considered imposed by the system operator to solve a grid constraint (e.g. congestion). Such constraint generally takes the form of a maximum power deliverable to the grid for a given lapse of time. However, more in general, it can be considered as a set of constraints concerning other characteristics

of the power generated by the WTs, such as the smoothness of the power curve. When these power specifications are operational, an interesting additional objective can be considered. Indeed, at wind farm level, particular set points could be imposed to the WTs in order to *reduce the mechanical stress* of the whole system structure by *fairly* distribute the power production among the WTs, (e.g. [Baros 2017]).

Frequency Support

In classic power plants, the system frequency inertia is naturally provided to the grid by the synchronous generators directly connected to it, as they exhibit a dynamic behavior that can be described by the following simplified *swing* equation

$$J\dot{\omega}_s = T_m - T_e \quad (2.13)$$

where J represents the generator inertia, $\omega_s = 2\pi f_s$, f_s being the grid frequency, T_m , and T_e are respectively the mechanical and electrical torque. From (2.13) it is clear that the frequency of the electric grid is dependent on the active power balance, and that a change in active power demand or production is reflected throughout the system by a change of frequency. However, if the cumulative inertial effect of the synchronous generators connected to the grid is high, then the change of frequency is limited. In Section 2.2, we mentioned how DFIG-based WTs are electrically decoupled from the grid thanks to the power converters. This holds true for general variable-speed WTs, basically due to their capability to vary the rotor angular speed while being connected to the grid via the employment of power converters. Thus, this is also the case of PMSG-based WTs. Following from the above discussion on system inertia, this electrical decoupling prevents the WTs from providing any natural contribution to the grid frequency stability, i.e. contributing to maintain a constant system frequency within acceptable limits. This sets a first interesting problem, namely the one of *artificially* producing the system inertia by means of control. This is essentially done by adding an auxiliary power reference ΔP_{ref} to the one of the control scheme of Fig. 2.9a, which is generally a linear combination of frequency deviation $\Delta f = f_{ref} - f_s$, where f_{ref} is the system frequency in normal operation ($f_{ref} = 50$ Hz in Europe), and frequency deviation time derivative $\frac{d\Delta f}{dt}$, (see e.g. [Morren 2006, Mauricio 2009, Zertek 2012b]), according to

$$\Delta P_{ref} = -K_1 \frac{d\Delta f}{dt} - K_2 \Delta f \quad (2.14)$$

where $K_1, K_2 \in \mathbb{R}^+$ are parameters to be tuned. The additional power reference (2.14) combines the attempt of producing an artificial inertial response, i.e. first term on the right side, and the classic *droop* control, i.e. second term on the right side, typically used for primary frequency control, (see [Kundur 2004]). Since (2.14) is added to the classic power reference described in Section 2.3, which is the maximum extractable from the wind, the defined inertial control can present some shortcomings. First of all, at high wind speed, when the classic power reference is the WT nominal one, the WT can only support a *downward* additional power

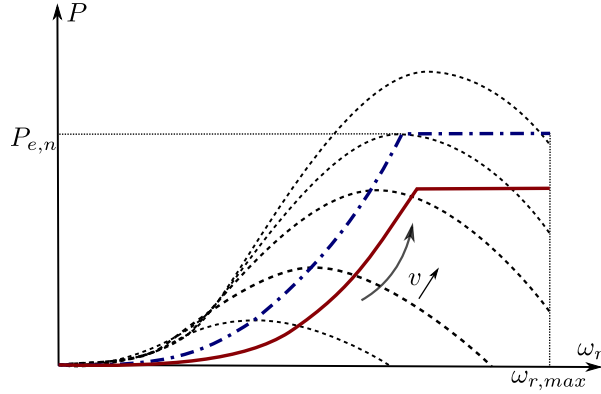


Figure 2.10: Power-rotor speed deloaded characteristic (solid-line) obtained via *over-speeding* the MPPT one (dash-dotted line).

reference, i.e. $\Delta P_{ref} \leq 0$. Secondly, when in MPPT mode of functioning, a positive ΔP_{ref} makes the WT get additional power from the kinetic energy of the rotating masses. This results in a deceleration of the rotor angular speed, thus making this additional power reserve available only for short-term operations, [Yingcheng 2011]. A second important problem concerning the grid frequency support is represented by the primary frequency control. In order to provide a primary regulation in both senses, i.e. *downward* when $\Delta P_{ref} \leq 0$, and *upward* when $\Delta P_{ref} \geq 0$, it has been proposed to control WTs permanently in deloaded mode, so that a *power reserve* is all time available. To do so, in the literature it is suggested to deload the optimal power-rotor speed characteristic of Fig. 2.8 of a chosen percentage, (e.g. [De Almeida 2007, Yingcheng 2011, Žertek 2012a, Zertek 2012b]). For any value of wind speed, and constant $\vartheta = \vartheta^o$, this can be done by either *under-speeding*, i.e. $\omega_r < \omega_r^o$, or by *over-speeding* the rotor angular speed, i.e. $\omega_r > \omega_r^o$. However, under-speeding is disadvantageous because, in order to increase the *electric* power output P_g , if $\Delta P_{ref} \geq 0$, the kinetic energy of the WT must be increased first to bring ω_r back to ω_r^o , [Loukarakis 2009]. Thus part of the wind power is first employed to increase the kinetic energy of the rotating masses, and the *mechanical* power. If for sake of simplicity we consider the wind turbine one-mass model of equation (2.10), we can see that if ω_r has to be increased, then a mechanical and electric power imbalance needs to occur. In particular the electric power injected into the grid needs to be firstly reduced to let ω_r increase and this delays the required increase in P_g . This is why over-speeding is preferred. The deloaded power-rotor speed characteristic is shown in Fig. 2.10.

Remark 2.3 *The described frequency support control is advantageous if one wants to make little adjustments to the control scheme of Fig. 2.9, used in classic mode of functioning. Indeed, these consist in simply changing the lookup table of Fig. 2.8 with the deloaded one of Fig. 2.10, and adding (2.14) to the power reference signal, (e.g. [De Almeida 2007, Ramtharan 2007]). The modified control scheme is shown*

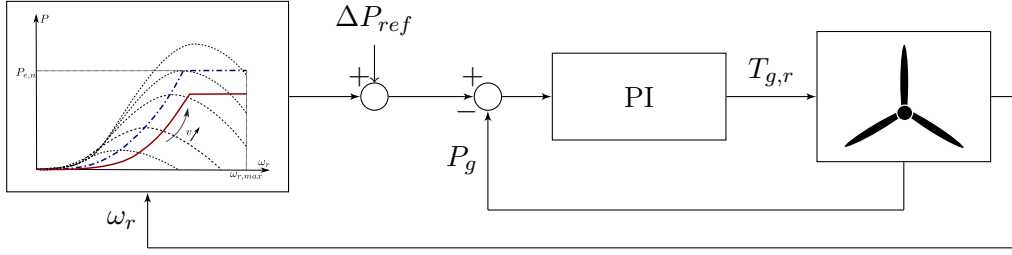


Figure 2.11: Modified classic control scheme for deloaded mode of operation.

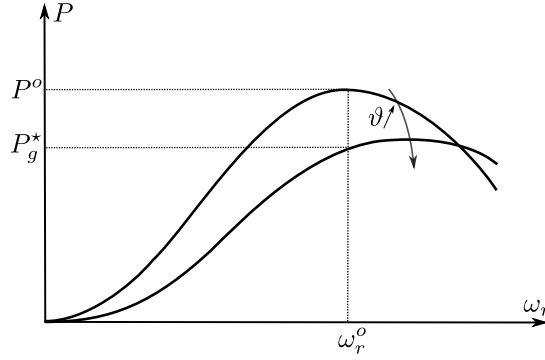


Figure 2.12: Deloading strategy via pitch control.

in Fig. 2.11.

2.4.2 Deloaded Control Strategies

In the sequel, we focus on the problem of deloading a WT of a general quantity that may also vary in time. This can be treated as the problem of making a WT track a general deloaded power reference, i.e. satisfying

$$0 \leq P_g(t) \leq \min(P^o, P_{e,n}) \quad \forall t \geq 0 \quad (2.15)$$

and it allows us to answer to both the control objectives of wind farm power maximization and grid support, seen in the previous subsection. Since according to power equation (2.1), for a given $P_g^* \leq \min(P^o, P_{e,n})$, the choice of (ω^*, ϑ^*) that yields P_g^* is not unique, there exist different strategies to deload a WT. These are typically based either on *pitch control* or *speed control*, [Yingcheng 2011]. The former consists in keeping ω_r at its optimal MPPT value ω_r^o , and modifying the pitch angle to move on other power-rotor speed curves. This principle is illustrated in Fig. 2.12. As far as the latter is concerned, we saw its working principle in the previous subsection. However, as mentioned in Remark 2.3, this is useful when a classic control scheme is preserved, and where the pitch control is only activated at high wind speed. More in general, the only speed control may not guarantee the required power deloading.

This is why we capitalize on the deloading strategy proposed by [Zertek 2012b] that, via a combined choice of both rotor speed and pitch angle, allows the WT to work at an optimal operating point with respect to the amount of kinetic energy stored in the rotating masses. In particular, when deloading needs to be performed the set point (ω^*, ϑ^*) are computed using

$$\begin{aligned} (\omega_r^*, \vartheta^*) &= \arg \max \omega_r \\ \text{subject to} \\ P_g^* &= P_r(\omega_r, \vartheta, v) \\ \omega_{r,min} &\leq \omega_r \leq \omega_{r,n} \\ \vartheta_{min} &\leq \vartheta \leq \vartheta_{max} \end{aligned} \quad (2.16)$$

This approach seems to be preferable because, if the WT has to be deloaded, part of the mechanical power P_r can be used to increase the rotor speed. As a result, part of the undelivered energy to the grid is stored in the rotating masses, yielding a surplus of stored kinetic energy ΔW_k with respect to pitch deloading control, given by

$$\Delta W_k = \frac{1}{2} J_t (\omega_r^{*2} - \omega_r^{o2}) \quad (2.17)$$

where, for sake of simplicity, we used the one-mass model. This surplus of kinetic energy could be in turns used for frequency control if required, as it represents a spinning reserve.

2.5 Combined FL and MPC for Power Tracking

2.5.1 Control Problem Formulation

According to the motivations discussed in Section 2.4, we address the problem of tracking a general power reference satisfying (2.15). The classic control architectures of Fig. 2.9 and Fig. 2.11 fail to be adequate for such purpose because, generally, both speed and pitch control have to be active in the whole WT operating envelope, i.e. at both high and low wind speed.

Let us consider the WT two-mass model of (2.7), where we further neglect the friction coefficients K_r , and K_g . As discussed, equation (2.11) describes the DFIG system, whose controlled output is the torque T_g , while we model the pitch one via the first order system

$$\tau_\vartheta \dot{\vartheta} = -\vartheta + \vartheta_r \quad (2.18)$$

The overall WT model considered for the control design is described by

$$\begin{pmatrix} \dot{\omega}_r \\ \dot{\omega}_g \\ \dot{\delta} \\ \dot{\vartheta} \\ \dot{T}_g \end{pmatrix} = \begin{pmatrix} \frac{1}{J_r} \frac{P_r(\omega_r, \vartheta, v)}{\omega_r} - \frac{D_s}{J_r} \omega_r + \frac{D_s}{J_r n_g} \omega_g - \frac{K_s}{J_r} \delta \\ \frac{D_s}{J_g n_g} \omega_r - \frac{D_s}{J_g n_g^2} \omega_g + \frac{K_s}{J_g n_g} \delta - \frac{1}{J_g} T_g \\ \omega_r - \frac{1}{n_g} \omega_g \\ -\frac{1}{\tau_\vartheta} \vartheta + \frac{1}{\tau_\vartheta} \vartheta_r \\ -\frac{1}{\tau_T} T_g + \frac{1}{\tau_T} T_{g,r} \end{pmatrix} \quad (2.19)$$

$$P_g = T_g \omega_g$$

The controlled input of the system is thus $u \triangleq \text{col}(\vartheta_r, T_{g,r})$ and its state vector is $x \triangleq \text{col}(\omega_r, \omega_g, \delta, \vartheta, T_g)$. It is easy to see that the system is affine in the control, i.e. of the form

$$\begin{cases} \dot{x} = f(x, v) + g(x)u \\ P_g = h(x) \end{cases}$$

where $f(x, v)$, $g(x)$, and $h(x)$ can be identified from equation (2.19). Note that v acts as a disturbance, and since it is a function of time, it makes the system *time-varying*. Moreover, for the sake of control design, for which an analytic representation of $f(x, v)$ is needed, in this chapter we consider the power coefficient approximation given by (2.2), and appearing in P_r according to (2.1). Let us consider system (2.19). Given an effective wind speed signal $v(t)$ and a time-varying reference trajectory of power $P_g^*(t)$ verifying (2.15), and such that it is an admissible steady state target for system (2.19), i.e. $\forall t \geq 0$ it is such that it always exists an admissible solution $(x_s(t), u_s(t))$ to the following set of nonlinear equations

$$\begin{cases} 0 = f(x_s(t), v(t)) + g(x_s(t))u_s(t) \\ P_g^*(t) = h(x_s(t)) \end{cases} \quad (2.20)$$

We can define the control problem as that of finding the input vector $u(t)$ that minimizes the distance between the system variables $(x(t), u(t))$ and the pair $(x_s(t), u_s(t))$, $\forall t \geq 0$. Note that (2.20) has to be solved employing the solution of (2.16), thus yielding a *unique* solution. During the following discussion, the wind speed v and state x are supposed to be known. However, in a second step, we will make use of an observer to determine, among other relevant variables of the system, an estimation of the effective wind speed v , namely \hat{v} . This estimation is treated in details in Section 2.6.

In order to solve the posed control problem we capitalize on the FL+MPC technique, whose basic concepts are shown in Chapter 1.

2.5.2 Feedback Linearization Step

The first control design step is concerned with the FL technique. As proposed by [Thomsen 2006], this is employed to directly target the system nonlinearities. These concern the relation between the input ϑ_r , and the rotor speed ω_r . Let us consider

the coordinate transformation

$$\xi = \Phi_\xi(x) = \begin{pmatrix} x_1 \\ L_f x_1 \\ x_2 \\ x_3 \\ x_5 \end{pmatrix} = \begin{pmatrix} x_1 \\ \frac{1}{J_r} \frac{P_r(x_1, x_4, v)}{x_1} - \frac{D_s}{J_r} x_1 + \frac{D_s}{J_r n_g} x_2 - \frac{K_s}{J_r} x_3 \\ x_2 \\ x_3 \\ x_5 \end{pmatrix} \quad (2.21)$$

Note that the above change of coordinates depends on v . Change of coordinates (2.21) leads to an exactly feedback linearizable system. This is shown in the following. The system in new coordinates has the form of

$$\begin{aligned} \dot{\xi} &= A_\xi \xi + B_\xi (a_\xi(\Phi_\xi^{-1}(\xi)) + b_\xi(\Phi_\xi^{-1}(\xi))u) = \\ &\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{D_s}{n_g J_g} & 0 & -\frac{D_s}{n_g^2 J_g} & \frac{K_s}{n_g J_g} & -\frac{1}{J_g} \\ 1 & 0 & -\frac{1}{n_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_T} \end{bmatrix} \xi + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{\tau_T} \end{bmatrix} \left(\begin{bmatrix} L_f^2 \xi_1 \\ 0 \end{bmatrix} + \begin{bmatrix} L_g L_f \xi_1 & 0 \\ 0 & 1 \end{bmatrix} u \right) \end{aligned} \quad (2.22)$$

where the pair (A_ξ, B_ξ) is controllable. Here, a_ξ is a 2-dimensional column vector of functions, and b_ξ a matrix of functions of dimension 2×2 , and they can be easily identify from (2.22). In other words, transformation (2.21) is an alternative change of coordinates with respect to the one transforming the system in its *normal form*, that still allows feedback linearization. This fact is reminded for the SISO case in Appendix A.4. According to it, and as shown by [Thomsen 2006], this means that the mapping Φ_ξ is related to the classic coordinate transformation Φ_z leading to a normal form, via a linear nonsingular transformation involving a matrix $M_{\xi z}$. Indeed, if we consider $y = \text{col}(x_2, x_3)$ as the output mapping for system (2.19), this naturally defined the change of coordinates Φ_z as

$$z = \Phi_z(x) = \begin{pmatrix} x_2 \\ L_f x_2 \\ \text{---} \\ x_3 \\ L_f x_3 \\ L_f^2 x_3 \end{pmatrix} = \begin{pmatrix} \omega_g \\ \dot{\omega}_g \\ \text{---} \\ \delta \\ \dot{\delta} \\ \ddot{\delta} \end{pmatrix}$$

Thus, the relationship between Φ_ξ and Φ_z is given by

$$\Phi_\xi = M_{\xi z} \Phi_z = \begin{bmatrix} \frac{1}{n_g} & 0 & 0 & 1 & 0 \\ 0 & \frac{1}{n_g} & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -J_g & \frac{K_s}{n_g} & \frac{D_s}{n_g} & 0 \end{bmatrix} \Phi_z$$

It is important to notice that the relative degree vector of system (2.19) with respect to the output $y = \text{col}(x_2, x_3)$ is $\text{col}(r_1, r_2) = \text{col}(2, 3)$, i.e. $r_1 + r_2 = n$, where $n = 5$ is system (2.19) dimension. As a consequence, system (2.19) can be *exactly* feedback linearized via Φ_z , and via Φ_ξ , since the transformation between Φ_z and Φ_ξ is linear. One of the advantages of considering Φ_ξ over Φ_z is that, in the new coordinates ξ , the former preserves a physical meaning, and this is helpful when designing a controller in cascade with the FL one, such that the closed-loop system has desired performance. In order to feedback linearize the system in the new coordinates (2.22), notice that all the system nonlinearities are concentrated in its second row. In particular, we can write it as

$$\dot{\xi}_2 = L_f^2 x_1 + L_g L_f x_1 \vartheta_r = \alpha(\xi, \vartheta, v, \dot{v}) + a_2 \xi + \beta(\xi, \vartheta, v) \vartheta_r \quad (2.23)$$

where a_2 is a row vector such that $a_2 \xi$ collects all the linearities in $L_f^2 x_1$, $\alpha(\xi, \vartheta, v, \dot{v}) + a_2 \xi = L_f^2 x_1$ and $\beta(\xi, \vartheta, v) = L_g L_f x_1$. Thus the feedback linearizing control law only concerns input ϑ_r . Moreover, at this point, we make the choice to use FL to *only* eliminate the nonlinearities appearing in (2.23). This is obtained via

$$\vartheta_{r,FL} \triangleq \vartheta_r = \frac{1}{\beta(\xi, \vartheta, v)} (-\alpha(\xi, \vartheta, v, \dot{v}) + v_\vartheta) \quad (2.24)$$

where v_ϑ is left as degree of freedom as in classic FL technique. The feedback linearized system is then

$$\begin{aligned} \dot{\xi} &= A\xi + B[v_\vartheta \quad T_{g,r}]^\top = \\ & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ a_{2,1} & 0 & a_{2,3} & a_{2,4} & a_{2,5} \\ \frac{D_s}{n_g J_g} & 0 & -\frac{D_s}{n_g^2 J_g} & \frac{K_s}{n_g J_g} & -\frac{1}{J_g} \\ 1 & 0 & -\frac{1}{n_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_T} \end{bmatrix} \xi + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{\tau_T} \end{bmatrix} \begin{bmatrix} v_\vartheta \\ T_{g,r} \end{bmatrix} \end{aligned} \quad (2.25)$$

where the pair (A, B) is controllable. In addition, note that under the assumption of known wind speed signal, system (2.25) no longer depends on v , thus it is made *time-invariant* via FL.

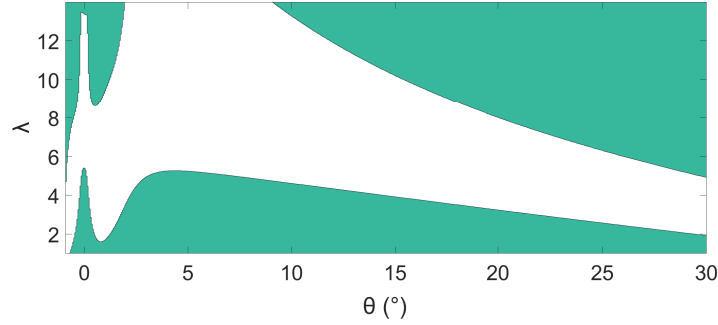


Figure 2.13: $\Lambda_t \triangleq \{(\lambda, \vartheta) \in \mathbb{R}^2 | \beta(\lambda, \vartheta) < 0\}$, white area, defines the set of points in which the WT relative degree is well-defined.

Remark 2.4 The choice of (2.24) is motivated by two main reasons. The first is that we want to reduce the number of exact cancellations as they are intrinsically non robust, leaving them just to the nonlinear terms. Secondly, if the term $a_2\xi$ was canceled by FL control, then variable ξ_1 , which is the rotor speed ω_r in the original coordinates, would be controlled only by v_ϑ . However, being ϑ_r limited in order to limit ϑ (see Table 2.1), this in turns results in constraints on v_ϑ . Thus, if we let a complete cancellation of $a_2\xi$, we implicitly affect the controllability of the system. Being a_2 a nonzero row, we are able to keep a coupling between the input $T_{g,r}$ and the variable ξ_1 .

2.5.3 Singular Points Analysis

We saw that Φ_ξ , defined in (2.21), leads to an exact feedback linearized system. Another way to interpret Φ_ξ is to consider x_1 , i.e. ω_r , as system (2.19) output. From this point of view, Φ_ξ is obtained by computing the relative degree associated to the output x_1 , which naturally defines the first two rows of mapping (2.21), and by completing the change of coordinates with its last chosen three rows. As a matter of fact, the relative degree of system (2.19) with respect to x_1 is not global. Namely, there exist points where $\beta(\xi, \vartheta, v) = 0$. Nonetheless, by simulation, β appears to be negative-valued in the points of functioning of interest, i.e. in the steady state values of the state space satisfying (2.20). In addition, numerical analysis shows that the domain in which β has negative value is *connected*, provided that $\vartheta > -1^\circ$. Indeed $\vartheta = -1^\circ$ is a singularity for (2.2). For this reason, we further limit ϑ to the range $(-1^\circ, \vartheta_{max}]$. Moreover $\beta(\xi, \vartheta, v) = \beta(\xi_1, \vartheta, v)$, i.e. it only depends on the variables ξ_1 , ϑ , and v and it can be further represented as a function of (λ, ϑ) , i.e. $\beta(\lambda, \vartheta)$. This allows us to show that the couples (λ, ϑ) such that $\beta < 0$, in the domain of interest, form a connected set. This, named $\Lambda_t \triangleq \{(\lambda, \vartheta) \in \mathbb{R}^2 | \beta(\lambda, \vartheta) < 0\}$, is shown in Fig. 2.13. As a consequence, if the system initial condition $x_0 \in \Lambda_t$, and the steady state solution of (2.16), (2.20), is such that $x_s(t) \in \Lambda_t \forall t \geq 0$, then, conditions of Proposition 1.2, and condition (i) of Proposition 1.1 are satisfied. In other words, the system relative degree is well-defined in x_0 and $x_s(t) \forall t \geq 0$, and

they belong to a connected set verifying $L_g L_f x_1 \neq 0$, i.e. $\beta < 0$ in this case. Thus, we aim at imposing condition (ii) of Proposition 1.1 by means of an MPC controller in cascade with the FL one, i.e. by constraining the closed-loop system trajectory to lie within $\Lambda_t \forall t \geq 0$. This is object of the next subsection.

Remark 2.5 *Unfortunately stability results of Theorem 1.1 cannot be applied in this framework. This is due to several reasons. First of all, the problem here addressed is the one of output tracking, which, in the MPC framework, itself requires other stability results than the ones holding for the problem of output regulation. Secondly, the condition $\beta < 0$, defining the set Λ_t , is time-varying with respect to the variables ξ_1 , and ϑ , because of the presence of v , whose exact time evolution cannot be predicted. Eventually, linearized system (2.25) exhibits additional nonlinear, possibly nonconvex constraints, due to the FL step. For such system, it is generally not a trivial task to find a convex approximation for its constraints. These points are shown in details in the next subsection.*

2.5.4 Model Predictive Control Step

We now have to design the MPC controller for system (2.25). First of all this is discretized, using a sampling time T_s , yielding the discrete-time system

$$\xi^+ = A_d \xi + B_d [v_\vartheta \quad T_{g,r}]^\top \quad (2.26)$$

where A_d , and B_d are obtained using, for instance, Euler method. System variables and signals are then considered at the sampling steps $i \in \mathbb{N}^+$. MPC is thus employed to make the WT track the reference signals $P_g^*(i)$, $\omega_r^*(i)$, $\forall i \geq 0$, obtained via (2.16). In order to compute the system steady state target, note that ξ has a clear physical meaning thanks to the choice of (2.21), and it is easy to find a steady state solution $(\xi_s, v_{\vartheta,s}, T_{g,r,s})$ corresponding to P_g^* and ω_r^* . Indeed, instead of solving the system of nonlinear equations (2.20), thanks to FL, one can use

$$\begin{cases} A\xi_s + B[v_{\vartheta,s} \quad T_{g,r,s}]^\top = 0 \\ \xi_{3,s}\xi_{5,s} = P_g^* \\ \xi_{1,s} = \omega_r^* \end{cases} \quad (2.27)$$

It is easy to see that (2.27) is a system of linear equations leading to a unique solution. We additionally name $\bar{u} \triangleq \text{col}(v_\vartheta, T_{g,r})$, and $\bar{u}_s \triangleq \text{col}(v_{\vartheta,s}, T_{g,r,s})$. Having introduced the steady state variables for MPC, we can define the error variables $\tilde{\xi} \triangleq \xi_s - \xi$, and $\tilde{u} \triangleq \bar{u}_s - \bar{u}$, which will be used in the optimization problem. The choice of MPC is mainly motivated by its capability to explicitly handle constraints. These come either from the WT physical constraints, and from additional ones introduced to let the system trajectories avoid the singular points. As a matter of

fact system (2.26) is subject to the following physical constraints

$$\omega_{r,min} \leq \xi_1 \leq \omega_{r,max} \quad (2.28)$$

$$0 \leq \xi_3 \quad (2.29)$$

$$0 \leq \xi_5 \quad (2.30)$$

$$\xi_3 \xi_5 \leq 1.05 P_{e,n} \quad (2.31)$$

$$\vartheta_{min} \leq \vartheta_{r,FL} \leq \vartheta_{max} \quad (2.32)$$

where (2.28), (2.29), and (2.30) are WT linear constraints, (2.31) is the nonlinear constraint describing the maximum allowed electric power during transients, and (2.32) is used to limit the pitch angle. Notice that the $\vartheta_{r,FL}$ is a nonlinear function of the system variables due to the FL step. A few words have to be spent on this last inequality. Since the second component of (2.21) is a complex expression of the system variables, it is not easy to derive an analytic relationship between ϑ and the new coordinates ξ . This is the reason why we choose to limit ϑ_r in order to respect the constraints on ϑ . Even though this solution is conservative, if the constraints on ϑ_r are satisfied, so they will be on ϑ via the pitch actuator model (2.18). Provided that the system initial state satisfies (i) of Proposition 1.1 and belongs to Λ_t , we only need to verify that the target state ξ_s belongs to Λ_t too at each time step i , and to impose (ii) of the mentioned proposition. This is done by adding the following constraint to the set of the WT physical ones in the MPC optimization problem

$$\beta(\xi_1, \vartheta, v) + \epsilon < 0, \quad \epsilon > 0 \quad (2.33)$$

where we added ϵ to let some margin. If (2.33) is satisfied for all the possible values of v in the prediction horizon, then the new state is guaranteed to lie in Λ_t , thus letting the system avoiding the singular points.

We can now provide the MPC problem formulation that leads to an *implicit* control law. The optimization problem $\mathbb{P}_N(\tilde{\xi}(i))$ that has to be solved at each time step i is defined as follows.

$$\begin{aligned} \min_{\{s, \tilde{u}\}} \quad & \sum_{k=0}^{N-1} \left(\tilde{\xi}(k)^\top Q_\xi \tilde{\xi}(k) + \tilde{u}(k)^\top R \tilde{u}(k) \right. \\ & \left. + r_s s_l(k)^2 + z_e(k)^\top Q_z z_e(k) + \Delta \tilde{u}(k)^\top R_\Delta \Delta \tilde{u}(k) \right) + \tilde{\xi}(N)^\top P \tilde{\xi}(N) \end{aligned} \quad (2.34)$$

subject to

$$\tilde{\xi}^+ = A_d \tilde{\xi} + B_d \tilde{u}, \quad \tilde{\xi}(0) = \tilde{\xi}(i)$$

$$\omega_{r,min} \leq \xi_1 \leq \omega_{r,max}$$

$$0 \leq \xi_3, \xi_5$$

$$\xi_3 \xi_5 \leq 1.05 \cdot P_{e,n} + s_l, \quad s_l \geq 0$$

$$\vartheta_{min} \leq \vartheta_{r,FL} \leq \vartheta_{max}$$

$$\beta(\xi_1, \vartheta, v) + \epsilon < 0, \quad \epsilon > 0$$

where, during the prediction horizon, ξ_s , and \bar{u}_s are considered constant and equal to the current value $\xi_s(i)$, and $u_s(i)$. We further introduced variable $\Delta\tilde{u}(k) \triangleq \tilde{u}(k) - \tilde{u}(k-1)$ to manage the input variation between two steps, and we enriched the MPC problem with two *integral actions* on the errors $\tilde{\xi}_3$, and $\tilde{\xi}_5$ respectively, and whose state variable is $z_e \in \mathbb{R}^2$. This helps to keep the error on P_g bounded in the presence of disturbances and model-plant mismatches. For instance, a source of error is given by the mismatch between the analytic expression of C_p , i.e. (2.2), used for the control design, and the CART C_p of Fig. 2.2. $Q_\xi \succeq 0$, Q_z , R , $R_\Delta \succ 0$ are the weight matrices, and P is the positive definite matrix solution of the Riccati equation that solves the infinite-horizon LQR problem associated to system (2.26). Note that by proper choice of Q_ξ we can keep variable ξ_4 , i.e. δ in the original coordinates, small. This in turns lets reduction of stress on the drive train. In addition we modified constraint (2.31) to help avoiding problem infeasibility by introducing slack variable s_l to the optimization problem and the corresponding weight r_s .

As mentioned in Remark 2.5, we cannot apply the results of Theorem 1.1 to conclude on the closed-loop stability of the system. Moreover, even though the optimization problem $\mathbb{P}_N(\tilde{\xi}(i))$ exhibits a *quadratic* cost function with an underlying LTI system, because of constraints (2.32), (2.33), and the modified constraint of (2.31), we cannot even conclude on the convexity of the problem. This is why, in this framework, we decide to treat the nonlinear constraints of (2.34) via the approximation procedures, object of the discussion of Section 1.1 in Chapter 1. In particular, we apply a *linear* approximation of the constraints, updated at each time step i in the current state, thus yielding constraints of the form

$$F_i\tilde{\xi} + G_i\tilde{u} \leq b_i \quad (2.35)$$

where the matrices F_i , G_i , and b_i represent the linear approximation at the current time step i , i.e. valued in the current state. Equation (2.35) describes a *time-varying* polytope, which is kept constant during the prediction horizon, and it makes the optimization problem *quadratic*.

Let us carry on the discussion on the constraints approximation a little bit further. In particular, we focus on constraints (2.32), and (2.33), where the former takes the extended form of

$$\alpha(\xi, \vartheta, v, \dot{v}) + \beta(\xi, \vartheta, v)\vartheta_{max} \leq v_\vartheta \leq \alpha(\xi, \vartheta, v, \dot{v}) + \beta(\xi, \vartheta, v)\vartheta_{min}$$

under the assumption of $\beta(\xi, \vartheta, v) < 0$. Both constraints depend on variable ϑ . However, as previously stated, it is difficult to have an analytic relationship between ϑ and the system new coordinates ξ . Nonetheless, the dynamics of ϑ is known, and its Euler approximation has the form of

$$\vartheta^+ = \left(1 - \frac{T_s}{\tau_\vartheta}\right)\vartheta + \frac{T_s}{\tau_\vartheta}\vartheta_{r,FL} \quad (2.36)$$

Thus (2.36) can be used to predict the behavior of ϑ in the prediction horizon of $\mathbb{P}_N(\tilde{\xi}(i))$, and to satisfy the mentioned constraints. However, since (2.36) is

nonlinear because of $\vartheta_{r,FL}$, in order to have a linear approximation of the constraints, the following steps are performed

- I Linearization of (2.36) in the system variables value at the current step i .
- II Linearization of constraints (2.32), and (2.33).
- III Composition of the linearization of steps I and II, yielding a linear constraint in the considered horizon.

Eventually, one last issue needs to be considered. Even though system (2.26) is time-invariant, because the dependency on v has been eliminated via FL, this is not the case for constraints (2.32), and (2.33). In fact, these depend on v and on its time derivative \dot{v} , thus, some additional steps have to be made in order to cancel the perturbation v out of such constraints in the prediction horizon. We identify two main possibilities

- Pre-assign a wind profile along the prediction horizon.
- Consider a worst case scenario to allow robust constraint satisfaction.

As far as the first is concerned, as done, for the target variables ξ_s , and \bar{u}_s , one may consider v *constant* along the prediction horizon, and equal to the current value $v(i)$. Another possibility is to let \dot{v} be constant, and equal to $\dot{v}(i)$. As a consequence v is *linearly* increasing, or decreasing, along the prediction horizon, according to

$$v(k+1) = v(k) + T_s \dot{v}(i), \quad v(0) = v(i)$$

These two approaches allow a simple MPC implementation when no information on the wind prediction is available. In both cases, if T_s and N are small, then the wind prediction error is relatively small too. However a proper tuning of parameter ϵ in (2.33), and possibly the introduction of a similar one for (2.32), should be considered in order to allow a *safe* constraint satisfaction under such bounded wind prediction error.

Differently, one may consider a worst case scenario to eliminate the dependency of the constraints on v . This is what concerns the second possibility. As an example, let us consider (2.33). Similar results hold for (2.32). Its linearization at step i can be expressed as

$$\begin{aligned} \beta(\xi_1, \vartheta, v) + \epsilon &\simeq \beta(\xi_1(i), \vartheta(i), v(i)) + \epsilon + \frac{\partial \beta}{\partial \xi_1} \Big|_{(\xi_1(i), \vartheta(i), v(i))} (\xi_1 - \xi_1(i)) \\ &+ \frac{\partial \beta}{\partial \vartheta} \Big|_{(\xi_1(i), \vartheta(i), v(i))} (\vartheta - \vartheta(i)) + \frac{\partial \beta}{\partial v} \Big|_{(\xi_1(i), \vartheta(i), v(i))} (v - v(i)) \end{aligned} \quad (2.37)$$

By employing system dynamics (2.26), the linearization of ϑ dynamics, and the following wind dynamics

$$\begin{aligned} v(k+1) &= v(k) + T_s \dot{v}(k), \quad v(0) = v(i) \\ |\dot{v}(k)| &\leq |\dot{v}|_{max} \quad k = 0, \dots, N \end{aligned}$$

where by $\dot{v}(k)$ we indicate the wind speed time derivative evaluated at time $t = kT_s$, and where we assume the value $|\dot{v}|_{max}$ to be known, then (2.37) can be limited from above by solving the following *linear programming* problem

$$\begin{aligned} & \max_{v(k), k=0, \dots, N} \quad (2.37) \\ & \text{subject to} \quad (2.38) \\ & v(i) - kT_s |\dot{v}|_{max} \leq v(k) \leq v(i) + kT_s |\dot{v}|_{max} \end{aligned}$$

The solution of (2.38) is then used instead of (2.33). It has to be stressed that this worst scenario is based on system dynamics and constraint function approximations, thus, also in this case, a proper parameter ϵ tuning has to be performed. All in all, if at a given step i , $\mathbb{P}_N(\tilde{\xi}(i))$ is feasible, then the next step state is guaranteed to respect the system constraints, and to stay within Λ_t . This fact does not have to be confused with the *recursive feasibility* property, i.e. feasibility of $\mathbb{P}_N(\tilde{\xi}(i))$ does not necessarily imply feasibility of $\mathbb{P}_N(\tilde{\xi}(i+1))$. As a consequence, for real implementation, some additional sub-routines should be considered to allow MPC to recover feasibility whenever it is lost. This problem is not further developed in this work.

2.6 Wind Estimation

So far the wind speed v has been considered as a known quantity. This assumption fails to be realistic for practical control implementation. This is why, in this section, we study a wind speed estimation technique. Such estimation needs to be accurate since, as shown in the previous sections, the wind speed value is directly employed several times in the control design. In particular, recall that v , with its first time derivative, has been used to *compute the WT set points*, to *feedback linearize the system*, and to *allow satisfaction of system constraints* in the MPC design step. One first possibility to have an on-line estimation of v is to use a wind measurement, e.g. the one provided by the WT anemometer, see Fig. 2.1, combined with known filtering techniques that exploit the WT model. For instance, a widely used technique is given by the Extended Kalman filter, see e.g. [Ma 1995]. As the wind speed value is generally not directly available, and the anemometer measurement poorly reliable, [Boukhezzar 2011], we capitalize on Kalman filtering technique for the sake of system state and wind speed estimation without the use of wind measurement. This is done in two steps. Firstly, an estimation of the aerodynamic torque T_r is obtained, based on the work of [Østergaard 2007], and [Boukhezzar 2011]. Secondly, the wind speed is deduced by employing the WT aerodynamic relation between the wind speed itself and T_r , [Boukhezzar 2011].

2.6.1 Aerodynamic Torque Estimation via Kalman Filter

As far as the aerodynamic torque estimation is concerned, we augment the WT state by introducing the following differential equation for T_r

$$\dot{T}_r = w_t$$

where w_t is a white Gaussian noise, whose covariance is given by $Q = E[w_t^2]$, where $E[\cdot]$ is the *expected value* operator. In other words, for the only estimation problem we do not exploit the model knowledge of the dynamics of T_r . Thus, using WT model (2.19), the augmented one is given by

$$\begin{aligned} \dot{x}_{ag} = A_{ag}x + B_{ag}u + Gw_t = \\ \begin{bmatrix} -\frac{D_s}{J_r} & \frac{D_s}{J_r n_g} & -\frac{K_s}{J_r} & 0 & 0 & \frac{1}{J_r} \\ \frac{D_s}{J_g n_g} & -\frac{D_s}{J_g n_g^2} & \frac{K_s}{J_g n_g} & 0 & -\frac{1}{J_g} & 0 \\ 1 & -\frac{1}{n_g} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_\vartheta} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau_T} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_g \\ \delta \\ \vartheta \\ T_g \\ T_r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\tau_\vartheta} & 0 \\ 0 & \frac{1}{\tau_T} \\ 0 & 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w_t \end{aligned} \quad (2.39)$$

The considered measured output is $y = Cx_{ag} \triangleq \text{col}(\omega_r, \omega_g, \vartheta, T_g)$, to which we add a white Gaussian noise w_m , whose covariance is $R = E[w_m w_m^\top]$, yielding the measurement equation

$$z = Cx_{ag} + w_m \quad (2.40)$$

One advantage of using the aforementioned system is that classic results of linear Kalman filtering can be applied. In particular, the stable stationary Kalman filter is given by

$$\dot{\hat{x}} = A_{ag}\hat{x} + B_{ag}u + K(z - C\hat{x}) \quad (2.41)$$

where \hat{x} is the estimated state, $K = PCR^{-1}$, and where P is the solution of the Riccati equation

$$A_{ag}P + PA_{ag}^\top + PC^\top R^{-1}CP + GQG^\top = 0$$

Thanks to the properties of the Kalman filter, the estimation \hat{x} is asymptotically optimal with respect to the minimization of the covariance of the error $E[(x - \hat{x})(x - \hat{x})^\top]$.

2.6.2 Indirect Wind Estimation

Once $\hat{x} = \text{col}(\hat{\omega}_r, \hat{\omega}_g, \hat{\delta}, \hat{\vartheta}, \hat{T}_g, \hat{T}_r)$ is obtained, v can be estimated by solving the following equation

$$\hat{T}_r \hat{\omega}_r - \frac{1}{2} \rho \pi R^2 v^3 C_p \left(\hat{\omega}_r R v^{-1}, \hat{\vartheta} \right) = 0 \quad (2.42)$$

which is simply obtained from (2.1). In this work (2.42) is solved on-line, thus fast available gradient-based optimization algorithms must be considered. However, for such purpose, the power coefficient function is required to be analytic and differentiable. In [Boukhezzer 2011], authors capitalize on a polynomial approximation of C_p . This is simplified by the fact that, in their work, C_p is function of the only tip speed ratio λ , since ϑ is constant and equal to its MPPT value. Here, we extend their proposal to the case in which the power coefficient is a function of both λ , and ϑ . In order to do so, by numerical simulation it appears that a general polynomial approximation of C_p is not sufficient to guarantee a reliable wind estimation. This is why, differently from [Boukhezzer 2011], at each time step we consider a new *local* approximation of the power coefficient, around the last obtained wind estimation $\hat{v}_{prev} \triangleq \hat{v}(i-1)$. We indicate such approximation with $C_p^{pol} \left(\frac{\hat{\omega}_r R}{v}, \hat{\vartheta} \right)$, where $v \in [\hat{v}_{prev} - 3\sigma_v, \hat{v}_{prev} + 3\sigma_v]$, and where, if we consider the simplified assumption of \hat{v} being a Gaussian process, σ_v^2 identifies its variance, and the range $\pm 3\sigma_v$ is selected to have the 99% of probability to have v within it. In practice, σ_v can be considered as a tunable parameter to choose the confidence interval of \hat{v} , and it defines the interval in which C_p is approximated yielding C_p^{pol} . Eventually, a solution to (2.42) is obtained solving the following nonlinear programming problem

$$\begin{aligned} & \min_v \left(\hat{T}_r \hat{\omega}_r - \frac{1}{2} \rho \pi R^2 v^3 C_p^{pol} \left(\hat{\omega}_r R v^{-1}, \hat{\vartheta} \right) \right)^2 \\ & \text{subject to} \\ & \hat{v}_{prev} - 3\sigma_v \leq v \leq \hat{v}_{prev} + 3\sigma_v \end{aligned} \quad (2.43)$$

Remark 2.6 Problem (2.43) can be seen as a maximum likelihood estimation if we consider

$$\hat{T}_r \hat{\omega}_r = \frac{1}{2} \rho \pi R^2 v^3 C_p^{pol} \left(\hat{\omega}_r R v^{-1}, \hat{\vartheta} \right) + w_p$$

as measurement equation, where $\hat{T}_r \hat{\omega}_r$ is treated as the measure and w_p is a Gaussian white noise.

2.7 Simulation Results

The overall control scheme used in the following simulation results is shown in Fig. 2.14. The WT model is given by (2.19). Although for control design purposes the analytic approximation of C_p , i.e. equation (2.2), was considered, in order to test the effectiveness of the proposed control approach, the CART lookup table power coefficient of Fig. 2.2 is used in the numerical simulations. Thus, the power coefficient represents a first model-plant mismatch. A noisy measure z_m of $col(\omega_r, \omega_g, \vartheta, T_g)$ is used to estimate the system state and wind speed. This is done via (2.41), and (2.43). The wind estimation \hat{v} is used together with the desired power output P_g^* to compute the set points for the MPC control stage. This is done

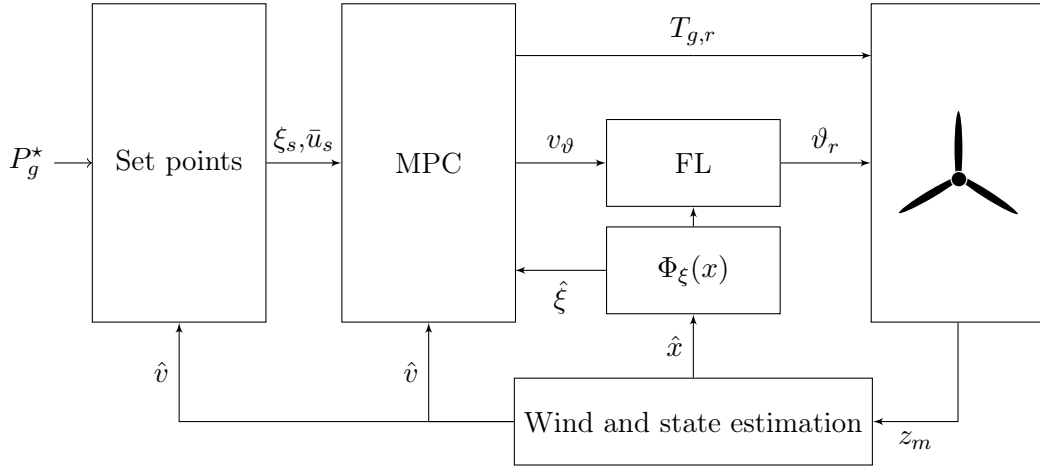


Figure 2.14: FL+MPC control scheme.

via (2.16), and (2.27). The MPC controller additionally makes use of the estimated state in the new coordinates $\hat{\xi}$, obtained from \hat{x} and diffeomorphism $\Phi_{\xi}(x)$. $T_{g,r}$ is directly applied to the WT model, while ϑ_r is recovered from v_{ϑ} via feedback linearizing input (2.24). To show the control scheme performance, we propose three simulation setups. First of all a comparison with a basic control scheme is provided on a mixed MPPT/power limiting and deloaded mode of functioning scenario. Robustness is then tested via Montecarlo simulation on classic mode of functioning, i.e. MPPT and power limiting. Eventually, the control approach is tested on some deloaded mode scenarios. For all the simulation setups the MPC prediction horizon is chosen to be $N = 3$ and the sampling time $T_s = 0.1$ s. Optimization problem (2.34) requires then a computational time of 4.9 ms¹. Moreover, concerning the problem of canceling the MPC constraints out of the wind signal v , we choose to pre-assign a wind profile during the prediction horizon. In particular we choose a constant \dot{v} value equal to the last estimated one $\dot{v}(i)$. By a proper selection of ϵ parameter of constraint (2.33), the controller provides good performance while letting a simpler control implementation with respect to the more robust possibility of worst case scenario shown in Subsection 2.5.4.

2.7.1 PI Control Comparison

We compare the proposed FL+MPC controller with a PI. The latter is composed of two separate feedback loops. The first one is concerned with the control of ω_r by acting on $T_{g,r}$, which is the output of a PI controller. The second one enables power limiting through the use of the pitch angle. Thus ϑ_r is output of a PI controller comparing the power reference and the actual mechanical power. This controller is

¹Computations are carried on an Intel®Core™i7-4790S, CPU @ 3.20 GHz, RAM of 16 Gb, with MATLAB®R2015a.

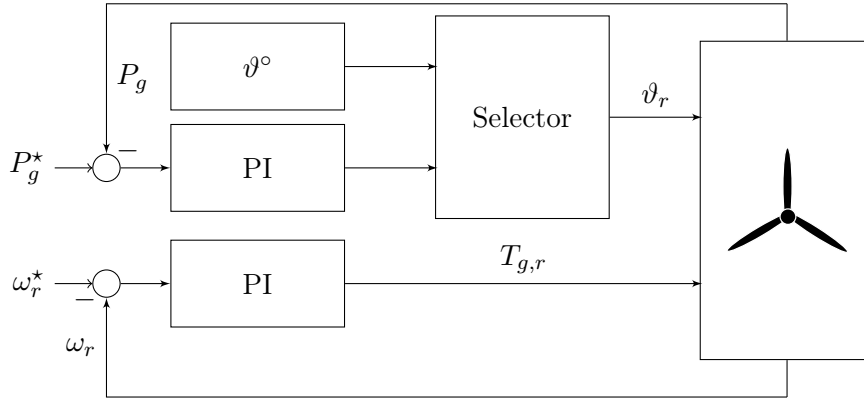


Figure 2.15: PI-based control for WT general power tracking.

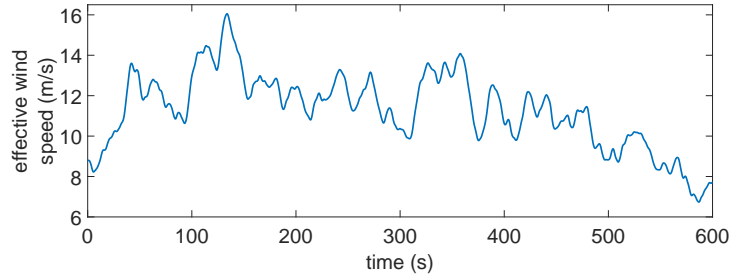
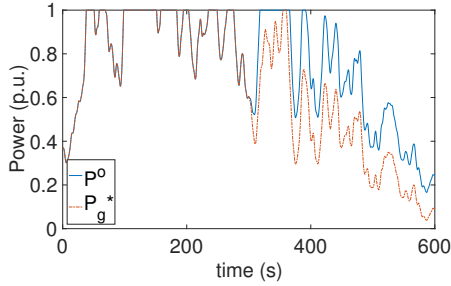
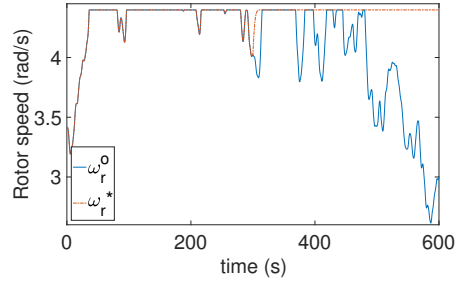


Figure 2.16: Effective wind speed.



(a) MPPT/power limiting and deloaded power reference.



(b) MPPT/power limiting and deloaded rotor speed reference.

Figure 2.17: MPPT/power limiting and deloaded.

activated only when the wind extractable power is higher than the reference one. ϑ_r is kept constant to its optimal MPPT value otherwise. The control scheme describing the chosen PI structure is shown in Fig. 2.15.

Concerning the simulation, during a time interval of 600 s the turbine is excited by the effective wind speed in the axial direction shown in Fig. 2.16. For such simulation, we only aim at comparing the FL+MPC performance against the PI one, thus we do not consider the wind estimation. In other words, the estimation

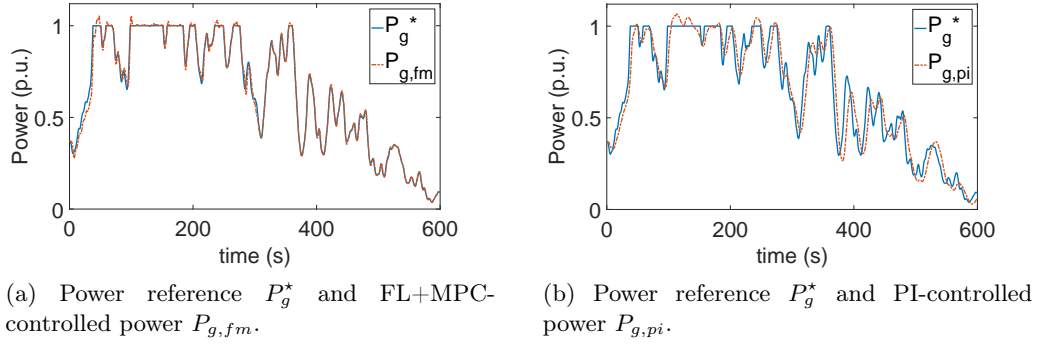


Figure 2.18: FL+MPC vs PI controlled power.

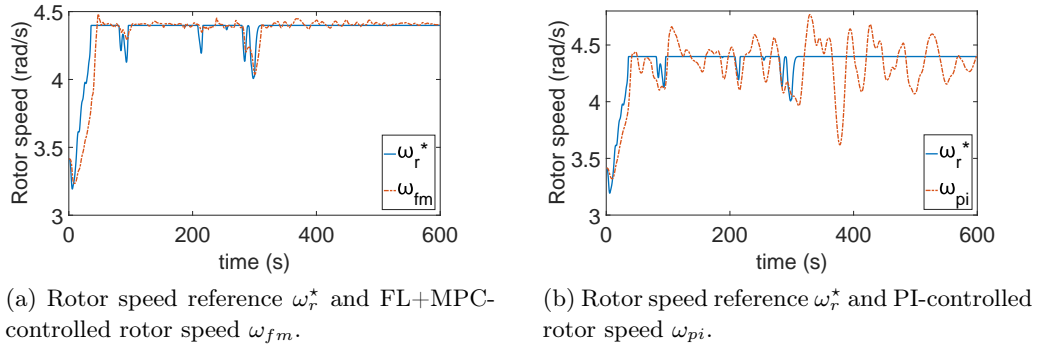


Figure 2.19: FL+MPC vs PI controlled rotor speed.

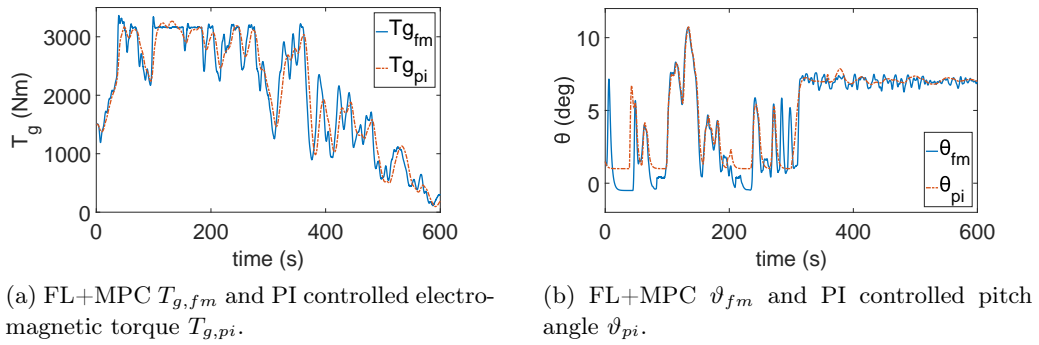


Figure 2.20: FL+MPC vs PI system inputs.

block of Fig. 2.14 is eliminated for this simulation purpose. For the first 300 s the turbine is controlled in the classic functioning of MPPT and power limiting, proving the capability of the proposed controller to work in the entire operating envelope. After 300 s the turbine is deloaded of a certain time-varying factor with respect to the maximum extractable power, as shown in Fig. 2.17a, where the reference power signal P_g^* (blue solid line) drifts from the maximum power P^o (red dash-dotted

Table 2.2: Simulation on FL+MPC vs PI

	Energy	$ \overline{err}_{\omega_r} $	$ \overline{err}_{P_g} $
FL+MPC	232.51 MJ	0.0434 rad/s	8.315 kW
PI	231.79 MJ	0.1420 rad/s	36.884 kW
(FL+MPC)/PI	1.0031	0.3059	0.2254

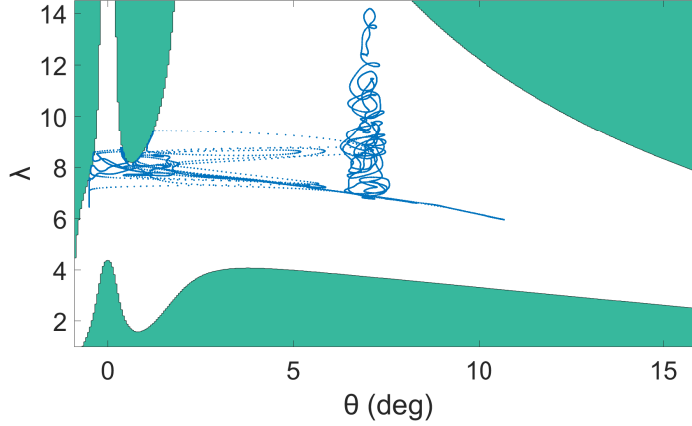
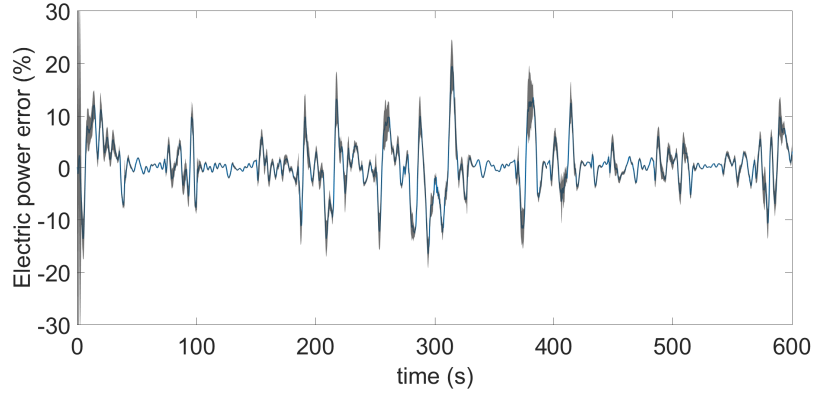


Figure 2.21: FL+MPC trajectories (*blue dots*) during the simulation. Thanks to the MPC constraints, these stay within Λ_t .

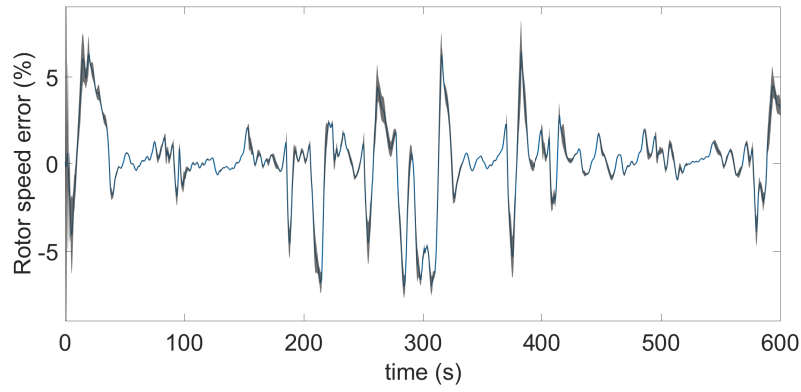
line). The ω_r reference corresponding to P_g^* is computed via (2.16). As shown in Fig. 2.17b, it reaches the nominal value $\omega_{r,n}$ (red dash-dotted line), drifting from ω_r^o (blue solid line) which corresponds to the P^o signal. FL+MPC (*fm*) and PI (*pi*) responses are compared throughout the whole simulation time. In particular the delivered power P_g is compared in Fig. 2.18, while the controlled rotor speed in Fig. 2.19. The main results are reported in Table 2.2 where we indicate by $|\overline{err}_{\omega_r}|$ and $|\overline{err}_{P_g}|$ the mean value of absolute rotor speed error and electric power error respectively, and they show that even if the delivered electrical energy is comparable, FL+MPC outperforms the PI in terms of precision, leading to smaller errors on the desired power and rotor speed. Moreover Fig. 2.19a shows how FL+MPC enables lower rotor speed oscillations during the deloaded mode of operation, thanks to a proper combined action of the controlled inputs T_g , and ϑ . These are reported in Fig. 2.20 for both FL+MPC and PI controller. Eventually, concerning the FL+MPC control scheme, Fig. 2.21 shows how during the entire simulation, constraint (2.33) is satisfied, i.e. system trajectories stay within Λ_t thus avoiding the system singular points.

2.7.2 Montecarlo Simulation

We employ Montecarlo method to test the controller robustness with respect to uncertainties in the model parameters D_s , K_s , J_r , and J_g . In particular we assign a

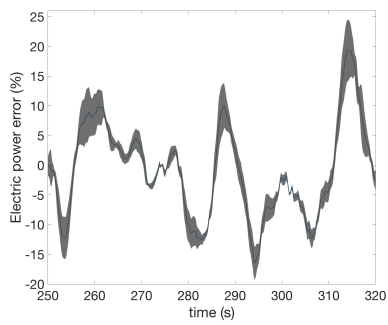


(a) Percentage error on the power reference.

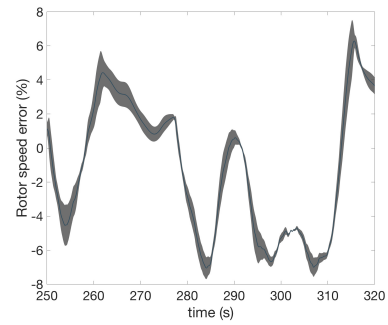


(b) Percentage error on the rotor speed reference.

Figure 2.22: Montecarlo simulation with respect to parameter uncertainties on the error on power and rotor speed reference. These have 99% of probability to lie within the grey area.



(a) Percentage error on power.



(b) Percentage error on rotor speed.

Figure 2.23: Zoom on Montecarlo simulation.

uniform distribution to the latter, centered in their nominal value, and we let them

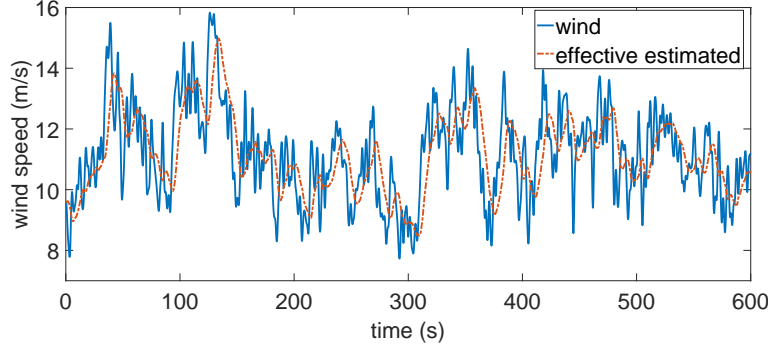
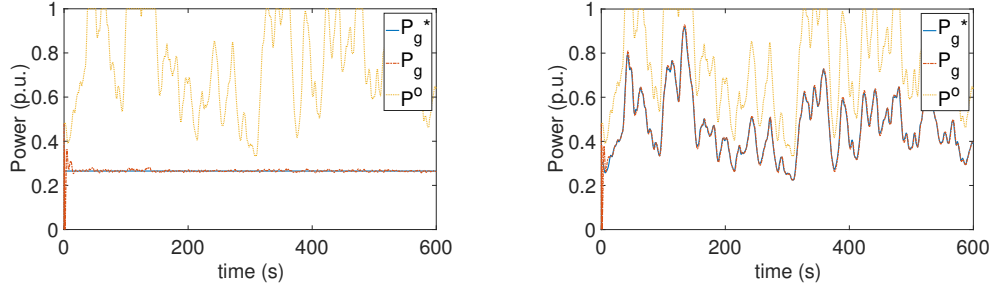


Figure 2.24: Real and effective estimated wind speed.

span an interval of $\pm 20\%$ of their nominal value. 100 simulations are performed on a 600 s time basis, and for each of them the system parameters are allowed to get a value according to the prescribed probability density. Concerning the simulation conditions, we let the turbine operate in classic MPPT/power limiting mode while being excited by a wind speed whose effective value is given in Fig. 2.16. Differently from the previous simulation setup, we do consider here the wind estimation. The reference signals P_g^* , and ω_r^* corresponding to the MPPT/power limiting operating mode for the given wind signal are thus equal to the MPPT ones (blue solid line) of Fig. 2.17. Fig. 2.22 shows that the error on the electric power P_g and on the rotor speed ω_r have approximately 99% of probability to lie within the grey area, i.e. an interval of ± 3 times the computed standard deviation around the computed mean value. For the sake of clarity, the same figure are illustrated in a zoomed window in Fig. 2.23. The error mean value (blue solid line) grows in respect to fast changes in the desired references, while it stabilizes around zero when the reference varies more slowly. Generally speaking though, the proposed control scheme is able to maintain stability and performance with respect to parametric model-plant mismatch since the errors remain within acceptable percentage ranges with respect to the reference values.

2.7.3 Deloaded Mode Scenarios

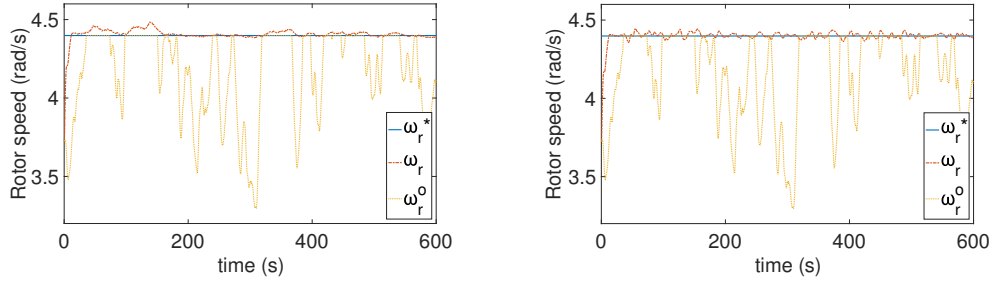
In this simulation setup we aim at testing the FL+MPC control scheme capability to let a WT track a deloaded power reference. In particular, we propose two possible scenarios. Namely, we first require the WT to track a *constant* power reference. Secondly, we choose a *varying* deloaded reference, obtained as a given percentage of the maximum extractable power from the wind. For both of them, we consider the wind speed signal perturbing the WT shown in Fig. 2.24, where the real wind speed blowing towards the WT blades is presented in blue solid line. In red dash-dotted line instead, we show, as an example, the effective estimated wind speed for one of the two scenarios. As far as the first one is concerned, we select $P_g^*(t) = 26, 5\% P_{e,n} \forall t \geq 0$, while, for the second one, we require the WT to track the 60% of the



(a) Controlled power for a constant deloaded power reference.

(b) Controlled power for a varying deloaded power reference.

Figure 2.25: Power for constant and varying deloaded mode of functioning.

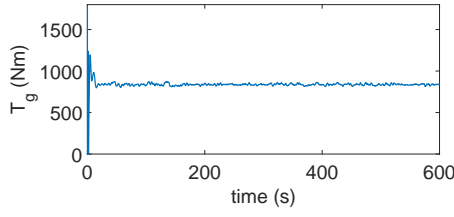


(a) Controlled rotor angular speed for a constant deloaded power reference.

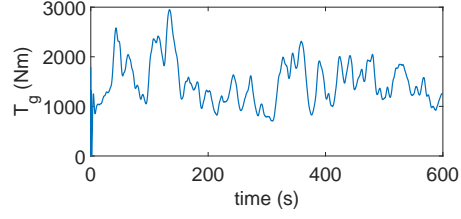
(b) Controlled rotor angular speed for a varying deloaded power reference.

Figure 2.26: Rotor angular speed signals for constant and varying deloaded mode of functioning.

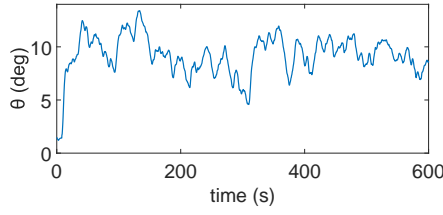
maximum extractable power signal. Performance on the controlled P_g are shown in Fig. 2.25, where the solid blue line identifies P_g^* , and the red dash-dotted one P_g . We additionally provide the MPPT/power limiting signal in yellow dotted line to show the comparison with the classic mode of functioning reference. Similar results are reported in Fig. 2.26 concerning performance on the controlled rotor speed ω_r . The corresponding controlled inputs T_g , and ϑ , for both scenarios are illustrated in Fig. 2.27. As mentioned in Section 2.4, during the deloaded mode, thanks to the chosen deloading strategy, the WT is able to store a surplus of kinetic energy ΔW_k with respect to the classic mode of operation in the rotating masses. This could be released to the grid, if needed, at any moment, by bringing the rotor speed back to its MPPT value. The ΔW_k signals are shown in Fig. 2.28. These are similar because, for the chosen P_g^* , and wind speed signals, the controller finds $\omega_r^*(t) = \omega_{r,n} \forall t \geq 0$ via (2.16), for both scenarios. Eventually, Fig. 2.29 shows that the MPC controller succeeds in keeping the system trajectories within Λ_t , thus letting the WT avoid the singular points.



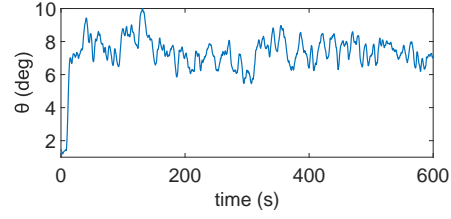
(a) Controlled T_g during a constant deloaded power reference.



(b) Controlled T_g during a varying deloaded power reference.

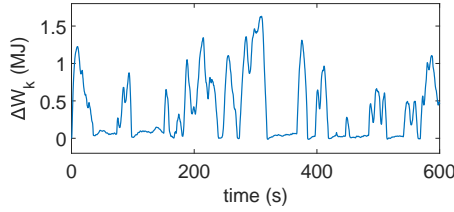


(c) Controlled ϑ during a constant deloaded power reference.

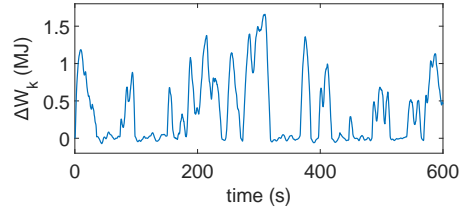


(d) Controlled ϑ during a varying deloaded power reference.

Figure 2.27: Controlled system inputs for constant and varying deloaded mode of functioning.



(a) Surplus of stored kinetic energy during a constant deloaded power reference.



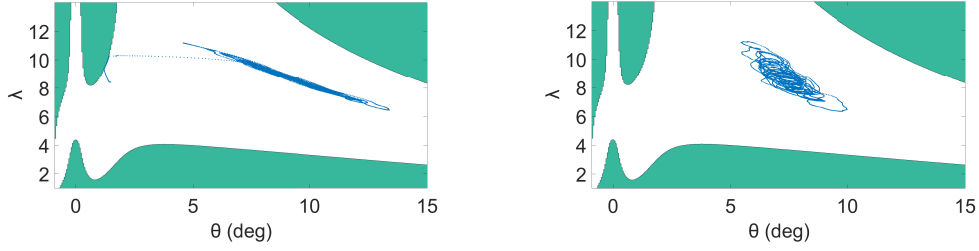
(b) Surplus of stored kinetic energy during a varying deloaded power reference.

Figure 2.28: Surplus of stored kinetic energy for constant and varying deloaded mode of functioning.

2.8 Conclusion and Future Perspectives

2.8.1 Conclusion

A novel approach to control a wind turbine for the general problem of tracking a given power reference was presented. Composing the two well-known techniques of FL and MPC showed clear benefit when treating a nonlinear system subject to physical constraints such as the wind turbine. The proposed controller lets the accomplishment of two main tasks, namely to work in classic MPPT/power limiting conditions for the entire operating envelope while allowing tracking of a general power reference when needed. We showed how the former task is necessary for important requirements as the satisfaction of grid constraints and the wind farm



(a) System trajectories in Λ_t during a constant deloaded power reference.

(b) System trajectories in Λ_t during a varying deloaded power reference.

Figure 2.29: System trajectories (blue dots). Thanks to the MPC constraints, these stay within Λ_t .

power maximization. As far as the former is concerned, we additionally saw that, by the chosen deloading strategy, a WT is able to store additional kinetic energy in the rotating masses. This happens to be useful to accomplish possibly required grid services such as frequency support. The proposed control scheme exhibits better performance when compared to the more classic approach of a PI-based controller. Moreover, Montecarlo simulation showed a certain inherent degree of robustness with respect to model-plant mismatches, as the one concerning the power coefficient, and parameter uncertainties.

As it has been showed, the FL+MPC scheme used for the specific case of controlling a WT is based on some approximations that lead to a convex quadratic problem to be solved on-line, thus allowing faster computation with respect to general nonlinear nonconvex programming problems.

2.8.2 Future Perspectives

In future work though, it would be interesting to consider additional problem approximations that could possibly allow to conclude on the mathematical stability of the closed-loop system, and lead to important MPC features such as the recursive feasibility of the optimization problem. For this last problem, a major issue is given by the presence of nonconvex constraints. In this sense, a possible way to be explored can be the one of considering a *scenario-based* FL+MPC approach according to the selected mode of functioning and the wind speed, possibly enabling a local convex scenario-invariant approximation of such constraints. Eventually, the proposed approach could be employed for a wider range of applications, e.g. for the reduction of the mechanical stress of a WT. For instance, being the pitch control active for all the operating points, the individual pitch control technique could be considered for such purpose, (see e.g. [Lio 2017]).

Part II

WIND FARM HIERARCHICAL CONTROL

Novel Distributed Optimization Algorithms

Contents

3.1	Finite-Time Average Consensus	70
3.1.1	Problem Formulation	70
3.1.2	Self-configuration	72
3.2	Particle Swarm Optimization	73
3.2.1	Basic Algorithm	73
3.2.2	Avoiding Premature Convergence	77
3.2.3	Constraints Handling	78
3.2.3.1	Problem Formulation and Available Solutions . .	78
3.2.3.2	Deb's rule	80
3.3	Gradient-based Distributed Optimization	81
3.4	Distributed Cooperative PSO: DPSO1	84
3.4.1	Related Works and Contribution	84
3.4.2	Problem Statement	85
3.4.3	Algorithm Description	86
3.4.3.1	Fitness Function	86
3.4.3.2	Variables Settings	86
3.4.3.3	Particles Update Law	87
3.4.3.4	Communication Settings	88
3.4.3.5	Distributed Functions Evaluation	89
3.4.3.6	Distributed Optimization Algorithm	90
3.4.4	Enhancement of Self-configuration Step	92
3.4.5	Conclusion	94
3.5	Distributed Cooperative PSO with reduced communication: DPSO2	94
3.5.1	Related Works and Contribution	94
3.5.2	Problem Statement	96
3.5.3	Dynamic Behavioral Analysis	97
3.5.3.1	CC-PSO Algorithm	97
3.5.3.2	Motivating Example	99
3.5.3.3	Damped CC-PSO Algorithm	100

3.5.4	Algorithm Description	103
3.5.4.1	Exploiting the Problem Structure	103
3.5.4.2	Role of Local Bests	104
3.5.4.3	Distributed Optimization Algorithm	105
3.5.5	Benchmark Test	106
3.5.5.1	Simulation Setup	106
3.5.5.2	Test results	108
3.5.6	Constrained Optimization Extension	111
3.5.7	Conclusion	112
3.6	Optimal Power Flow Application	113
3.6.1	Related Works and Contribution	113
3.6.2	Power System Model	114
3.6.2.1	Power Flow Equations	114
3.6.2.2	Power Flow Constraints	116
3.6.3	Distributed Optimal Power Flow	117
3.6.3.1	Problem Formulation	117
3.6.3.2	Application of DPSO1 and DPSO2	119
3.6.4	Conclusion	121

3.1 Finite-Time Average Consensus

THE aim of this section is to provide a description of the distributed *finite-time average consensus* algorithm as it is employed in the first distributed PSO, proposed in Section 3.4. A more detailed introduction to the general problem of consensus control is treated in Part III, as this is where consensus control techniques will be extensively used for the sake of controlling a wind farm. Moreover, the reader may refer to [Olfati-Saber 2007] for the basic important concepts of consensus algorithms. For some basic notions and preliminaries about graph theory please refer to Appendix C. The results of this section are mainly taken from [Dung 2013].

3.1.1 Problem Formulation

We consider a set of N agents each of which disposes of a personal piece of information $h_i \in \mathbb{R}$, called state. The agents can communicate on an undirected connected graph, and they are identified with the nodes of the communication graph \mathcal{G} . In the consensus problem we are interested in letting the agent agree on the state variable, i.e. to reach consensus. In other words, by means of local interaction on a given communication graph, and with no information relay, we aim at finding a distributed protocol, which makes $h_i, i = 1, \dots, N$ converge to a value satisfying the agreement condition $h_i = h_j \forall i, j$. Moreover, if the addressed problem is the *average*

consensus one, then by means of a distributed protocol we want the agents state to reach the value $\frac{1}{N} \sum_{i=1}^N h_{i,0}$, where we named $h_{i,0}$ agent i state at the initial time step $k = 0$. Among the possible distributed protocols letting the multi-agent system achieve average consensus, we are interested in those guaranteeing convergence in a *minimum number of finite steps* rather than asymptotic convergence. Thus, we restrict the attention on *finite-time* algorithms, i.e. those that make the system reach consensus in a finite number of steps. The work in [Dung 2013] is thus concerned with the design of such fast distributed protocol by means of *self-configuration*, i.e. the design step is solved autonomously in a distributed way by the agents themselves by exploiting the communication graph.

