Maximal Closed Loop Admissible Set for Linear Systems with Non-Convex Polyhedral Constraints

Emilio Pérez[a], Carlos Ariño[a], F.Xavier Blasco[b], Miguel A. Martínez[b]

[a]*Departamento de Ingeniería de Sistemas Industriales y Diseño, Universitat Jaume I, Avenida Vicent Sos Baynat, s/n. 12071 Castelló de la Plana, Spain.*
[b]*Instituto Universitario de Automática e Informática Industrial. Universidad Politécnica de Valencia. Camino de Vera S/N 46022 - Valencia, Spain.*

Abstract

This paper deals with the computation of the maximal closed-loop admissible set for linear systems with non-convex polyhedral constraints. These constraints are modeled as the union of a finite number of convex polyhedra. An efficient algorithm for the computation of this set, based on removing subsets of the maximal closed-loop invariant set for the convex hull of the original constraints, is proposed and compared with the existing generic algorithm. Next, stability conditions for a general predictive control scheme are applied to the particular problem proposed. Finally, two examples showing the performance of the maximal admissible set algorithm are given.

Keywords: Maximal Admissible Set, Non-convex Constraints, Model Predictive Control, Stability

1. Introduction

Closed loop admissible sets for discrete systems are defined as the set of initial states from which the evolution of a closed loop system satisfies specified pointwise-in-time constraints, both in the states an the inputs. The characterization of such sets is an important propriety for the performance of the closed loop system.

Email addresses: pereze@esid.uji.es (Emilio Pérez), arino@esid.uji.es (Carlos Ariño), xblasco@isa.upv.es (F.Xavier Blasco), mmiranzo@isa.upv.es (Miguel A. Martínez)

Furthermore, closed loop admissible sets have an important role in the field of stability of model predictive control (MPC). MPC is nowadays a mature control framework based on a finite horizon open loop optimal control problem which yields an optimal control sequence. The control action is obtained by a receding horizon strategy, i.e. applying only the first control action in the optimal sequence.

Sufficient conditions that actively guarantee the *a priori* stability of predictive control schemes include the use as a terminal constraint of a closed loop admissible set for an specified terminal controller [1].

Non-convex polyhedral constraints, defined as the non-convex union of a finite number of convex polyhedra, arise naturally in problems such as obstacle avoidance, which is inherently non-convex. The importance of this problem is stressed in [2], and it appears naturally in many practical control engineering problems, such as robot path planning [3] or aircraft traffic planning [4].

Another field of application of non-convex polyhedral constraints comes from linear systems with non-linear constraints. These problems usually appear when controlling Hammerstein systems, defined by an static non-linearity followed by a linear system. These models are often controlled by an inversion of the non-linearity and an appropriate controller, such as predictive control, for the linear part [5][6]. However, constraints on the system inputs have to be transformed by the non-linearity in order to be coped with by the predictive controller. If the constraints are non-convex, precise approximation require the use of several polyhedra with a non-convex union.

Furthermore, non-convex constraints also arise in feedback linearization control with linear constraints. Feedback linearization control is a widely used technique for controlling non-linear systems [7]. Clearly, the linear constraints of the system become non-linear for the linearized model [8], [9]. Once again, if these non-linear constraints are non-convex, they can be approximated by the union of several polyhedra.

Linear systems with non-convex polyhedral constraints can also be treated as piecewise affine (PWA) systems. For this kind of models, which also allow to represent a wide range of discrete-time hybrid systems [10], several approaches for the computation of closed-loop admissible sets have been presented in the literature in the context of stability of finite-time constrained optimal control (FTCOC) and predictive control.

