Stability of Barrier Model Predictive Control

Maxime Pouilly-Cathelain, Philippe Feyel, Gilles Duc, Guillaume Sandou

Stability of Barrier Model Predictive Control

Maxime Pouilly-Cathelain[1,2], Philippe Feyel[1], Gilles Duc and Guillaume Sandou[2]
[1] *Safran Electronics & Defense, Massy, France*
[2] *Université Paris-Saclay, CNRS, L2S, CentraleSupélec, Gif-sur-Yvette, France*
{*maxime.pouilly-cathelain, philippe.feyel*}@*safrangroup.com* {*gilles.duc, guillaume.sandou*}@*centralesupelec.fr*

Keywords: Model Predictive Control, Nonlinear Constraints, Barrier functions, Stability analysis, Invariant set

Abstract: In the last decades, industrial problems have tried to take into account constraints explicitly in the design of the control law. Model Predictive Control is one way to do so and has been extensively studied. However, most papers related to constrained Model Predictive Control often omit to consider nondifferentiable constraints and stability is not ensured when constraints are not satisfied. The aim of this paper is to propose a formulation of the cost function of a Model Predictive Control to ensure stability in face with input and state nondifferentiable constraints. For this purpose, a set where all constraints are satisfied is defined by means of the invariant set theory. Once this set is defined, the system is enforced to reach it and stay in, while guaranteeing stability thanks to the choice of a well suited Lyapunov function based on the cost function.

1 INTRODUCTION

In industrial problems, system design is usually based on high-level system specifications. In order to reach these specifications with high performances, requirements have to be taken into account during the design of the control law. Model predictive control (MPC) is one way to control systems with state and input constraints. In this paper, we aim to design a stabilizing controller for a discrete linear system subject to nondifferentiable constraints.

As presented in (Rawlings et al., 2017), MPC method elaborates the control input as the result of an optimization problem. At each time step, a vector of future inputs is determined to get the optimal predicted behavior of the system on a finite prediction horizon according to a cost function. Following the principle of the receding horizon, only the first input of the obtained input sequence is applied to the system. Stability, feasibility and constraint considerations are important points that have been studied in MPC theory. The following paragraphs present some works related to these topics in order to capture the underlying context of our study.

To begin with, MPC stability has been extensively studied in recent decades. (Chen and Shaw, 1982) is one of the first papers using a terminal constraint and a Lyapunov function based on the cost function of the MPC in order to prove stability. At this point no constraint was taken into account. (Keerthi and Gilbert, 1988) extended this work by using an equality terminal constraint and a Lyapunov function to prove the stability for a nonlinear discrete time varying system subject to constraints. An equality constraint is in general really challenging for an optimization algorithm. Proof given in previous cited articles suppose that the system is able to reach the final state before the end of the prediction horizon in order to avoid the problem of recursive feasibility. More recently, in order to relax the terminal constraint while keeping stability, (Rawlings and Muske, 1993) have used the cost of the infinite horizon problem to define a terminal cost. The proof of stability is based on the use of a final controller related to the infinite horizon control problem. (Chisci et al., 1996) and (Scokaert et al., 1999) impose a terminal set where a state feedback controller stabilizes the system. One drawback of this method is the need to reach this set before the end of the prediction horizon.

Next, one main issue of constrained MPC is the feasibility of the problem because infeasibility can lead to instability. In fact, in presence of constraints, the system can be unable to reach the target point regardless of the feasibility of the optimization problem (this defines MPC infeasibility). Recursive feasibility is an important point, meaning that if the system can reach the target point at a given time step, the problem is also feasible at the next time step. To obtain this, some improvements of the MPC problem have been proposed. To avoid infeasibility to reach an imposed terminal set, (Michalska and Mayne, 1993) propose a solution with a prediction horizon of variable length. In this case, the problem is that the prediction horizon could become really large and the optimization algorithm could be unable to find a solution. In order to relax the problem, it is possible to use slack variables, as presented in (Camacho and Bordons, 2007). The main drawback of this method is that it increases the number of optimization parameters. Finally, (Zeilinger et al., 2010) propose to relax the terminal set constraint that can grow to make the problem feasible.

Barrier model predictive control (BMPC) has emerged in order to take into account constrains directly in the cost function. Barrier functions method is similar to the use of slack variables in that it permits to relax constraints. The first proposition has been done by (Wills and Heath, 2004) for differentiable constraints. Regardless of BMPC, economic MPC has been developed to enforce reduction of some quantities. According to (Amrit et al., 2011), economic MPC minimizes a cost function that includes convex constraints. (Müller et al., 2014) presents a formulation that permits to take into account the mean value of a state in the cost function.

Stability of BMPC has already been studied several times. For instance in (Wills and Heath, 2004) the stability has been proven by using a quadratic upper bound of the barrier function and an ellipsoidal terminal set. The same method has been used in (Feller and Ebenbauer, 2015) with a polytopic terminal set. This work is extended to suboptimal solution in (Feller and Ebenbauer, 2017) where polytopic constraints on states and inputs are included in the cost function using relaxed logarithmic barrier. Furthermore, (Petsagkourakis et al., 2019) propose conditions to ensure robust stability according to system uncertainties, yet this paper only deals with differentiable constraints. To the best of our knowledge, there is no literature dealing with BMPC and nondifferentiable constraints. Moreover, the case where the system is far from the set where all constraints are satisfied is often omitted.

In this paper, we aim to propose a BMPC for linear discrete systems that can deal with any evaluable constraint, being differentiable or not, by means of barrier functions. For this purpose, a gradient free optimization algorithm is needed to deal with any formulation of barrier function. For instance, differential evolution algorithms (Price et al., 2006) are interesting candidates to solve this problem. This choice has been done in (Merabti and Belarbi, 2014) with a multi-objective model predictive control using metaheuristic. Nevertheless this solution is only suitable for a relatively small number of constraints, as the difficulty of choosing the best solution in the Pareto front increases with the number of objectives.