Let us pre-assign the following distributed law to the generic agent i of the system

$$h_i(k) = w_{ii}^k h_i(k-1) + \sum_{j \in \mathcal{N}_i} w_{ij}^k h_j(k-1), \quad h_i(0) = h_{i,0} \quad (3.1)$$

where $w_{ii}^k, \{w_{ij}^k | j \in \mathcal{N}_i\} \in \mathbb{R}, i = 1, \dots, N, \forall k > 0$ are the weights at time step k to be chosen, and where \mathcal{N}_i is the set of agent i neighbors. Difference equation (3.1) describes how h_i is updated at each step, based on only its local information, i.e. its neighbors state. The dynamics of the whole multi-agent system can be written in matrix form as

$$\mathbf{h}(k) = W_k \mathbf{h}(k-1), \quad \mathbf{h}(0) = \mathbf{h}_0$$

where we defined $\mathbf{h} \triangleq \text{col}(h_1, \dots, h_N)$, and \mathbf{h}_0 the corresponding initial state. Matrices W_k , which have to be tuned to solve the consensus problem, have a well-defined structure imposed by the adjacency matrix \mathcal{A} , (see Appendix C.1). Indeed W_k belongs to set of matrices that can be factorized as $W_k = Q_k \circ (I_N + \mathcal{A})$, where $Q_k \in \mathbb{R}^{N \times N}$ is an arbitrary matrix at time step k , I_N the identity matrix of dimension $\mathbb{R}^{N \times N}$, and \circ the Hadamard matrix product¹. Thus, the design problem can be reformulated as the one of finding a finite sequence of matrices $W_k, \{W_1, \dots, W_D\}$, belonging to the mentioned set, such that the system state reaches average consensus in a finite number of steps D , i.e.

$$\mathbf{h}(D) = J_N \mathbf{h}_0 \triangleq \frac{1}{N} \mathbf{1} \mathbf{1}^\top \mathbf{h}_0, \quad \forall \mathbf{h}_0 \in \mathbb{R}^N$$

where $\mathbf{1} \in \mathbb{R}^N$ is a column vector whose entries are all equal to 1. Since $\mathbf{h}(D) = \prod_{k=D}^1 W_k \mathbf{h}_0$, then the problem is to solve the following factorization

$$\prod_{k=D}^1 W_k = J_N \quad (3.2)$$

The minimum number of steps D required to solve factorization (3.2) is known to be $d(\mathcal{G}) \leq D \leq 2r(\mathcal{G})$, (see Appendix C.1). Intuitively, the lower bound is given by the fact the diameter of the graph characterizes the necessary steps for a given piece of information to reach all the agents in the network, and thus, it cannot be lower than $d(\mathcal{G})$. The upper bound is provided by [Hendrickx 2012].

¹Given $A = [a_{i,j}], B = [b_{i,j}] \in \mathbb{R}^{m \times n}$, then $A \circ B \triangleq [a_{i,j} b_{i,j}] \in \mathbb{R}^{m \times n}$.

3.1.2 Self-configuration

Among the results in [Dung 2013], the authors provide a distributed solution to the identified matrix factorization problem based on *gradient backpropagation* method. This is based on solving an optimization problem by exploiting known input-output learning sequences. Each agent i has Q couples of input-output values $\{h_{i,q}(0), y_{i,q}\}$, $q = 1, \dots, Q$, where $y_{i,q} = y_q = \frac{1}{N} \sum_{j=1}^N h_{j,q}(0)$. In order to estimate the D matrices W_k , $k = 1, \dots, D$, the factorization problem is reformulated as the minimization of the following quadratic error

$$E(W_1, \dots, W_D) \triangleq \frac{1}{N} \sum_{i=1}^N \sum_{q=1}^Q (h_{i,q}(D) - y_q)^2 \quad (3.3)$$

where the W_k matrices appear in $h_{i,q}(D)$ via (3.1). In order to see that (3.3) is equal to the square of the Frobenius norm² of the factorization error $\frac{1}{2} \left(\prod_{k=D}^1 W_k - J_N \right)$, first notice that

$$E(W_1, \dots, W_D) = \frac{1}{2} \left\| \prod_{k=D}^1 W_k H(0) - Y \right\|_F^2 \quad (3.4)$$

where we defined $H(0) \in \mathbb{R}^{N \times Q}$ the matrix whose entries are $h_{i,q}(0)$, and $Y \in \mathbb{R}^{N \times Q}$ the one whose entries are $y_{i,q}$. The identity between (3.3) and (3.4) can be seen by defining $\varepsilon_q(\mathbf{W})$ as

$$\varepsilon_q(\mathbf{W}) \triangleq \prod_{k=D}^1 W_k \mathbf{h}_q(0) - \mathbf{y}_q = \mathbf{h}_q(D) - \mathbf{y}_q$$

where $\mathbf{W} \triangleq (W_1, \dots, W_D)$, $\mathbf{h}_q(k) \triangleq \text{col}(h_{1,q}(k), \dots, h_{N,q}(k))$, $\mathbf{y}_q \triangleq y_q \mathbf{1}$. Thus, we have that

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^N \sum_{q=1}^Q (h_{i,q}(D) - y_q)^2 &= \frac{1}{2} \sum_{q=1}^Q \varepsilon_q(\mathbf{W})^\top \varepsilon_q(\mathbf{W}) = \frac{1}{2} \sum_{q=1}^Q \text{tr} \left(\varepsilon_q(\mathbf{W}) \varepsilon_q(\mathbf{W})^\top \right) = \\ &= \frac{1}{2} \text{tr} \left(\sum_{q=1}^Q \left(\varepsilon_q(\mathbf{W}) \varepsilon_q(\mathbf{W})^\top \right) \right) = \frac{1}{2} \text{tr} \left((H(D) - Y)(H(D) - Y)^\top \right) \end{aligned}$$

Notice that $Y = J_N H(0)$, and that the input learning sequence can be chosen such that $H(0)H(0)^\top = I_N$, for instance by selecting the vectors of the canonical basis of \mathbb{R}^N as inputs. Thus, $E(W_1, \dots, W_D) = \frac{1}{2} \left\| \prod_{k=D}^1 W_k - J_N \right\|_F^2$, meaning that (3.4)

²Recall that the Frobenius norm of a real matrix A is defined as $\|A\|_F \triangleq \sqrt{\text{tr}(AA^\top)}$, where $\text{tr}(\cdot)$ is the *trace* operator. The trace of a matrix $B \in \mathbb{R}^{n \times n}$ is defined as $\text{tr}(B) \triangleq \sum_{i=1}^n b_{ii}$, being b_{ii} the diagonal entries of B .

is equivalent with minimizing the square of the Frobenius norm of the factorization error. All in all, the optimization problem to be solved in a distributed way is

$$\min_{\{W_k, k=1, \dots, D\}} \frac{1}{2} \sum_{q=1}^Q \text{tr} \left(\varepsilon_q(\mathbf{W}) \varepsilon_q(\mathbf{W})^\top \right) \quad (3.5)$$

By naming $E_q(\mathbf{W}) \triangleq \frac{1}{2} \text{tr} \left(\varepsilon_q(\mathbf{W}) \varepsilon_q(\mathbf{W})^\top \right)$, problem (3.5) can be solved via a gradient descent method, where the generic matrix W_k is updated via

$$W_k(m+1) = W_k(m) - \alpha_c \sum_{q=1}^Q \frac{\partial E_q(\mathbf{W})}{\partial W_k}$$

where $\alpha_c \in]0, 1[$ to guarantee convergence to a local minimum, and where m stands for the m -th iteration of the optimization process. In [Dung 2013] though, a stochastic gradient one is selected, i.e. the gradient of the cost function is approximated by the gradient at a single input-output sequence, yielding

$$W_k(m+1) = W_k(m) - \alpha_c \frac{\partial E_q(\mathbf{W})}{\partial W_k} \quad (3.6)$$

In the mentioned reference it is shown that the derivatives of $E_q(\mathbf{W})$ can be computed via

$$\frac{\partial E_q(\mathbf{W})}{\partial W_k} = \delta_{k,q} \mathbf{h}_q^\top(k-1), \quad k = 1 \dots, D$$

where $\delta_{D,q} \triangleq \mathbf{h}_q(D) - \mathbf{y}_q$, i.e. the difference between the actual output $\mathbf{h}_q(D)$, and the desired one \mathbf{y}_q , and where $\delta_{k-1,q} = W_k^\top \delta_{k,q}$, $k = 1 \dots, D$. The overall update law for (3.6) can be rewritten as

$$W_k(m+1) = W_k(m) - \alpha_c \delta_{k,q(m)} \mathbf{h}_{q(m)}^\top(k-1) \quad (3.7)$$

where $q(m)$ identifies the q -th sequence obtained via the system matrices $W_k(m)$, $k = 1 \dots, D$. Update law (3.7) can be written entry-wise as

$$w_{ij}^k(m+1) = w_{ij}^k(m) - \alpha_c \delta_{i,q} h_j(k-1) \quad (3.8)$$

where $\delta_{i,k}$ is the i -th entry of $\delta_{k,q(m)}$, and $h_j(k-1)$ the j -th entry of $\mathbf{h}_{q(m)}(k-1)$. In a gradient backpropagation fashion, the optimization problem is solved in a distributed way by alternating the following steps. Firstly, the learning sequence is propagated forward in the network. Then, the error between the actual output and the target one is computed and backpropagated. The overall distributed self-configuration algorithm is described in Algorithm 1.

3.2 Particle Swarm Optimization

3.2.1 Basic Algorithm

Particle swarm optimization is a stochastic optimization method, belonging to the class of nature-inspired metaheuristics, and it was first introduced in the pioneer

Algorithm 1 Finite-time average consensus self-configuration, [Dung 2013]

Output: Matrix sequence W_k , $k = 1, \dots, D$

Initialization :

- 1: Number of steps D , number of patterns Q
- 2: Learning sequence $\{h_{i,q}(0), y_q\}$, where $y_q = \frac{1}{N} \sum_{i=1}^N h_{i,q}(0)$, $i = 1, \dots, N$, $q = 1, \dots, Q$
- 3: Randomly initialize $W_k(0)$, $W_k(-1)$, $k = 1, \dots, D$
- 4: Choose the learning rate α_c
- 5: Select threshold γ_c , and $E_i > \gamma_c$
- 6: Set $m = 0$

LOOP Process :

- 7: **while** $E_i > \gamma_c$ **do**
- 8: **for** $q = 1$ to Q **do**
- 9: Select the learning sequence $h_i(0) \triangleq h_{i,q}(0)$, $\bar{h} \triangleq y_q$
- 10: Learning sequence propagation

$$h_i(k) = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij}^k(m) h_j(k-1), \quad k = 1, \dots, D$$

- 11: Error computation $\delta_{i,D} = h_i(D) - \bar{h}$, $e_{i,q} \triangleq \delta_{i,D}^2$
- 12: Error propagation

$$\delta_{i,k-1} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ji}^k(m) \delta_{j,k}, \quad k = D, \dots, 2$$

- 13: For $k = 1, \dots, D$, $i = 1, \dots, N$, and $j \in \mathcal{N}_i \cup \{i\}$, update matrices via (3.8)
 - 14: $m = m + 1$
 - 15: **end for**
 - 16: $E_i = \frac{1}{N} \sum_{q=1}^Q e_{i,q}$
 - 17: **end while**
 - 18: **return** W_k , $k = 1, \dots, D$
-

work of [Eberhart 1995].

Let us consider the generic optimization problem

$$\min_x f(x) \tag{3.9}$$

where $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \rightarrow \mathbb{R}$. PSO is concerned with finding a solution to (3.9), possibly the global minimizer of f , via an evolutionary algorithm. Moreover, belonging to the class of metaheuristic optimization, PSO does not exploit any prior knowledge of the problem structure in order to search for a solution. The only requirement is that we have to be able to *evaluate* f in its domain. This is why

this algorithm is also *derivative-free*. Moreover, under the stated only requirement, the cost function is not restricted to any particular form, i.e. it does not even have to be in analytic closed form. This opens up to a wide range of applications, e.g. simulation-based optimization. For instance, authors of [Sandou 2008] capitalize on PSO for the sake of autonomously tune an \mathcal{H}_∞ -based controller for a LTI system. In order to solve problem (3.9), the classic PSO algorithm of [Eberhart 1995] is defined as follows. Let us consider a set of N_p particles $p = 1, \dots, N_p$, in the search space, i.e. each particle represents a position $x_p \in \mathbb{R}^n$. During the performance of the algorithm, these are allowed to move in the search space according to a prescribed law of evolution, and they update and store in memory their latest *personal best* visited position b_p in respect of the cost function f to be minimized. Moreover, they have access to the other particles personal best and the according cost function value. On the basis of this knowledge then each particle is able to elect the best particle among the swarm particle bests, i.e. the global best g . By defining a *speed* variable s_p , the generic particle of the swarm evolves according to the following discrete time system.

$$\begin{cases} s_p(k+1) &= \omega s_p(k) + \phi_{1,p}(k)(g(k) - x_p(k)) + \phi_{2,p}(k)(b_p(k) - x_p(k)) \\ x_p(k+1) &= x_p(k) + s_p(k+1) \end{cases} \quad (3.10)$$

Where, as stated above

$$b_p(k) \triangleq \arg \min_{x \in \{x_p(0), \dots, x_p(k)\}} f(x)$$

while $g(k)$ is the *global best* of the swarm at the current step k , i.e.

$$g(k) \triangleq \arg \min_{x \in \{b_1(k), \dots, b_{N_p}(k)\}} f(x)$$

$\omega \in \mathbb{R}$ is called the *inertia factor*, and where $\phi_{1,p}$ and $\phi_{2,p}$ are two aleatory variables with uniform distribution of probability in the respective intervals $[0, c_1]$, and $[0, c_2]$, where $c_1, c_2 \in \mathbb{R}^+$. PSO algorithm (3.10) thus realizes at each iteration step, and for each particle, an aleatory linear combination of three trends, namely to follow its path, to retrace its steps towards its personal best, to move towards the global best current value. An illustration of the dynamic behavior of (3.10) is given in Fig. 3.1. We additionally consider lower and upper bounds of the variables of the form: $\underline{x} \leq x \leq \bar{x}$, by limiting the maximal speed of each particle, and forcing x_p to stay within the aforementioned bounds. This is simply obtained by adding the following to (3.10).

$$s_p(k+1) \triangleq \max\{\min\{s_p(k+1), \bar{s}\}, -\bar{s}\} \quad (3.11)$$

$$x_p(k+1) \triangleq \max\{\min\{x_p(k+1), \bar{x}\}, \underline{x}\} \quad (3.12)$$

Where we place respectively (3.11) right after the first equation in (3.10), and (3.12) after the second equation in (3.10), and where $\bar{s} \triangleq 1/2(\bar{x} - \underline{x})$. The overall algorithm is thus shown in Algorithm 2, where `max_iter` is the max iteration step to be set. The

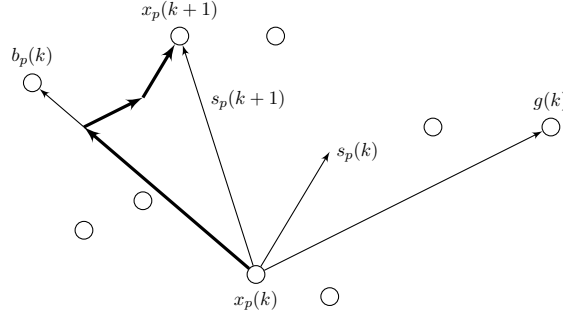


Figure 3.1: An illustration of PSO classic dynamics of system (3.10).

Algorithm 2 Classic PSO algorithm**Output:** Global best: g *Initialization :*

- 1: Randomly initialize $x_p \in [\underline{x}, \bar{x}]$, $s_p \in [-\bar{s}, \bar{s}]$, $p = 1, \dots, N_p$
- 2: $b_p = x_p$, $p = 1, \dots, N_p$
- 3: Evaluate $f_p^b \triangleq f(b_p)$, $p = 1, \dots, N_p$
- 4: Initialize global bests $g = \arg \min_{\{b_p\}} \{f_p^b\}$

LOOP Process

- 5: **for** $k = 1$ to max_iter **do**
- 6: Evaluate the particles in f : $f_p^x \triangleq f(x_p)$, $p = 1, \dots, N_p$
- 7: update personal bests for $p = 1, \dots, N_p$

$$(b_p^{\text{new}}, f_p^{b,\text{new}}) = \begin{cases} (x_p, f_p^x) & \text{if } f_p^x < f_p^b \\ (b_p, f_p^b) & \text{otherwise} \end{cases}$$

- 8: $b_p = b_p^{\text{new}}$; $f_p^b = f_p^{b,\text{new}}$
- 9: update global bests $g = \arg \min_{\{b_p\}} \{f_p^b\}$
- 10: perform (3.10), (3.11), (3.12)
- 11: **end for**
- 12: **return** g

choice of the parameters is one first important step in the tuning of the algorithm as they highly influence its dynamic behavior, i.e. the stability of its trajectories and their behavior during the transient, and the quality of the solution to which particles converge. As far as the latter issue is concerned, since PSO does not guarantee the convergence to the global minimizer of f , we are generally interested in ensuring the achievement of a *good-enough* solution to (3.9). A standard parameters setting often employed in the literature, and validated by [Kennedy 2006], is given by

- $N_p = 10 + \sqrt{n}$

- $\omega = \frac{1}{2 \ln 2}$
- $c_1 = c_2 = \frac{1}{2} + \ln 2$

More in general, plenty of works have been carried out in order to give a parameters setting that could at least guarantee the stability of (3.10), i.e. the particles convergence to a common value. A first work in this sense is provided by [Trelea 2003] where, based on a deterministic analysis of (3.10), the author gives some insight on how the dynamic behavior of the system is influenced by the parameter tuning. Authors of [Kadirkamanathan 2006] suggest a system analysis based on the well-known Lure's stability problem, under the simplifying assumption that personal bests b_p , $p = 1, \dots, N_p$ and global best g are time-invariant. To give a further interesting example, we can cite the work of [Pan 2014]. Here, since the stability problem concerns the convergence to a common swarm value, the PSO algorithm is analyzed from the perspective of stochastic consensus. Thus, under chosen negative values of the inertia factor ω , the authors give conditions under which the swarm converges to an aleatory variable with defined expected value and variance.

3.2.2 Avoiding Premature Convergence

The classic PSO of Algorithm 2 can be affected by the problem of premature convergence. This phenomenon, also known as *stagnation*, occurs when b_p and g do not change over a prolonged period of time. In this case, according to (3.10), the best particle speed would rapidly go to zero, i.e. it would give up its research, [Clerc 2006]. As it is shown by system (3.10), PSO is based on two main behavioral trends, namely *exploitation* of currently known information, i.e. the discovered b_p , and g , and *exploration* of the unvisited areas of the search space. Thus, it is clear that if a PSO algorithm suffers from premature convergence, the exploration trend should be boosted. This can be achieved either via a different tuning of the parameters, or via other algorithm modifications, (see for instance [van den Bergh 2002]). However, a first enhancement can be simply achieved by letting each particle have only access to the personal bests values of the particles belonging to a defined restricted subset \mathcal{S}_p of the swarm. In this case, each particle performs a modified Step 9 of Algorithm 2, that is, computing *its own* knowledge of global best, which is *local* because restricted to the mentioned subset. The modified Step 9 takes the form of

$$l_p \triangleq \arg \min_{\{b_p \in \mathcal{S}_p\}} \{f_p^b\}$$

where we named l_p the new *local best* to depict the concept of local information, replacing the global one, (see e.g. [Aguirre 2007]).

There exist several techniques in the literature to define the neighborhood \mathcal{S}_p . In this work though, we provide the one proposed by [Aguirre 2007], named *singly-linked list*, as it shows good convergence performance. First of all the size of the neighborhood must be set via the introduction of a new parameter, N_m . For $N_m =$

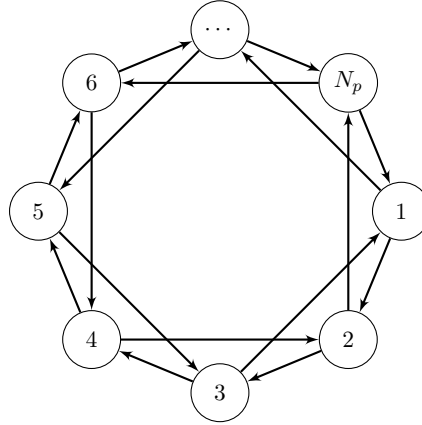


Figure 3.2: Singly-linked list definition for $N_m = 2$. Each particle p gives access to its information to particles $p + 1$, and $p - 2$.

Algorithm 3 Singly-linked list, [Aguirre 2007].

Output: \mathcal{S}_p

Initialization :

1: Set $\mathcal{S}_p = \emptyset$

2: Set N_m

3: **step** = 1

4: **switch** = 1

LOOP Process

5: **while** neighborhood_size < N_m **do**

6: Include the particle labeled $p + \text{switch} \cdot \text{step}$ in \mathcal{S}_p

7: **step** = **step** + 1

8: **switch** = -**switch**

9: **end while**

10: **return** \mathcal{S}_p

2, the mentioned technique associates a concatenated communication among the particles as shown in Fig. 3.2. For a general size N_m , for each particle p , the singly-linked list neighborhood can be obtained via the pseudocode shown in Algorithm 3.

3.2.3 Constraints Handling

3.2.3.1 Problem Formulation and Available Solutions

With the exception of *box* constraints, which can be handled in the PSO framework via (3.11), and (3.12), PSO was first introduced to solve *unconstrained* optimization problems of the form of (3.9), for which generally it exhibits good performance.

Let us now consider the following *constrained* optimization problem

$$\begin{aligned}
 & \min_x f(x) \\
 & \text{subject to} \\
 & g(x) \leq 0 \\
 & h(x) = 0
 \end{aligned} \tag{3.13}$$

where the optimization variable and the cost function are defined in Subsection 3.2.1, and where, for the sake of generality, functions $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $h : \mathbb{R}^n \rightarrow \mathbb{R}^q$ are considered nonlinear, and may define a nonconvex feasible set. As it is well-known, one of the major difficulties when employing PSO to solve optimization problem (3.13) is the lack of an explicit method to direct the optimum search towards the feasible region, [Aguirre 2007]. In classic optimization algorithms, constraints handling methods can be classified in two groups [Deb 2000], namely *generic* methods, which do not exploit the mathematical structure of the constraints, and *specific* methods, which do exploit the structure but are only applicable to a specific class of constraints. Within the former group, one would usually capitalize on penalty function, Lagrange multiplier techniques, etc. As far as the latter one is concerned, typical approaches include cutting plane, reduced gradient, gradient projection ones, etc. [Deb 2012]. When employing a metaheuristic optimization algorithm, such as PSO, though, in which the optimization problem is treated as a *black box*, one tends to capitalize on generic methods to treat the constraints. One of the most successful techniques belonging to this class is given by the well-known *Deb's rule*, whose principle is presented in Subsection 3.2.3.2. However, in addition to the chosen constraint handling method, when incorporating constraints into the cost function of an evolutionary algorithm, it is particularly important to maintain *diversity* in the swarm, and to be able to keep solutions both inside and outside the feasible region, [Cabrera 2007]. The diversity of the swarm can be interpreted as a measure of the degree of exploration and exploitation ability of the algorithm, via the particles position and speed distribution, [Cheng 2013]. Keeping a sufficient level of diversity thus means to let the swarm have a good balance between exploration and exploitation in order to avoid premature convergence. The works in the literature are thus mainly concerned with solutions combining a generic method to handle constraints, and techniques to preserve the diversity of the swarm. Works of [Aguirre 2007, Pulido 2004, Cabrera 2007, Liu 2010] are based on similar ideas of combining the mentioned Deb's rule with different techniques concerning the diversity problem. In particular, authors of [Aguirre 2007] propose to add a dynamic tolerance to handle inequality constraints, and the storage of tolerant particles, i.e. laying in the feasible region, to enhance the constraints handling. Perturbation operators are then defined to keep diversity. In [Pulido 2004], a turbulence operator is employed to perturb the particles speed, and in [Cabrera 2007], the authors suggest to use a combined action of algorithm reinitialization and mutation operator. Finally [Liu 2010] propose a combination of PSO and differential evolution algorithm to help preserving the diversity. A similar approach to the aforementioned ones is

shown in [Venter 2010] where, focusing on the only inequality constraints, it is proposed to transform the optimization problem in a bi-objective one in order to treat the constraints. Thus the problem is solved in the framework of multi-objective optimization by searching the Pareto front. Authors of [Ali 2014] provide a test on the penalty function approach, in both its static and dynamic variant, for the electromagnetic-like method of optimization. The penalty function technique is concerned with introducing the problem constraints in the cost function, and by weighting them via a tunable penalty gain, which, for the dynamic case, can be made iteration-varying. The resulting optimization problem is thus unconstrained. Nonetheless, such approach exhibits some important pitfalls that limit its applicability. Indeed, it generally introduces a cost function *distortion* if the penalty function is not applied in the only unfeasible domain, and for which usually the global minimizer of the original problem no longer preserves the same stationary property in the modified one. PSO is combined with the augmented Lagrange multiplier method and with a nonstationary penalty function in [Sedlaczek 2005]. If, for the sake of brevity, we consider the only inequality constraints in (3.13), authors of [Sedlaczek 2005] consider the augmented Lagrange function

$$L_A(x, \lambda, r) \triangleq f(x) + \lambda g(x) + rg(x)^\top g(x)$$

obtained by adding the quadratic extension $rg(x)^\top g(x)$ to the Lagrange function, and where $\lambda \in \mathbb{R}^m$ having all strictly positive components, $r \in \mathbb{R}$. Thus, they suggest to alternate the classic update of the Lagrange multiplier λ , and an heuristic update for r , with the classic PSO equations. An example of a specific method coupled with PSO can be found in [Wakasa 2015a], where the authors make use of a projection method to deal with convex constraints, and a modified one to treat non-convex constraints. Eventually, a mixed generic and specific approach is proposed in [Wakasa 2015b], where the authors capitalize on primal-dual decomposition method. Here, PSO equations are employed for the primal variable update, combined with a projection method to satisfy the convex constraints.

3.2.3.2 Deb's rule

Having made no constraints assumption in optimization problem (3.13), we are interested in generic methods to handle them. In particular, in the following we provide the mentioned Deb's rule combined with the basic PSO algorithm. This was first introduced by [Deb 2000], who applied the proposed technique for the genetic algorithm. This technique exhibits some interesting properties which make it eligible for the application of plenty constrained optimization problems. Indeed, even if belonging to the penalty function approaches, it does not require any penalty parameter. Moreover, the aforementioned rule allows avoiding any cost function distortion that may occur when incorporating the constraints in the problem via penalty functions. Deb's rule consists of a tournament selection in which, when comparing two solutions of (3.13), the following criteria is adopted

- any feasible solution is preferred to any infeasible solution

- among two feasible solutions, the one having better objective function value is preferred
- among two infeasible solutions, the one having smaller constraint violation is preferred

In addition, equality constraints are handled via a transformation into inequality constraints with the introduction of a positive threshold $\epsilon \in \mathbb{R}^+$ such that $|h(x)| - \epsilon \leq 0$, $i = 1, \dots, N$, where ϵ is a vector of proper dimension with all its entries equal to ϵ . For ease of notation we define $\tilde{g}(x) \triangleq \text{col}(g(x), |h(x)| - \epsilon)$, where \tilde{g} is a vector of functions of dimension $m + q$.

In order to implement Deb's rule the agents have to evaluate a modified cost function, called *fitness* function, which takes into account the tournament selection described above. Thus, instead of $f(x)$, the particles have to be evaluated in the following

$$F(x) \triangleq \begin{cases} f(x) & \text{if } \tilde{g}(x) \leq 0 \\ f_{max} + \sum_{i=1}^{m+q} \chi(\tilde{g}_i(x)) & \text{otherwise} \end{cases} \quad (3.14)$$

being \tilde{g}_i the i -th component of \tilde{g} , and where $\chi : \mathbb{R} \rightarrow \mathbb{R}$ is defined as

$$\chi(y) \triangleq \begin{cases} y & \text{if } y > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

The second term of the second case of (3.14) accounts for the degree of constraint violation, and it is called *sum of constraints*. $f_{max} \in \mathbb{R}^+$ is introduced so that the feasible solutions are always preferred to the infeasible ones. It is set as the objective function value of the worst feasible current solution in the swarm. If there is no feasible solution, then it is simply set as 0. f_{max} thus needs to be updated each time that new feasible solutions are found. However, for an easier implementation, if some little prior information of the involved function of (3.13) is known, then f_{max} can be set as a constant sufficiently high value, in order to guarantee that the first case of (3.14) has always lower value with respect to the second one. Algorithm 2 is thus modified by substituting the f -evaluations with F .

Remark 3.1 *As shown in [Aguirre 2007] the tolerance ϵ used to handle the equality constraints can be made iteration-varying to let better search space exploration. Typically a linearly decreasing value is employed. This can be done by changing its value at each iteration of modified Algorithm 2.*

3.3 Gradient-based Distributed Optimization

In this section we aim at providing a brief introduction to the problem of distributed optimization, and the main concepts of the available solutions in the literature. In particular, we focus here on gradient-based algorithms as, even if we will not make use of them for the WF optimization problem, we believe that it is useful to have

an all-round view of the distributed optimization problem in order to contextualize our work on evolutionary algorithms of Sections 3.4, and 3.5.

When dealing with multi-agent systems on large size networks, it might be too computationally demanding to consider standard centralized optimization techniques. Moreover, if the global network topology of the system is not known, it might even be unfeasible. The common point among the available works on gradient-based distributed optimization is to blend consensus control techniques with classic convex optimization algorithms. These are usually based on Lagrange multipliers, primal-dual methods, augmented Lagrange function, etc. The dissertation of these well-known notions and techniques goes beyond the scopes of this work. The reader may refer to [Boyd 2004] for an introduction to convex optimization. The basic optimization problem treated in the literature, from which all its variants are obtained, is the *sum of local cost functions*, and it can be stated as follows. Consider a set of N agents at the nodes of a connected graph \mathcal{G} , then the problem to be solved is

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^N f_i(x) \quad (3.16)$$

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function, only available at node i . Notice also that, in this general formulation, each local cost function f_i depends on the whole optimization variable. In other words, agents of the system share a common optimization variable. Thus a typical choice in the literature is to endow each agent with a local estimation $x_i \in \mathbb{R}^n$ of the optimization variable, even if f_i does not necessarily depends on the whole x . More refined solutions are available though, [Wang 2011a]. One of the first work, concerned with solving (3.16) in a distributed way, can be found in [Nedic 2009], where the generic agent i estimation of the optimization variable is updated according to

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j(k) - \beta_i d_i(k) \quad (3.17)$$

where w_{ij} are properly chosen weights on the communication graph edges, β_i a tunable parameter, and $d_i(k)$ the subgradient of f_i evaluated at $x_i(k)$. As one can see, (3.17) is given by a combined action of consensus, i.e. averaging with local neighbors in order to meet condition $x_i = x_j \forall i, j \in \mathcal{V}$, in the first term on the right side, and a contribution towards the descent direction of f_i in the second term, obtained exploiting gradient information of f_i . The algorithm is shown to converge to an approximated optimal solution of (3.16), according to a trade-off between accuracy and speed of convergence, managed via the tunable parameters. Such algorithm is analyzed in [Kvaternik 2011] by a control theory perspective, where the authors show that each x_i converges to a neighborhood of the optimal solution, which can be made little at will by acting on β_i . A first variation of (3.16) is given in [Sundhar Ram 2012], where a common convex constrained set is considered, and the global cost function is modified as $h\left(\sum_{i=1}^N f_i(x)\right)$, where $h : \mathbb{R} \rightarrow \mathbb{R}$ is known by every agent. Constrained optimization is then treated via projection method.

Function h basically brings an additional coupling term among the agents, and to solve the optimization problem, differently from (3.17), the global cost function needs to be estimated at each algorithm step. This is managed by employing an additional variable for each agent, which is updated in parallel with x_i , according to average consensus techniques.

More recently, problem (3.16) has been treated in the primal-dual framework, via an interesting alternative formulation, (e.g. [Wang 2010, Wang 2011a]). Indeed, condition $x_i = x_j \forall i, j \in \mathcal{V}$ can be included in the optimization problem as an equality constraint, yielding the equivalent problems

$$\begin{aligned} \min_x \sum_{i=1}^N f_i(x) &= \min_{\{x_i, i=1, \dots, N\}} \sum_{i=1}^N f_i(x_i) &= \min_{\{x_i, i=1, \dots, N\}} \sum_{i=1}^N f_i(x_i) \\ &\text{subject to } x_i = x_j \quad \forall i, j \in \mathcal{V} &\text{subject to } (\mathcal{L} \otimes I_n) \mathbf{x} = \mathbf{0} \end{aligned} \quad (3.18)$$

where \otimes is the Kronecker product (see Appendix A.8), I_n the identity matrix of dimension n , $\mathbf{x} \triangleq \text{col}(x_1, \dots, x_N) \in \mathbb{R}^{Nn}$, $\mathbf{0}$ a vector of same dimension with all its entries equal to 0, and where $\mathcal{L} \in \mathbb{R}^{N \times N}$ is the Laplacian matrix associated to \mathcal{G} . From Appendix C, we know that under some graph connectivity assumptions $\mathcal{L}\mathbf{1}_N = \mathbf{0}_N$, i.e. 0 is an eigenvalue of \mathcal{L} with corresponding eigenvector equal to $\mathbf{1}_N$. Thus, constraint $(\mathcal{L} \otimes I_n) \mathbf{x} = \mathbf{0}$, if satisfied, implies that $\mathbf{x} \in \text{span}(\mathbf{1}_N \otimes v)$, where $v \in \mathbb{R}^n$ is an arbitrary vector, i.e. $x_i = x_j \forall i, j \in \mathcal{V}$. The last problem formulation of (3.18) on the right is useful because, by writing the associated Lagrange function

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^N f_i(x_i) + \boldsymbol{\lambda}^\top (\mathcal{L} \otimes I_n) \mathbf{x}$$

with $\boldsymbol{\lambda} \triangleq \text{col}(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{Nn}$, the problem can be solved via

$$\begin{cases} \dot{\mathbf{x}} = -(\mathcal{L} \otimes I_n) \boldsymbol{\lambda} - G(\mathbf{x}) \\ \dot{\boldsymbol{\lambda}} = (\mathcal{L} \otimes I_n) \mathbf{x} \end{cases} \quad (3.19)$$

where $G(\mathbf{x})$ is a concatenation of g_i , i.e. the gradient of f_i , and for which it is easy to derive the discrete time iterative implementable version [Wang 2010]. System (3.19) describes the dynamics of the overall multi-agent system. The local law of the generic agent i is thus simply obtained from (3.19) yielding

$$\begin{cases} \dot{x}_i = - \sum_{j \in \mathcal{N}_i} a_{ij} (\lambda_i - \lambda_j) - g_i(x_i) \\ \dot{\lambda}_i = \sum_{j \in \mathcal{N}_i} a_{ij} (x_i - x_j) \end{cases}$$

Eventually, one efficient distributed algorithm often employed in the literature is the Alternating Direction Method of Multipliers (ADMM), which is obtained by defining the augmented Lagrange function associated to the last problem formulation of (3.18), e.g. [Zhang 2014].

Belonging to gradient-based optimization algorithms, the aforementioned approaches

inherit the pitfalls of centralized convex optimization, which may result in suboptimality of the solution when treating nonconvex problems, and it basically limits its application to problems that can be described via analytic smooth functions. These are among the major motivations that lead us to the development of the algorithms object of the next sections.

3.4 Distributed Cooperative PSO: DPSO1

3.4.1 Related Works and Contribution

Since its first formulation [Eberhart 1995], particle swarm optimization algorithm has been widely studied, and a great number of modified versions has been proposed, especially for *centralized* problems (see e.g. [Del Valle 2008]). Nonetheless PSO still gains great attention, mainly due to the research effort to apply this successful algorithm to complex engineering applications. Among them, large scale multi-agent systems certainly provide new opportunities for the engineering community as they represent the most natural way of treating many problems encountered in the domains of telecommunication, transports, power systems, etc. Moreover the requirements for such systems to be scalable, modular, and resilient, logically imposes the *distributed* framework as the one to be considered when dealing with them. This motivates the increasing interest for distributed optimization techniques to which distributed PSO (DPSO) belongs.

In this regard, numerous solutions are available in the literature. However, in order to be consistent with the definitions that we will use in the sequel, and with the problem addressed, we have to make a distinction clear. DPSO, and distributed evolutionary algorithms more in general, are often related to *parallel* computation in order to speed up the convergence to a solution. In this case the optimization problem is usually centralized, and it is split among several units. In other words, distributing the algorithm is a *choice*, [Rivera 2001]. We will refer to parallel PSO (PPSO) when dealing with this class.

The problem we want to address though, is concerned with those systems that are *physically* distributed, i.e. distribution is not a parameter of the algorithm. This is, again, the case of multi-agent systems. In this second class of distributed systems, typically each agent would be endowed with some *private* parameters, e.g. personal cost functions, and would *cooperate* with the other agents to solve a common optimization problem, which is usually given by the sum of the personal cost functions. We will refer to DPSO to indicate the PSO algorithm applied to this class of systems. In the literature, there exist only a few works belonging to this latter class of algorithms. These usually combine PSO, and consensus techniques. In [Wakasa 2015a] each agent has knowledge of its own cost function that depends on only its own optimization variable, i.e. they do not share a common variable. Coupling among the agents is then given by a common objective function known by them all. A modified consensus technique, based on the work of [Sundhar Ram 2012] is employed to estimate the sum of local cost functions at each step of PSO. Un-

fortunately such estimation may fail to be sufficiently accurate to guarantee proper convergence of the algorithm when the agents do share common variable. This issue is treated in more details in Subsection 3.4.3. Authors of [Wakasa 2015b] propose a distributed primal-dual optimization method, based on the work of [Chang 2014], where the primal variable update, usually provided by sub-gradient methods, is replaced by the PSO algorithm. As in [Wakasa 2015a], agents do not share common variables, and the only coupling among them is given by the requirement of satisfying a common inequality constraint obtained by the sum of local private constraint functions. Consensus technique is thus employed to reach an agreement among the agents on the dual optimization variable as well as to estimate the sum of local constraints.

Examples of DPSO in which the agents share a common optimization variable, i.e. their local cost function depends also on the other agents variables, can be found for instance in [Gazi 2014, Navarro 2015]. However, they are both specific to the problem they address, and they are not readily extendable to a more general class of optimization problems. In particular, [Gazi 2014] capitalizes on DPSO to solve a distributed agreement problem, while in [Navarro 2015] the technique is used to tune the controllers of cooperative robots.

In this section we propose a first novel approach to DPSO, which we name *DPSO1* and that allows a more general problem formulation and applies to a class of distributed constrained optimization problems. The algorithm is mainly concerned with reproducing, in a distributed way, the exact behavior of a centralized PSO. This is made possible by combining it with the means of average consensus techniques. Moreover, in order to provide a fully-distributed solution, we capitalize on the finite-time average consensus self-configuration shown in Section 3.1. Concerning the centralized PSO that we aim to reproduce, we choose the work of [Aguirre 2007], i.e. one of the PSO algorithms available in the literature, handling the optimization problem constraints via Deb's rule.

3.4.2 Problem Statement

Consider a group of N agents, each of which disposes of a private control variable $x_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$, private cost function, inequality and equality constraints. By *private* we mean that they are only known by the associated agent i . We provide the following useful definition.

Definition 3.1 In a constrained optimization problem, an agent i is *physically* coupled to j if at least one among its private cost function, inequality and equality constraints depends on agent j private control variable. Agent j is thus said to be a *physical neighbor* of agent i .

We then consider a graph $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{E}_p)$ that keeps track of the physical relations among the agents. In particular $\mathcal{V}_p = \{1, \dots, N\}$ is the set of the agents, i.e. the nodes of the graph, and $\mathcal{E}_p \subseteq \mathcal{V}_p \times \mathcal{V}_p$ is the set of edges among them, where the edge $(i, j) \in \mathcal{E}_p$ if and only if agent j is a *physical neighbor* of agent i . Note

that according to the given definition of physical neighbor, \mathcal{G}_p generally defines a *digraph*, i.e. the graph edges have a direction associated to them. The set of physical neighbors of agent i is defined as $\mathcal{N}_i^p \triangleq \{j \in \mathcal{V}_p : (i, j) \in \mathcal{E}_p\}$. By defining $\mathbf{x}_{ij} \triangleq \{x_j \in \mathbb{R}^{n_j} : j \in \mathcal{N}_i^p\}$ as the set of physical neighbors variables of agent i , we are able to represent the generic agent i cost function, inequality, and equality constraints respectively as $f_i(x_i, \mathbf{x}_{ij})$, $g_i(x_i, \mathbf{x}_{ij}) \leq \mathbf{0}$, and $h_i(x_i, \mathbf{x}_{ij}) = \mathbf{0}$, where $f_i : \mathbb{R}^{n_i + \sum_{j \in \mathcal{N}_i^p} n_j} \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^{n_i + \sum_{j \in \mathcal{N}_i^p} n_j} \rightarrow \mathbb{R}^{m_i}$, $h_i : \mathbb{R}^{n_i + \sum_{j \in \mathcal{N}_i^p} n_j} \rightarrow \mathbb{R}^{q_i}$, and where $\mathbf{0}$ is a vector of proper dimension with all zero entries. Note that no assumptions were made concerning these functions. We are now able to state the optimization problem that agents have to *cooperatively* solve, as

$$\begin{aligned} \min_{\mathbf{x} \triangleq [x_1 \dots x_N]^\top} F(\mathbf{x}) &\triangleq \min_{\{x_i, i=1, \dots, N\}} \sum_{i=1}^N f_i(x_i, \mathbf{x}_{ij}) \\ \text{subject to } g_i(x_i, \mathbf{x}_{ij}) &\leq \mathbf{0}, \quad i = 1, \dots, N \\ h_i(x_i, \mathbf{x}_{ij}) &= \mathbf{0}, \quad i = 1, \dots, N \end{aligned} \quad (3.20)$$

where F is the global, or *common* cost function.

Remark 3.2 A first difference with the distributed algorithms of Section 3.3 is that the agents do not have a copy of the whole optimization variable. Rather, they modify their own optimization variable x_i in parallel with the other agents ones.

3.4.3 Algorithm Description

3.4.3.1 Fitness Function

Having made no constraints assumption we choose a generic method to handle constraints of (3.20). In particular we apply Deb's rule, (see Subsection 3.2.3.2). As shown, equality constraints are handled via a transformation in inequality constraints with the introduction of a positive threshold $\epsilon \in \mathbb{R}^+$, and by defining $\tilde{g}_i(x_i, \mathbf{x}_{ij}) \triangleq \text{col}(g_i(x_i, \mathbf{x}_{ij}), |h_i(x_i, \mathbf{x}_{ij})| - \epsilon)$, $\forall i \in \mathcal{V}_p$, where \tilde{g}_i is a vector of functions of dimension $m_i + q_i$. In order to implement Deb's rule, the following *common* fitness function is defined

$$\tilde{F}(\mathbf{x}) \triangleq \begin{cases} F(\mathbf{x}) & \text{if } \tilde{g}_i(\mathbf{x}) \leq 0 \quad \forall i \in \mathcal{V}_p \\ f_{\max} + \sum_{i=1}^N \sum_{k=1}^{m_i+q_i} \chi(\tilde{g}_{i,k}(\mathbf{x})) & \text{otherwise} \end{cases} \quad (3.21)$$

where $\tilde{g}_{i,k}$ is the k -th component of \tilde{g}_i , and where function χ , and value f_{\max} are defined in (3.15). Recall that the second term in second case of (3.21) is the sum of constraints.

3.4.3.2 Variables Settings

Each agent i has N_p particles associated to its private variable, and they represent a position in the search space $x_{i,p} \in \mathbb{R}^{n_i}$, $p = 1, \dots, N_p$. As done for the centralized

PSO of Section 3.2, we associate to each particle a personal best $b_{i,p}$, and a local best $l_{i,p}$ of a defined subset $\mathcal{S}_{i,p}$. Moreover, we require such subset to be the same for each particle having same p label, i.e. $\mathcal{S}_{i,p} = \mathcal{S}_p, \forall i \in \mathcal{V}_p$. In this work, such subset is defined according to the singly-linked list of Algorithm 3. In addition we define two new variables associated to each particle and personal best, namely, $h_{i,p}^1$, and $h_{i,p}^2$ as far as the former is concerned, and $h_{i,p}^{b,1}$, and $h_{i,p}^{b,2}$ regarding the latter. These are employed for average consensus purposes. In particular, variables with superscript 1 are the local estimate of the original common cost function F averaged among the N agents, while variables with superscript 2 are used for the average of the sum of constraints. Both average functions are evaluated in the optimization variable $\mathbf{x}_p \triangleq \text{col}(x_{1,p}, \dots, x_{N,p})$ concerning variables $h_{i,p}^1$, and $h_{i,p}^2$, and in $\mathbf{b}_p \triangleq \text{col}(b_{1,p}, \dots, b_{N,p})$ concerning variables $h_{i,p}^{b,1}$, and $h_{i,p}^{b,2}$, i.e composed by all agents particles, respectively personal bests, having the same p index. In other words, as it will be clear in the sequel, the mentioned additional variables are used to get knowledge of the common fitness function (3.21), by running an average consensus algorithm that makes them reach the following values

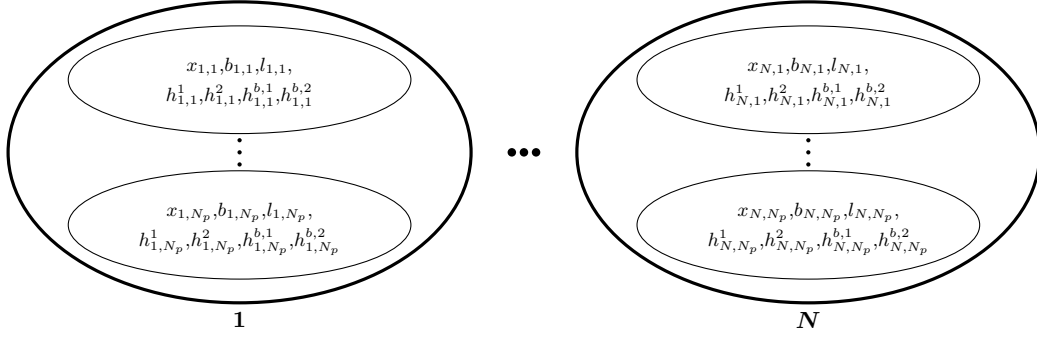
$$\begin{aligned}
h_{i,p}^1 &= \frac{1}{N} \sum_{l=1}^N f_l(x_{l,p}, \mathbf{x}_{l,j,p}) \\
h_{i,p}^{b,1} &= \frac{1}{N} \sum_{l=1}^N f_l(b_{l,p}, \mathbf{b}_{l,j,p}) \\
h_{i,p}^2 &= \frac{1}{N} \sum_{l=1}^N \left(\frac{1}{m_l + q_l} \sum_{k=1}^{m_l + q_l} \chi(\tilde{g}_{l,k}(x_{l,p}, \mathbf{x}_{l,j,p})) \right) \\
h_{i,p}^{b,2} &= \frac{1}{N} \sum_{l=1}^N \left(\frac{1}{m_l + q_l} \sum_{k=1}^{m_l + q_l} \chi(\tilde{g}_{l,k}(b_{l,p}, \mathbf{b}_{l,j,p})) \right)
\end{aligned} \tag{3.22}$$

where we defined $\mathbf{x}_{l,j,p} \triangleq \{x_{j,p} \in \mathbb{R}^{n_j} | j \in \mathcal{N}_l^p\}$, resp. $\mathbf{b}_{l,j,p} \triangleq \{b_{j,p} \in \mathbb{R}^{n_j} | j \in \mathcal{N}_l^p\}$, i.e. the sets of physical neighbors position and personal best associated to the particle labeled p . Notice that we keep index i in (3.22) to stress the fact that, once performed the average consensus, agent i estimate of average functions can have a small difference with respect to the other agents one. If this difference can be neglected, then we can approximate the consensus result via $h_{i,p}^1 = h_{j,p}^1 = h_p^1, \forall i, j \in \mathcal{V}_p$. Same conclusions hold for $h_{i,p}^2, h_{i,p}^{b,1}, h_{i,p}^{b,2}$.

An illustration of all the variables for each agent is shown in Fig. 3.3.

3.4.3.3 Particles Update Law

As shown in Section 3.2, PSO equations determine the particles motion in the search space in order to find the global optimum. For ease of notation, at iteration k of the algorithm we note $x_{i,p} \triangleq x_{i,p}(k)$, and $x_{i,p}^+ \triangleq x_{i,p}(k+1)$. Same notations hold for the speed $s_{i,p}$, particle best $b_{i,p}$, and local best $l_{i,p}$. We rewrite then PSO equations,

Figure 3.3: Variables associated to the N agents of the network for DPSO1.

which for each particle, are given by

$$\begin{cases} s_{i,p}^+ = \omega s_{i,p} + \phi_{1,i,p}(l_{i,p} - x_{i,p}) + \phi_{2,i,p}(b_{i,p} - x_{i,p}) \\ x_{i,p}^+ = x_{i,p} + s_{i,p}^+ \end{cases} \quad (3.23)$$

$$(b_{i,p}^+, h_{i,p}^b) = \begin{cases} (x_{i,p}^+, h_{i,p}^+) & \text{if } h_{i,p}^+ < h_{i,p}^b \\ (b_{i,p}, h_{i,p}^b) & \text{otherwise} \end{cases} \quad (3.24)$$

$$l_{i,p}^+ = \arg \min_{\{b_{i,p}^+, p \in S_p\}} \{h_{i,p}^b\} \quad (3.25)$$

$$\text{where } h_{i,p} \triangleq h_{i,p}(k) = \begin{cases} h_{i,p}^1 & \text{if } h_{i,p}^2 = 0 \\ f_{max} + h_{i,p}^2 & \text{otherwise} \end{cases}$$

and $h_{i,p}^+ \triangleq h_{i,p}(k+1)$. A similar definition holds for $h_{i,p}^b, h_{i,p}^{b,1}, h_{i,p}^{b,2}$. Equation (3.24) is the decision step of the algorithm in which the new particles are compared to the according personal best in terms of average fitness function evaluation. $\phi_{1,i,p} \triangleq \phi_{1,i,p}(k)$, and $\phi_{2,i,p} \triangleq \phi_{2,i,p}(k)$ are defined as in Subsection 3.2.1, and here they are allowed to have different value for any particle labeled (i, p) . Eventually, box constraints of the form $\underline{x}_i \leq x_i \leq \bar{x}_i$ are handled by adding equations of the form of (3.11), (3.12) to (3.23), which rewritten for each particle p of each agent i are

$$s_{i,p}(k+1) \triangleq \max\{\min\{s_{i,p}(k+1), \bar{s}_i\}, -\bar{s}_i\} \quad (3.26)$$

$$x_{i,p}(k+1) \triangleq \max\{\min\{x_{i,p}(k+1), \bar{x}_i\}, \underline{x}_i\} \quad (3.27)$$

3.4.3.4 Communication Settings

The algorithm relies on the following assumption on the communication graph.

Assumption 3.1 *It exists a communication graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$ such that $\mathcal{V}_c \equiv \mathcal{V}_p$, and $\mathcal{E}_c \subseteq \mathcal{V}_c \times \mathcal{V}_c$ is such that if $(i, j) \in \mathcal{E}_p$ then both (i, j) , and $(j, i) \in \mathcal{E}_c$.*

Assumption 3.1 implies that each agent i has to be able to exchange information in both senses, i.e. receive and transmit information, with both its physical neighbors $j : j \in \mathcal{N}_i^p$, and the agents for which i is a physical neighbor, i.e. $k : i \in \mathcal{N}_k^p$. The

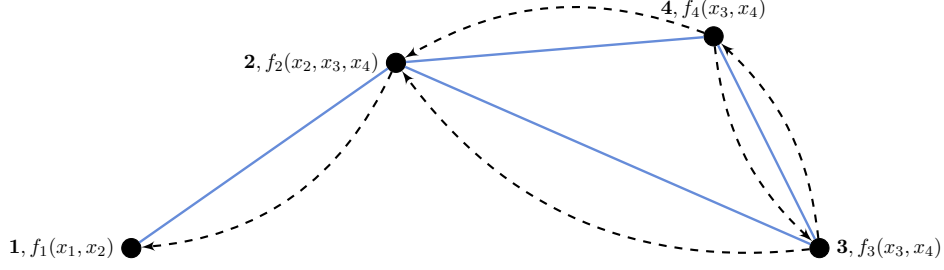


Figure 3.4: Physical graph \mathcal{G}_p (dash line), and communication graph \mathcal{G}_c (solid line).

Algorithm 4 Distributed average common fitness function evaluation

Input: i component of $\hat{\mathbf{x}}_p$

Output: Average fitness function value: $\hat{h}_{i,p}$

- 1: Send $\hat{x}_{i,p}$ to all agents $j : i \in \mathcal{N}_j^p$, via \mathcal{G}_c
 - 2: Wait to receive $\hat{x}_{j,p}$ from the physical neighbors, via \mathcal{G}_c
 - 3: $\hat{h}_{i,p}^1 = f_i(\hat{x}_{i,p}, \hat{\mathbf{x}}_{ij,p})$
 - 4: $\hat{h}_{i,p}^2 = \frac{1}{m_i + q_i} \sum_{k=1}^{m_i + q_i} \chi(\tilde{g}_{i,k}(\hat{x}_{i,p}, \hat{\mathbf{x}}_{ij,p}))$
 - 5: Run *average consensus* on variables $\hat{h}_{i,p}^1$, and $\hat{h}_{i,p}^2$, $p = 1, \dots, N_p$, via \mathcal{G}_c (see Section 3.1)
 - 6: **if** ($\hat{h}_{i,p}^2 == 0$) **then**
 - 7: $\hat{h}_{i,p} = \hat{h}_{i,p}^1$
 - 8: **else**
 - 9: $\hat{h}_{i,p} = f_{max} + \hat{h}_{i,p}^2$
 - 10: **end if**
 - 11: **return** $\hat{h}_{i,p}$
-

communication graph is thus *undirected*. An example for $N = 4$ is given in Fig. 3.4.

3.4.3.5 Distributed Functions Evaluation

Before providing the overall algorithm, we need to introduce a distributed subroutine that allows the evaluation of the average common cost function and sum of constraints in order to compute the average common fitness function. This has to be run each time that a decision step, as the one of (3.24), has to be performed. Since this subroutine makes use alternatively of variables $h_{i,p}^1$, $h_{i,p}^2$, and $h_{i,p}^{b,1}$, $h_{i,p}^{b,2}$ according to the algorithm requirements, and the functions evaluations is done in different points of the search space, in order to present it in a general way, we refer to a new definition of generic variables $\hat{h}_{i,p}^1$, $\hat{h}_{i,p}^2$, and $\hat{\mathbf{x}}_p \triangleq \text{col}(\hat{x}_{1,p}, \dots, \hat{x}_{N,p})$. These will be replaced by the needed variables according to the situation. The distributed subroutine is shown in Algorithm 4, where each component $\hat{x}_{i,p}$ of the generic position $\hat{\mathbf{x}}_p$, is associated with the estimation of the average common fitness function

Algorithm 5 DPSO1**Output:** Local best among S_p : $l_{i,p}$ *Initialization :*

- 1: Randomly initialize $x_{i,p} \in [\underline{x}_i, \bar{x}_i]$, $s_{i,p} \in [-\bar{s}_i, \bar{s}_i]$, where $\bar{s}_i \triangleq \frac{1}{2}(\bar{x}_i - \underline{x}_i)$
- 2: $b_{i,p} = x_{i,p}$
- 3: Set $\epsilon = \bar{\epsilon}$
- 4: Run Algorithm 4 with input $x_{i,p}$
- 5: Perform (3.25), and set $l_{i,p} = l_{i,p}^+$

LOOP Process

- 6: **for** $k = 1$ to **max_iter** **do**
- 7: Perform (3.23)
- 8: Run Algorithm 4 with input $x_{i,p}^+$
- 9: Perform (3.24)
- 10: Set $\varepsilon = \varepsilon(k)$
- 11: Run Algorithm 4 with input $b_{i,p}^+$
- 12: **if** $(h_{i,p}^+ < h_{i,p}^b)$ **then**
- 13: $(b_{i,p}^+, h_{i,p}^b) = (x_{i,p}^+, h_{i,p}^+)$
- 14: **end if**
- 15: Perform (3.25)
- 16: Set $x_{i,p} = x_{i,p}^+$, $b_{i,p} = b_{i,p}^+$, and $l_{i,p} = l_{i,p}^+$
- 17: **end for**
- 18: **return** $l_{i,p}$

 $\hat{h}_{i,p}$.

Remark 3.3 *It is important to notice that the functions evaluation in Algorithm 4 is dependent on the chosen value of constraints threshold ϵ , via the definition of \tilde{g}_i . In order to let better performance, as shown in the centralized PSO algorithm of [Aguirre 2007], we can employ an additional constraints handling technique, named dynamic tolerance. This consists in letting the mentioned threshold be iteration-varying, typically linearly decreasing from an initial value $\bar{\epsilon}$ to a final target $\underline{\epsilon}$.*

3.4.3.6 Distributed Optimization Algorithm

The overall distributed algorithm to be run by each agent on each of its particles $x_{i,p}$ is shown in Algorithm 5, where iterations are performed until a prescribed number **max_iter**. In this algorithm, Steps 10–14 are only necessary if ϵ is iteration-varying. Indeed if ϵ decreases at each algorithm step, then the personal bests need to be reevaluated to see how they *fit* the new, ϵ -dependent, fitness function. If not performed, a $b_{i,p}$ computed at the beginning of the iterations has higher probability to have better fitness function value with respect to one computed afterwards. As previously mentioned, the basic idea behind Algorithm 5 is to perform a distributed optimization algorithm that, from a global perspective, acts as a centralized PSO. In this case, the latter is chosen to be the one of [Aguirre 2007], as it is one among

the centralized PSO algorithms combining PSO with Deb's rule. In order to do so, iteratively, variables $x_{i,p}$, and $b_{i,p}$ are updated only where needed, according to Definition 3.1, and their effect on the common fitness function is evaluated by means of average consensus. Thus, the only source of error between the performance of DPSO1 and its centralized version is given by the consensus errors, if any. As a result, because of the structure of the proposed algorithm, and under the assumption of negligible average consensus errors, we are able to conclude the following

Proposition 3.1 *Consider the centralized PSO algorithm obtained by substitution of Algorithm 4 with direct computation of $\hat{h}_{i,p}$. If it converges, so it does Algorithm 5. In addition, they converge with same performance in quality of the solution and speed of convergence.*

Because of Proposition 3.1, DPSO1 has the same statistical properties of its centralized counterpart. For this reason, we do not provide here statistical simulations to prove its convergence properties.

Remark 3.4 *It is important to note that thanks to Assumption 3.1 we are able to communicate variables $x_{i,p}$, and $b_{i,p}$ with one communication step. This allows us to avoid adding any consensus or consistency constraint to (3.20), as it is usually done in the mentioned references of Subsection 3.4.1.*

Remark 3.5 *It is important to stress that even if Assumption 3.1 allows direct communication of the particles and the associated personal bests among the agents, this fact does not have to be confused or interpreted as if each agent only needed to retrieve information about its direct neighbors according to \mathcal{G}_c . Indeed, in order to properly move its own particles, recall that each agent has to have knowledge of the average common fitness function via Algorithm 4. In other words, each agent has to have knowledge of how every other agent in the network contributes to the common fitness function, including those agents to which it is not directly physically coupled. To go further in practical implementation details, this is needed to perform decision steps (3.24) in DPSO1. This fact sets a major difference with respect to the algorithm proposal of Section 3.5.*

Remark 3.6 *In the centralized PSO described in [Aguirre 2007], an additional step is considered to help keeping diversity of the flock of particles, which in turns helps preventing from premature convergence. This is obtained by applying two perturbation operators to the particle personal bests $b_{i,p}$ each time that they are updated. These operators are named C-Perturbation, and M-Perturbation. The reader may refer to the mentioned reference for further details. This additional step can be easily implemented in Algorithm 5. However, in order to evaluate the new perturbed $b_{i,p}$ in the common fitness function, the average consensus step needs to be performed, and this clearly lengthen the time of convergence. Thus, it should be considered only if it actually enhances the optimality of the solution in a remarkable way.*

3.4.4 Enhancement of Self-configuration Step

The main communication burden is due to the need for performing the average consensus in Algorithm 4 to evaluate the problem functions. This step cannot be approximated as a one-step communication as shown in [Wakasa 2015a], as the accuracy of consensus would not be sufficient to perform decision steps on particles, as (3.24). Nonetheless we employ a *finite-time* average consensus algorithm to reduce the communication burden as much as possible. In addition, being interested in a whole-distributed solution, we capitalize on the distributed self-configuration shown in Section 3.1. Recall that such self-configuration step is performed via solving the *nonconvex* optimization problem of minimizing quadratic error (3.3), with respect to matrices W_1, \dots, W_D . However, being the presented approach gradient-based, the solution is strongly dependent on the algorithm initialization, i.e. the initialization of the mentioned self-configuration step. We notice that (3.3) can be written in the form of (3.20), with no constraints, by taking the agent i private cost function as $\sum_{q=1}^Q (h_{i,q}(D) - y_q)^2$, and (3.3) as common cost function. Thus, we propose to use Algorithm 5 to provide an initial point to Algorithm 1, as it could help finding a better solution. In this self-configuration step though, clearly, the average consensus used in Algorithm 4 cannot be performed via the finite-time algorithm, which is the aim of this stage. For this reason we consider to update $\hat{h}_{i,p}$ variables at Step 5 in Algorithm 4, for this only self-configuration step, via

$$\begin{aligned} \hat{h}_{i,p}^+ &= \varphi_{ii} \hat{h}_{i,p} + \sum_{j \in \mathcal{N}_i} \varphi_{ij} \hat{h}_{j,p} \\ \varphi_{ij} &= \begin{cases} \frac{1}{\max\{d_i, d_j\} + 1} & \text{if } (i, j) \in \mathcal{E}_c \\ 1 - \sum_{j \in \mathcal{N}_i} \frac{1}{\max\{d_i, d_j\} + 1} & \text{if } i = j \end{cases} \end{aligned} \quad (3.28)$$

i.e. φ_{ij} are Metropolis-Hastings weights, where d_i is the *degree* of node i , i.e. the number of edges connected to it. Update law (3.28) thus only requires each agent to know little information about its neighbors, and it has to be run for a sufficient number of steps in order to ensure proper convergence of Algorithm 5. In addition, for the particular problem of minimizing (3.3), there is no need for the agents to send, their private variables to the other ones. Agents do share common variables, but this link is kept implicit in the computation of $h_{i,q}(D)$, appearing in (3.3), via (3.1).

Example

We consider MATLAB[®] *bucky* graph in Fig. 3.5, to show the enhancement that one can achieve via a combined action of DPSO1 with self-configuration of Algorithm 1. This graph has radius, and diameter $r = d = 9$, and a total number of agents N equal to 60. Good results are found for a value of $D = 11$, i.e. 11 steps are needed to find the average consensus on the network with an acceptable small error. Fig. 3.6 shows the results of Algorithm 1 whose initialization is provided by

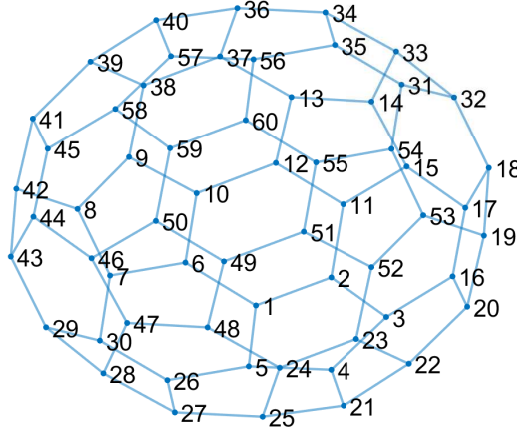
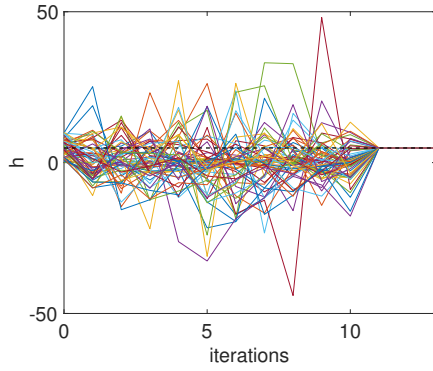
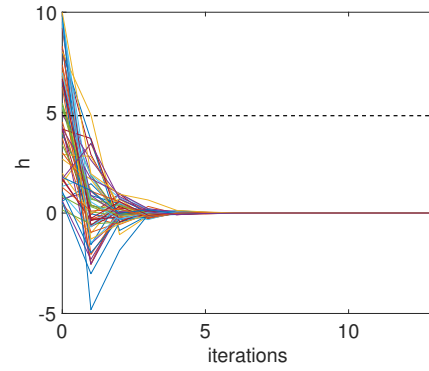


Figure 3.5: Bucky communication graph.



(a) Gradient backpropagation with DPSO1 solution as initialization. \mathbf{h} components reach their average value after 11 step with a small steady state error.



(b) Gradient backpropagation with random initialization. \mathbf{h} components fail to reach their average value.

Figure 3.6: Combined DPSO1 and backpropagation versus backpropagation.

DPSO1 versus the solution provided by the only gradient backpropagation approach of Algorithm 1. The dash-dot black line represents the average of the initial network state $1/N \sum_{i=1}^N h_i(0)$, where $h_i(0)$, $i = 1, \dots, N$ are randomly selected, and thus the value to which the agents state should converge in D steps. From this example it is clear that DPSO1 not only can enhance the optimality of the solution of Algorithm 1, but it also helps finding one when gradient back-propagation fails. Indeed, weights found via gradient back-propagation do not let convergence to the average consensus value as it shows a steady state constant error. By employing the algorithm initialization provided by DPSO1, the steady state error is negligible.

3.4.5 Conclusion

We presented a novel distributed PSO algorithm that enables to solve general distributed constrained optimization problems that can be written in the form of (3.20). Potentiality of the algorithm was also shown when solving a distributed factorization problem to tune the finite-time average consensus algorithm.

The main drawback of the approach is due to the communication burden, in turns due to the need for performing a finite-time average consensus algorithm for each step of DPSO1. This basically raises the time for convergence especially for communication graphs having high diameter, and radius. This is one of the reasons that motivated the work of next section.

3.5 Distributed Cooperative PSO with reduced communication: DPSO2

3.5.1 Related Works and Contribution

In Subsection 3.4.1, in order to distinguish the available distributed PSO algorithms from those treating the particular case of physically distributed systems, we introduced the class of PPSO. Recall that this is concerned with parallel computation of originally centralized problems. How these problems are distributed among several units is thus a choice, i.e. a problem parameter. On the one hand, the two classes of distributed problems, namely PPSO and DPSO, have many points in common, and this justifies the interest for studying the literature concerning them both. On the other hand, they refer to essentially different problems, and they cannot be treated in the same framework. Please refer to Subsection 3.4.1, for the available works in the literature concerning DPSO.

When a centralized, usually large-scale, optimization problem has to be solved, parallel computing is often employed to speed-up convergence. In this regard, the author in [Rivera 2001] identifies different strategies to do so, namely *global*, *fine-grained*, and *coarse-grained* parallelization. As far as PPSO is concerned, for the first type of strategy one can cite, for instance, the works of [Schutte 2004, Chu 2003] for synchronous parallelization, and [Venter 2006, Koh 2006] for the asynchronous one. However, their main goal is to exactly reproduce the PSO algorithm while reducing the run time [Chu 2003]. Thus the main structure of the algorithm remains unchanged. For this reason, this approach cannot be employed for DPSO. In fine-grained parallelization the update of each particle of the PSO algorithm is performed by a dedicated agent. A formalization of such strategy can be found in [Akat 2008], and an interesting application for multi-robots in [Hereford 2006]. From a DPSO perspective each agent shares the whole optimization variable, and the common cost function is known to each of them, which is not the case of the problem addressed in this work. Moreover, its use for physically distributed systems is confined to limited application, as the one of [Hereford 2006], where the number of agent has to be sufficiently high. Coarse-grained parallel computing becomes an interesting

strategy when it is combined with techniques of *cooperative co-evolution* (CC). Instead of simply sharing the PSO particles among multiple agents the problem is divided in smaller sub-problems assigned to each agent. CC was first introduced by [Potter 1994], while its first application to centralized PSO (CC-PSO) can be found in [Van den Bergh 2004]. The main idea, known also as *divide and conquer* strategy, can be summarized in three important steps, namely *problem decomposition*, *sub-component optimization*, and *cooperative combination* [Mahdavi 2015, Yang 2008]. In the first step, a n -dimensional problem is divided in sub-problems of lower dimension, i.e. each of them, during the subcomponent optimization step, is responsible for searching a solution in only a nonoverlapping subset of the possible dimensions. Such decomposition is critical as it is strongly sensible to the degree of *separability* of the problem. The concerning literature is thus mainly due to the different methods of grouping the optimization problem in subcomponents whose inter-dependency is minimized. A review of such techniques can be found in [Mahdavi 2015]. Once the decomposition is chosen, CC can be coupled with PPSO (CC-PPSO) by assigning a different group to each agent. As an example, this technique is proposed by the authors of [Calazan 2013] to solve high-dimension optimization problems, and in [Atashpendar 2016] for multi-objective optimization. However, the success of this approach strongly relies on the decomposition itself. Thus, its main short-come, and the reason why it cannot be generally applied to DPSO is that, in physically distributed systems, the decomposition is *imposed* by the physics of the problem itself, and it cannot be modified.

In this section we propose a novel DPSO algorithm to solve a class of physically distributed optimization problems among a set of agents, and we name it *DPSO2*. In particular, differently from [Wakasa 2015a], and [Wakasa 2015b], the agents are allowed to share a common optimization variable. Moreover, based on some communication assumptions, we exploit the problem structure rather than employing consensus techniques in order to estimate the common objective function.

In the original CC formulation, the populations belonging to each subcomponent are processed in *sequential* order, i.e. at any time step of the algorithm, only one population is active. In this case we say that the *update timing* parameter is sequential [Popovici 2006]. Moreover, the population to be evaluated is usually coupled with the latest local best values found by the other subcomponent populations. This strategy, known as *single best collaboration* [Wiegand 2003], guarantees the objective function to be strictly nonincreasing in the local best trajectories during the run of the algorithm [Van den Bergh 2004]. Nonetheless, when the optimization is distributed among the agents, as for DPSO, and PPSO, where the goal is to reduce the run time, we are interested in a *parallel* update timing, i.e. each agent population is active at any algorithm time step, without the need for waiting its own turn. In our work we focus on this last case, as the algorithm that we present has some important common points with CC-PPSO algorithms where the update timing is parallel. In particular, our contribution has some similarities with the approach shown in [Calazan 2013], which is though conceived for PPSO. However, the latter does not provide any analysis of the impact of choosing a parallel update

timing. How this parameter changes the behavior of the algorithm has been studied by [Popovici 2006], and [Jansen 2003]. In particular, [Popovici 2006] provides an analysis built on the so-called *best-response curves*. Based on them, we carry out a new dynamic behavior analysis. This in turns suggests the introduction of a new parameter, as well as a modification of the classic PSO algorithm, which reveals to have an important impact in the performance of the proposed DPSO. Eventually, the presented algorithm can be extended to the constrained optimization case in which each agent has a private constraint allowed to depend also on the other agents optimization variables. On this regard, as we have previously done for DPSO1, while [Wakasa 2015b] makes use of a perturbed primal-dual method, and [Wakasa 2015a] capitalizes on projection functions, we choose to handle constraints by employing Deb's rule.

3.5.2 Problem Statement

Differently from what done in Section 3.4, we state here the *unconstrained* distributed optimization problem on which the main result of this section and the following discussion are based. The reader may refer to Subsection 3.5.6 for the constrained extension. The problem statement is based on similar definitions, functions and variables settings of Subsection 3.4.2. For the convenience of the reader, in the following we recall and adapt them for the unconstrained version. Let us consider a group of N agents, each of which disposes of a private control variable $x_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$, and a private cost function f_i . In this context, it is useful to provide an unconstrained version of Definition 3.1.

Definition 3.2 In an unconstrained optimization problem, an agent i is *physically* coupled to agent j if its private cost function depends on agent j private control variable. Agent j is thus said to be a *physical neighbor* of agent i .

We again consider a graph $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{E}_p)$ that keeps track of the physical coupling among the agents according to the above definition. In particular $\mathcal{V}_p = \{1, \dots, N\}$ is the set of the agents, i.e. the nodes of the graph, and $\mathcal{E}_p \subseteq \mathcal{V}_p \times \mathcal{V}_p$ is the set of edges among them, where the edge $(i, j) \in \mathcal{E}_p$ if and only if agent j is a *physical neighbor* of agent i . Thus, the set of physical neighbors of agent i is defined as $\mathcal{N}_i^p \triangleq \{j \in \mathcal{V}_p : (i, j) \in \mathcal{E}_p\}$. By defining $\mathbf{x}_{ij} \triangleq \{x_j \in \mathbb{R}^{n_j} : j \in \mathcal{N}_i^p\}$ as the set of physical neighbors variables of agent i , we are able to represent the generic agent i cost function as $f_i(x_i, \mathbf{x}_{ij})$, where $f_i : \mathbb{R}^{n_i + \sum_{j \in \mathcal{N}_i^p} n_j} \rightarrow \mathbb{R}$. The overall unconstrained optimization problem is

$$\min_{\mathbf{x} \triangleq [x_1 \dots x_N]^\top} F(\mathbf{x}) \triangleq \min_{\{x_i, i=1, \dots, N\}} \sum_{i=1}^N f_i(x_i, \mathbf{x}_{ij}) \quad (3.29)$$

Thus, agents have to *cooperatively* minimize a common cost function $F : \mathbb{R}^{\sum_{i=1}^N n_i} \rightarrow \mathbb{R}$ while sharing the common optimization variable \mathbf{x} .

Remark 3.7 Note that agent i optimization variable x_i is a subcomponent of problem (3.29) optimization variable \mathbf{x} . Reflecting the physics of the problem, such decomposition is given, and it cannot be altered to perform particular CC algorithms.

3.5.3 Dynamic Behavioral Analysis

Before providing the distributed algorithm to solve (3.29), let us consider it in its *centralized* formulation. In particular, we focus the analysis of this section on a CC-PSO algorithm based on *parallel* update timing. The reason for doing so is two-fold. On the one hand, the distributed algorithm described in Subsection 3.5.4 has its same convergence properties. On the other hand, this analysis suggests the modification of the classic PSO equations by introducing a new parameter.

3.5.3.1 CC-PSO Algorithm

If (3.29) is considered in the CC-PSO framework, then F is the cost function to be minimized, and \mathbf{x} the optimization variable. The problem is then divided in N sub-problems, i.e. the vector \mathbf{x} is decomposed in N subcomponents $x_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$, each of which represents a nonoverlapping subset of the dimension of the search space. A swarm of N_p particles is associated to each subcomponent. In order to evaluate each of them, i.e. how each of its particles fits the cost function F , a *context vector* has to be constructed [Van den Bergh 2004]. This is usually done by concatenating each subcomponent particle with the global best particles of the other subcomponents swarms [Wiegand 2003]. This practice will be detailed in the algorithm description. The swarm is then updated independently one from another by employing the previously introduced classic PSO equations

$$\begin{cases} s_{i,p}(k+1) = \omega s_{i,p}(k) + \phi_{1,p}(k)(g_i(k) - x_{i,p}(k)) \\ \quad + \phi_{2,p}(k)(b_{i,p}(k) - x_{i,p}(k)) \\ x_{i,p}(k+1) = x_{i,p}(k) + s_{i,p}(k+1) \end{cases} \quad (3.30)$$

where k is the current algorithm step, $x_{i,p}$, $s_{i,p}$, and $b_{i,p}$ are respectively the position, the speed and the personal best position values associated to the p -th particle of the i -th subcomponent, g_i is the global best of the i -th swarm. See Section 3.2 for the other parameters definition and settings.