A simple one is taken in [11], where the origin is chosen as the admissible

set. However, this approach typically deteriorates the overall performance of the system. A different suboptimal approach, which uses a polyhedral terminal set and a reference governor, is proposed in [12]. This methodology reduces the computational burden of the online controller at the cost of suboptimality and a smaller region of attraction. A less conservative solution is achieved by using a piecewise quadratic terminal cost and piecewise linear terminal control, as in [13]. This requires the computation of a terminal set, either the maximal positively invariant set, as in [14], or a more simple subset of it, as in [13].
Although the system with non-convex polyhedral constraints can be modeled as piecewise affine, fully exploiting the structure of the initial problem presents the fundamental advantage that invariant sets can be calculated in a more simple way.

The structure of the paper is as follows. Next section introduces the problem formulation. In section 3 an efficient algorithm for the computation of the maximal closed-loop admissible set is presented. In section 4, the application of stability conditions to a predictive control scheme with non-convex polyhedral constraints is discussed. Finally, section 5 gives two examples showing the performance of the invariant set calculation algorithm.

2. Problem statement

Let us consider a discrete-time linear time invariant system of the form (1)

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \tag{1}$$

with input and states constraints, which we assume that contain the origin of their respective spaces, defined as the non-convex union of a finite number of convex polytopes

$$u_k \in \mathbb{U} = \bigcup_{i=0}^{\gamma_u} \mathbb{U}_i, \qquad x_k \in \mathbb{X} = \bigcup_{i=0}^{\gamma_x} \mathbb{X}_i \tag{2}$$

and a linear control law $u_k = \mathcal{K}_f(x) = -Kx$.

In this work, we aim to find the the maximal closed-loop admissible set $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$, that is, the set of states for which the closed-loop trajectory satisfies the state and input constraints, $\mathbb{X}$ and $\mathbb{U}$:

$$\begin{aligned}\mathcal{O}_{\infty}^{\mathcal{K}_f}(\mathbb{U},\mathbb{X}) = \{x \in \mathbb{R}^n | K(A-BK)^k x \in \mathbb{U},\\ (A-BK)^k x \in \mathbb{X} \text{ for } k = 0,1,\ldots\}\end{aligned}$$

It is well known that for discrete-time systems, if $\mathcal{O}_{\infty}^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$ is not empty, it is a positively invariant set [15].

In order to propose our algorithm for the computation of the maximal closed-loop admissible set, we introduce here set definitions from [16]. In the following, Ω represents a non-empty arbitrary set in the x space, $h(x)$ is a given control law, and $\mathbb{U}$ is a constraint set in the inputs space.

Definition 1. *Given a control law $u_k = h(x_k)$, the input admissible subset of $\Omega \subset \mathbb{R}^n$ is given by*

$$\Omega^h(\mathbb{U}) = \{x_k \in \Omega | h(x_k) \in \mathbb{U}\}$$

Definition 2. *The closed loop input admissible one-step set $\mathcal{Q}^h(\mathbb{U},\Omega)$ is the set of states in $\mathbb{R}^n$ from which the closed loop system is guaranteed to evolve to Ω with an admissible input $u_k = h(x_k) \in \mathbb{U}$, i.e.*

$$\mathcal{Q}^h(\mathbb{U},\Omega) = \{x_k \in \mathbb{R}^n | h(x_k) \in \mathbb{U}; f(x_k, h(x_k)) \in \Omega\}$$

Definition 3. *The i-step controllable set $\mathbb{K}_i(\mathbb{U},\Omega,\mathbb{T})$ is the largest set of states in Ω for which there exists an admissible sequence of inputs such that an arbitrary target set $\mathbb{T} \subset \Omega$ is reached in exactly i steps, while keeping the state inside Ω for the first $i-1$ steps, i.e.*

$$\begin{aligned}\mathbb{K}_i(\mathbb{U},\Omega,\mathbb{T}) = \{x_0 \in \mathbb{R}^n | \exists \{u_k \in \mathbb{U}\}_0^{i-1};\\ \{x_k \in \Omega\}_0^{i-1}, x_i \in \mathbb{T}\}\end{aligned}$$