In addition to the proposed methodology, a proof of stability using Lyapunov and invariant set theory is also established in the case of linear discrete systems under some realistic assumptions. The proposed method is based on the following steps:

1. Constraints formulation. Constraints directly result from the initial set of specifications and no reformulation should be required at this stage.
2. Constraints classification. Constraints are classified into two different categories and some of them are used in the cost function as barrier functions. This classification is important for stability purpose as it will be shown in the sequel.
3. Unconstrained set definition. Specifications permit to define a set where all constraints are satisfied and the corresponding backward reachable set. These sets are defined in such a way that the stability can be proven.
4. Cost function definition. The cost function is defined by using barrier functions defined in step (2) and according to sets defined in step (3).

This paper is organized as follows. First of all, we will introduce some notations in section 2. Section 3 introduces conditions about the barrier functions to be chosen. Then, section 4 is related to the determination of the maximal unconstrained set. The cost function of the proposed BMPC is developed in section 5. Thereafter, section 6 provides a proof of stability based on the Lyapunov theory. An application of the proposed BMPC is given in section 7. Finally, section 8 concludes and proposes some future works.

2 NOTATIONS

Linear discrete-time systems of interests in this paper are defined by (1).

$$\begin{cases} x_{K+1} &= Ax_K + Bu_K \\ y_K &= Cx_K + Du_K \end{cases} . \tag{1}$$

where $K \in \mathbb{N}$ is the time index, $x_K \in \mathbb{X} \subseteq \mathbb{R}^n$ ($n \in \mathbb{N}^*$), $u_K \in \mathbb{U} \subseteq \mathbb{R}^p$ ($p \in \mathbb{N}^*$), $y \in \mathbb{R}^m$ ($m \in \mathbb{N}^*$), $A \in \mathbb{R}^{n\times n}$, $B \in \mathscr{R}^{n\times p}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times p}$. $\mathbb{X}$ and $\mathbb{U}$ are two compact sets. Without any loss of generality, the system will be stabilized at the origin. Assumption 1 and 2 are considered for the system.

Assumption 1. (A,B) *is controllable.*

Assumption 2. *A is non singular. Among others, this is the case for systems obtained by discretization of continuous systems without time delay.*

Our prediction model is defined by (2) where predicted states and outputs will be respectively denoted by $\bar{x}$ and $\bar{y}$.

$$\forall k \in [\![0; N-1]\!], \left\{ \begin{array}{lcl} \bar{x}_{k+1} & = & A\bar{x}_k + Bu_k \\ \bar{y}_k & = & C\bar{x}_k + Du_k \end{array} \right. . \quad (2)$$

$\bar{x}_0 \in \mathbb{R}^n$ is the initial state of the system and $\bar{x}_N \in \mathbb{R}^n$ the final predicted state where $N \in \mathbb{N}^*$ is the prediction horizon. It is assumed that the state at time K, x_K, can be measured thus $\bar{x}_0 = x_K$. The vector of predicted states is $\bar{\mathbf{x}} = [\bar{x}_0, \ldots, \bar{x}_N]$.

J_N denotes the cost function. Mathematical formatting of the cost function will be more precisely presented in section 5. We also define the optimal cost J_N^* and the problem $\mathbb{P}_N(x_K)$ which is solved by the MPC optimizer at each time step as (3).

$$\mathbb{P}_N(x_K) : J_N^*(x_K) = \min_{\mathbf{u}} \left\{ J_N(\bar{\mathbf{x}}, \mathbf{u}) \mid \mathbf{u} \in \mathbb{U}^N \right\}. \quad (3)$$

$\mathbf{u} \in \mathbb{U}^N$ denotes the vector of control inputs. According to the receding horizon principle, the first component of the vector $\mathbf{u}$ is applied to the system and (3) is solved again at the following time step.

$\|x\|_Q$ denotes the norm of $x \in \mathbb{R}^n$ defined by $\|x\|_Q = x^{\mathrm{T}} Q x$ where $Q \in \mathbb{R}^{n \times n}$ is a symmetric strictly definite positive matrix.

$\forall x \in \mathbb{R}^n$, $d(x, \mathbb{S}) = \inf\{\mathscr{N}(x,s) | s \in \mathbb{S}\}$ represents the distance between x and the set $\mathbb{S}$ where $\mathscr{N}$ is a norm on $\mathbb{R}^n$. For instance, $\mathscr{N}$ could be the Euclidean norm.

The Minkowski addition will be denoted by $\oplus$ and is defined by (4).

$$\forall (\mathbb{A}, \mathbb{B}) \in \mathbb{R}^n \times \mathbb{R}^n, \ \mathbb{A} \oplus \mathbb{B} = \{a + b, a \in \mathbb{A} \text{ and } b \in \mathbb{B}\}. \quad (4)$$

Equation 5 defines the product between a matrix M and a set $\mathbb{A}$.

$$\forall (M, \mathbb{A}), \ M\mathbb{A} = \{Ma, a \in \mathbb{A}\} \quad (5)$$

where M and $\mathbb{A}$ have compatible sizes.

The following operator can be applied to sets:

$$\forall (\mathbb{A}, \mathbb{B}) \subseteq \mathbb{R}^n \times \mathbb{R}^n, \ \mathbb{A} \backslash \mathbb{B} = \{x \mid x \in \mathbb{A} \text{ and } x \notin \mathbb{B}\}. \quad (6)$$

3 Formulation of the barrier functions

System design is often based on end-user specifications. Only specifications that can be mathematically formatted as constraints are considered in this paper. The main goal of this section is to present a way to consider constraints as barrier functions that will be used in the proposed BMPC. Properties required for sake of stability proof (section 6) are also provided. Once constraints are satisfied, the goal is to make the system remain as long as possible inside the corresponding set; in the case where the initial system state does not satisfy the constraints, the goal is to reach the feasible set.