It is well known that the key point affecting the algorithm convergence is the problem decomposition step. By giving the definition of *separable* function [Weise 2012]:

Definition 3.3 A function $F(x_1, \dots, x_N)$ is separable if and only if

$$\begin{aligned} \arg \min_{x_1, \dots, x_N} F(x_1, \dots, x_N) = \\ \left\{ \arg \min_{x_1} F(x_1, \dots), \dots, \arg \min_{x_N} F(\dots, x_N) \right\} \end{aligned}$$

then, in order to assure proper convergence of the algorithm, the cost function F should be separable in the chosen decomposition. As pointed out in Remark 3.7, in

Algorithm 6 CC-PSO with parallel update timing**Output:** Global best: g_i *Initialization :*

- 1: Randomly initialize $x_{i,p} \in [\underline{x}_i, \bar{x}_i]$, $s_{i,p} \in [-\bar{s}_i, \bar{s}_i]$, $p = 1, \dots, N_p$
- 2: $b_{i,p} = x_{i,p}$, $p = 1, \dots, N_p$
- 3: Randomly initialize $g_i \in [\underline{x}_i, \bar{x}_i]$

LOOP Process

- 4: **for** $k = 1$ to max_iter **do**
- 5: Compose the context vectors associated to the particles $x_{i,p}$:
 $\mathbf{x}_{i,p}^g \triangleq (g_1, \dots, x_{i,p}, \dots, g_N)$, $p = 1, \dots, N_p$
- 6: Compose the context vectors associated to the personal bests $b_{i,p}$:
 $\mathbf{b}_{i,p}^g \triangleq (g_1, \dots, b_{i,p}, \dots, g_N)$, $p = 1, \dots, N_p$
- 7: Evaluate the particle context vectors in F : $F_{i,p}^x \triangleq F(\mathbf{x}_{i,p}^g)$, $p = 1, \dots, N_p$
- 8: Evaluate the personal best context vectors in F : $F_{i,p}^b \triangleq F(\mathbf{b}_{i,p}^g)$, $p = 1, \dots, N_p$
- 9: Update personal bests for $p = 1, \dots, N_p$

$$(b_{i,p}^{new}, F_{i,p}^{b,new}) = \begin{cases} (x_{i,p}, F_{i,p}^x) & \text{if } F_{i,p}^x < F_{i,p}^b \\ (b_{i,p}, F_{i,p}^b) & \text{otherwise} \end{cases}$$

- 10: $b_{i,p} = b_{i,p}^{new}$, $F_{i,p}^b = F_{i,p}^{b,new}$
- 11: Update global bests $g_i^{new} = \arg \min_{\{b_{i,p}\}} \{F_{i,p}^b\}$
- 12: $g_i = g_i^{new}$
- 13: Perform (3.30)
- 14: **end for**
- 15: **return** g_i

the addressed problem, the problem decomposition is given and it may not reflect the structure of the cost function, i.e. F may not be separable in such decomposition. Moreover, as previously mentioned, we consider each swarm associated to each subcomponent to be active at any algorithm step. In other words the update timing is parallel. Eventually, as done in Section 3.4 for DPSO1 algorithm, box constraints of the form of $\underline{x}_i \leq x_i \leq \bar{x}_i$, $\underline{x}_i, \bar{x}_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$, can be handled by placing respectively (3.26) right after the first equation in (3.30), and (3.27) after the second equation in (3.30). The overall proposed CC-PSO algorithm is described in Algorithm 6 for the generic i -th swarm.

Remark 3.8 *Since at each iteration the global bests of each swarm are likely to change, the cost function value associated to the personal best $b_{i,p}$ may provide a false comparison reference for the corresponding particle $x_{i,p}$ context vector evaluation because they may refer to different values of global bests. This is why we propose a first algorithm modification by requiring to evaluate the personal best at each algorithm step, by composing the context vectors associated to them too. This is performed by Steps 6, and 8 of Algorithm 6.*

3.5.3.2 Motivating Example

The given subcomponents affect the convergence properties as well as the dynamic behavior of the algorithm because they define the *best-response curves*. Following [Popovici 2006] and Definition 3.3, the generic best-response curves associated to (3.29) are

$$\begin{aligned} \text{bestResponse}X_i(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N) &\triangleq \\ \arg \min_{x_i} F(\dots, x_i, \dots), \quad i = 1, \dots, N \end{aligned} \quad (3.31)$$

which clearly depend on the given problem decomposition. Authors of [Popovici 2006] provide some insights of how the update timing influences the performance of the algorithm according to the aforementioned best-response curves. However, motivated by the following example, we provide a new dynamic behavior analysis as well as the synthesis of a modified CC-PSO algorithm with parallel update timing. Let us consider the cost function $F(x_1, x_2) = (x_1 - x_2)^2$ to be minimized, where $x_1, x_2 \in \mathbb{R}$. Note that $F(x_1, x_2)$ is not separable. However let us consider the decomposition given by the two components of the optimization variable, x_1 , and x_2 . Using (3.31), as in [Popovici 2006], we analyze the deterministic system associated to the CC-PSO algorithm, which, for this example, is given by

$$\begin{cases} x_1(k+1) = \text{bestResponse}X_1(x_2(k)) = x_2(k) \\ x_2(k+1) = \text{bestResponse}X_2(x_1(k)) = x_1(k) \end{cases} \quad (3.32)$$

Such system reproduces the parallel update timing of the CC-PSO algorithm as if the swarms associated to the two components were able to find the global best at each time step. Even though $F(x_1, x_2)$ is a simple convex function, having its minima in $\Omega_{ex} \triangleq \{(x_1, x_2) \in \mathbb{R}^2 : x_1 = x_2\}$, system (3.32) shows an *oscillatory* behavior, given by the fact of having its eigenvalues in -1 , and 1 . Thus, it does not converge to Ω_{ex} for any initial condition $\{(x_1(0), x_2(0)) \in \mathbb{R}^2 : x_1(0) \neq x_2(0)\}$. This fact has an important implication on the dynamics of the CC-PSO algorithm. Although system (3.32) does not exactly reproduce Algorithm 6 behavior, it is likely to have similar oscillatory dynamics as the number of particles of each swarm increases. Fig. 3.7a shows the swarms global best trajectories during the run of CC-PSO algorithm for such example, where we set $N_p = 100$. Inspired by classic results in control theory, where typically an oscillatory system can be stabilized by introducing *additional damping*, we modify system (3.32) equations, according to

$$\begin{cases} x_1(k+1) = x_1(k) + \beta (\text{bestResponse}X_1(x_2(k)) - x_1(k)) \\ \quad = x_1(k) + \beta (x_2(k) - x_1(k)) \\ x_2(k+1) = x_2(k) + \beta (\text{bestResponse}X_2(x_1(k)) - x_2(k)) \\ \quad = x_2(k) + \beta (x_1(k) - x_2(k)) \end{cases} \quad (3.33)$$

Such system converges to Ω_{ex} for any initial condition, and $\beta \in]0, 1[$. Indeed, for the considered values of β , system (3.33) has one eigenvalue inside the unit circle,

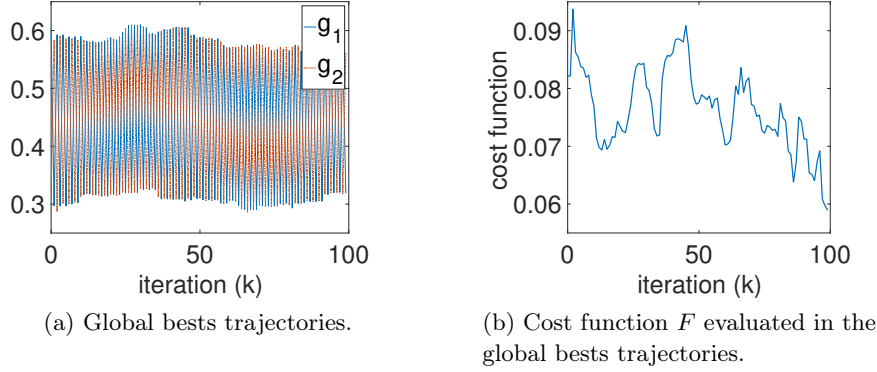


Figure 3.7: CC-PSO with parallel update timing dynamic behavior.

and one eigenvalue in 1 with corresponding eigenvector equal to $\mathbf{1}_2 \triangleq \text{col}(1, 1)$. Thus, system (3.33) trajectories converge to the subspace given by $\text{span}(\mathbf{1}_2)$, i.e. $\{(x_1, x_2) \in \mathbb{R}^2 : x_1 = x_2\}$, which is exactly Ω_{ex} . This motivates the CC-PSO algorithm modification shown in next subsection.

Remark 3.9 *The original CC-PSO is conceived for a sequential update timing parameter. In this case, only one subcomponent is active at any algorithm step. In particular, this implies that the cost function F is evaluated after the update of each subcomponent, and this guarantees F to be strictly nonincreasing in the best context vector trajectory [Van den Bergh 2004]. Such property is no longer satisfied in the case of parallel update timing. For instance, Fig. 3.7b shows the cost function of the aforementioned example evaluated in the best context vector trajectory, i.e. the one show in Fig. 3.7a.*

3.5.3.3 Damped CC-PSO Algorithm

In the previous subsection we saw how the introduction of the *damping factor* β let the deterministic system (3.33) converge to Ω_{ex} . Here we generalize this result for the minimization of a class of quadratic convex functions, providing the following

Theorem 3.1 *Given a cost function F of the form*

$$F(x, z) = [x^\top \quad z^\top]Q \begin{bmatrix} x \\ z \end{bmatrix} + [x^\top \quad z^\top]b + d$$

where $x \in \mathbb{R}^{n_x}$, $z \in \mathbb{R}^{n_z}$, $n_x, n_z \in \mathbb{N}^+$, $F : \mathbb{R}^{n_x+n_z} \rightarrow \mathbb{R}$, $d \in \mathbb{R}$, $b \in \mathcal{R}(Q)$, $Q = \begin{bmatrix} Q_1 & Q_2 \\ Q_2^\top & Q_3 \end{bmatrix} \succeq 0$, $Q_1 \in \mathbb{R}^{n_x \times n_x} \succ 0$, and $Q_3 \in \mathbb{R}^{n_z \times n_z} \succ 0$; then the system given by the following difference equations

$$\begin{cases} x(k+1) = x(k) + \beta (\text{bestResponseX}(z(k)) - x(k)) \\ z(k+1) = z(k) + \beta (\text{bestResponseZ}(x(k)) - z(k)) \end{cases} \quad (3.34)$$

converges to $\Omega \triangleq \{(x, z) \in \mathbb{R}^{n_x+n_z} : (x, z) = \arg \min_{x,z} F(x, z)\}$ for any initial condition $x(0) \in \mathbb{R}^{n_x}$, and $z(0) \in \mathbb{R}^{n_z}$ if the damping factor $\beta \in \mathbb{R} : \beta \in]0, \min\left\{\frac{1}{\lambda(B)} - \epsilon, 1\right\}]$, where $\epsilon \in \mathbb{R}^+$ is small, and $B \triangleq \begin{bmatrix} I_{n_x} & Q_1^{-1}Q_2 \\ Q_3^{-1}Q_2^\top & I_{n_z} \end{bmatrix}$, being I_{n_x}, I_{n_z} the identity matrices of respective dimension n_x , and n_z , and $\lambda(B)$ the maximum eigenvalue of B .

Proof: First of all note that $Q \succeq 0$ is required for the convexity of the optimization problem. Ω is given by the first order condition, i.e. by the couples (x, z) that satisfy the following system of linear equations

$$\begin{cases} \frac{\partial F(x,z)}{\partial x} = 0 \\ \frac{\partial F(x,z)}{\partial z} = 0 \end{cases} \Rightarrow \begin{cases} 2Q_1x + 2Q_2z + b_1 = 0 \\ 2Q_3z + 2Q_2^\top x + b_2 = 0 \end{cases} \quad (3.35)$$

where $b \triangleq [b_1^\top \ b_2^\top]^\top$. In order to let this system have a solution we require $b \in \mathcal{R}(Q)$, [Boyd 2004]. From (3.35) it is immediate to compute the *best-responses curves*:

$$\begin{cases} \arg \min_x F(x, z) = -Q_1^{-1}Q_2z - \frac{1}{2}Q_1^{-1}b_1 \\ \arg \min_z F(x, z) = -Q_3^{-1}Q_2^\top x - \frac{1}{2}Q_3^{-1}b_2 \end{cases}$$

Thus the difference equations under analysis are given by

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} (1-\beta)I_{n_x} & -\beta Q_1^{-1}Q_2 \\ -\beta Q_3^{-1}Q_2^\top & (1-\beta)I_{n_z} \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} - \frac{\beta}{2} \begin{bmatrix} Q_1^{-1}b_1 \\ Q_3^{-1}b_2 \end{bmatrix} \quad (3.36)$$

Note that $\beta = 0$ does not let the system move from its initial condition, and $\beta = 1$ provide the original undamped system. Thus, we additionally confine a priori β in the interval $]0, 1]$. A more general analysis, though possible, is considered to be beyond the interest of this work and it will not be carried out. The equilibria (x_e, z_e) of system (3.36) do not depend on β , and they are given by the solution of the following system of linear equations

$$\begin{bmatrix} I_{n_x} & Q_1^{-1}Q_2 \\ Q_3^{-1}Q_2^\top & I_{n_z} \end{bmatrix} \begin{bmatrix} x_e \\ z_e \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}Q_1^{-1}b_1 \\ -\frac{1}{2}Q_3^{-1}b_2 \end{bmatrix} \quad (3.37)$$

Clearly (3.37) is the same as (3.35). As a consequence Ω coincides with set of equilibrium points of (3.36). A first conclusion on the stability of system (3.36) can be drawn by employing Gerschgorin theorem (see Appendix A.9). Thus, a sufficient condition for the system to be stable is that $\|Q_1^{-1}Q_2\|_\infty < 1$, and $\|Q_3^{-1}Q_2^\top\|_\infty < 1$. In this case the stability of the system does not depend on β , as long as $\beta \in]0, 1]$. In the sequel we provide a more general analysis. We first name $B \triangleq \begin{bmatrix} I_{n_x} & Q_1^{-1}Q_2 \\ Q_3^{-1}Q_2^\top & I_{n_z} \end{bmatrix}$.

Noticing that $B = \begin{bmatrix} Q_1^{-1} & \mathbf{0} \\ \mathbf{0} & Q_3^{-1} \end{bmatrix} Q \triangleq Q_d^{-1}Q$, we can conclude that $\sigma(B) \subset \mathbb{C}_{\geq 0}$ by applying Lemma A.1 of Appendix A. Moreover, according to Lemma A.2 of

Appendix A its eigenvalues are real. By naming $y \triangleq [x^\top \ z^\top]^\top$, the autonomous system associated to (3.36) can be written as $y(k+1) = (I - \beta B)y(k)$. On the one hand, if $B \succ 0$, by employing $V(y) = y^\top y$ as a Lyapunov function, we have that $V(k+1) - V(k) < 0$ if $\beta < 1/\bar{\lambda}(B)$, where $\bar{\lambda}(B)$ is the greatest among the eigenvalues in $\sigma(B)$. On the other hand, if $B \succeq 0$, suppose that B has $r < n_x + n_z$ nonzero eigenvalues, then it exists a nonsingular matrix $T : TBT^{-1} = \Lambda = \text{diag}(\Lambda_r, \mathbf{0})$, where $\Lambda_r \triangleq \text{diag}(\lambda_1, \dots, \lambda_r)$. The linear transformation $\tilde{y} = Ty$ yields the system equations in the new coordinates $\tilde{y}(k+1) = (I - \beta\Lambda)\tilde{y}(k) - \beta\tilde{b}/2$, where $\tilde{b} \triangleq TQ_d^{-1}b = [\tilde{b}_1^\top \ \mathbf{0}^\top]^\top$, and $\tilde{b}_1 \in \mathbb{R}^r$. This is due to the fact that $\mathcal{R}(B) \equiv \mathcal{R}(Q)$, thus $Q_d^{-1}b \in \mathcal{R}(B)$ because $b \in \mathcal{R}(Q)$ by assumption. Eventually, we can split the dynamics of \tilde{y} in two, according to

$$\begin{cases} \tilde{y}_1(k+1) = (I_r - \beta\Lambda_r)\tilde{y}_1(k) - \frac{\beta}{2}\tilde{b}_1 \\ \tilde{y}_2(k+1) = \tilde{y}_2(k) \end{cases} \quad (3.38)$$

where $\tilde{y}_1 \in \mathbb{R}^r$. As for the previous case, we can employ the Lyapunov function $V_1 \triangleq \tilde{y}_1^\top \tilde{y}_1$ for the reduced order autonomous system associated to the first equation of (3.38). We obtain that $V_1(k+1) - V_1(k) < 0$ if $\beta < 1/\bar{\lambda}(\Lambda_r) = 1/\bar{\lambda}(B)$. ■

We can see how the result of Theorem 3.1 is only sufficient if we apply it to (3.33). According it, we are able to enforce stability of the system by choosing $\beta \in]0, 1/2[$, while we saw in the previous subsection that any value of β in $]0, 1[$ guarantees proper convergence. Although the results of Theorem 3.1 are not readily extendable to more general cost functions, and despite the fact that the considered system of equations (3.34) does not fully capture the CC-PSO algorithm dynamics, in the author's opinion such system still sheds light on the parallel update timing strategy behavior and it allows us to deduce an important heuristic. The main idea is to add the damping factor β in Algorithm 6 in order to reproduce a similar behavior to the one seen in the example of the deterministic system (3.33). In particular, we propose to substitute Step 11 of Algorithm 6 with

$$g_i^{new} = g_i + \beta \left(\arg \min_{\{b_{i,p}\}} \left\{ F_{i,p}^b \right\} - g_i \right)$$

This has the effect of damping the update of the global bests, and possibly reducing unwanted oscillations. When applied in the algorithm, it is difficult to compute β using mathematical tools. Thus, it is left as an additional parameter to be tuned by selecting a value in $]0, 1[$. Generally, such value should be chosen to reduce possible oscillations while letting proper convergence of the algorithm, which could be unnecessarily worsened by excessive damping. We name the modified algorithm as the *damped CC-PSO with parallel update timing*. Let us now apply this algorithm to the example of the previous subsection for different values of β . The results are shown in Fig. 3.8. We remark how the damping reduces the oscillations seen in Fig. 3.7a. Moreover, as expected, the convergence time increases as the damping increases, i.e. as $\beta \rightarrow 0$.

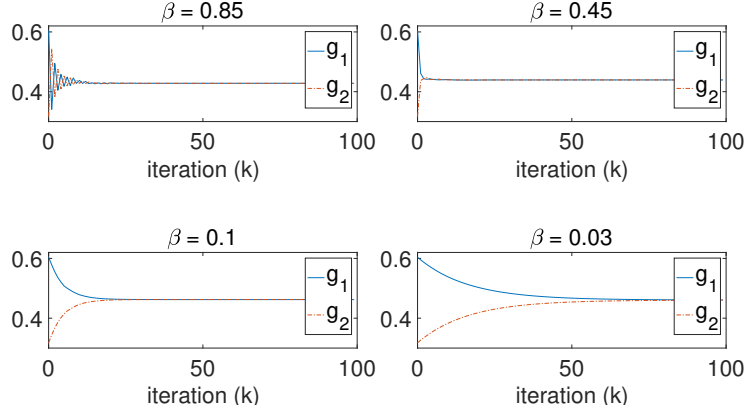


Figure 3.8: Global bests trajectories for different values of damping factor β .

Remark 3.10 We considered $\beta \in \mathbb{R}$. However, more in general, if the private variables x_i have dimension $n_i > 1$, and if x_i and x_j have same component-wise physical meaning for $i, j = 1, \dots, N$, then a different β value can be associated to different components of the private variables. An example is given in Section 3.6.

3.5.4 Algorithm Description

The *distributed* PSO proposed to solve (3.29) is concerned with reproducing the damped CC-PSO algorithm with parallel update timing shown in Subsection 3.5.3 in a distributed way, and it is based on the same communication assumption used for DPSO1, i.e. Assumption 3.1.

3.5.4.1 Exploiting the Problem Structure

Differently from the existing DPSO algorithms, which mainly employ consensus approaches to let the agents evaluate their contribution in the common cost function, we rather exploit problem (3.29) structure, and Assumption 3.1. The idea is that each agent i only needs to evaluate its contribution, i.e. its context vectors, in the agents private cost functions in which its own private control variable appears. In other words, agent i has to evaluate its context vectors in all f_k such that $k : i \in \mathcal{N}_k^p$, and, according to Assumption 3.1, this can be done by employing a direct communication without the need for consensus techniques. This is due to the fact that

$$\arg \min_{\{\mathbf{x}_{i,p}^g\}} \left\{ F(\mathbf{x}_{i,p}^g) \right\} = \arg \min_{\{\mathbf{x}_{i,p}^g\}} \left\{ \sum_{k:i \in \mathcal{N}_k^p \cup \{i\}} f_k(\mathbf{x}_{i,p}^g) \right\} \quad (3.39)$$

where we remind that $\mathbf{x}_{i,p}^g$ is the context vector associated to the particle $x_{i,p}$, defined in Step 5 of Algorithm 6. Clearly the same conclusions hold true for the personal best context vectors. As an example, let us consider agent 2 of Fig. 3.4. This only

needs to evaluate its context vectors $\mathbf{x}_{i,p}^g$, (and $\mathbf{b}_{i,p}^g$), $p = 1, \dots, N_p$, in f_1 , and f_2 , and compute $f_1 + f_2$, because f_3 and f_4 do not depend on variable x_2 . Indeed, if f_3 and f_4 were taken into account, this would simply add the same constant value, $f_3(g_3, g_4) + f_4(g_3, g_4)$, to each context vector evaluated in $f_1 + f_2$. Thus, this would provide the same result since the comparison of solution is done via the personal best update, which is performed in similar manners to Step 9 of Algorithm 6.

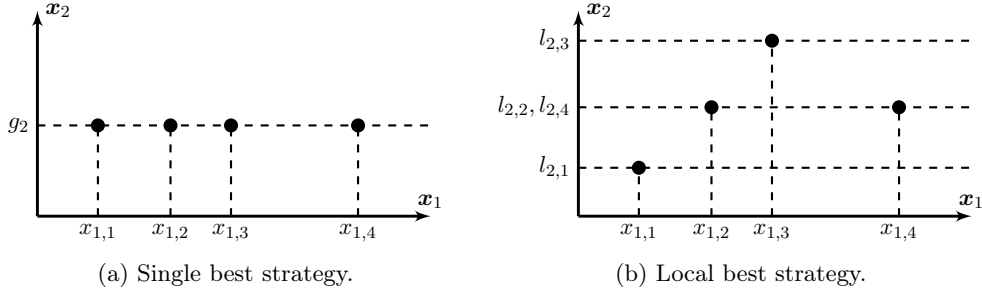
Remark 3.11 Notice that (3.39) is only possible thanks to the use of context vectors strategy. By recalling Remark 3.5 on DPSO1 requirements of information communication, this is precisely what allows DPSO2 to avoid performing an average consensus algorithm at each iteration step. In DPSO2, each agent is able to properly move its own particles on the only knowledge of how they fit in a partial common fitness function, thanks to (3.39). This was not possible in DPSO1, where functions are not evaluated in context vectors causing relationship (3.39) to be generally not verified. This fact has an important implication in the application of DPSO1 and DPSO2 to the wind farm optimization problem in Chapter 4.

3.5.4.2 Role of Local Bests

Instead of letting each particle compare its personal best with the ones of all the other $N_p - 1$ particles for the global best computation, usually a smaller subset is considered. As shown in Section 3.2, this practice is well-known to reduce the possibility of premature convergence. In particular, each particle $x_{i,p}$ has access to the personal bests of the particles belonging to a defined subset $S_{i,p}$. Thus, $x_{i,p}$ has *its own* knowledge of global best, which is *local* because it is restricted to the mentioned subset. Again, we indicate the local best associated to each particle with $l_{i,p}$. PSO equations (3.30) are then modified by simply replacing g_i with $l_{i,p}$. As done for DPSO1, we additionally require the subset $S_{i,p}$ to be the same for each particle among the agents having same p index. Thus we can drop its i index: $S_{i,p} = S_p$, $i = 1, \dots, N$. As for DPSO1, we make use of the singly-linked list structure for S_p , described in Algorithm 3, where we can select the number N_m of particles belonging to each subset S_p .

In the context considered in this work, the local best strategy has an important additional role in the convergence of the algorithm. CC-PSO with sequential update timing is known to suffer from a stagnation problem that is caused by the restriction that only one swarm is updated at a time [Van den Bergh 2004]. The parallel update timing version can present a similar problem. This is mainly due to the fact that each swarm puts together its context vectors by employing the single best collaboration technique, i.e. using the other swarm *global* bests. Thus the particles, and the personal bests, are restricted to *one* hyperplane of the search space, and this limits its exploration. The algorithm can be remarkably improved by employing the local bests when composing the context vectors, i.e.

$$\begin{aligned}\mathbf{x}_{i,p}^l &\triangleq (l_{1,p}, \dots, x_{i,p}, \dots, l_{N,p}) \\ \mathbf{b}_{i,p}^l &\triangleq (l_{1,p}, \dots, b_{i,p}, \dots, l_{N,p})\end{aligned}$$


 Figure 3.9: Local best based context vectors (*dots*) for the x_1 component.

where we renamed $\mathbf{x}_{i,p}^l$, and $\mathbf{b}_{i,p}^l$, respectively the context vector, and the personal best context vector associated to the particle $x_{i,p}$. This strategy is depicted in the example of Fig. 3.9, where we show a search space of dimension 2, $N_p = 4$ particles, and the corresponding context vectors for the swarm associated to the x_1 component. In order to show the improvement brought by the described strategy, let us provide the following example. Consider the cost function

$$F(x_1, x_2) = 3(1 - x_1)^2 e^{-x_1^2 - (x_2 + 1)^2} - 10(x_1/5 - x_1^3 - x_2^5) e^{-x_1^2 - x_2^2} - 1/3 e^{-(x_1 + 1)^2 - x_2^2} \quad (3.40)$$

In the region defined by $[-2.5, 2.5] \times [-2.5, 2.5]$, among its minima, function (3.40) has a global minimum in $A = (0.2282; -1.626)$ where its value is -6.5511 , and a higher local minimum in $B = (-1.347; 0.2045)$ where its value is -3.0498 . This is shown in Fig. 3.10. We run 1000 times the damped CC-PSO algorithm with parallel update timing for both the case of single best, and local best strategy. As far as the latter is concerned we select $N_m = 2$, while for both strategies we choose $N_p = 40$, and $\beta = 0.4$. While for the local best strategy the algorithm converges to the global minimum for the totality of the trials, the single best one attains A approximately the 70% of them while for the remaining 30% it converges to B . This fact is representative of the benefit gained by using the local bests to compose the context vectors.

3.5.4.3 Distributed Optimization Algorithm

The overall DPSO2 algorithm that we propose, is presented in Algorithm 7, and it is referred to the generic agent i . Each iteration of the algorithm requires each agent to exchange information with its neighbor agents according to \mathcal{G}_c twice. Moreover agent i evaluates its private cost function f_i , $2N_p(|\mathcal{N}_i^p| + 1)$ times, for a total number of $2N_p(|\mathcal{N}_i^p| + 1)\text{max_iter}$ evaluations, where $|\mathcal{N}_i^p|$ indicates the cardinality of the set \mathcal{N}_i^p . However note that, from experimental results, both max_iter , and \mathcal{N}_i^p appear to be small with respect to the usual values used in classic PSO.

Remark 3.12 *It has to be mentioned that the choice of N_m for the set S_p plays an additional role on the convergence of the algorithm. Indeed a too small value can prevent the particles to converge to the same value, i.e. $\forall i = 1, \dots, N: \lim_{k \rightarrow \infty} \|x_{i,p}(k) -$*

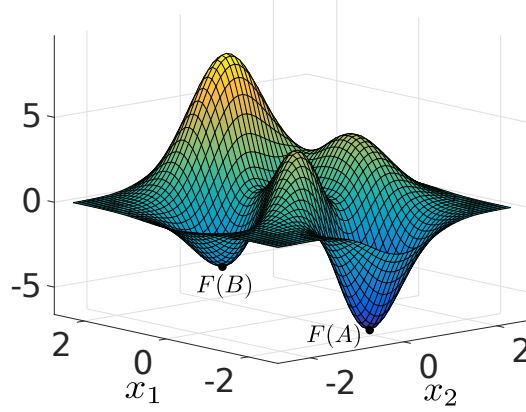


Figure 3.10: Plot of function (3.40).

$x_{i,t}(k) = 0, \forall p, t = 1, \dots, N_p$. In order to recover from this issue while maintaining a sufficient level of exploration as shown in the previous section, N_m can be made iteration-varying. For instance, it can be linearly increased at each iteration.

3.5.5 Benchmark Test

3.5.5.1 Simulation Setup

In the context of physically distributed agents, it seems impracticable to provide a rigorous test that is able to grasp the whole variety of real world optimization problems. Indeed, they may vary in the degree of interdependence among the agents, i.e. the structure of the physical graph \mathcal{G}_p , in the number of the agents involved, and in the structure of the private parameters, i.e. the structure of the cost functions. This is why, in this subsection, we choose to focus on the analysis of Algorithm 7 behavior with respect to the degree of *separability* of the functions involved as well as to the dimension of the optimization problem, as in the author's opinion, these are two major factors of influence in the convergence of the algorithm itself. The case of a real world distributed optimization problem is treated in Section 3.6.

We choose to consider the following setup. Based on [Li 2013a], we select a number of classic benchmark functions $F : \mathbb{R}^n \rightarrow \mathbb{R}$, and we distribute them among $N = n$ agents, each of which has a nonoverlapping private variable $x_i \in \mathbb{R}$, and a private cost function $f_i(x_i, \mathbf{x}_{ij}) = 1/N F(\mathbf{x})$. Note that the physical graph associated to such distributed problem is *complete*. In other words, each agent is influenced by and influences every other one, i.e. $|\mathcal{N}_i^p| = n - 1, \forall i$. This can be considered as the extreme case of what one can find in applications to real world multi-agent systems where the number of mutual interconnections among the agents is usually limited. The benchmark functions F chosen in this test are the well-known *Ackley*, *Griewank*, *Alpine*, and *Rastrigin* ones, which all have global minimum value equal to 0. The respective considered domains are $[-32, 32]^n$, $[-600, 600]^n$, $[-10, 10]^n$, and $[-5, 5]^n$.

Algorithm 7 DPSO2

Output: Local bests: $l_{i,p}$, $p = 1, \dots, N_p$

Initialization :

1: Randomly initialize $x_{i,p} \in [\underline{x}_i, \bar{x}_i]$, $s_{i,p} \in [-\bar{s}_i, \bar{s}_i]$, $p = 1, \dots, N_p$

2: $b_{i,p} = x_{i,p}$, $p = 1, \dots, N_p$

3: Randomly initialize $l_{i,p} \in [\underline{x}_i, \bar{x}_i]$, $p = 1, \dots, N_p$

LOOP Process

4: **for** $k = 1$ to **max_iter** **do**

5: Send $x_{i,p}$, $b_{i,p}$, $l_{i,p}$ $p = 1, \dots, N_p$ to all agents $k : i \in \mathcal{N}_k^p$, via \mathcal{G}_c

6: Wait to receive $x_{j,p}$, $b_{j,p}$, $l_{j,p}$ $p = 1, \dots, N_p$ from all agents $j \in \mathcal{N}_i^p$, via \mathcal{G}_c

7: Compose the context vectors associated to the its own particles $x_{i,p}$:

$$\mathbf{x}_{i,p}^l \triangleq (x_{i,p}, l_{j,p} : j \in \mathcal{N}_i^p), p = 1, \dots, N_p$$

8: Compose the context vectors associated to its own personal bests $b_{i,p}$:

$$\mathbf{b}_{i,p}^l \triangleq (b_{i,p}, l_{j,p} : j \in \mathcal{N}_i^p), p = 1, \dots, N_p$$

9: Compose the context vectors associated to the particles $x_{j,p}$ of its neighbors $j \in \mathcal{N}_i^p$:

$$\mathbf{x}_{j,p}^l \triangleq (x_{j,p}, l_{i,p}, l_{k,p} : k \in \mathcal{N}_i^p \wedge k \neq j), p = 1, \dots, N_p$$

10: Compose the context vectors associated to the personal bests $b_{j,p}$ of its neighbors $j \in \mathcal{N}_i^p$: $\mathbf{b}_{j,p}^l \triangleq (b_{j,p}, l_{i,p}, l_{k,p} : k \in \mathcal{N}_i^p \wedge k \neq j), p = 1, \dots, N_p$

11: Evaluate particle context vectors $\mathbf{x}_{i,p}^l$ in f_i : $f_{i,p}^x \triangleq f_i(\mathbf{x}_{i,p}^l)$, $p = 1, \dots, N_p$

12: Evaluate personal best context vectors $\mathbf{b}_{i,p}^l$ in f_i :

$$f_{i,p}^b \triangleq f_i(\mathbf{b}_{i,p}^l), p = 1, \dots, N_p$$

13: Evaluate particle context vectors $\mathbf{x}_{j,p}^l \forall j \in \mathcal{N}_i^p$ in f_i : $f_{i,p}^{x,j} \triangleq f_i(\mathbf{x}_{j,p}^l)$, $p = 1, \dots, N_p$

14: Evaluate personal best context vectors $\mathbf{b}_{j,p}^l \forall j \in \mathcal{N}_i^p$ in f_i : $f_{i,p}^{b,j} \triangleq f_i(\mathbf{b}_{j,p}^l)$, $p = 1, \dots, N_p$

15: Send $f_{i,p}^{x,j}$, $f_{i,p}^{b,j}$, $p = 1, \dots, N_p$ to the corresponding agent $j \in \mathcal{N}_i^p$, via \mathcal{G}_c

16: Wait to receive $f_{k,p}^{x,i}$, $f_{k,p}^{b,i}$, $p = 1, \dots, N_p$ from the corresponding agent $k : i \in \mathcal{N}_k^p$, via \mathcal{G}_c

17: Compute $F_{i,p}^x \triangleq f_{i,p}^x + \sum_{k:i \in \mathcal{N}_k^p} f_{k,p}^{x,i}$, $p = 1, \dots, N_p$

18: Compute $F_{i,p}^b \triangleq f_{i,p}^b + \sum_{k:i \in \mathcal{N}_k^p} f_{k,p}^{b,i}$, $p = 1, \dots, N_p$

19: Update personal bests for $p = 1, \dots, N_p$

$$(b_{i,p}^{new}, F_{i,p}^{b,new}) = \begin{cases} (x_{i,p}, F_{i,p}^x) & \text{if } F_{i,p}^x < F_{i,p}^b \\ (b_{i,p}, F_{i,p}^b) & \text{otherwise} \end{cases}$$

20: $b_{i,p} = b_{i,p}^{new}$; $F_{i,p}^b = F_{i,p}^{b,new}$

21: Update local bests for $p = 1, \dots, N_p$

$$l_{i,p}^{new} = l_{i,p} - \beta \left(\arg \min_{\{b_{i,p} \in S_p\}} \{F_{i,p}^b\} - l_{i,p} \right)$$

22: $l_{i,p} = l_{i,p}^{new}$, $p = 1, \dots, N_p$

23: Perform *modified* (3.30)

24: **end for**

25: **return** $l_{i,p}$, $p = 1, \dots, N_p$

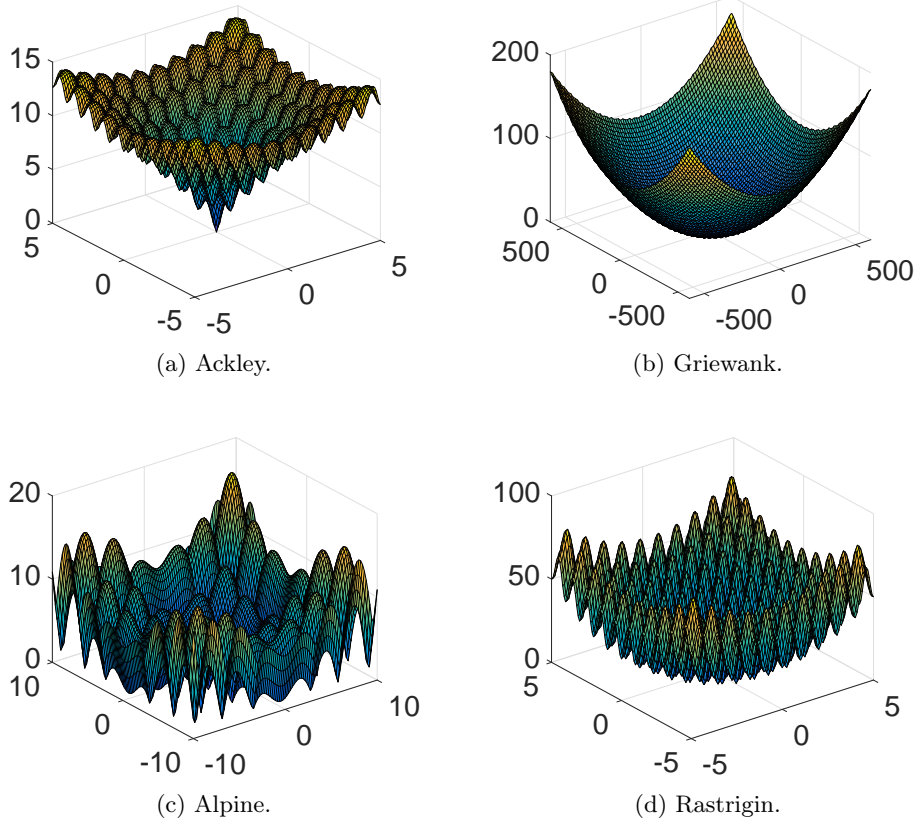


Figure 3.11: Benchmark functions.

Their plot for $n = 2$ is shown in Fig. 3.11. These functions are *fully separable*, i.e., following Definition 3.3, each of their components $x_i \in \mathbb{R}$ can be optimized independently from the other ones. In order to act on their degree of separability it is well-known practice to *rotate* them, i.e. the rotated function F_{rot} associated to F is defined as $F_{rot}(\mathbf{x}) \triangleq F(M\mathbf{x})$, where $M = \text{diag}(\bar{M}, I) \in \mathbb{R}^{n \times n}$, $I \in \mathbb{R}^{(n-m) \times (n-m)}$ is the identity matrix, $\bar{M} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix, and where $m \leq n$. By acting on m we are able to test different degrees of separability. For instance, $m = 0$, and $m = n$ respectively represent the fully separable, and fully nonseparable case. Moreover, for each function F we consider different problem dimension values n . For each n value, m is selected as a percentage $m\%$ of n , i.e. $m = \lceil m\%n \rceil$. Note that these functions have been chosen as they represent different degrees of complexity, and they are expected to provide different results, thus enabling a better comprehension of the algorithm behavior.

3.5.5.2 Test results

Two tests are proposed. The first one aims at providing insights of the algorithm performance and behavior for different values of n , and $m\%$ on the selected bench-

Table 3.1: Algorithm Parameters

$m\%$	N_p	N_m	β
0	15	2	1
33	15	2	1
66	50	2	0.8
100	50	2	0.8

(a) Ackley

$m\%$	N_p	N_m	β
0	25	25	0.5
33	40	2	0.5
66	40	2	0.5
100	60	2	0.5

(b) Griewank

$m\%$	N_p	N_m	β
0	15	2	0.8
33	15	2	0.8
66	30	2	0.8
100	30	2	0.8

(c) Alpine

$m\%$	N_p	N_m	β
0	75	15	0.1
33	80	25	0.1
66	80	25	0.1
100	120	25	0.1

(d) Rastrigin

mark functions. In the second one, we compare the algorithm rate of convergence against the required evaluations per agent with DPSO1 algorithm. As far as the first test is concerned, the main algorithm parameters associated to each function, and each chosen $m\%$ value are shown in Table 3.1. These have been chosen empirically. `max_iter` is set to 200. The mean and standard deviation values out of 30 trials for each function in the corresponding dimension and $m\%$ values are shown in Table 3.2. The algorithm performs well on both Ackley and Griewank functions, with no apparent dependence on $m\%$. Results slightly worsen as n increases. Similar trends are shown concerning Rastrigin function, i.e. the algorithm presents similar performances for fixed n , and variable $m\%$, while they generally decrease in comparable ways for different $m\%$ values as n increases. Moreover they do so with a faster rate with respect to the other two mentioned functions. This can be ascribed to the high multimodality of Rastrigin function as well as to the presence of several local minima, as shown in Fig 3.11d. Alpine test presents two trends. Performance decreases as either n or $m\%$ increases.

As far as the second test is concerned, for each benchmark function we consider one couple of parameters n , and $m\%$. Simulation results, and the corresponding parameters are shown in Fig. 3.12, where we compare DPSO1 and DPSO2 performance. The figure shows the mean values out of 30 trials of the benchmark functions evaluated in the global bests trajectories, i.e. the bests among the local bests. DPSO1 requires N_p function evaluations per agent at each iteration step as there is no constraint in the optimization problem, and thus no threshold ϵ to be set, while, for this particular test configuration, DPSO2 requires $2N_p n$ evaluations per iteration. This, again, can be considered as the worst case, as typically $|\mathcal{N}_i^p|$ is limited and consequently the number of evaluations too. Yet the two algorithms generally exhibit similar performance, and, for the case of Ackley and Rastrigin functions,

Table 3.2: Test Results

		n					
		4	10	25	45	70	100
$m\%$	0	(2.16±1.78)e-5	(12.4±6.36)e-5	(6.27±3.06)e-4	(14.9±5.06)e-4	(2.99±1.17)e-3	(4.22±1.47)e-3
	33	(2.20±1.95)e-5	(15.2±8.53)e-5	(8.32±3.58)e-4	(24.0±6.93)e-4	(4.31±1.78)e-3	(6.75±1.20)e-3
	66	(7.81±6.53)e-6	(5.82±2.46)e-5	(25.1±5.33)e-5	(6.80±1.88)e-4	(13.6±3.65)e-4	(22.4±6.06)e-4
	100	(10.6±7.37)e-6	(6.28±2.78)e-5	(28.9±5.84)e-5	(9.23±2.59)e-4	(19.9±4.79)e-4	(3.96±1.14)e-3

(a) Ackley

		n					
		4	10	25	45	70	100
$m\%$	0	(1.55±3.18)e-1	(6.07±7.72)e-3	(1.08±3.34)e-3	(7.76±9.56)e-3	(2.19±1.35)e-2	(11.3±9.95)e-2
	33	(13.3±9.20)e-3	(8.25±25.4)e-4	(8.56±8.74)e-5	(11.4±8.55)e-4	(10.3±4.06)e-3	(4.27±2.35)e-2
	66	(9.71±7.44)e-3	(4.40±23.6)e-2	(5.54±4.62)e-5	(1.78±2.93)e-3	(11.0±7.37)e-3	(4.55±2.72)e-2
	100	(8.14±6.14)e-3	(3.15±7.24)e-6	(6.83±7.01)e-5	(1.32±1.47)e-3	(11.9±7.52)e-3	(4.79±4.63)e-2

(b) Griewank

		n					
		4	10	25	45	70	100
$m\%$	0	(8.21±29.4)e-5	(8.41±13.9)e-4	(4.73±5.27)e-3	(3.79±9.74)e-2	(1.00±2.99)e-1	(5.25±14.1)e-1
	33	(3.33±7.10)e-4	(7.51±12.9)e-4	(3.75±8.80)e-2	(4.02±7.42)e-1	2.16±2.39	4.94±3.57
	66	(9.73±24.9)e-4	(4.34±7.17)e-3	(3.49±12.4)e-1	1.00±2.35	2.17±2.77	4.44±5.10
	100	(1.29±3.27)e-3	(2.17±11.7)e-1	(3.39±13.1)e-1	(4.09±3.31)e-1	3.83±8.52	8.64±11.90

(c) Alpine

		n					
		4	10	25	45	70	100
$m\%$	0	(5.57±13.2)e-1	5.96±1.21e+1	1.19e+1±3.38	2.59e+1±5.91	(5.19±1.23)e+1	(8.19±1.55)e+1
	33	1.35±6.67	2.71±2.25	1.28e+1±4.88	2.77e+1±8.07	(4.87±1.25)e+1	(7.43±1.50)e+1
	66	(2.32±4.28)e-1	2.68±1.59	1.18e+1±3.07	2.85e+1±8.40	(5.07±1.70)e+1	(7.95±2.06)e+1
	100	(6.93±30.6)e-1	(49.0±1.11)e-1	1.10e+1±3.20	2.54e+1±6.63	(4.31±1.23)e+1	(6.84±1.43)e+1

(d) Rastrigin

DPSO2 shows better convergence rate. This is mainly due to the fact that the proposed DPSO algorithm of this section requires less iterations to converge, as, unlike DPSO1, it belongs to the class of CC algorithms, which typically have good convergence rate properties. Note that, as pointed out in Remark 3.9, although the DPSO2 cost functions globally decrease along the global bests trajectories, this is not guaranteed at a local level. The overall experiment results suggest that

- Performance decreases as n increases at a pace that seems to depend on the degree of multimodality of the cost function.
- $m\%$ seems to have a more important role on asymmetric cost functions.
- DPSO2 presents good convergence rate properties when compared to DPSO1.

Remark 3.13 *Even though in general the algorithm exhibits lower performance for high values on n , we need to stress the fact that in this particular test $N = n =$*

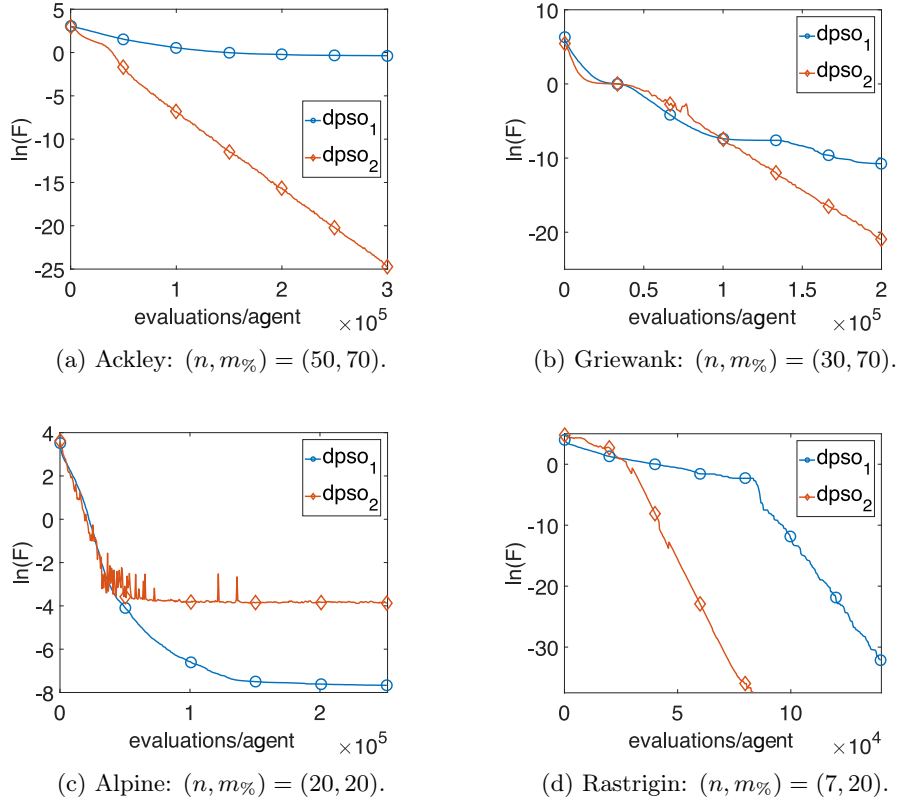


Figure 3.12: Mean convergence plots for the selected benchmark functions.

$|\mathcal{N}_i^p| + 1, \forall i$. Thus high n value both indicates a high number of agents, and high physical interdependence among them. However, in the problems we wish to address in this report, the number of physical connections among the agents of the considered physical systems is usually limited and does not grow with problem dimension.

Remark 3.14 Both DPSO1, and DPSO2 rely on Assumption 3.1. However, the former requires a consensus algorithm to be run on the whole system of agents at each iteration step. Thus, in practice, its convergence is worsened by a potentially high communication burden.

3.5.6 Constrained Optimization Extension

The results shown in Subsection 3.5.4 can be extended to the case in which the agents have private constraints of the form $g_i(x_i, \mathbf{x}_{ij}) \leq \mathbf{0}$, and $h_i(x_i, \mathbf{x}_{ij}) = \mathbf{0}$, where $g_i : \mathbb{R}^{n_i + \sum_{j \in \mathcal{N}_i^p} n_j} \rightarrow \mathbb{R}^{m_i}$, $h_i : \mathbb{R}^{n_i + \sum_{j \in \mathcal{N}_i^p} n_j} \rightarrow \mathbb{R}^{q_i}$, and where $\mathbf{0}$ is a vector of proper dimension with all zero entries. For this case, we refer to the definition of physical neighbor given in Section 3.4, i.e. Definition 3.1. The overall optimization problem takes thus the form of problem (3.20). As done for DPSO1, we employ Deb's rule to

handle the problem constraints, and we transform the equality ones into inequalities via the introduction of a positive threshold $\epsilon \in \mathbb{R}^+$. As a result, agent i generalized constraint functions assume the form of $\tilde{g}_i(x_i, \mathbf{x}_{ij}) \triangleq (g_i(x_i, \mathbf{x}_{ij}), |h_i(x_i, \mathbf{x}_{ij})| - \epsilon)$, where \tilde{g}_i is a vector of functions of dimension $m_i + q_i$.

In order to implement Deb's rule the agents have to evaluate a modified cost function, i.e. the fitness function, which takes into account the tournament selection described in Subsection 3.2.3.2. Referring to Algorithm 7, agent i additionally has to evaluate the context vectors of Steps 11 – 14 in the constraints functions \tilde{g}_i . Thus we additionally define $\tilde{g}_{i,p}^x \triangleq \tilde{g}_i(\mathbf{x}_{i,p}^l)$, and $\tilde{g}_{i,p}^b \triangleq \tilde{g}_i(\mathbf{b}_{i,p}^l)$ as far as agent i own context vectors are concerned, and $\tilde{g}_{i,p}^{x,j} \triangleq \tilde{g}_i(\mathbf{x}_{j,p}^l)$, and $\tilde{g}_{i,p}^{b,j} \triangleq \tilde{g}_i(\mathbf{b}_{j,p}^l)$ for agent i neighbor context vectors. These values are exchanged as done for the cost function values in Steps 15 – 16. The fitness function evaluation concerns Steps 17 – 18, where the cost function associated to the p -th particle of the i -th agent is modified as follows

$$\tilde{F}_{i,p}^x \triangleq \begin{cases} F_{i,p}^x & \text{if } \left\{ \tilde{g}_{i,p}^x, \tilde{g}_{k,p}^{x,i} : k : i \in \mathcal{N}_k^p \right\} \leq 0 \\ f_{max} + \sum_{t=1}^{m_i+q_i} \chi \left(\tilde{g}_{i,p}^x(t) \right) + \sum_{k:i \in \mathcal{N}_k^p} \sum_{t=1}^{m_k+q_k} \chi \left(\tilde{g}_{k,p}^{x,i}(t) \right) & \text{otherwise} \end{cases} \quad (3.41)$$

where f_{max} and χ are defined in Subsection 3.2.3.2. Same results hold true for the personal bests evaluation, in which case we name (3.41) $\tilde{F}_{i,p}^b$.

Remark 3.15 *As done for DPSO1 and pointed out in Remark 3.3, the tolerance ϵ used to handle the equality constraints can be made iteration-varying to let better search space exploration. Typically a linearly decreasing value is employed. This can be done by changing its value at each iteration of Algorithm 7. Differently from DPSO1, this modification does not require any additional function evaluation nor communication step.*

3.5.7 Conclusion

We presented a novel PSO algorithm applied to physically distributed systems to solve the class of optimization problems of (3.29), for which the agents have to cooperate in order to find the best common solution. A similar problem is typically tackled by cooperative co-evolutionary algorithms. However these are not readily applicable to the systems considered in this work because such algorithms rely on a problem decomposition that grasps the structure of the problem itself. We additionally focused on cooperative algorithms with parallel update timing parameter as they represent the most natural way to treat distributed systems while aiming at fast convergence. Thus, we carried out a new dynamic behavior analysis, and we proposed the introduction of a damping parameter, which has the role of reducing unnecessary algorithm trajectories oscillations caused by the parallel update timing parameter. Based on the aforementioned analysis, and on the assumption that each agent can directly communicate with its physical neighbors, we introduced a distributed PSO algorithm. Simulation results on four benchmark functions showed that it presents a decrease in performance as *both* the number of agents, and $|\mathcal{N}_i^p|$

increase. From a practical point of view though, this does not limit the field of application, as $|\mathcal{N}_i^p|$ has typically limited value. Belonging to the class of CC algorithms, the proposed one inherits fast convergence properties with respect to the required algorithm iterations. The price paid is the number of function evaluations, which increases as the agents neighborhoods increase. However, when compared to DPSO algorithm of Section 3.4, it shows good convergence properties with respect to the required function evaluations too. Eventually, we extended the algorithm application to the constrained case by employing Deb's rule. An engineering example of this last framework is provided in Section 3.6.

The main algorithm drawback is concerned with stagnation, especially when both the number of agents and their physical interdependence increase. This can be mainly ascribed to the parallel update timing parameter, and to the context vector composition. This problem was treated by employing the local bests strategy via the introduction of parameter N_m that allows to set the particles neighborhood dimension, and that enhances the search space exploration. Nonetheless, on the one hand this technique may not be sufficient to guarantee convergence to a good solution, especially for cost functions with high multimodality. On the other hand, a too small value of N_m , selected for better exploration, can prevent the agents particles to converge to a common global best value. Techniques that enrich the search space exploration while reducing the problem of premature stagnation still represent an open research question for the considered kind of systems. An interesting future perspective could be given by considering new algorithm modifications that would allow the common cost function to be decreasing in the global bests trajectories. Concerning this last topic, some insights could be unraveled by a deeper analysis of the influence of the introduced damping factor on the algorithm dynamics.

3.6 Optimal Power Flow Application

3.6.1 Related Works and Contribution

Optimal power flow (OPF) is a well-known engineering problem having a central role in the power dispatching and planning of the electric grid. Since its first formulation [Carpentier 1962], the problem has been widely studied and a great number of centralized algorithms have been proposed to solve it (see for instance [Lavai 2012] and references therein). Such research effort is motivated by the complexity of the problem itself, which happens to be highly nonconvex, mainly due to the power flow equations. Nonetheless, nowadays OPF is still object of great attention. Distributed generation, free electricity market as well as the increasing penetration of renewable energy sources in the grid require the OPF solutions to adapt to a definitely complexified electric network. In particular, the requirements for scalability and efficiency make centralized solutions no longer tractable and justify the research for distributed ones [Liu 2015].

In this regard, numerous solutions are available in the literature. Generally speaking, they usually split the problem in two steps. Firstly, it is reduced to a convex one,

either by approximation or exact relaxation. Secondly, the obtained optimization problem is distributed among the agents, i.e. the buses of the grid. This step usually capitalizes on ADMM technique because it proved good convergence and scalability properties (e.g. [Dall’Anese 2013, Peng 2014, Magnússon 2015]). Moreover, typically, the optimization problem is formulated in a distributed way via the introduction of an additional constraint called *consistency* or *consensus* constraint, which is responsible for the consistency, or agreement, of the variables which are shared among the agents of the network (e.g. [Peng 2014, Magnússon 2015, Liu 2015]). Considerable attention has been devoted to exact convex relaxation methods (e.g. [Peng 2014, Dall’Anese 2013]) because they enable to find the OPF global optimum in polynomial time. Unfortunately they are restricted to special network classes, such as radial ones. Other approaches (e.g. [Sun 2013, Magnússon 2015]), even if removing the assumptions on the network topology, are based on some convex approximations of the original OPF. In [Magnússon 2015], for instance, the authors rely on convexification of nonconvex constraints. Thus, neither optimality of the solution nor feasibility of the original problem are guaranteed. In [Liu 2015] a consensus based distributed OPF is proposed by employing available convex optimization tools. In addition the consensus constraint is taken into account via the method of the penalty function. Thus, it inherits the problem of distortion of the original objective function [Deb 2000].

In the following, we employ the two previously described algorithms of Section 3.4, and Section 3.5 to solve the OPF problem. Application of the first proposed algorithm to OPF is object of our work of [Gionfra 2017c]. As in [Magnússon 2015], this allows us to aim at scalability of the solution and to refer to an OPF problem formulation which is as general as possible. The use of PSO technique in this framework is also legitimated by the works of [Abido 2002, Wannakarn 2010, Syai’in 2012]), in which *centralized* PSO algorithms show good performance in solving the OPF problem. On the one hand, the discussion carried out in the sequel results in a first contribution, as to the author’s knowledge, PSO was never applied to solve OPF in a distributed manner. On the other hand, the following results allow us to show how the proposed DPSO algorithms perform in a real engineering problem.

3.6.2 Power System Model

The following notes are a simplified version of the power system modeling of reference [Andersson 2008]. The reader may refer to it for further details.

3.6.2.1 Power Flow Equations

The system equations are obtained via lumped-circuit model, or π -model, of an AC transmission line between two nodes of a power network k , and m , also called *buses*, and by using the equivalent mono-phase electric scheme. This is represented in Fig. 3.13a, where the complex variables E_k , and E_m are the phasors associated to the voltages, respectively, at bus k , and m , i.e. $E_k \triangleq U_k e^{j\theta_k}$, $E_m \triangleq U_m e^{j\theta_m}$,

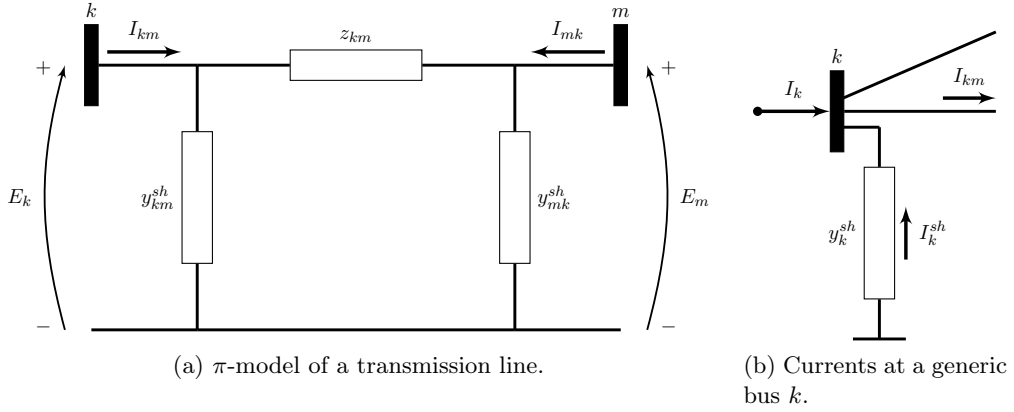


Figure 3.13: Circuits for power system modeling.

being U_k, U_m the voltage amplitudes, θ_k, θ_m their phase, and j the imaginary unit. Moreover, z_{km} is the series line impedance, in Ω , $z_{km} \triangleq r_{km} + jx_{km}$, with $r_{km}, x_{km} > 0$, and y_{km}^{sh} is the shunt admittance, in *siemens*, $y_{km}^{sh} \triangleq g_{km}^{sh} + jb_{km}^{sh}$, with $g_{km}^{sh}, b_{km}^{sh} > 0$. Here, for the sake of simplicity, the shunt admittances are supposed to be equal. In general $y_{km}^{sh} \neq y_{mk}^{sh}$. By considering the series admittance $y_{km} \triangleq z_{km}^{-1} = g_{km} + jb_{km}$, with $g_{km} > 0, b_{km} < 0$, and by employing Kirchhoff law, the complex current I_{km} in the transmission line is given by

$$I_{km} = y_{km}(E_k - E_m) + jb_{km}^{sh}E_k \quad (3.42)$$

where we neglected the term g_{km}^{sh} as it is usually very small. The complex power flowing from k to m is thus given by $S_{km} \triangleq P_{km} + jQ_{km}$, where P_{km} is the *active* power, and Q_{km} the *reactive* one, and it can be computed using (3.42), via

$$S_{km} = E_k I_{km}^* = y_{km}^* U_k e^{j\theta_k} (U_k e^{-j\theta_k} - U_m e^{-j\theta_m}) - jb_{km}^{sh} U_k^2 \quad (3.43)$$

The expression of P_{km} , and Q_{km} can be identified from respectively the real and imaginary parts of (3.43), yielding

$$\begin{aligned} P_{km} &= U_k^2 g_{km} - U_k U_m g_{km} \cos(\theta_k - \theta_m) - U_k U_m b_{km} \sin(\theta_k - \theta_m) \\ Q_{km} &= -U_k^2 (b_{km} + b_{km}^{sh}) + U_k U_m b_{km} \cos(\theta_k - \theta_m) - U_k U_m g_{km} \sin(\theta_k - \theta_m) \end{aligned}$$

Similar computations can be performed for the complex power flowing in the opposite direction, i.e. $S_{mk} = P_{mk} + jQ_{mk}$. Thus, the active and reactive losses on the transmission line between buses k , and m are given by

$$\begin{aligned} P_{km}^{loss} &\triangleq P_{km} + P_{mk} = g_{km}(U_k^2 + U_m^2 - 2U_k U_m \cos(\theta_k - \theta_m)) \\ Q_{km}^{loss} &\triangleq Q_{km} + Q_{mk} = -b_{km}^{sh}(U_k^2 + U_m^2) - b_{km}(U_k^2 + U_m^2 - 2U_k U_m \cos(\theta_k - \theta_m)) \end{aligned} \quad (3.44)$$

Let us now consider the net current injection from generators and loads at a network bus k , I_k , as shown in Fig. 3.13b. By Kirchhoff law, $I_k + I_k^{sh} = \sum_{m \in \mathcal{N}_k} I_{km}$, where

I_k^{sh} is the current injection from the shunt, whose admittance is y_k^{sh} , \mathcal{N}_k is the set of neighbor buses of k , and I_{km} is given by (3.42). Using this relation, it yields

$$I_k = \left(y_k^{sh} + \sum_{m \in \mathcal{N}_k} (y_{km}^{sh} + y_{km}) \right) E_k - \sum_{m \in \mathcal{N}_k} y_{km} E_m \quad (3.46)$$

By writing (3.46) for $k = 1, \dots, N$, where N is the number of buses in the network, the expression can be rewritten in matrix form as

$$\mathbf{I} = \mathbf{Y} \mathbf{E}$$

where $\mathbf{I} \triangleq \text{col}(I_1, \dots, I_N)$, $\mathbf{E} \triangleq \text{col}(E_1, \dots, E_N)$, and $\mathbf{Y} \triangleq \mathbf{G} + j\mathbf{B}$ is the nodal admittance matrix, whose entries are

$$\begin{aligned} Y_{km} &= -y_{km} \\ Y_{kk} &= y_k^{sh} + \sum_{m \in \mathcal{N}_k} (y_{km}^{sh} + y_{km}) \end{aligned}$$

The general matrix entry Y_{km} can be further detailed in real and imaginary part as $Y_{km} = G_{km} + jB_{km}$. Thus we can rewrite the expression of I_k as

$$I_k = \sum_{m \in \mathcal{N}_k \cup \{k\}} (G_{km} + jB_{km}) U_m e^{j\theta_m}$$

Thus, the complex power injection at bus k is

$$S_k = P_k + jQ_k = E_k I_k^* = U_k e^{j\theta_k} \sum_{m \in \mathcal{N}_k \cup \{k\}} (G_{km} - jB_{km}) (U_m e^{-j\theta_m}) \quad (3.47)$$

Eventually, the expressions of the active and reactive power injections at bus k can be obtained by identifying the real and imaginary parts of (3.47), yielding

$$P_k = U_k \sum_{m \in \mathcal{N}_k \cup \{k\}} U_m (G_{km} \cos(\theta_k - \theta_m) + B_{km} \sin(\theta_k - \theta_m)) \quad (3.48)$$

$$Q_k = U_k \sum_{m \in \mathcal{N}_k \cup \{k\}} U_m (G_{km} \sin(\theta_k - \theta_m) - B_{km} \cos(\theta_k - \theta_m)) \quad (3.49)$$

3.6.2.2 Power Flow Constraints

Each bus k of the network disposes of two independent variables, i.e. U_k , and θ_k , and two dependent ones, i.e. P_k via (3.48), and Q_k via (3.49). According to which variables are given and which are to be computed, we can define three main type of buses:

- PQ bus: P_k , and Q_k are specified, U_k , and θ_k are calculated.
- PU bus: P_k , and U_k are specified, Q_k , and θ_k are calculated.

- U θ bus: U_k , and θ_k are specified, P_k , and Q_k are calculated.

PQ buses usually represent *load* buses, as they impose a given demanded active power value P , and they are operated at constant power factor³ $\cos \varphi$. PU buses are used to represent *generation* buses with voltage control. Eventually, at least one U θ bus, also called reference or slack bus, is needed in the network to balance active power generation, load and losses. Moreover, it usually provides the voltage reference angle, typically set as 0° .

In the OPF problem addressed in the next subsection, we consider a slight modification to the above bus definition. In particular, we let each bus k act on its independent variables U_k , and θ_k in order to satisfy the active and reactive power constraints defining the type of bus itself. Thus, with a slight abuse of notation, let us consider:

- PQ bus: U_k , and θ_k are such that the given P_k , and Q_k values are satisfied.
- PU bus: U_k , and θ_k are such that the given P_k value is satisfied.
- U θ bus: θ_k is set as reference, and U_k is chosen to balance the active power in the network.

In this sense, according to the first given bus type definition, in PU and U θ buses, U_k can be considered *specified* by the solution of the OPF problem described in the next subsection.

Remark 3.16 *The voltage amplitude value is allowed in the typical interval of $[0.95, 1.05]$ p.u.⁴, which is treated as a box constraint on the independent variable U_k .*

3.6.3 Distributed Optimal Power Flow

3.6.3.1 Problem Formulation

OPF is concerned with minimizing a given cost function while satisfying the electric grid constraints. For its distributed setup, we consider each bus k of the network as an agent of the system that can act on its private optimization variables U_k , and θ_k . Conforming to the model developed in Subsection 3.6.2, each agent is electrically coupled, i.e. *physically* coupled, to those buses to which it is connected via a transmission line. According to such model, and Definition 3.1, we are thus able to define the set of physical neighbors \mathcal{N}_k^p associated to each agent k . Among the possible cost functions to be minimized usually employed for OPF, in this work

³Recall that the power factor $\cos \varphi$ is defined as the ratio of the active power over the apparent power $|S|$, i.e. $\cos \varphi \triangleq \frac{P}{\sqrt{P^2 + Q^2}}$.

⁴Recall that the per-unit (p.u.) system is the expression of system quantities as fractions of a defined base unit quantity. In the case of voltage p.u., this is done by normalizing the voltage values with respect to the network nominal voltage U_n .

we choose the sum of power losses in the transmission lines, which can be expressed as

$$\sum_{k=1}^N \sum_{m \in \mathcal{N}_k^p} (P_{km}^{loss} + Q_{km}^{loss}) \quad (3.50)$$

where P_{km}^{loss} , and Q_{km}^{loss} are given by respectively (3.44), and (3.45). In the sequel we neglect the shunt susceptance b_{ij}^{sh} , for it is usually verified that $|b_{ij}^{sh}| \ll |b_{ij}|$. Thus, we can simply define the generic transmission line loss as

$$L_{km}^{loss} \triangleq P_{km}^{loss} + Q_{km}^{loss} \simeq (g_{km} - b_{km})(U_k^2 + U_m^2 - 2U_k U_m \cos(\theta_k - \theta_m)) \quad (3.51)$$

This can be considered as a cost function associated to each transmission line of the network. Since the chosen agents coincide with the grid buses, in order to pose the optimization problem in the form of (3.20), i.e. to associate a private cost function to each agent, we consider

$$L_k^{loss} \triangleq \sum_{m \in \mathcal{N}_k^p} (g_{km} - b_{km})(U_k^2 - U_k U_m \cos(\theta_k - \theta_m))$$

It is then easy to see that $\sum_{k=1}^N L_k^{loss}$ is equivalent to (3.50), i.e. the sum of power losses in the network. Notice also that $L_k^{loss}(U_k, \theta_k, \mathbf{U}_{km}, \boldsymbol{\theta}_{km})$, where, according to the problem formulation used for both DPSO1 and DPSO2, $\mathbf{U}_{km} \triangleq \{U_m \in \mathbb{R} : m \in \mathcal{N}_k^p\}$, and $\boldsymbol{\theta}_{km} \triangleq \{\theta_m \in \mathbb{R} : m \in \mathcal{N}_k^p\}$. In other words, each agent private cost function generally depends on its own private variables, and on the ones of its physical neighbors.

As far as the optimization problem constraints are concerned, they are given by the box constraints on the voltage amplitude, (see Remark 3.16), and by constraints on active and reactive power according to the type of bus defined in Subsection 3.6.2, and via equations (3.48), and (3.49). Notice that, having considered (U_k, θ_k) as the private variable of the generic agent k , this allows a constraint function formulation in the form of the ones appearing in (3.20). Moreover, we consider an additional typical constraint concerning the generators, i.e. the PU buses. These are required to keep their reactive power within given limits with respect to their active power production. Indeed the reactive power supplied above a Q/P ratio of $(\tan \varphi)_{max}$ or absorbed under $(\tan \varphi)_{min}$ is subject to the application of a charge⁵. The overall

⁵In France, for instance, it amounts to 16.3 €/MVARh for electric grids whose nominal voltage is in [50, 130] kV. In particular $(\tan \varphi)_{max}$, $(\tan \varphi)_{min}$ are set respectively as 0.4, and -0.35 . According to the document about TURPE 4 (Tarifs d'Utilisation du Réseau Public de Distribution d'Électricité), the mean cost for the active power losses is estimated as 53 €/MWh

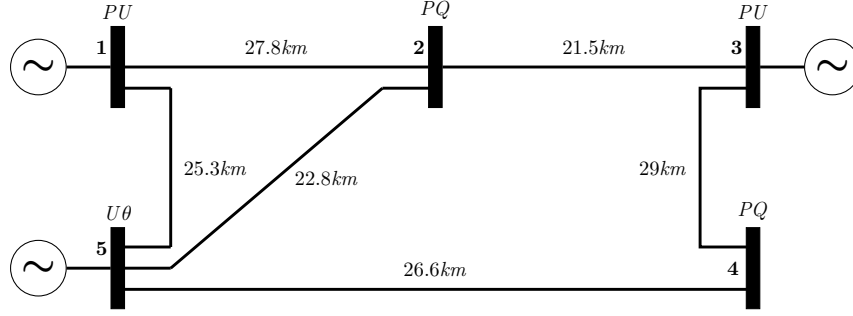


Figure 3.14: 5-bus electric grid example.

OPF optimization problem, written in the form of (3.20), is given by

$$\begin{aligned}
 & \min_{\{(U_k, \theta_k), k=1, \dots, N\}} \sum_{k=1}^N L_k^{loss} \\
 & \text{subject to} \\
 & 0.95 \leq U_k \leq 1.05, \quad k = 1, \dots, N \\
 & \text{via (3.48), (3.49)} : \begin{cases} P_k = \bar{P}_k, Q_k = \bar{Q}_k & \text{if } k \text{ is a PQ bus} \\ P_k = \bar{P}_k, Q_{k_{min}} \leq Q_k \leq Q_{k_{max}} & \text{if } k \text{ is a PU bus} \end{cases}
 \end{aligned}$$

where \bar{P}_k , and \bar{Q}_k are problem data. In particular, in this work, \bar{Q}_k is obtained from \bar{P}_k by imposing the PQ buses to operate at constant power factor $\cos \varphi = 0.97$.

3.6.3.2 Application of DPSO1 and DPSO2

Having posed the OPF problem in the form of (3.20), we are now able to employ the distributed algorithms described in Section 3.4, and Section 3.5. We additionally consider satisfied communication Assumption 3.1, on which both algorithms rely. We illustrate and compare their performance on a 5-bus grid, shown in Fig. 3.14, which is a modified scheme of a grid example in [Andersson 2008]. The considered grid has nominal voltage equal to 63 kV, and the grid cables impedance are set to be $R = 0.15 \Omega/km$, and $X = 0.21 \Omega/km$. Bus 5 is the $U\theta$ bus, and it can be considered as a source substation. The generators are prescribed to produce $\bar{P}_1 = 50 MW$, $\bar{P}_3 = 70 MW$, and load buses require $\bar{P}_2 = 60 MW$, $\bar{P}_4 = 85 MW$.

DPSO1 Settings

We choose $N_p = 70$ particles for each bus, and `max_iter` = 600. Each particle has access to the whole set of particles belonging to the same bus, i.e. $N_m = N_p$. After having normalized the power and voltage values, as far as the equality constraints are concerned, as suggested in Remark 3.3, we consider a threshold ϵ linearly decreasing from $\bar{\epsilon} = 1$ to $\underline{\epsilon} = 10^{-5}$ until the 90% of the iterations. For the considered graph, finite-time average consensus algorithm to be run in Algorithm 4 to let the agents

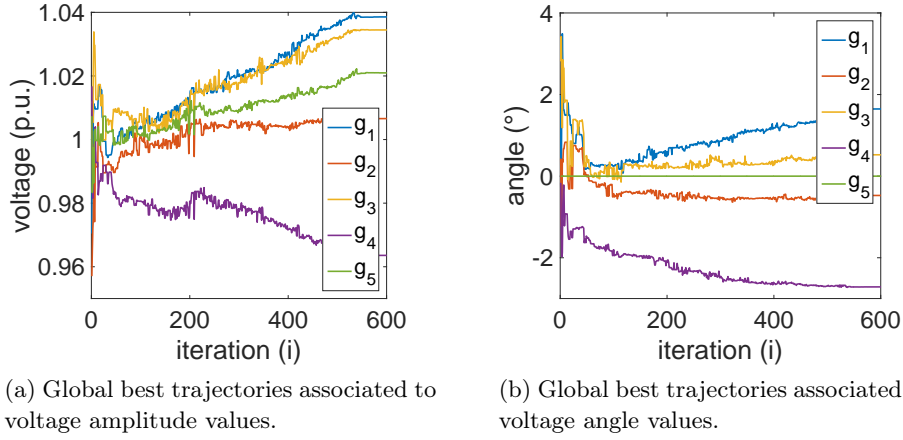


Figure 3.15: Global best trajectories for the OPF problem during the run of DPSO1.

get knowledge of the common fitness function evaluation, only requires 2 steps to be solved. Eventually, for the specific application considered in this example, as pointed out in Remark 3.6, we do not consider any perturbation operator, as by numerical simulations it does not seem to bring any enhancement in the optimality of the solution.

DPSO2 Settings

We choose $N_p = 70$ particles for each bus, and $\text{max_iter} = 70$. Each particle has access to an iteration-varying set of particles belonging to the same bus. In particular, N_m is chosen to be linearly increasing from a value of 2, (see Remark 3.12). Differently from DPSO1 settings, we do not consider here an iteration-varying value for ϵ . After normalization of power and voltage values, it is set to $\epsilon = 10^{-4}$. Eventually, as suggested in Remark 3.10, we choose two different damping factors for U , and θ variables, respectively $\beta_U = 0.6$, and $\beta_\theta = 0.8$.

Simulations

Let us name $g(i)$ the best among all the personal bests $b_{k,p}(i)$ in the network at iteration i , i.e. the global best. Thus, in Fig. 3.15, and Fig. 3.16 we are able to see a typical global best trajectory during the performance of respectively DPSO1 and DPSO2, where g_k , $k = 1, \dots, 5$ identifies the k -th component of g . As a first remark, we see how, in terms of speed convergence, DPSO2 outperforms DPSO1, requiring ~ 60 iterations to converge, against the 600 iterations of DPSO1. Moreover, recall that DPSO2 requires less communication steps as it does not need any average consensus algorithm to be performed. By a further comparison of DPSO1 trajectories of Fig. 3.15 with the one of DPSO2 of Fig. 3.16, we can also observe that the former exhibit a particular slower pace. This is essentially due to the need, in DPSO1, for considering an iteration-varying threshold ϵ . This fact can be further

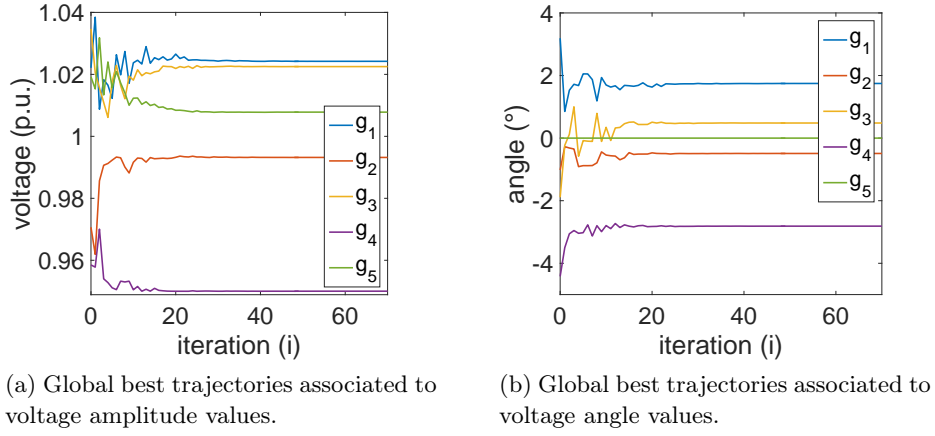


Figure 3.16: Global best trajectories for the OPF problem during the run of DPSO2.

analyzed in Fig. 3.17. Indeed, recalling Deb's rule, a particle is assigned with a sum of constraints value whenever it is not feasible, and with a common cost function value otherwise. At the beginning, being randomly initialized, they are very likely to be unfeasible. Nonetheless they reach feasibility quite quickly since ϵ is still large enough. As iterations grow, ϵ becomes smaller, thus, the common cost function respecting constraints is updated, and this explains its increasing value from about iteration 100. At the very end, all particles move according to their sum of constraints value since ϵ reaches a very small value. However, this fact does not have to be interpreted as nonconvergence. Indeed, at this point, the particles would only adjust their position in a small neighborhood of their last feasible visited position. This fact is confirmed by a maximum convergence quadratic error: $\max_{i,p} \|x_{i,p}(\text{max_iter}) - g(\text{max_iter})\|^2$ being of the order of 10^{-7} , which shows convergence of the algorithm. Performance is then compared out of 15 trials. Optimal values, and equality constraints are shown for both DPSO1, and DPSO2 respectively in Table 3.3, and Table 3.4. These show that good performance is achieved, especially concerning the respect of the power flow constraints, as the ideal ratio \bar{P}/P_k , and \bar{Q}/Q_k should be 1. Eventually, the total optimized active and reactive power losses are $(6.16 \pm 0.13) \text{ MW}$, and $(8.63 \pm 0.19) \text{ MVAR}$ concerning DPSO1, and $(6.53 \pm 0.17) \text{ MW}$, and $(9.15 \pm 0.23) \text{ MVAR}$ for DPSO2. All in all, we can conclude that DPSO1 performs better in terms of optimality of the solution. However, this is attained at the cost of a much higher number of iterations, and the according communication burden, which makes DPSO2 a more appealing solution for practical implementation. Indeed, according to the situation, OPF may be required to be solved with almost real-time timing.