Definition 4. *The set $\mathcal{KO}_i^h(\mathbb{U},\Omega,\mathbb{T})$ for the system $x_{k+1} = f(x_k,u_k)$ in closed-loop with the control law $u_k = h(x_k)$ is defined as $\mathbb{K}_i(\mathbb{U},\Omega^h,\mathbb{T})$ for the system $x_{k+1} = f(x_k, h(x_k))$:*

$$\begin{aligned}\mathcal{KO}_i^h(\mathbb{U},\Omega,\mathbb{T}) = \{x_0 \in \mathbb{R}^n | \{h(x_k) \in \mathbb{U}\}_0^{i-1};\\ \{x_k \in \Omega\}_0^{i-1}, x_i \in \mathbb{T}\}\end{aligned}$$

3. Maximal closed-loop admissible set computation

The most direct way of computing the maximal closed-loop admissible set $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$ is to note that it is equal to the set $\mathcal{KO}_\infty^{\mathcal{K}_f}(\mathbb{U},\Omega,\Omega^{\mathcal{K}_f})$ and that, by definition, this set can be calculated as the infinite-time input admissible controllable set for the closed loop system $x_{k+1} = f(x_k, \mathcal{K}_f(x_k))$: $\mathbb{K}_\infty(\mathbb{U},\Omega^{\mathcal{K}_f},\Omega^{\mathcal{K}_f})$.
This set can be computed by means of the recursive algorithm 1 from [16].

Algorithm 1 Calculation of the closed-loop N-step controllable set $\mathbb{K}_N(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T})$

```
1. Make $i = 0$ and $\mathbb{K}_0(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T}) = \mathbb{T}$
2. While $i < N$:
   (a) $\mathbb{K}_{i+1}(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T}) = \mathcal{Q}^{\mathcal{K}_f}\left(\mathbb{U},\mathbb{K}_i(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T})\right) \cap \Omega^{\mathcal{K}_f}$
   (b) If $\mathbb{K}_{i+1}(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T}) = \mathbb{K}_i(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T})$, end algorithm and
       $\mathbb{K}_N(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T}) = \mathbb{K}_\infty(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T}) = \mathbb{K}_i(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T})$.
   (c) i=i+1
```

This algorithm starts by taking the target set $\mathbb{T}$ and then it calculates the subset of Ω that takes the system to $\mathbb{T}$ in one step, $\mathbb{K}_1(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T})$. Next, this calculated set is defined as the new target set and the procedure is repeated. The algorithm finishes when it arrives to step N or when a $\mathbb{K}_i(\mathbb{U},\Omega^{\mathcal{K}_f},\mathbb{T})$ equal to the previous iteration one is found.

Algorithm 1 is used in [14] for the computation of the maximal closed-loop invariant set for piecewise affine systems.

As can be seen, algorithm 1 makes recursive use of the one-step input admissible set, $\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U},\Omega)$. In order to calculate this set, when Ω is a non-convex polyhedron the following property is needed:

Proposition 1. *[16] If Ω is given as the union*

$$\Omega = \bigcup_i \Omega_i,$$

then

$$\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U},\Omega) = \bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U},\Omega_i)$$

In an analogous way, when $\mathbb{U}$ is a non-convex polyhedra we can formulate the following proposition:

Proposition 2. *If $\mathbb{U}$ is given as the union*

$$\mathbb{U} = \bigcup_i \mathbb{U}_i,$$

then

$$\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega) = \bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$$

Proof. If $x_k \in \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega)$, by definition, $u_k = \mathcal{K}_f(x_k) \in \mathbb{U}$ and $x_{k+1} \in \Omega$. But if $u_k \in \mathbb{U}$, then $u_k \in \mathbb{U}_i$ for one or more i, so $x_k \in \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$, which implies $x_k \in \bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$. Therefore, it is proved $\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega) \subseteq \bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$. In the opposite way, if $x_k \in \bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$, for some i $x_k \in \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$ holds. Using the definition of the closed-loop one-step set, $u_k = \mathcal{K}_f(x_k) \in \mathbb{U}_i$ for some i and $x_{k+1} \in \Omega$. But for any $u_k \in \mathbb{U}_i$, $u_k \in \mathbb{U}$ holds, so $x_k \in \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega)$. This way it is proved that $\bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega) \subseteq \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega)$ and therefore $\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega) = \bigcup_i \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega)$. □