We only consider inequality constraints defined by

$$c\left(\hat{X}, \hat{U}\right) \leqslant c_{max} \quad (7)$$

where $\hat{X}$ is any set of states evaluated at any sample time and $\hat{U}$ is any set of inputs evaluated at any sample time and $c_{max} \in \mathbb{R}$.

We admit that all constraints are at least satisfied at the origin ($c(0,0) \leqslant c_{max}$) which is the target point.

Equality constraints are unusual in industrial problems and can be easily converted into inequality constraint using (8), where $\varepsilon \in \mathbb{R}_+$ is a small number.

$$c\left(\hat{X}, \hat{U}\right) = 0 \qquad \rightarrow \qquad \left\| c\left(\hat{X}, \hat{U}\right) \right\| - \varepsilon \leqslant 0 \quad (8)$$

Constraints can be classified into two categories:

1. Constraints that are applied independently to each predicted state. For instance, an upper bound on a given state variable.
2. Constraints that are applied to a combination of different predicted states. For instance, the variance of a state over the predicted horizon.

The total number of constraints and the number of constraints which belong to the first type will be respectively denoted by N_c^* and N_c.

In the following, the constraint $u \in \mathbb{U}$ is not considered using a barrier function because the search space of a gradient free optimization algorithm is usually a compact set.

The largest set included in $\mathbb{X}$ where all constraints are satisfied will be denoted by $\tilde{\mathscr{C}} \subseteq \mathbb{X}$.

For stability purpose (see section 6), only constraints that belong to the first type will be taken into account in the cost function of the MPC as barrier functions. However, all constraints will be taken into account during the determination of $\tilde{\mathscr{C}}$.

It is worth emphasizing that barrier functions are only here to penalize the cost function when constraints are not satisfied, thus it cannot be ensured that constraints will always be satisfied. Moreover, because barrier functions can only be evaluated on the prediction horizon we cannot guarantee that constraints will be satisfied from a global point of view. Yet, increasing the value of N can increase probability to globally fulfill requirements.

The use of barrier functions permits to relax the problem and improve feasibility. This relaxation permits to always find a trajectory to reach the target point even if all constraints are not satisfied.

We will now give some properties required to prove stability of our proposed BMPC. The barrier function l_c for constraint c should be defined to satisfy:

$$\forall(\mathbf{u},k) \in \mathbb{U}^N \times [\![0;N]\!], \quad \left\{ \begin{array}{lll} l_c(\bar{x}_k,\mathbf{u}) = 0 & \text{if} & \bar{x}_k \in \tilde{\mathbb{X}}_c \\ l_c(\bar{x}_k,\mathbf{u}) \geqslant 0 & \text{if} & \bar{x}_k \notin \tilde{\mathbb{X}}_c \end{array} \right. \tag{9}$$

where $\tilde{\mathbb{X}}_c$ is the set where the constraint c is satisfied.

l_b is defined as the global barrier function:

$$\forall(\mathbf{u},k) \in \mathbb{U}^N \times [\![0;N]\!], \quad l_b(\bar{x}_k,\mathbf{u}) = \sum_{c=1}^{N_c} l_c(\bar{x}_k,\mathbf{u}). \tag{10}$$

Using (9) the following property holds:

$$\forall(\mathbf{u},k) \in \mathbb{U}^N \times [\![0;N]\!], \quad \left\{ \begin{array}{lll} l_b(\bar{x}_k,\mathbf{u}) = 0 & \text{if} & \bar{x}_k \in \tilde{\mathscr{C}} \\ l_b(\bar{x}_k,\mathbf{u}) \geqslant 0 & \text{if} & \bar{x}_k \notin \tilde{\mathscr{C}} \end{array} \right. \tag{11}$$

where $\tilde{\mathscr{C}}$ is defined by (12).

$$\tilde{\mathscr{C}} = \bigcap_{c=1}^{N_c^*} \tilde{\mathbb{X}}_c. \tag{12}$$

Since l_b is a barrier function that may be applied to all elements of the predicted state vector $\bar{\mathbf{x}}$, the notation $l_b(\bar{\mathbf{x}},\mathbf{u})$ will be used in the sequel. Properties presented in (11) still hold true for each element $\bar{x}_k$ of the predicted state $\bar{\mathbf{x}}$.

4 Determination of the unconstrained set

In this section a method to determine a set $\mathscr{C}$ as an approximation of $\tilde{\mathscr{C}}$ is presented. This set $\mathscr{C}$ has to be included or equal to $\tilde{\mathscr{C}}$ and to be positive invariant. Positive invariant sets correspond to definition 1 below, which will be used in section 6 to prove stability.

Definition 1. *A set $\mathscr{C}$ is positive invariant with respect to the system $x_{K+1} = Ax_K + Bu_K$ for the control law $u_K = \bar{K}x_K$ if for a state $x_K \in \mathscr{C}$ then for all $i \in N^*$, $x_{K+i} \in \mathscr{C}$.*

The set which corresponds to the set of all points that can reach $\mathscr{C}$ in N steps is denoted by $\mathscr{C}_r(N)$. The following property 1 holds true and the corresponding proof is given by proof 1.