3.6.4 Conclusion

We presented a novel distributed approach to solve the OPF problem. Since the proposed algorithms belong to the class of metaheuristics optimization, no assump-

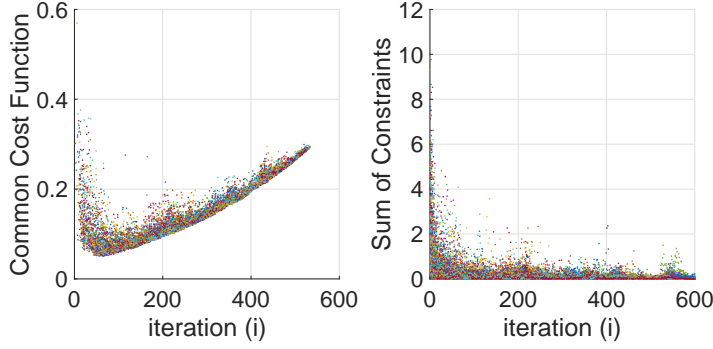


Figure 3.17: Particles common cost function, and sum of constraints during iterations of DPSO1.

Table 3.3: DPSO1 Optimal Variables and Equality Constraints

bus n ^o	1	2	3	4	5
U (p.u.)	1.0377 ± 0.0093	1.0046 ± 0.0114	1.0313 ± 0.0119	0.9612 ± 0.0127	1.0194 ± 0.0122
θ (°)	1.6261 ± 0.1020	-0.4530 ± 0.0693	0.6074 ± 0.1232	-2.6792 ± 0.0855	0
\bar{P}/P_{dps0}	1.0000 ± 0.0000	1.0000 ± 0.0001	1.0000 ± 0.0001	1.0028 ± 0.0093	—
\bar{Q}/Q_{dps0}	—	1.0001 ± 0.0001	—	1.0000 ± 0.0001	—

Table 3.4: DPSO2 Optimal Variables and Equality Constraints

bus n ^o	1	2	3	4	5
U (p.u.)	1.0263 ± 0.0085	0.9946 ± 0.0040	1.0155 ± 0.0045	0.9518 ± 0.0032	1.0175 ± 0.0067
θ (°)	2.0407 ± 0.3298	-0.1387 ± 0.2086	1.2091 ± 0.4225	-2.4581 ± 0.2092	0
\bar{P}/P_{dps0}	1.0000 ± 0.0395	1.0001 ± 0.0263	1.0000 ± 0.0401	1.0001 ± 0.0059	—
\bar{Q}/Q_{dps0}	—	1.0001 ± 0.1427	—	1.0002 ± 0.0812	—

tion are needed concerning either the cost function of the optimization problem, and the network topology. Both DPSO1 and DPSO2 were tested and evaluated on the addressed OPF problem. While the former achieves better results in terms of optimality of the solution, this is obtained at the cost of both high number of iterations, and communication burden, which result in high convergence time. On the other hand, as expected, DPSO2 converges to a relatively good solution in a few number of iterations, which is a typical characteristic of CC algorithms. Together with the lower number of required communication steps, this fact leads DPSO2 to

a much lower convergence time with respect to DPSO1. One may conclude that, if the communication graph of the multi-agent system is well-connected, i.e. it has a relatively small radius and diameter allowing the finite-time average consensus to be solved in a few steps, then DPSO1 should be considered as, at least in the given example, it provides a better solution. For more complex networks, the pitfalls of DPSO1 can be widely recovered by employing DPSO2, still obtaining good solutions.

Wind Farm Control

Contents

4.1	Introduction	126
4.1.1	Problem Motivation	126
4.1.2	Related Works	127
4.1.3	Contribution	131
4.2	Wake Model	132
4.2.1	Modified Park Wake Model	133
4.2.2	Single Wake Model	134
4.2.3	Wind Farm Power Function	138
4.3	Hierarchical Control	139
4.3.1	High Level Optimization Problem	139
4.3.2	Yaw Angle Influence on the Optimization Problem	142
4.4	Distributed Optimization	146
4.4.1	Application of DPSO1	146
4.4.1.1	Communication Settings	146
4.4.1.2	Problem Formulation	147
4.4.1.3	Simulation	148
4.4.1.4	Wake Model Approximation	150
4.4.2	Application of DPSO2	152
4.4.2.1	Discussion on Wake Model Approximation	152
4.4.2.2	Communication Settings	153
4.4.2.3	DPSO2 Algorithm for Wind Farm Optimization	156
4.4.2.4	Simulation	158
4.5	Hierarchical Control Evaluation	162
4.5.1	Combining High Level Optimization and WT Controllers	162
4.5.2	A Wind Farm Example Test	163
4.6	Conclusion and Future Perspectives	165
4.6.1	Conclusion	165
4.6.2	Future Perspectives	166

4.1 Introduction

4.1.1 Problem Motivation

IN recent years, we have witnessed a relevant increase in the installation of wind farms composed of several WTs, e.g. more than one hundred of units for some off-shore WFs. In the meantime, developments in the field of control and optimization, along with the mature technology level of variable-speed variable-pitch variable-yaw WTs, have pushed the WF control targets further ahead towards a better exploitation of the wind source. In particular, the great adaptability of modern WTs to a wide range of wind conditions for the maximum power capture as well as a better understanding of the aerodynamic phenomena involved in the WFs, suggested to take in consideration the aerodynamic interaction among the WTs, when the power maximization of large wind farms is concerned. Indeed, when extracting kinetic energy from the wind, a wind turbine causes a reduction of the wind speed in the downstream wake. As a result a turbine, standing in the wake of an upstream one, experiences a reduction of available wind power. Intuitively as the number of wind turbines of a wind farm increases, the *wake effect* becomes more important, so that considering it when optimizing the wind production proves potential gain with respect to classic individual turbine MPPT operating mode. As a matter of fact, in such situation, a *greedy* control, according to which each WT tracks its *own* maximum available power, no longer guarantees the maximization of the power extraction at the WF level. This mainly justifies a growing interest in *cooperative* methods to control wind turbines belonging to large wind farms.

As shown in Chapter 2, if at WT level there exist several situations for which the control objective deviates from the power maximization one, it is reasonable to think that this holds true at the WF level too. When a WT is required to track a de-loaded power reference to sustain the grid, e.g. for frequency support purposes or power curtailment constraints, this has to be managed by a higher control level that ensures that the WF, as a whole, respects the aforementioned grid requirements. From the above discussion it is clear that a WF controller is needed to manage the *power production sharing*, among its WTs, according to different scenarios, which define the control objectives. A comprehensive survey on the WF control state of the art can be found in [Knudsen 2015], where the authors identify three main WF objectives summed up as follows

- I Maximize the total WF active power.
- II Follow a reference for the total WF active power.
- III Do I and II while minimizing fatigue loading for the WTs in the WF.

Notice that a third point has been added to the list of the discussed ones, namely the fatigue reduction of the WF. Indeed, when a WF is required to follow a particular power profile, usually imposed by the grid operator, and this happens to be lower than the WF extractable power, than the power production can be *fairly*

shared among the WT's to evenly distribute the mechanical load.

A WF controller has to deal with a larger quantity of information with respect to the WT local ones. It has to process and communicate a growing number of variables with the increasing number of WT units, while respecting the dynamics of the system involved. This is why, recently, a great research effort has been given in developing adequate control architectures to fulfill requirements of system reliability while being fast enough to grasp its dynamics. A first major WF control architecture widely used in the literature is the *centralized* hierarchical one. This classic approach is based on a higher centralized slower controller, responsible for delivering the power references to the local WT faster ones, on the base of grid operator requirements and system measurements. Typically, when WF aerodynamic effects are taken into account, the problem solved by this higher level controller reduces to an optimization one, i.e. dynamics of the system are neglected in the problem formulation because of the complexity of the system itself. Still the requirements for fast computation have to be fulfilled. More recently, there has been a growing research interest in *distributed* solutions to the WF control problem. Even though, typically, such approaches provide suboptimal solutions with respect to their centralized counterpart, they allow some important practical features that justify their interest. Notice, that they still preserve a hierarchical architecture. In other words, here, the adjectives *centralized* and *distributed* refer to the higher control level. In this sense, in this part, both the aforementioned hierarchical architectures present a *decentralized* lower control level. That is to say that, once the optimal power references are computed by the higher optimization level, they are delivered to the local WT controllers, as the one treated in Part I, and, at this point, no additional information is exchanged by them. The case in which *also* the local WT controllers are allowed to exchange information is treated in Part III. In this case the lower control level in the hierarchical pyramid is no longer decentralized, and it is said to be *distributed*.

4.1.2 Related Works

Modeling the aerodynamic coupling among the WT's of a WF, i.e. their wake interaction, also known as *wake effect*, with high fidelity degree is not a trivial task. This involves numerical solutions to systems of partial differential equations. In other words, in principle, computational fluid dynamics (CFD) simulation should be employed. However, for the sake of engineering applications, such as the real time control of a WF, for which, as mentioned above, fast algorithms should be employed, a simplified wake model should be considered. As far as the latter is concerned, even though in the literature there exist many wake model representations, they basically are variants of the pioneer work of [Jensen 1983], who first proposed the single wake model, i.e. describing the wake interaction between two WT's, and [Katic 1986], who introduced the wake model accounting for multiple WT's interactions. This, also known as *Park wake model*, describes a piece-wise linear wind speed profile distribution within a WF, and it is based on the assumption

that the wake behind the WTs expands linearly, according to a *wake constant* k_w , in the downstream direction, and that the wind inside the wake is constant at equal distance from the according WT. Many software are nowadays available based on Park model. For instance, one can cite WAsP (Wind Atlas Analysis and Application Program) [Mortensen 1998], which, among its applications, allows calculation of the wind farm production. Based on a constant selected k_w value, given the WT thrust coefficient C_T/λ , and the WF layout, it estimates the wake power losses and, thus, the net annual energy production (AEP). Software like WAsP let good estimation of WF power generation for the AEP, via constant k_w values, by employing high time scale averaged quantities, as the wind signal one, in the simulation. As a consequence, it has been argued that model variations should be considered in order to grasp the variability of the wind source, and the according validity of the model, for more real time applications. Authors of [Herp 2015] propose a finer Park model calibration, based on *ensemble models*, to depict the flow variability in a lower time scale. This is done by considering a variable k_w value, and by relating it to the upstream free wind direction and modulus via a probability density function. Similarly, in [Peña 2014], the authors suggest a Park model modification based on different k_w values to meet different atmospheric stability conditions.

One of the disadvantages of Park wake model is that it does not include the influence of the WTs yaw angle in the wake profile. In [Fleming 2015], a CFD tool named SOWFA (Simulation for Onshore/Offshore Wind Farm Applications) was developed to analyze the yaw impact, and authors of [Gebraad 2016] proposed a parametric wake model, considering the yaw effect, to be tuned either on CFD simulation or on real wind farm data. Based on it, [Park 2015a] provided a *continuous* analytic wake model, which can be used for efficient gradient-based optimization algorithms, as well as for model validation purposes. All the aforementioned wake models are based on underlying steady state assumptions, i.e. given the WTs operating conditions, and the free upstream wind value, the wind distribution is computed by a *static* wake model. In other words, the dynamics involved in the wind propagation within a WF are neglected. This approximation allows these models to be suitable to describe the problem addressed in terms of an optimization one, e.g. layout optimization and power maximization. In [Soleimanzadeh 2010], it is presented a first attempt to consider the dynamics of the wind within a WF, while allowing the use of efficient control algorithms that are not based on CFD simulation. In particular, neglecting the yaw influence, a state space wake representation is obtained from a system of partial differential equations via finite volume method, and this allows the employment of control theory results rather than static optimization ones.

A first straightforward application of the wake model and its influence on a WF power production is represented by the problem of *optimal layout* of the wind turbines, i.e. where they should be placed within a WF in order to minimize the wake interaction. Plenty of works have been proposed in this regard. One can cite for instance [Larsen 2011, Park 2015b, Feng 2015]. In this case, the optimization problem has to be solved off-line, and this allows more complex wake models and validation via CFD simulation to be used. Moreover, the layout optimization problem is con-

cerned with large time scales, typically annual, and the objective function is usually enriched with other terms taking into account the costs of the WTs, infrastructures, etc.

When the WF layout is given, still some optimal control of a WF can be seek via optimization and control methods, and by employing the simplified wake models mentioned above. The control objectives are, thus, mainly the one identified in Subsection 4.1.1. The research literature is vast in this regard, and it can be mainly divided in two main groups according to the chosen control architecture, namely centralized and distributed solutions. As previously introduced, we restrict the case of the former class to the centralized *hierarchical* one, which are typically based on SCADA system. In the centralized architectures that actively consider the wake effect in the control problem, typically a *static* WF power generation model is employed, leading to an optimization problem formulation. Research works are then mainly conceived only with the solution of such optimization problem, which is the task of the higher level controller, under the assumption of existing local WT ones. Moreover, the addressed optimization problem is usually concerned with the *unconstrained* WF power maximization one. In other words, neither the system physical constraints such as the WTs nominal power $P_{e,n}$, and the one imposed by the grid operator, are directly considered in the problem formulation. Thus, the provided solutions are confined to the case in which the total available WF power is lower than the maximum allowed one. Alternatively, one can *a posteriori* constrain the solution of the unconstrained problem. However, this generally leads to a suboptimal solution. As it will be explained in the following sections, the optimization problem is nonlinear and nonconvex, and adding the mentioned constraints to it brings the problem to a higher level of complexity, because they are typically nonlinear nonconvex too. An exception to the described solutions can be found in the work of [Soleimanzadeh 2012], where the authors capitalize on the *dynamic* wake model of [Soleimanzadeh 2010]. The problem addressed is then posed as an optimal constrained control one, and it is solved by MPC means. A similar solution is also hypothesized by [Spudis 2010].

Generally speaking the most employed centralized architectures are based on gradient-based optimization algorithms. One can cite, for instance, [Heer 2014, Park 2015a, Herp 2015]. Authors of [Park 2015a] capitalize on sequential convex programming techniques, with yaw influence consideration, enabled by the choice of a continuous and differentiable wake model. In [Herp 2015], authors aim at speeding up the optimization problem convergence, by using a sequential optimization method. In particular, with the knowledge of the upstream free wind direction, they exploit the problem structure by letting only the parameters of a WT to be active at a time, starting from the most upstream WT towards the most downstream one. The aforementioned optimization approaches suffer from convergence to local optima, and their solution is affected by the initial guess (see Appendix A.7). This is why *global optimization* methods have been proposed for the sake of controlling a WF. An example is given by authors of [Tian 2014], who capitalize on particle swarm optimization technique. In this reference, system constraints are also actively con-

sidered in the problem formulation. However, the authors do not mention how these are managed in the algorithm, which is known to be a nontrivial problem in metaheuristic optimization. Although, as shown above, there exist simplified wake models to be used in the optimization problem, they may lack of reliability because of the complexity of the aerodynamic phenomena. Moreover, they make the optimization problem generally difficult to be solved. This is what motivates the search for model-free data-driven solutions to the power maximization problem. An example is given in [Gebraad 2016] where the authors considered a game-theoretic approach to optimize the WTs yaw, and by [Park 2016], where the wake model is learned on-line by testing the WTs yaw and axial induction factors, and Bayesian optimization technique is employed aiming at reducing the convergence time to a solution.

The centralized approaches suffer from some important drawbacks that can be summarized as follows

- High computation and communication burden.
- Non-scalability of the architecture, meaning that the problem gets more complex as the system size increases.
- Non-modularity of the architecture, meaning that the removal or addition of a WT unit makes the entire control solution to be re-designed.
- Single point of failure, i.e. if the central controller fails so it does the entire system.

These disadvantages can be at least partially recovered by distributed architectures. In this case, the wind turbines would solve the optimization problem cooperatively by exploiting a communication graph on which they share their own local information. In the literature, the works concerning this kind of architecture are mainly based on model-free approaches. One can cite [Gebraad 2013, Barreiro-Gomez 2015] for methods based on gradient estimation, [Marden 2013] as far as game-theoretic approaches are concerned, and again [Park 2016] for a Bayesian approach. In [Ebegbullem 2016], the authors make use of the *extremum seeking* technique to estimate the cost function gradient, coupled with results of consensus control, to solve the distributed power maximization problem. The main drawback of the mentioned data-driven distributed approaches mainly regards the speed of convergence. This is due to the fact that the WF control parameters have to be first tested on the real plant before knowing their real effect. Moreover, these approaches limit the system parameters exploration to those that guarantee a safe system exploitation while being tested.

Distributed architectures are also widely employed in the literature to address the problem of fatigue minimization while satisfying grid power constraints. In this framework though, the wake model is usually not considered in the problem formulation, and this allows it to be treated from a control perspective, i.e. generally the system dynamics are considered. However, if in the considered WF, the wake effect

cannot be neglected, then the control should be validated on a model that takes it into account. Usually, the WF dynamic model is first linearized, thus allowing classic results of linear control theory to be applied. Generally, consensus techniques are also employed in order to let satisfaction of the maximum power production. One can cite for instance [Madjidian 2011, Biegel 2013, Zhang 2013].

4.1.3 Contribution

In our work we focus on the problem of controlling wind farms in which the wake interaction among WTs cannot be neglected, and it is taken into account in the problem formulation. Both the centralized and the distributed hierarchical control architectures are analyzed. However, our main first contribution is concerned with the distributed WF optimization for power maximization under the system physical constraints, and maximum allowed power one imposed by the grid operator. In particular, we suggest to apply the two novel distributed versions of the PSO algorithm, proposed in the previous chapter, as they allow the optimization problem to be formulated by directly considering the power constraints. Being the optimization problem nonconvex, we chose the PSO technique, as it belongs to the global optimization methods class. The first proposed algorithms that we aim to test is DPSO1, enabling a distributed architecture to be implemented by means of PSO combined with available techniques of finite-time consensus. Here, recall that the basic structure of classic PSO remains unchanged, while distribution among a set of agents, i.e. the wind turbines, is allowed by a proper re-organization of the system variables, and the choice of a particular communication graph. Our second proposed distributed PSO algorithm, i.e. DPSO2, enables less communication burden with respect to DPSO1. Recall that, differently from DPSO1, this is achieved by introducing a new algorithm modification, which changes the classic PSO structure. Such modification has a remarkable impact on the convergence time of the algorithm to a solution, opening up to a more practical implementation in real plants. Both DPSO1 and DPSO2 are model-based, as we seek for fast convergence of the algorithm, for real time implementation. Nonetheless, they are not restricted to WF power models of specific form, as it happens for gradient-based optimization algorithms, which need the WF power function to be continuous and differentiable with respect to the system variables. This feature is inherited from metaheuristic algorithms, which only need the WF power function to be *evaluated* in selected search space points. The results of DPSO2 applied for the WF optimization are object of a patent deposited by EDF (Électricité de France) in July 2017, and entitled *Pilotage d'un parc éolien* ('Control of a wind farm').

As stated by [Marden 2013], the optimization algorithms mentioned in Section 4.1.2, generally rely on the assumption of the existence of local control strategies for individual wind turbines that can stabilize around any feasible optimal set point, solution of the wind farm optimization problem. Even if proving a potential benefit in the amount of extracted wind power, these assumptions are not necessarily realistic as either the dynamics of the optimization variables and the performance

of the local controllers play an important role in the actual gain. Thus, as a second contribution, we analyze the overall control architecture performance, i.e. we consider the problem of maximizing a wind farm power production, based on a static optimization at high level for optimal set points generation and local control at low level to stabilize the wind turbines around them. Nonetheless, in this second framework, we mainly focus on the local controller performance to demonstrate what can actually be achieved by means of such hierarchical architecture. The local control employed in this approach is the one developed in Chapter 2. As a consequence, this analysis enables us to create the first important connection between the presented tools in Part I and the purpose of controlling a wind farm. In the literature, few works have analyzed the effective gain of wind farm optimization under wake effect when the dynamics of the controlled turbines are considered. An example is given in [Heer 2014], where a local controller based on system linear approximation is employed, and it shows $\sim 1\%$ energy gain with respect to classic greed control, where farm optimization would not be performed. The central aspect that motivated this work is that, when cooperative optimization is employed, the optimal set points delivered to each turbine in the wind farm can deviate from the classic power references typically used in greed control. This confirms the importance of the deloading WT control techniques developed in Chapter 2. The described system analysis has been object of our work of [Gionfra 2016a].

4.2 Wake Model

In this section, we aim at providing a model of the wind farm power function. As it has been mentioned in the previous section, in many cases, the WTs of a wind farm are aerodynamically coupled via the previously introduced wake effect. Thus, a proper modeling of such effect is essential for the description of the available power extraction at the WF level. The wake model is generally complex, and involves solving partial differential equations via CFD tools. If the WF power function though is to be used in real-time for WF optimization and control, i.e. for model-based techniques, then a model simplification is needed to let practical implementation feasible. As it will be shown in the sequel, typically, a new optimization needs to be run each time that wind conditions change. Thus, an important feature of the employed optimization algorithm, and, as far as this work is concerned, a model-based one, is fast convergence.

Among the available simplified analytic wake models, as mentioned, the Park one is by far the most employed. In the sequel we present the wake model introduced in the works of [Gebraad 2016, Park 2015a]. This is still based on the some basic Park model assumptions and formulations, but it additionally takes into account the WTs yaw angle influence in the wake profile. Moreover, while Park wake model results in a nonsmooth power function, the model here considered is continuous and differentiable, and it is useful if gradient-based optimization algorithms are to be

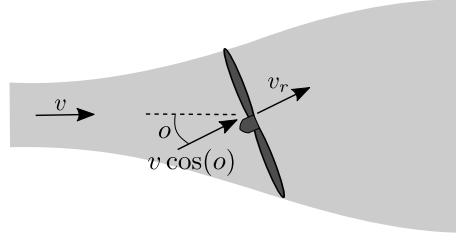


Figure 4.1: Wind flow and wind speed at rotor disc, $v_r = v(\cos o - \alpha)$, [Park 2015a].

employed. This is obtained by considering a particular mathematical model characterizing the wake shape, which, in addition, allows a better degree of model fidelity with respect to Park one, without introducing much model complexification that may result in higher computational burden. Eventually, when applied to a given WF, such model is tuned with CFD simulation data, by acting on some parameters left as degree of freedom. For instance in [Park 2015a], it shows a high degree of fidelity when calibrated on the *Horns Rev 1* wind farm located in Denmark.

4.2.1 Modified Park Wake Model

Recall from Section 2.2 of Chapter 2, that for a wind blowing in the *axial* direction of a WT, the available extractable power can be expressed as a function of α , the axial induction factor¹. This is done by providing the theoretical value of the WT power coefficient C_p , as the function expressed in (2.3), and that we report here for the convenience of the reader

$$C_p(\alpha) = 4\alpha(1 - \alpha)^2$$

This function additionally enables to compute the theoretical α value for which C_p attains its maximum, i.e. $1/3$, and that we previously named α_{betz} . However, when the WT rotor plane is yawed with respect to the wind direction by a yaw offset angle o , as shown in Fig. 4.1, then the expression of α is given by $\alpha = \frac{v \cos o - v_r}{v}$, as the wind component in the axial direction is $v \cos o$. The power extracted by a WT from the wind can be expressed as

$$P = \frac{1}{2} \rho \pi R^2 (v \cos o)^3 4\tilde{\alpha}(1 - \tilde{\alpha})^2 \quad (4.1)$$

where $\tilde{\alpha} = \frac{v \cos o - v_r}{v \cos o}$ is the axial induction factor as if the wind blew in the axial direction with a speed value equal to $v \cos o$. Thus $\tilde{\alpha} = \alpha / \cos o$, and by replacing it in (4.1), it yields

$$P = \frac{1}{2} \rho \pi R^2 v^3 4\alpha(\cos o - \alpha)^2$$

¹Recall that α is defined as $\alpha \triangleq \frac{v - v_r}{v}$, i.e. as the ratio of the difference between the wind speed value in the axial direction and the wind speed at the rotor plane, and the upstream value, (see Fig. 2.3).

from which it is easy to identify the power coefficient C_p in this yawed case as a function of both α , and o , i.e. $C_p(\alpha, o) = 4\alpha(\cos o - \alpha)^2$. As it has been mentioned, the model described in this section presents some tunable parameters. Thus, the power coefficient expression is modified by introducing a first parameter $\tau_1 \in \mathbb{R}$, yielding

$$C_p(\alpha, o) = 4\alpha(\cos(\tau_1 o) - \alpha)^2$$

Park wake model describes the wind speed value in a downstream wake, by introducing a deficit factor $\delta_v(d, r, \alpha)$, i.e. function of the distance d behind the wake, the radial wake distance r , and the axial induction factor α of the WT causing the wake. Such wind speed value is computed according to

$$v(d, r, \alpha) = v(1 - \delta_v(d, r, \alpha))$$

In such model, it is assumed that the radius of the wake linearly increases with the downstream distance, according to $R(d) = R + \kappa_w d$, where R is the radius of the WT rotor plane, and κ_w the wake expansion, mentioned in Section 4.1, which depends on the surface roughness of the WF site. Deficit δ_v is then assumed to be null outside the wake, constant inside on equal distance values d , and proportional to the rapport between the area of the rotor plane, πR^2 , and the wake plane area at d , $\pi R^2(d)$, having value $2\alpha \left(\frac{R}{R(d)}\right)^2$. This is where Park model introduces a discontinuity. However, it has been observed that for distance values $d > 10R$, the cross sectional wind speed profile resembles a Gaussian function [Park 2015a]. Thus, $\delta_v(d, r, \alpha)$ can be modeled as

$$\delta_v(d, r, \alpha) = 2\alpha \left(\frac{R}{R(d)}\right)^2 e^{-\left(\frac{r}{R(d)}\right)^2} \quad (4.2)$$

The aforementioned considerations on the wake expansion, and shape behind the WT are depicted in Fig. 4.2. Eventually, in order to take into account the yaw effect, (4.2) can be further modified as, more in general, when $o \neq 0$, the area swept by the WT blades assume an ellipsoidal form whose surface is given by $\pi R^2 \cos o$. By introducing a second parameter $\tau_2 \in \mathbb{R}$ to let model tuning, (4.2) takes the generalized expression

$$\delta_v(d, r, \alpha, o) = 2\alpha \cos(\tau_2 o) \left(\frac{R}{R(d)}\right)^2 e^{-\left(\frac{r}{R(d)}\right)^2} \quad (4.3)$$

4.2.2 Single Wake Model

Having introduced the wind deficit factor in the downstream wake of a given WT j , we now need to quantify the wind speed reduction experienced by a downstream WT i . Intuitively, this depends on the wake position with respect to the downstream WT rotor plane. This can be modeled by introducing two parameters, namely d_{ij} ,

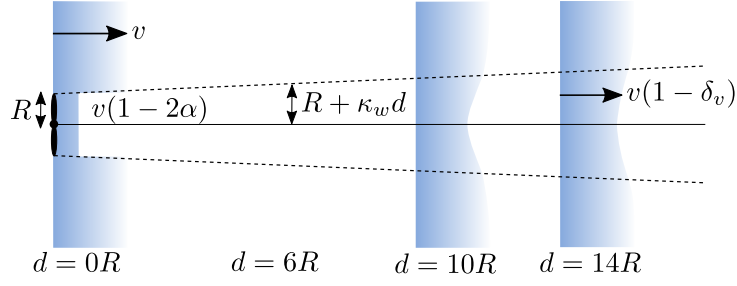


Figure 4.2: Modified Park wake expansion model, [Park 2015a].

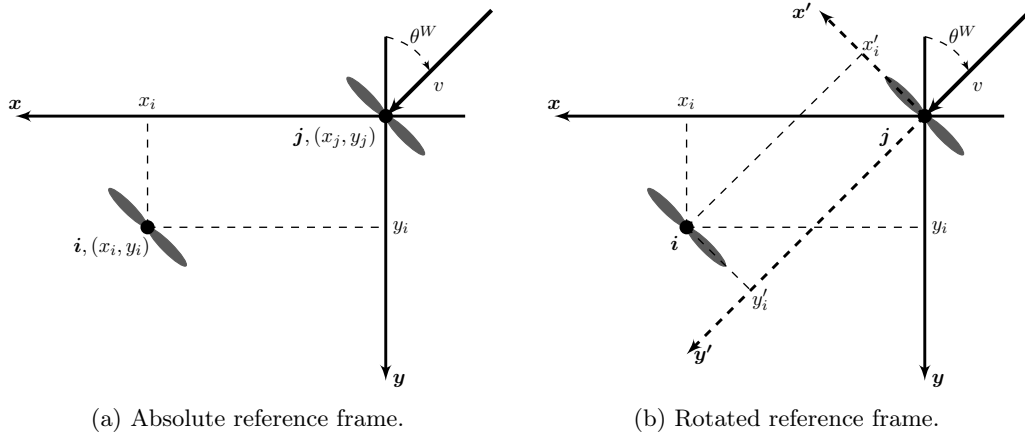


Figure 4.3: Wind farm reference frame.

and r_{ij} . The former represents the axial distance according to the free stream wind direction θ^W , while the latter represents the radial distance of the wake centerline from WT i rotor plane center. Moreover, in order to take into account the effect of the rotating blades and the yaw effect on the wake trajectory, r_{ij} is composed of three terms, i.e.

$$r_{ij} = r_{ij}^l + r_{ij}^r + r_{ij}^o \quad (4.4)$$

Both d_{ij} , and r_{ij} are described in the sequel. Let us consider an *absolute* reference frame (x, y) . In the following we will always make reference to the axis disposition and orientation shown in Fig. 4.3a. We can thus define the position of each turbine i via its coordinates (x_i, y_i) . The wind direction θ^W in front of the considered upstream WT j can be defined with respect to the mentioned reference frame. Then we can consider a *rotated* reference (x', y') by an angle equal to θ^W , as shown in Fig. 4.3b, and given by the coordinate transformation

$$\begin{cases} y' &= x \sin \theta^W + y \cos \theta^W \\ x' &= x \cos \theta^W - y \sin \theta^W \end{cases} \quad (4.5)$$

On this basis, we can compute the distance d_{ij} , and r_{ij} , illustrated in Fig. 4.4a, as

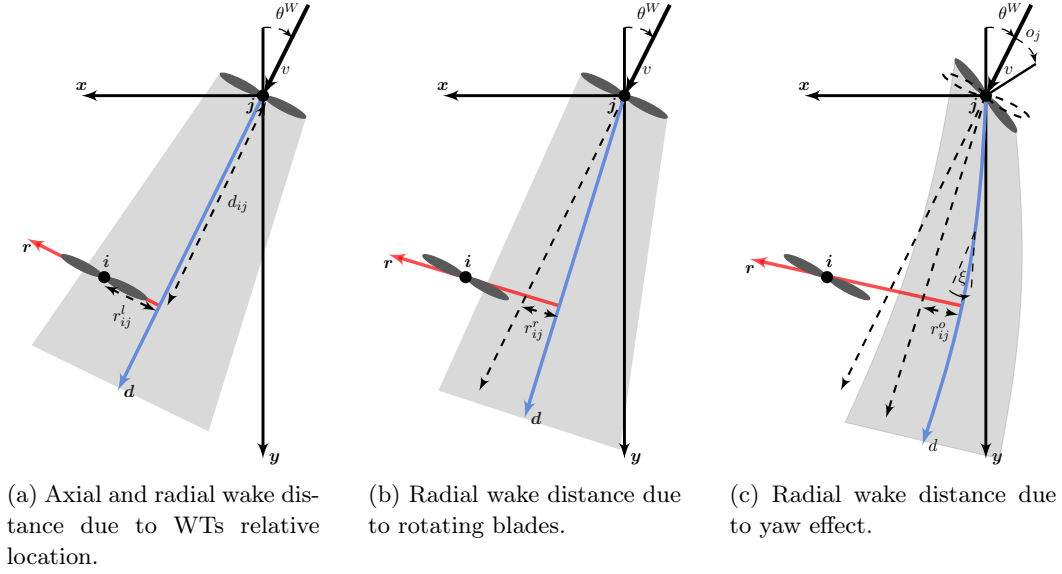


Figure 4.4: Axial and radial wake distance, [Park 2015a].

$$d_{ij} = |y'_i - y'_j|$$

$$r_{ij}^l = |x'_i - x'_j|$$

where r_{ij}^l is the first term in (4.4), and it represents the radial wake distance due to the relative locations of WT i , and j with respect to θ^W , i.e. in the rotated coordinates. The second term of r_{ij} , i.e. r_{ij}^r , allows to take into account the wake deviation caused by the rotating blades. In [Gebraad 2016], it is shown that this term depends linearly on d_{ij} via a tunable parameter $\tau_3 \in \mathbb{R}^+$. Moreover, without loss of generality, we consider a positive deflection towards the right of the wake centerline, i.e.

$$r_{ij}^r = \tau_3 d_{ij} \text{sign}(x'_i - x'_j)$$

where $\text{sign}(x'_i - x'_j) = 1$ if $x'_i - x'_j \geq 0$, and -1 otherwise. This second term contribution is illustrated in Fig. 4.4b. The last term in r_{ij} , r_{ij}^o , is due to the yaw offset angle of WT j , o_j . The wake is deflected by the yaw, then, influenced by the free stream, it deviates in its direction. The result is a *curved* trajectory, as shown in Fig. 4.4c. Authors of [Jiménez 2010] describe this curvature at a generic downstream distance d via the angle of the centerline of the wake $\xi(d)$, which assumes the following expression [Gebraad 2016]

$$\xi(d) = \frac{\xi_{init}(\alpha_j, o_j)}{\left(1 + \frac{\tau_4 d}{R}\right)^2}$$

where $\tau_4 \in \mathbb{R}^+$ is left as an additional tunable parameter, and

$$\xi_{init}(\alpha_j, o_j) \triangleq \frac{1}{2}(\cos o_j)^2(\sin o_j)4\alpha_j(1 - \alpha_j)$$

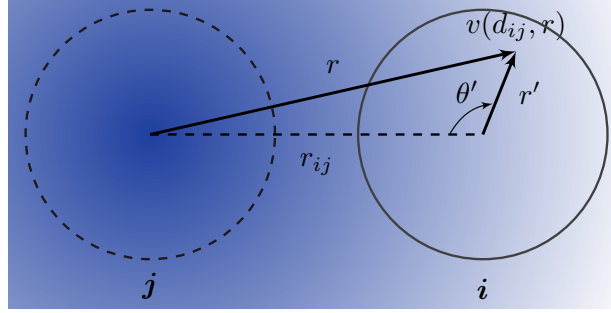


Figure 4.5: Local polar coordinates, [Park 2015a].

Moreover, the yaw-induced lateral offset $\delta_\gamma(d_{ij})$ can be approximated by Taylor expansion to solve the following integral

$$\begin{aligned} \delta_\gamma(d_{ij}) &= \int_0^{d_{ij}} \tan(\xi(x)) dx \\ &\simeq \frac{\xi_{init}(\alpha_j, o_j) R}{15\tau_4} \left(\frac{15 \left(\frac{\tau_4 d_{ij}}{R} + 1 \right)^4 + \xi_{init}(\alpha_j, o_j)^2}{\left(\frac{\tau_4 d_{ij}}{R} + 1 \right)^5} - 15 - \xi_{init}(\alpha_j, o_j)^2 \right) \end{aligned}$$

As shown in Fig. 4.4c, a positive yaw offset o_j increases the yaw-induced offset. However, this latter can increase or decrease the total value of r_{ij} depending on the relative locations of the wind turbines and the wind direction. According to the above considerations, r_{ij}^o can be computed as

$$r_{ij}^o = |\delta_\gamma(d_{ij})| \text{sign}(x'_i - x'_j)$$

We are now ready to determine the wind speed deficit of a WT i caused by the single wake of WT j . Indeed, using the expressions of the wind deficit in (4.3), d_{ij} , and r_{ij} , the wind speed profile can be computed using local polar coordinates (r', θ') as shown in Fig. 4.5. Thus, we have that the wind speed value captured by WT i at the generic position (r', θ') is

$$v_{ij}(r', \theta', \alpha_j, o_j, v, \theta^W) = v(1 - \delta_v(d_{ij}, r, \alpha_j, o_j))$$

where δ_v is computed in (d_{ij}, α_j, o_j) , and where $r = \sqrt{(r_{ij} - r' \cos \theta')^2 + (r' \sin \theta')^2}$. Notice that d_{ij} , and r , on which δ_v depends, are function of (α_j, o_j) , and θ^W , but not on v . For the purpose of simplification we can now derive the *average* wind speed \bar{v}_{ij} on the rotor disc of the downstream WT i . By applying the mean value theorem for integrals, this is given by

$$\bar{v}_{ij}(\alpha_j, o_j, v, \theta^W) = \frac{1}{\pi R^2} \int_{\theta'=0}^{\theta'=2\pi} \int_{r'=0}^{r'=R} v_{ij}(r', \theta', \alpha_j, o_j, v, \theta^W) r' dr' d\theta'$$

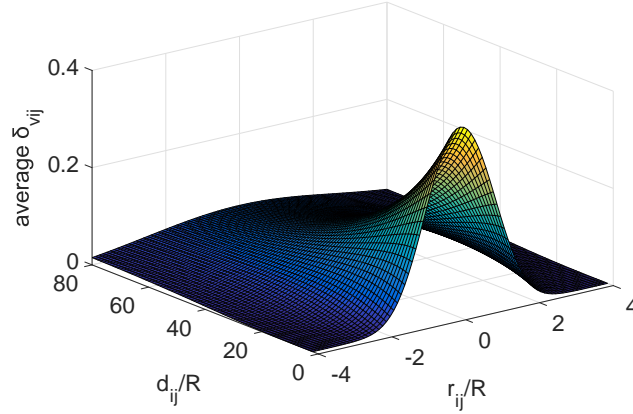


Figure 4.6: Wind deficit as a function of d_{ij} and r_{ij} caused by WT j on WT i , when $o_j = 0$, $\alpha = \alpha_{betz}$.

From the above expression, the power extracted from the downstream WT is

$$P_i = \frac{1}{2} \rho \pi R^2 \bar{v}_{ij}^3(\alpha_j, o_j, v, \theta^W) C_p(\alpha_i, o_i) \quad (4.6)$$

We are able to see how P_i is function of the according WT i operating conditions via C_p , as well as function of the upstream WT j ones via the wake model, defining $\bar{v}_{ij}(\alpha_j, o_j, v, \theta^W)$. The model parameters tuned by CFD simulation on the Horns Rev 1 wind farm, whose WTs have $R = 63$ m, provided by [Park 2015a], are given in Table 4.1. An example of average wind deficit $\bar{\delta}_{v,ij} \triangleq 1 - \frac{\bar{v}_{ij}}{v}$ as a function of d_{ij} , and r_{ij} , caused by a WT j with $o_j = 0$, $\alpha_j = \alpha_{betz}$, is given in Fig. 4.6, where the model parameters of the aforementioned table were used.

Table 4.1: Horns Rev 1 wind farm parameters

κ_w	τ_1	τ_2	τ_3	τ_4
0.0313	0.7850	1.5410	0.0211	0.5617

4.2.3 Wind Farm Power Function

In a wind farm, a wind turbine i is likely to experience a wind speed deficit caused by *multiple* wakes, i.e. from all the upstream WTs. In order to take into account the wake interference among multiple WTs, the kinetic energy conservation method proposed by [Katic 1986] in Park model, is by far the most employed one. In this model it is assumed that *the kinetic energy deficit by the mixed wake is equal to the sum of the kinetic energy deficits by individual wakes*, [Park 2015a]. Before providing the expression of the wind deficit in the multi-wake case, it is useful to employ the definition of physical neighborhood given in the previous chapter. Each WT i is physically coupled to any upstream WT j via the wake model, since the average

captured wind speed \bar{v}_{ij} , affecting turbine i power function via (4.6), is function of WT j operating points, i.e. α_j , and o_j . Thus, WT j is a physical neighbor of WT i . Notice that, differently from the OPF case of Section 3.6 in Chapter 3, here the vice versa is not true. For each WT i we are thus able to define the associated physical neighborhood as $\mathcal{N}_i^p \triangleq \{j : y'_i - y'_j > 0\}$, i.e. if WT j is an upstream turbine to i . In the same way of what done in the previous chapter, we can also define the associated variables $\alpha_{ij} \triangleq \{\alpha_j : j \in \mathcal{N}_i^p\}$, and $\mathbf{o}_{ij} \triangleq \{o_j : j \in \mathcal{N}_i^p\}$. Notice that $\mathcal{N}_i^p(\theta^W)$, i.e. it is a function of the wind farm free stream wind direction θ^W via (4.5). Thus, if θ^W is a function of time, then \mathcal{N}_i^p is time-varying too. According to the mentioned method of conservation of kinetic energy, we can thus describe the total average wind speed deficit $\bar{\delta}_{v,i}$ experienced by WT i in a WF as

$$\bar{\delta}_{v,i} = \sqrt{\sum_{j \in \mathcal{N}_i^p} \bar{\delta}_{v,ij}^2}$$

If we now name v_∞ the free stream wind speed blowing towards the wind farm, we have that the average wind speed captured by WT i is

$$\bar{v}_i(\alpha_{ij}, \mathbf{o}_{ij}, v_\infty, \theta^W) = v_\infty(1 - \bar{\delta}_{v,i}(\alpha_{ij}, \mathbf{o}_{ij}, \theta^W))$$

As a result, its available power is

$$P_i(\alpha_i, o_i, \alpha_{ij}, \mathbf{o}_{ij}, v_\infty, \theta^W) = \frac{1}{2} \rho \pi R^2 \bar{v}_i^3(\alpha_{ij}, \mathbf{o}_{ij}, v_\infty, \theta^W) C_p(\alpha_i, o_i)$$

Eventually, the total wind farm available power P_{wf} , given the free stream wind parameters (v_∞, θ^W) , and the WTs operating conditions $\alpha_i, o_i, i = 1, \dots, N$, where N is the number of WTs in the WF, can be simply computed by summing the single power productions, yielding

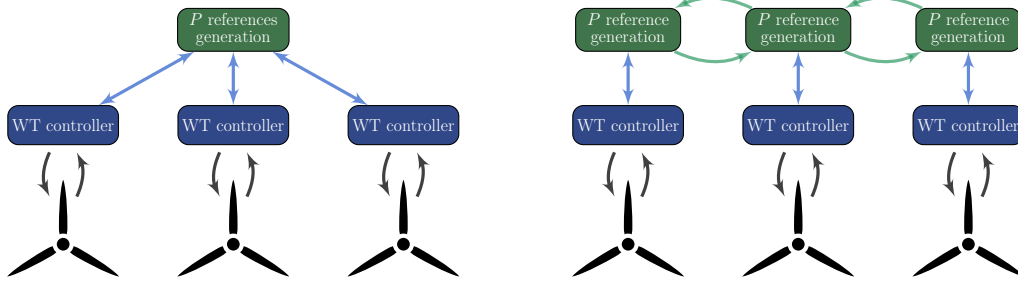
$$P_{wf} \triangleq \sum_{i=1}^N P_i(\alpha_i, o_i, \alpha_{ij}, \mathbf{o}_{ij}, v_\infty, \theta^W) \quad (4.7)$$

Remark 4.1 Notice that, according to the considered wake model, the wind deficit experienced by each turbine is function of the free stream wind direction θ^W , but not of its speed value v_∞ . This influences the power production, but it intervenes as a factor in (4.7). This fact has important consequences in the optimization problem formulation, as it will be discussed in the next sections.

4.3 Hierarchical Control

4.3.1 High Level Optimization Problem

As mentioned in Section 4.1, we focus on a hierarchical structure for the sake of controlling a wind farm. This is developed on a two-level pyramidal structure,



(a) Centralized hierarchical architecture.

(b) Distributed hierarchical architecture.

Figure 4.7: Hierarchical architectures.

whose low control level coincides with the local WT control, as the one developed in Chapter 2. Thus, it is responsible for making the WT tracking the power reference signal computed by the higher control level. In this work, according to the considerations done in Section 4.1, the higher control level reduces to an optimization step. The associated optimization problem is built upon the wind farm model shown in Section 4.2, which is *static*. In other words, we do not consider the *dynamics* ruling the wind field in a WF. Wind speed values can be thus considered as *steady state* ones. The hierarchical structure can be centralized or distributed according to how the optimization problem is solved. In the former case, a central controller communicates with the WTs ones to get information about the wind conditions and to deliver optimized power references. Typically such communication architecture is realized via SCADA systems. The optimization is performed by a single central controller, based on WF measurements and on additional power constraints, which for the sake of simplicity, can be considered as imposed by the grid operator. Such system suffers from the main drawbacks listed in Section 4.1, and this motivates the interest for distributed hierarchical architectures. In this case, the optimization step is executed cooperatively by high level controllers associated to each WT and that are able to exchange information among them. Two examples of communication will be discussed in Section 4.4. A schematic illustrating the considered architectures is given in Fig. 4.7.

Let us now consider the optimization problem in details. Given a WF of N WTs, this usually takes the form of an *unconstrained* wind farm power maximization, which can be formulated as follows

$$\min_{(\boldsymbol{\alpha}, \boldsymbol{o}) \triangleq \{(\alpha_i, o_i), i=1, \dots, N\}} -P_{wf}(\boldsymbol{\alpha}, \boldsymbol{o}, v_\infty, \theta^W) \quad (4.8)$$

Some considerations need to be done concerning problem (4.8). First of all, notice that according to Remark 4.1, it follows that its argument $(\boldsymbol{\alpha}^*, \boldsymbol{o}^*)$, i.e. the optimal optimization variables, *does not* change if only the wind speed value v_∞ does. This is due to the fact that (4.8) cost function has invariant minima with respect to

v_∞ , as it appears as a factor in it. The important practical implication is that, in this case, the optimization needs to be performed only when the wind direction θ^W changes.

In a real world implementation though, the solution to (4.8) may not be feasible due to system constraints. We identify two main sources of them, which will be considered in the sequel, namely

- physical constraints, i.e. WT nominal power, $P_{e,n}$.
- grid constraints, expressed in the form of maximum allowed power injection in the grid, P_{wf}^{max} .

As a result, from a practical point of view, the optimal solution to (4.8) can be applied to the system only if, in the absence of a P_{wf}^{max} constraint, the free stream wind speed value v_∞ is such that $P_i(\alpha^*, \theta^*) \leq P_{e,n}$, $i = 1, \dots, N$. Another approach that could be considered is to constrain the solution of (4.8) *a posteriori*. However, in this case, being the optimization problem nonconvex, and because the considered power constraints generally define a nonconvex feasible set, such solution is likely to be suboptimal. For this reason, we rather consider such constraints actively in the optimization problem, by modifying (4.8) as follows

$$\begin{aligned} & \min_{(\alpha, \theta)} -P_{wf}(\alpha, \theta, v_\infty, \theta^W) \\ & \text{subject to} \\ & P_i(\alpha, \theta) \leq P_{e,n}, \quad i = 1, \dots, N \\ & P_{wf}(\alpha, \theta, v_\infty, \theta^W) \leq P_{wf}^{max} \end{aligned} \tag{4.9}$$

It is important to notice that, differently from (4.8), problem (4.9) has to be solved each time that either v_∞ , or θ^W changes, as v_∞ modifies the feasible region via the problem constraints. It is clear that under this requirement, the optimization step solving (4.9) has to be faster than the variation of the wind conditions. Moreover the system has to have the time to actualize such solution via the local controllers. The above considerations can be summarized in the following

Assumption 4.1 *The couple (v_∞, θ^W) varies slowly with respect to the convergence time of the optimization algorithm and the dynamics of the controlled WTs.*

Under this assumption, it seems reasonable to consider an optimization step with respect to averaged values of wind speed and direction signals, in order to grasp their main trends affecting the wind farm power function, and to sufficiently filter out the turbulence components. For this reason, for now on, we will refer to (v_∞, θ^W) as the filtered values of the original wind speed and direction signals. For instance, from wind measurements, this can be achieved via a moving average filter. Eventually we consider the additional

Assumption 4.2 *The couple (v_∞, θ^W) is uniform along the wind farm length.*

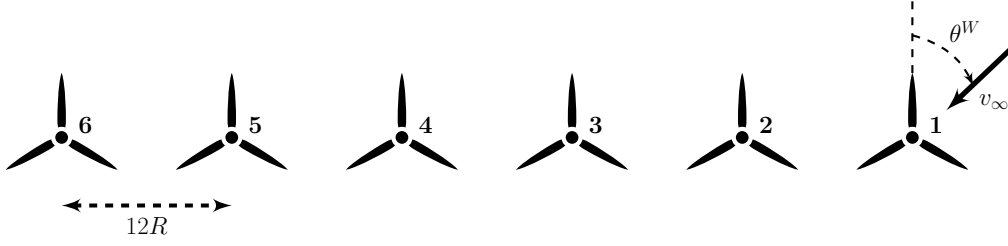


Figure 4.8: 6-WT wind farm example for yaw effect analysis.

4.3.2 Yaw Angle Influence on the Optimization Problem

As shown in the wake model of Section 4.2, the yaw angle influences the wake shape trajectory, and for this reason it should be considered as an optimization variable, as done in the optimization problem formulation in (4.9). However, a suboptimal yet simpler choice for the WTs yaw angles of a WF could be considered if it simplifies and enhances the convergence of the optimization algorithm. Intuitively such choice is $\gamma_i = 0$, $i = 1, \dots, N$, i.e. the yaw angle γ_i of each WT is $\gamma_i = \theta^W$. In this case the WTs are always oriented such that their rotor plane is perpendicular to the wind direction. This is why in this subsection we aim at analyzing the sensitivity of the solution to (4.9) with respect to the yaw angle, by comparing it with the mentioned case of $\gamma_i = 0$, $i = 1, \dots, N$. For such purpose we consider a modified problem (4.9), in which we add the penalizing term $\beta_o \sum_{i=1}^N (\gamma_i - \theta^W)^2$ to its cost function, where $\beta_o \in \mathbb{R}_0^+$ is a tunable parameter. The idea is to compare the problem optimal solutions for the cases $\beta_o = 0$, and $\beta_o > 0$, i.e. respectively corresponding to the original problem, and to the case in which γ_i are pushed to be equal to θ^W . Moreover, in this analysis, we employ a classic centralized PSO with Deb's rule to solve the described modified problem. In view of using the distributed PSO versions previously proposed in order to solve the WF optimization problem, this additionally allows us to anticipate the behavior of such metaheuristic algorithms for this specific problem. The test is performed on a 6-WT wind farm example shown in Fig. 4.8. The simulations are carried out for $\theta^W = 0$, and $\theta^W = \pi/2$, i.e. respectively the negligible wake effect case, and the worst wake effect one, and for both low and high wind speed values v_∞ , to test the problem solution in respectively the active and inactive constraint case. As far as the constraints are concerned, only the physical ones are considered, i.e. a maximum allowed power value for each WT equal to $P_{e,n} = 5 \text{ MW}$, corresponding to WTs whose rotor radius² is $R = 63 \text{ m}$. Concerning the algorithm parameters, we consider 500 as maximum iteration, and for this particular problem setup we additionally modify the inequality constraints $P_i \leq P_{e,n}$, $i = 1, \dots, N$ by introducing a linearly iteration-decreasing positive threshold ϵ such that $P_i \leq P_{e,n} + \epsilon$, $i = 1, \dots, N$, in order to enhance the quality of the algorithm solution. Notice that this technique is usually

²Note that this values belong to the WTs of the Horns Rev 1 wind farm.

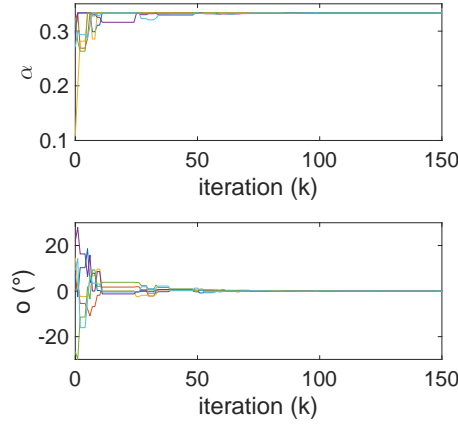


Figure 4.9: Global best trajectories for $\theta^W = 0$ and low wind speed, and for both $\beta_o = 0$, and $\beta_o > 0$: convergence to p_1^* .

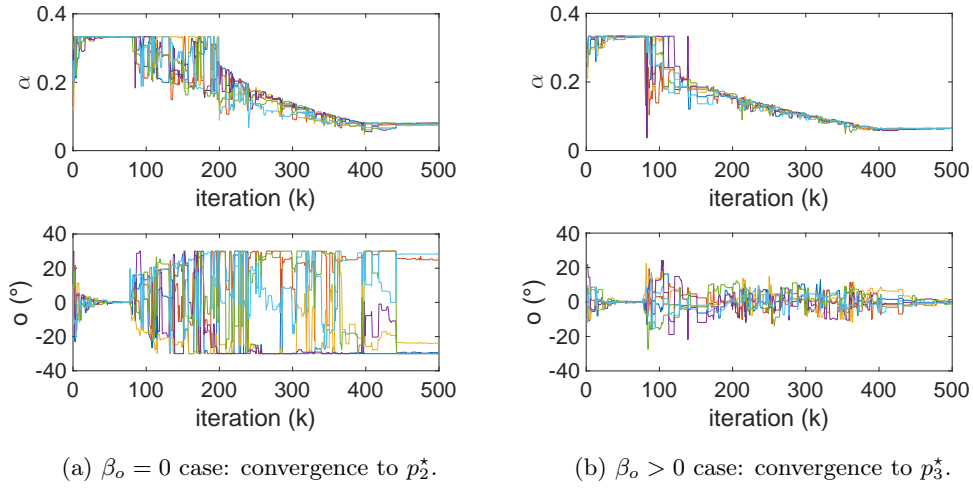


Figure 4.10: Global best trajectories for $\theta^W = 0$ and high wind speed.

employed to manage the equality constraints (see Section 3.2 of Chapter 3). In the sequel we only report the global best trajectories associated to the six couples of variables (α, o) , during the run of the PSO algorithm. The use of ϵ explains their particular behavior shown in the following figures.

Let us start our test for the case $\theta = 0$, i.e. negligible wake effect. At low wind speed, i.e. $v_\infty = 7$ m/s in this example, the algorithm converges to the same optimal solution $p_1^* \triangleq (\alpha_1^*, o_1^*)$ for both the cases $\beta_o = 0$, and $\beta_o > 0$, to which it corresponds a total WF power $P_{wf,1}^* = 9.81$ MW. The corresponding global best trajectories are shown in Fig. 4.9. As expected, in order to maximize the power extraction, $o_1^* = 0$, and $\alpha_1^* = 1\alpha_{betz}$. At high wind speed, i.e. $v_\infty = 14$ m/s in this example, simulations are shown in Fig. 4.10a, and Fig. 4.10b for respectively $\beta_o = 0$, and $\beta_o > 0$, where

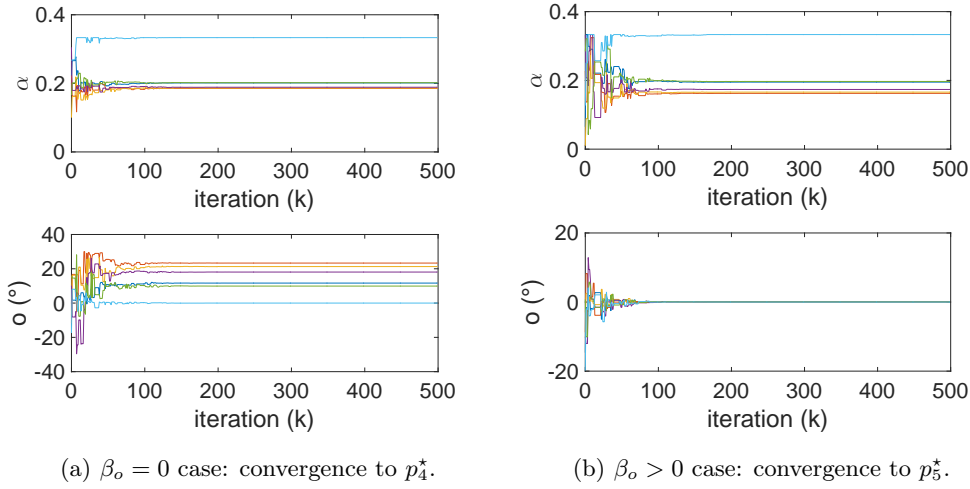


Figure 4.11: Global best trajectories for $\theta^W = \pi/2$ and low wind speed.

the algorithm converges respectively to $p_2^* \triangleq (\alpha_2^*, \mathbf{o}_2^*)$, and $p_3^* \triangleq (\alpha_3^*, \mathbf{o}_3^*)$, and where $\mathbf{o}_3^* \simeq \mathbf{0}$. It has to be stressed that in the case of active constraints, there exist infinite feasible couples (α, \mathbf{o}) that let their satisfaction. This has an important effect in the algorithm convergence as it can be seen in Fig. 4.10a. Indeed here, the algorithm is not able to *choose* a specific couple among the feasible ones, and this results in the oscillatory behavior of the global best trajectories. Moreover, this behavior seems to particularly affect the yaw global best ones, and this is indicative of how the problem solution is less sensitive to the choice of yaw values as long as they stay within a bounded interval around θ^W . In this test, for instance, we chose the interval $-\pi/6 \leq o \leq \pi/6$. This is further motivated if we compare p_2^* , and p_3^* . As a matter of fact, $\alpha_2^* \simeq \alpha_3^*$, while \mathbf{o}_2^* , and \mathbf{o}_3^* are quite different. Nonetheless, their associated WF power values, $P_{wf,2}^* = 29.85 \text{ MW}$, and $P_{wf,3}^* = 29.90 \text{ MW}$ are similar. Again, the solution appears to be more sensitive to the choice of α variables.

Let us now consider the case of $\theta^W = \pi/2$, i.e. maximum wake effect. At low wind speed, the optimal solutions are $p_4^* \triangleq (\alpha_4^*, \mathbf{o}_4^*)$, and $p_5^* \triangleq (\alpha_5^*, \mathbf{o}_5^*)$, for respectively $\beta_o = 0$, and $\beta_o > 0$, where $\mathbf{o}_5^* \simeq \mathbf{0}$. Simulations are reported in Fig. 4.11. The WF power obtained via p_4^* is $P_{wf,4}^* = 5.30 \text{ MW}$, while it is $P_{wf,5}^* = 5.25 \text{ MW}$ for p_5^* . From these results, it is clear that when constraints are not active and wake effect is not negligible, the yaw angles do have a role in determining the optimal solution. Thus, in this case, pre-assigning the value $\gamma_i = \theta^W$, $i = 1, \dots, N$ results in a suboptimal solution. Nonetheless, the difference between $P_{wf,4}^*$ and $P_{wf,5}^*$ is not large, and this confirms again the less sensitivity of the solution with respect to the yaw angles. Eventually, similar results to the case of $\theta^W = 0$ are obtained for high wind speed values. These are illustrated in Fig. 4.12. Indeed, if among the feasible solutions, we select the one having yaw values $\gamma_i = \theta^W$, $i = 1, \dots, N$, the quality of the solution stays unchanged, yielding for the $\beta_o = 0$ case $p_6^* \triangleq (\alpha_6^*, \mathbf{o}_6^*)$, and corresponding

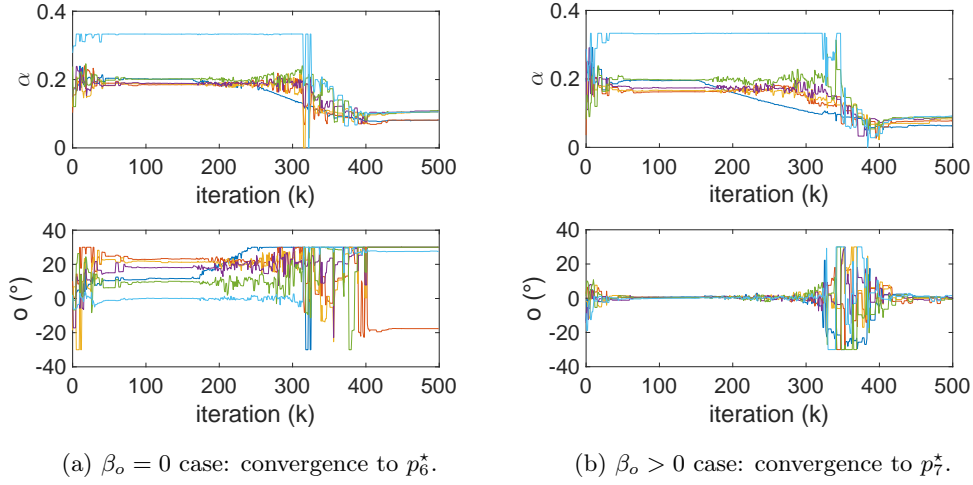


Figure 4.12: Global best trajectories for $\theta^W = \pi/2$ and high wind speed.

$P_{wf,6}^* = 29.92 \text{ MW}$, and for the $\beta_o > 0$ case $p_7^* \triangleq (\alpha_7^*, o_7^*)$, and corresponding $P_{wf,7}^* = 29.70 \text{ MW}$. The overall test results are collected in Table 4.2. According to the simulations and remarks provided above, we can conclude the following. As long as it stays within a given bounded interval around the θ^W value, e.g. $[-\pi/6, \pi/6]$, the yaw angle has generally little importance in determining the optimal solution of the optimization problem, when constraints are active. Moreover, considering it as an optimization variable may lead to unclear algorithm convergence when PSO-like algorithms are employed. As expected, the yaw angle has an active role in providing the optimal solution when constraints are not active and in the presence of the wake effect. However, the optimality of the solution seems to be poorly sensitive to its choice, with respect to the case in which $\gamma_i = \theta^W$, $i = 1, \dots, N$. In the light of the above considerations, we make a choice that can be stated through the following

Assumption 4.3 *Given the free stream wind direction θ^W , the WTs yaw angles are set as $\gamma_i = \theta^W$, $i = 1, \dots, N$.*

As a results, optimization problem (4.9) reduces to

$$\begin{aligned}
 & \min_{\alpha} -P_{wf}(\alpha, \mathbf{0}, v_{\infty}, \theta^W) \\
 & \text{subject to} \\
 & P_i(\alpha, \mathbf{0}) \leq P_{e,n}, \quad i = 1, \dots, N \\
 & P_{wf}(\alpha, \mathbf{0}, v_{\infty}, \theta^W) \leq P_{wf}^{max}
 \end{aligned} \tag{4.10}$$

where we set $\mathbf{o} = \mathbf{0}$.

Remark 4.2 *Under Assumption 4.3, the radial wake distance term r_{ij}^o , described in Section 4.2, is 0.*

Table 4.2: Yaw Influence Test Results

		v_∞	
		7 m/s	14 m/s
		inactive constraints	active constraints
β_o	0	<ul style="list-style-type: none"> • p_1^* • $P_{wf,1}^* = 9.81 \text{ MW}$ 	<ul style="list-style-type: none"> • constraints satisfied ✓ • $\alpha_2^* \simeq \alpha_3^*$ • σ_2^*: unclear convergence • $P_{wf,2}^* = 29.85 \text{ MW}$
	> 0	<ul style="list-style-type: none"> • p_1^* • $P_{wf,1}^* = 9.81 \text{ MW}$ 	<ul style="list-style-type: none"> • constraints satisfied ✓ • $\alpha_3^* \simeq \alpha_2^*$ • $\sigma_3^* \simeq 0$ • $P_{wf,3}^* = 29.90 \text{ MW}$

(a) $\theta^W = 0$, negligible wake effect.

		v_∞	
		7 m/s	14 m/s
		inactive constraints	active constraints
β_o	0	<ul style="list-style-type: none"> • p_4^* • $P_{wf,4}^* = 5.30 \text{ MW}$ 	<ul style="list-style-type: none"> • constraints satisfied ✓ • $\alpha_6^* \simeq \alpha_7^*$ • σ_6^*: unclear convergence • $P_{wf,6}^* = 29.92 \text{ MW}$
	> 0	<ul style="list-style-type: none"> • p_5^* • $P_{wf,5}^* = 5.25 \text{ MW}$ 	<ul style="list-style-type: none"> • constraints satisfied ✓ • $\alpha_7^* \simeq \alpha_6^*$ • $\sigma_7^* \simeq 0$ • $P_{wf,7}^* = 29.70 \text{ MW}$

(b) $\theta^W = \pi/2$, maximum wake effect.

4.4 Distributed Optimization

4.4.1 Application of DPSO1

4.4.1.1 Communication Settings

As seen in the previous section, the functions involved in the optimization problem are the WTs power functions, which are obtained via the considered wake model of Section 4.2. Then, the WT physical neighbors defined in Subsection 4.2.3, along with their variables (α, o) , respect the definition of physical neighbors of Definition 3.1, given in Section 3.4 of Chapter 3. Note that, because of Assumption 4.3, the involved optimization variables reduce to the only axial induction factors. As described in the DPSO1 section, the physical relationships define the physical graph \mathcal{G}_p , which as mentioned, for this application case, is time-varying, as it depends on θ^W . Recall that, according to Assumption 4.1, θ^W is supposed to be slowly varying with respect

to the WTs dynamics. DPSO1 relies on communication Assumption 3.1, for which each agent, i.e. WT, is required to communicate with all its physical neighbors, and with all the WTs for which it is in turns a physical neighbor. As a result, the defined communication graph \mathcal{G}_c is a priori time-varying too. For the application of DPSO1 algorithm though, we consider the following

Assumption 4.4 *The communication graph \mathcal{G}_c is time-invariant, and equal to the undirected graph associated to the digraph \mathcal{G}_p corresponding to the θ^W values for which \mathcal{G}_p has higher connectivity.*

As an example of the above assumption, consider the wind farm layout of Fig. 4.8. According to the continuous wake model of Section 4.2, the higher graph connectivity is obtained $\forall \theta^W : \theta^W \neq k\pi, k \in \mathbb{N}_0^+$, for which cases the graph is *complete*, i.e. every pair of WTs is connected by a unique communication edge. In the following, to simplify the analysis, we consider the worst case of complete graph \mathcal{G}_c associated to any given WF. Such solution will serve as an analysis tool rather than a real implementation proposition. Indeed, clearly, it may be unfeasible to implement when big-size WFs are considered, and especially if DPSO1 is employed to solve the optimization problem. Recall that DPSO1 needs a finite-time average consensus to be run upon \mathcal{G}_c at each algorithm step. However, we complete the analysis under this strong communication assumption as it enables to shed a light on how DPSO1, and more in general *classic* PSO-like algorithms behave in the WF optimization problem resolution. Indeed, because of Proposition 3.1, DPSO1 reproduces the behavior of a centralized PSO. A different analysis will be carried out for DPSO2 in Subsection 4.4.2, as it differs from the classic PSO algorithms. Eventually, Assumption 4.4 can be relaxed, as shown in Subsection 4.4.1.4.

4.4.1.2 Problem Formulation

We formulate the WF optimization problem under the following

Assumption 4.5 *The couple (v_∞, θ^W) , and value P_{wf}^{max} are known by all the WTs of the wind farm.*

In practice, this can be achieved by letting a subset of WTs on the borders of the WF measure (v_∞, θ^W) , and a possibly different subset of WTs receive information about grid constraints. These values can be then forwarded³ to the rest of WTs of the WF via \mathcal{G}_c . The optimization problem, written in the form of (3.20), can be formulated as

$$\begin{aligned} & \min_{\{\alpha_i, i=1, \dots, N\}} - \sum_{i=1}^N P_i(\alpha_i, \alpha_{ij}, v_\infty, \theta^W) \\ & \text{subject to} \end{aligned} \tag{4.11}$$

$$P_i(\alpha_i, \alpha_{ij}, v_\infty, \theta^W) \leq \min \left\{ P_{e,n}, \frac{P_{wf}^{max}}{N} \right\}, \quad i = 1, \dots, N$$

³For instance, classic *leader-follower* consensus algorithms can be employed for this purpose, see e.g. [Olfati-Saber 2007], or Part III for basic notions on leader-follower consensus technique.

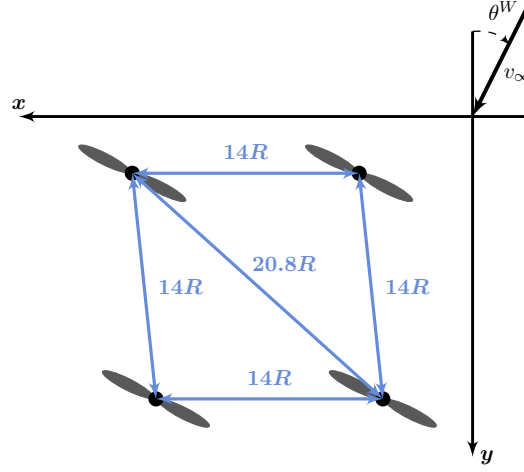


Figure 4.13: Horns Rev 1 wind farm layout.

where the private cost function associated to WT i is $-P_i$, and the private constraint function the i -th one appearing in (4.11). Because of Assumption 4.3, and thanks to Assumption 4.5, WTs can always be oriented such that $\gamma_i = \theta^W$, $1, \dots, N$, thus $o_i = 0, 1, \dots, N$, and they are not shown in (4.11) for ease of reading. Notice also that we make the choice of equally distribute the maximum allowed power production P_{wf}^{max} among the WTs, by imposing a *local* maximum power constraint of P_{wf}^{max}/N , whenever the latter is inferior to the WTs nominal power $P_{e,n}$. Other choices though possible are beyond the scope of this work and will not be treated further. As an example, the total power sharing could be computed in order to minimize the overall WF mechanical fatigue. Eventually, notice that to set the power constraint P_{wf}^{max}/N , each WT is required to have knowledge of the number of WTs in the WF.

4.4.1.3 Simulation

We test the DPSO1 algorithm for the sake of solving optimization problem (4.11). For this purpose, we consider a wind farm layout as illustrated in Fig. 4.13, where WTs coordinates are given with respect to the shown (x, y) reference frame. Here we can additionally set parameters N_x , and N_y , which are respectively the number of WTs along the x direction, and along the y direction, for a total number of $N = N_x N_y$. Such layout is the one of Horns Rev 1 wind farm in Denmark, where $N_x = 10$, and $N_y = 8$, [Park 2015a]. The tunable wake model parameters are the ones given in Table 4.1. As far as the DPSO1 parameters are concerned, we choose a maximum iteration number of 600 steps, and $N_p = 70$ particles associated to each WT, with a particle neighborhood \mathcal{S}_p of dimension $N_m = 5$.

First of all, we propose a test at low wind speed, in particular $v_\infty = 7$ m/s, with no P_{wf}^{max} constraint, i.e. no constraint is active. Simulations are carried out on Horns Rev 1 wind farm, for $\theta^W = 0, \pi/4, \pi/2$. The maximized WF power is compared with

Table 4.3: Horns Rev 1: cooperative vs greedy control

θ^W		P_{wf}^*	P_{wf}^{greedy}	Gain
	0	90.80 MW	84.90 MW	6.95%
	$\frac{\pi}{4}$	86.20 MW	78.91 MW	10.31%
	$\frac{\pi}{2}$	68.01 MW	52.34 MW	29.95%

the *greedy* control one, i.e. with the obtained WF power when all the WTs are operated at an induction axial factor equal to α_{betz} . Results are reported in Table 4.3, and they confirm the interest in considering the wake effect when maximizing the WF power extraction. This test also shows that DPSO1 achieves good performance for big-size WFs when constraints are not active. Moreover the algorithm is robust, as the results of Table 4.3 stay basically unchanged if the experience is reproduced. Here, we tested algorithm robustness on 20 trials.

As a second simulation test, we aim at analyzing the algorithm performance, with respect to the WF size, when constraints are active, since for inactive constraints, DPSO1 showed good and robust convergence properties even when N is large. From an algorithm perspective, the physical and the grid constraints are equal, as they are treated in the same way. For this reason, without loss of generality, in the sequel we only consider the physical constraints, and we select $v_\infty = 15$ m/s to be sure that they are active. Moreover, thanks to this particular choice we are also aware of the optimal solution to (4.11), and thus able to compare it with the solution provided by DPSO1 to evaluate performance. Indeed, the chosen v_∞ value is high enough to let the WTs operate at their nominal power value $P_{e,n}$, which is thus the optimal solution to (4.11). Notice that in the range of feasible wind speed values of a WT, $[v_{cutin}, v_{cutout}]$, (see Chapter 2), it always exists a value of α , to which it corresponds at least one couple of pitch angle ϑ , and rotor speed ω_r , such that the extracted power can be limited at $P_{e,n}$. It could be argued that, in this case of high wind speed value, the high optimization level is useless as it would be sufficient to saturate the extracted power of each WT at $P_{e,n}$, which is done via the local controller in classic power limiting mode of operation, (see Chapter 2). However, information of when limiting the extracted power at $P_{e,n}$ is not known a priori on the basis of the only knowledge of v_∞ value, when wake effect is considered, and WF power maximized. Such piece of information is indeed output of the high level optimization. Moreover, according to the WF layout and wind direction θ^W , there exist different ranges of v_∞ values for which some WTs should be operated at $P_{e,n}$, and some others should not. In other words, it is not a trivial task to determine a priori which WT has its physical constraint active for given ranges of v_∞ , and the problem gets even more complex if grid constraints are taken into account too.

We thus proceed as follows. We set $\theta^W = \pi/4$, and progressively increment parameters N_x , and N_y , and for each associated WF, we test DPSO1 on 20 trials. On each trial, we count the number of WTs that, according to DPSO1, would have to extract

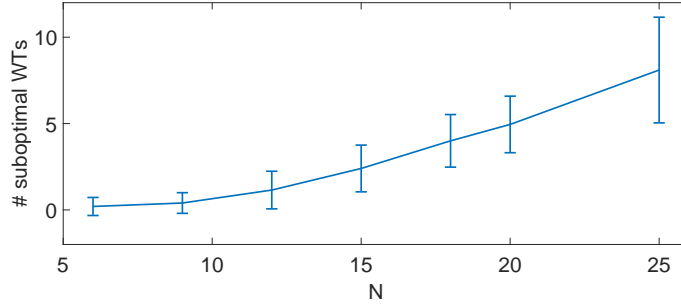


Figure 4.14: Mean and standard values out of 20 trials of the number of suboptimal WT's, for $\theta^W = \frac{\pi}{4}$, and $v_\infty = 15$ m/s.

a power value which is lower then $99\%P_{e,n}$, where 99% is considered to have some margin⁴. Such test allows us to quantify how much DPSO1 solution is suboptimal since, as mentioned, in this case the optimal one is given by all WT's in the WF operated at $P_{e,n}$. The mean and standard value of the number of suboptimal WT's for each considered couple of parameters (N_x, N_y) is shown in Fig. 4.14, where the number of WT's in the axis of abscissae are obtained with the couples (3, 2), (3, 3), (4, 3), (5, 3), (6, 3), (5, 4), and (5, 5). We can conclude that, even if DPSO1 provides good robust solutions to (4.11) when constraints are not active, this is no longer the case if they are active. Moreover performance worsens as N grows.