Using both propositions in our particular problem:

$$\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \Omega) = \bigcup_i \bigcup_j \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_i, \Omega_j) \tag{3}$$

Recursive algorithm 1 has an important drawback motivated from the fact that the one-step set (3) will in general be also non-convex. As the algorithm needs recursively each one-step set to calculate the following one, $\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X}))$, the number of non-convex sets will grow up in a combinatorial explosion way. This fact, apart from making the algorithm more computationally expensive, will mean that if the set is used as a terminal set in an MPC scheme without further simplification[1], the complexity of the optimization problem associated to the predictive controller will be high.

In order to design a different algorithm, we first introduce the following set definitions:

[1] An algorithm for the union of regions, such as the one in [17] can be used at this point, with the drawback of a considerably increased computation time.

Definition 5. *The* i-step input admissible closed loop set $\mathcal{Q}_i^h(\mathbb{U},\Omega,\mathbb{T})$ *is the set of states for which the closed loop evolution given by $x_{k+1}=f(x_k,h(x_k))$ drives the state to $\mathbb{T}$ in i steps or less, while keeping the evolution of the state and the control action $u_k=h(x_k)$ inside, respectively, Ω and $\mathbb{U}$:*

$$\mathcal{Q}_i^h(\mathbb{U},\Omega,\mathbb{T})=\{x_0\in\mathbb{R}^n|\exists N\le i:\\ \{u_k=h(x_k)\in\mathbb{U}\}_0^{N-1},\{x_k\in\Omega\}_0^{N-1},x_N\in\mathbb{T}\}$$

This set can be calculated by obtaining the sets that drive the closed loop system to $\mathbb{T}$ in exactly i steps, i.e. $\mathcal{KO}_i^h(\mathbb{U},\Omega,\mathbb{T})$:

$$\mathcal{Q}_i^h(\mathbb{U},\Omega,\mathbb{T})=\bigcup_{j=1}^{i}\mathcal{KO}_j^h(\mathbb{U},\Omega,\mathbb{T}) \tag{4}$$

Definition 6. *The i-step input admissible closed loop set for $i=\infty$ is the* maximal input admissible closed loop set $\mathcal{Q}_\infty^h(\mathbb{U},\Omega,\mathbb{T})$ *and is said to be* finitely determined *if there exists some $i^*\in\mathbb{N}$ such that $\mathcal{Q}_\infty^h(\mathbb{U},\Omega,\mathbb{T})=\mathcal{Q}_{i^*}^h(\mathbb{U},\Omega,\mathbb{T})$.*

Note that we do not assume that target set $\mathbb{T}$ is invariant, therefore an initial condition $x_0\in\mathcal{Q}_\infty^h(\mathbb{U},\Omega,\mathbb{T})$ will drive the closed-loop system to $\mathbb{T}$ in a given number of steps but it is unknown if the trajectory will remain inside this set. If the target set is defined as a subset of the state space for which some problem constraints do not hold, the set $\mathcal{Q}_\infty^h(\mathbb{U},\Omega,\mathbb{T})$ represents the states that violate those constraints at some point in their closed loop trajectories.