Property 1. *If $\mathscr{C}$ is control invariant with respect to $\mathbb{U}$ then $\mathscr{C}_r(N)$ is control invariant with respect to $\mathbb{U}$.*

Proof 1. *Let x_K be an element of $\mathscr{C}_r(N)$. It means that it exists an input sequence $\boldsymbol{u} = [u_0,\ldots,u_{N-1}]$ which leads to a final predicted state $\bar{x}_N \in \mathscr{C}$. It has been supposed that the predictor is identical to the system thus, at the next time step, the current state will be $x_{K+1} = \bar{x}_1 = A\bar{x}_0 + Bu_0$. Because $\mathscr{C}$ is control invariant with respect to $\mathbb{U}$, it exists $u^+ \in \mathbb{U}$ such that $\bar{x}_N^+ = A\bar{x}_N + Bu^+ \in \mathscr{C}$. Therefore, the control vector $\boldsymbol{u}^+ = [u_1,\ldots,u_{N-1},u^+]$ leads to $\bar{x}_N^+ \in \mathscr{C}$. Thus $\mathscr{C}_r(N)$ is control invariant with respect to $\mathbb{U}$.* □

4.1 First approximation of $\tilde{\mathscr{C}}$: $\bar{\mathscr{C}}$

To begin with, by taking into account all specifications, a first approximation $\bar{\mathscr{C}}$ of $\tilde{\mathscr{C}}$ should be determined. This first approximation is not necessarily positive invariant. In some cases and particularly in the case of nonlinear constraints, the set $\tilde{\mathbb{X}}_c$ where a constraint c is satisfied is difficult to compute. That is why a set $\bar{\mathbb{X}}_c \subseteq \tilde{\mathbb{X}}_c$ is used as an approximation. Finally, a first approximation $\bar{\mathscr{C}}$ of $\tilde{\mathscr{C}}$ is:

$$\bar{\mathscr{C}} = \bigcap_{c=1}^{N_c^*} \bar{\mathbb{X}}_c. \tag{13}$$

For all constraint c, assumption 3 is done on $\bar{\mathbb{X}}_c$.

Assumption 3. *Every set $\bar{\mathbb{X}}_c$ contains the origin. As a consequence, $0 \in \bar{\mathscr{C}}$ which means that $\bar{\mathscr{C}}$ is not empty.*

As an example, in a stochastic context, considering a constraint on the variance σ^2 over the prediction horizon of the first element $\bar{x}^1$ of the predicted state $\bar{\mathbf{x}}$:

$$\sigma^2\left(\left[\bar{x}_0^1,\ldots,\bar{x}_N^1\right]\right) \leqslant a. \tag{14}$$

where $a \in \mathbb{R}_+$, a set $\bar{\mathbb{X}}_c$ for this constraint can be defined by:

$$\bar{\mathbb{X}}_c = \left\{x_k \ , \ \forall k \in [\![0;N]\!] \ , \ \bar{x}_k^1 \in \mathbb{R} \ , \ |\bar{x}_k^1| \leqslant \sqrt{a}\right\}. \tag{15}$$

4.2 Definition of the set $\mathscr{C}$

In this part, it is supposed that the first approximation $\bar{\mathscr{C}} \subseteq \tilde{\mathscr{C}}$ is known. In the following, we will deal with a convex polytope approximation of $\bar{\mathscr{C}}$. Convex polytope approximations of sets are discussed in (Bronstein, 2008).

The maximal positive invariant set $\mathscr{C}$ included in $\bar{\mathscr{C}}$ is determined using algorithm described in (Gilbert and Tan, 1991) for the system defined by (16). K_{LQ} is computed by solving a Linear Quadratic problem.

$$\left\{ \begin{array}{rcl} x_{K+1} & = & Ax_K + B(K_{LQ}x_K) \\ y_K & = & Cx_K + D(K_{LQ}x_K) \end{array} \right. . \tag{16}$$

This method will be illustrated by the example in section 7.

In the following, it is assumed that $\mathscr{C}$ is known and defined, therefore property 1 holds for $\mathscr{C}_r(N)$.

Remark 1. *Equation (11) still holds by substituting $\tilde{\mathscr{C}}$ for $\mathscr{C}$ because $\mathscr{C} \subseteq \tilde{\mathscr{C}}$.*

5 Definition of the cost function

5.1 Theoretical concept

We would like to ensure that $\bar{x}_N$ will reach $\mathscr{C}$ when the current state $x_K = \bar{x}_0$ is in $\mathbb{X}\backslash\mathscr{C}$. The set of trajectories corresponding to N steps forward from $\bar{x}_0$ such that $d(\bar{x}_N, \mathscr{C}) < d(\bar{x}_0, \mathscr{C})$ is denoted by $\mathbb{T}$. $\mathbb{T}_c \subset \mathbb{T}$ denotes the set of trajectories such that $\bar{x}_N \in \mathscr{C}$ which means that $d(\bar{x}_N, \mathscr{C}) = 0$.

Assumption 4 is considered.

Assumption 4. *$\bar{x}_0$ is in the maximal controllable set included in $\mathbb{X}$. As a consequence, $\mathbb{T}$ is not empty.*

Furthermore, by definition of $\mathscr{C}_r(N)$, $\mathbb{T}_c$ is not empty if and only if the current state $x_K = \bar{x}_0$ is in $\mathscr{C}_r(N)$.

The two following cases are considered:

1. If $\mathbb{T}_c$ is empty, then a LQ control law is applied. With this control law, the state is ensured to reach $\mathscr{C}_r(N)$
2. If $\mathbb{T}_c$ is not empty, then the best trajectory in $\mathbb{T}_c$ is found according to the cost function and constraints that belong to the first category presented in section 3.

Figure 1 shows the different cases for a two states system.

Following statements propose a mathematical formulation of the cost function that can be used as the criterion to be optimized in both cases.