4.4.1.4 Wake Model Approximation

As we mentioned in Subsection 4.4.1.1, Assumption 4.4 may result in unfeasible implementation for the case of big-size WFs, because it generally leads to high communication burden. Recall that low communication burden is within the reason why a distributed architecture was considered in the first place. This is why we propose to relax the aforementioned assumption to restore this important feature. The main idea is that even if, according to the considered continuous model, each WT j influences every other downstream WT i , i.e. such that $y'_i - y'_j > 0$, (see Subsection 4.2.3), the deficit in the wind speed caused by WT's that are sufficiently far away can be neglected. We thus propose to let lower communication burden by considering an approximated wake model. This is such that the number of physical couplings among the WT's, defining \mathcal{N}_i^p , $i = 1, \dots, N$, and \mathcal{G}_p , is reduced. Since DPSO1 is based on communication Assumption 3.1, also the connectivity of \mathcal{G}_c is reduced. In particular, we consider each WT in the WF to be influenced by the WT's that are immediately adjacent to it and within a given distance. If, as an example, we consider Horns Rev 1 wind farm, the communication graph \mathcal{G}_c associated to the mentioned approximated wake couplings, reduces to the one shown in Fig. 4.15, where each WT communicates with a maximum number of 8

⁴In the considered numerical simulations, DPSO1 showed to be always able to find a feasible solution to the optimization problem. This is why potential solutions above $P_{e,n}$ are not counted.

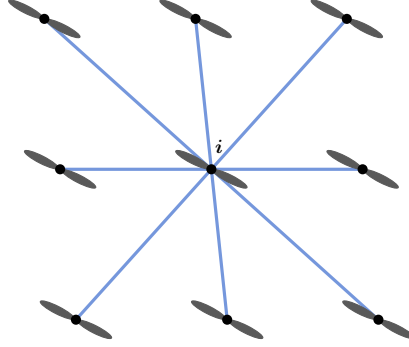


Figure 4.15: Generic WT i reduced communication graph \mathcal{G}_c for DPSO1, based on wake model approximation.

WTs. Notice that also in this case we consider \mathcal{G}_c to be time-invariant. According to this approximated wake model, each WT computes its optimal axial induction factor as if the reduced upstream WTs intercepted a wind speed value equal to v_∞ . Of course, this fact is generally not true as the majority of the WTs experiences a wind speed reduction because of the wake effect. However, the solution to (4.11) by employing the considered model approximation still provides a feasible solution when applied to the original model, and thus it is implementable. Clearly, such solution would be suboptimal with respect to the original problem. The reason for such solution to be feasible in the original model is due to the fact that the model approximation is *conservative* with respect to the constraints satisfaction. In other words, if each WT considers every upstream one to be excited by v_∞ , then it would over-reduce its own α in order to respect the power constraints. We show the above considerations on some simple examples. First of all, we apply the optimization problem based on the model approximation to Horns Rev 1 when constraints are not active, and we employ the same model parameters used in Subsection 4.4.1.3. Results are shown in Table 4.4, where we can see the percentage of power gain lost because of the model approximation. Eventually, we compare the solution of the optimization problem for the original wake model and the approximated one, in the case of active constraints. In particular we consider a WF of $(N_x, N_y) = (3, 3)$, and wind $(v_\infty, \theta^W) = (15, \pi/4)$. While in the original model, DPSO1 provides a solution in which each turbine is operated at its nominal power $P_{e,n} = 5 \text{ MW}$, in the approximated one some WTs are required to operate at 4.95 MW , which show the level of conservatism introduced by the approximated model. Overall, even though the considered model approximation applied to WFs provides a solution which is mathematically suboptimal with respect to the case of complete communication, still the obtained power gains, and power values are similar. Moreover, the results are obtained with highly reduced communication burden, and this justify the interest for this approximated solution. Unfortunately DPSO1 reveals worse performance when constraints are active and the number of WTs in the WF grows. This issue is recovered by the application of DPSO2, described in the next subsection.

Table 4.4: Horns Rev 1: original vs approximate wake model control

		P_{wf}^*	P_{wf}^{approx}	Gain
θ^w	0	90.80 MW	90.78 MW	0.02%
	$\frac{\pi}{4}$	86.20 MW	85.99 MW	0.24%
	$\frac{\pi}{2}$	68.01 MW	67.04 MW	1.45%

Remark 4.3 *As it has been stressed out in Remark 3.5 in the general DPSO1 description of Chapter 3, even if the wake model approximation considered in this subsection lets reduction of communication burden, still DPSO1 needs to perform an average consensus algorithm over the whole farm at each iteration step. In other words, each WT needs to estimate information about the whole wind farm WTs via the average consensus algorithm in order to properly update its particles and personal bests and converge to a meaningful solution.*

4.4.2 Application of DPSO2

4.4.2.1 Discussion on Wake Model Approximation

Differently from what done for DPSO1, we evaluate DPSO2 performance directly with respect to an approximated model that allows to reduce the communication burden. Indeed, because of the continuous wake model and the fact that DPSO2 relies on the same communication Assumption 3.1 as DPSO1, it would be generally required to implement a complete communication graph \mathcal{G}_c . However the wake model approximation of Subsection 4.4.1.4 considered for DPSO1 cannot be applied to DPSO2 as, in this case, it would not let a proper common fitness function evaluation, thus leading to a meaningless algorithm convergence. This fact is intimately related to the algorithm common fitness function evaluation. This is explained in the following. First of all, recall from Section 4.2, that in order to compute its own power function $P_i = \frac{1}{2}\rho\pi R^2 \bar{v}_i^3 C_p(\alpha_i)$, each WT i needs to compute the wind speed value \bar{v}_i . According to the wake model, this depends on the upstream WTs axial induction factors, and on v_∞ , the free stream wind speed or the wind captured by the most upstream WTs in the farm. In Subsection 4.4.1.4, the wake model reduction is concerned with approximating the wind deficit experienced by each WT as caused by a *smaller* subset of the actual upstream WTs. Thus, in order to compute \bar{v}_i , the approximation also consists in letting the wind speed value of the most upstream WT in this smaller subset of upstream WTs with respect to WT i , be v_∞ . We name WT_i^{sup} such upstream WT. This is generally not true as the wind captured by WT_i^{sup} , v_i^{sup} , depends in turns on v_∞ via the wake model. As already discussed, this leads to a conservative yet implementable solution. However, the point we need to stress here is that the approximation due to the choice $v_i^{sup} = v_\infty$, $i = 1, \dots, N$, still leads to a meaningful DPSO1 convergence because WT particles

are compared on how they fit the *same* common fitness function, via the finite-time average consensus algorithm, (see Remark 3.5). In DPSO2 algorithm, particles are evaluated on a *reduced* common fitness function thanks to the use of the context vectors technique, (see Remark 3.11). Nonetheless, in order to be sure that relationship (3.39) is satisfied, i.e. that the particles evaluation on the reduced common fitness function leads to the same result as if they were evaluated on the whole one, P_i , $i = 1, \dots, N$ need to be representative of the real wake model. Thus, to compute \bar{v}_i , approximation $v_i^{sup} = v_\infty$ cannot be considered. Its value needs to be the one determined by the wake model. We can summarize the above discussion in the following points

- I v_i^{sup} is needed by WT i to compute \bar{v}_i .
- II v_i^{sup} generally depends on WT_i^{sup} upstream WTs axial induction factors.
- III We want to reduce direct communication.

Thus, if v_i^{sup} is computed by WT i itself, then direct communication with the WTs influencing its value is needed. This is why we require v_i^{sup} to be computed locally by WT_i^{sup} , and its value communicated to WT i . Because of the above point II, its value is likely to change during the run of DPSO2. As a result, it needs to be sent to the corresponding WT i at each algorithm iteration. The details of DPSO2 algorithm applied to solve the wind farm optimization problem are described in Subsection 4.4.2.3. How to choose WT_i^{sup} and, more in general, each WT i neighborhood, is object of the next subsection.

Remark 4.4 *If WT_i^{sup} is such that the wind blowing in front of it is actually the free stream wind, i.e. $v_i^{sup} = v_\infty$, then its value needs to be sent to the corresponding WT i only once at the beginning of DPSO2 algorithm, as during the run of the algorithm, v_∞ is supposed to be constant, (see Assumption 4.1).*

Remark 4.5 *One may think that since v_i^{sup} is computed by WT_i^{sup} according to the wake model, no model approximation is actually introduced. This is unfortunately not true because of the distributed nature of the optimization algorithm and the subsequent communication steps. This will be more clear in Subsection 4.4.2.3, where the detailed DPSO2 algorithm is illustrated. We can though anticipate that at algorithm iteration k , each WT i computes its current power function P_i employing v_i^{sup} , received from WT_i^{sup} . However, v_i^{sup} refers to the value computed by WT_i^{sup} at the previous iteration $k - 1$, by employing its own upstream WTs variables available at $k - 1$. Such relay of information concerning v_i^{sup} introduce delays, in turns responsible for the errors in the computation of the power functions.*

4.4.2.2 Communication Settings

The aim of the communication settings is to find the minimal communication graph \mathcal{G}_c sufficient to let the solution of optimization problem (4.11), via DPSO2, i.e. to guarantee satisfaction of the problem constraints and to let convergence to a good-enough solution.

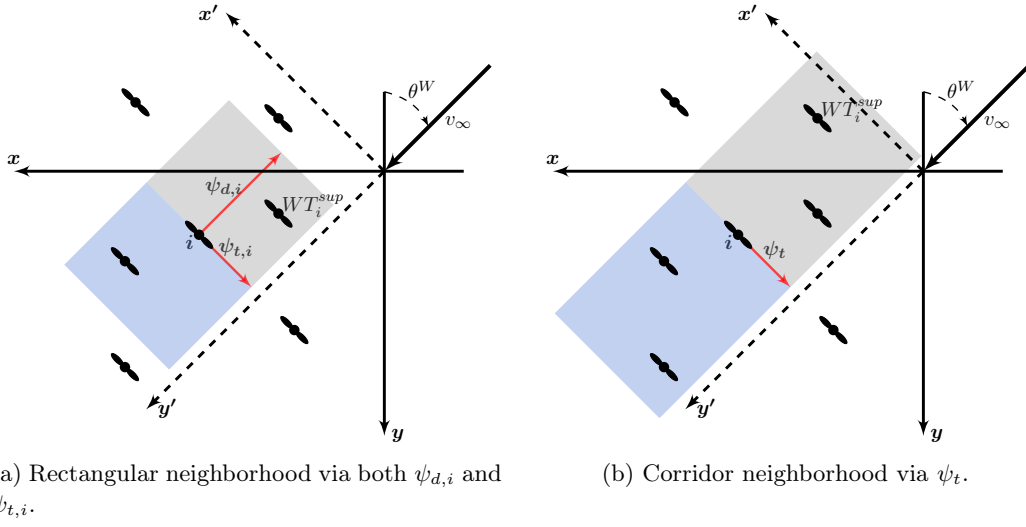


Figure 4.16: WT i neighborhood for DPSO2: WT_i^{up} in light gray area, WT_i^{down} in light blue area.

Wind Turbine Neighborhood

The definition of the communication neighborhood of WTs associated to WT i is done via the introduction of two parameters, $\psi_{d,i}$, and $\psi_{t,i}$, for each WT i in the farm. These define respectively a distance in the wind direction, and a distance in the transversal direction of the wind. Such parameters are in fact function of θ^W : $\psi_{d,i}(\theta^W)$, $\psi_{t,i}(\theta^W)$. The neighborhood of WT i is obtained as shown in Fig. 4.16a. For a given wind direction θ^W , by selecting a value for the aforementioned parameters, a *rectangle* centered in WT i is obtained. Then, all the WTs laying within it are part of the communication neighborhood of WT i . Notice that such neighborhood is associated to the θ^W value. Thus, differently from the DPSO1 case, here \mathcal{G}_c is *time-varying*, as θ^W may change in time. The neighborhood thus defined via $\psi_{d,i}$, and $\psi_{t,i}$ can be further divided in the upstream turbines subset, which we name WT_i^{up} , i.e. WT i physical neighbors, and the downstream turbines subset, which we name WT_i^{down} , i.e. those turbines for which WT i is a physical neighbor. Thus WT_i^{sup} , defined as the most upstream WT in the neighborhood of WT i , belongs to WT_i^{up} . The problem of finding the minimal communication graph can be thus reformulated as the one of finding the minimum values for $\psi_{d,i}$, and $\psi_{t,i}$ for each possible wind direction that guarantee a satisfactory solution to the optimization problem.

Computation of Wind Turbine Neighborhood Parameters

For the sake of simplicity, from now on we consider the distance parameters defined above to be the same for each WT in the wind farm. We can thus drop index i : ψ_d , ψ_t . This generally holds true for WFs having a regular geometry. Notice that, for

every set value of $\psi_t(\theta^W)$, parameter $\psi_d(\theta^W)$ defines the WT_i^{sup} turbine associated to WT i , and thus v_i^{sup} . This is of particular importance since, as discussed in Subsection 4.4.2.1, v_i^{sup} directly influences the computation of \bar{v}_i , and ultimately P_i . From numerical simulations, it appears that different minimal values of parameters $\psi_d(\theta^W)$, $\psi_t(\theta^W)$ need to be considered according to whether the optimization problem constraints are active or not. If for the sake of simplicity we consider the only physical constraints, this means that for any set θ^W value, different parameters are required according to different wind speed v_∞ ranges of values. As it will be shown in Subsection 4.4.2.4, typically when constraints are inactive, i.e. for low v_∞ values, the required ψ_d , ψ_t values to guarantee a good-enough solution are smaller than for the active constraints case. This implies that when constraints are inactive, less communication is required, and convergence time can be reduced. However, as it has been mentioned, whether constraints are active or not is not known a priori, and it is output of the optimization problem solution itself. For this reason, we provide here the computation technique of the neighborhood parameters in the worst case of active constraints, as it includes the inactive one. In particular, in order to get a problem solution that satisfies the constraints, it appears important to reduce the model approximation due to the information relay, (see Remark 4.5). This is why we choose to consider an infinite value for parameter $\psi_d(\theta^W)$, $\forall \theta^W$, and leave the only $\psi_t(\theta^W)$ parameter as degree of freedom. This means that the generic WT i neighborhood is given by all the WTs laying within the *corridor* defined via ψ_d . Notice that WT_i^{sup} is now only determined by ψ_t . This is shown in Fig. 4.16b. The computation of ψ_t for every possible θ^W in the WF is done *off-line* and via *simulation*. In particular, for every considered θ^W , starting from a low $\psi_t(\theta^W)$ value, the optimization is run via DPSO2, and $\psi_t(\theta^W)$ value incremented until reaching a satisfactory solution to the optimization problem. The obtained function $\psi_t(\theta^W)$ can be thus stored in each WT, for instance, via a lookup table.

Wind Turbine Neighborhood Computation

Each WT can compute its own neighborhood based on the knowledge of $\psi_t(\theta^W)$ and on the following assumptions.

Assumption 4.6 *Each WT has knowledge of the number N of WTs in the wind farm, and their position (x_i, y_i) , $i = 1, \dots, N$ with respect to a given reference frame.*

Differently from DPSO1 case we consider

Assumption 4.7 *Each WT measures (v_i, θ_i) , i.e. the speed and direction values of the wind blowing in the front of their rotor plane.*

Thus, according to Assumption 4.2, and to the wake model, $\theta_i \simeq \theta^W$, $i = 1, \dots, N$. Given θ^W , the subroutine that the generic WT i has to perform to compute its neighborhood is shown in Algorithm 8, where (x'_i, y'_i) values are computed via (4.5).

Algorithm 8 Wind Turbine i Neighborhood Computation**Input:** $(x'_i, y'_i), i = 1, \dots, N$ **Output:** $WT_i^{sup}, WT_i^{up}, WT_i^{down}$

```

1: distance =  $y'_i$ 
2:  $WT_i^{sup} = \{\}, WT_i^{up} = \{\}, WT_i^{down} = \{\}$ 
   LOOP Process
3: for ( $j = 1$  to  $N$ ) and ( $j \neq i$ ) do
4:   if  $|x'_i - x'_j| < \psi_t$  then
5:     Establish communication between WT  $i$  and WT  $j$ 
6:     if  $y'_j < y'_i$  then
7:       Add WT  $j$  to  $WT_i^{up}$ 
8:     else
9:       Add WT  $j$  to  $WT_i^{down}$ 
10:    end if
11:    if  $y'_j < \text{distance}$  then
12:       $WT_i^{sup} = j$ 
13:      distance =  $y'_j$ 
14:    end if
15:  end if
16: end for
17: return  $WT_i^{sup}, WT_i^{up}, WT_i^{down}$ 

```

Algorithm 9 Wind Farm Optimization Process

```

1:  $v_i = \{\}, \theta_i = \{\}, i = 1, \dots, N$ 
2:  $P_{wf}^{max} = \text{Inf}$ 
   LOOP Process
3: while WF is active do
4:   Each WT measures wind values  $(v_i, \theta_i)$ 
5:   Each WT receives  $P_{wf}^{max}$  if imposed by the grid operator
6:   if  $v_i, \theta_i$  or  $P_{wf}^{max}$  has changed then
7:     Update yaw value  $\gamma_i = \theta_i, i = 1, \dots, N$ 
8:     Each WT computes its communication neighborhood via Algorithm 8
9:     Run DPSO2, i.e. Algorithm 10
10:    Apply the optimal axial induction factors
11:   end if
12: end while

```

4.4.2.3 DPSO2 Algorithm for Wind Farm Optimization

The wind farm optimization process employing DPSO2 algorithm is based on the previously introduced Assumptions 4.1, 4.2, 4.6, 4.7 and, as for DPSO1 application, on the following

Assumption 4.8 Value P_{wf}^{max} is known by all the WTs in the wind farm.

Algorithm 10 DPSO2 for Wind Farm Optimization Problem**Output:** Local bests: $l_{i,p}$, $p = 1, \dots, N_p$ *Initialization :*

- 1: Randomly initialize $x_{i,p} \in [0, \alpha_{betz}]$, $s_{i,p} \in [-\bar{s}_i, \bar{s}_i]$, $p = 1, \dots, N_p$
- 2: $b_{i,p} = x_{i,p}$, $p = 1, \dots, N_p$
- 3: Randomly initialize $l_{i,p} \in [0, \alpha_{betz}]$, $p = 1, \dots, N_p$
- 4: Initialize $v_{i,p} = v_i$, $p = 1, \dots, N_p$, being v_i the measured wind speed

LOOP Process

- 5: **for** $k = 1$ to **max_iter** **do**
- 6: Send $x_{i,p}$, $b_{i,p}$, $l_{i,p}$, $v_{i,p}$ $p = 1, \dots, N_p$ to all WT $k \in WT_i^{down}$, via \mathcal{G}_c
- 7: Wait to receive $x_{j,p}$, $b_{j,p}$, $l_{j,p}$, $v_{j,p}$ $p = 1, \dots, N_p$ from all WT $j \in WT_i^{up}$, via \mathcal{G}_c
- 8: Compose the context vectors associated to the its own particles $x_{i,p}$:
 $\mathbf{x}_{i,p}^l \triangleq (x_{i,p}, l_{j,p} : j \in WT_i^{up})$, $p = 1, \dots, N_p$
- 9: Compose the context vectors associated to its own personal bests $b_{i,p}$:
 $\mathbf{b}_{i,p}^l \triangleq (b_{i,p}, l_{j,p} : j \in WT_i^{up})$, $p = 1, \dots, N_p$
- 10: Compose the context vectors associated to the particles $x_{j,p}$ of its neighbors $j \in WT_i^{up}$: $\mathbf{x}_{j,p}^l \triangleq (x_{j,p}, l_{i,p}, l_{k,p} : k \in WT_i^{up} \wedge k \neq j)$, $p = 1, \dots, N_p$
- 11: Compose the context vectors associated to the personal bests $b_{j,p}$ of its neighbors $j \in WT_i^{up}$: $\mathbf{b}_{j,p}^l \triangleq (b_{j,p}, l_{i,p}, l_{k,p} : k \in WT_i^{up} \wedge k \neq j)$, $p = 1, \dots, N_p$
- 12: Among the received $v_{j,p}$, select $v_{k,p}$ where $k = WT_i^{sup}$, for $p = 1, \dots, N_p$
- 13: Compute $v_{i,p}$ according to the wake model, using $v_{k,p}$ as upstream wind value, and $\mathbf{x}_{i,p}^l$ as operating points, for $p = 1, \dots, N_p$
- 14: Evaluate particle context vectors $\mathbf{x}_{i,p}^l$ in P_i : $P_{i,p}^x \triangleq P_i(\mathbf{x}_{i,p}^l)$, $p = 1, \dots, N_p$
- 15: Evaluate personal best context vectors $\mathbf{b}_{i,p}^l$ in P_i :
 $P_{i,p}^b \triangleq P_i(\mathbf{b}_{i,p}^l)$, $p = 1, \dots, N_p$
- 16: Evaluate particle context vectors $\mathbf{x}_{j,p}^l \forall j \in WT_i^{up}$ in P_i : $P_{i,p}^{x,j} \triangleq P_i(\mathbf{x}_{j,p}^l)$, $p = 1, \dots, N_p$
- 17: Evaluate personal best context vectors $\mathbf{b}_{j,p}^l \forall j \in WT_i^{up}$ in P_i : $P_{i,p}^{b,j} \triangleq P_i(\mathbf{b}_{j,p}^l)$, $p = 1, \dots, N_p$
- 18: Send $P_{i,p}^{x,j}$, $P_{i,p}^{b,j}$, $p = 1, \dots, N_p$ to the corresponding WT $j \in WT_i^{up}$, via \mathcal{G}_c
- 19: Wait to receive $P_{k,p}^{x,i}$, $P_{k,p}^{b,i}$, $p = 1, \dots, N_p$ from the corresponding WT $k \in WT_i^{down}$, via \mathcal{G}_c
- 20: Compute *fitness function* values $\tilde{F}_{i,p}^x$, $p = 1, \dots, N_p$
- 21: Compute *fitness function* values $\tilde{F}_{i,p}^b$, $p = 1, \dots, N_p$
- 22: Update personal bests for $p = 1, \dots, N_p$

$$(b_{i,p}^{new}, \tilde{F}_{i,p}^{new}) = \begin{cases} (x_{i,p}, \tilde{F}_{i,p}^x) & \text{if } \tilde{F}_{i,p}^x < \tilde{F}_{i,p}^b \\ (b_{i,p}, \tilde{F}_{i,p}^b) & \text{otherwise} \end{cases}$$

- 23: $b_{i,p} = b_{i,p}^{new}$, $\tilde{F}_{i,p}^b = \tilde{F}_{i,p}^{new}$
- 24: Update local bests for $p = 1, \dots, N_p$

$$l_{i,p}^{new} = l_{i,p} - \beta \left(\arg \min_{\{b_{i,p} \in S_p\}} \{ \tilde{F}_{i,p}^b \} - l_{i,p} \right)$$

- 25: $l_{i,p} = l_{i,p}^{new}$, $p = 1, \dots, N_p$
- 26: Perform PSO update (3.30), with box constraints handled via
- 27: **end for**
- 28: **return** $l_{i,p}$, $p = 1, \dots, N_p$

Table 4.5: Minimum and maximum number of direct communications.

		(C_{min}, C_{max})	
θ^W		Reduced Communication	Min. Opt. Communication
	0	(1, 1)	(7, 12)
	$\frac{\pi}{4}$	(0, 1)	(0, 9)
	$\frac{\pi}{2}$	(1, 1)	(9, 9)

The overall process is illustrated in Algorithm 9, whose main part is DPSO2 algorithm, which, for the reader's convenience, we report in this subsection in Algorithm 10, and where variables are referred to the specific case of wind farm optimization problem (4.11). Following the discussion of Subsection 4.4.2.1, we need to introduce a wind speed variable that allows the power function computation. In particular, we associate a wind speed value $v_{i,p}$ to each particle in WT i . These values are initialized in the WT i measured one, v_i . At each iteration, each WT i receives the wind speed variables $v_{j,p}$ from WTs j in its physical neighborhood WT_i^{up} , for $p = 1, \dots, N_p$. Then, in order to update its own wind speed values $v_{i,p}$, $p = 1, \dots, N_p$, in Step 12, WT i selects the wind values $v_{k,p}$, $p = 1, \dots, N_p$, belonging to the most upstream WT in its neighborhood, i.e. $k = WT_i^{sup}$. The generic value $v_{k,p}$ thus has the role of v_i^{sup} , introduced in Subsection 4.4.2.1. WT i values $v_{i,p}$, $p = 1, \dots, N_p$ may serve in turns as v_i^{sup} for some downstream WTs.

Once the power functions have been evaluated in the required context vectors, and sent back to the according WTs, each WT i can compute the *fitness function* values associated to its particles context vectors, $\tilde{F}_{i,p}^x$, and to its personal bests context vectors, $\tilde{F}_{i,p}^b$. This is done in Steps 17, and 18 respectively. We provide here the computation formula for $\tilde{F}_{i,p}^x$. Same results hold for $\tilde{F}_{i,p}^b$. $\tilde{F}_{i,p}^x$ is computed according to Deb's rule as

$$\begin{aligned}
 F_{i,p}^x &\triangleq -P_{i,p}^x - \sum_{k \in WT_i^{down}} P_{k,p}^{x,i} \\
 G_{i,p}^x &\triangleq P_{i,p}^x - \min \left\{ P_{e,n}, \frac{P_{wf}^{max}}{N} \right\} \\
 \tilde{F}_{i,p}^x &\triangleq \begin{cases} F_{i,p}^x & \text{if } G_{i,p}^x < 0 \\ G_{i,p}^x & \text{otherwise} \end{cases}
 \end{aligned}$$

4.4.2.4 Simulation

In order to test DPSO2 algorithm, in the following we provide simulations for both inactive and active constraints in the optimization problem. We first analyze the former case. For this purpose we consider Horns Rev 1 wind farm, where $N = 80$, and its layout is the previously shown one in Fig. 4.13. The test is carried out for a free stream wind speed value $v_\infty = 7$ m/s, and directions $\theta^W = 0, \pi/4, \pi/2$. As a far

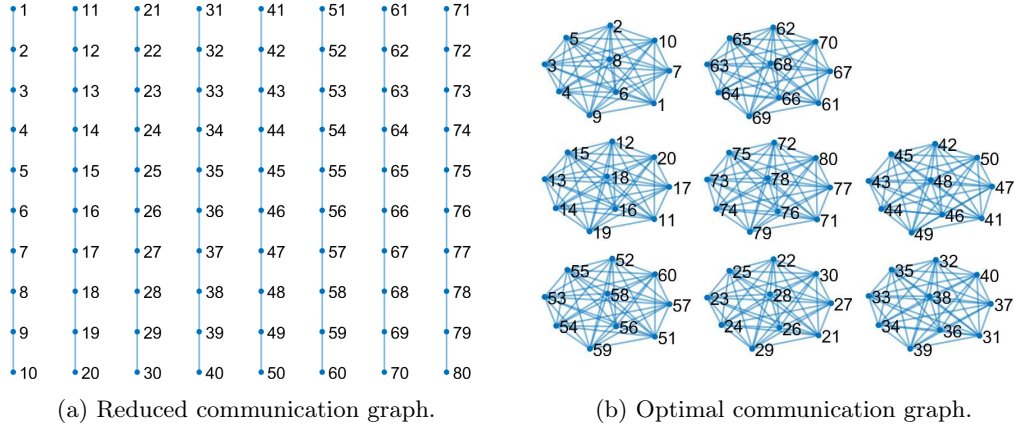


Figure 4.17: Horns Rev 1 communication graphs for $(v_\infty, \theta^W) = (7, \frac{\pi}{2})$.

as the algorithm parameters are concerned, we associate $N_p = 5$ particles to each WT. Each particle has access to the information concerning a subset of particles of dimension $N_m = 2$. Moreover we set the damping factor β as 0.8, (see Section 3.5 in Chapter 3). As we mentioned in Subsection 4.4.2.2, when constraints are not active, the required direct communication among the WTs can be further reduced with respect to the active constraints case, and this can still lead to a good-enough solution. This is why, for each chosen wind direction, we aim at testing the algorithm performance in both the case of reduced communication and minimal optimal communication. By minimal optimal communication we mean the minimum values for parameters ψ_d , and ψ_t that guarantee a near-to-optimum solution. For the sake of simplicity though, let us set $\psi_d = \infty$, and only choose parameter ψ_t . This implies that we reduce the wake model approximation due to the issue of information relay, (see Remark 4.5), and we manage the lateral wake interference among the WTs in the wake model via ψ_t . Thus, ψ_t is chosen in such a way that any additional increase in its value leads to a negligible or absent improvement in the solution to the optimal problem. In the reduced communication case instead, we set ψ_d , and ψ_t parameters so that any WT only communicates with its immediately upstream and downstream WT. Notice that in this case the model approximation due to information relay is maximal. The considered communication cases represent two opposite extreme situations as far as the model approximation is concerned.

Rather than providing ψ_d , and ψ_t values for each selected case, we give $C_{min} \triangleq \min_{i=1,\dots,N}(|WT_i^{up}| + |WT_i^{down}|)$, and $C_{max} \triangleq \max_{i=1,\dots,N}(|WT_i^{up}| + |WT_i^{down}|)$, where $|\cdot|$ indicates the set cardinality, as they give an idea of the required direct communication among the WTs in the WF. This is shown in Table 4.5. An example of communication graph \mathcal{G}_c for $\theta^W = \pi/2$ for both the reduced and optimal communication is given in Fig. 4.17, while an example of global best trajectories during the run of DPSO2 for the same wind direction and both communication cases is illustrated in Fig. 4.18. Similar trajectories are obtained for the other considered wind directions, and this allows us to conclude on the good speed of convergence

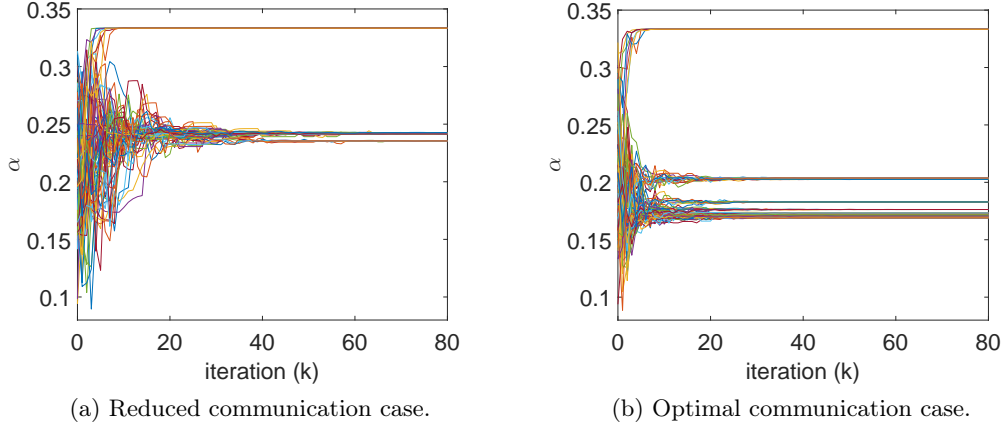


Figure 4.18: Global bests trajectories during DPSO2 iterations for Horns Rev 1, and $(v_\infty, \theta^W) = (7, \frac{\pi}{2})$.

Table 4.6: Optimal solution and power gain for different communication settings.

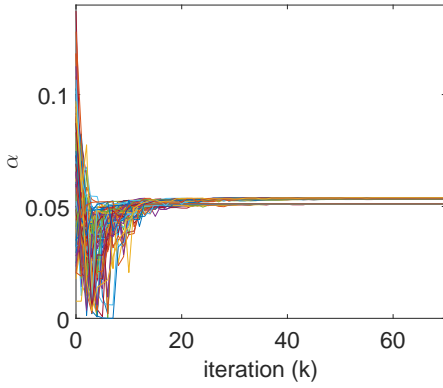
θ^W		Reduced Communication P	Min. Opt. Communication P	P gain
	0	90.30 MW	90.80 MW	0.56%
	$\frac{\pi}{4}$	85.22 MW	86.20 MW	1.14%
	$\frac{\pi}{2}$	64.56 MW	68.01 MW	5.35%

of the algorithm for the wind farm case too. The performance in terms of extracted power for the two communication graphs cases are compared and results are reported in Table 4.6. From this it is clear that there might be situations, as for $\theta^W = \pi/2$, where a more extensive communication is justified by the resulting power gains. In the same way, for other wind directions, considering a reduced communication, and thus an according more approximated wake model, does not lead to important gain loss, and the overall algorithm speed of convergence is improved.

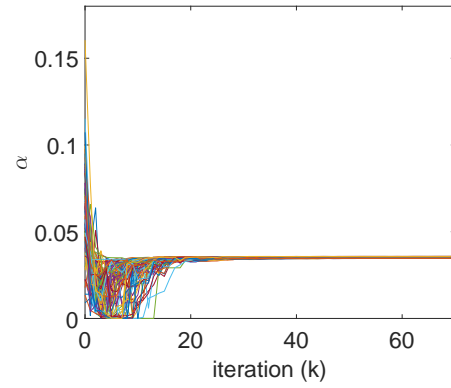
If for the inactive constraints case, considering an approximated wake model via the use of both ψ_t and ψ_d parameters, leads to suboptimal yet acceptable problem solutions, this is no longer the case when constraints are active. In particular, from numerical simulations, it appears necessary to highly reduce or even eliminate any model approximations due to information relay. This is why, as anticipated, for this case, ψ_d is set as ∞ , and the minimal required direct communication among the WTs is managed via the only ψ_t parameter. As a matter of fact, this implies more extensive communication to let the distributed optimization provide a feasible and implementable solution. Thus, in order to quantify the needed direct communication, we first test DPSO2 algorithm for different WF sizes with equal N_x , and N_y parameters and layout of Fig. 4.13. As for the previous inactive constraints case, we carry out simulations for $\theta^W = 0, \pi/4, \pi/2$, while we set $v_\infty = 15$ m/s to let the

Table 4.7: (C_{min}, C_{max}) for WFs of different size and $v_\infty = 15$ m/s, i.e. with active physical constraints.

	N			
	25	49	81	169
0	(4, 4)	(6, 9)	(7, 13)	(15, 34)
$\frac{\pi}{4}$	(0, 4)	(0, 7)	(0, 11)	(1, 24)
$\frac{\pi}{2}$	(4, 4)	(6, 6)	(8, 8)	(12, 12)



(a) Physical constraints.



(b) Maximum allowed power constraint.

Figure 4.19: Global best trajectories during DPSO2 iterations for Horns Rev 1, and $(v_\infty, \theta^W) = (15, \frac{\pi}{3})$.

physical constraints be active. For the sake of simplicity, these are the only considered active constraints. Results are illustrated in Table 4.7, where for each WF configuration we report C_{min} and C_{max} associated to the selected ψ_t value. This is chosen to be the minimal value letting the problem constraints be satisfied. Eventually we show DPSO2 convergence properties on the case of Horns Rev 1 for active constraints. In particular, in Fig. 4.19a the global best trajectories during the run of the algorithm are shown for the case of active physical constraints, while the case of grid constraints is illustrated in Fig. 4.19b. In this last case the maximum power constraint is set to $P_{wf}^{max} = 280$ MW, i.e 70% of the maximum extractable one from the wind. Both simulations are carried out for $v_\infty = 15$ m/s, and $\theta^W = \pi/3$, and parameter ψ_t such that $(C_{min}, C_{max}) = (0, 6)$. Moreover, for this last simulation example we augmented N_p to 15 as far as the associated particles to each WT are concerned. Constraints are respected in all the aforementioned cases with obtained power values which are not less the 99% of the maximum allowed one.

4.5 Hierarchical Control Evaluation

4.5.1 Combining High Level Optimization and WT Controllers

In the previous sections, by comparing the power gains with respect to the classic greedy WF operating mode, we justified the interest in taking into account the wake model when controlling a wind farm. However, as mentioned in Section 4.1, these gains may not reflect the reality, as the high level optimization is based on some simplified model assumptions for which the dynamics of the system are not taken into account. For instance, it is often assumed that the local WT controllers are able to perfectly stabilize the WTs in the optimal references provided by the optimization step. This is generally not the case since, as we saw in Chapter 2, controlling a WT poses a nontrivial problem itself. This is why in this section we aim at evaluating the WF actual gains and performance by considering the whole hierarchical architecture. In other words, the WF control is analyzed from the high level optimization, performed in a distributed way via DPSO2, to the local WT controller, which in this work is the one presented in Chapter 2. Note that the study of this section is still based on a static wake model, i.e. any change in (v_∞, θ^W) is supposed to cause an instantaneous change in the WF wind field.

As a first step, we need to let a proper interface between the high and low control levels. Recall that optimization problem (4.11) provides the optimal axial induction factor α^* associated to each WT, while the WT local control is based on variables ω_r , ϑ , i.e. respectively the WT rotor angular speed, and its pitch angle. Moreover, the optimization is based on a WT power function description that depends on a theoretical expression of the power coefficient, which we rename here $C_p^{th} \triangleq 4\alpha(1 - \alpha)^2$. On the other hand, the local controller is based on a power coefficient expressed as a function of (ω_r, ϑ) , and the wind speed v , which is usually obtained experimentally and provided by the WT producer as a lookup table. Recall that in Chapter 2 we made use of CART turbine one, which we rename here C_p^{cart} . Thus, we first set a link between C_p^{th} and C_p^{cart} , by assuming the following relationship

$$C_p^{cart}(\lambda, \vartheta) = C_p^{th}(\alpha)\eta$$

where $\eta \in]0, 1[$ is supposed to be a constant, and thus independent from α . This implies that the optimal α provided by the optimization step stays unchanged as η does not modify the WF cost function minima. In particular η is set as

$$\eta = \frac{C_p^{cart}(\lambda^o, \vartheta^o)}{C_p^{th}(\alpha_{betz})}$$

where we remind that λ^o , ϑ^o are respectively the tip speed ratio and pitch angle operating points corresponding to the WT MPPT operating mode.

Once the optimization step is completed and α^* delivered to the corresponding WT, each wind turbine can compute its *deloaded* power reference to be tracked. This can be simply done by deloading the MPPT power reference $P^o(v)$ of a certain factor, where we remind that $P^o(v)$ is obtained from the estimated or measured wind speed

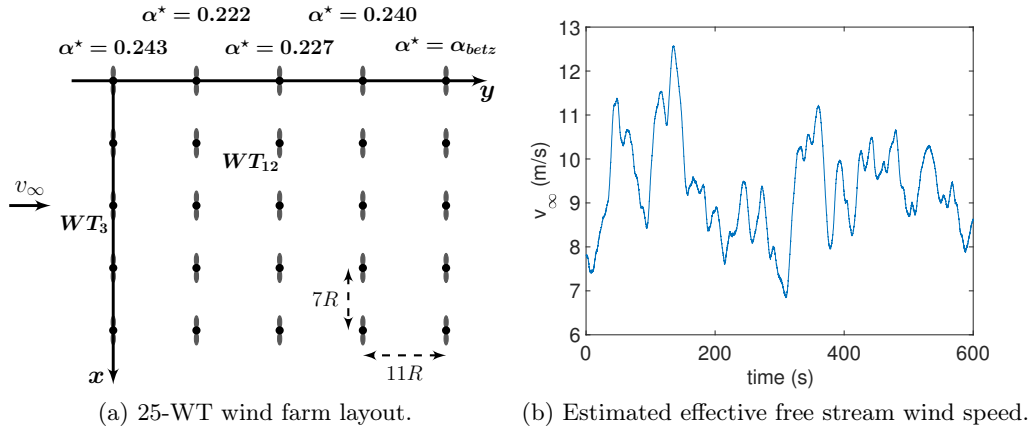


Figure 4.20: Wind farm layout and wind values.

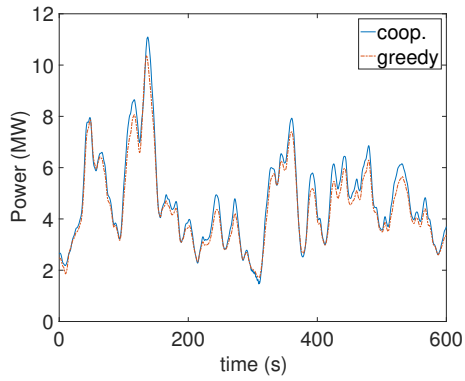
v . Thus, for any given v value, the deloaded optimal power reference P^* is computed as

$$P^*(v) = \frac{C_p^{th}(\alpha^*)\eta}{C_p^{cart}(\lambda^o, \vartheta^o)} P^o(v)$$

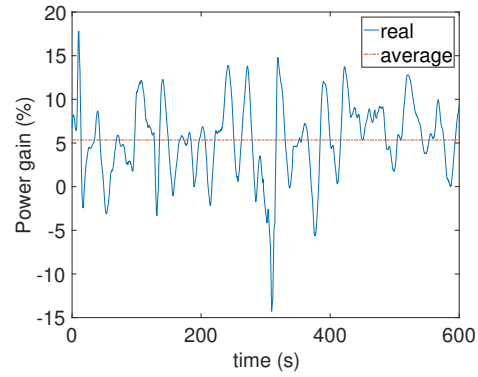
This is tracked by the WT by means of the local controller shown in Chapter 2. Recall that when tracking a deloaded power reference, the proposed controller additionally allows to store a surplus of kinetic energy in the rotating masses.

4.5.2 A Wind Farm Example Test

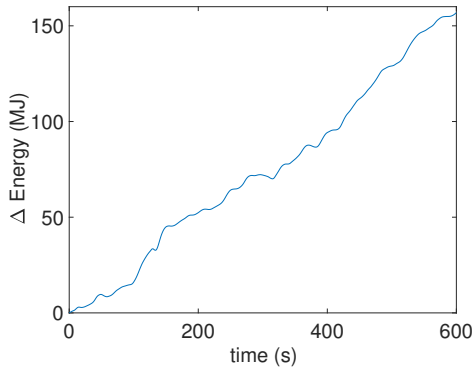
In order to evaluate the proposed hierarchical control performance and power gain we consider a 25-WT wind farm whose layout is shown in Fig. 4.20a. Its units are the CART turbines introduced in Chapter 2, which we remind having $P_{e,n} = 600 \text{ kW}$. In this simulation we aim at analyzing the actual WF power gain when the system dynamics is taken into account. For this reason, we consider no active grid constraints, and we choose a free stream wind speed value such that no nominal power constraint is active. In Fig. 4.20b we report the free stream effective wind speed signal $v_\infty(t)$ during a time interval of 600 s. This is the estimated signal from one of the most upstream WTs in the farm via the Kalman filter shown in Chapter 2. The wind direction, shown in Fig. 4.20a, is set as $\theta^W = 0$. Both $v_\infty(t)$, and θ^W are uniform along the WF length. For such WF configuration, the high level optimization provides the optimal axial induction factors shown in Fig. 4.20a. These happen to be equal for each WT having same y -coordinate. To such optimal values it corresponds a theoretical power gain of $\sim 7.5\%$ with respect to the greedy WF operation. Fig. 4.21a shows the total WF mechanical power extracted from the wind for both the cooperative and greedy operation, where the cooperative one is illustrated in blue solid line, and the greedy one in red dash-dotted line. The corresponding power gain percentage is reported in Fig. 4.21b in blue solid line together with its average value in red dash-dotted line. This is about 5.4%.



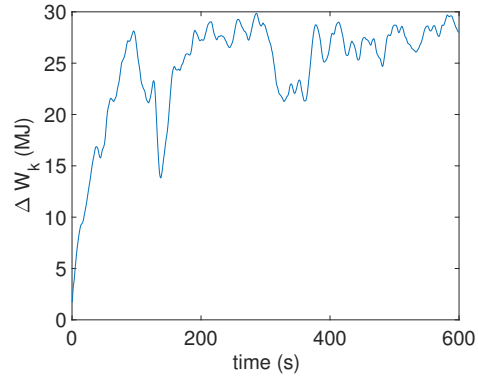
(a) Wind farm power.



(b) Power gain.



(c) Difference of injected energy in the grid.



(d) Surplus of wind farm stored kinetic energy.

Figure 4.21: Wind farm greedy vs cooperative performance.

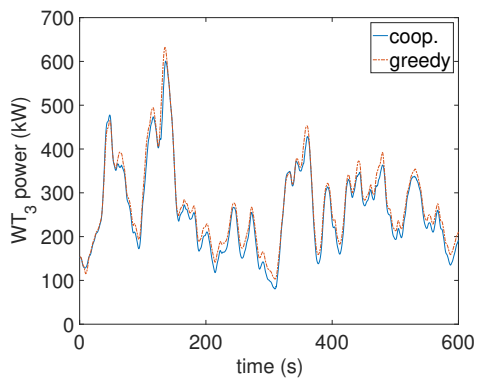
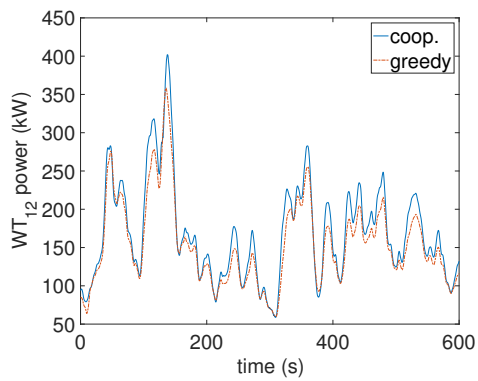
(a) WT_3 mechanical power.(b) WT_{12} mechanical power.

Figure 4.22: Greedy vs cooperative power signals.

As anticipated, the theoretical gain is not attained, basically due to the WT local controllers performance in tracking the reference optimal power signal. However,

thanks to a good control of the WTs at local level, still a relatively good gain is achieved, and this further motivates the wake effect phenomenon consideration when controlling a wind farm. The energy gain deriving from the power one is shown in Fig. 4.21c, where we report the surplus of energy delivered to the grid during the simulation time with respect to the greedy case. Moreover, since in order to achieve the aforementioned gain, the upstream WTs operate in deloaded mode, a surplus of kinetic energy stored in the rotating masses can be obtained. The cumulative surplus over the whole wind farm during the simulation is shown in Fig. 4.21d. As stated in Chapter 2, this could be in turns used to let the WF participate to some grid frequency control duties. Eventually, to get more insight into the wake interaction phenomenon, we compare the power extracted from the WTs labeled WT_3 , and WT_{12} in Fig. 4.22, for both the cooperative and greedy operation. In particular, as expected, the most upstream WTs, such as WT_3 , extract less wind power in the cooperative mode of operation. This is evidenced in Fig. 4.22a. As a result, more power is left for the downstream WTs, such as WT_{12} , which are thus able to extract more power with respect to the greedy case. This fact is illustrated in Fig. 4.22b. For both figures in Fig. 4.22, we indicate the power corresponding to the cooperative case in blue solid line, and the greedy one in via red dash-dotted line.

4.6 Conclusion and Future Perspectives

4.6.1 Conclusion

A hierarchical control for wind farm optimization under the effect of wake interaction was presented. The control architecture is composed of two levels. The higher one is responsible for solving a problem that is reduced to an optimization one via some wake model approximations. A first novelty with respect to existing research works is presented by considering the system constraints actively in the problem formulation. This is then solved in a distributed way, which appears to be the most natural way to treat large multi-agent systems such as the WFs.

The two proposed distributed metaheuristic optimization algorithms, DPSO1 and DPSO2, were tested. These were chosen to treat the nonconvex WF optimization problem. DPSO2 presented better performance in both speed of convergence and optimality of the solution over DPSO1.

The overall system performance were then tested by also considering the local WT controllers developed in Chapter 2. Even if it exists a gap between the theoretical static optimization and the actual attainable gain when system dynamics is considered, good performance is still achieved, and it consolidates the interest for cooperative control of large wind farms.

4.6.2 Future Perspectives

One main drawback of the considered distributed optimization algorithms when applied to the WF optimization problem is the difficulty in treating problem constraints. If for DPSO1, in practice this issue restricts its applicability to small-size WFs, on the other hand DPSO2 presents good scalability properties as the WF size grows. This is obtained at the price of increasing in the number of needed direct communications among the WTs. An interesting problem is thus represented by finding other wake model approximations or algorithm modifications that would let a reduced communication burden, in the same way as done when system constraints are not active.

Another pitfall may be presented by the fact of having considered a static wake model, in turns used to let the optimization problem formulation. This approximation is stronger as the WF size increases. Considering the wind dynamics in the WF high level control sets an interesting problem and an opportunity for future research work.

Eventually we conclude the chapter by recalling that in the presented control architecture, the low level control is decentralized, as the local WT controllers are not allowed to exchange information. The optimal power references are tracked with no additional modifications based on on-line local measurements. Since a distributed communication among the WTs is set because of the high level optimization, better performance could be achieved by letting the WT local controllers communicate exploiting the existing communication graph. This case of *distributed* low control level is treated in the next part.

Part III

WIND FARM DISTRIBUTED CONTROL

PID-like Consensus Protocol

Contents

5.1 Preliminaries on Consensus Control	169
5.1.1 Consensus Concept Introduction	169
5.1.2 Fundamental Consensus Algorithms	170
5.2 Discrete-time PID Consensus Control for LTI Systems .	171
5.2.1 Related Works and Contribution	171
5.2.2 Leaderless Consensus Under the Presence of Disturbances	173
5.2.2.1 Problem Formulation	173
5.2.2.2 \mathcal{H}_∞ Weighted Output Consensus	174
5.2.2.3 Fast Weighted Consensus	178
5.2.3 Leader-follower Consensus with Time-varying Reference	
State	180
5.2.3.1 Problem Formulation	180
5.2.3.2 \mathcal{H}_∞ Weighted Leader-follower Consensus	181
5.2.3.3 Fast Weighted Leader-follower Consensus	183
5.2.4 Simulation Example	183
5.3 Conclusion	188

5.1 Preliminaries on Consensus Control

ALTHOUGH a particular form of consensus control technique was anticipated in Chapter 3, we decide to devote this section to a more general introduction to its main concepts and algorithms, as we believe they may be useful for a non-expert reader in this control branch. The following notes are mainly based on [Ren 2008].

5.1.1 Consensus Concept Introduction

In recent years much research effort has been devoted to the area of multi-agent cooperative control because of its wide range of applications and potential benefits. Co-operation of a coordinated multi-agent network is sought via distributed algorithms as they present some interesting advantages over their centralized counterpart. For instance, as discussed in the previous chapters, they allow avoiding single point of

failure, reducing communication and computational burden, etc. Distributed coordination of multi-agent systems is based on the axiom that *shared information is a necessary condition for cooperation*, [Ren 2008]. In other words, the piece of information that has to be exchanged among the agents is the one needed to achieve cooperation, and it is called *coordination variable*. This is the key to cooperation, and it does not have to be confused with the control goal, which is in fact the one of letting each agent coordination variable converge to a *consistent* value, i.e. meet on a common one. The *consensus* problem is thus concerned with letting a group of agent agree on the aforementioned variable of interest, via only neighbor-to-neighbor communication, and with no information relay. In other words, the ultimate goal is to impose a global system behavior via local interactions.

5.1.2 Fundamental Consensus Algorithms

Let us suppose a set of N agents that are able to communicate on a given undirected¹ graph, whose adjacency matrix is $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$, (see Appendix C for graph theory preliminaries). The goal is to let the multi-agent system reach consensus on a coordination variable $x \in \mathbb{R}$, which for the sake of simplicity is here supposed to be a scalar. Each agent disposes of a copy x_i of x . Agreement on x is reached when the set of copies of the coordination variable in the network are consistent, i.e. when condition $x_i = x_j$, $i, j = 1, \dots, N$ is satisfied.

If the communication network is sufficiently fast, then the update law of x_i can be described by a first-order differential equation of the form

$$\dot{x}_i = u_i, \quad i = 1, \dots, N \quad (5.1)$$

where $u_i \in \mathbb{R}$ is the distributed protocol to be chosen to let consensus. The fundamental consensus algorithms were initially developed on single integrator system (5.1). If on the other hand, communication among the agents is set in discrete-time, then similarly to what done for the continuous-time case, the local agent update law can be described via a first-order difference equation.

The most common distributed protocol for system (5.1) is such that in closed-loop it takes the form of

$$\dot{x}_i = - \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j), \quad i = 1, \dots, N \quad (5.2)$$

where \mathcal{N}_i is the set of neighbors of agent i according to the communication network. Intuitively, the distributed protocol in (5.2) pushes each agent state x_i towards its neighbors ones x_j , $j \in \mathcal{N}_i$. Such control is sometimes referred to as a P-like distributed protocol, as the control action is proportional to the relative errors $x_i - x_j$. By naming $\mathbf{x} \triangleq \text{col}(x_1, \dots, x_N)$, then (5.2) can be written in matrix form as

$$\dot{\mathbf{x}} = -\mathcal{L}\mathbf{x} \quad (5.3)$$

¹The more general case of a direct graph, *digraph*, is beyond the interest of this work and it will not be treated.

where \mathcal{L} is the Laplacian matrix associated to the communication network, (see Appendix C). Similarly, for discrete-time communication the most simple consensus algorithm defines the following update law

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i \cup \{i\}} b_{ij} x_j(k)$$

where b_{ij} are such that $\sum_{j=1}^N b_{ij} = 1$, $i = 1, \dots, N$, and $b_{ij} \geq 0$, $i, j = 1, \dots, N$, being 0 when there is no direct communication between the corresponding agent i , and j . The overall system is thus described by

$$\mathbf{x}(k+1) = D\mathbf{x}(k) \quad (5.4)$$

where $D = [b_{ij}] \in \mathbb{R}^{N \times N}$ is *row-stochastic*². Classic consensus results are then based on analyzing conditions on the network topology under which systems (5.3), and (5.4) reach consensus, i.e. $x_i - x_j \rightarrow 0$, $i, j = 1, \dots, N$, and to which common value the agents agree.

Note that consensus algorithm (5.2), and its discrete-time counterpart, let in general every agent in the network have influence on the final common coordination variable value. Another well-known consensus technique is concerned with letting one agent, called *leader*, communicate with the remaining agents, called *followers*, without *receiving* information from them. Intuitively, if consensus is reached over the network, then the coordination variable value on which agents agree is *imposed* by the leader, as its dynamics is not influenced by the followers one. This technique, known as *leader-follower* consensus, is useful when a particular desired global behavior has to be imposed to the multi-agent system.

5.2 Discrete-time PID Consensus Control for LTI Systems

5.2.1 Related Works and Contribution

In the previous section we introduced the consensus control problem for a system of agents whose dynamics are governed by a single integrator differential, or difference equation. Finding a distributed protocol to solve the aforementioned problem has been extensively treated for both single and double integrator dynamic agents, e.g. [Ren 2008]. However, in a more general framework, general dynamics need to be considered in order to describe the agents behavior. The consensus problem for this case has been discussed for both continuous and discrete-time multi-agent systems. In addition, it can be further divided in two main classes of problems, namely leaderless and leader-follower ones. As far as the former is concerned, the most employed

²A square matrix $D = [b_{ij}]$ is said to be row-stochastic if all its entries are nonnegative real values and such that $\sum_{j=1}^N b_{ij} = 1$, $i = 1, \dots, N$. From this fact we know that D has an eigenvalue in 1 with corresponding eigenvector $\mathbf{1}$. This is used to conclude on system consensus reachability.

distributed protocol is given by a static state feedback law, also called P-like distributed control. One can cite, for instance, [Xi 2010, Li 2013b, Yang-Zhou 2014] for the continuous-time framework, and [Li 2013b, You 2011, Su 2012, Ge 2013] for the discrete one, where the consensus problem is led back to the one of simultaneously stabilizing multiple LTI systems. References [Li 2013b, Su 2012] also solve a leader-follower problem where the leader has an autonomous time-invariant dynamics. Another interesting problem is the one of finding the optimal P-like protocol gain in order to improve consensus under system uncertainties, as in [Li 2012], and disturbances as in [Oh 2014, Li 2011], for continuous time systems, and [Wang 2011b] for discrete-time ones. The proposed approaches usually make use of some \mathcal{H}_2 or \mathcal{H}_∞ constraints to be respected, and they are in general more involved than the one of simultaneously stabilizing multiple systems. For instance, authors of [Li 2011] provide necessary and sufficient conditions, for the continuous-time case to solve the consensus problem while guaranteeing some properties on the aforementioned norms. On the other hand, for discrete-time systems only sufficient conditions are provided using results from robust control as in [Wang 2011b]. Dynamic distributed controllers are also proposed for consensus achievement based on local output measurements, e.g. [Li 2013b]. In the continuous-time framework, authors of [Xi 2012] provide a controller with limited energy, while a general full order one is presented in [Liu 2009] to achieve some \mathcal{H}_∞ performance.

Other possible structures have been explored too. Indeed, given the common P-like controller, one can easily think of a more general PID-like structure. In continuous-time, for instance, authors of [Carli 2008] propose a PI-like distributed algorithm for single integrator dynamic agents, and in [Ou 2014] authors provide a PID-like controller for general high-order SISO systems. A similar control design is applied to solve the leader-follower consensus under time-varying reference state, as in [Ren 2007], and in its sampled-data counterpart [Cao 2009], where the authors provide a PD-like protocol. Even though the presented literature review is far from being exhaustive, one can remark that poorer attention has been devoted to discrete-time dynamic protocols for general LTI MIMO systems, and this is on what we wish to focus our attention in the sequel.

In particular, we propose a PID-like distributed controller for the aforementioned systems, where the agents can communicate on a connected undirected graph, and we provide two possible ways of tuning the controller parameters, based on the solution of given LMIs. This contribution has been object of our work of [Gionfra 2017a]. The approach we propose is used to solve two similar problems, namely the *leaderless consensus under the presence of disturbances*, and the *leader-follower consensus under a time-varying reference state*. Our main results are based on the work of [Wu 2011], which we adapted for distributed coordination purposes. The fundamental feature of the aforesaid work is that MIMO PID parameter tuning can be performed via LMIs, avoiding in this way, the need for solving BMIs. As for classic control, the PID controller allows good performance despite being rather simple. Concerning the leaderless consensus problem, for instance, it enhances the disturbance rejection, and achieves results that a simple P-like protocol would not

permit if the dynamics of the agents are general. Similar conclusions hold for the leader-follower consensus problem with a time-varying reference state, where a P-like control would undoubtedly reach lower performance.

5.2.2 Leaderless Consensus Under the Presence of Disturbances

5.2.2.1 Problem Formulation

We consider N identical agents governed by general discrete-time linear dynamics, according to

$$\begin{cases} x_i^+ = Ax_i + B_2u_i + B_1\omega_i, & i = 1, \dots, N \\ y_i = Cx_i \end{cases} \quad (5.5)$$

where $A \in \mathbb{R}^{n \times n}$, $B_2 \in \mathbb{R}^{n \times l}$, $B_1 \in \mathbb{R}^{n \times h}$, $C \in \mathbb{R}^{m \times n}$, $x_i \triangleq x_i(k) \in \mathbb{R}^n$ and $x_i^+ \triangleq x_i(k+1) \in \mathbb{R}^n$ are respectively the agent state at the current step k , and at the next step $k+1$, $u_i \triangleq u_i(k) \in \mathbb{R}^l$ is the agent control, $\omega_i \triangleq \omega_i(k) \in \mathbb{R}^h$ its disturbance, and $y_i \triangleq y_i(k) \in \mathbb{R}^m$ is the measured output and the variable on which agreement among the agents is sought. Moreover we require the system to satisfy $l \geq m$, i.e. to have a greater or equal number of inputs with respect to its outputs. For the sake of leaderless consensus, a priori we do not require A to be Schur stable. Indeed, as shown by [Ge 2013], A has a role in determining the consensus function to which the agents converge under proper control. Here it can be thought to be assigned by a previous control design step. The agents can communicate on an undirected connected graph whose Laplacian matrix \mathcal{L} has positive minimum nonzero and maximum eigenvalues respectively equal to $\underline{\lambda}_{\mathcal{L}}$, and $\bar{\lambda}_{\mathcal{L}}$. Thus, we can address the problem of finding a distributed control law for u_i such that $\|y_i/\chi_i - y_j/\chi_j\|$ is minimized for $i, j = 1, \dots, N$ with respect to the disturbance $\omega \triangleq \text{col}(\omega_1, \dots, \omega_N)$, and where the *weights* are $\chi_i \in \mathbb{R}^+$, $i = 1, \dots, N$, i.e. for the sake of simplicity of analysis they are considered to be scalars. If error $y_i/\chi_i - y_j/\chi_j = 0$, $i, j = 1, \dots, N$, then we say that *weighted consensus* is achieved. By naming $D \triangleq \text{diag}(1/\chi_1, \dots, 1/\chi_N)$, we additionally define matrix $\hat{\mathcal{L}} \triangleq D\mathcal{L}$, which satisfies Lemma C.3, and whose positive minimum nonzero and maximum eigenvalues are respectively $\underline{\lambda}_{\hat{\mathcal{L}}}$, and $\bar{\lambda}_{\hat{\mathcal{L}}}$. In this work we focus on local controllers of the form

$$\begin{cases} x_{c_i}^+ = A_c x_{c_i} + B_c s_i, & i = 1, \dots, N \\ u_i = C_c x_{c_i} + D_c s_i \end{cases} \quad (5.6)$$

where $x_{c_i} \triangleq x_{c_i}(k) \in \mathbb{R}^{2l}$ is the agent controller state, and

$$\begin{aligned} A_c &= \begin{bmatrix} I_l & I_l \\ 0_{l \times l} & 0_{l \times l} \end{bmatrix}_{2l \times 2l} & B_c &= \begin{bmatrix} (K_i - K_d) \\ K_d \end{bmatrix}_{2l \times m} \\ C_c &= [I_l \quad 0_{l \times l}]_{l \times 2l} & D_c &= [(K_p + K_i + K_d)]_{l \times m} \end{aligned} \quad (5.7)$$

where $K_p, K_i, K_d \in \mathbb{R}^{l \times m}$ are gain matrices to be tuned, and where $s_i \triangleq s_i(k) \in \mathbb{R}^m$ is defined as

$$s_i \triangleq \sum_{j=1}^N a_{ij} \left(\frac{y_i}{\chi_i} - \frac{y_j}{\chi_j} \right) \quad (5.8)$$

where a_{ij} is the (i, j) -th element of the adjacency matrix \mathcal{A} defined in Appendix C. Thus the closed-loop system for agent i has dimension $\bar{n} \triangleq n + 2l$. As shown by [Wu 2011], system (5.6) is a state representation of the discrete-time PID MIMO controller, whose z -transform between s_i and u_i is the following transfer matrix

$$K_p + K_i \frac{z}{z-1} + K_d \frac{z-1}{z} \quad (5.9)$$

Refer to Appendix D.4 for an extended representation of (5.9). The problem can be now restated as the one of finding matrices B_c , and D_c such that the effect of disturbance ω on the weighted consensus is minimized.

Remark 5.1 *Note that a more general analysis could be carried out by considering different measured and controlled output variables for system (5.5). In particular this is possible for all the results provided in this chapter that involve the solution of the described LMIs in Theorem D.1 in order to impose an \mathcal{H}_∞ constraint. However, for the sake of simplicity, this problem extension is not treated further in this work.*

5.2.2.2 \mathcal{H}_∞ Weighted Output Consensus

Refer to Appendix D.1 to recall the definition of \mathcal{H}_∞ -norm of a linear system. In order to state the main result we introduce the following definition, similar to the one given in [Wang 2015].

Definition 5.1 System (5.5) is said to achieve an \mathcal{H}_∞ weighted output consensus with a performance index $\gamma \in \mathbb{R}^+$ if, for any initial condition, $\lim_{k \rightarrow \infty} \|y_i/\chi_i - y_j/\chi_j\| = 0$ for $i, j = 1, \dots, N$ when $\omega = \mathbf{0}$, and the \mathcal{H}_∞ norms of the transfer matrices, for $i = 1, \dots, N$, between ω and $\left(\frac{y_i}{\chi_i} - \frac{1}{N} \sum_{j=1}^N \frac{y_j}{\chi_j} \right)$ are inferior to γ .

The following result is based on Theorem D.1 in Appendix D.4.

Theorem 5.1 *Given N agents described by (5.5) on an undirected connected graph; consider the distributed protocol of equations (5.6), (5.7), (5.8); then the agents achieve \mathcal{H}_∞ weighted output consensus with performance index γ if there exist two symmetric positive definite matrices $\bar{P}, \bar{P} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ such that the LMI conditions of Theorem D.1 are simultaneously satisfied for two LTI systems whose dynamic, input and measure matrices are respectively $(A, B_2, \underline{\lambda}_{\bar{C}} C)$, and $(A, B_2, \bar{\lambda}_{\bar{C}} C)$, and they both have controlled output matrix C , and disturbance input matrix B_1 .*

Proof: The closed-loop dynamics for the generic agent i , by using (5.5), (5.6), and by defining the augmented state $\xi_i \triangleq \text{col}(x_i, x_{c_i}) \in \mathbb{R}^{\bar{n}}$, and matrices $\bar{C} \triangleq$

$[C \ 0_{m \times 2l}]$, $\tilde{B} \triangleq [B_1^\top \ 0_{h \times (2l)}]^\top$ is given by

$$\begin{cases} \xi_i^+ = \hat{A}\xi_i + \hat{B} \sum_{j=1}^N a_{ij} \left(\frac{\xi_i}{\chi_i} - \frac{\xi_j}{\chi_j} \right) + \tilde{B}\omega_i \\ y_i = \bar{C}\xi_i \end{cases} \quad (5.10)$$

where

$$\hat{A} = \begin{bmatrix} A & B_2 C_c \\ 0 & A_c \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B_2 D_c \bar{C} \\ B_c \bar{C} \end{bmatrix}$$

By naming $\xi \triangleq \text{col}(\xi_1, \dots, \xi_N)$, $y \triangleq \text{col}(y_1, \dots, y_N)$, gathering together the closed-loop agents dynamic, and performing the change of coordinates $\bar{\xi} = (D \otimes I_{\bar{n}})\xi$, it yields

$$\begin{cases} \bar{\xi}^+ = (I_N \otimes \hat{A} + D\mathcal{L} \otimes \hat{B}) \bar{\xi} + (I_N \otimes \tilde{B}) \bar{\omega} \\ \bar{y} = (I_N \otimes \bar{C}) \bar{\xi} \end{cases} \quad (5.11)$$

where we named $\bar{\omega} \triangleq (D \otimes I_h)\omega$, $\bar{y} \triangleq (D \otimes I_m)y$, and we used point (i) of Lemma A.3. Similar to [Liu 2009], and [Wang 2011b], we define

$$\begin{aligned} \zeta_i &\triangleq \bar{y}_i - \frac{1}{N} \sum_{j=1}^N \bar{y}_j \\ \delta_i &\triangleq \bar{\xi}_i - \frac{1}{N} \sum_{j=1}^N \bar{\xi}_j \end{aligned}$$

Thus $\zeta_i = \bar{C}\delta_i$. Note that if $\zeta_i = 0$ for $i = 1, \dots, N$ then $\bar{y}_i = \bar{y}_j$, i.e. weighted consensus is achieved. If we now name $\delta \triangleq \text{col}(\delta_1, \dots, \delta_N)$, and $\zeta \triangleq \text{col}(\zeta_1, \dots, \zeta_N)$, we have that $\zeta = (I_N \otimes \bar{C}) \delta$, and $\delta = \bar{\xi} - \mathbf{1} \otimes \frac{1}{N} \sum_{j=1}^N \bar{\xi}_j = (\bar{\mathcal{L}} \otimes I_{\bar{n}}) \bar{\xi}$, where $\bar{\mathcal{L}}$ satisfies the conditions of Lemma C.2. Thus ζ and $\bar{\xi}$ variables are linked by relationship $\zeta = (I_N \otimes \bar{C}) (\bar{\mathcal{L}} \otimes I_{\bar{n}}) \bar{\xi} = (\bar{\mathcal{L}} \otimes \bar{C}) \bar{\xi}$. Considering the change of coordinates $\delta = (\bar{\mathcal{L}} \otimes I_{\bar{n}}) \bar{\xi}$ for system (5.11), it yields

$$\begin{aligned} \delta^+ &= (\bar{\mathcal{L}} \otimes I_{\bar{n}}) (I_N \otimes \hat{A} + \hat{\mathcal{L}} \otimes \hat{B}) \bar{\xi} + (\bar{\mathcal{L}} \otimes I_{\bar{n}}) (I_N \otimes \tilde{B}) \bar{\omega} \\ &= (\bar{\mathcal{L}} \otimes \hat{A} + \bar{\mathcal{L}} \hat{\mathcal{L}} \otimes \hat{B}) \left(\delta + \mathbf{1} \otimes \frac{1}{N} \sum_{k=1}^N \bar{\xi}_k \right) + (\bar{\mathcal{L}} \otimes \tilde{B}) \bar{\omega} \\ &= (\bar{\mathcal{L}} \otimes \hat{A} + \bar{\mathcal{L}} \hat{\mathcal{L}} \otimes \hat{B}) \delta + (\bar{\mathcal{L}} \otimes \tilde{B}) \bar{\omega} \end{aligned}$$

where we used points (i) of Lemmas C.2, C.3, and A.3. According to the (ii) point of Lemma C.2, we employ the orthogonal matrix $U \in \mathbb{R}^{N \times N}$ to define the change of coordinates: $\hat{\delta} \triangleq (U^\top \otimes I_{\bar{n}}) \delta$, $\hat{\omega} \triangleq (U^\top \otimes I_h) \bar{\omega}$, $\hat{\zeta} \triangleq (U^\top \otimes I_m) \zeta$, so that the

system equations in the new coordinates are given by

$$\begin{cases} \hat{\delta}^+ &= (U^\top \otimes I_{\bar{n}}) \left(\bar{\mathcal{L}} \otimes \hat{A} + \bar{\mathcal{L}} \hat{\mathcal{L}} \otimes \hat{B} \right) (U \otimes I_{\bar{n}}) \hat{\delta} \\ &+ (U^\top \otimes I_{\bar{n}}) \left(\bar{\mathcal{L}} \otimes \hat{B} \right) \bar{\omega} \\ &= \left(\bar{\Lambda} \otimes \hat{A} + \bar{\Lambda} U^\top \hat{\mathcal{L}} U \otimes \hat{B} \right) \hat{\delta} + \left(\bar{\Lambda} \otimes \hat{B} \right) \hat{\omega} \\ \hat{\zeta} &= (U^\top \otimes I_m) (I_N \otimes \bar{C}) (U \otimes I_{\bar{n}}) \hat{\delta} = (I_N \otimes \bar{C}) \hat{\delta} \end{cases} \quad (5.12)$$

As shown in Lemma C.2, and C.3, being the last row and column of $\bar{\Lambda}$ zeros, and the last column of $U^\top \hat{\mathcal{L}} U$ zero, we can split (5.12) in two by dividing the system variables as $\hat{\delta} = \text{col}(\hat{\delta}_1, \hat{\delta}_2)$, $\hat{\omega} = \text{col}(\hat{\omega}_1, \hat{\omega}_2)$, and $\hat{\zeta} = \text{col}(\hat{\zeta}_1, \hat{\zeta}_2)$. It follows that, to conclude on system stability, we can study the reduced order system described by

$$\begin{cases} \hat{\delta}_1^+ = \left(I_{N-1} \otimes \hat{A} + \hat{\mathcal{L}}_1 \otimes \hat{B} \right) \hat{\delta}_1 + \left(I_{N-1} \otimes \hat{B} \right) \hat{\omega}_1 \\ \hat{\zeta}_1 = (I_{N-1} \otimes \bar{C}) \hat{\delta}_1 \end{cases}$$

From Lemma C.3, it exists an invertible matrix $V \in \mathbb{R}^{(N-1) \times (N-1)} : V^{-1} \hat{\mathcal{L}}_1 V \triangleq \Lambda = \text{diag}(\lambda_1, \dots, \lambda_{N-1})$, where $0 < \underline{\lambda}_{\hat{\mathcal{L}}} \leq \lambda_i \leq \bar{\lambda}_{\hat{\mathcal{L}}}$ for $i = 1, \dots, N-1$. Thus we can define a further change of coordinates, such that $\tilde{\delta}_1 \triangleq (V^{-1} \otimes I_{\bar{n}}) \hat{\delta}_1$, $\tilde{\omega}_1 \triangleq (V^{-1} \otimes I_h) \hat{\omega}_1$, and $\tilde{\zeta}_1 \triangleq (V^{-1} \otimes I_m) \hat{\zeta}_1$. This yields

$$\begin{cases} \tilde{\delta}_1^+ = \left(I_{N-1} \otimes \hat{A} + \Lambda \otimes \hat{B} \right) \tilde{\delta}_1 + \left(I_{N-1} \otimes \hat{B} \right) \tilde{\omega}_1 \\ \tilde{\zeta}_1 = (I_{N-1} \otimes \bar{C}) \tilde{\delta}_1 \end{cases} \quad (5.13)$$

Note that the transfer function matrix of (5.13) satisfies

$$\|T_{\tilde{\zeta}_1 \tilde{\omega}_1}(z)\|_\infty = \|T_{\hat{\zeta}_1 \hat{\omega}_1}(z)\|_\infty = \|T_{\hat{\zeta} \hat{\omega}}(z)\|_\infty = \|T_{\bar{\zeta} \bar{\omega}}(z)\|_\infty \quad (5.14)$$

It follows that we can impose an \mathcal{H}_∞ constraint on transfer matrix $T_{\bar{\zeta} \bar{\omega}}(z)$ by acting on $T_{\hat{\zeta}_1 \hat{\omega}_1}(z)$. We can now separate (5.13) in $N-1$ subsystems, each of them being governed by

$$\begin{cases} \tilde{\delta}_{1_i}^+ = \begin{bmatrix} \tilde{x}_{1_i}^+ \\ \tilde{x}_{1,c_i}^+ \end{bmatrix} = \begin{bmatrix} (A + B_2 D_c(\lambda_i C)) & B_2 C_c \\ B_c(\lambda_i C) & A_c \end{bmatrix} \begin{bmatrix} \tilde{x}_{1_i} \\ \tilde{x}_{1,c_i} \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tilde{\omega}_{1_i} \\ \tilde{\zeta}_{1_i} = C \tilde{x}_{1_i} \end{cases} \quad (5.15)$$

System (5.15) can be equivalently seen as the closed-loop form of the two following systems

$$\begin{cases} \tilde{x}_{1_i}^+ = A \tilde{x}_{1_i} + B_2 \tilde{u}_i + B_1 \tilde{\omega}_{1_i} \\ \tilde{y}_{1_i} \triangleq (\lambda_i C) \tilde{x}_{1_i} \\ \tilde{\zeta}_{1_i} = C \tilde{x}_{1_i} \\ \tilde{x}_{1,c_i}^+ = A_c \tilde{x}_{1,c_i} + B_c \tilde{y}_{1_i} \\ \tilde{u}_i \triangleq C_c \tilde{x}_{1,c_i} + D_c \tilde{y}_{1_i} \end{cases} \quad (5.16)$$

where \tilde{g}_{1_i} , and $\tilde{\zeta}_{1_i}$ are respectively the *measured* and *controlled* output variables of the controlled system. Thus, we can reformulate the problem as the one finding matrices B_c , and D_c such that for $i = 1, \dots, N - 1$ the closed-loop system of (5.16) is Schur stable when $\omega_{1_i} = 0$, and to guarantee that $\|T_{\tilde{\zeta}_{1_i}\tilde{\omega}_{1_i}}(z)\|_\infty < \gamma$. A sufficient condition to prove the existence of such a solution and a relatively simple way to calculate the controller matrices are obtained by employing Theorem D.1. Indeed, if we only had to stabilize *one* generic system i of the form of (5.16), then the mentioned theorem could be directly applied as it states that, if it exists a symmetric positive definite matrix $P_i \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ such that if the given LMI conditions are satisfied, then closed-loop system (5.15) using controller (5.6),(5.7),(5.8) is such that

$$\tilde{\delta}_{1_i}^\top(k+1)P_i\tilde{\delta}_{1_i}(k+1) - \tilde{\delta}_{1_i}^\top(k)P_i\tilde{\delta}_{1_i}(k) < \gamma^2\tilde{\omega}_{1_i}^\top(k)\tilde{\omega}_{1_i}(k) - \tilde{\zeta}_{1_i}^\top(k)\tilde{\zeta}_{1_i}(k)$$

which is an equivalent way to express stability condition on system (5.15) with prescribed \mathcal{H}_∞ gain inferior to γ , (see recall on Bounded Real Lemma in Appendix D.2). It is important to stress that the mentioned LMI conditions are *affine* in the system matrices, variables and matrix P_i . We make use of this fact to provide sufficient conditions for which it exists a controller of the considered form such that the mentioned LMI is *simultaneously* verified for $i = 1, \dots, N - 1$. Since the generic eigenvalue of $\hat{\mathcal{L}}_1 : \lambda_i$ is such that $\underline{\lambda}_{\hat{\mathcal{L}}} \leq \lambda_i \leq \bar{\lambda}_{\hat{\mathcal{L}}}$, then it always exists $\rho_i \in \mathbb{R} : 0 \leq \rho_i \leq 1$ so that $\lambda_i = \rho_i \underline{\lambda}_{\hat{\mathcal{L}}} + (1 - \rho_i) \bar{\lambda}_{\hat{\mathcal{L}}}$. Notice that the systems to be stabilized, appearing in the first set of equation in (5.16), can be seen as *one* single system with an uncertain measurement matrix, whose parameter is λ_i . In other words, $C_i \triangleq \lambda_i C$, and $\exists \rho_i : C_i = \rho_i C_{low} + (1 - \rho_i) C_{up}$, where $C_{low} \triangleq \underline{\lambda}_{\hat{\mathcal{L}}} C$, and $C_{up} \triangleq \bar{\lambda}_{\hat{\mathcal{L}}} C$, i.e. it can be written as a *convex combination* of the extreme matrices C_{low} , and C_{up} . Thus, as in [Wang 2011b], we make use of classic results of robust linear control, and in particular by introducing an affine parameter dependent Lyapunov matrix $P(\rho_i) \triangleq \rho_i \underline{P} + (1 - \rho_i) \bar{P}$, where \underline{P} , \bar{P} are Lyapunov matrices solution of simultaneous LMI of Theorem D.1 written for respectively C_{low} , and C_{up} . See Appendix D.3 to recall classic results of robust control for LTI systems with parametric polytopic uncertainties. Eventually, it is easy to show that if \underline{P} , \bar{P} exist, then the controller solves the problem $\forall \lambda \in \mathbb{R} : \underline{\lambda}_{\hat{\mathcal{L}}} \leq \lambda \leq \bar{\lambda}_{\hat{\mathcal{L}}}$, and in particular for $\lambda = \lambda_i$, for $i = 1, \dots, N - 1$. Such a controller is easily found from the solution of the aforementioned LMI condition. Indeed among the LMI variables there are matrices B_c , and D_c , from which it is easy to deduce the PID gain matrices K_p , K_i , and K_d by employing relations in (5.7). ■

Remark 5.2 *The last part of the proof gives evidence of the sufficiency of the given conditions. This is due to the sufficiency of Theorem D.1 itself, as well as to the fact of having imposed the stability condition for any real value between $\underline{\lambda}_{\hat{\mathcal{L}}}$, and $\bar{\lambda}_{\hat{\mathcal{L}}}$, while just a subset is actually required.*

Remark 5.3 *Note that the mentioned LMI conditions, if satisfied, guarantee that the consensus error is minimized with respect to the disturbance. However they still have a role in determining the common function to which the agents converge, called consensus function.*

Remark 5.4 *If the mentioned LMI has a solution, then the closed-loop multi-agent system is guaranteed to be stable. In addition, having employed a PID structure for the distributed controller may suggest that consensus should be reached for any constant disturbance vector ω . Unfortunately, this is not automatically guaranteed in the MIMO case by the mentioned LMI conditions, and in this framework it is only verified a posteriori. Note that the MIMO PID controller is not block diagonal. Nonetheless, if such LMI has a solution then, according to the well-known Francis³ equations, a necessary condition for the proposed controller to reject constant exogenous signals is that $l \geq m$.*

5.2.2.3 Fast Weighted Consensus

In the previous subsection we saw how to tune a distributed controller such that the closed-loop multi-agent system has a minimized additive disturbance effect on weighted consensus reaching. This is achieved by imposing an \mathcal{H}_∞ constraint via LMIs. There might be situations though in which different LMI conditions could be used thus leading to a *different tuning* for the distributed controller. By proposing the following definition, we aim at focusing on multi-agent system *fast response* with respect to exogenous signals, such as disturbances, to achieve weighted consensus, rather than imposing an \mathcal{H}_∞ constraint.