In order to find a practical way to calculate $\mathcal{Q}_\infty^h(\mathbb{U},\Omega,\mathbb{T})$ we formulate the following proposition:

Proposition 3. *If the closed-loop system $x_{k+1}=f(x_k,h(x_k))$ is asymptotically stable and $0\notin\mathbb{T}$, there exists some $i^*\in\mathbb{N}$ such that $\mathcal{KO}_i^h(\mathbb{U},\Omega,\mathbb{T})=\emptyset$ $\forall i\ge i^*$.*

Proof. If $0\notin\mathbb{T}$, there exists some $\epsilon>0$ such that if $||x||<\epsilon$, then $x\notin\mathbb{T}$. On the other hand, if the closed-loop system is asymptotically stable, there exists some $i^*\in\mathbb{N}$ such that $||x_i||<\epsilon$ $\forall i\ge i^*$. Therefore, for $i\ge i^*$ $||x_i||\notin\mathbb{T}$ and $\mathcal{KO}_i^h(\mathbb{U},\Omega,\mathbb{T})=\emptyset$. □

Remark 1. *Note that if the closed-loop system is asymptotically stable and $0\notin\mathbb{T}$, from proposition 3 and* (4)*, $\mathcal{Q}_\infty^h(\mathbb{U},\Omega,\mathbb{T})$ is finitely determined.*

Previous definitions and proposition can be used to design a different algorithm for the calculation of $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$.

To start the procedure, we need the convex hulls of the constraint sets $\bar{\mathbb{U}} = \mathbf{conv}(\mathbb{U})$ and $\bar{\mathbb{X}} = \mathbf{conv}(\mathbb{X})$ and the complement of these constraint sets in their respective hulls. These complementary sets will in general be also non-convex, defined as the union of a problem-dependent number of convex polyhedra:

$$\bar{\mathbb{U}} \backslash \mathbb{U} = \bar{\mathbb{U}} \backslash \left(\bigcup_i \mathbb{U}_i \right) = \bigcup_l \mathbb{U}_l^* \qquad \bar{\mathbb{X}} \backslash \mathbb{X} = \bar{\mathbb{U}} \backslash \left(\bigcup_j \mathbb{U}_j \right) = \bigcup_m \mathbb{X}_m^*.$$

In many cases, such as obstacle avoidance problems, it is not necessary to calculate the hulls and complementary sets, as the problem constraints are defined as a convex set and forbidden subsets inside it.

Note that, as by assumption $\mathbb{U}$ and $\mathbb{X}$ contain the origin of their spaces, their complements do not contain it:

$$\begin{aligned} 0 \notin \mathbb{U}_l^*, &\quad \forall l \\ 0 \notin \mathbb{X}_m^*, &\quad \forall m \end{aligned}$$

The algorithm is started by calculating the maximal admissible set $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})$. As $\bar{\mathbb{U}}$ and $\bar{\mathbb{X}}$ are convex, this set is easily obtained by algorithm 1.
As the original constraints are included in their respective hulls, by definition we have $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X}) \subset \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})$.
Therefore, the idea for the algorithm is to start from this set and eliminate from it any state for which the closed-loop trajectory violates at some point either the state or the input constraints. This set of states that need to be eliminated is denoted as $\mathbb{X}_f^*$.

It is clear that, for any $\mathbb{U}_l^*$, any state for which $u_k = \mathcal{K}_f(x_k) \in \mathbb{U}_l^*$ must be eliminated. From our sets definitions, this states are represented as $\mathcal{T}_l = \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_l^*, \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}}))$. Furthermore, any state that, in any number of steps, drives the future trajectory of the closed-loop system to the previous set, must be eliminated too. These states can be represented as $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}},\mathcal{T}_l)$.

As $\mathbb{U}_l^*$ and $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})$ are convex, so is $\mathcal{T}_l$, which is easily calculated from the definition of closed loop one step set. On the other hand, $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}},\mathcal{T}_l)$

is also defined over convex sets. Furthermore, as $0 \notin \mathbb{U}_l^*$, proposition 3 can be applied and the set is finitely determined.

In an analogous way, every state inside any $\mathbb{X}_m^*$ has to be eliminated: $\mathcal{D}_m = \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}) \cap \mathbb{X}_m^*$. Once again, it also has to be eliminated every state that drives the future trajectory to the previous set: $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{D}_m)$. Clearly, as $0 \notin \mathbb{X}_m^*$, the last set is also finitely determined by application of proposition 3.