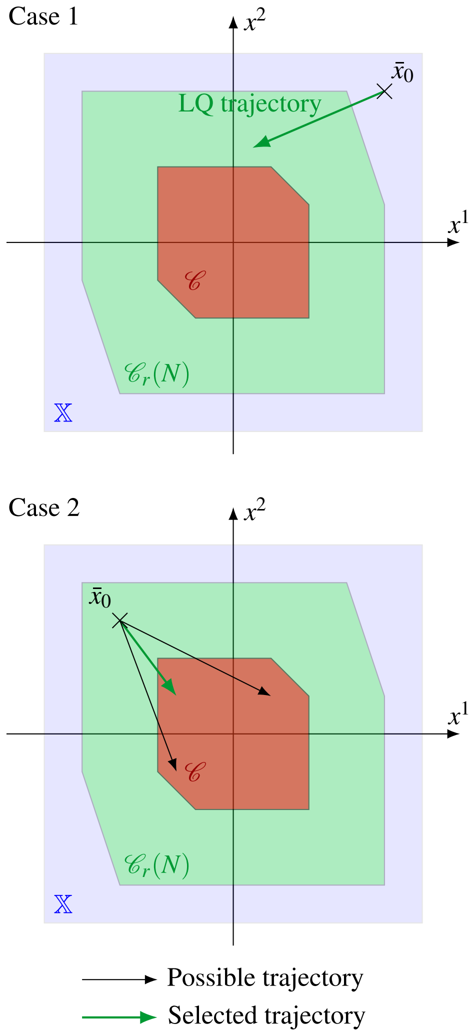

Figure 1: Graphic presentation of the two different cases

5.2 Mathematical formulation

When $\bar{x}_0 \notin \mathscr{C}_r(N)$, convergence has to be prioritized against constraints thus two objectives have to be considered corresponding to cases: $\bar{x}_0 \notin \mathscr{C}$ and $\bar{x}_0 \in \mathscr{C}_r(N)$.

5.2.1 Mathematical requirements for convergence for $x_K = \bar{x}_0 \notin \mathscr{C}$

In this case, we have to find if it exists a trajectory that can reach $\mathscr{C}$ which means that $d(\bar{x}_N, \mathscr{C}) = 0$ thus the global cost function is required to be the distance $d(\bar{x}_N, \mathscr{C})$:

$$J_1(\bar{\mathbf{x}}, \mathbf{u}) = d(\bar{x}_N, \mathscr{C}). \tag{17}$$

With this cost function the optimization algorithm is enforced to find a trajectory with the final predicted state in (or at least as close as possible to) the set $\mathscr{C}$ where all constraints are satisfied. This is equivalent to a terminal constraint.

5.2.2 Mathematical requirements for finding the best trajectory with regard to constraints for $x_K = \bar{x}_0 \in \mathscr{C}_r(N)$

For the second case, it is required to find the best trajectory in $\mathbb{T}_c$ according to the classical MPC cost function and the barrier functions. Thus, the cost function to be minimized is defined by (18).

$$J_2(\bar{\mathbf{x}}, \mathbf{u}) = \sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_N(\bar{x}_N) + l_b(\bar{\mathbf{x}}, \mathbf{u}). \tag{18}$$

where $l(\bar{x}_k, u_k) = \|\bar{x}_k\|_Q + \|u_k\|_R$ is the classical MPC cost, $l_N(\bar{x}_N)$ is a terminal cost and l_b corresponds to barrier functions (defined in section 3).

5.2.3 Mathematical formatting of the cost function

We want to consider one single optimization problem thus a cost function that takes into account both cases presented in section 5.1 is defined. For this purpose, a positive and a negative cost function will be respectively used for the first and the second case by using the involutory[1] transformation t defined by (19).

$$\forall x \in \mathbb{R}_+^*, \quad t(x) = -1/x. \tag{19}$$

The optimization process is finally based on the global cost function J_g defined by (20).

$$J_g(\bar{\mathbf{x}},\mathbf{u}) = \begin{cases} J_1(\bar{\mathbf{x}},\mathbf{u}) & \text{if } \bar{x}_0 \notin \mathscr{C}_r(N) \\ t\left(J_2(\bar{\mathbf{x}},\mathbf{u})\right) & \text{if } \bar{x}_0 \in \mathscr{C}_r(N) \text{ and } J_2(\bar{\mathbf{x}},\mathbf{u}) \neq 0 \\ -\infty & \text{if } \bar{x}_0 \in \mathscr{C}_r(N) \text{ and } J_2(\bar{\mathbf{x}},\mathbf{u}) = 0 \end{cases} \tag{20}$$

In the next section, it will be shown that this cost function permits to impose stability.

Remark 2. *This methods bears similarities with the one proposed in (Feyel, 2017) where H_∞ controllers are tuned using stochastic algorithms. The first step of the method consists of reducing the distance between the largest magnitude of eigenvalues of the closed loop and the unit circle and the second optimization step consists of finding the best controller according to specifications.*

6 Asymptotic stability

To begin with, we tried to apply recommendations proposed in (Mayne et al., 2000) in order to prove stability of a BMPC using the cost function defined by (18), however we have faced the need to assume that the barrier function defined by (10) is decreasing along the prediction horizon. This assumption is unsatisfying when we deal with some specific constraints such as standard deviation of the output which can only grow if the system starts from a stationary state. That is why the cost function (20) has been proposed.

In this section assumption 5 is considered.

Assumption 5. *The optimization algorithm always finds the global minimum of $\mathbb{P}_N$ defined by (3).*

In practice it is almost impossible to find the exact global solution at each step thus some works have been done in order to prove stability without assumption 5, (Reble and Allgöwer, 2012), (Allan et al., 2017), (Scokaert et al., 1999). Using these works, the following proof can be modified to tackle suboptimal solutions given by the optimizer. This point will be tackled in future works.

[1] An involutory transformation defines a transformation that is its own inverse.

In order to prove stability, we need to ensure that the following properties hold when solving $\mathbb{P}_N(\bar{x}_0)$:

1. If $\bar{x}_0$ is not in $\mathscr{C}_r(N)$, the convergence of the system state is ensured by applying a LQ control law.
2. If $\bar{x}_0$ is in $\mathscr{C}_r(N)$ we have Lyapunov stability.

With these two statements it is possible to prove that the system will reach the unconstrained set $\mathscr{C}$ and that the system is stable in the sense of Lyapunov in $\mathscr{C}_r(N)$.