Definition 5.2 System (5.5) is said to achieve *fast weighted consensus* with performance index $\tau \in \mathbb{R}^+$ if for $\omega = \mathbf{0}$, and any initial condition, $\lim_{k \rightarrow \infty} \|y_i/\chi_i - y_j/\chi_j\| = 0$ for $i, j = 1, \dots, N$, and $(1 - e^{-1})\%$ of consensus is achieved in a maximum number of steps equal to $\lceil \tau \rceil$.

Note that the same kind of definition can be considered for *sampled-data* systems, by saying that system (5.5) achieves fast weighted consensus with a time constant inferior to τT_s , where T_s is the system sampling time.

The following result is based on Theorem D.2 in the Appendix D.4.

Theorem 5.2 *Given the system described by (5.5), where N agents can communicate on an undirected connected graph; consider the distributed protocol of equations (5.6), (5.7), (5.8); then the systems achieve fast weighted consensus with performance index $\tau = -1/\log(\psi)$, where $\psi \in \mathbb{R} : 0 \leq \psi < 1$, if there exist two symmetric positive definite matrices $\underline{P}, \bar{P} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ such that the LMI conditions of Theorem D.2 are simultaneously satisfied for two LTI systems whose dynamic, input, and output matrices are respectively $(A, B_2, \underline{\lambda}_{\hat{C}}C)$, and $(A, B_2, \bar{\lambda}_{\hat{C}}C)$, and where the real constants (a, b) to be set in Theorem D.2 are chosen to be $(a, b) = (0, \psi)$.*

Proof: We employ the same changes of coordinates used in the proof of Theorem 5.1, yielding the $N - 1$ systems of the form of (5.15), which can be alternatively

³If a dynamic LTI controller is capable of rejecting exogenous signals that can be described as the output of an autonomous LTI exosystem (e.g. constants, ramps, etc.) from a linear system, then it necessarily satisfies Francis equations, [Francis 1976]. These are linear matrix equations built from the system matrices, the controller matrices, and the matrices describing the exosystem. A necessary condition for Francis equations to have a solution is $l \geq m$.

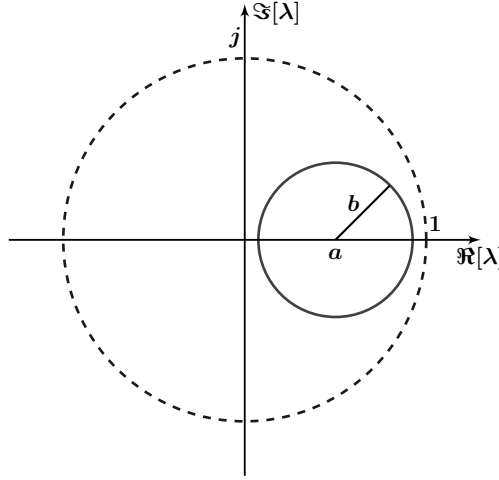


Figure 5.1: \mathcal{F}_D region in the complex plane defined via parameters (a, b) .

seen as the closed-loop form of the two systems appearing in (5.16), where matrices A_c , B_c , C_c , and D_c are defined in (5.7). Thus we can formulate the control problem as the one of finding matrices B_c and D_c such that for $i = 1, \dots, N-1$ system (5.15) is Schur stable when $\omega_{1_i} = 0$. Moreover, since we are interested in speeding up consensus reaching with respect to exogenous signals, we want to push the overall closed-loop system eigenvalues closed to zero as much as possible. For this purpose we invoke Theorem D.2, whose results can be directly applied to *one* generic system of the form of (5.16). Here it is shown that, given two constants $a \in \mathbb{R}$, and $b \in \mathbb{R}_0^+$, if there exists a symmetric positive definite matrix P_i such that the given LMI condition in the theorem is satisfied, then system (5.16) is stable with all its eigenvalues laying in the complex plane region defined by

$$\mathcal{F}_D \triangleq \{(\Re[\lambda], \Im[\lambda]) : (\Re[\lambda] + a)^2 + \Im[\lambda]^2 < b^2\}$$

where λ is the complex variable, (see Fig. 5.1). Since the LMI condition is *affine* in the system matrices, variables and P_i , exactly as done for Theorem 5.1, we employ classic results of robust control to simultaneously satisfy the mentioned LMI for (5.16), and $i = 1, \dots, N-1$, (see Appendix D.3). Eventually, in order to place the closed-loop system eigenvalues closed to 0, we set $a = 0$, and $b = \psi$, where $\psi : 0 \leq \psi < 1$. Thus, all system eigenvalues are guaranteed to have a module inferior to ψ . As a result, the system has the slowest time-constant inferior to $-T_s/\log(\psi)$. In terms of number of iterations it is easy to see that such performance is equal to a maximum value $\lceil -1/\log(\psi) \rceil$ of iterations. Eventually, from the LMI solution, the PID gain matrices are found as in the previous theorem. ■

Note that similar concluding remarks to the ones in 5.2, 5.3, and 5.4 hold in this case too.

5.2.3 Leader-follower Consensus with Time-varying Reference State

5.2.3.1 Problem Formulation

The results shown in Subsection 5.2.2 can be easily applied to solve the following leader-follower problem. Consider $N+1$ discrete-time linear agents, whose dynamics are described by

$$\begin{cases} x_0^+ = Ax_0 + B_1 u_0 \\ y_0 = Cx_0 \\ x_i^+ = Ax_i + B_2 u_i, \quad i = 1, \dots, N \\ y_i = Cx_i \end{cases} \quad (5.17)$$

where $A \in \mathbb{R}^{n \times n}$, $B_1 \in \mathbb{R}^{n \times h}$, $B_2 \in \mathbb{R}^{n \times l}$, $C \in \mathbb{R}^{m \times n}$, $x_0 \triangleq x_0(k) \in \mathbb{R}^n$ is the state of the $N+1$ agent, called *leader*, $y_0 \triangleq y_0(k) \in \mathbb{R}^m$ is its measured output and the variable on which we want the follower measured and controlled outputs y_i to converge, and $u_0 \triangleq u_0(k) \in \mathbb{R}^h$ is a time-varying unknown control acting on the leader dynamics. We additionally supposed that $l \geq m$. Concerning the remaining N follower agents, system description equivalent to (5.5) holds. The followers are assumed to communicate on an undirected connected graph whose Laplacian matrix is \mathcal{L} . The leader can pass information to a subset of followers. If agent i receives information from the leader, then we set a_{i0} to 1, and 0 otherwise. Thus we can define $\mathcal{M} \triangleq \mathcal{L} + \text{diag}(a_{10}, \dots, a_{N0})$, which is symmetric and positive definite, [Cao 2012]. Differently from the leaderless consensus case, without loss of generality we consider A to be Schur stable. The aim of the present problem is indeed not the one of stabilizing each single agent, but rather to steer the follower agents state to the leader one despite the presence of u_0 , which makes the leader dynamics time-varying. Moreover, as done for the leaderless case, we consider the general case of weighted consensus. In other words we aim at finding a distributed control law to minimize $\|y_i/\chi_i - y_0/\chi_0\|$ for $i = 1, \dots, N$, where $\chi_i, \chi_0 \in \mathbb{R}^+$. In order to accomplish such objective we aim to employ the controller of form (5.6), (5.7), where we consider a modified variable s_i to take into account the communication with the leader agent, according to

$$s_i = \sum_{j=1}^N a_{ij} \left(\frac{y_i}{\chi_i} - \frac{y_j}{\chi_j} \right) + a_{i0} \left(\frac{y_i}{\chi_i} - \frac{y_0}{\chi_0} \right) \quad (5.18)$$

Eventually, by using D , which we recall to be $D = \text{diag}(1/\chi_1, \dots, 1/\chi_N)$, we can additionally define $\hat{\mathcal{M}} \triangleq D\mathcal{M}$, which satisfies Lemmas A.1, and A.2 in Appendix A, and it has minimum and maximum positive real eigenvalues equal to $\underline{\lambda}_{\hat{\mathcal{M}}}$, and $\bar{\lambda}_{\hat{\mathcal{M}}}$ respectively.

Remark 5.5 Note that, similarly to Remark 5.1, a more general analysis could be considered by taking into account different measured and controlled output matrices. This extension is not treated in the sequel.

5.2.3.2 \mathcal{H}_∞ Weighted Leader-follower Consensus

Intuitively the proposed controller is not capable of solving the leader-follower *tracking* problem, i.e. $\lim_{k \rightarrow \infty} \|y_i/\chi_i - y_0/\chi_0\| \neq 0$ for $i = 1, \dots, N$, and for any vector signal u_0 , because it acts as an unknown exogenous signal for the overall system including the $N + 1$ agents. This is why we will focus on tuning the controller matrices B_c , and D_c such that $\|y_i/\chi_i - y_0/\chi_0\|$ is minimized for $i = 1, \dots, N$.

We provide the following

Definition 5.3 System (5.17) is said to achieve an \mathcal{H}_∞ *weighted output leader-follower consensus* with a performance index $\gamma \in \mathbb{R}^+$ if, for any initial condition, $\lim_{k \rightarrow \infty} \|y_i/\chi_i - y_0/\chi_0\| = 0$ for $i = 1, \dots, N$ when $u_0 = 0$, and the \mathcal{H}_∞ norms of the transfer matrices, for $i = 1, \dots, N$, between u_0 and $y_i/\chi_i - y_0/\chi_0$ are inferior to γ .

The following result is similar to Theorem 5.1, and it is based on Theorem D.1 in Appendix D.4.

Theorem 5.3 *Given the system described by (5.17), where N follower agents can communicate on an undirected connected graph, and one leader can communicate with a non-empty subset of followers; consider the distributed protocol of equations (5.6), (5.7), (5.18); then the systems achieve \mathcal{H}_∞ output leader-follower consensus with performance index γ if there exist two symmetric positive definite matrices $\underline{P}, \bar{P} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ such that the LMI conditions of Theorem D.1 are simultaneously satisfied for two LTI systems whose dynamic, input and measured output matrices are respectively $(A, B_2, \underline{\lambda}_{\mathcal{M}}C)$, and $(A, B_2, \bar{\lambda}_{\mathcal{M}}C)$, and they both have controlled output matrix C , and disturbance input matrix $-B_1/\chi_0$.*

Proof: The proof is similar to proof of Theorem 5.1. By defining error $e_i \triangleq x_i - x_0\chi_i/\chi_0$, $\xi_i \triangleq \text{col}(e_i, x_{c_i})$, and $\zeta_i \triangleq Ce_i$ the closed-loop system for the generic follower agent i is given by

$$\begin{cases} \xi_i^+ = \hat{A}\xi_i + \hat{B} \left(\sum_{j=1}^N a_{ij} \left(\frac{\xi_i}{\chi_i} - \frac{\xi_j}{\chi_j} \right) + a_{i0} \frac{\xi_i}{\chi_i} \right) + \chi_i \tilde{B}u_0 \\ \zeta_i = \bar{C}\xi_i \end{cases}$$

where \hat{A} , \hat{B} , \bar{C} are defined in (5.10), and $\tilde{B} \triangleq [-B_1^\top/\chi_0 \quad 0_{h \times 2l}]^\top$. Defining $\mathbf{u}_0 \triangleq \mathbf{1}_N \otimes u_0$, and $\tilde{\mathbf{u}}_0 \triangleq (D \otimes I_{\bar{n}})\tilde{\mathbf{u}}_0$, we then gather the N agent equations together

$$\begin{cases} \xi^+ = (I_N \otimes \hat{A} + \mathcal{M}D \otimes \hat{B})\xi + (I_N \otimes \tilde{B})\tilde{\mathbf{u}}_0 \\ \zeta = (I_N \otimes \bar{C})\xi \end{cases} \quad (5.19)$$

As done for the proof of Theorem 5.1, we consider the change of coordinates $\bar{\xi} \triangleq (D \otimes I_{\bar{n}})\xi$, and define $\bar{\mathbf{u}}_0 \triangleq (D \otimes I_{\bar{n}})\tilde{\mathbf{u}}_0$, $\bar{\zeta} \triangleq (D \otimes I_{\bar{n}})\zeta$, system (5.19) can be rewritten in the new coordinates as

$$\begin{cases} \bar{\xi}^+ = (I_N \otimes \hat{A} + \hat{\mathcal{M}} \otimes \hat{B})\bar{\xi} + (I_N \otimes \tilde{B})\bar{\mathbf{u}}_0 \\ \bar{\zeta} = (I_N \otimes \bar{C})\bar{\xi} \end{cases}$$

From the definition of $\hat{\mathcal{M}}$ in Subsection 5.2.3.1, there exists an orthogonal matrix $U : U^\top \hat{\mathcal{M}} U \triangleq \Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$, where $\lambda_i \in \mathbb{R} : \lambda_i > 0$ for $i = 1, \dots, N$, so that we can define the change of coordinates $\bar{\xi} \triangleq (U \otimes I_{\bar{n}})\hat{\xi}$, $\tilde{\mathbf{u}}_0 \triangleq (U \otimes I_h)\hat{\mathbf{u}}_0$, $\bar{\zeta} \triangleq (U \otimes I_m)\hat{\zeta}$. By applying similar calculation as in the previous subsections, the global system in the new coordinates is

$$\begin{cases} \hat{\xi}^+ = (I_N \otimes \hat{A} + \Lambda \otimes \hat{B}) \hat{\xi} + (I_N \otimes \tilde{B}) \hat{\mathbf{u}}_0 \\ \hat{\zeta} = (I_N \otimes \bar{C}) \hat{\xi} \end{cases} \quad (5.20)$$

As a matter of fact, it holds that

$$\|T_{\hat{\zeta}\hat{\mathbf{u}}_0}(z)\|_\infty = \|T_{\zeta\mathbf{u}_0}(z)\|_\infty$$

In other words, we can minimize the effect of \mathbf{u}_0 on the consensus error by acting on system (5.20). Splitting (5.20) in N subsystems yields the following equation for subsystem i

$$\begin{cases} \hat{\xi}_i^+ = \begin{bmatrix} (A + B_2 D_c(\lambda_i C)) & B_2 C_c \\ B_c(\lambda_i C) & A_c \end{bmatrix} \hat{\xi}_i + \begin{bmatrix} -B_1/\chi_0 \\ 0 \end{bmatrix} \hat{\mathbf{u}}_0 \\ \hat{\zeta}_i = C \hat{e}_i \end{cases} \quad (5.21)$$

where $\hat{\xi}_i \triangleq \text{col}(\hat{e}_i, \hat{x}_{c_i})$. System (5.21) can be equivalently described as the connection of the two following systems

$$\begin{cases} \hat{e}_i^+ = A \hat{e}_i + B_2 \hat{u}_i - B_1/\chi_0 \hat{\mathbf{u}}_0 \\ \hat{y}_i \triangleq (\lambda_i C) \hat{e}_i \\ \hat{\zeta}_i = C \hat{e}_i \end{cases} \quad (5.22)$$

$$\begin{cases} \hat{x}_{c_i}^+ = A_c \hat{x}_{c_i} + B_c \hat{y}_i \\ \hat{u}_i \triangleq C_c \hat{x}_{c_i} + D_c \hat{y}_i \end{cases}$$

The rest of the proof is equivalent to the last part of Theorem 5.1 proof, and it is concluded by invoking Theorem D.1, whose LMI conditions have to be simultaneously satisfied for the two systems at the vertexes of the polytope having matrices respectively $(A, B_2, \underline{\lambda}_{\hat{\mathcal{M}}} C)$, and $(A, B_2, \bar{\lambda}_{\hat{\mathcal{M}}} C)$, and same controlled output, and disturbance input matrices $C, -B_1/\chi_0$. From the solution of the aforementioned LMIs the controllers gains matrices are easily found as in the proof of Theorem 5.1. ■

Remark 5.6 *Similar remarks to the ones in 5.2, and 5.4 hold for the leader-follower consensus case too. In particular, recalling the latter, also in this case, having imposed a PID structure for the distributed controller is not sufficient to guarantee rejection of constant \mathbf{u}_0 vectors in the MIMO case. A necessary condition though is given by $l \geq m$.*

5.2.3.3 Fast Weighted Leader-follower Consensus

Similar to what done for the leaderless consensus case we can consider a different tuning for the distributed PID controller other than the one obtained from having imposed an \mathcal{H}_∞ constraint on the transfer matrix between u_0 and the consensus error. Thus, similar to Definition 5.2, we provide the following

Definition 5.4 System (5.17) is said to achieve *fast weighted leader-follower consensus* with performance index $\tau \in \mathbb{R}^+$ when $u_0 = 0$, if for any initial condition, $\lim_{k \rightarrow \infty} \|y_i/\chi_i - y_0/\chi_0\| = 0$ for $i = 1, \dots, N$, and $(1 - e^{-1})\%$ of consensus is achieved in a maximum number of steps equal to $\lceil \tau \rceil$.

The following result, similar to Theorem 5.2, is based on Theorem D.2 in the Appendix D.4.

Theorem 5.4 *Given the system described by (5.17), where N follower agents can communicate on an undirected connected graph, and one leader can communicate with a non-empty subset of followers; consider the distributed protocol of equations (5.6), (5.7), (5.18); then the systems achieve fast leader-follower consensus with performance index $\tau = -\frac{1}{\log(\psi)}$, where $\psi \in \mathbb{R} : 0 \leq \psi < 1$, if there exist two symmetric positive definite matrices $\underline{P}, \bar{P} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ such that the LMI conditions of Theorem D.2 are simultaneously satisfied for two LTI systems whose dynamic, input and output matrices are respectively $(A, B_2, \underline{\lambda}_{\mathcal{M}}C)$, and $(A, B_2, \bar{\lambda}_{\mathcal{M}}C)$, and where the real constants (a, b) to be set in Theorem D.2 are chosen to be $(a, b) = (0, \psi)$.*

Proof: The proof employs the same change of coordinates as in the previous one, so that we can restate the problem as the one of stabilizing system (5.21), for $i = 1, \dots, N$, via a PID controller whose matrices are defined in (5.7). The rest of the proof is similar to the last part of the one of Theorem 5.2. In particular, since system (5.21) can be seen as one system with uncertain parameter $\lambda_i \in [\underline{\lambda}_{\mathcal{M}}, \bar{\lambda}_{\mathcal{M}}]$, we make use of LMI conditions of Theorem D.2, and we impose them to be simultaneously satisfied for two systems at the vertexes of the polytope having matrices $(A, B_2, \underline{\lambda}_{\mathcal{M}}C)$, and $(A, B_2, \bar{\lambda}_{\mathcal{M}}C)$. The proof is concluded as for Theorem 5.2. ■
Eventually, same remark as Remark 5.6 holds in this case too.

5.2.4 Simulation Example

First of all we carry out a numerical simulation to test the \mathcal{H}_∞ leaderless consensus control. The case of fast leaderless consensus is not shown here as a more practical example will be provided in the next chapter. For the sake of simplicity we consider all consensus weights to be equal to 1, i.e. $D = I$. A more general example of $D \neq I$ is shown in the next chapter. We consider a network of 5 agents able to communicate on a connected undirected graph as show in Fig. 5.2a. Each of them

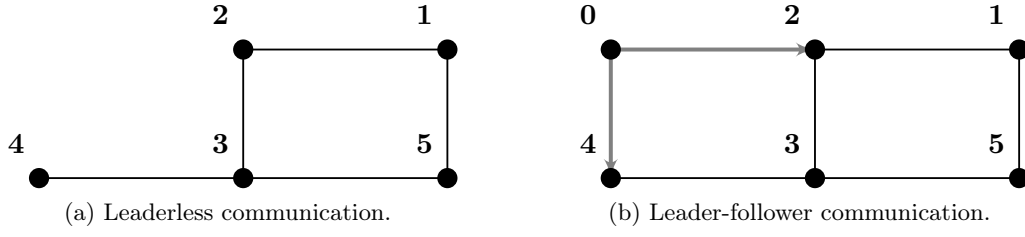
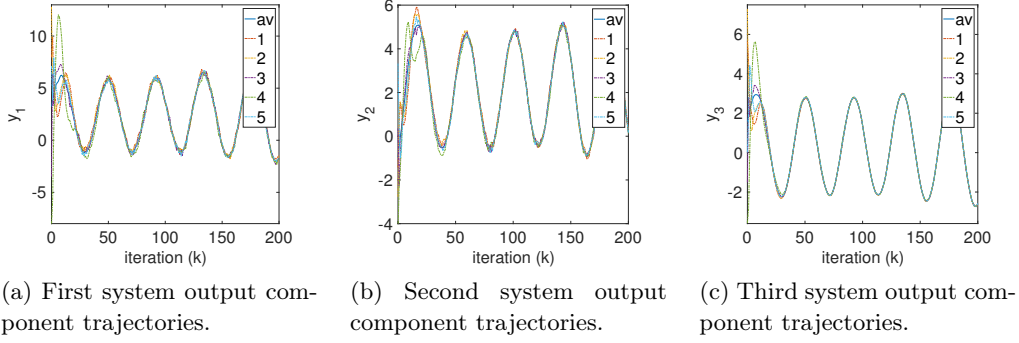
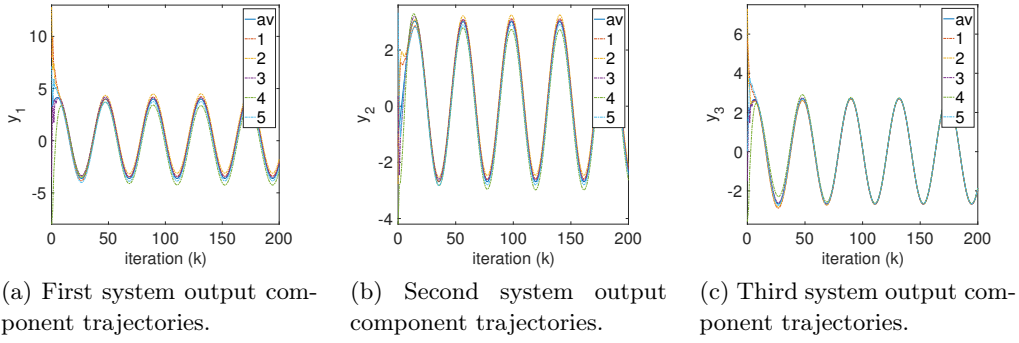
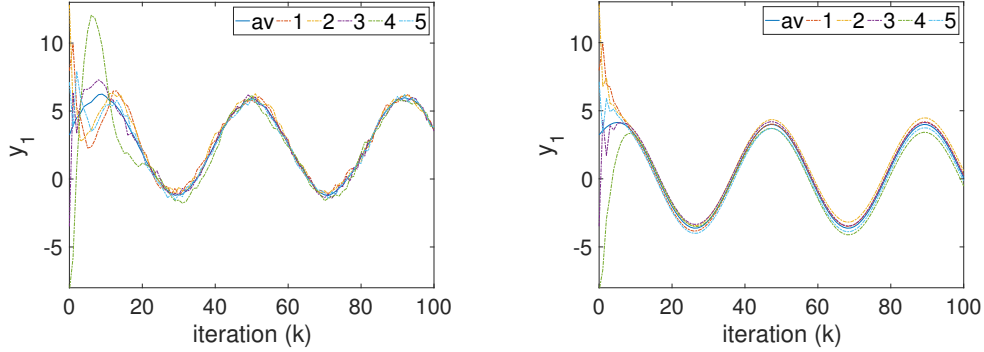


Figure 5.2: 5-agent communication graphs.

Figure 5.3: System output trajectories for each agent and network average for $\gamma = 0.9$, when $\omega_i(k) = 4\nu_i(k) + c_i$, $i = 1, \dots, 5$.Figure 5.4: \mathcal{H}_∞ tuning: system output trajectories for each agent and network average for $\gamma = 0.63$, when $\omega_i(k) = c_i$, $i = 1, \dots, 5$.

is governed by (5.5), where

$$\begin{aligned}
 A &= \begin{bmatrix} 0.8182 & 0.0452 & -0.0034 \\ 0 & 0.9888 & -0.1492 \\ 0 & 0.1492 & 0.9888 \end{bmatrix}, B_1 = \begin{bmatrix} 0.1 \\ 0.05 \\ 0 \end{bmatrix}, \\
 B_2 &= \begin{bmatrix} 1 & 0.4 & 0.2 \\ 0 & 1 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}, C = \begin{bmatrix} 1.2 & 0.8 & 1.4 \\ 1.4 & -1.2 & 0.8 \\ -0.5 & 0.7 & 1.2 \end{bmatrix}
 \end{aligned} \tag{5.23}$$



(a) First system output component trajectories for $\gamma = 0.9$, when $\omega_i(k) = 4\nu_i(k) + c_i$, $i = 1, \dots, 5$.

(b) First system output component trajectories for $\gamma = 0.63$, when $\omega_i(k) = c_i$, $i = 1, \dots, 5$.

Figure 5.5: Zoom on first system output component for $\gamma = 0.9$ and $\gamma = 0.63$.

Note that (5.23) is not Schur stable because two of its eigenvalues lay on the unit circle. Each agent is perturbed by a disturbance of the form $\omega_i(k) = 4\nu_i(k) + c_i$, where ν_i is an aleatory variable with uniform distribution of probability in $[0, 1]$, and c_i is some constant nonzero value. The PID gain matrices corresponding to an \mathcal{H}_∞ performance index $\gamma = 0.9$ are

$$K_p = \begin{bmatrix} -0.1194 & -0.0173 & 0.2030 \\ 0.1256 & -0.1354 & -0.6305 \\ -0.6705 & 0.7183 & 1.6086 \end{bmatrix}, K_i = \begin{bmatrix} 0.0465 & -0.1292 & -0.0848 \\ -0.0269 & 0.0419 & -0.0795 \\ -0.0459 & 0.0749 & 0.0571 \end{bmatrix},$$

$$K_d = \begin{bmatrix} -0.0002 & 0.0111 & 0.0102 \\ 0.0079 & -0.0085 & 0.0081 \\ 0.0117 & -0.0144 & -0.0219 \end{bmatrix}$$

Fig. 5.3 shows the 5 agents trajectories, in colored dash-dotted lines, as well as their average, in blue solid line. It is interesting to notice that for the considered system, the aforementioned γ value is *not* the minimum achievable. The minimum γ value allowing LMI feasibility is in fact, for this case, ~ 0.63 . As a matter of fact γ influences the PID gain matrices and, ultimately, how the closed-loop system respond to exogenous signals. Thus, there might be different *optimal* γ values according to different kinds of exogenous signals, i.e. values that minimize a particular signal influencing the consensus error. In our example, for the considered disturbance signals, even if both 0.9, and 0.63 γ values allow problem feasibility, i.e. in both cases the closed-loop system is Schur stable, yet a too small value of γ lets a much slower disturbance rejection. This can be seen in Fig. 5.4, which shows worse performance with respect to the previously illustrated system outputs in Fig. 5.3. Moreover, in this latter simulation we considered only $\omega_i = c_i$, where c_i are the ones used in the previous simulation, in order to evidence the system response to constant disturbances when γ is too small. For the reader's convenience we report the first

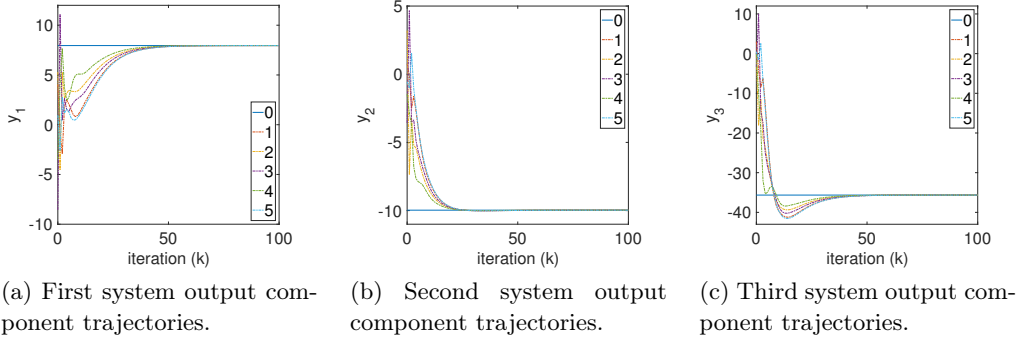


Figure 5.6: Fast consensus tuning: system output trajectories for each follower and leader for step response when $u_0 = 20$.

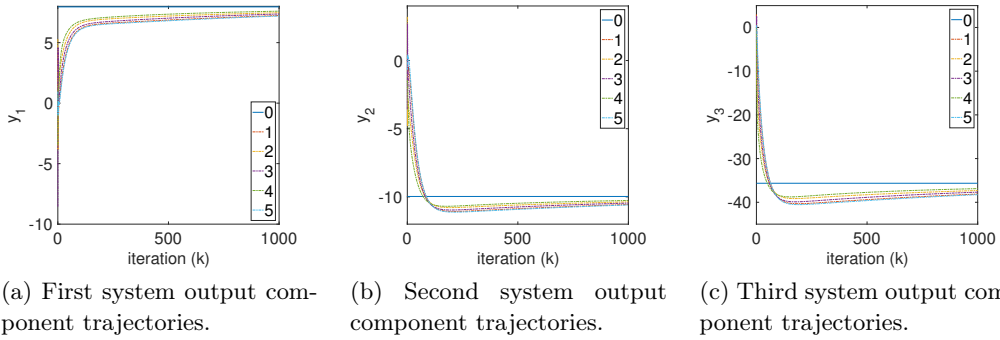


Figure 5.7: \mathcal{H}_∞ tuning: system output trajectories for each follower and leader for step response when $u_0 = 20$.

multi-agent system output component for $\gamma = 0.9$ and $\gamma = 0.63$ in a zoomed window in Fig. 5.5. Such behavior is not surprising as in the proposed controller, its tuning is performed by acting on only one degree of freedom represented by γ , through which we force the same \mathcal{H}_∞ gain to the closed-loop system for all frequencies. Despite being rather simple, this technique is generally not optimal if some prior knowledge of the exogenous signal is available to let a more fine tuning. Finally, by comparing Fig. 5.3 where a random perturbation affects the system, and Fig. 5.4 where only constant disturbances are considered, it is interesting to notice how in the former case the system disturbances distort the consensus function to which agents converge. This gives evidence of what anticipated in Remark 5.3.

As a second simulation, we compare the two proposed PID gain tuning, namely \mathcal{H}_∞ and fast consensus, for a leader-follower consensus problem. For this example we consider communication graph of Fig. 5.2b, where the added agent 0 is the leader.

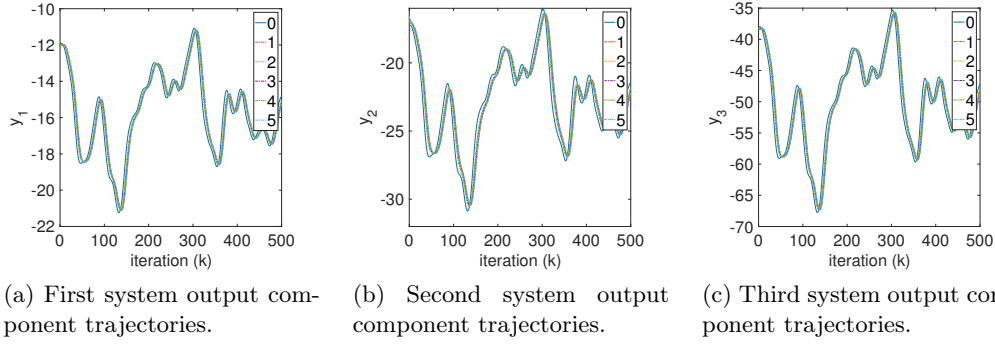


Figure 5.8: Fast consensus tuning: system output trajectories for each follower and leader for step response when u_0 is time-varying.

The system dynamics is governed by equation (5.17), where

$$A = \begin{bmatrix} 0.7711 & 0.4744 & 0.2475 \\ 0.1646 & 0.4487 & 0.1036 \\ -0.8959 & -0.8534 & -0.2198 \end{bmatrix}, B_1 = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}$$

and matrices B_2 , and C are as in (5.23). The PID controller has the following gain matrices as far as \mathcal{H}_∞ tuning is concerned

$$K_p = \begin{bmatrix} -0.1294 & 0.0542 & 0.0992 \\ 0.5981 & -0.2985 & -0.4420 \\ -1.3653 & 0.7569 & 0.9839 \end{bmatrix}, K_i = \begin{bmatrix} -0.0319 & -0.0876 & 0.0306 \\ -0.0456 & 0.0357 & 0.0022 \\ -0.0173 & 0.0265 & 0.0045 \end{bmatrix},$$

$$K_d = \begin{bmatrix} 0.0006 & -0.0026 & -0.0006 \\ -0.0011 & -0.0000 & 0.0010 \\ 0.0002 & -0.0012 & -0.0004 \end{bmatrix}$$

and gain matrices

$$K_p = \begin{bmatrix} -0.1295 & 0.0412 & 0.0863 \\ 0.6440 & -0.2598 & -0.4116 \\ -1.4194 & 0.5845 & 0.8851 \end{bmatrix}, K_i = \begin{bmatrix} 0.0091 & -0.1075 & 0.1110 \\ -0.4595 & 0.0779 & -0.2943 \\ 0.6309 & 0.1865 & 0.7436 \end{bmatrix},$$

$$K_d = \begin{bmatrix} -0.0025 & 0.0003 & 0.0011 \\ -0.0011 & 0.0037 & 0.0002 \\ 0.0008 & -0.0006 & -0.0003 \end{bmatrix}$$

as far as fast consensus tuning is concerned. In order to have some insight on how the two different kinds of tuning work on the considered leader-follower consensus problem, we firstly test them on a classic step response example. Thus u_0 is constant and set equal to 20. According to the above discussion on the choice of γ , notice that its selected value is not necessarily the lowest achievable. As a matter of fact, the lowest γ value lets worse performance in this case too. Concerning the choice of τ for fast consensus, since the only degree of freedom in the employed LMI

is parameter ψ , allowing to push the system eigenvalues towards 0, it can cause unwanted overshoots and oscillations for the sake of speed of convergence. As a result, its value needs to be chosen according to the physics of the problem in order to avoid too large stress on the controlled system. In this example, for instance, we set $\psi = 0.9$ corresponding to performance index $\tau = 9.5$, while the lowest τ value achievable is ~ 6.15 . Results are illustrated in Fig. 5.6 concerning the fast consensus, and in Fig. 5.7 for the \mathcal{H}_∞ one. From these, we are able to conclude that fast consensus tuning leads to better performance in terms of rejection of constant disturbances, as \mathcal{H}_∞ one lets the consensus error go to zero at a much slower pace. Eventually, we test fast consensus tuning performance in the case of a *time-varying* u_0 signal. Results are shown in Fig. 5.8, where followers outputs are plotted against the leader reference signal y_0 . In this case the chosen τ index is 6.15. The controller shows good performance even if, as expected, consensus error cannot go to zero in this case because of the presence of time-varying u_0 .

5.3 Conclusion

We presented a PID-like distributed protocol for general LTI MIMO discrete-time agents communicating on an undirected connected graph. By employing LMIs we showed how the controller gains can be tuned to solve two different, yet similar, problems, namely a leaderless under system disturbances and a leader-follower under time-varying reference state consensus problem. Treating the system disturbances in the \mathcal{H}_∞ framework revealed good performance, whereas a gain tuning based on fast response seems to be preferable when dealing with a leader-follower problem. Both the employed LMI conditions, i.e. the \mathcal{H}_∞ and the fast consensus approach, allow a simple MIMO PID controller tuning by acting on only one parameter, thus letting a reduction in the implementation complexity. Nonetheless, the price paid for having only one degree of freedom is that this may itself introduce an unnecessary too large level of conservatism. This was presented in the simulation example, where setting the only γ value in the \mathcal{H}_∞ design does not allow to exploit prior knowledge on system disturbances in an optimized way, and its tuning has to be performed via a trial and error approach. In future perspectives, an interesting alternative tuning technique that may bring additional degree of freedom with only minor changes to the control design approach, in the author's opinion, is presented by the \mathcal{H}_∞ loop shaping one. For instance, by adding a tunable filter on the disturbance input, prior knowledge on the system disturbance could allow focusing on the reduction of particular frequency ranges influence on the consensus error. Eventually another drawback posed by the proposed control design is that our results are based on robust control to deal with the problem of simultaneous stabilization of a given number of systems. The given conditions are sufficient and therefore conservative. In near future work it would be interesting to study less conservative conditions when treating the discrete-time consensus problem via LMI.

Wind Farm Consensus Control

Contents

6.1	Introduction	189
6.1.1	Problem Motivation	189
6.1.2	Related Works	190
6.1.3	Contribution	191
6.2	Problem Formulation	192
6.2.1	Wind Turbine Model	192
6.2.2	Problem Statement	193
6.3	Control Design	195
6.3.1	Wind Turbine Control for Deloaded Mode	195
6.3.2	Disturbance Effect and Additional Local Control Settings	198
6.3.3	Wind Farm Distributed Protocol	200
6.4	Simulations Results	202
6.5	Conclusion and Future Perspectives	209
6.5.1	Conclusion	209
6.5.2	Future Perspectives	209

6.1 Introduction

6.1.1 Problem Motivation

IN Part II we saw that when considering the aerodynamic coupling among the wind turbines, far from being a classic *modus operandi* of wind farms, the power maximization problem under system constraints is among the recent research interests in new ways to exploit the wind source. In the framework of power maximization under wake interaction, the control problem is typically handled via a first step of optimization under the approximation of static system. As shown in Chapter 4, this control step ascribes to a hierarchical control architecture in which it holds position at the higher level. Its main role is to let cooperation among the WTs by taking into account the knowledge of their physical couplings. In the case of large WFs, this can be obtained by means of distributed algorithms, as the ones shown in Chapter 3, since they typically let important benefits with respect to their centralized counterparts. As we shown in Chapter 4, the actual attainable power gain in

such hierarchical architecture can be highly affected by the system dynamics, and this validates the need for the design of efficient local WT controllers to support the optimization step. Moreover, recall that the control architecture proposed in the previous part is *decentralized* in the local WT controllers. In other words, once they receive the power references, these are *blindly* tracked without adding any further cooperation step among the WTs, which could allow taking into account disturbances and local measurements to enhance the power extraction maximization. Indeed, the optimal power references, computed according to the approximated model employed in the optimization step, are only subjected to modifications if either the free stream wind conditions or the grid requirements change. From these considerations, a resulting idea would be the one of letting the WTs local controllers exploit the existing distributed communication channels, employed at the higher control level, to add cooperation at the lower one. To move this idea forward, such cooperation control should serve as an interface between the optimization level and the local WT control, while being allowed to slightly modify the optimal power references around their original value on the basis of local measurements. The purpose of adding the aforesaid cooperation step at lower control level is mainly what motivates the work of this part.

6.1.2 Related Works

It is important to point out that wind farm power maximization can be alternatively seen as the problem of finding the optimal *power sharing* of the available wind source among the WTs. Similar power sharing problems in the wind farm control framework have been treated for instance in [Zhang 2013, Biegel 2013, Spudić 2015, Baros 2017]. In [Zhang 2013], the problem of feeding a microgrid in islanded mode via wind energy generation in order to meet the demanded power load is treated. Even though in this case the wind turbines may not belong to a common wind farm, the tackled problem is interesting from a power sharing perspective. In particular, when the overall demanded power P_d is lower than the WF available one, the power load is assigned to each WT in a simple proportional way via a prior step of WF information discovery, which let knowledge of the demanded and available power in a distributed way.

The remaining cited references all address a similar problem, namely the one of letting a WF meet the required P_d , while sharing power among the WTs in order to reduce the mechanical fatigue. These works are all based on the common assumption that the WF operates in deloaded mode in order to have the additional degree of freedom to share production in particular ways. However, in real plants this is not the general case. Both the works of [Biegel 2013], and [Spudić 2015] are based on a linearized wind turbine model. In the former one, the control problem is posed as an LQR one, where the system inputs are chosen to be the WTs power references. These have to satisfy an hard constraint in order to meet the required P_d value. This is in turns eliminated by employing a suitable linear relationship among the power references that parametrizes the feasible set. The overall control problem is then

solved by imposing a distributed structure to the control law. In [Spudić 2015], differently from [Biegel 2013], the optimization problem yielding the control law is solved in a receding horizon fashion via MPC technique. This is implemented in a distributed way by employing approaches that we previously mentioned in Section 3.3 of Chapter 3, namely ADMM and dual decomposition technique. In both the latter two references, fatigue reduction is achieved by minimizing a quadratic cost function in the system variables, which in turns are related to the system mechanical stress. Eventually, in [Baros 2017], the power constraint according to which the sum of WT power productions has to meet P_d is coupled with the pre-assignment of a fair load sharing condition of the form of

$$\frac{P_i}{P_i^o} = \frac{P_j}{P_j^o}, \quad i, j = 1, \dots, N$$

where N is the number of WTs in the wind farm and, P_i and P_i^o the generic i -th WT power and MPPT one respectively. The aforesaid relationship makes unique the combination of P_i , $i = 1, \dots, N$ yielding P_d . Power sharing is then solved via leader-follower consensus control in which a selected WT, namely the leader, has knowledge of P_d value via communication with the grid operator.

In all the mentioned works the aerodynamic coupling among the WTs is not taken into account in the control design. With the exception of the work in [Biegel 2013], the wake effect is not considered in the validation step either.

6.1.3 Contribution

Our purpose is to propose a distributed control to let proper power sharing among the WTs in order to maximize the power generation. To the author's knowledge such distributed control framework has never been applied to the problem here addressed, and, despite having some common ideas concerning power sharing, it substantially differs from the mentioned references.

Our contribution is two-fold. Firstly, an asymptotic output tracking control is applied at the WT level via FL to let the distributed problem be treated in the linear systems framework. Moreover, this control step has to allow the WT to track a general deloaded power reference, which is a necessary condition for wind farm power maximization as discussed in Chapters 2 and 4. Although the local controller that we presented in Chapter 2 allows the WT to work in the whole operating envelope while tracking general power references, because of the MPC control step, it does not particularly simplify the design of a distributed control law, which is the aim of this chapter. The WT local controller here considered is in fact different from the aforementioned one, and it is based on the work of [Boukhezzer 2005]. Here, the authors propose an AOT-based controller for variable-speed *fixed*-pitch WTs to perform power regulation in full load operation mode, i.e. pitch control is not employed to let power tracking. This is why we modify the control design shown therein, by proposing an approximated FL technique to let WTs track general power references at low wind speed, as in this wind speed range deloading is performed for

WF power maximization purposes.

As a second contribution, under the assumption of discrete-time communication among the WTs, we first employ a leaderless distributed PID-like control technique shown in the previous chapter to force the system to stabilize around the optimal power sharing set points provided by the higher optimization level, under zero-mean wind disturbances. The set power sharing problem has some common ideas with the work of [Schiffer 2016], where weighted consensus control is employed to solve the reactive power sharing problem in inverter-based microgrids via distributed voltage control. Secondly, we show how, under power reference errors, the optimal WF power sharing can be restored via a leader-follower technique. In both the leaderless and the leader-follower cases the PID structure is justified as a simpler P-like protocol would not allow a satisfactory disturbance rejection, as the dynamics of the agents, i.e. the AOT-controlled WTs, are described by LTI systems with no particular structure. Some of the results reported in the sequel have been object of our work of [Gionfra 2017b].

6.2 Problem Formulation

6.2.1 Wind Turbine Model

The wind turbine model has been extensively presented in Section 2.2 of Chapter 2, where a two-mass model was employed for control design and validation. Here, for the sake of simplicity, we consider the one-mass model representation for the local controller design. For the reader convenience we report in the following the main system variables and equations. For further details refer to the aforesaid chapter. Recall that the extracted power from the wind has the form of

$$P_r = \frac{1}{2} \rho \pi R^2 v^3 C_p(\lambda, \vartheta) \quad (6.1)$$

where, once again, we employ the CART power coefficient shown in Fig. 2.2 in Chapter 2 for control validation. As far as the control design is concerned though, differently from what done in the FL+MPC design in Chapter 2, here we consider a *polynomial* approximation of C_p . The differential equation describing the dynamics of the WT rotor speed ω_r is

$$J_t \dot{\omega}_r = T_r - K_t \omega_r - T_{em} \quad (6.2)$$

where $T_r = P_r/\omega_r$, J_t and K_t are respectively the total equivalent shaft inertia and mass friction on the slow shaft side, and $T_{em} = n_g T_g$, being T_g the electromagnetic torque of the generator and n_g the gear box ratio. Moreover we neglect the first order equation describing the dynamics of T_g (recall equation (2.11) in Chapter 2), while, as done in Chapter 2, we consider the one used to describe the pitch angle dynamics. In addition, we consider the pitch actuator system to be endowed with a saturation system that can be modeled via a *sigmoid* function $\sigma : \mathbb{R} \rightarrow [\vartheta_{min}, \vartheta_{max}]$.

The overall WT dynamics can be described by

$$\begin{cases} \tau_{\vartheta} \dot{\vartheta}_s = -\vartheta_s + \vartheta_r \\ \vartheta = \sigma(\vartheta_s) \\ J_t \dot{\omega}_r = \frac{P_r(\omega_r, \vartheta, v)}{\omega_r} - K_t \omega_r - T_{em} \\ P = T_{em} \omega_r \end{cases} \quad (6.3)$$

where P is the electric power delivered to the grid if we neglect the generator losses. The system inputs are T_{em} , and ϑ_r , while the wind speed v acts as a disturbance. The feasible domain is $\mathcal{X} \triangleq \{(\omega_r, \vartheta) \in \mathbb{R}^2 : \omega_r \in [\omega_{r,min}, \omega_{r,max}], \vartheta \in [\vartheta_{min}, \vartheta_{max}]\}$.

6.2.2 Problem Statement

In the sequel, for consistency of notation, we add the index i to the WT variables described in the previous subsection when referring to turbine i variables, and we drop it when the results hold for any WT. Recall that when controlling a WT in classic MPPT mode at low wind speed, the main control objective is to let the WT track the constant tip-speed ratio

$$\lambda^o \triangleq \arg \max_{\lambda} P_r(v, \vartheta^o, \lambda) = \arg \max_{\lambda} C_p(\lambda, \vartheta^o)$$

being ϑ^o the constant value for which the maximum power extraction is attained for any admissible wind speed value. For the considered CART turbine $\lambda^o \cong 8$. It has been shown in the previous part how, when considering the wake effect in the optimization step of a farm of N WTs, the optimal value of C_p related to the generic turbine i is such that $C_{p,i}^* \leq C_p^o$, where $C_{p,i}^*$ is the power coefficient related to the optimal axial induction factor α_i^* , and $C_p^o = C_p(\lambda^o, \vartheta^o)$. As a matter of fact, this implies that for any low wind speed value v_i , a turbine i should track an optimal deloaded power reference $P_i^*(v_i)$, i.e. satisfying $P_i^*(v_i) \leq P_i^o(v_i)$, where P_i^o is the MPPT power value. According to the wake model employed in Chapter 4, as well as the following

Assumption 6.1 *For $i = 1, \dots, N$, v_i is such that $P_i^*(v_i) < P_n$, where P_n is the WT nominal power.*

Then, the static optimization step needs to be run only when the wind direction changes, as optimal values $C_{p,i}^*$ do not depend on the wind speed value. Moreover, we consider the following

Assumption 6.2 *The average wind direction is considered to be slowly varying with respect to the system dynamics. Thus, it is considered to be constant.*

We can now formulate the control problem in two subproblems, the first of which being

Problem 6.1 Consider the system described by (6.3). Given an effective wind speed signal $v(t)$, and a time-varying reference trajectory $P^{ref}(t)$, verifying $P^{ref}(t) \leq P^o(t) \forall t \geq 0$, find the signals $(\vartheta_r(t), T_g(t)) \forall t \geq 0$ such that $\lim_{t \rightarrow \infty} |P^{ref}(t) - P(t)| = 0$ for every initial condition $(\omega_r(0), \vartheta(0)) \in \mathcal{X}$ such that $P(0) \leq P^o(0)$.

Let us now assume that each local WT controller can measure, or estimate, the effective wind speed $v_{m,i}(t)$ such that

$$v_i(t) = v_{m,i}(t) + v_{d,i}(t), \quad \forall t \geq 0$$

Thus $v_{d,i}$ represents a nonmeasurable time varying disturbance for turbine i .

Assumption 6.3 We consider small zero-mean disturbances $v_{d,i}$ with respect to $v_{m,i}$, and slowly-varying with respect to the dynamics of (6.3).

Each WT can compute the optimal power reference, as described in Section 4.5 of Chapter 4, from its maximum available power P_i^o , and optimal axial induction factor, according to

$$P_i^{fw} \triangleq P_i(v_{m,i}) = \frac{C_{p,i}^*}{C_p^o} P_i^o(v_{m,i}) \quad (6.4)$$

which is *optimal* in nominal conditions, i.e. $P_i^*(v_i) = \frac{C_{p,i}^*}{C_p^o} P_i^o(v_i)$, when $v_{d,i} \equiv 0$.

We can additionally require the WTs to meet an optimal *relative* power sharing condition given by

$$\frac{P_i}{\chi_i} = \frac{P_j}{\chi_j} \quad i, j = 1, \dots, N \quad (6.5)$$

Indeed, by naming P_∞^o the maximum power that a WT can extract from the wind if there is no wake effect, from (6.4) we have that

$$\begin{aligned} \frac{P_i}{C_{p,i}^*} &= \frac{P_i^o}{C_p^o} = \nu_i \frac{P_\infty^o}{C_p^o}, \quad i = 1, \dots, N \\ \frac{P_i}{\nu_i C_{p,i}^*} &= \frac{P_j}{\nu_j C_{p,j}^*} \quad i, j = 1, \dots, N \end{aligned}$$

We name $\chi_i \triangleq \nu_i C_{p,i}^* \in \mathbb{R}^+$, and where

$$\nu_i = \frac{P_i^o}{P_\infty^o} = \left(\frac{v_i}{v_\infty} \right)^3 \quad i = 1, \dots, N$$

are constant values for any value of v_∞ according to Assumptions 6.1, and 6.2, being v_∞ the free stream wind speed. In the sequel, in order to make the difference with condition (6.5) clear, we will refer to (6.4) as the *absolute* power reference. Once again, this is optimal if the corresponding $v_{d,i} = 0$. Despite being redundant information with respect to (6.4) in nominal conditions, as it will be made clear in the sequel, condition (6.5) provides additional signals that can be exploited when the system is subject to disturbances. We can now state the second subproblem.

Problem 6.2 Given N identical WTs, allowed to communicate on an undirected connected graph \mathcal{G}_c ; given optimal values $C_{p,i}^*$, and χ_i , $i = 1, \dots, N$; find $P_i^{ref}(t) \forall t \geq 0$, $i = \dots, N$ such that each P_i tracks (6.4), while minimizing the error $|P_i/\chi_i - P_j/\chi_j|$, $i, j = \dots, N$, under the presence of $v_{d,i}(t)$.

The idea behind Problem 6.2 is to exploit additional information concerning optimal WTs *relative* power values in order to even out the system disturbances on the optimal power sharing defined by the optimization step.

6.3 Control Design

6.3.1 Wind Turbine Control for Deloaded Mode

According to the optimization step, it turns out that every WT causing a reduction of available wind power of another one, is very likely to be subject to an optimal C_p value such that $C_{p,i}^* < C_p^o$, i.e. *strictly* inferior. Thus, WTs whose $C_{p,i}^*$ verifies $C_{p,i}^* = C_p^o$ should simply perform classic MPPT regardless the disturbances of the system and the other WTs operating points, and they can be controlled with classic local controllers. In the sequel we only consider WTs that have to be *strictly* deloaded with respect to their P_i^o .

Following [Boukhezzer 2005], the local control is composed of a first loop to control ω_r . We impose a first order dynamics to the rotor speed tracking error $\varepsilon_\omega \triangleq \omega^{ref} - \omega_r$, i.e.

$$\dot{\varepsilon}_\omega + a_0 \varepsilon_\omega = 0$$

by choosing $a_0 \in \mathbb{R}^+$. If we name $w \triangleq a_0 \omega^{ref} + \dot{\omega}^{ref}$, this is attained by using (6.2) as

$$T_{em} = T_r - (K_t - a_0 J_t) \omega_r - J_t w \quad (6.6)$$

Differently from [Boukhezzer 2005], we choose to regulate the power output P by acting on the pitch angle. We impose a first order dynamics to the electric power tracking error $\varepsilon_p \triangleq P^{ref} - P$, i.e.

$$\dot{\varepsilon}_p + b_0 \varepsilon_p = 0 \quad (6.7)$$

by choosing $b_0 \in \mathbb{R}^+$. This is attained via FL on (6.3) by choosing the feedback linearizing input

$$\begin{aligned} \vartheta_r = \frac{1}{\beta(\omega_r, \vartheta_s, v)} & \left(\dot{P}^{ref} - \omega_r \frac{\partial T_r}{\partial v} \dot{v} + \frac{\omega_r}{\tau_\vartheta} \frac{\partial T_r}{\partial \vartheta} \frac{d\sigma}{d\vartheta_s} \vartheta_s + J_t \dot{w} \omega_r \right. \\ & \left. + \left(2(K_t - a_0 J_t) \omega_r - T_r + J_t w - \omega_r \frac{\partial T_r}{\partial \omega_r} \right) (-a_0 \omega_r + w) + b_0 \varepsilon_p \right) \end{aligned} \quad (6.8)$$

where $\frac{\partial T_r}{\partial \omega_r}$, $\frac{\partial T_r}{\partial \vartheta}$, and $\frac{\partial T_r}{\partial v}$ are functions of (ω_r, ϑ, v) , and where $\beta(\omega_r, \vartheta_s, v) \triangleq \frac{\omega_r}{\tau_\vartheta} \frac{d\sigma}{d\vartheta_s} \frac{\partial T_r}{\partial \vartheta}$.

As pointed out in Chapter 2, there exist points in which $\beta = 0$, called *singular points*, i.e. points in which (6.8), feedback linearizing input with respect to output P , is not defined. These points are determined by the solution of

$$\frac{\partial C_q}{\partial \vartheta}(\omega_r, \vartheta, v) = \frac{\partial C_q}{\partial \vartheta}(\lambda, \vartheta) = 0$$

being $C_q \triangleq \frac{C_p}{\lambda}$, and because

$$\beta(\omega_r, \vartheta_s, v) = \frac{\omega_r}{2\tau_\vartheta} \rho \pi R^3 v^2 \frac{d\sigma}{d\vartheta_s} \frac{\partial C_q}{\partial \vartheta}(\omega_r, \sigma(\vartheta_s), v) \cong \frac{\omega_r}{2\tau_\vartheta} \rho \pi R^3 v^2 \frac{\partial C_q}{\partial \vartheta}(\omega_r, \vartheta, v)$$

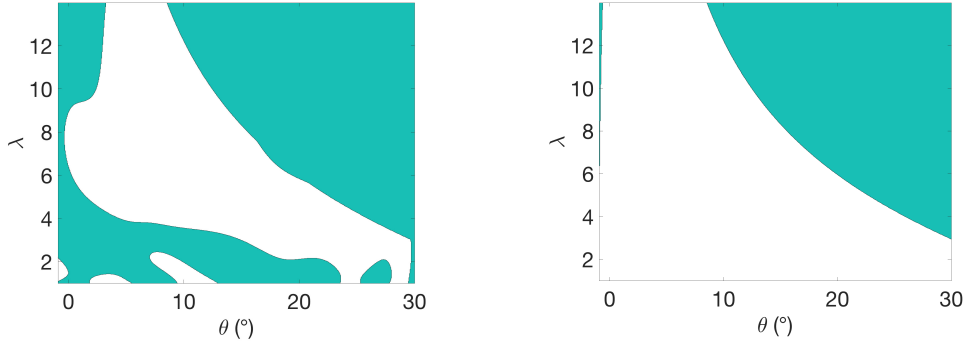
in the domain of interest of ϑ , and $\omega_r, v > 0$. In Fig. 6.1a the white area represents $\Lambda = \{(\lambda, \vartheta) : (\omega_r, \vartheta) \in \mathcal{X} \wedge \beta < 0\}$. If ω^{ref} is chosen to let λ be in a neighborhood of λ^o , and $\vartheta > \vartheta^o$ in order to deload the WT, where $\vartheta^o \simeq 0^\circ$ for CART turbine, then it is clear that β is *negative*-valued in the points of functioning of interest. In order to ensure that the trajectories of the closed loop system, defined by (6.3), (6.6), (6.8), do not pass through singular points, differently from what done in Chapter 2, we consider a *modified* FL function for ϑ_r , by replacing the β function appearing in (6.8) with

$$\begin{aligned} \hat{\beta} &\triangleq \frac{\omega_r}{2\tau_\vartheta} \rho \pi R^3 v^2 \frac{d\sigma}{d\vartheta_s} \left(\frac{\partial C_q}{\partial \vartheta}(\lambda, \vartheta) - \varepsilon(\lambda, \vartheta) \right) \\ \varepsilon(\lambda, \vartheta) &\triangleq \begin{cases} \varrho \max \left\{ \frac{\partial C_q}{\partial \vartheta}(\lambda, \vartheta), 0 \right\} & \text{if } \frac{\partial C_q}{\partial \vartheta}(\lambda, \vartheta) \neq 0 \\ \epsilon_1 & \text{otherwise} \end{cases} \end{aligned} \quad (6.9)$$

where ϵ_1 is a small positive value, and $\varrho > 1$ is a tunable parameter to let some margin to have $\hat{\beta}$ negative-valued in the system trajectories. Thus we obtain an expanded negative-valued area $\hat{\Lambda} = \{(\lambda, \vartheta) : (\omega_r, \vartheta) \in \mathcal{X} \wedge \hat{\beta} < 0\}$, shown in Fig. 6.1b. The idea is thus to perform an approximated FL *only* when the system trajectories come close to a singular point. Clearly, in this case, the chosen ϑ_r no longer guarantees satisfaction of (6.7). Nonetheless, under proper choice of ω^{ref} , and de-loading technique, approximation (6.9) may occur only during transients. We can summarize the results in this subsection by stating the following

Theorem 6.1 *Given system (6.3), controlled via (6.6), and (6.8), where the β function is substituted with (6.9). For any initial condition $(\omega_r(0), \vartheta(0)) \in \hat{\Lambda}$, the system trajectories are bounded if parameters b_0 , ϵ_1 , and ϱ are chosen such that $\epsilon_1 > 0$ is sufficiently small, $\varrho > 1$, and $b_0 > -\frac{\varrho}{1-\varrho}$. In addition, if $\exists \bar{t} \geq 0$ such that $\frac{\partial C_q}{\partial \vartheta}(\lambda(t), \vartheta(t)) < 0 \forall t \geq \bar{t}$, then $\lim_{t \rightarrow \infty} |P^{ref}(t) - P(t)| = 0$.*

Proof: First of all, initial conditions in $\hat{\Lambda}$ imply $\hat{\beta}(0) < 0$, then $\epsilon_1 > 0$, $\varrho > 1$ allow $\hat{\beta}(t) < 0 \forall t \geq 0$. In particular $\hat{\beta}(t) \neq 0 \forall t \geq 0$, thus (6.8) is well-defined. Note



(a) Λ , white area, set of (λ, ϑ) such that $\beta(\lambda, \vartheta) < 0$. (b) $\hat{\Lambda}$, white area, set of (λ, ϑ) such that $\hat{\beta}(\lambda, \vartheta) < 0$.

Figure 6.1: Singular points with and without β approximation.

that initial conditions considered in Problem 6.1 satisfy $\hat{\beta}(0) < 0$, belonging to $\hat{\Lambda}$ shown in Fig. 6.1b. The system dynamics in closed loop is given by

$$\dot{\varepsilon}_p = \left(-b_0 + 1 - \frac{\beta}{\beta + \beta_\varepsilon} \right) \varepsilon_p + \left(1 - \frac{\beta}{\beta + \beta_\varepsilon} \right) \varphi(\varsigma) \quad (6.10)$$

where we named $\beta_\varepsilon \triangleq \hat{\beta} - \beta$, and $\varphi(\varsigma)$ the function composed of all the terms appearing in the right factor of (6.8) deprived of the term $b_0 \varepsilon_p$, and being $\varsigma \triangleq (\omega_r, v, \vartheta_s, \dot{v}, w, \dot{w}, \dot{P}^{ref})$, i.e.

$$\begin{aligned} \varphi(\varsigma) \triangleq & \dot{P}^{ref} - \omega_r \frac{\partial T_r}{\partial v} \dot{v} + \frac{\omega_r}{\tau_\vartheta} \frac{\partial T_r}{\partial \vartheta} \frac{d\sigma}{d\vartheta_s} \vartheta_s + J_t \dot{w} \omega_r \\ & + \left(2(K_t - a_0 J_t) \omega_r - T_r + J_t w - \omega_r \frac{\partial T_r}{\partial \omega_r} \right) (-a_0 \omega_r + w) \end{aligned}$$

The term $\left(1 - \frac{\beta}{\beta + \beta_\varepsilon} \right) \varphi(\varsigma)$ is bounded in the trajectories thanks to the choice of β_ε , and being φ a continuous function on a compact set. This is compact because the wind is limited, $w, \dot{w}, \dot{P}^{ref}$ are chosen to be so, ω_r is bounded thanks to (6.6), and term $\frac{d\sigma}{d\vartheta_s} \vartheta_s$ is bounded. Thus it will be considered as a bounded input of (6.10) to simplify the analysis. Finally system (6.10) with $\varphi(\varsigma) \equiv 0$, given by

$$\dot{\varepsilon}_p = \begin{cases} \left(-b_0 + 1 - \frac{1}{\epsilon_1} \right) \varepsilon_p & \text{if } \beta = 0 \\ \left(-b_0 + 1 - \frac{1}{1 - \varrho} \right) \varepsilon_p & \text{if } \beta > 0 \\ -b_0 \varepsilon_p & \text{otherwise} \end{cases} \quad (6.11)$$

is stable if, for instance, we choose $b_0 > -\frac{\varrho}{1 - \varrho}$, and $\epsilon_1 < 1$. This can be proved by choosing $V(\varepsilon_p) \triangleq \frac{1}{2} \varepsilon_p^2$ as a common Lyapunov function for the family of sys-

tems (6.11), (see [Liberzon 2012]). Eventually, if for some $\bar{t} \geq 0 : \frac{\partial C_q}{\partial \vartheta} < 0 \ \forall t \geq \bar{t}$, then (6.10) reduces to $\dot{\varepsilon}_p = -b_0 \varepsilon_p$, thus $P \rightarrow P^{ref}$ for $t \rightarrow \infty$. ■

Remark 6.1 Concerning ω^{ref} , we make the choice to use the MPPT signal $\omega_r^o = \frac{\lambda^o v}{R}$ sufficiently filtered of its high frequency components. There are different motivations to support this choice. First of all, if v varies rapidly so it does ω_r^o , then if we consider $\omega^{ref} = \omega_r^o$, its variation would directly effect ϑ_r via (6.8), and in turns ϑ . This fact risks to make ϑ hit the saturation constraints of the sigmoid function, and more in general, to not let the constraints on $\dot{\vartheta}$ be respected, as in this framework they are only verified a posteriori. Secondly, if ω^{ref} varies too rapidly, by empirical results it turns out that the closed-loop system trajectories are more likely to approach singular points, letting the activation of $\varepsilon(\lambda, \vartheta)$ defined in (6.9), and not allowing satisfaction of (6.7). On the other hand, filtering ω_r^o let (6.8) be defined, i.e. it fulfils the requirement of tracking the desired deloaded power reference. The physical explanation of this fact is that for a given deloaded P^{ref} there exist infinite pairs $(\omega_r, \vartheta) \in \mathcal{X}$ that let a WT track it, (see Section 2.4 of Chapter 2), and by filtering ω_r^o , we are simply considering another possible choice of couples (ω_r, ϑ) than the one in which $\omega_r = \omega_r^o$.

Remark 6.2 The chosen deloading approach is similar to the deloading via pitch control technique shown in Section 2.4 of Chapter 2. Differently from the one employed in the FL+MPC design, however this technique does not allow any additional energy storage in the rotating masses.

Remark 6.3 The particular choice of ω_r to deload the WT is made a priori, and this allows to avoid reference computation for ω_r by solving an optimization problem, as done for FL+MPC control scheme of Chapter 2 in problem (2.16).

Remark 6.4 The described control approach is conceived for a WT working in strict deloaded mode. Even though a priori it could be also employed for MPPT control, the closed-loop system would not present a satisfactory tracking of P^o power reference, i.e. the MPPT one. This is basically due to the fact that it is impossible to track ω_r^o , letting a mechanical power equal to P^o , while simultaneously reproducing the same power value P^o as electric power output.

6.3.2 Disturbance Effect and Additional Local Control Settings

From now on we carry out the analysis under the following

Assumption 6.4 Trajectories of the closed-loop system described by (6.3), (6.6), (6.8) verify $\frac{\partial C_q}{\partial \vartheta} < 0$.

As previously mentioned, we assume that turbine i local controller is able to measure $v_{m,i}$ such that $v_i = v_{m,i} + v_{d,i}$. The effect of $v_{d,i}$ on the closed loop dynamics can be thus approximated as

$$\dot{\varepsilon}_{p,i} = -b_0 \varepsilon_{p,i} + \mu_1(\hat{\zeta}_i) v_{d,i} + \mu_2(\hat{\zeta}_i) v_{d,i}^2 + \mu_3(\hat{\zeta}_i) \dot{v}_{d,i} \quad (6.12)$$

obtained via first order Taylor expansion of the functions depending on v_i , in a neighborhood of $v_{m,i}$, e.g. $T_r(v_i) \cong T_r(v_{m,i}) + \frac{\partial T_r}{\partial v}(v_{m,i})v_{d,i}$, and where $\hat{\zeta}_i \triangleq (\omega_{r,i}, \vartheta_i, v_{m,i}, \dot{v}_{m,i})$. Functions μ_1, μ_2, μ_3 are reported in Appendix B.4. According to Assumption 6.3 we neglect $\mu_3(\hat{\zeta}_i)\dot{v}_{d,i}$. Moreover, by numerical simulation, the contribution of term $\mu_2 v_{d,i}^2$ can be neglected with respect to $\mu_1 v_{d,i}$. On the compact set on which μ_1 is defined, the function satisfies $\mu_{1,min} \leq \mu_1 \leq 0$, thus in the sequel we treat μ_1 as a parametric uncertainty, and we drop its dependency on $\hat{\zeta}_i$ for ease of notation. Being interested in a discrete-time communication set-up among the WTs we shall consider system (6.12) discretized at sampling time T_s , given by

$$\begin{bmatrix} \xi_i(k+1) \\ P_i(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -1 & (1 - T_s b_0) \end{bmatrix} \begin{bmatrix} \xi_i(k) \\ P_i(k) \end{bmatrix} + \begin{bmatrix} 1 \\ (1 + T_s b_0) \end{bmatrix} P_i^{ref}(k) + \begin{bmatrix} 0 \\ \mu_1 T_s \end{bmatrix} v_{d,i}(k) \quad (6.13)$$

where we used Euler method, we approximated $\dot{P}_i^{ref} \cong \frac{(P_i^{ref}(k) - P_i^{ref}(k-1))}{T_s}$, and we named $\xi_i(k) \triangleq P_i^{ref}(k-1)$.

System (6.13) is required to track the optimal power reference provided by the higher optimization step according to (6.4). For the moment, let us assume that P_i^{fw} , $i = 1, \dots, N$ are not affected by wind measurement error, i.e. $P_i^{fw}(v_i)$. Because of the presence of $v_{d,i}$ in (6.13), affecting the controlled WT dynamics, setting $P_i^{ref} = P_i^{fw}$ does not guarantee P_i to asymptotically converge to P_i^{fw} , nor satisfaction of relationship (6.5), describing the optimal relative power sharing values among the WTs. Under the assumption of communicating WTs on an undirected connected graph \mathcal{G}_c , whose Laplacian matrix is \mathcal{L} , one could think to exploit the leaderless PID-like distributed protocols developed in Chapter 5 to reduce the effect of $v_{d,i}$ on weighted consensus (6.5) among the WTs, by acting on P_i^{ref} . Nonetheless, even if the distributed PID reveals good performance in rejecting system disturbances on the consensus error, as pointed out in Remark 5.3 in Chapter 5, still the consensus function to which the system converges depends on the disturbance signals. As a result, even if *relative* distances on power values according to (6.5) are respected thanks to consensus control, there is no general guarantee for the power values to reach P_i^{fw} . This is why before continuing our analysis on WF consensus control, we introduce an additional *local* PI loop to system (6.13) to let convergence to the absolute power value P_i^{fw} . Note that the two integral actions, namely the internal loop one controlling P_i to P_i^{fw} , and the distributed control one controlling P_i to satisfy condition (6.5), are not in contradiction if P_i^{fw} are measurement error free. Indeed, by naming $D \triangleq \text{diag}(\frac{1}{\chi_1}, \dots, \frac{1}{\chi_N})$ as done in Chapter 5, and $P^{fw} \triangleq \text{col}(P_1^{fw}, \dots, P_N^{fw})$, then by construction, $\mathcal{L}DP^{fw} = \mathbf{0}$, which is exactly the weighted consensus condition seek by the distributed protocol. As a consequence, in such case of perfect information on P_i^{fw} , the distributed consensus is not required to let condition (6.5) be satisfied, as it is already ensured by the internal PI. However, even in this case, consensus control can be employed to enhance closed-loop performance.

An example of the fact will be provided in Section 6.4.

The above discussion holds for the case in which the wind disturbance $v_{d,i}$ only affects the system equations as shown in (6.13). However, $v_{d,i}$ has also a role in the computation of P_i^{fw} , since in reality we have $P_i^{fw}(v_{m,i}) \neq P_i^*(v_i)$. In this case, condition $\mathcal{LDP}^{fw} = \mathbf{0}$ generally does not hold, i.e. the power references provided to the local WT controller may not let satisfaction of the optimal relative power sharing. Under these circumstances, the distributed PID does have a role in forcing weighted consensus among the WTs. Moreover, the two integral actions may come to a conflict. In the following we decide to give priority to consensus seeking by allowing the distributed control action modify the local error $P_i^{fw} - P_i$. This is achieved by considering the dashed arrow shown in the control scheme of Fig. 6.2, where the overall WT control is illustrated. The idea behind this choice is that zero-mean disturbances on the local power references P_i^{fw} can be globally evened out by enforcing relative power distances among the network of wind turbines.

By naming K_I^l , and K_P^l respectively the integral, and proportional gains of the PI, we can write (6.13) in closed-loop as $x_i(k+1) = Ax_i(k) + B_2u_i(k) + B_{fw}P_i^{fw}(k) + B_wv_{d,i}(k)$, where

$$\begin{aligned} A &\triangleq \begin{bmatrix} 1 & 0 & -K_I^l T_s \\ 1 & 0 & -K_P^l \\ (1+T_s b_0) & -1 & (1-T_s b_0 - K_P^l(1+T_s b_0)) \end{bmatrix}, B_w \triangleq \begin{bmatrix} 0 \\ 0 \\ \mu_1 T_s \end{bmatrix} \\ B_2 &\triangleq \begin{bmatrix} K_I^l T_s \\ 1 \\ (1+T_s b_0) \end{bmatrix}, B_{fw} \triangleq \begin{bmatrix} K_I^l T_s \\ K_P^l \\ (1+T_s b_0)K_P^l \end{bmatrix} \end{aligned} \quad (6.14)$$

and where we named $x_i \triangleq \text{col}(\varsigma_i, \xi_i, P_i)$, being ς_i the state of the local integral action, P_i^{fw} a forward signal, and u_i is left as a degree of freedom to let distributed control. Note that other choices for the introduction of u_i in the internal WT control loop would have been possible.

6.3.3 Wind Farm Distributed Protocol

Leaderless Consensus

Consensus control over the WF can be employed to let satisfaction of relationship (6.5) over a set of N controlled WTs of the form of (6.14). This can be done by making use of the tools concerning leaderless consensus shown in Chapter 5. Both Theorems 5.1, and 5.2 can be applied. This is simply obtained by choosing as A , B_2 the matrices of (6.14), $C = [0 \ 0 \ 1]$, i.e. P_i is the measured and controlled output, and $B_1 = [B_{fw} \ B_w]$, i.e. P_i^{fw} , and $v_{d,i}$ are both considered as disturbances with respect to the weighted consensus. However, notice that if the \mathcal{H}_∞ approach of Theorem 5.1 is selected, then the disturbance matrix B_1 needs to be considered as an uncertain one as it depends on μ_1 via B_w . Eventually, in order to respect Assumption 6.4, we require

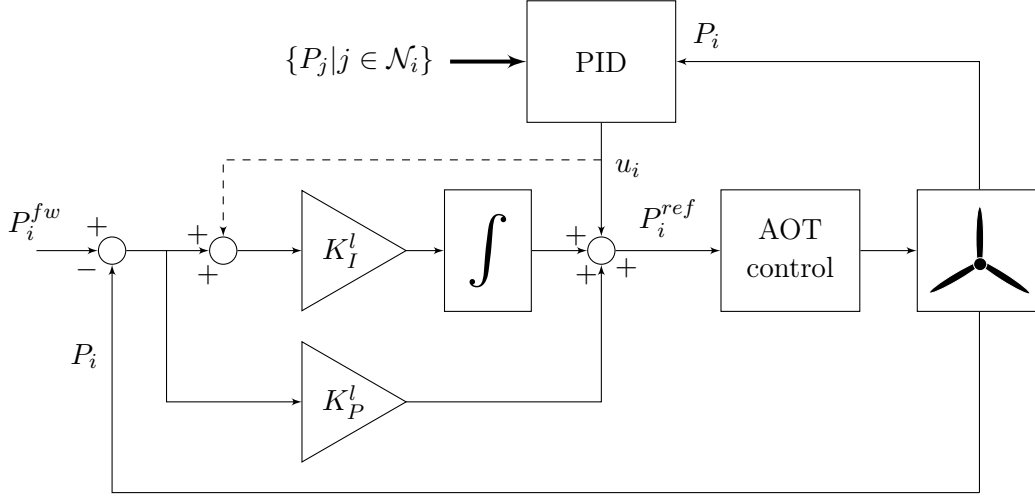


Figure 6.2: WT control scheme: the local control is composed of an AOT step and of a PI; the distributed control has a PID structure.