Therefore, the set of states $\mathbb{X}_f^*$ that have to be eliminated from $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}})$ is:

$$\mathbb{X}_f^* = \bigcup_l (\mathcal{T}_l \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_l)) \cup \bigcup_m (\mathcal{D}_m \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{D}_m))$$

Finally, the maximal input admisible invariant set can be computed as:

$$\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X}) = \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}) \backslash \mathbb{X}_f^*$$

Algorithm 2 summarizes the described procedure.

3.1. Efficiency of the algorithms

As previously stated, algorithm 1 has the fundamental drawback of the exponential growth of the worst-case number of regions. This growth is defined by the number of iterations that the algorithm needs to converge, i^*, and depends on the closed-loop system dynamics.

Let us consider input and state constraints defined as (2). Algorithm 1 starts by making

$$\mathbb{K}_0(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f}) = \mathbb{X}^{\mathcal{K}_f} = \bigcup_{k=0}^{\gamma_x} \mathbb{X}_k$$

Next, in any given iteration, in step 2.b one needs to calculate the set:

$$\mathbb{K}_{i+1}(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f}) = \mathcal{Q}\left(\mathbb{U}, \mathbb{K}_i(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f})\right) \cap \mathbb{X}^{\mathcal{K}_f}$$

By (3), this set can be written as:

$$\begin{aligned} \mathbb{K}_{i+1}(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f}) &= \left(\bigcup_j^{\gamma_u} \bigcup_k^{\gamma_{\mathbb{K}_i}} \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_j, \mathbb{K}_{ik}(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f}))\right) \cap (\bigcup_l^{\gamma_x} \mathbb{X}_l) \\ &= \bigcup_j^{\gamma_u} \bigcup_k^{\gamma_{\mathbb{K}_i}} \bigcup_l^{\gamma_x} \left(\mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_j, \mathbb{K}_{ik}(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f})) \cap \mathbb{X}_l\right) \end{aligned}$$

where $\gamma_{\mathbb{K}_i}$ represents the number of convex polyhedra that define $\mathbb{K}_i(\mathbb{U}, \mathbb{X}^{\mathcal{K}_f}, \mathbb{X}^{\mathcal{K}_f})$.

Algorithm 2 Computation of $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$

```
1. Initialize the subset of states to eliminate, $\mathbb{X}_f^* = \emptyset$.
2. Calculate the convex hull of each constraints set, if not given: $\bar{\mathbb{U}} = \mathbf{conv}(\mathbb{U})$, $\bar{\mathbb{X}} = \mathbf{conv}(\mathbb{X})$.
3. Calculate the maximal input admissible invariant set for them: $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})$.
4. Calculate the complement of $\mathbb{U}$ and $\mathbb{X}$ in their hulls, if not given: $\bar{\mathbb{U}}\backslash\mathbb{U} = \bigcup_l \mathbb{U}_l^*$, $\bar{\mathbb{X}}\backslash\mathbb{X} = \bigcup_m \mathbb{X}_m^*$.
5. For each $\mathbb{U}_l^*$:
    (a) Obtain the subset of $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})$ such that $u_k = \mathcal{K}_f(x_k) \in \mathbb{U}_l^*$: $\mathcal{T}_l = \mathcal{Q}^{\mathcal{K}_f}(\mathbb{U}_l^*, \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}}))$
    (b) Calculate the maximal input admissible closed loop set: $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}},\mathcal{T}_l)$
    (c) Add those sets to $\mathbb{X}_f^*$:
        $\mathbb{X}_f^* = \mathbb{X}_f^* \cup \mathcal{T}_l \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}},\mathcal{T}_l)$
6. For each $\mathbb{X}_m^*$:
    (a) Obtain the subset of $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})$ such that $x_k \in \mathbb{X}_m^*$: $\mathcal{D}_m = \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}}) \cap \mathbb{X}_m^*$.
    (b) Calculate the maximal input admissible closed loop set: $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}},\mathcal{D}_m)$
    (c) Add those sets to $\mathbb{X}_f^*$:
        $\mathbb{X}_f^* = \mathbb{X}_f^* \cup \mathcal{D}_m \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}},\mathcal{D}_m)$
7. Compute the final set as:

        $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X}) = \mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}},\bar{\mathbb{X}})\backslash\mathbb{X}_f^*$
```