Remark 3. *The decrease of the distance in the first statement does not imply that the value of the cost function decreases and the barrier function can be non convex. That is why, as said at the beginning of the section, we cannot follow the recommendation of (Mayne et al., 2000) to ensure stability.*

6.1 Convergence for $x_K = \bar{x}_0 \notin \mathscr{C}_r(N)$

The cost function has been formulated in order to firstly determine a trajectory such that $d(\bar{x}_N, \mathscr{C}) = 0$, but in case where $x_K = \bar{x}_0 \notin \mathscr{C}_r(N)$ such trajectory does not exist. In practice, it corresponds to a positive cost at the end of the optimization. In this case, a LQ control law is applied to the system. With this law, the system state is ensure to converge to $\mathscr{C}_r(N)$.

Remark 4. *The idea that consists of applying the sequence of input determined by the MPC algorithm that decreases the distance from $\mathscr{C}$ is not sufficient to ensure convergence thus it is not a viable solution.*

6.2 Convergence for $x_K = \bar{x}_0 \in \mathscr{C}_r(N)$

We now tackle the second point. To begin with, the following function V is defined.

$$\begin{aligned} &\forall (\bar{\mathbf{x}},\mathbf{u}) \in \mathscr{C}_r(N)^{N+1} \times \mathbb{U}^N, \\ &V(\bar{\mathbf{x}},\mathbf{u}) = t\left(J_g(\bar{\mathbf{x}},\mathbf{u})\right) = \frac{-1}{J_g(\bar{\mathbf{x}},\mathbf{u})}. \end{aligned} \tag{21}$$

By simplifications using the fact that $\bar{x}_0 \in \mathscr{C}_r(N)$ and (20):

$$\begin{aligned} &\forall (\bar{\mathbf{x}},\mathbf{u}) \in \mathscr{C}_r(N)^{N+1} \times \mathbb{U}^N, \\ &V(\bar{\mathbf{x}},\mathbf{u}) = J_2(\bar{\mathbf{x}},\mathbf{u}) \\ &\qquad = \sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_N(\bar{x}_N) + l_b(\bar{\mathbf{x}},\mathbf{u}). \end{aligned} \tag{22}$$

Because l and l_N are positive functions on $\mathbb{X} \times \mathbb{U}$ and l_b is a positive function on $\mathbb{X}^{N+1} \times \mathbb{U}^N$, we can conclude that V is a positive function on $\mathscr{C}_r(N)^{N+1} \times \mathbb{U}^N$. Moreover, $V(0,0) = 0$.

We can choose similarly to the warm start method presented in (Rawlings et al., 2017) (see Fig. 2):

$$\forall k \in [\![0; N-2]\!], \; u_k^+ = u_{k+1}, \tag{23}$$

u_{N-1}^+ is chosen such that $\bar{x}_N^+ \in \mathscr{C}$. The existence of u_{N-1}^+ such that $\bar{x}_N^+ \in \mathscr{C}$ is directly derived from definition 1 and property 1. The optimization algorithm could find an other vector $\mathbf{u}^+$, but this solution can only be better in terms of cost function (thanks to assumption 5) thus the proof proposed below still holds.

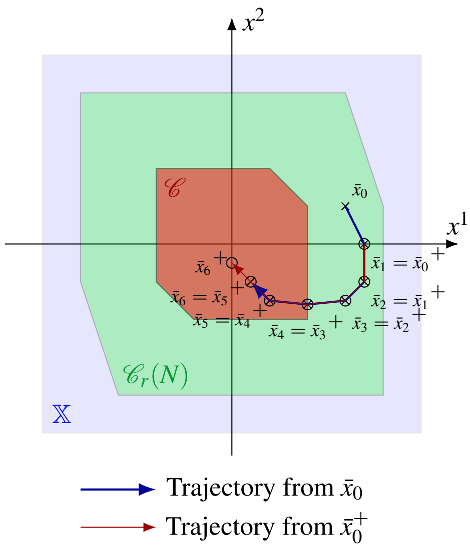

Figure 2: Example for $N = 6$

The next predicted state vector corresponding to $\mathbf{u}^+$ will be denoted by $\bar{\mathbf{x}}^+$. By definition, $\bar{\mathbf{x}}^+ = \left[\bar{x}_1, \ldots, \bar{x}_N, \bar{x}_N^+\right]$. The function V evaluated at the next time step is thus $V(\bar{\mathbf{x}}^+, \mathbf{u}^+)$.

We now evaluate $V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u})$,

$$\begin{aligned}
& V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) \\
= & \sum_{k=0}^{N-1} l(\bar{x}_k^+, u_k^+) + l_N(\bar{x}_N^+) + l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) \\
& - \left[\sum_{k=0}^{N-1} l(\bar{x}_k, u_k) + l_N(\bar{x}_N) + l_b(\bar{\mathbf{x}}, \mathbf{u})\right] \\
= & \sum_{k=0}^{N-1} \left[l(\bar{x}_k^+, u_k^+) - l(\bar{x}_k, u_k)\right] \\
& + l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - l_b(\bar{\mathbf{x}}, \mathbf{u}) \\
& + l_N(\bar{x}_N^+) - l_N(\bar{x}_N)
\end{aligned} \tag{24}$$

As a reminder, only barrier functions that correspond to constraints that belong to the first category presented in 3 are taken into account. Moreover, $\bar{\mathbf{x}}^+ = \left[\bar{x}_1, \ldots, \bar{x}_N, \bar{x}_N^+\right]$ by definition. Thus, the expression $l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - l_b(\bar{\mathbf{x}}, \mathbf{u})$ that appears in (24) can be simplified by looking only on influence of the first component: $\bar{x}_0$ and the last one: $\bar{x}_N^+$. In order to make it appears explicitly, we will note:

$$l_b(\bar{\mathbf{x}}^+, \mathbf{u}^+) - l_b(\bar{\mathbf{x}}, \mathbf{u}) = l_b(\bar{x}_N^+, \mathbf{u}^+) - l_b(\bar{x}_0, \mathbf{u}). \tag{25}$$