Assumption 6.5 All $v_{d,i}$ are such that the corresponding P_i^{ref} , depending on (6.4), on the local PI control, and on the distributed PID, is always lower than the real maximum power that turbine i can extract from the wind.

Remark 6.5 The considered control approach relies on Assumption 6.3. In particular disturbances $v_{d,i}$ are supposed to be zero-mean signals. If there exist WTs for which $v_{d,i}$ has not zero mean, then the corresponding absolute power reference P_i^{fw} would be in average different from the optimal one. Unfortunately under these circumstances, there is no hope for the optimal absolute power sharing to be satisfied by the only means of consensus control, even if optimal relative power sharing condition (6.5) is satisfied. This is due to the fact that in a network of agents communicating on an undirected graph, each of them has a role in determining the consensus function to which they converge. Thus, WTs affected by nonzero-mean wind disturbance would make the whole network deviate from the optimal absolute power values in average.

Improving Performance via Multirate

We conclude the discussion on WF leaderless consensus by mentioning the possibility to enhance the presented control capabilities in rejecting disturbances by means of the well-known *multirate* technique, (see e.g. [Monaco 2001, Cimino 2009]). Without going into the details of the aforementioned digital control technique, we propose to show a simple application to the particular case of WF control, object of this chapter. The basic idea of multirate sampling consists in letting the control u_i of system (6.14) vary q times between two sampling instants, during the sampling time T_s . This is attained by resampling system (6.14) at $T_c \triangleq T_s/q$. For the sake of

simplicity, let us not consider disturbance matrix B_1 . Resampled system (6.14) is thus described at T_c sampling time by

$$x_i(h+1) = Fx_i(h) + Gu_i(h)$$

$$\text{where } F \triangleq A^{1/q}, \quad G \triangleq \left(\sum_{j=0}^{q-1} F^j \right)^{-1} B_2$$

and where h indicates sampling instants at T_c sampling time. Let us assume that system (6.14) measured and controlled output is $\hat{C} = I \in \mathbb{R}^{3 \times 3}$. Then, recalling Remark 5.4, a reasonable choice for T_c is $q = 3$. Thus, system (6.14) with multirate evolves at the original sampling time T_s according to

$$x_i((k+1)T_s) = Ax_i(kT_s) + \underbrace{\begin{bmatrix} F^2G & FG & G \end{bmatrix}}_{\triangleq B_{2,mr}} \begin{bmatrix} u_i(kT_s) \\ u_i(kT_s + T_c) \\ u_i(kT_s + 2T_c) \end{bmatrix} \quad (6.15)$$

where we made the dependence on T_s , and T_c explicit, for the sake of clarity. Matrices \hat{C} , and $B_{2,mr}$ can be now employed in Theorem 5.1, or 5.2 to compute the MIMO PID distributed controller.

Leader-follower Consensus

As we mentioned in Remark 6.5, leaderless consensus technique fails to restore optimal absolute power references when some WTs in the wind farm are subject to nonzero-mean wind measurement errors. This is due to the fact that all the WTs communicating on the undirected graph contribute to the common consensus function. However, if the WTs being affected by nonzero-mean wind disturbances can be detected and isolated then, for instance, the leader-follower techniques developed in Chapter 5 can be employed to restore the optimal power references in the concerned WTs. This could be achieved by letting the faulty WTs be *follower* agents, and the unfaulty WT network serve as a *leader*.

If, for the sake of simplicity, we consider only one WT not to be affected by absolute power reference error, and we thus let it be the only leader in the WF, then either Theorem 5.3 or 5.4 can be applied by choosing A , B_2 the matrices of (6.14), $C = [0 \ 0 \ 1]$, and $B_1 = B_{fw}$. In addition to what shown in the leader-follower development of Chapter 5, in this case either the followers and the leader are additionally subject to disturbances via matrix $[B_{fw} \ B_w]$, i.e. P_i^{fw} , and $v_{d,i}$ are both considered as disturbances with respect to the weighted consensus.

6.4 Simulations Results

In this section we test the proposed WF consensus control via some numerical simulations. First of all though, we aim at showing the AOT control acting on a generic wind turbine. Recall that the WT model is described by system (6.3). In order

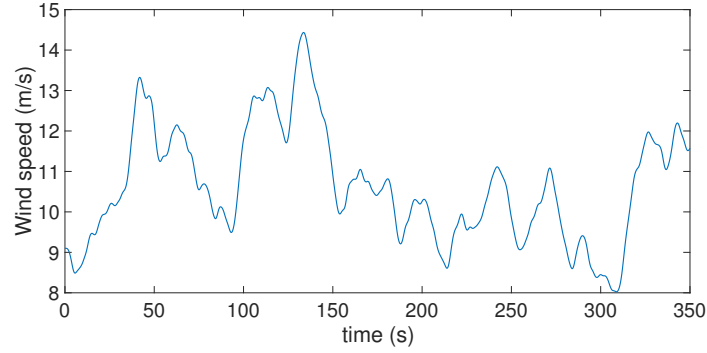
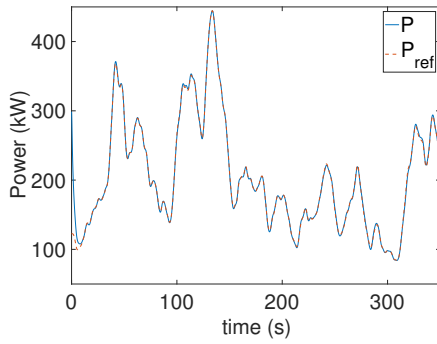
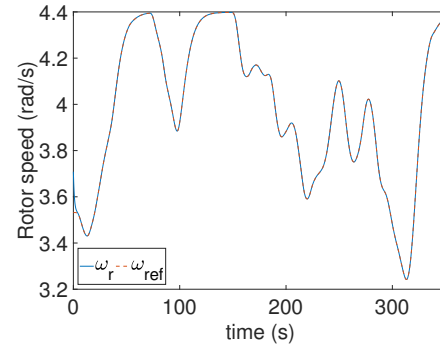


Figure 6.3: Effective wind speed.

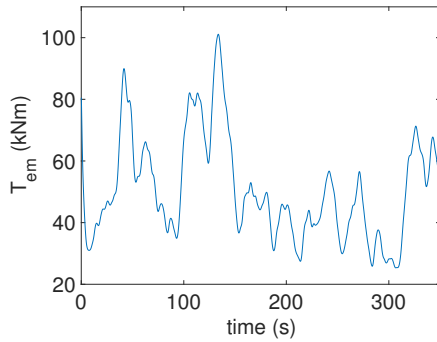


(a) Deloaded power reference and output.

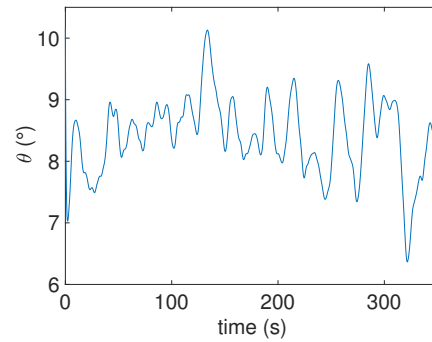


(b) Rotor angular speed reference and output.

Figure 6.4: Wind turbine power and rotor angular speed for deloading mode.



(a) Electromagnetic torque.



(b) Pitch angle.

Figure 6.5: Wind turbine inputs for deloading mode.

to test the controller performance we consider CART power coefficient, which we remind to be provided as a lookup table, and it is shown in Fig. 2.2 in Chapter 2. This represents the only source of model-plant mismatch for the local AOT-based controller, as the controller design is developed via the use of a polynomial approx-

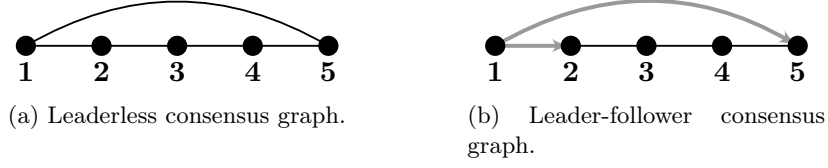


Figure 6.6: WF example communication graph.

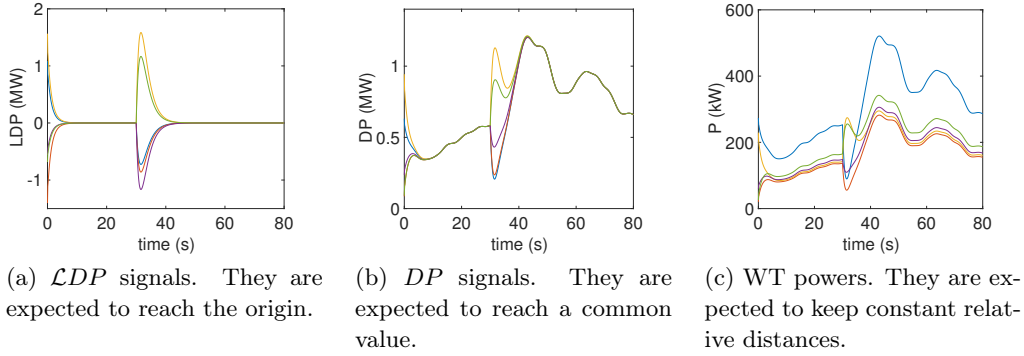


Figure 6.7: Wind farm control with no distributed protocol.

imation of C_p . Test under more realistic uncertainties are beyond the scope of this chapter and it will not be considered further. During a time interval of 350 s the WT is excited by the effective wind speed shown in Fig. 6.3. The WT is required to track a deloaded power reference $P^{ref}(t) = 50\%P^o(t)$. The chosen wind speed and deloading factor are such that $P^{ref}(t)$ respects Assumption 6.1. Moreover, the closed loop system trajectories are such that $\frac{\partial C_q}{\partial \vartheta} < 0$, which would let the error $|P - P^{ref}| \rightarrow 0$ as $t \rightarrow \infty$ in nominal conditions. However, even in the presence of the uncertainty caused by C_p , good performance is reached, as it can be seen in Fig. 6.4a. Input signals T_{em} , and ϑ , are shown in Fig. 6.5a, and Fig. 6.5b respectively. As it can be remarked, power deloading is achieved mainly via pitch angle control, since the rotor angular speed, illustrated in Fig. 6.4b, tracks a filtered ω_r^o reference which does not allow a surplus of energy storage in the rotating masses as done in the FL+MPC design.

In the remaining simulation test we focus on the distributed control capability in letting a WF reach weighted consensus. For the following tests we consider a WF composed of $N = 6$ WTs aligned one after the other according to the wind direction. We suppose the 6-th WT to be the last one of the row according to the wind direction. Thus it is required to always operate in classic MPPT mode. In the following tests, it is supposed to not intervene in the consensus control, and its power signals will not be reported. Let us first consider the leaderless case. Here the remaining WTs are supposed to communicate on the communication graph shown in Fig. 6.6a. The free stream wind speed v_∞ blowing in front of WT 1 is chosen to be

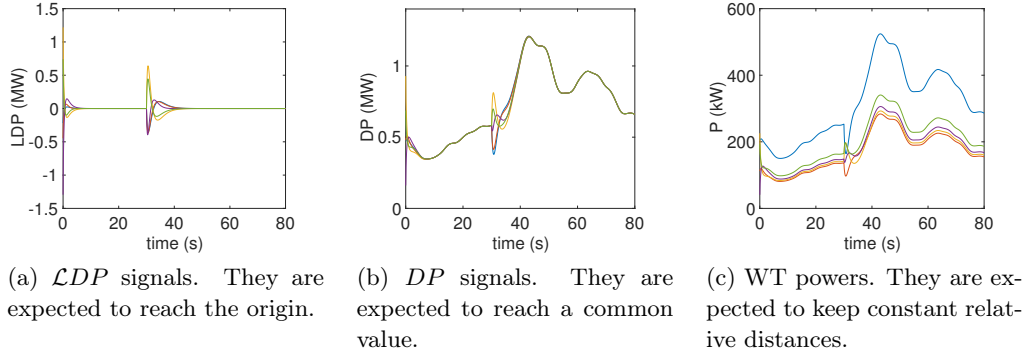


Figure 6.8: Wind farm control with distributed PID.

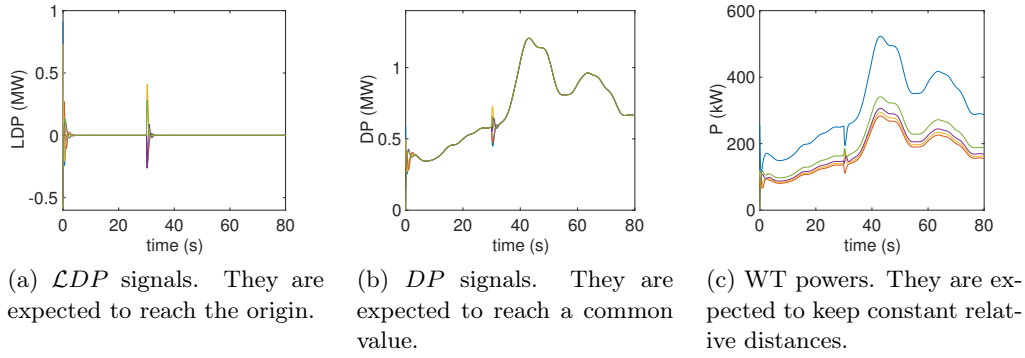


Figure 6.9: Wind farm control with distributed multirate PID.

the one employed in the previous simulation, and shown in Fig. 6.3. The other wind speed signals v_i , $i = 2, \dots, 5$ are obtained from v_∞ according to the wake model of Chapter 4, and the optimal induction factor values provided by the optimization step. First of all, let us analyze the system response to *step disturbances* in the ideal case of perfect AOT control, and in the case in which the considered wind approximation is exact. In other words, we assume each WT dynamics to be exactly described by system (6.14), where $T_s = 0.1$ s. This one depicts the AOT-controlled WT with the additional PI control loop, whose tunable parameters are set as $K_P^l = 0.2$, and $K_I^l = 0.8$. Concerning the disturbance $v_d \triangleq \text{col}(v_{d,1}, \dots, v_{d,5})$ acting on the system, after a system initialization where $v_d = \mathbf{0}$, we let $v_d = \text{col}(2, 1, -1.5, 0.5, -1)$ from time $t = 30$ s. Moreover, we let the effect of v_d on system (6.14) be maximum by selecting $\mu_1 = \mu_{1,min}$, as recall that $\mu_1 \in [\mu_{1,min}, 0]$. Even if such test does not reproduce any real situation, it enables us to evaluate the distributed protocol performance in rejecting disturbances. Moreover, we propose to consider $P^{fw} = \text{col}(P_1^{fw}, \dots, P_5^{fw})$ as a vector of signals not corrupted by wind measurement error. As mentioned in Subsection 6.3.2, in such case the distributed protocol is not necessary to reach consensus condition (6.5) because of the presence of the local PI control. However, we aim at showing that the employment of the additional

distributed PID can lead to better performance. This is reported in the following. The PID gains for the aforementioned multi-agent system are tuned via the LMI technique of Theorem 5.2 as, for this case, it showed better performance over the one of Theorem 5.1. These are

$$K_p = 0.0072, \quad K_i = -0.0638, \quad K_d = 0.0013$$

and they allow weighted consensus achievement with performance index $\tau = 24.5$. Results are shown on signals \mathcal{LDP} , DP , and P , in a time window of 80 s, and where $P = \text{col}(P_1, \dots, P_5)$. Indeed recall that weighted consensus is reached if $\mathcal{LDP} = 0$. This condition can be alternatively seen as weighted power signals DP reaching a *common* consensus function, or as power signals P keeping defined constant relative distances. In Fig. 6.7 we show the results obtained via the only local PI control, while in Fig. 6.8 we show the signals of interest obtained with the additional use of the distributed control. Even if in both cases consensus is achieved, the use of the distributed protocol leads to better performance, as it can be seen by comparing Fig. 6.7a with Fig. 6.8a, and Fig. 6.7b with Fig. 6.8b.

As a second simulation we test multirate technique introduced in Subsection 6.3.3, where we assume each agent to measure and control the whole state x_i for consensus purposes. With the choice of $q = 3$, the PID gain matrices tuned via Theorem 5.2 are

$$K_p = \begin{bmatrix} -0.1810 & 0.5018 & -0.4645 \\ 0.3797 & -1.0529 & 0.9745 \\ -0.1983 & 0.5501 & -0.5091 \end{bmatrix} \cdot 10^6,$$

$$K_i = \begin{bmatrix} -0.8394 & 2.4028 & -2.2273 \\ 1.7606 & -5.0410 & 4.6729 \\ -0.9198 & 2.6336 & -2.4413 \end{bmatrix} \cdot 10^5, \quad K_d = \begin{bmatrix} 0.0028 & 0.0049 & -0.0048 \\ 0.0038 & 0.0003 & -0.0022 \\ 0.0047 & -0.0038 & 0.0001 \end{bmatrix}$$

referring to a MIMO controller as, in this case, the input is allowed to vary 3 times within sampling time T_s , (see equation (6.15)). The achieved performance index is $\tau = 3.5$. As a result, consensus is obtained in a much faster way with respect to the previous cases, as it can be seen in Fig. 6.9.

We conclude the leaderless consensus simulation test by analyzing the described WF example performance when considering the whole control, i.e. from the AOT one to the distributed PID. In such case the system is subject to two sources of model-plant mismatch. The first one is caused by differences between the polynomial C_p approximation used for the AOT-based controller design and the CART power coefficient. The second one is given by the way $v_{d,i}$ acts on WT i . Indeed, each $v_{d,i}$ affects the according WT i dynamics via the mechanical power P_r , (see equation (6.1)). This causes an additional mismatch as recall that $v_{d,i}$ effect was approximated in Subsection 6.3.2, leading to the approximated model of system (6.13). For this simulation the PID controller is chosen to be the previously obtained SISO PID, whose performance index is $\tau = 24.5$. The disturbance v_d affecting the system is chosen to be the one shown in Fig. 6.10. In this case P^{fw} is computed on the

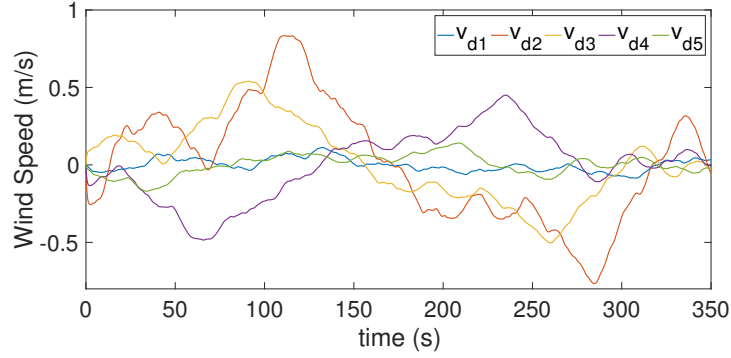


Figure 6.10: Wind disturbance.

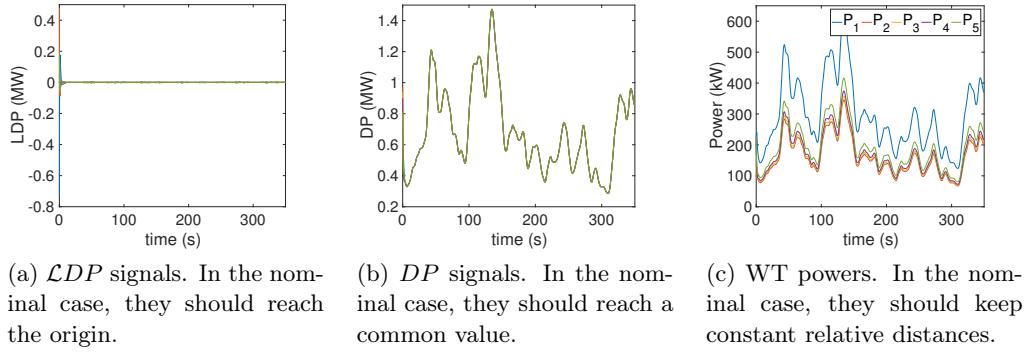
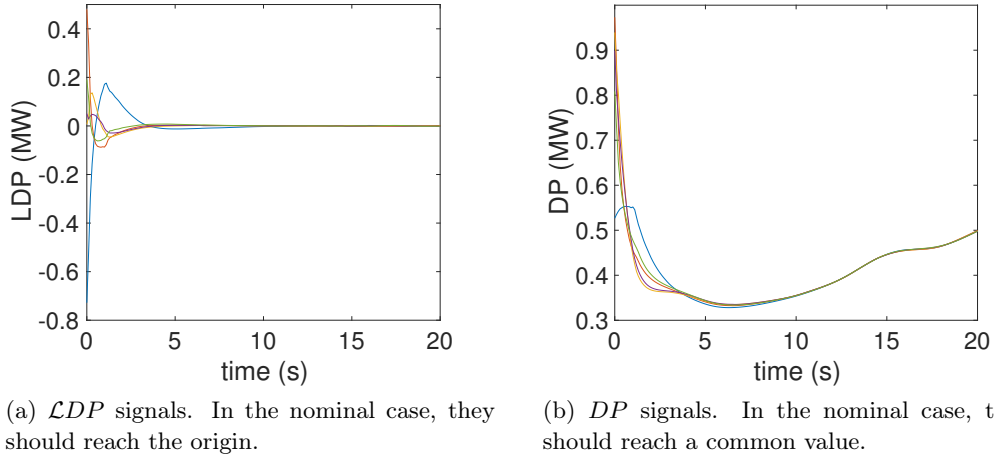


Figure 6.11: Wind farm control with distributed PID.

Figure 6.12: Zoom on \mathcal{LDP} and DP during WF consensus control.

available measurements $v_{m,i}$. Thus, they are corrupted by measurement error. As done for the previous simulations, we report the results by showing \mathcal{LDP} , DP , and

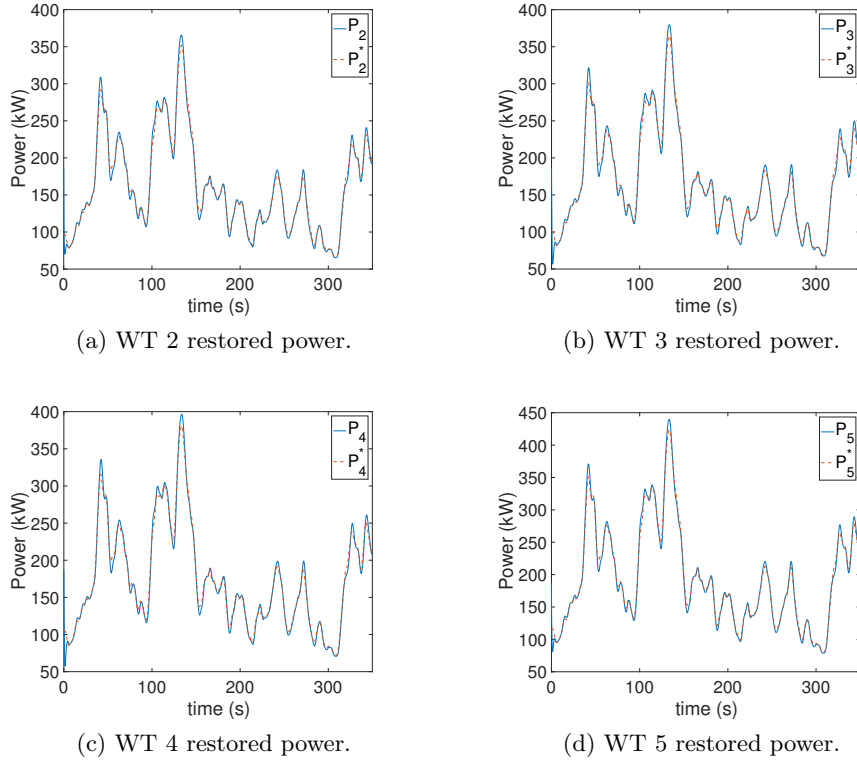


Figure 6.13: Wind farm leader-follower control.

P . These are illustrated in Fig. 6.11, from which we can also notice that the aforesaid model-plant mismatches, as well as temporary dissatisfaction of Assumption 6.4, cause persistent small oscillations around the reached weighted consensus. For the sake of clarity, \mathcal{LDP} , and DP are also shown in a zoomed window in Fig. 6.12.

Eventually, we aim at showing how optimal absolute power references P_i^* can be restored via leader-follower consensus when some WTs in the WF are affected by power reference error. In order to do, we consider a simple example in which only the first WT in the row is error-free, and thus acts as the leader. In such case, we consider the WF communication graph to be modified according to Fig. 6.6b, where the leader, WT 1, can communicate with WT 2 and WT 5 directly. The remaining WTs from 2 to 5 are followers and they can communicate on an undirected connected graph. We suppose the leader $v_{d,1}$ to be equal to the one shown in Fig. 6.10, whereas the followers wind disturbances are *nonzero*-mean signals of the form of $v_{d,i} = \bar{v}_{d,i} + \tilde{v}_{d,i}$, where $\tilde{v}_{d,i}$ is a zero-mean signal, and $\bar{v}_{d,i}$ a constant nonzero value. $\bar{v}_{d,i}$, $i = 2, \dots, 5$ are supposed to be equal to the corresponding i -th signal in Fig. 6.10 used in the previous simulation, while $\bar{v}_{d,i}$ are such that the computed power references via (6.4) are $P_2^{fw} = 80\%P_2^*$, $P_3^{fw} = 60\%P_3^*$, $P_4^{fw} = 30\%P_4^*$, $P_5^{fw} = 40\%P_5^*$, i.e. without leader-follower control action they would track a non-optimal power reference. By employing fast leader-follower weighted consensus technique, the PID gains are $K_p = 0.0082$, $K_i = -0.0656$, and $K_d = -0.0028$, and they allow a

performance index equal to $\tau = 24.5$. Simulation results are illustrated in Fig. 6.13, where the red dashed signals represent the original optimal power reference P_i^* , and the blue solid line the obtained power output via consensus control. We are able to conclude that consensus control shows good performance in restoring the optimal absolute power values. Small persistent errors are due to model-plant mismatches, by the fact that the PID structure cannot reject general time-varying reference state, and because wind disturbance $\tilde{v}_{d,i}$ also affect the system consensus function.

6.5 Conclusion and Future Perspectives

6.5.1 Conclusion

We presented a novel distributed approach to control a WF for power maximization under the presence of wind measure disturbances. After a first higher optimization step, which can be performed via the presented tools of Part II, the low control level is based on two additional layers. First, the proposed AOT controller allows a generic WT to track a general deloaded power reference. Differently from what done in Chapter 2, the employed deloading technique is mainly based on pitch control, as the WT rotor angular speed ω_r tracks a filtered ω_r^o reference. Unfortunately this no longer allows to store additional kinetic energy in the rotating masses when deloading is performed. However the considered AOT-based control scheme has the advantage of letting a simpler implementation since, as pointed out in Remark 6.3, the rotor angular speed reference does not have to be computed by solving an optimization problem. Moreover, since MPC technique is not used, this enables a simpler problem formulation concerning the WF distributed problem.

Secondly, the PID-like weighted leaderless consensus described in Chapter 5 is employed to let the WF turbines maintain the optimal power sharing relative values under the presence of wind disturbances. In particular, the distributed control helps out keeping the relative optimal power sharing condition when they are zero-mean signals. In the presence of nonzero-mean wind disturbances causing in turns an absolute power reference error, we additionally showed how leader-follower control technique can be used to restore the optimal absolute power sharing, if the faulty WTs can be identified.

6.5.2 Future Perspectives

One of the main drawbacks of the presented AOT technique to control a WT is identified in Remark 6.4 where we claim that it should not be generally applied for classic WT MPPT operating mode. On the basis of the FL+MPC approach of Chapter 2, it would be interesting to modify the WT P^{ref} on-line by means of MPC in order to impose condition $\frac{\partial C_q}{\partial \vartheta}(t) < 0 \ \forall t \geq 0$ in the system closed-loop trajectories, and in turns to let AOT control in the whole operating envelope.

Concerning the proposed wind farm consensus control, in the near future, it would be interesting to evaluate the power gain which can be achieved when considering

the wake effect and more real world wind disturbances. Moreover, different scenarios could be considered to see the benefits coming from the potential use of both leaderless and leader-follower control schemes. All in all, consensus techniques represent a fairly new approach to control wind farms and they have the potential to lead to a great variety of applications. For instance one can imagine to use distributed estimation techniques to enhance wind field estimation in a wind farm by directly employing the wake model.

GENERAL CONCLUSION AND PERSPECTIVES

THE main objectives of the research work presented in this report are met. These consisted in identifying some of the major issues and opportunities that are to be considered nowadays, aiming at improving wind generation and penetration in the electric grid, as well as suggesting and developing advanced control tools to help the transition of wind energy generation towards its integration in the existing and future power systems. Bearing in mind the new wind farm challenges and requirements to be met, we first translated them into control objectives, then conjugated them according to the system components under analysis, starting from its foundation, i.e. the wind turbines, until reaching the system as a whole, i.e. the wind farm. This led us to define different control levels, and for each of them we proposed solutions that could integrate the existing system functioning while optimizing it whenever possible, and that could allow the introduction of new modes of operation.

At the wind turbine level, the benefit of the proposed nonlinear controller with respect to more classic ones is evidenced by the fact that the controlled WT is either capable to operate in classic MPPT and power limiting mode and able to track general deloaded power references. Moreover this is achieved by employing one controller for the whole WT operating envelope. Tracking general deloaded power references is essential in the perspective of integrating the frequency control and power curtailment requirements in the WT operating modes as well as to let possible power gains at the WF level when wake effect is considered.

As far as WF control operations are concerned, while respecting the grid active power requirements, our main objective was to propose a distributed architecture which could lead to improved power gains when considering the wake interaction among the WTs. In the considered hierarchical control structure, the coordination level is devoted to the solution of an optimization problem by considering the wake model in its formulation. Aiming at fast convergence of such optimization problem, while taking into consideration the system constraints, two novel distributed optimization algorithms were proposed in the metaheuristic optimization framework. Testing the overall WF controller let us conclude on the interest in considering a cooperative control solution with respect to the existing greedy ones. Perhaps the most interesting feature of the proposed distributed optimization algorithms is that, a priori, they could be implemented on any existing wind farm without the need for extra mechanical equipment, under the assumption that the WTs can track general power references. Moreover the considered architecture is scalable, resilient, and little modifications need to be considered if wind turbines are added to or removed from the wind farm.

Eventually, an additional WF control step was considered, in the promising framework of consensus control. Here, testing the proposed distributed controller suggests that better performance can be achieved if additional cooperation is allowed at the WT control level. Indeed, it enables taking into account system disturbances that

could make the WTs deviate from the optimal operating points set by the optimization level, and reduce their effect on the WF optimization.

In future perspectives new ways to explore can be hypothesized according to the concerned WF control level. Regarding the WT one, as the employed FL+MPC approach showed good versatility in controlling a WT in different operating modes, it would be worth considering further developments that could lead to conclude on the mathematical stability of the system as well as to new applications, such as fatigue minimization, noise reduction, etc. Moreover, in the last part of this report, we proposed a different WT control approach based on AOT technique. Despite presenting a suboptimal functioning with respect to the stored kinetic energy when compared to the FL+MPC technique, AOT-based control could allow a simpler integration of new modes of functioning in the existing real world WT controllers. This is why in the author's opinion, tests under more realistic scenarios would be worth considering.

Concerning the WF optimization, it would be interesting to implement the proposed distributed algorithms on some experiments mocking the functioning of real wind farms, in order to prove the real achievable power gains. Being the proposed algorithms applicable to a class of optimization problems, one could find other new possible applications in distributed generation and smart grids.

Consensus control for WF operations represents a fairly new application and it may lead to promising avenues to explore. For instance, further developments could be carried out for wind estimation purposes. This is an important task not only for proper WF control, but it could be also useful in order to estimate the WF power capabilities in real time for a better integration in the electric grid.

Eventually, some of the presented tools could be employed to address another leading issue in WF control, which is the fatigue reduction. Both WT and WF level controllers could be adapted in order to integrate the system mechanical stress reduction.

APPENDICES

Basic Mathematical Notions and Definitions

A.1 Positive Definite Matrix

A square matrix $A \in \mathbb{R}^{n \times n}$ is said to be *positive definite* if the scalar $x^\top Ax$, where $x \in \mathbb{R}^n$, is positive, i.e. $x^\top Ax > 0$, for every non-zero vector x . We indicate a positive definite matrix with the notation $A \succ 0$.

Similarly, A is said to be *positive semidefinite* if $x^\top Ax \geq 0, \forall x$. In this case, we use the notation $A \succeq 0$.

A.2 Lie Derivative

A mapping f assigning to each point $x \in \mathcal{X} \subseteq \mathbb{R}^n$ a vector of \mathbb{R}^n , i.e. $f(x) = \text{col}(f_1(x), \dots, f_n(x))$, is said to be a *vector field* defined on \mathcal{X} . In general, we will consider the vector fields to be *smooth*, i.e. having continuous partial derivatives of any order. Given a real-valued function λ , and a vector field f , both defined on \mathcal{X} , we name *Lie derivative*, or *derivative of λ along f* , the following

$$L_f \lambda(x) \triangleq \sum_{i=1}^n \frac{\partial \lambda}{\partial x_i} f_i(x) \quad (\text{A.1})$$

The result of (A.1) can be applied to another vector field g , yielding $L_g L_f \lambda(x) = \frac{\partial(L_f \lambda(x))}{\partial x} g(x)$. Moreover, if $L_f^0 \lambda(x) \triangleq \lambda(x)$, then (A.1) satisfies the recursion $L_f^k \lambda(x) = \frac{\partial(L_f^{k-1} \lambda(x))}{\partial x} f(x)$.

A.3 Global and Local Diffeomorphism

A nonlinear change of coordinates in \mathbb{R}^n of the form $z = \phi(x)$, where ϕ is a \mathbb{R}^n -valued function, is a *global diffeomorphism* if

- ϕ is invertible, i.e. it exists $\phi^{-1}(z): \phi^{-1}(\phi(x)) = x \quad \forall x \in \mathbb{R}^n$.
- ϕ and $\phi^{-1}(z)$ have both continuous partial derivatives for any order.

When such coordinate transformation is only defined in a *neighborhood* of a given point, then it is said to be a *local diffeomorphism*. If the jacobian matrix of ϕ computed at x_0 is nonsingular, then $\phi(x)$ defines a local diffeomorphism.

A.4 Feedback Linearization

Feedback linearization is a well-known technique to control nonlinear dynamic systems. The results of this section are mainly taken from [Isidori 2013]. No proof is provided, as our aim is to only provide the basic notions that will be used in this report. Notation is intentionally chosen to be the one used in the mentioned reference.

A.4.1 Single-Input Single-Output Systems

The nonlinear system affine in the control u , given by

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (\text{A.2})$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, is said to have *relative degree* $r \leq n$ at a point x_0 if

- $L_g L_f^k h(x) = 0 \quad \forall x$ in a neighborhood of x_0 , and all $k < r - 1$.
- $L_g L_f^{r-1} h(x_0) \neq 0$.

where Lie derivative defined in Section A.2 was used. Note that the relative degree definition depends on mapping h . There might exist points in which the relative degree is not defined. From a practical point of view, it is equal to the number of times one has to differentiate the output $y(t)$ with respect to time in order to have the input u appearing explicitly. Indeed, from the above definition and from (A.1), we have

$$\begin{aligned} \dot{y} &= \frac{\partial h}{\partial x} f(x) = L_f h(x) \\ &\vdots \\ y^{(r-1)} &= \frac{\partial y^{(r-2)}}{\partial x} f(x) = L_f^{r-1} h(x) \\ y^{(r)} &= \frac{\partial y^{(r-1)}}{\partial x} f(x) = L_f^r h(x) + L_g L_f^{r-1} h(x)u \end{aligned} \quad (\text{A.3})$$

If we suppose $r < n$, system output $h(x)$, together with the first $r - 1$ functions of (A.3) naturally define a partial change of coordinates for system (A.2). This can be always completed with $n - r$ additional *arbitrary* functions $\phi_{r+1}(x), \dots, \phi_n(x)$ such that the mapping

$$\Phi(x) = \text{col} \left(h(x), L_f h(x), \dots, L_f^{r-1} h(x), \phi_{r+1}(x), \dots, \phi_n(x), \right) \quad (\text{A.4})$$

is a local diffeomorphism in a neighborhood of x_0 . In addition, it is always possible to choose the additional $n - r$ functions such that $L_g \phi_i(x) = 0 \quad \forall r + 1 \leq i \leq n$.

Thus, if we name $z = \Phi(x)$, then system (A.2) is described in the new coordinates by

$$\begin{cases} \dot{z}_1 &= z_2 \\ &\dots \\ \dot{z}_{r-1} &= z_r \\ \dot{z}_r &= b(z) + a(z)u \\ \dot{z}_{r+1} &= q_{r+1}(z) \\ &\dots \\ \dot{z}_n &= q_n(z) \\ y &= z_1 \end{cases} \quad (\text{A.5})$$

System (A.5) is said to be in *strict normal form*. More in general, if we are not able to find additional functions such that the condition $L_g \phi_i(x) = 0 \ \forall r+1 \leq i \leq n$ is satisfied, system (A.2) can be described in *normal form*, which can be represented in compact form by

$$\begin{cases} \dot{\xi} &= A\xi + B(b(\xi, \eta) + a(\xi, \eta)u) \\ \dot{\eta} &= q(\xi, \eta) + p(\xi, \eta)u \\ y &= C\xi \end{cases} \quad (\text{A.6})$$

where we named $\xi \triangleq \text{col}(z_1, \dots, z_r)$, $\eta \triangleq \text{col}(z_{r+1}, \dots, z_n)$, and where

$$A \triangleq \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad B \triangleq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad C \triangleq \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}^\top \quad (\text{A.7})$$

A.4.1.1 Case $r = n$: Exact Linearization

In this subsection we address the problem of transforming nonlinear system (A.2) into a linear and controllable one, by means of the change of coordinates presented in the previous subsection. Under the assumption of measurable state, this is achieved by a *static state feedback control* of the form $u = \alpha(x) + \beta(x)v$, where v is the external reference input, and it is left as degree of freedom.

If system (A.2) has relative degree $r = n$ at some point x_0 , then no extra functions are needed to complete the diffeomorphism (A.4), and there will be no η coordinates. In such situation the feedback linearizing input is given by

$$u = \frac{1}{a(z)}(-b(z) + v) \quad (\text{A.8})$$

which can be also expressed in the original coordinates via

$$u = \frac{1}{L_g L_f^{n-1} h(x)}(-L_f^n h(x) + v) \quad (\text{A.9})$$

The closed loop system resulting from the application of (A.8) is thus linear and controllable, as it is governed by a system that is in the canonical controllable form, and its matrices are given by (A.7). Note that in this case of $r = n$, matrix A in (A.7) has dimension $n \times n$. On the obtained linear system one can impose a new state feedback by acting on input v to impose local asymptotic stability. Note that stability is in general only local if the diffeomorphism is local.

A.4.1.2 Case $r < n$: Input-Output Linearization

Since the relative degree depends on the output function h , more in general, the problem of exact linearization can be posed as the one of finding a *different* output function $\lambda(x)$ such that $r = n$. The existence of such function is a necessary and sufficient condition for the existence of an exact feedback linearizing input for system (A.2). In particular, the given conditions can be interpreted as a nonlinear version of the controllability conditions for nonlinear systems. The reader may refer to the mentioned reference for further details. If the system happens to be feedback linearizable still a set of partial differential equations need to be solved in order to find $\lambda(x)$. Moreover, in general, the real system output would be nonlinear in the new coordinates. For this reason, if the system has relative degree $r < n$ for a given output, still one can perform a feedback linearization yielding the *partially* linear system

$$\begin{cases} \dot{\xi} &= A\xi + Bv \\ \dot{\eta} &= q(\xi, \eta) + \frac{p(\xi, \eta)}{a(\xi, \eta)}(-b(\xi, \eta) + v) \\ y &= C\xi \end{cases} \quad (\text{A.10})$$

where the feedback linearizing input has been replaced in system (A.6). In order to conclude on the stability of (A.10), it is useful to recall the concept of *zero dynamics*. This is naturally defined in the context of the so-called *problem of zeroing the output*, that is to find all pairs $(x(0), u(t))$, i.e. initial state and input, such that the corresponding output is *identically* zero. Considering the system in its normal form, then $y(t) \equiv 0$ implies $\dot{z}_1 = \dots = \dot{z}_r = 0$. In particular, $\dot{z}_r = 0$ imposes the input to be

$$u(t) = -\frac{b(0, \eta(t))}{a(0, \eta(t))} \quad (\text{A.11})$$

Thus, if the output is identically zero, then the system initial condition in the new coordinates must be set as $(\xi(0), \eta(0)) = (0, \eta_0)$, where η_0 is arbitrary, and u must satisfy (A.11), where $\eta(t)$ is the solution of

$$\dot{\eta} = q(0, \eta) - p(0, \eta) \frac{b(0, \eta)}{a(0, \eta)}; \quad \eta(0) = \eta_0 \quad (\text{A.12})$$

The dynamics (A.12) are called *zero dynamics*. In the original coordinates, the same problem of zeroing the output provides as initial condition $x(0)$ all the points belonging to the manifold $Z^* = \left\{ x \in \mathbb{R}^n : h(x) = \dots = L_f^{r-1}h(x) = 0 \right\} =$

$\{x \in \mathbb{R}^n : \Phi(x) = 0\}$, and input $u^*(x) = -\frac{L_f^r h(x)}{L_g L_f^{r-1} h(x)}$. Thus the zero dynamics is

the dynamics of system $\dot{x} = f(x) + g(x)u^*(x)$ on Z^* .

Without loss of generality, let us now consider x_0 to be an equilibrium point for $\dot{x} = f(x)$, and that $h(x_0) = 0$. Moreover $(\xi, \eta) = (0, 0)$ at x_0 . The important result linking the zero dynamics with the problem of stabilizing (A.10) is that the control law

$$\begin{cases} u &= \frac{1}{L_g L_f^{r-1} h(x)} (-L_f^r h(x) + v) \\ v &= -c_0 z_1 - \dots - c_{r-1} z_r \end{cases} \quad (\text{A.13})$$

locally asymptotically stabilize (A.10) if $\eta = 0$ is locally asymptotically stable for the zero dynamics (A.12), and $p(s) = c_0 + c_1 s + \dots + c_{r-1} s^{r-1} + s^r$ has its roots in the left half complex plane.

Remark A.1 *Note that if the employed diffeomorphism is global and the zero dynamics (A.12) is globally asymptotically stable, still only local stability of the closed loop system via (A.13) can be concluded. In order to have global stability of the closed loop system, the zero dynamics needs to be input-to-state stable, [Khalil 2002].*

Remark A.2 *A stabilizing control can be found even if we do not know the explicit expression of the normal form (A.6). We only need to know that the zero dynamics is asymptotically stable. Then the control law can be described in the original coordinates as*

$$u = \frac{1}{L_g L_f^{r-1} h(x)} (-L_f^r h(x) - c_0 h(x) - \dots - c_{r-1} L_f^{r-1} h(x))$$

A.4.2 Multi-Input Multi-Output Systems

The results of the previous subsection can be extended to MIMO systems. For the sake of simplicity, here we shall consider the analysis of systems having equal number of inputs and outputs m . The MIMO system to be controlled is described by

$$\begin{cases} \dot{x} = f(x) + \sum_{i=1}^m g_i(x) u_i \\ y_i = h_i(x), \quad i = 1, \dots, m \end{cases} \quad (\text{A.14})$$

being $f(x), g_i(x)$, $i = 1, \dots, m$ smooth vector fields, and $h_i(x)$, $i = 1, \dots, m$ are smooth functions on an open set of \mathbb{R}^n . As for the SISO case, the multivariable version of the FL technique is based on the concept of relative degree and on the consequent capability to represent (A.14) in normal form. System (A.14) is said to have vector relative degree $[r_1 \dots r_m]$ at a point x_0 if

- For $i = 1, \dots, m$: $L_{g_j} L_f^k h_i(x) = 0$, $j = 1, \dots, m$, for all $k < r_i - 1$, and for all x in a neighborhood of x_0 .

- the $m \times m$ matrix

$$D(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \cdots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_1(x) & \cdots & L_{g_m} L_f^{r_m-1} h_1(x) \end{bmatrix} \quad (\text{A.15})$$

is nonsingular at x_0 .

Note that each r_i is associated to the output y_i . Thus, for each i there is at least one input u_j such that the system having output y_i and input u_j has relative degree r_i at x_0 , and for any other possible input the relative degree is greater or equal to r_i . From a computational point of view, the relative degree associated to the output y_i is equal to the number of times the output can be differentiated until at least one of the inputs appears. The relative degree allows to define a natural change of coordinates. Indeed, if system (A.14) has vector relative degree in such a way that $r_1 + \cdots + r_m < n$, then, for $i = 1, \dots, m$ we can set

$$\begin{aligned} \phi_1^i(x) &= h_i(x) \\ &\vdots \\ \phi_{r_i}^i(x) &= L_f^{r_i-1} h_i(x) \end{aligned} \quad (\text{A.16})$$

and complete the mapping (A.16) with $n - r$ additional arbitrary functions $\phi_{r+1}(x), \dots, \phi_n(x)$ such that

$$\Phi(x) = \text{col}(\phi_1^1(x), \dots, \phi_{r_1}^1(x), \dots, \phi_1^m(x), \dots, \phi_{r_m}^m(x), \phi_{r+1}(x), \dots, \phi_n(x))$$

defines a local diffeomorphism at x_0 . We now employ the latter to perform a change of coordinates. By naming

$$\xi^i \triangleq \begin{pmatrix} \xi_1^i \\ \vdots \\ \xi_{r_i}^i \end{pmatrix} = \begin{pmatrix} \phi_1^i(x) \\ \vdots \\ \phi_{r_i}^i(x) \end{pmatrix}, \quad \eta = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_{n-r} \end{pmatrix} = \begin{pmatrix} \phi_{r+1}(x) \\ \vdots \\ \phi_n(x) \end{pmatrix}$$

$\xi \triangleq \text{col}(\xi^1, \dots, \xi^m)$, and $a_{ij}(\xi, \eta) \triangleq L_{g_j} L_f^{r_i-1} h_i(\Phi^{-1}(\xi, \eta))$ for $i, j = 1, \dots, m$, $b_i(\xi, \eta) \triangleq L_f^{r_i} h_i(\Phi^{-1}(\xi, \eta))$ for $i = 1, \dots, m$, $u \triangleq \text{col}(u_1, \dots, u_m)$, system (A.14) can be written in the new coordinates as

$$\begin{cases} \dot{\xi}_1^i &= \xi_2^i \\ &\vdots \\ \dot{\xi}_{r_i-1}^i &= \xi_{r_i}^i \\ \dot{\xi}_{r_i}^i &= b_i(\xi, \eta) + \sum_{j=1}^m a_{ij}(\xi, \eta) u_j \\ y_i &= \xi_1^i \end{cases} \quad i = 1, \dots, m \quad (\text{A.17})$$

$$\dot{\eta} = q(\xi, \eta) + \sum_{j=1}^m p_j(\xi, \eta) u_j = q(\xi, \eta) + p(\xi, \eta) u$$

Note that a_{ij} are the entries of matrix (A.15). Equations (A.17) describes the *normal form* associated to (A.14). As for the SISO counterpart, the feedback linearizing input in the original coordinates is given by

$$u = D^{-1}(x)(-b(x) + v) \quad (\text{A.18})$$

where $b(x) \triangleq \text{col}(b_1(x), \dots, b_m(x))$, and where $v \in \mathbb{R}^m$ is the external reference input, left as degree of freedom. In close analogy with the results of the previous subsection, by reproducing the problem of zeroing the output, system (A.17) dynamics reduce to

$$\dot{\eta} = q(0, \eta) - p(0, \eta)D^{-1}(0, \eta)b(0, \eta) \quad (\text{A.19})$$

if $\text{col}(y_1, \dots, y_m)$ is identically zero. System (A.19) is the *zero dynamics* of system (A.14). If the latter has vector relative degree so that $r_1 + \dots + r_m = n$, then there is no zero dynamics and the system can be exactly linearized via (A.18).

Remark A.3 *If system (A.14) can be exactly linearized, then by employing input (A.18), the closed loop system is decoupled in m independent channels. For the generic channel i , v_i controls only y_i through a chain of r_i integrators. Control law (A.18) is thus said to solve the problem of noninteractive control.*

A.4.3 Alternative Change of Coordinates

As shown, a feedback linearizing input transforms the linearizable part of a nonlinear system in a chain of integrators or, in the MIMO case, in decoupled chains of integrators. As a consequence, once the system is linearized, often the physical meaning of the state variables is lost. For this reason it is sometimes convenient to find a change of coordinates that tackles the system nonlinearities while preserving the physical meaning of the system variables. In this case, the linearized system is no longer guaranteed to be a chain of integrators. From a mathematical point of view, this means that FL is not confined to systems in normal form, [Khalil 2002]. Of course, a connection with the FL theory presented in the previous two subsections needs to be established. For the sake of simplicity let us consider an exactly linearizable SISO system. Similar results hold for the MIMO case. If the system can be feedback linearized, then it exists a local diffeomorphism $T_\zeta(x)$, being $x \in \mathbb{R}^n$, such that in the new coordinates ζ , the system is described by

$$\dot{\zeta} = A_\zeta \zeta + B_\zeta(b_\zeta(\zeta) + a_\zeta(\zeta)u)$$

Note that in general the pair (A_ζ, B_ζ) is not necessarily the one shown in (A.7). The pair (A_ζ, B_ζ) is controllable, thus it exists a nonsingular linear change of coordinates $z = M\zeta$ such that the system described in the z coordinates is in canonical form of controllability. In other words, $MA_\zeta M^{-1} = A + B\lambda^\top$, $MB_\zeta = B$, where $\lambda \in \mathbb{R}^n$, and where (A, B) are the matrices in (A.7). The system can be thus written in normal form

$$\dot{z} = Az + B(b(z) + a(z)u)$$

where $a(z) = a_\zeta(M^{-1}z)$, $b(z) = \lambda^\top z + b_\zeta(M^{-1}z)$. In conclusion, $T_\zeta(x)$ is an alternative change of coordinates leading to FL, and it is related to the one putting the system in normal form via a nonsingular linear transformation.

A.5 Positive Invariant Set and Local Stability

We provide the following definitions for discrete time systems, as they will be used in this context.

Definition A.1 A set $X \subseteq \mathbb{R}^n$ is said to be *positive invariant* for the discrete time system $x(k+1) = f(x(k))$ if $x \in X$ implies $f(x) \in X$.

Definition A.2 A set $X \subseteq \mathbb{R}^n$ is said to be *positive control invariant* for the discrete time system $x(k+1) = f(x(k), u(k))$, where $\mathbb{U} \in \mathbb{R}^m$ if $\forall x \in X$ it exists $u \in \mathbb{U}$ such that $f(x, u) \in X$.

Theorem A.1 The system $x(k+1) = f(x(k))$, $x \in \mathbb{R}^n$, has the origin asymptotically stable with a region of attraction X , if X is positive invariant for the given system, and if there exist two functions α_1, α_2 of class \mathcal{K}_∞ , a positive definite function α_3 , and a function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\forall x \in X$

$$\begin{aligned} (i) \quad & V(x) \geq \alpha_1(|x|) \\ (ii) \quad & V(x) \leq \alpha_2(|x|) \\ (iii) \quad & V(f(x)) \leq V(x) - \alpha_3(|x|) \end{aligned} \tag{A.20}$$

Where recall that a function is of class \mathcal{K}_∞ if it is continuous, strictly increasing, zero at zero, and is unbounded, and a function is positive definite if it is continuous and positive everywhere except at the origin. Function V is called *Lyapunov function*. Condition (iii) is named *descent property*. On the same basis we can define a *control Lyapunov function* (CLF). A function V is a CLF for $x(k+1) = f(x(k), u(k))$, where $u \in \mathbb{R}^m$, if there exists a control law $\kappa : \mathbb{R}^n \rightarrow \mathbb{R}^m$, such that V satisfies conditions (A.20) for the closed loop system $x(k+1) = f(x(k), \kappa(x))$.

A.6 Model Predictive Control

Model predictive control is a control technique widely employed in engineering problems. Probably, its main interesting feature is its capability to explicitly handle system constraints, i.e. constraints on its inputs, state and outputs, in the design of a control law. The following notes are mainly taken from [Rawlings 2009].

A.6.1 Concept and Implementation

A.6.1.1 Optimization Problem

MPC differs from those controllers whose control law is pre-computed off-line. Indeed, MPC implements an *implicit* control law. At first instance, MPC can be seen

as a practical way to treat a more difficult problem of *optimal control*. By defining $V_\infty(x_0, u) \triangleq \int_0^\infty l(x(t), u(t))dt$, where $x_0, x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, l is definite positive, the optimal control problem $\mathbb{P}(x_0)$ consists in solving the following problem for every initial condition x_0

$$\begin{aligned} & \min_u V_\infty(x_0, u) \\ & \text{subject to } \dot{x} = f(x, u); \quad x(0) = x_0 \\ & u \in \mathbb{U}, x \in \mathbb{X} \end{aligned} \tag{A.21}$$

The latter is in general too complex to be solved in its original formulation. Its solution, if it exists, is a function of time and state, thus describing a feedback control law. As a matter of fact (A.21) is an infinite-dimensional problem, as the optimization variable is a function of time. Moreover the considered time horizon is infinite, and $V_\infty(x_0, u)$ is in general nonconvex if $f(x, u)$ is nonlinear. Thus, in MPC one considers some approximations of the original optimal control problem, which allows to draw conclusions on the optimality of the solution as well as on the stability of the closed-loop system having as input the argument of the solution of a simplified version of $\mathbb{P}(x_0)$. In particular, in the MPC approach, a discrete time approximation of (A.21) is considered, thus leading to a finite-dimensional problem, the horizon is reduced to a finite one, and typically a *terminal region* and *cost* are added to the problem formulation in order to approximate the cost-to-go of the infinite horizon formulation. Thus, let us consider the discrete time system

$$x^+ = f(x, u), \quad x(0) = x_0$$

where $x \triangleq x(i)$, $u \triangleq u(i)$ are the state and input at current time i , and $x^+ \triangleq x(i+1)$ is the state at the next step. f is continuous and, without loss of generality, we consider $f(0, 0) = 0$. MPC is concerned with solving an optimization problem that depends on the system condition at the current time i , (x, i) . However, in the sequel we only consider time-invariant systems and time-invariant functions in the cost function. As a result, the optimization problem is also time-invariant, and we refer to it with the notation $\mathbb{P}_N(x)$, where x is the current state (x, i) . This can be formulated as follows

$$\begin{aligned} & \min_{\mathbf{u} \triangleq \{u(0), \dots, u(N-1)\}} V_N(x, \mathbf{u}) \triangleq \min_{\mathbf{u}} \sum_{k=0}^{N-1} l(x(k), u(k)) + V_f(x(N)) \\ & \text{subject to } x^+ = f(x, u) \\ & u(k) \in \mathbb{U}, \quad k = 0, \dots, N-1 \\ & x(k) \in \mathbb{X}, \quad k = 0, \dots, N-1 \\ & x(N) \in \mathbb{X}_f \end{aligned}$$

where N is the horizon length, V_f and \mathbb{X}_f are respectively the final term cost function and set. By naming $\phi(k, x, \mathbf{u})$ the system solution at time step k given the initial condition x , and the input control sequence $\mathbf{u} = \{u(0), \dots, u(k-1)\}$, it is

clear that the state at each time step in the horizon can be described as a function of x , and \mathbf{u} . For the same reason, the state constraints can be described as a function of the mentioned variables. Let us name $\mathcal{U}_N(x) \triangleq \{\mathbf{u} | (x, \mathbf{u}) \in \mathbb{Z}_N\}$, where $\mathbb{Z}_N \triangleq \{(x, \mathbf{u}) | u(k) \in \mathbb{U}, \phi(k, x, \mathbf{u}) \in \mathbb{X}, k = 0, \dots, N-1, \phi(N, x, \mathbf{u}) \in \mathbb{X}_f\}$, $\mathbb{P}_N(x)$ can be restated as

$$V_N^o(x) \triangleq \min_{\mathbf{u}} \{V_N(x, \mathbf{u}) | \mathbf{u} \in \mathcal{U}_N(x)\} \quad (\text{A.22})$$

The latter is a problem of *parametric optimization*, where \mathbf{u} is the optimization variable, and x the parameter. We additionally name $\mathcal{X}_N \triangleq \{x \in \mathbb{X} | \mathcal{U}_N(x) \neq \emptyset\}$, which is the state such that the couple (x, \mathbf{u}) belongs to \mathbb{Z}_N . For the existence of a solution to (A.22) the following assumption is usually made.

Assumption A.1 *Functions f , l , V_f are continuous, and $f(0, 0) = 0$, $l(0, 0) = 0$, $V_f(0) = 0$. \mathbb{X} is closed, \mathbb{U}, \mathbb{X}_f are compact, and each set contains the origin.*

Under such assumption, V_N is continuous on \mathbb{Z}_N and $\forall x \in \mathcal{X}_N$, $\mathcal{U}_N(x)$ is compact. Thus, for Weierstrass theorem it exists a solution to $\mathbb{P}_N(x)$, $\forall x \in \mathcal{X}_N$.

A.6.1.2 Receding Horizon Control

Given the system initial state x_0 , in the nominal case, i.e. if the system model is perfect and there is no disturbance, then, in principle, problem (A.22) could be solved once and off-line, providing the optimal *time-varying* control sequence $\mathbf{u}_N^o(x_0) = \{u^o(0, x_0), u^o(1, x(1)), \dots, u^o(N-1, x(N-1))\}$ bringing the initial state x_0 to the target set \mathbb{X}_f in N steps. Note that, even though the system and cost function are time-invariant, $\mathbf{u}_N^o(x_0)$ is *time-varying* as it associates a control action to both the system state and the time step. In such case, MPC realizes an *open-loop* control. Of course, assumptions on nominal system are not realistic in practical applications. This is why MPC is implemented in a *receding horizon* fashion. That is, problem (A.22) is solved at each time step, and only the *first* control action of the optimal control sequence $\mathbf{u}_N^o(x)$ is applied, i.e. $u^o(0, x)$, being x the current state. The process reiterates by solving problem (A.22) initialized in the new current *measured* (or more in general estimated) system state. As a consequence, the implicit control law $\kappa_N(x) \triangleq u^o(0, x)$ is *time-invariant*, and the control is in *closed-loop*.

A.6.2 Nominal Stability Results

In the following we state some basic results concerning the stability properties of nominal MPC, as they shed a light on the convergence property of the control algorithm. We do not provide any proof. The reader may refer to the mentioned reference for further details. The shown results are valid under the assumption that the solution to (A.22) is *globally optimal*, (see Section A.7). As it is well-known, since *optimality does not imply stability*, in the MPC framework the latter is attained via a proper choice of the MPC *ingredients*, namely, the horizon length, and the terminal cost function and set, V_f and \mathbb{X}_f . Stability analysis is performed by choosing a

Lyapunov function for the closed loop system

$$x^+ = f(x, \kappa_N(x)) \quad (\text{A.23})$$

i.e. a function satisfying Lyapunov conditions (A.20), (see Section A.5) in the region of attraction \mathcal{X}_N . A candidate Lyapunov function for (A.23) is the value function $V_N^o(x)$ defined in (A.22). First of all, for the unconstrained optimal control problem, such function satisfies the mentioned Lyapunov conditions if

$$V_f(f(x, u)) \leq V_f(x) - l(x, \kappa_f(x))$$

That is, the terminal cost function V_f is a global control Lyapunov function for the system to be controlled, i.e. there exists a control law $\kappa_f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that the system $x^+ = f(x, \kappa_f(x))$ is globally asymptotically stable. Although such condition implies global asymptotic stability for (A.23), this is attained at the price of finding a CLF for the system we want to stabilize. Thus, κ_f could be simply used instead of κ_N , although MPC could lead to better performance. As previously mentioned though, the interest in employing MPC is very often related to the possibility to treat constrained systems. For the latter, the basic stability assumption that allows $V_N^o(x)$ to satisfy condition (iii) of (A.20) $\forall x \in \mathcal{X}_N$ is the following

Assumption A.2

$$\min_{u \in \mathbb{U}} \{V_f(f(x, u)) + l(x, u) | f(x, u) \in \mathbb{X}_f\} \leq V_f(x) \quad \forall x \in \mathbb{X}_f$$

This assumption implies that \mathbb{X}_f is a controlled invariant set. As far as conditions (i), (ii) are concerned, the following assumption let them be satisfied by $V_N^o(x)$, $\forall x \in \mathcal{X}_N$.

Assumption A.3 \mathbb{X} is bounded. Moreover, there exist functions α_1 , and α_2 of class \mathcal{K}_∞ such that $l(x, u) \geq \alpha_1(|x|)$, $\forall x \in \mathcal{X}_N$, $\forall u \in \mathbb{U}$, and $V_f(x) \leq \alpha_2(|x|)$.

All in all, Assumptions A.1, A.2, A.3 let $V_N^o(x)$ satisfy Lyapunov conditions (A.20) $\forall x \in \mathcal{X}_N$.

Remark A.4 As shown, Assumption A.1 guarantees $\mathbb{P}_N(x)$ to have solution $\forall x \in \mathcal{X}_N$. However, this is not sufficient for a stronger property, according to which, if $\mathbb{P}_N(x)$ is solvable at the initial condition x_0 , then it has a feasible solution for any following step. This MPC feature is called recursive feasibility, and it is fundamental in the receding horizon technique, since in this case the control action is given by the solution of $\mathbb{P}_N(x)$. For the nominal constrained case, recursive feasibility is guaranteed by the fact that \mathbb{X}_f is control invariant.

A.7 Mathematical Optimization

An *optimization problem* can be described in the form

$$\begin{aligned} & \min_x f_0(x) \\ & \text{subject to } f_i(x) \leq 0, \quad i = 1, \dots, m \end{aligned} \tag{A.24}$$

where $x \in \mathbb{R}^n$ is the optimization variable, $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is the *cost* or *objective* function, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ are the *constraint* functions. *Linear* and *quadratic* programming (or optimization) are two important subclasses of (A.24). In the former, functions f_i , $i = 0, \dots, m$ are affine in the optimization variable, while in the latter f_0 is *quadratic*, i.e. of the form $f_0(x) = \frac{1}{2}x^\top Hx + c^\top x$, and f_i , $i = 1, \dots, m$ are affine.

A.7.1 Convex Optimization

Definition A.3 A set Ω is convex if the line segment between any two points in Ω lies in Ω , i.e. if for any $x_1, x_2 \in \Omega$ and any θ with $0 \leq \theta \leq 1$, we have $\theta x_1 + (1 - \theta)x_2 \in \Omega$.

An example of convex set is given by the *polyhedra*. A polyhedron \mathcal{P} in a n -dimensional space can be defined as the intersection of a finite number of halfspaces and hyperplanes via

$$\mathcal{P} \triangleq \left\{ x \in \mathbb{R}^n \mid a_j^\top x \leq b_j, j = 1, \dots, m, c_j^\top x = d_j, j = 1, \dots, p \right\}$$

A bounded polyhedron is called *polytope*.

Definition A.4 A function $f : \Omega \rightarrow \mathbb{R}$ is convex if Ω is convex and if for all $x_1, x_2 \in \Omega$, and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$$

On the basis of the two previous definitions, an optimization problem (A.24) is *convex* if functions f_i , $i = 0, \dots, m$ are convex. In general there exists no explicit analytic solution, but efficient methods and algorithms can be employed to solve it. These are typically *gradient-based* methods, i.e. they use gradient information of the cost function to provide directions in the search space along which the cost function decreases.

Definition A.5 A point x^* is a *local minimizer* of the cost function f_0 , if there is a neighborhood \mathcal{N} of x^* such that $f_0(x^*) \leq f_0(x) \forall x \in \mathcal{N}$.

From a practical point view, an important result concerning convex optimization is that *a local minimizer of a convex optimization problem is also global*.

A.7.2 Nonlinear Optimization

When the cost function f_0 or the constraint functions f_i , $i = 1, \dots, m$ are nonlinear and not known to be convex, we define (A.24) as a *nonlinear optimization* problem. In this case, there is no general method to solve it. One typical approach is to seek only local minimizers by adapting well-known algorithms for convex optimization. In this case there is no guarantee in finding the globally optimal solution. Moreover, these methods require an initial guess for the optimization variable, and this greatly influences the optimality of the solution, [Boyd 2004].

A.8 Matrix Lemmas

Given a square real matrix A , via $\sigma(A)$ we indicate its *spectrum*. We name $\mathbb{C}_{\geq 0} \triangleq \{a + ib : a \in \mathbb{R}_0^+, b \in \mathbb{R}, i^2 = -1\}$. I is the identity matrix. Given a rectangular matrix $B \in \mathbb{R}^{m \times n}$, by $\mathcal{R}(B)$ we indicate the *range* of matrix B , i.e. the *span* of the columns of B . By $\|B\|_\infty$ we indicate the \mathcal{H}_∞ norm of matrix B , given by $\max_{i=1, \dots, m} \sum_{j=1}^n |b_{ij}|$, where b_{ij} is the generic element of B . By $\mathbf{0}$ we indicate either a vector or a matrix of proper dimension according to the situation, and with all its entries equal to 0.

We introduce the two following lemmas.

Lemma A.1 *Given two symmetric matrices A , and B of equal dimension such that $A \succeq 0$, and $B \succ 0$; then $\sigma(AB) \subset \mathbb{C}_{\geq 0}$.*

Proof: It exists the symmetric matrix $B^{1/2} : B^{1/2} B^{1/2} = B$, where $B^{1/2} \succ 0$, and it exists $B^{-1/2} : B^{-1/2} B^{1/2} = B^{1/2} B^{-1/2} = I$. We have that $AB = AB^{1/2} B^{1/2} = B^{-1/2} (B^{1/2} A B^{1/2}) B^{1/2}$, thus AB is similar to $B^{1/2} A B^{1/2}$, i.e. $\sigma(AB) \equiv \sigma(B^{1/2} A B^{1/2})$. Moreover $B^{1/2} A B^{1/2} \succeq 0$ because $A \succeq 0$ and $B^{1/2}$ is symmetric. Thus $\sigma(B^{1/2} A B^{1/2}) \subset \mathbb{C}_{\geq 0}$. ■

Lemma A.2 *Given two symmetric matrices A , and B of equal dimension; if $A \succ 0$ then AB is diagonalizable in \mathbb{R} .*

Proof: It exists the symmetric matrix $A^{1/2} : A^{1/2} A^{1/2} = A$, where $A^{1/2} \succ 0$, and it exists $A^{-1/2} : A^{-1/2} A^{1/2} = A^{1/2} A^{-1/2} = I$. We have $A^{-1/2} A B A^{1/2} = A^{1/2} B A^{1/2}$, and the latter is symmetric. Thus AB is similar to a symmetric matrix, so it is diagonalizable in \mathbb{R} . ■

We remind the definition of *Kronecker product*. Given two matrices $A = [a_{ij}] \in \mathbb{R}^{m \times n}$, and $B \in \mathbb{R}^{p \times q}$, then their Kronecker product is

$$A \otimes B \triangleq \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp \times nq}$$

We recall the useful lemma on Kronecker product.

Lemma A.3 [*Graham 1981*] Suppose that $U \in \mathbb{R}^{p \times p}$, $V \in \mathbb{R}^{q \times q}$, $X \in \mathbb{R}^{p \times p}$, and $Y \in \mathbb{R}^{q \times q}$. The following hold: (i) $(U \otimes V)(X \otimes Y) = UX \otimes VY$; (ii) suppose U , and V invertible, then $(U \otimes V)^{-1} = U^{-1} \otimes V^{-1}$.

A.9 Gershgorin's Disc Theorem

Let $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, and let

$$R_i(A) \triangleq \sum_{j=1, j \neq i}^n |a_{ij}|, \quad i = 1, \dots, n$$

denote the deleted absolute row sums of A . Then all eigenvalues of A are located in the union of n discs

$$\bigcup_{i=1}^n \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i(A)\}$$

Additional Elements for Wind Turbines

B.1 dq Transformation

These notes are mainly taken from [Hamon 2010]. For the sake of simplicity we consider $p = 1$, thus $\omega_g = \omega_{eg}$, which is the electrical angular speed in the rotor winding. The electric quantities in the stator winding have angular speed ω_s . The voltage and current relations on rotor and stator side are described by the following equations

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = R_s \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{as} \\ \phi_{bs} \\ \phi_{cs} \end{bmatrix}, \quad \begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix} = R_r \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \phi_{ar} \\ \phi_{br} \\ \phi_{cr} \end{bmatrix}$$

where

$$\begin{bmatrix} \phi_{as} \\ \phi_{bs} \\ \phi_{cs} \end{bmatrix} = \mathbf{L}_s \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \mathbf{L}_m \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix}, \quad \begin{bmatrix} \phi_{ar} \\ \phi_{br} \\ \phi_{cr} \end{bmatrix} = \mathbf{L}_r \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} + \mathbf{L}_m^\top \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix}$$

where v , and i variables are the voltage and current quantities, subscripts a, b, c indicate the corresponding phase quantity, and subscripts s, r are used to indicate the stator and rotor quantities respectively. R_s , and R_r are the stator and rotor winding resistances, \mathbf{L}_s , and \mathbf{L}_r constant inductance matrices, and $\mathbf{L}_m(\theta_r(t))$ is the mutual inductance matrix depending on the angle describing the rotor electrical quantities position with respect to the fixed stator frame, i.e.

$$\theta_r(t) \triangleq \int_0^t \omega_g dt + \theta_r(0)$$

The dq -transformation is thus employed to eliminate the aforementioned dependency of matrix \mathbf{L}_m on θ_r , which is time-varying.

The $dq0$ -reference frame is a *rotating* frame with respect to the fixed stator frame, at angular speed ω_{dq} . Thus, the dq axis angular displacement with respect to the stator frame can be expressed as function of

$$\beta_{dq}(t) \triangleq \int_0^t \omega_{dq} dt + \beta_{dq}(0)$$

The dq -transformation is thus given by the following rotation matrix which transforms the abc -quantities in $dq0$ -quantities

$$T_{dq0}(\beta) \triangleq \frac{\sqrt{3}}{2} \begin{bmatrix} \cos(\beta) & \cos(\beta - \frac{2}{3}\pi) & \cos(\beta + \frac{2}{3}\pi) \\ -\sin(\beta) & -\sin(\beta - \frac{2}{3}\pi) & -\sin(\beta + \frac{2}{3}\pi) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (\text{B.1})$$

where β is set equal to

$$\beta_s(t) \triangleq \int_0^t \omega_{dq} dt + \beta_{dq}(0)$$

for the stator quantities of interest, and equal to

$$\beta_r(t) \triangleq \int_0^t (\omega_{dq} - \omega_g) dt + \beta_{dq}(0) - \theta_r(0)$$

for the rotor quantities of interest. Notice that for symmetrical abc -quantities, the 0-component in the $dq0$ -reference frame is zero because of the last row of (B.1). This is why the 0-component is often omitted.

B.2 DFIG Working Principle and Electrical Model

The following notes are based on [Poller 2003, Arifujaman 2010, Hamon 2010]. We provide the differential equations describing the electrical variables involved in the DFIG-based electric machine. These are presented in the dq frame, which is a rotating reference at the angular speed ω_{dq} with respect to the stationary stator reference frame. The transformation matrix allowing the change of coordinates in the dq frame is *power invariant* with respect to the chosen value of ω_{dq} . For details of the dq transformation see Section B.1. By proper choice of ω_{dq} , the dynamic system describing the stator and rotor voltages, (v_{ds}, v_{qs}) , (v_{dr}, v_{qr}) , and currents, (i_{ds}, i_{qs}) , (i_{dr}, i_{qr}) , is given by the following, [Arifujaman 2010]

$$\begin{cases} v_{ds} &= R_s i_{ds} + \dot{\phi}_{ds} - \omega_s \phi_{qs} \\ v_{qs} &= R_s i_{qs} + \dot{\phi}_{qs} + \omega_s \phi_{ds} \\ v_{dr} &= R_r i_{dr} + \dot{\phi}_{dr} - (\omega_s - \omega_{eg}) \phi_{qr} \\ v_{qr} &= R_r i_{qr} + \dot{\phi}_{qr} + (\omega_s - \omega_{eg}) \phi_{dr} \end{cases} \quad (\text{B.2})$$

where R_s , R_r are the stator and rotor resistances, (ϕ_{ds}, ϕ_{qs}) the dq -axis stator flux linkages, and (ϕ_{dr}, ϕ_{qr}) the dq -axis rotor flux linkages. The latter can be expressed as a function of the currents via, [Arifujaman 2010]

$$\begin{aligned} \phi_{ds} &= L_{ss} i_{ds} + L_m i_{dr} \\ \phi_{qs} &= L_{ss} i_{qs} + L_m i_{qr} \\ \phi_{dr} &= L_{rr} i_{dr} + L_m i_{ds} \\ \phi_{qr} &= L_{rr} i_{qr} + L_m i_{qs} \end{aligned} \quad (\text{B.3})$$

where $L_{ss} = L_s + L_m$, and $L_{rr} = L_r + L_m$, and where L_s , L_r are the stator and rotor inductance, and L_m the mutual inductance.

In the previous subsection we defined the electric power delivered to the grid as $P_g = T_g \omega_g$ if the electric power losses are neglected. If now these are taken into account, P_g can be more correctly described as

$$P_g = P_{loss} + P_{mag} + P_e$$

In other words, part of the active power P_g is lost in ohmic losses P_{loss} , and in magnetizing power P_{mag} . Indeed, more in general, P_g can be described as a function of the electric variables according to

$$P_g = (v_{ds}i_{ds} + v_{qs}i_{qs}) + (v_{dr}i_{dr} + v_{qr}i_{qr})$$

Thus, by employing (B.2), and a general value for ω_{dq} , we have (see [Hamon 2010])

$$\begin{aligned} P_g = & \underbrace{R_s(i_{ds}^2 + i_{qs}^2) + R_r(i_{dr}^2 + i_{qr}^2)}_{\text{ohmic losses}} \\ & + \underbrace{\dot{\phi}_{ds}i_{ds} + \dot{\phi}_{qs}i_{qs} + \dot{\phi}_{dr}i_{dr} + \dot{\phi}_{qr}i_{qr}}_{\text{magnetizing power}} \\ & + \underbrace{\omega_{dq}(\phi_{ds}i_{qs} - \phi_{qs}i_{ds}) + (\omega_{dq} - \omega_{eg})(\phi_{dr}i_{qr} - \phi_{qr}i_{dr})}_{\text{transferred electric power}} \end{aligned} \quad (\text{B.4})$$

Thus, the power injected into the grid P_e is given by the last term of (B.4), where we rename P_{stator} its first addend, and P_{rotor} its second one. P_{stator} , and P_{rotor} thus represent the power flowing through the stator and rotor winding respectively. Since, as mentioned, the expression is valid for any value of ω_{dq} , for a choice of $\omega_{dq} = 0$, and $\omega_{dq} = \omega_{eg}$, we have

$$\begin{aligned} \omega_{dq} = 0 & \Rightarrow P_e = \omega_{eg}(\phi_{qr}i_{dr} - \phi_{dr}i_{qr}) \\ \omega_{dq} = \omega_{eg} & \Rightarrow P_e = \omega_{eg}(\phi_{ds}i_{qs} - \phi_{qs}i_{ds}) \end{aligned}$$

from which we obtain the relationship

$$\phi_{ds}i_{qs} - \phi_{qs}i_{ds} = \phi_{qr}i_{dr} - \phi_{dr}i_{qr} \quad (\text{B.5})$$

Notice that relationship (B.5) can be also verified using (B.3). By employing P_e expression with $\omega_{dq} = \omega_s$, as well as relation (B.5) and the above definition of P_{stator} , and P_{rotor} , it yields

$$P_e = (1 - s)P_{stator}$$

B.3 Decoupling Active and Reactive Power

In DFIG-based WTs, the rotor is connected to the grid through two power electronic converters. The rotor-side one is usually operated in a stator-flux dq reference, which enables a power control decomposition in active component along the q -axis, and in reactive one along the d -axis. This is obtained under the following assumptions (see [Arifujjaman 2010])

- The stator resistance R_s is neglected.
- The grid is stiff, i.e. the stator voltage v_s , and ω_s are constant.
- The magnetizing current of the stator is determined by the grid.
- The q -axis is $\pi/2$ rad ahead with respect to the d -axis in the frame rotational sense, and $\omega_{dq} = \omega_s$.
- The stator flux vector is aligned with the stator d -axis.

The assumptions listed above result in

$$\begin{cases} v_{ds} &= 0 \\ v_{qs} &= v_s \\ \phi_{ds} &= \phi_s \\ \phi_{qs} &= 0 \end{cases} \quad (\text{B.6})$$

By plugging (B.6) in (B.2), (B.3), it first yields the reduced order system

$$\begin{cases} v_{dr} &= R_r i_{dr} + \left(L_{rr} - \frac{L_m^2}{L_{ss}} \right) \frac{di_{dr}}{dt} - (\omega_s - \omega_{eg}) \left(L_{rr} - \frac{L_m^2}{L_{ss}} \right) i_{qr} \\ v_{qr} &= R_r i_{qr} + \left(L_{rr} - \frac{L_m^2}{L_{ss}} \right) \frac{di_{qr}}{dt} + (\omega_s - \omega_{eg}) \left(\left(L_{rr} - \frac{L_m^2}{L_{ss}} \right) i_{dr} + \frac{L_m v_s}{L_{ss} \omega_s} \right) \end{cases}$$

from which we see that i_{dr} , and i_{qr} can be controlled by acting on v_{dr} , and v_{qr} . Secondly, the active and reactive power of the stator, P_s , and Q_s reduce to

$$\begin{cases} P_s &= -\frac{L_m v_s}{L_{ss}} i_{qr} \\ Q_s &= \frac{v_s^2}{\omega_s L_{ss}} - \frac{L_m v_s}{L_{ss}} i_{dr} \end{cases}$$

Thus, i_{qr} can be used to regulate P_s , and i_{dr} to regulate Q_s .

Similar results hold for the grid-side converter, which is operated in an ac-voltage dq reference frame, enabling a decoupling control for active and reactive power. Mathematical computation are not reported here though. All in all, reactive power control is possible through the d -axis component of the rotor-side and grid-side converters.

B.4 Residual due to Wind Disturbance

Functions μ_1 , μ_2 , and μ_3 are obtained by considering a first order Taylor approximation in v_m for all the functions appearing in the closed-loop WT dynamics. In this way we can make explicit, in an approximated way, the presence of v_d , which in turns does not allow perfect cancellations via FL, and ultimately makes appear a residual term in the closed-loop system, i.e.

$$\dot{\varepsilon}_p = -b_0 \varepsilon_p + \mu_1 v_d + \mu_2 v_d^2 + \mu_3 \dot{v}_d$$

We do not report the computation. However the obtained functions are

$$\begin{aligned}\mu_1 &\triangleq -\frac{\omega_r}{J_t} \left(\frac{\partial T_r}{\partial v} \right)^2 - \omega_r \frac{\partial \dot{T}_r}{\partial v} + 2 \frac{\omega_r}{J_t} (K_t - a_0 J_t) \frac{\partial T_r}{\partial v} + a_0 \omega_r \frac{\partial T_r}{\partial v} - \frac{T_r}{J_t} \frac{\partial T_r}{\partial v} \\ \mu_2 &\triangleq -\frac{1}{J_t} \left(\frac{\partial T_r}{\partial v} \right)^2 \\ \mu_3 &\triangleq -\omega_r \frac{\partial T_r}{\partial v}\end{aligned}$$

where functions T_r , $\frac{\partial T_r}{\partial v}$, and $\frac{\partial \dot{T}_r}{\partial v}$ are computed in the system variables and in v_m .

Graph Theory Preliminaries

C.1 Basic Notions and Definitions

A directed graph \mathcal{G} , called *digraph*, is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of unordered pairs of nodes, named edges. Two nodes i, j are said to be adjacent if $(i, j) \in \mathcal{E}$. In such case the communication is supposed to be directed from i to j . The *weighted* adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ associated with the digraph \mathcal{G} , is defined by $a_{ii} = 0$, i.e. self-loops are not allowed, and $a_{ij} > 0$ if $(i, j) \in \mathcal{E}$. The Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N \times N}$ is defined as $l_{ii} = \sum_{j \neq i} a_{ij}$ and $l_{ij} = -a_{ij}$, $i \neq j$. Typically, if the adjacency matrix is not weighted, then we simply assign $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$. Moreover, under the assumption of *undirected* graph, $(i, j) \in \mathcal{E}$ implies that $(j, i) \in \mathcal{E}$ too. In this report an undirected graph is always considered to be not weighted. An undirected graph is connected if there exists a path between every pair of distinct nodes, otherwise it is disconnected. If there exist an edge between any two nodes, the graph is said to be complete. We provide the following basic notions

- \mathcal{N}_i identifies the set of neighbors of node i , i.e. $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$.
- Given two nodes i , and j , the distance $\text{dist}(i, j)$ is the length of the shortest path between i and j .
- The *eccentricity* $\text{ecc}(i)$ of a node i is the greatest distance between i and any other $j \in \mathcal{V}$.
- The radius $r(\mathcal{G})$ of a graph is the minimum eccentricity of any node, i.e. $r(\mathcal{G}) \triangleq \min_i \text{ecc}(i)$.
- The diameter $d(\mathcal{G})$ of a graph is the maximum eccentricity of any node in the graph, i.e. $d(\mathcal{G}) \triangleq \max_{i,j} \text{dist}(i, j)$.

C.2 Lemmas

Lemma C.1 [*Ren 2005*] *The Laplacian matrix has the following properties: (i) if \mathcal{A} refers to an undirected graph, then \mathcal{L} is symmetric and all its eigenvalues are either strictly positive or equal to 0, and $\mathbf{1}$ is the corresponding eigenvector to 0; (ii) 0 is a simple eigenvalue of \mathcal{L} if and only if the graph is connected.*

Lemma C.2 [*Lin 2008*] Let $\bar{\mathcal{L}} = [\bar{l}_{ij}] \in \mathbb{R}^{N \times N}$ be a Laplacian matrix such that $\bar{l}_{ij} = N-1/N$ if $i = j$, and $\bar{l}_{ij} = -1/N$ otherwise, then the following hold: (i) the eigenvalues of $\bar{\mathcal{L}}$ are 1 with multiplicity $N - 1$, and 0 with multiplicity 1. $\mathbf{1}^\top$ and $\mathbf{1}$ are respectively the left and right eigenvector associated to eigenvalue 0; (ii) there exists an orthogonal matrix $U \in \mathbb{R}^{N \times N}$, i.e. $U : U^\top U = UU^\top = I$, and whose last column is equal to $\mathbf{1}/\sqrt{N}$, such that for any Laplacian matrix \mathcal{L} associated to any undirected graph we have

$$\begin{aligned} U^\top \bar{\mathcal{L}} U &= \begin{bmatrix} I_{N-1} & 0_{(N-1) \times 1} \\ 0_{1 \times (N-1)} & 0 \end{bmatrix} \triangleq \bar{\Lambda}, \\ U^\top \mathcal{L} U &= \begin{bmatrix} \mathcal{L}_1 & 0_{(N-1) \times 1} \\ 0_{1 \times (N-1)} & 0 \end{bmatrix} \end{aligned}$$

where $\mathcal{L}_1 \in \mathbb{R}^{(N-1) \times (N-1)}$ is symmetric and positive definite if the graph is connected.

Moreover we deduce the following extension of Lemma C.2.

Lemma C.3 Let $\mathcal{L} \in \mathbb{R}^{N \times N}$ be the Laplacian matrix associated to an undirected connected graph, and let $D \in \mathbb{R}^{N \times N} \succ 0$, and symmetric, then the following hold: (i) $\hat{\mathcal{L}} \triangleq D\mathcal{L} \succeq 0$, all its eigenvalues are real, and 0 is a simple eigenvalue with associated eigenvector $\mathbf{1}$; (ii) consider the orthogonal matrix $U \in \mathbb{R}^{N \times N}$ defined in Lemma C.2, then

$$U^\top \hat{\mathcal{L}} U = \begin{bmatrix} \hat{\mathcal{L}}_1 & 0_{(N-1) \times 1} \\ * & 0 \end{bmatrix}$$

where $\hat{\mathcal{L}}_1 \in \mathbb{R}^{(N-1) \times (N-1)} \succ 0$, and its eigenvalues are real.

Proof: We have that $D\mathcal{L} = D^{1/2}(D^{1/2}\mathcal{L}D^{1/2})D^{-1/2}$, thus $D\mathcal{L}$ is similar to a symmetric semi-definite positive matrix, so its eigenvalues are positive real. $\hat{\mathcal{L}}$ preserves the 0 eigenvalue, and its associated eigenvector $\mathbf{1}$, as $D\mathcal{L}\mathbf{1} = \mathbf{0}$. 0 is a simple eigenvalue as D is nonsingular, and \mathcal{L} has one simple 0 eigenvalue by hypothesis. The last column of $U^\top \hat{\mathcal{L}} U$ has all its entries equal to 0 because the last column of U is $\mathbf{1}/\sqrt{N}$. Being $U^\top \hat{\mathcal{L}} U$ block triangular, and similar to $\hat{\mathcal{L}}$, $\hat{\mathcal{L}}_1$ has all real strictly positive eigenvalues. \blacksquare

LMI-based Design

D.1 \mathcal{H}_∞ Norm of a System

In the following we recall the \mathcal{H}_∞ -norm definition for a *continuous-time* LTI system. A similar definition holds for discrete-time LTI systems.

Consider the following LTI MIMO system

$$\begin{cases} \dot{x} &= Ax + Bu \\ y &= Cx \end{cases} \quad (\text{D.1})$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$. System (D.1) transfer matrix is $G(s) \triangleq C(sI - A)^{-1}B$. Its *singular values* are defined as the square roots of eigenvalues λ_i of $G(j\omega)G^*(j\omega)$, i.e.

$$\sigma_i(G(j\omega)) \triangleq \sqrt{\lambda_i(G(j\omega)G^*(j\omega))}$$

Moreover $\sigma_i \geq 0 \ \forall \omega$, and by naming $\bar{\sigma}(G(j\omega))$, and $\underline{\sigma}(G(j\omega))$, respectively the largest and the smallest singular value of $G(s) \ \forall \omega$, we have

$$\underline{\sigma}(G(j\omega)) \leq \frac{\|G(j\omega)U(j\omega)\|_2}{\|U(j\omega)\|_2} \leq \bar{\sigma}(G(j\omega)) \quad (\text{D.2})$$

where $U(j\omega)$ is the Laplace transform of u , computed in $j\omega$, and $\|\cdot\|_2$ indicates the \mathcal{L}_2 -norm. Relationship (D.2) means that the frequency gain of a system lies between the smallest and the largest singular values. The \mathcal{H}_∞ -norm of a system represents the maximal possible amplification between the input u and the output y . This can be mathematically defined by employing the aforementioned singular values: Thus, the \mathcal{H}_∞ -norm of system (D.1), $\|G(s)\|_\infty$ is

$$\|G(s)\|_\infty \triangleq \sup_{\omega} \bar{\sigma}(G(j\omega))$$

For the discrete-time case, given a transfer matrix $G(z)$, $\|G(z)\|_\infty$ is given by

$$\|G(z)\|_\infty \triangleq \sup_{\omega \in [0, 2\pi]} \bar{\sigma}(G(e^{j\omega}))$$

D.2 Bounded Real Lemma

In this section we report two of the equivalents set by the Bounded Real Lemma. Given $\gamma \in \mathbb{R}^+$, and the system

$$\begin{cases} \dot{x} &= Ax + Bu \\ y &= Cx \end{cases}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$. The following statements are equivalent

- It exists $P \in \mathbb{R}^{n \times n} : P = P^\top \succ 0$ such that

$$2x^\top P(Ax + Bu) < \gamma^2 u^\top u - y^\top y$$

- $\|T_{uy}(s)\|_\infty < \gamma$, being $T_{uy}(s) \triangleq C(sI - A)^{-1}B$.

Similar results hold for the discrete-time case for which we have that, given $\gamma \in \mathbb{R}^+$, and the system

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{cases}$$

- It exists $P \in \mathbb{R}^{n \times n} : P = P^\top \succ 0$ such that

$$x^\top(k+1)Px(k+1) - x^\top(k)x(k) < \gamma^2 u^\top(k)u(k) - y^\top(k)y(k)$$

- $\|T_{uy}(z)\|_\infty < \gamma$, being $T_{uy}(z) \triangleq C(zI - A)^{-1}B$.

D.3 Robust Control for Parametric Polytopic Uncertainties

For the sake of simplicity we recall classic results of robust stability for an autonomous linear system. Let us consider the following continuous-time system

$$\dot{x} = A(\theta)x \tag{D.3}$$

where $x \in \mathbb{R}^n$, and $\theta \in \mathbb{R}^p$ is an unknown parameter whose components θ_i , $i = 1, \dots, p$ are known to be confined in the interval $[\theta_i, \bar{\theta}_i]$. Thus θ is confined in a known polytopic set whose vertexes are π_i , $i = 1, \dots, 2^p$, i.e. for a given value of θ , there exist corresponding α_i , $i = 1, \dots, 2^p$ such that

$$\begin{aligned} \theta &= \sum_{i=1}^{2^p} \alpha_i \pi_i \\ \text{where } \sum_{i=1}^{2^p} \alpha_i &= 1, \quad \alpha_i \geq 0 \end{aligned} \tag{D.4}$$

If $A(\theta)$ is *affine* in θ then, for every set value of θ , it can be described as a *convex combination* of matrices computed at vertexes π_i , $A_i(\pi_i)$, by using the same weights α_i corresponding to θ , given in (D.4). In other words, by naming $\alpha = \text{col}(\alpha_1, \dots, \alpha_{2^p})$, $A(\theta)$ can be equivalently described as a function of α , i.e. $A(\alpha)$ belonging to

$$\mathbf{A} \triangleq \left\{ A(\alpha) \mid A(\alpha) = \sum_{i=1}^{2^p} \alpha_i A_i, \quad \sum_{i=1}^{2^p} \alpha_i = 1, \quad \alpha_i \geq 0 \right\}$$

In the sequel we will refer to system (D.3) by making explicit its dependency on α , i.e.

$$\dot{x} = A(\alpha)x \quad (\text{D.5})$$

System (D.5) is known to be *robustly stable* with respect to all the admissible values of α , if there exists a Lyapunov matrix $P \in \mathbb{R}^{n \times n}$, such that $P = P^\top \succ 0$, and

$$A_i^\top P + P A_i < 0, \quad i = 1, \dots, 2^p \quad (\text{D.6})$$

Note that such condition is quite conservative as it employs *one* Lyapunov matrix P to solve (D.6), for $i = 1, \dots, 2^p$. Although less conservative solutions exist for the continuous-time case, we will treat them for the discrete-time one. Consider the following system

$$x(k+1) = A(\alpha)x(k) \quad (\text{D.7})$$

Authors of [de Oliveira 1999] showed that (D.7) is robustly stable if there exist matrices $P_i \in \mathbb{R}^{n \times n} : P_i = P_i^\top \succ 0$, and a matrix $G \in \mathbb{R}^{n \times n}$ such that

$$\begin{bmatrix} P_i & A_i^\top G^\top \\ G A_i & G + G^\top - P_i \end{bmatrix} \succ 0, \quad i = 1, \dots, 2^p \quad (\text{D.8})$$

The key point in the stability proof is that inequality (D.8) is *linear* on P_i and A_i , and this allows to consider the following *parameter-dependent* Lyapunov matrix to establish stability.

$$P(\alpha) = \sum_{i=1}^{2^p} \alpha_i P_i$$

This technique enables a less conservative solution thanks to the introduced extra degree of freedom represented by G .

The aforementioned results can be applied to more general control design problems, e.g. the one of finding a controller to impose a desired closed-loop \mathcal{H}_∞ gain. Again, the key point that allows to consider a parameter-dependent Lyapunov matrix to conclude on stability is that the matrix inequalities resulting from the according considered control problem have to be *affine* in the system matrices at the vertexes of the polytope, in the problem variables, and in P_i , [De Oliveira 2002].

D.4 PID for MIMO LTI Discrete-time Systems

In the following we report the two cited theorems of [Wu 2011]. For their proof please refer to the mentioned reference. Both theorems are concerned with the PID controller design for MIMO LTI discrete-time systems, based on the solution of given LMIs. In particular, the first theorem is based on imposing an \mathcal{H}_∞ norm to the closed-loop system, while the second one is concerned with constraining the closed-loop system eigenvalues in a given region of the complex plane. This is explained in

the following.

Consider the system of equations

$$\begin{cases} x^+ = Ax + B_1\omega + B_2u \\ z = C_1x, \quad y = C_2x \end{cases} \quad (\text{D.9})$$

where $A \in \mathbb{R}^{n \times n}$, $B_2 \in \mathbb{R}^{n \times l}$, $B_1 \in \mathbb{R}^{n \times h}$, $C_1 \in \mathbb{R}^{r \times n}$, $C_2 \in \mathbb{R}^{m \times n}$, $x \triangleq x(k) \in \mathbb{R}^n$ and $x^+ \triangleq x(k+1) \in \mathbb{R}^n$ are respectively the system state at the current step k , and at the next step $k+1$, $u \triangleq u(k) \in \mathbb{R}^l$ is the control input, $\omega \triangleq \omega(k) \in \mathbb{R}^h$ is an exogenous input signal, $z \triangleq z(k) \in \mathbb{R}^r$ the controlled output, and $y \triangleq y(k) \in \mathbb{R}^m$ is the measured one. Define the matrices $C_d \triangleq [C_1 \quad 0_{r \times (2l)}]$, $\tilde{B} \triangleq [B_1^\top \quad 0_{h \times (2l)}]^\top$, $K \triangleq [D_c^\top \quad B_c^\top]^\top$, and

$$\tilde{A} \triangleq \begin{bmatrix} A & B_2C_c \\ 0_{2l \times n} & A_c \end{bmatrix}$$

where A_c , B_c , C_c , and D_c have the following defined structure

$$\begin{aligned} A_c &= \begin{bmatrix} I_l & I_l \\ 0_{l \times l} & 0_{l \times l} \end{bmatrix}_{2l \times 2l} & B_c &= \begin{bmatrix} (K_i - K_d) \\ K_d \end{bmatrix}_{2l \times m} \\ C_c &= [I_l \quad 0_{l \times l}]_{l \times 2l} & D_c &= [(K_p + K_i + K_d)]_{l \times m} \end{aligned}$$

where $K_p = [k_{p,ij}]$, $K_i = [k_{i,ij}]$, $K_d = [k_{d,ij}] \in \mathbb{R}^{l \times m}$. Note that the system having the aforesaid matrices (A_c, B_c, C_c, D_c) has transfer matrix equal to

$$\begin{aligned} C_c(zI - A_c)^{-1}B_c + D_c &= K_p + K_i \frac{z}{z-1} + K_d \frac{z-1}{z} = \\ &= \begin{bmatrix} \left(k_{p,11} + k_{i,11} \frac{z}{z-1} + k_{d,11} \frac{z-1}{z} \right) & \cdots & \left(k_{p,1m} + k_{i,1m} \frac{z}{z-1} + k_{d,1m} \frac{z-1}{z} \right) \\ \vdots & \ddots & \vdots \\ \left(k_{p,l1} + k_{i,l1} \frac{z}{z-1} + k_{d,l1} \frac{z-1}{z} \right) & \cdots & \left(k_{p,lm} + k_{i,lm} \frac{z}{z-1} + k_{d,lm} \frac{z-1}{z} \right) \end{bmatrix} \end{aligned}$$

Assuming B_2 to be of full column rank without loss of generality, there exists an invertible $T_b \in \mathbb{R}^{n \times n}$: $T_b B_2 = [0_{l \times (n-l)} \quad I_{l \times l}]^\top$. Finally define

$$T \triangleq \begin{bmatrix} T_b & 0_{n \times 2l} \\ 0_{2l \times n} & I_{2l \times 2l} \end{bmatrix}$$

Thus, we have the following theorems

Theorem D.1 Consider system (D.9). If there exists a positive definite matrix $P \in \mathbb{R}^{\bar{n} \times \bar{n}}$, where $\bar{n} \triangleq n + 2l$, matrices

$$F = \begin{bmatrix} F_{11} & 0_{(\bar{n}-g) \times 3l} \\ F_{21} & F_{22} \end{bmatrix}$$

$F_{22} \in \mathbb{R}^{q \times 3l}$, $1 \leq q \leq 3l$, $G_1 \triangleq [G_{11} \ 0] \in \mathbb{R}^{\bar{n} \times \bar{n}}$, $G_{11} \in \mathbb{R}^{\bar{n} \times (\bar{n}-3l)}$, $G_2 \triangleq [G_{21} \ 0] \in \mathbb{R}^{h \times \bar{n}}$, $G_{21} \in \mathbb{R}^{h \times (\bar{n}-3l)}$, $G_3 \triangleq [G_{31} \ 0] \in \mathbb{R}^{r \times \bar{n}}$, $G_{31} \in \mathbb{R}^{r \times (\bar{n}-3l)}$, $H_1 \in \mathbb{R}^{\bar{n} \times r}$, $H_2 \in \mathbb{R}^{\bar{n} \times r}$, $H_3 \in \mathbb{R}^{h \times r}$, $H_4 \in \mathbb{R}^{r \times r}$, $Y \in \mathbb{R}^{q \times m}$, and

$$N_1 = \begin{bmatrix} 0_{(\bar{n}-q) \times n} & 0_{(\bar{n}-q) \times 2l} \\ YC_2 & 0_{q \times 2l} \end{bmatrix}$$

and we further name $\Psi_{11} \triangleq P - FT - (FT)^\top$, $\Psi_{21} \triangleq N_1^\top + (FT\tilde{A})^\top - G_1T + (H_1C_{cl})^\top$, $\Psi_{22} \triangleq -P + G_1T\tilde{A} + (G_1T\tilde{A})^\top + H_2C_{cl} + (H_2C_{cl})^\top$, $\Psi_{31} \triangleq (FT\tilde{B})^\top - G_2T$, $\Psi_{32} \triangleq G_2T\tilde{A} + H_3C_{cl} + (G_1T\tilde{B})^\top$, $\Psi_{33} \triangleq -\gamma^2 I + G_2T\tilde{B} + (G_2T\tilde{B})^\top$, $\Psi_{41} \triangleq -G_3T - H_1^\top$, $\Psi_{42} \triangleq G_3T\tilde{A} + H_4C_{cl} - H_2^\top$, $\Psi_{43} \triangleq G_3T\tilde{B} - H_3^\top$, and $\Psi_{44} \triangleq I - H_4 - H_4^\top$, such that the following LMI has a solution

$$\begin{bmatrix} \Psi_{11} & * & * & * \\ \Psi_{21} & \Psi_{22} & * & * \\ \Psi_{31} & \Psi_{32} & \Psi_{33} & * \\ \Psi_{41} & \Psi_{42} & \Psi_{43} & \Psi_{44} \end{bmatrix} < 0$$

where $*$ indicates the transposed corresponding element in the subdiagonal part of the matrix, and if exists K such that $F_{22}K = Y$, then the \mathcal{H}_∞ norm of the closed-loop system given by (D.9) and

$$\begin{cases} x_c^+ = A_c x_c + B_c y \\ u = C_c x_c + D_c y \end{cases}$$

satisfies $\|T_{zw}\|_\infty < \gamma$.

Theorem D.2 Consider system (D.9). If there exists a positive definite matrix $P \in \mathbb{R}^{\bar{n} \times \bar{n}}$, and a matrix

$$J = \begin{bmatrix} J_{11} & 0_{(\bar{n}-q) \times 3l} \\ J_{21} & J_{22} \end{bmatrix}$$

$J_{22} \in \mathbb{R}^{3l \times 3l}$, and $X \in \mathbb{R}^{3l \times m}$, and we further name

$$\Omega \triangleq \begin{bmatrix} 0_{(\bar{n}-3l) \times n} & 0_{(\bar{n}-3l) \times 2l} \\ XC_2 & 0_{3l \times 2l} \end{bmatrix}$$

such that the following LMI has a solution

$$\begin{bmatrix} bP & * \\ \Omega + JT\tilde{A} + aJT & b(JT + (JT)^\top - P) \end{bmatrix} > 0$$

and if J is nonsingular, then by choosing $K = J_{22}^{-1}X$, the eigenvalues of the following matrix

$$A_{cl} \triangleq \begin{bmatrix} (A + B_2D_cC_2) & B_2C_c \\ B_cC_2 & A_c \end{bmatrix}$$

lie in the region $\mathcal{F}_D \triangleq \{(\Re[\lambda], \Im[\lambda]) : (\Re[\lambda] + a)^2 + \Im[\lambda]^2 < b^2\}$.

Bibliography

- [Abdullah 2012] Majid A Abdullah, AHM Yatim, CW Tan and R Saidur. *A review of maximum power point tracking algorithms for wind energy systems*. Renewable and sustainable energy reviews, vol. 16, no. 5, pages 3220–3227, 2012. (Cited on pages 28 and 39).
- [Abido 2002] MA Abido. *Optimal power flow using particle swarm optimization*. International Journal of Electrical Power & Energy Systems, vol. 24, no. 7, pages 563–571, 2002. (Cited on page 114).
- [Ackermann 2005] Thomas Ackermann. Wind power in power systems. John Wiley & Sons, 2005. (Cited on page 28).
- [Aguirre 2007] A Hernandez Aguirre, A Muñoz Zavala, E Villa Diharce and S Botello Rionda. *COPSO: Constrained Optimization via PSO algorithm*. Center for Research in Mathematics (CIMAT). Technical report No. I-07-04/22-02-2007, 2007. (Cited on pages 77, 78, 79, 81, 85, 90 and 91).
- [Akat 2008] S Burak Akat and Veysel Gazi. *Decentralized asynchronous particle swarm optimization*. In Swarm Intelligence Symposium, 2008. SIS 2008. IEEE, pages 1–8. IEEE, 2008. (Cited on page 94).
- [Ali 2014] MM Ali, Mohsen Golalikhani and Jun Zhuang. *A computational study on different penalty approaches for solving constrained global optimization problems with the electromagnetism-like method*. Optimization, vol. 63, no. 3, pages 403–419, 2014. (Cited on page 80).
- [Andersson 2008] Göran Andersson. *Modelling and analysis of electric power systems*. ETH Zurich, september, 2008. (Cited on pages 114 and 119).
- [Arifujjaman 2010] Md Arifujjaman, MT Iqbal and John E Quaicoe. *Vector control of a DFIG based wind turbine*. Istanbul University-Journal of Electrical & Electronics Engineering; Vol 9, No 2 (2009); 1057-1066, 2010. (Cited on pages ix, 7, 29, 36, 38, 232 and 233).
- [Atashpendar 2016] Arash Atashpendar, Bernabé Dorronsoro, Grégoire Danoy and Pascal Bouvry. *A parallel cooperative coevolutionary SMPSO algorithm for multi-objective optimization*. In High Performance Computing & Simulation (HPCS), 2016 International Conference on, pages 713–720. IEEE, 2016. (Cited on page 95).
- [Barakati 2009] S Masoud Barakati, Mehrdad Kazerani and J Dwight Aplevich. *Maximum power tracking control for a wind turbine system including a matrix converter*. IEEE Transactions on Energy Conversion, vol. 24, no. 3, pages 705–713, 2009. (Cited on page 39).

- [Baros 2017] Stefanos Baros and Marija D Ilic. *Distributed Torque Control of De-loaded Wind DFIGs for Wind Farm Power Output Regulation*. IEEE Transactions on Power Systems, 2017. (Cited on pages [xi](#), [10](#), [42](#), [190](#) and [191](#)).
- [Baroudi 2007] Jamal A Baroudi, Venkata Dinavahi and Andrew M Knight. *A review of power converter topologies for wind generators*. Renewable Energy, vol. 32, no. 14, pages 2369–2385, 2007. (Cited on pages [vi](#) and [5](#)).
- [Barreiro-Gomez 2015] Julian Barreiro-Gomez, Carlos Ocampo-Martinez, F Bianchi and Nicanor Quijano. *Model-free control for wind farms using a gradient estimation-based algorithm*. In Control Conference (ECC), 2015 European, pages 1516–1521. IEEE, 2015. (Cited on page [130](#)).
- [Bhutta 2012] Muhammad Mahmood Aslam Bhutta, Nasir Hayat, Ahmed Uzair Farooq, Zain Ali, Sh Rehan Jamil and Zahid Hussain. *Vertical axis wind turbine—A review of various configurations and design techniques*. Renewable and Sustainable Energy Reviews, vol. 16, no. 4, pages 1926–1939, 2012. (Cited on pages [vi](#) and [4](#)).
- [Biegel 2013] Benjamin Biegel, Daria Madjidian, Vedrana Spudic, Anders Rantzer and Jakob Stoustrup. *Distributed low-complexity controller for wind power plant in derated operation*. In Control Applications (CCA), 2013 IEEE International Conference on, pages 146–151. IEEE, 2013. (Cited on pages [xi](#), [10](#), [131](#), [190](#) and [191](#)).
- [Boukhezzar 2005] Boubekeur Boukhezzar and Houria Siguerdidjane. *Nonlinear control of variable speed wind turbines for power regulation*. In Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on, pages 114–119. IEEE, 2005. (Cited on pages [x](#), [8](#), [29](#), [191](#) and [195](#)).
- [Boukhezzar 2006a] Boubekeur Boukhezzar. *Sur les stratégies de commande pour l’optimisation et la régulation de puissance des éoliennes à vitesse variable*. PhD thesis, Université Paris Sud-Paris XI, 2006. (Cited on pages [ix](#), [7](#), [33](#) and [35](#)).
- [Boukhezzar 2006b] Boubekeur Boukhezzar, Houria Siguerdidjane and M Maureen Hand. *Nonlinear control of variable-speed wind turbines for generator torque limiting and power optimization*. Journal of solar energy engineering, vol. 128, no. 4, pages 516–530, 2006. (Cited on pages [x](#) and [8](#)).
- [Boukhezzar 2007] Boubekeur Boukhezzar, L Lupu, Houria Siguerdidjane and M Hand. *Multivariable control strategy for variable speed, variable pitch wind turbines*. Renewable Energy, vol. 32, no. 8, pages 1273–1287, 2007. (Cited on pages [x](#), [8](#) and [29](#)).
- [Boukhezzar 2009] Boubekeur Boukhezzar and Houria Siguerdidjane. *Nonlinear control with wind estimation of a DFIG variable speed wind turbine for power*

- capture optimization*. Energy Conversion and Management, vol. 50, no. 4, pages 885–892, 2009. (Cited on page 38).
- [Boukhezzar 2010] Boubekur Boukhezzar and Houria Siguerdidjane. *Comparison between linear and nonlinear control strategies for variable speed wind turbines*. Control Engineering Practice, vol. 18, no. 12, pages 1357–1368, 2010. (Cited on pages ix, 8, 29 and 30).
- [Boukhezzar 2011] Boubekur Boukhezzar and Houria Siguerdidjane. *Nonlinear control of a variable-speed wind turbine using a two-mass model*. IEEE Transactions on Energy Conversion, vol. 26, no. 1, pages 149–162, 2011. (Cited on pages 29, 33, 54 and 56).
- [Boyd 2004] Stephen Boyd and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004. (Cited on pages 82, 101 and 229).
- [Burkart 2011] Ralph Burkart, Kostas Margellos and John Lygeros. *Nonlinear control of wind turbines: An approach based on switched linear systems and feedback linearization*. In Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 5485–5490. IEEE, 2011. (Cited on pages 8, 29 and 33).
- [Burton 2011] Tony Burton, Nick Jenkins, David Sharpe and Ervin Bossanyi. Wind energy handbook. John Wiley & Sons, 2011. (Cited on pages vi, 4 and 33).
- [Cabrera 2007] Juan C Fuentes Cabrera and Carlos A Coello Coello. *Handling constraints in particle swarm optimization using a small population size*. In Mexican International Conference on Artificial Intelligence, pages 41–51. Springer, 2007. (Cited on page 79).
- [Calazan 2013] Rogerio De Moraes Calazan, Nadia Nedjah and Luiza De Macedo Mourelle. *A cooperative parallel particle swarm optimization for high-dimension problems on GPUs*. In Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on, pages 356–361. IEEE, 2013. (Cited on page 95).
- [Camblong 2012] Haritza Camblong, Said Nourdine, Ionel Vechiu and Gerardo Tapia. *Control of wind turbines for fatigue loads reduction and contribution to the grid primary frequency regulation*. Energy, vol. 48, no. 1, pages 284–291, 2012. (Cited on page 30).
- [Cao 2009] Yongcan Cao, Wei Ren and Yan Li. *Distributed discrete-time coordinated tracking with a time-varying reference state and limited communication*. Automatica, vol. 45, no. 5, pages 1299–1305, 2009. (Cited on page 172).
- [Cao 2012] Yongcan Cao and Wei Ren. *Distributed coordinated tracking with reduced interaction via a variable structure approach*. IEEE Transactions on Automatic Control, vol. 57, no. 1, pages 33–48, 2012. (Cited on page 180).

- [Carli 2008] Ruggero Carli, Alessandro Chiuso, Luca Schenato and Sandro Zampieri. *A PI consensus controller for networked clocks synchronization*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 10289–10294, 2008. (Cited on page 172).
- [Carlin 2003] Palmer W Carlin, Alan S Laxson and EB Muljadi. *The history and state of the art of variable-speed wind turbine technology*. Wind Energy, vol. 6, no. 2, pages 129–159, 2003. (Cited on pages vi and 4).
- [Carpentier 1962] J Carpentier. *Contribution to the economic dispatch problem*. Bulletin de la Societe Francoise des Electriciens, vol. 3, no. 8, pages 431–447, 1962. (Cited on page 113).
- [Chang 2014] Tsung-Hui Chang, Angelia Nedic and Anna Scaglione. *Distributed constrained optimization by consensus-based primal-dual perturbation method*. IEEE Transactions on Automatic Control, vol. 59, no. 6, pages 1524–1538, 2014. (Cited on page 85).
- [Cheng 2013] Shi Cheng. *Population diversity in particle swarm optimization: Definition, observation, control, and application*. University of Liverpool, England, 2013. (Cited on page 79).
- [Chu 2003] Shu-Chuan Chu, John F Roddick and Jeng-Shyang Pan. *Parallel particle swarm optimization algorithm with communication strategies*. submitted to IEEE Transactions on Evolutionary Computation, 2003. (Cited on page 94).
- [Cimino 2009] Mauro Cimino and Prabhakar R Pagilla. *A design technique for multirate linear systems*. IEEE Transactions on Control Systems Technology, vol. 17, no. 6, pages 1342–1349, 2009. (Cited on page 201).
- [Clerc 2006] Maurice Clerc. *Stagnation analysis in particle swarm optimisation or what happens when nothing happens*. 2006. (Cited on page 77).
- [Crabtree 2014] Christopher J Crabtree, Donatella Zappalá and Peter J Tavner. *Survey of commercially available condition monitoring systems for wind turbines*. 2014. (Cited on pages vi and 5).
- [Dall’Anese 2013] Emiliano Dall’Anese, Hao Zhu and Georgios B Giannakis. *Distributed optimal power flow for smart microgrids*. IEEE Transactions on Smart Grid, vol. 4, no. 3, pages 1464–1475, 2013. (Cited on pages xi, 9 and 114).
- [de Alegría 2007] Iñigo Martinez de Alegría, Jon Andreu, José Luis Martín, Pedro Ibanez, José Luis Villate and Haritza Camblong. *Connection requirements for wind farms: A survey on technical requierements and regulation*. Renewable and Sustainable Energy Reviews, vol. 11, no. 8, pages 1858–1872, 2007. (Cited on pages vii, 6 and 28).

- [De Almeida 2007] Rogério G De Almeida and JA Peças Lopes. *Participation of doubly fed induction wind generators in system frequency regulation*. IEEE transactions on power systems, vol. 22, no. 3, pages 944–950, 2007. (Cited on page 43).
- [de Oliveira 1999] Maurício C de Oliveira, Jacques Bernussou and José C Geromel. *A new discrete-time robust stability condition*. Systems & control letters, vol. 37, no. 4, pages 261–265, 1999. (Cited on page 241).
- [De Oliveira 2002] Mauricio C De Oliveira, José C Geromel and Jacques Bernussou. *Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems*. International Journal of Control, vol. 75, no. 9, pages 666–679, 2002. (Cited on page 241).
- [Deb 2000] Kalyanmoy Deb. *An efficient constraint handling method for genetic algorithms*. Computer methods in applied mechanics and engineering, vol. 186, no. 2, pages 311–338, 2000. (Cited on pages 79, 80 and 114).
- [Deb 2012] Kalyanmoy Deb. Optimization for engineering design: Algorithms and examples. PHI Learning Pvt. Ltd., 2012. (Cited on page 79).
- [Del Valle 2008] Yamille Del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez and Ronald G Harley. *Particle swarm optimization: basic concepts, variants and applications in power systems*. IEEE Transactions on evolutionary computation, vol. 12, no. 2, pages 171–195, 2008. (Cited on page 84).
- [Delille 2013] Gauthier Delille, Gilles Malarange and Christophe Gaudin. *Analysis of the options to reduce the integration costs of renewable generation in the distribution networks. Part 2: A step towards advanced connection studies taking into account the alternatives to grid reinforcement*. 2013. (Cited on pages ix, 8 and 28).
- [Dung 2013] Thi Minh Dung, Tran Alain and Y Kibangou. *Distributed Design of Finite-time Average Consensus Protocols*. IFAC Proceedings Volumes, vol. 46, no. 27, pages 227–233, 2013. (Cited on pages xiii, 11, 70, 71, 72, 73 and 74).
- [Ebegbulem 2016] Judith Ebegbulem. *Distributed control of multi-agent systems using extremum seeking*. PhD thesis, 2016. (Cited on page 130).
- [Eberhart 1995] Russell Eberhart and James Kennedy. *A new optimizer using particle swarm theory*. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, pages 39–43. IEEE, 1995. (Cited on pages 74, 75 and 84).

- [ENTSO-E 2013] ENTSO-E. Network code for requirements for grid connection applicable to all generators. European Network of Transmission System Operators for Electricity, 2013. (Cited on page 28).
- [Feng 2015] Ju Feng and Wen Zhong Shen. *Solving the wind farm layout optimization problem using random search algorithm*. Renewable Energy, vol. 78, pages 182–192, 2015. (Cited on page 128).
- [Fleming 2015] Paul Fleming, Pieter MO Gebraad, Sang Lee, Jan-Willem Wingerden, Kathryn Johnson, Matt Churchfield, John Michalakes, Philippe Spalart and Patrick Moriarty. *Simulation comparison of wake mitigation control strategies for a two-turbine case*. Wind Energy, vol. 18, no. 12, pages 2135–2143, 2015. (Cited on page 128).
- [Francis 1976] Bruce A Francis and W Murray Wonham. *The internal model principle of control theory*. Automatica, vol. 12, no. 5, pages 457–465, 1976. (Cited on page 178).
- [Gazi 2014] Veysel Gazi and Raul Ordonez. *Particle swarm optimization based distributed agreement in multi-agent dynamic systems*. In Swarm Intelligence (SIS), 2014 IEEE Symposium on, pages 1–7. IEEE, 2014. (Cited on pages x, 9 and 85).
- [Ge 2013] Yanrong Ge, Yangzhou Chen, Yaxiao Zhang and Zhonghe He. *State consensus analysis and design for high-order discrete-time linear multiagent systems*. Mathematical Problems in Engineering, vol. 2013, 2013. (Cited on pages 172 and 173).
- [Gebraad 2013] Pieter MO Gebraad, Filip C van Dam and Jan-Willem van Wingerden. *A model-free distributed approach for wind plant control*. In American Control Conference (ACC), 2013, pages 628–633. IEEE, 2013. (Cited on page 130).
- [Gebraad 2016] PMO Gebraad, FW Teeuwisse, JW Wingerden, Paul A Fleming, SD Ruben, JR Marden and LY Pao. *Wind plant power optimization through yaw control using a parametric model for wake effects-a CFD simulation study*. Wind Energy, vol. 19, no. 1, pages 95–114, 2016. (Cited on pages xi, xiii, 9, 11, 128, 130, 132 and 136).
- [Gionfra 2016a] Nicolo Gionfra, Houria Siguerdidjane, Guillaume Sandou and Damien Faille. *Hierarchical Control of a Wind Farm for Wake Interaction Minimization*. IFAC-PapersOnLine, vol. 49, no. 27, pages 330–335, 2016. (Cited on page 132).
- [Gionfra 2016b] Nicolo Gionfra, Houria Siguerdidjane, Guillaume Sandou, Damien Faille and Philippe Loevenbruck. *Combined feedback linearization and MPC for wind turbine power tracking*. In Control Applications (CCA), 2016 IEEE Conference on, pages 52–57. IEEE, 2016. (Cited on page 30).

- [Gionfra 2017a] Nicolo Gionfra, Guillaume Sandou, Houria Siguerdidjane and Damien Faille. *A Distributed PID-like Consensus Control for Discrete-time Multi-agent Systems*. In ICINCO, 14th International Conference on Informatics in Control, Automation and Robotics, 2017. (Cited on page 172).
- [Gionfra 2017b] Nicolo Gionfra, Guillaume Sandou, Houria Siguerdidjane, Damien Faille and Philippe Loevenbruck. *A Distributed Consensus Control Under Disturbances for Wind Farm Power Maximization*. In Decision and Control (CDC), 2017 IEEE 56th Annual Conference on. IEEE, 2017. (Cited on page 192).
- [Gionfra 2017c] Nicolo Gionfra, Guillaume Sandou, Houria Siguerdidjane, Philippe Loevenbruck and Damien Faille. *A novel distributed particle swarm optimization algorithm for the optimal power flow problem*. In Control Technology and Applications (CCTA), 2017 IEEE Conference on. IEEE, 2017. (Cited on page 114).
- [Graham 1981] Alexander Graham. Kronecker products and matrix calculus with applications. Holsted Press, New York, 1981. (Cited on page 230).
- [Gros 2013] Sébastien Gros. *An economic NMPC formulation for wind turbine control*. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pages 1001–1006. IEEE, 2013. (Cited on page 29).
- [Hamon 2010] Camille Hamon, Katherine Elkington and Mehrdad Ghandhari. *Doubly-fed induction generator modeling and control in DigSilent PowerFactory*. In Power System Technology (POWERCON), 2010 International Conference on, pages 1–7. IEEE, 2010. (Cited on pages 29, 36, 231, 232 and 233).
- [Hansen 2006] Anca D Hansen, Poul Sørensen, Florin Iov and Frede Blaabjerg. *Centralised power control of wind farm with doubly fed induction generators*. Renewable Energy, vol. 31, no. 7, pages 935–951, 2006. (Cited on pages vi and 5).
- [Hauser 1992] John Hauser, Shankar Sastry and Petar Kokotovic. *Nonlinear control via approximate input-output linearization: The ball and beam example*. IEEE transactions on automatic control, vol. 37, no. 3, pages 392–398, 1992. (Cited on page 31).
- [Heer 2014] Flavio Heer, Peyman Mohajerin Esfahani, Maryam Kamgarpour and John Lygeros. *Model based power optimisation of wind farms*. In Control Conference (ECC), 2014 European, pages 1145–1150. IEEE, 2014. (Cited on pages xi, 10, 129 and 132).
- [Heier 2014] Siegfried Heier. *Wind energy conversion systems*. Grid Integration of Wind Energy: Onshore and Offshore Conversion Systems, pages 31–117, 2014. (Cited on page 33).

- [Hendrickx 2012] J Hendrickx, R Jungers, A Olshevsky and G Vankeerberghen. *Diameter, optimal-time consensus, and graph eigenvalues*, 2012. (Cited on page 71).
- [Herceg 2013] M. Herceg, M. Kvasnica, C.N. Jones and M. Morari. *Multi-Parametric Toolbox 3.0*. In Proc. of the European Control Conference, pages 502–510, Zürich, Switzerland, July 17–19 2013. <http://control.ee.ethz.ch/~mpt>. (Cited on page 26).
- [Hereford 2006] James M Hereford. *A distributed particle swarm optimization algorithm for swarm robotic applications*. In Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, pages 1678–1685. IEEE, 2006. (Cited on page 94).
- [Herp 2015] Jürgen Herp, Uffe V Poulsen and Martin Greiner. *Wind farm power optimization including flow variability*. Renewable Energy, vol. 81, pages 173–181, 2015. (Cited on pages xi, 9, 128 and 129).
- [Isidori 2013] Alberto Isidori. Nonlinear control systems. Springer Science & Business Media, 2013. (Cited on page 218).
- [Jansen 2003] Thomas Jansen and R Paul Wiegand. *Sequential versus parallel cooperative coevolutionary (1+ 1) EAs*. In Evolutionary Computation, 2003. CEC’03. The 2003 Congress on, volume 1, pages 30–37. IEEE, 2003. (Cited on page 96).
- [Jensen 1983] Niels Otto Jensen. A note on wind generator interaction. 1983. (Cited on page 127).
- [Jiménez 2010] Ángel Jiménez, Antonio Crespo and Emilio Migoya. *Application of a LES technique to characterize the wake deflection of a wind turbine in yaw*. Wind energy, vol. 13, no. 6, pages 559–572, 2010. (Cited on page 136).
- [Kadirkamanathan 2006] Visakan Kadirkamanathan, Kirusnapillai Selvarajah and Peter J Fleming. *Stability analysis of the particle dynamics in particle swarm optimizer*. IEEE Transactions on Evolutionary Computation, vol. 10, no. 3, pages 245–255, 2006. (Cited on page 77).
- [Katic 1986] I Katic, J Højstrup and Niels Otto Jensen. *A simple model for cluster efficiency*. In European wind energy association conference and exhibition, pages 407–410, 1986. (Cited on pages 127 and 138).
- [Kennedy 2006] J Kennedy and M Clerc. *Standard PSO*, 2006. (Cited on page 76).
- [Khalil 2002] H.K. Khalil. Nonlinear systems. Pearson Education. Prentice Hall, 2002. (Cited on pages 221 and 223).

- [Khezami 2010] Nadhira Khezami, Naceur Benhadj Braiek and Xavier Guillaud. *Wind turbine power tracking using an improved multimodel quadratic approach*. ISA transactions, vol. 49, no. 3, pages 326–334, 2010. (Cited on page 30).
- [Knudsen 2015] Torben Knudsen, Thomas Bak and Mikael Svenstrup. *Survey of wind farm control-power and fatigue optimization*. Wind Energy, vol. 18, no. 8, pages 1333–1351, 2015. (Cited on page 126).
- [Koh 2006] Byung-Il Koh, Alan D George, Raphael T Haftka and Benjamin J Fregly. *Parallel asynchronous particle swarm optimization*. International Journal for numerical methods in engineering, vol. 67, no. 4, pages 578–595, 2006. (Cited on page 94).
- [Kundur 2004] Prabha Kundur, John Paserba, Venkat Ajjarapu, Göran Andersson, Anjan Bose, Claudio Canizares, Nikos Hatziargyriou, David Hill, Alex Stankovic, Carson Tayloret al. *Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions*. IEEE transactions on Power Systems, vol. 19, no. 3, pages 1387–1401, 2004. (Cited on pages vii, 5 and 42).
- [Kvaternik 2011] Karla Kvaternik and Lacra Pavel. *Lyapunov analysis of a distributed optimization scheme*. In Network Games, Control and Optimization (NetGCooP), 2011 5th International Conference on, pages 1–5. IEEE, 2011. (Cited on page 82).
- [Larsen 2011] Gunner Chr Larsen, Helge Aagaard Madsen, Niels Troldborg, Torben J Larsen, Pierre-Elouan Réthoré, Peter Fuglsang, Søren Ott, Jakob Mann, Thomas Buhl, Morten Nielsen et al. *TOPFARM-next generation design tool for optimisation of wind farm topology and operation*. Technical report, Danmarks Tekniske Universitet, Risø Nationallaboratoriet for Bæredygtig Energi, 2011. (Cited on page 128).
- [Lavaei 2012] Javad Lavaei and Steven H Low. *Zero duality gap in optimal power flow problem*. IEEE Transactions on Power Systems, vol. 27, no. 1, pages 92–107, 2012. (Cited on page 113).
- [Li 2009] Shuhui Li, Rajab Chaloo and Marty J Nemmers. *Comparative study of DFIG power control using stator-voltage and stator-flux oriented frames*. In Power & Energy Society General Meeting, 2009. PES'09. IEEE, pages 1–8. IEEE, 2009. (Cited on page 30).
- [Li 2011] Zhongkui Li, Zhisheng Duan and Guanrong Chen. *On H_∞ and H_2 performance regions of multi-agent systems*. Automatica, vol. 47, no. 4, pages 797–803, 2011. (Cited on page 172).
- [Li 2012] Zhongkui Li, Zhisheng Duan, Lihua Xie and Xiangdong Liu. *Distributed robust control of linear multi-agent systems with parameter uncertainties*.

- International Journal of Control, vol. 85, no. 8, pages 1039–1050, 2012. (Cited on page 172).
- [Li 2013a] Xiaodong Li, Ke Tang, Mohammad N Omidvar, Zhenyu Yang, Kai Qin and Hefei China. *Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization*. *gene*, vol. 7, no. 33, page 8, 2013. (Cited on page 106).
- [Li 2013b] Zhongkui Li, Wei Ren, Xiangdong Liu and Mengyin Fu. *Distributed containment control of multi-agent systems with general linear dynamics in the presence of multiple leaders*. International Journal of Robust and Nonlinear Control, vol. 23, no. 5, pages 534–547, 2013. (Cited on page 172).
- [Liberzon 2012] Daniel Liberzon. *Switching in systems and control*. Springer Science & Business Media, 2012. (Cited on page 198).
- [Lin 2008] Peng Lin, Yingmin Jia, Junping Du and Fashan Yu. *Distributed leaderless coordination for networks of second-order agents with time-delay on switching topology*. In American Control Conference, 2008, pages 1564–1569. IEEE, 2008. (Cited on page 238).
- [Lio 2017] Wai Hou Lio, Bryn Ll Jones, Qian Lu and JA Rossiter. *Fundamental performance similarities between individual pitch control strategies for wind turbines*. International Journal of Control, vol. 90, no. 1, pages 37–52, 2017. (Cited on page 65).
- [Liserre 2011] Marco Liserre, Roberto Cardenas, Marta Molinas and Jose Rodriguez. *Overview of multi-MW wind turbines and wind parks*. IEEE Transactions on Industrial Electronics, vol. 58, no. 4, pages 1081–1095, 2011. (Cited on pages vi and 5).
- [Liu 2009] Yang Liu, Yingmin Jia, Junping Du and Shiyang Yuan. *Dynamic output feedback control for consensus of multi-agent systems: an H_∞ approach*. In 2009 American Control Conference, pages 4470–4475. IEEE, 2009. (Cited on pages 172 and 175).
- [Liu 2010] Hui Liu, Zixing Cai and Yong Wang. *Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization*. Applied Soft Computing, vol. 10, no. 2, pages 629–640, 2010. (Cited on page 79).
- [Liu 2015] Jie Liu, Mouhacine Benosman and AU Raghunathan. *Consensus-based distributed optimal power flow algorithm*. In Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society, pages 1–5. IEEE, 2015. (Cited on pages 113 and 114).

- [Loukarakis 2009] Emmanouil Loukarakis, Ioannis Margaritis and Panayiotis Moutis. *Frequency control support and participation methods provided by wind generation*. In Electrical Power & Energy Conference (EPEC), 2009 IEEE, pages 1–6. IEEE, 2009. (Cited on pages 30 and 43).
- [Ma 1995] Xin Ma, Niels Kjølstad Poulsen and Henrik Bindner. *Estimation of wind speed in connection to a wind turbine*. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 1995. (Cited on page 54).
- [Madjidian 2011] Daria Madjidian, Karl Mårtensson and Anders Rantzer. *A distributed power coordination scheme for fatigue load reduction in wind farms*. In American Control Conference (ACC), 2011, pages 5219–5224. IEEE, 2011. (Cited on page 131).
- [Magnússon 2015] Sindri Magnússon, Pradeep Chathuranga Weeraddana and Carlo Fischione. *A distributed approach for the optimal power-flow problem based on ADMM and sequential convex approximations*. IEEE Transactions on Control of Network Systems, vol. 2, no. 3, pages 238–253, 2015. (Cited on pages xi, 9 and 114).
- [Mahdavi 2015] Sedigheh Mahdavi, Mohammad Ebrahim Shiri and Shahryar Rahnamayan. *Metaheuristics in large-scale global continues optimization: A survey*. Information Sciences, vol. 295, pages 407–428, 2015. (Cited on page 95).
- [Marden 2013] Jason R Marden, Shalom D Ruben and Lucy Y Pao. *A model-free approach to wind farm control using game theoretic methods*. IEEE Transactions on Control Systems Technology, vol. 21, no. 4, pages 1207–1214, 2013. (Cited on pages xi, 9, 10, 130 and 131).
- [Mauricio 2009] Juan Manuel Mauricio, Alejandro Marano, Antonio Gómez-Expósito and José Luis Martínez Ramos. *Frequency regulation contribution through variable-speed wind energy conversion systems*. IEEE Transactions on Power Systems, vol. 24, no. 1, pages 173–180, 2009. (Cited on pages vii, 6 and 42).
- [Minaud 2013] Antoine Minaud, Christophe Gaudin and Laurent Karsenti. *Analysis of the options to reduce the integration costs of renewable generation in the distribution networks. part 1: Impact of pv development in france and global analysis of considered alternatives to reinforcement*. 2013. (Cited on page 28).
- [Monaco 1988] S Monaco and D Normand-Cyrot. *Zero dynamics of sampled nonlinear systems*. Systems & control letters, vol. 11, no. 3, pages 229–234, 1988. (Cited on page 17).

- [Monaco 2001] Salvatore Monaco and Dorothée Normand-Cyrot. *Issues on non-linear digital control*. European Journal of Control, vol. 7, no. 2-3, pages 160–177, 2001. (Cited on page 201).
- [Morren 2006] Johan Morren, Sjoerd WH De Haan, Wil L Kling and JA Ferreira. *Wind turbines emulating inertia and supporting primary frequency control*. IEEE Transactions on power systems, vol. 21, no. 1, pages 433–434, 2006. (Cited on pages vii, 6 and 42).
- [Mortensen 1998] Niels Gylling Mortensen, Lars Landberg, Ib Troen and Erik Lundtang Petersen. *Wind Atlas Analysis and Application program (WAsP): Vol. 1: Getting started*. Technical report, Risø National Laboratory, 1998. (Cited on page 128).
- [Navarro 2015] Inaki Navarro, Ezequiel Di Mario and Alcherio Martinoli. *Distributed Particle Swarm Optimization-particle allocation and neighborhood topologies for the learning of cooperative robotic behaviors*. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 2958–2965. IEEE, 2015. (Cited on pages x, 9 and 85).
- [Nedic 2009] Angelia Nedic and Asuman Ozdaglar. *Distributed subgradient methods for multi-agent optimization*. IEEE Transactions on Automatic Control, vol. 54, no. 1, pages 48–61, 2009. (Cited on page 82).
- [Nevistic 1995] V Nevistic and M Morari. *Constrained control of feedback-linearizable systems*. In Proc. 3rd European Control Conference ECC'95, pages 1726–1731, 1995. (Cited on pages 17, 18, 19 and 30).
- [Oh 2014] Kwang-Kyo Oh, Kevin L Moore and Hyo-Sung Ahn. *Disturbance Attenuation in a Consensus Network of Identical Linear Systems: An H_∞ Approach*. IEEE Transactions on Automatic Control, vol. 59, no. 8, pages 2164–2169, 2014. (Cited on page 172).
- [Olfati-Saber 2007] Reza Olfati-Saber, J Alex Fax and Richard M Murray. *Consensus and cooperation in networked multi-agent systems*. Proceedings of the IEEE, vol. 95, no. 1, pages 215–233, 2007. (Cited on pages 70 and 147).
- [Østergaard 2007] Kasper Zinck Østergaard, Per Brath and Jakob Stoustrup. *Estimation of effective wind speed*. In Journal of Physics: Conference Series, volume 75, page 012082. IOP Publishing, 2007. (Cited on page 54).
- [Ou 2014] Lin-Lin Ou, Jun-Jie Chen, Dong-Mei Zhang, Lin Zhang and Wei-Dong Zhang. *Distributed H_∞ PID feedback for improving consensus performance of arbitrary-delayed multi-agent system*. International Journal of Automation and Computing, vol. 11, no. 2, pages 189–196, 2014. (Cited on page 172).

- [Pan 2014] Feng Pan, Qianqian Zhang, Jun Liu, Weixing Li and Qi Gao. *Consensus analysis for a class of stochastic PSO algorithm*. Applied Soft Computing, vol. 23, pages 567–578, 2014. (Cited on page 77).
- [Park 2015a] Jinkyoo Park and Kincho H Law. *Cooperative wind turbine control for maximizing wind farm power using sequential convex programming*. Energy Conversion and Management, vol. 101, pages 295–316, 2015. (Cited on pages viii, xi, xiii, xxii, 6, 9, 11, 128, 129, 132, 133, 134, 135, 136, 137, 138 and 148).
- [Park 2015b] Jinkyoo Park and Kincho H Law. *Layout optimization for maximizing wind farm power production using sequential convex programming*. Applied Energy, vol. 151, pages 320–334, 2015. (Cited on page 128).
- [Park 2016] Jinkyoo Park and Kincho H Law. *A data-driven, cooperative wind farm control to maximize the total power production*. Applied Energy, vol. 165, pages 151–165, 2016. (Cited on pages xi, 9 and 130).
- [Peña 2014] Alfredo Peña, Pierre-Elouan Réthoré and Ole Rathmann. *Modeling large offshore wind farms under different atmospheric stability regimes with the Park wake model*. Renewable Energy, vol. 70, pages 164–171, 2014. (Cited on page 128).
- [Penarrocha 2014] I Penarrocha, D Dolz, N Aparicio and R Sanchis. *Synthesis of nonlinear controller for wind turbines stability when providing grid support*. International Journal of Robust and Nonlinear Control, vol. 24, no. 16, pages 2261–2284, 2014. (Cited on page 30).
- [Peng 2014] Qiuyu Peng and Steven H Low. *Distributed Optimal Power Flow Algorithm for Balanced Radial Distribution Networks*. arXiv preprint arXiv:1404.0700, 2014. (Cited on pages xi, 9 and 114).
- [Perdana 2008] Abram Perdana. Dynamic models of wind turbines. Chalmers University of Technology, 2008. (Cited on page 29).
- [Petru 2002] Tomas Petru and Torbjörn Thiringer. *Modeling of wind turbines for power system studies*. IEEE transactions on Power Systems, vol. 17, no. 4, pages 1132–1139, 2002. (Cited on page 32).
- [Poller 2003] Markus A Poller. *Doubly-fed induction machine models for stability assessment of wind farms*. In Power Tech Conference Proceedings, 2003 IEEE Bologna, volume 3, pages 6–pp. IEEE, 2003. (Cited on pages 29, 36 and 232).
- [Popovici 2006] Elena Popovici and Kenneth De Jong. *Sequential versus parallel co-operative coevolutionary algorithms for optimization*. In Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, pages 1610–1617. IEEE, 2006. (Cited on pages 95, 96 and 99).

- [Potter 1994] Mitchell A Potter and Kenneth A De Jong. *A cooperative coevolutionary approach to function optimization*. In International Conference on Parallel Problem Solving from Nature, pages 249–257. Springer, 1994. (Cited on page 95).
- [Pulido 2004] Gregorio Toscano Pulido and Carlos A Coello Coello. *A constraint-handling mechanism for particle swarm optimization*. In Evolutionary Computation, 2004. CEC2004. Congress on, volume 2, pages 1396–1403. IEEE, 2004. (Cited on page 79).
- [Ramtharan 2007] G Ramtharan, Nick Jenkins and JB Ekanayake. *Frequency support from doubly fed induction generator wind turbines*. IET Renewable Power Generation, vol. 1, no. 1, pages 3–9, 2007. (Cited on pages 30 and 43).
- [Rawlings 2009] James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009. (Cited on pages 22, 23 and 224).
- [Ren 2005] Wei Ren and Randal W Beard. *Consensus seeking in multiagent systems under dynamically changing interaction topologies*. IEEE Transactions on automatic control, vol. 50, no. 5, pages 655–661, 2005. (Cited on page 237).
- [Ren 2007] Wei Ren. *Multi-vehicle consensus with a time-varying reference state*. Systems & Control Letters, vol. 56, no. 7, pages 474–483, 2007. (Cited on page 172).
- [Ren 2008] Wei Ren and Randal W Beard. *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008. (Cited on pages 169, 170 and 171).
- [Rivera 2001] Wilson Rivera. *Scalable parallel genetic algorithms*. Artificial intelligence review, vol. 16, no. 2, pages 153–168, 2001. (Cited on pages 84 and 94).
- [Sandou 2008] Guillaume Sandou, Gilles Duc and Dominique Beauvois. *Optimisation par essaim particulaire du réglage d’un correcteur H-infini*. In 5ème Conférence Internationale Francophone d’Automatique, pages CD–ROM, 2008. (Cited on page 75).
- [Schiffer 2016] Johannes Schiffer, Thomas Seel, Jörg Raisch and Tevfik Sezi. *Voltage stability and reactive power sharing in inverter-based microgrids with consensus-based distributed voltage control*. IEEE Transactions on Control Systems Technology, vol. 24, no. 1, pages 96–109, 2016. (Cited on page 192).
- [Schutte 2004] Jaco F Schutte, Jeffrey A Reinbolt, Benjamin J Fregly, Raphael T Haftka and Alan D George. *Parallel global optimization with the particle swarm algorithm*. International journal for numerical methods in engineering, vol. 61, no. 13, pages 2296–2315, 2004. (Cited on page 94).

- [Sedlaczek 2005] Kai Sedlaczek and Peter Eberhard. *Constrained particle swarm optimization of mechanical systems*. Proceedings of the 6th WCSMO, Rio de Janeiro, Brazil, 2005. (Cited on page 80).
- [Soleimanzadeh 2010] Maryam Soleimanzadeh and Rafael Wisniewski. *Wind speed dynamical model in a wind farm*. In Control and Automation (ICCA), 2010 8th IEEE International Conference on, pages 2246–2250. IEEE, 2010. (Cited on pages 128 and 129).
- [Soleimanzadeh 2012] Maryam Soleimanzadeh, Rafael Wisniewski and Stoyan Kanev. *An optimization framework for load and power distribution in wind farms*. Journal of Wind Engineering and Industrial Aerodynamics, vol. 107, pages 256–262, 2012. (Cited on page 129).
- [Spudic 2010] Vedrana Spudic, M Jelavic, M Baotic and Nedjeljko Peric. *Hierarchical wind farm control for power/load optimization*. The science of making torque from wind (Torque2010), 2010. (Cited on page 129).
- [Spudić 2015] Vedrana Spudić, Christian Conte, Mato Baotić and Manfred Morari. *Cooperative distributed model predictive control for wind farms*. Optimal Control Applications and Methods, vol. 36, no. 3, pages 333–352, 2015. (Cited on pages xi, 10, 190 and 191).
- [Su 2012] Youfeng Su and Jie Huang. *Two consensus problems for discrete-time multi-agent systems with switching network topology*. Automatica, vol. 48, no. 9, pages 1988–1997, 2012. (Cited on page 172).
- [Sun 2013] Andy X Sun, Dzung T Phan and Soumyadip Ghosh. *Fully decentralized AC optimal power flow algorithms*. In Power and Energy Society General Meeting (PES), 2013 IEEE, pages 1–5. IEEE, 2013. (Cited on pages xi, 9 and 114).
- [Sundhar Ram 2012] S Sundhar Ram, A Nedić and Venugopal V Veeravalli. *A new class of distributed optimization algorithms: Application to regression of distributed data*. Optimization Methods and Software, vol. 27, no. 1, pages 71–88, 2012. (Cited on pages 82 and 84).
- [Syai'in 2012] Mat Syai'in, Kuo Lung Lian, Nien-Che Yang and Tsai-Hsiang Chen. *A distribution power flow using particle swarm optimization*. In Power and Energy Society General Meeting, 2012 IEEE, pages 1–7. IEEE, 2012. (Cited on page 114).
- [Tarnowski 2009] Germán Claudio Tarnowski, Philip Carne Kjar, Poul E Sorensen and Jacob Ostergaard. *Variable speed wind turbines capability for temporary over-production*. In Power & Energy Society General Meeting, 2009. PES'09. IEEE, pages 1–7. IEEE, 2009. (Cited on page 30).

- [Thomsen 2006] Sven Creutz Thomsen. Nonlinear control of a wind turbine. Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2006. (Cited on pages 29, 33, 46 and 47).
- [Tian 2014] Jie Tian, Chi Su, M Soltani and Zhe Chen. *Active power dispatch method for a wind farm central controller considering wake effect*. In Industrial Electronics Society, IECON 2014-40th Annual Conference of the IEEE, pages 5450–5456. IEEE, 2014. (Cited on page 129).
- [Tomlin 1998] CJ Tomlin and S Shankar Sastry. *Switching through singularities*. Systems & control letters, vol. 35, no. 3, pages 145–154, 1998. (Cited on page 20).
- [Trelea 2003] Ioan Cristian Trelea. *The particle swarm optimization algorithm: convergence analysis and parameter selection*. Information processing letters, vol. 85, no. 6, pages 317–325, 2003. (Cited on page 77).
- [Ulbig 2015] Andreas Ulbig, Theodor S Borsche and Göran Andersson. *Analyzing Rotational Inertia, Grid Topology and their Role for Power System Stability*. IFAC-PapersOnLine, vol. 48, no. 30, pages 541–547, 2015. (Cited on pages vii and 6).
- [van den Bergh 2002] Frans van den Bergh and Andries P Engelbrecht. *A new locally convergent particle swarm optimiser*. In Systems, Man and Cybernetics, 2002 IEEE International Conference on, volume 3, pages 6–pp. IEEE, 2002. (Cited on page 77).
- [Van den Bergh 2004] Frans Van den Bergh and Andries Petrus Engelbrecht. *A cooperative approach to particle swarm optimization*. IEEE transactions on evolutionary computation, vol. 8, no. 3, pages 225–239, 2004. (Cited on pages 95, 97, 100 and 104).
- [Venne 2009] Philippe Venne and X Guillaud. *Impact of wind turbine controller strategy on deloaded operation*. In Integration of Wide-Scale Renewable Resources Into the Power Delivery System, 2009 CIGRE/IEEE PES Joint Symposium, pages 1–1. IEEE, 2009. (Cited on page 40).
- [Venter 2006] Gerhard Venter and Jaroslaw Sobieszczanski-Sobieski. *Parallel Particle Swarm Optimization Algorithm Accelerated by Asynchronous Evaluations*. JACIC, vol. 3, no. 3, pages 123–137, 2006. (Cited on page 94).
- [Venter 2010] Gerhard Venter and RT Haftka. *Constrained particle swarm optimization using a bi-objective formulation*. Structural and Multidisciplinary Optimization, vol. 40, no. 1, pages 65–76, 2010. (Cited on page 80).
- [Wakasa 2015a] Yuji Wakasa and Sosuke Nakaya. *Distributed particle swarm optimization using an average consensus algorithm*. In Decision and Control

- (CDC), 2015 IEEE 54th Annual Conference on, pages 2661–2666. IEEE, 2015. (Cited on pages [x](#), [9](#), [80](#), [84](#), [85](#), [92](#), [95](#) and [96](#)).
- [Wakasa 2015b] Yuji Wakasa and Sho Yamasaki. *Distributed Particle Swarm Optimization Based on Primal-Dual Decomposition Architectures*. In Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications, volume 2015, pages 97–101. The ISCIE Symposium on Stochastic Systems Theory and Its Applications, 2015. (Cited on pages [x](#), [9](#), [80](#), [85](#), [95](#) and [96](#)).
- [Wang 2010] Jing Wang and Nicola Elia. *Control approach to distributed optimization*. In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, pages 557–561. IEEE, 2010. (Cited on page [83](#)).
- [Wang 2011a] Jing Wang and Nicola Elia. *A control perspective for centralized and distributed convex optimization*. In Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 3800–3805. IEEE, 2011. (Cited on pages [82](#) and [83](#)).
- [Wang 2011b] Liyong Wang and Lixin Gao. *H_∞ Consensus Control for Discrete-Time Multi-Agent Systems with Switching Topology*. Procedia Engineering, vol. 15, pages 601 – 607, 2011. (Cited on pages [172](#), [175](#) and [177](#)).
- [Wang 2014] Shu Wang and Peter Seiler. *Gain scheduled active power control for wind turbines*. In AIAA Atmospheric Flight Mechanics Conference, 2014. (Cited on page [30](#)).
- [Wang 2015] Xiaoping Wang and Jinliang Shao. *Consensus for Discrete-Time Multiagent Systems*. Discrete Dynamics in Nature and Society, 2015. (Cited on page [174](#)).
- [Wannakarn 2010] P Wannakarn, S Khamsawang, S Pothiya and S Jiriwibhakorn. *Optimal power flow problem solved by using distributed Sobol particle swarm optimization*. In Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on, pages 445–449. IEEE, 2010. (Cited on page [114](#)).
- [Weise 2012] Thomas Weise, Raymond Chiong and Ke Tang. *Evolutionary optimization: Pitfalls and booby traps*. Journal of Computer Science and Technology, vol. 27, no. 5, pages 907–936, 2012. (Cited on page [97](#)).
- [Weiss 2017] Avishai Weiss, Claus Danielson, Karl Berntorp, Ilya Kolmanovsky and Stefano Di Cairano. *Motion planning with invariant set trees*. In Control Technology and Applications (CCTA), 2017 IEEE Conference on, pages 1625–1630. IEEE, 2017. (Cited on page [26](#)).

- [Wiegand 2003] R Paul Wiegand. *An analysis of cooperative coevolutionary algorithms*. PhD thesis, George Mason University Virginia, 2003. (Cited on pages 95 and 97).
- [Wu 2011] Zhizheng Wu, Azhar Iqbal and Foued Ben Amara. *LMI-based multivariable PID controller design and its application to the control of the surface shape of magnetic fluid deformable mirrors*. IEEE Transactions on Control Systems Technology, vol. 19, no. 4, pages 717–729, 2011. (Cited on pages 172, 174 and 241).
- [Xi 2010] Jianxiang Xi, Ning Cai and Yisheng Zhong. *Consensus problems for high-order linear time-invariant swarm systems*. Physica A: Statistical Mechanics and its Applications, vol. 389, no. 24, pages 5619–5627, 2010. (Cited on page 172).
- [Xi 2012] Jianxiang Xi, Zongying Shi and Yisheng Zhong. *Output consensus analysis and design for high-order linear swarm systems: partial stability method*. Automatica, vol. 48, no. 9, pages 2335–2343, 2012. (Cited on page 172).
- [Yang-Zhou 2014] CHEN Yang-Zhou, GE Yan-Rong and Ya-Xiao ZHANG. *Partial stability approach to consensus problem of linear multi-agent systems*. Acta Automatica Sinica, vol. 40, no. 11, pages 2573–2584, 2014. (Cited on page 172).
- [Yang 2008] Zhenyu Yang, Ke Tang and Xin Yao. *Large scale evolutionary optimization using cooperative coevolution*. Information Sciences, vol. 178, no. 15, pages 2985–2999, 2008. (Cited on page 95).
- [Yingcheng 2011] Xue Yingcheng and Tai Nengling. *Review of contribution to frequency control through variable speed wind turbine*. Renewable Energy, vol. 36, no. 6, pages 1671–1677, 2011. (Cited on pages x, 8, 30, 43 and 44).
- [You 2011] Keyou You and Lihua Xie. *Network topology and communication data rate for consensusability of discrete-time multi-agent systems*. IEEE Transactions on Automatic Control, vol. 56, no. 10, pages 2262–2275, 2011. (Cited on page 172).
- [Žertek 2012a] A Žertek, G Verbič and M Pantoš. *Optimised control approach for frequency-control contribution of variable speed wind turbines*. IET Renewable power generation, vol. 6, no. 1, pages 17–23, 2012. (Cited on pages 30 and 43).
- [Zertek 2012b] Andraž Zertek, Gregor Verbic and Miloš Pantos. *A novel strategy for variable-speed wind turbines’ participation in primary frequency control*. IEEE Transactions on sustainable energy, vol. 3, no. 4, pages 791–799, 2012. (Cited on pages x, 8, 42, 43 and 45).

- [Zhang 2006] Fu Zhang and B Fernandez-Rodriguez. *Feedback linearization control of systems with singularities: a ball-beam revisit*. In Proc. of the Int. Conf. on Complex Systems, 2006. (Cited on page 20).
- [Zhang 2013] Wei Zhang, Yinliang Xu, Wenxin Liu, Frank Ferrese and Liming Liu. *Fully distributed coordination of multiple DFIGs in a microgrid for load sharing*. IEEE Transactions on Smart Grid, vol. 4, no. 2, pages 806–815, 2013. (Cited on pages 131 and 190).
- [Zhang 2014] Ruiliang Zhang and James Kwok. *Asynchronous distributed ADMM for consensus optimization*. In International Conference on Machine Learning, pages 1701–1709, 2014. (Cited on page 83).

Titre : Stratégies de commande distribuée pour l'optimisation de la production des fermes éoliennes

Mots clés : ferme éolienne, effet de sillage, pilotage de puissance active, contrôle-commande distribué, optimisation métaheuristique distribuée, systèmes multi-agents

Résumé : Les travaux de thèse s'intéressent au réglage de la puissance active injectée dans le réseau, ce qui représente aujourd'hui l'une des problématiques principales du pilotage des parcs éoliens participant à la gestion du réseau. Dans le même temps, l'un des buts reste de maximiser la puissance extraite du vent en considérant les effets de couplage aérodynamique entre les éoliennes.

La structure du contrôle-commande choisie est de type hiérarchisée et distribuée. Dans la première partie de la thèse, les travaux portent sur la commande de la turbine d'une éolienne autour des points de fonctionnement classiques mais également autour des points à puissance extraite réduite. En fait, cela relève d'une condition de fonctionnement nécessaire pour l'atteinte des objectifs imposés au pilotage d'un parc éolien.

Dans la deuxième partie, le problème du contrôle à

l'échelle d'un parc est posé sous la forme d'une optimisation distribuée parmi les turbines. Deux nouveaux algorithmes d'optimisation métaheuristique sont proposés et leur performance testée sur différents exemples de parcs éoliens. Les deux algorithmes s'appuient sur la méthode d'optimisation par essaim particulaire, qui est ici modifiée et adaptée pour les cas d'application aux systèmes multi agents. L'architecture de contrôle-commande globale est enfin évaluée en considérant les dynamiques des turbines contrôlées. Les simulations effectuées montrent des gains potentiels significatifs en puissance.

Finalement, dans la troisième partie de la thèse, l'introduction d'une nouvelle étape de coopération au niveau des contrôleurs locaux des turbines, par l'utilisation de la technique de contrôle par consensus, permet d'améliorer les performances du système global.

Title : Distributed control strategies for wind farm power production optimization

Keywords : wind farm, wake effect, active power control, distributed control, distributed metaheuristic optimization, multi-agent systems

Abstract : In this PhD work we focus on the wind farm (WF) active power control since some of the new set grid requirements of interest can be expressed as specifications on its injection in the electric grid. Besides, one of our main objectives is related to the wind farm power maximization problem under the presence on non-negligible wake effect. The chosen WF control architecture has a two-layer hierarchical distributed structure. First of all, the wind turbine (WT) control is addressed. Here, a nonlinear controller lets a WT work in classic zones of functioning as well as track general deloaded power references. This last feature is a necessary condition to accomplish the WF control specifications. Secondly, the high level WF control problem is formulated as an optimization problem distributed among the WTs.

Two novel distributed optimization algorithms are proposed, and their performance tested on different WF examples. Both are based on the well-known particle swarm optimization algorithm, which we modify and extend to be applicable in the multi-agent system framework. Finally, the overall WF control is evaluated by taking into account the WTs controlled dynamics. Simulations show potential significant power gains. Eventually, the introduction of a new control level in the hierarchical structure between the WF optimization and the WTs controllers is proposed. The idea is to let further cooperation among the WT local controllers, via a consensus-based technique, to enhance the overall system performance.