Moreover, by using (23), we can simplify (24):

$$\begin{aligned}
V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) = \; & l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) \\
& + l_b(\bar{x}_N^+, \mathbf{u}^+) - l_b(\bar{x}_0, \mathbf{u}) \\
& + l_N(\bar{x}_N^+) - l_N(\bar{x}_N)
\end{aligned} \tag{26}$$

By definition of $\mathscr{C}$, $\bar{x}_N^+ \in \mathscr{C} \Rightarrow l_b(\bar{x}_N^+, \mathbf{u}^+) = 0$ (see (11)). We finally have

$$\begin{aligned}
V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) = \; & l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) \\
& - l_b(\bar{x}_0, \mathbf{u}) \\
& + l_N(\bar{x}_N^+) - l_N(\bar{x}_N)
\end{aligned} \tag{27}$$

From (11), l_b is a positive function thus:

$$\begin{aligned}
& V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) \\
\leqslant \; & l_N(\bar{x}_N^+) - l_N(\bar{x}_N) + l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0)
\end{aligned} \tag{28}$$

The right part of inequality (28) corresponds to the classical equation that is found in the proof of stability of the original Model Predictive Control. According to (Rawlings et al., 2017), if we choose for instance $l_N(\bar{x}_N) = \bar{x}_N^{\mathrm{T}} P \bar{x}_N$ where P is the positive definite solution of the Ricatti equation:

$$P = Q + A^{\mathrm{T}} P A - A^{\mathrm{T}} P B (B^{\mathrm{T}} P B + R)^{-1} B^{\mathrm{T}} P A, \tag{29}$$

we can prove that

$$l_N(\bar{x}_N^+) - l_N(\bar{x}_N) + l(\bar{x}_{N-1}^+, u_{N-1}^+) - l(\bar{x}_0, u_0) < 0 \tag{30}$$

From (28), we finally have $V(\bar{\mathbf{x}}^+, \mathbf{u}^+) - V(\bar{\mathbf{x}}, \mathbf{u}) < 0$.

To conclude, V is a positive decreasing function with $V(0,0) = 0$ thus V is a Lyapunov function and we can conclude that the proposed BMPC is Lyapunov stable on $\mathscr{C}_r(N)$.

7 Application example

In this example, a motor with angular position and velocity respectively denoted by θ and Ω is considered. The considered motor can be modeled by a first order transfer function: $\theta(s)/u(s) = 0.9/(s * (1 + 0.015s))$ The sampled time is chosen to be $T_e = 1\text{ms}$ and the whole state is measured. The corresponding state space representation is:

$$\begin{cases}
x_{K+1} = \begin{pmatrix} 1.0 & 1.0e-3 \\ 0 & 9.4e-1 \end{pmatrix} x_K + \begin{pmatrix} 1.0e-4 \\ 5.8e-2 \end{pmatrix} u_K \\
y_K = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x_K
\end{cases} \tag{31}$$

where $x_K = \begin{pmatrix}\theta_K & \Omega_K\end{pmatrix}^{\mathrm{T}}$.

It is assumed that the set of constraint $\tilde{\mathscr{C}}$ is defined for all K in $\mathbb{N}$ by

$$\begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} x_K \leqslant \begin{pmatrix} 8 \\ 8 \end{pmatrix}. \tag{32}$$

Furthermore, $\forall K \in \mathbb{N},\ -10 \leqslant u_K \leqslant 10$.

In this particular case, all constraints are taken into account in the cost function because they all belong to the first category presented in section 3. In this example, exponential barrier functions defined by (33) have been used.

$$l_b(\bar{\mathbf{x}}, \mathbf{u}) = \alpha \max\left(0, \exp\left(\beta \max_{k \in [\![1,N]\!]} \{d(\Omega_k, \mathbb{X}_\Omega)\}\right) - 1\right) \tag{33}$$

where α and β are two tuning parameters respectively fixed to 5 and 1000. $\mathbb{X}_\Omega$ is the set where constraints over Ω are satisfied.

Algorithm developed by (Gilbert and Tan, 1991) is now applied to find a set $\mathscr{C} \subseteq \tilde{\mathscr{C}}$ that is positive invariant. As a result, the algorithm has converged in 14 steps to $\mathscr{C}$, this result is presented in Fig. 3. Moreover, algorithms developed by (Herceg et al., 2013) have been used to compute all sets.

Figure 4 shows some trajectories from different initial points of the proposed algorithm. The BMPC problem is solved using Perturbed Differential Evolution algorithm proposed in (Feyel, 2015) which is known to give results with small standard deviation from a run to the next. In order to emphasize the interest of the cost function defined by (20), Fig. 4 also shows results obtained without using the barrier function. Finally, as it can be seen in Fig. 5, constraints are satisfied faster with the proposed method than without using barrier functions.

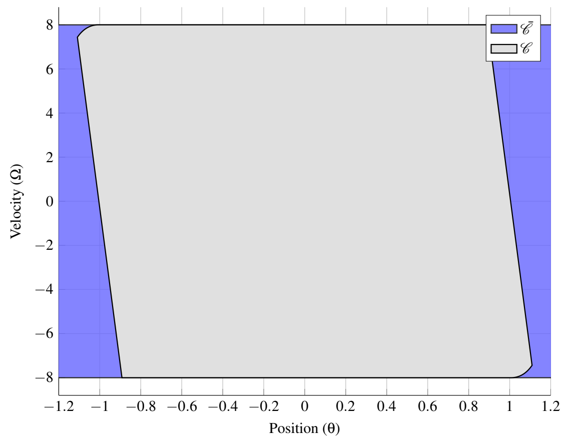

Figure 3: $\mathscr{C}$ computed by algorithm developed by (Gilbert and Tan, 1991)

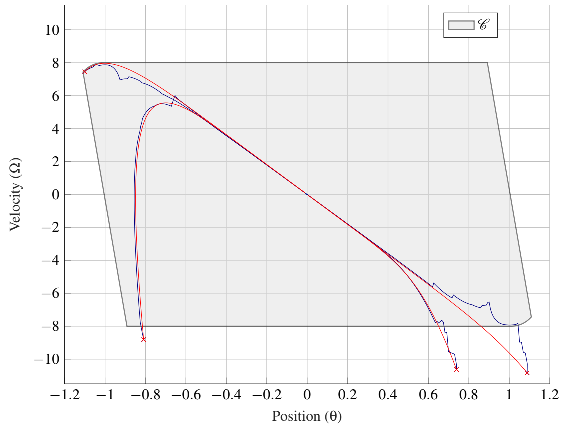

Figure 4: Trajectories using the same initial point using the proposed algorithm (in blue) and not using it (in red)

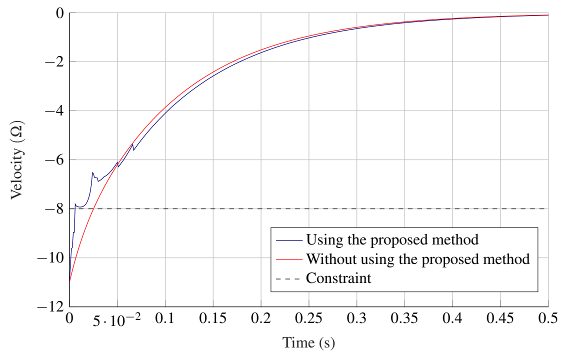

Figure 5: Temporal comparison of trajectories using the proposed method or not

8 Conclusion

We have proposed a MPC which can handle any kind of nonlinear constraints by including barrier functions in a particular cost function. Using this method requires to define a set $\mathscr{C}$ where all constraints are satisfied. Some important properties of the set have been presented and a method to define it has also been provided. Furthermore, we have proposed a proof of stability for our Barrier Model Predictive Control. The interest of the method has been shown on a numerical example.

In return of the use of a gradient free optimization algorithm, real time implementation will need an efficient code optimization for parallelization, or the use of a compact algorithm if the memory is limited. Moreover the characterization of $\mathscr{C}$ is sometimes difficult and can lead to conservative results. Future works will deal with a more efficient determination of this set.

This paper is restricted to the case of linear systems because application of algorithm developed by (Gilbert and Tan, 1991) and by extension definition of $\mathscr{C}$ is not direct for nonlinear systems. That is why future works will also deal with extension to nonlinear systems. We will also extend stability to the case of suboptimal solution given by the optimizer and improve robustness with respect to system uncertainties.

REFERENCES

Allan, D. A., Bates, C. N., Risbeck, M. J., and Rawlings, J. B. (2017). On the inherent robustness of optimal and suboptimal nonlinear MPC. *Systems & Control Letters*, 106:68 – 78.

Amrit, R., Rawlings, J. B., and Angeli, D. (2011). Economic optimization using model predictive control with a terminal cost. *Annual Reviews in Control*, 35(2):178 – 186.

Bronstein, E. M. (2008). Approximation of convex sets by polytopes. *Journal of Mathematical Sciences*, 153(6):727–762.

Camacho, E. F. and Bordons, C. (2007). *Model predictive control, Second Edition*. Springer International Publishing.

Chen, C. and Shaw, L. (1982). On receding horizon feedback control. *Automatica*, 18(3):349–352.

Chisci, L., Lombardi, A., and Mosca, E. (1996). Dual-receding horizon control of constrained discrete time systems. *European Journal of Control*, 4(2):278–285.

Feller, C. and Ebenbauer, C. (2015). Weight recentered barrier functions and smooth polytopic terminal set formulations for linear model predictive control. In *2015 American Control Conference (ACC)*, pages 1647–1652. IEEE.

Feller, C. and Ebenbauer, C. (2017). A stabilizing iteration scheme for model predictive control based on relaxed barrier functions. *Automatica*, 80:328–339.

Feyel, P. (2015). Perturbed differential evolution algorithm–enhancing the differential evolution algorithm by perturbing the mutants.

Feyel, P. (2017). *Robust Control Optimization with Metaheuristics*. John Wiley & Sons.

Gilbert, E. G. and Tan, K. T. (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic control*, 36(9):1008–1020.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland. http://control.ee.ethz.ch/mpt.

Keerthi, S. a. and Gilbert, E. G. (1988). Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of optimization theory and applications*, 57(2):265–293.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

Merabti, K. and Belarbi, H. (2014). Multi-objective predictive control: a solution using metaheuristics. *International Journal of Computer Science & Information Technology (IJCSIT)*.

Michalska, H. and Mayne, D. Q. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE transactions on automatic control*, 38(11):1623–1633.

Müller, M. A., Angeli, D., Allgöwer, F., Amrit, R., and Rawlings, J. B. (2014). Convergence in economic model predictive control with average constraints. *Automatica*, 50(12):3100–3111.

Petsagkourakis, P., Heath, W. P., Carrasco, J., and Theodoropoulos, C. (2019). Input-output stability of barrier-based model predictive control.

Price, K., Storn, R. M., and Lampinen, J. A. (2006). *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.

Rawlings, J. B., Mayne, D. Q., and Diehl, M. (2017). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing.

Rawlings, J. B. and Muske, K. R. (1993). The stability of constrained receding horizon control. *IEEE transactions on automatic control*, 38(10):1512–1516.

Reble, M. and Allgöwer, F. (2012). Unconstrained model predictive control and suboptimality estimates for nonlinear continuous-time systems. *Automatica*, 48(8):1812–1817.

Scokaert, P. O., Mayne, D. Q., and Rawlings, J. B. (1999). Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654.

Wills, A. G. and Heath, W. P. (2004). Barrier function based model predictive control. *Automatica*, 40(8):1415–1422.

Zeilinger, M. N., Jones, C. N., and Morari, M. (2010). Robust stability properties of soft constrained mpc. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5276–5282. IEEE.