In a worst-case scenario, none of the polyhedra that form $\mathbb{K}_{i+1}$ can be united with each other to form a new convex polyhedra, and therefore, this set is defined by $\gamma_u \cdot \gamma_x \cdot \gamma_{\mathbb{K}i}$ regions.

Considering this worst-case scenario for each iteration of the algorithm, we have the region in each step defined by the following number of convex polyhedra:

$$\gamma_{\mathbb{K}i} = \gamma_{\mathbb{K}0} \cdot (\gamma_u \gamma_x)^i = \gamma_x (\gamma_u \gamma_x)^i$$

Therefore, the number of sets that define $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ in the worst case is given by

$$\gamma_{\mathbb{K}i^*} = \gamma_x (\gamma_u \gamma_x)^{i^*}$$

which is clearly exponential on i^* and polynomial on γ_u and γ_x.

On the other hand, algorithm 2 requires the computation of the set $\mathbb{X}_f^*$. Therefore, the efficiency of the algorithm will be analyzed in terms of the cost of the computation of this set. Taking the complements of the constraint sets, $\mathbb{U}^* = \bigcup_l^{\gamma_u^*} \mathbb{U}_l^*$ and $\mathbb{X}^* = \bigcup_m^{\gamma_x^*} \mathbb{X}_m^*$, for each of the convex polyhedra we obtain the set $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_l)$ or $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{D}_m)$.

These sets are defined by (4) and the analogous expression for $\mathcal{D}_m$:

$$\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_l) = \bigcup_{j=1}^{i_l^*} \mathbb{K}_j(\bar{\mathbb{U}}, \bar{\mathbb{X}}^{\mathcal{K}_f}, \mathcal{T}_l^{\mathcal{K}_f})$$

where i_l^* depends on the closed-loop system dynamics.

Each of the $\mathbb{K}_j$ has to be calculated by algorithm 1. However, unlike for the previous case, as $\bar{\mathbb{U}}$, $\bar{\mathbb{X}}$ and $\mathcal{T}_l$ (or $\mathcal{D}_m$) are convex sets, $\mathbb{K}_j$ is a single convex polyhedra. Therefore, $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_l)$ $(\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{D}_m))$ is formed in the worst case by the non-convex union of i_l^* (i_m^*) convex polyhedra. Now, considering that each $\mathcal{T}_l$ and $\mathcal{D}_m$ is formed by a single polyhedron, the total number of convex polyhedra that form $\mathbb{X}_f^*$ in the worst case can be calculated as:

$$\gamma_{xf}^* = \sum_{l=1}^{\gamma_u^*}(1 + i_l^*) + \sum_{m=1}^{\gamma_x^*}(1 + i_m^*) \leq (\gamma_u^* + \gamma_x^*)(2 + i_{max}^*)$$

where i_{max}^* is the maximum value of every i_l^* and i_m^*. It can be seen that γ_{xf}^* is bounded by a linear expression on i_{max}^*, γ_u and γ_x.

The number of polyhedra that define $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ when calculated by algorithm 2 is not γ_{xf}^* but a number dependent on each particular problem. However, it is in general in the same order that γ_{xf}^*.

Therefore, for closed-loop systems with relatively high values for the parameters of i^*, i_l^* and i_m^*, in the worst-case $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ obtained by algorithm 2 will be defined by a considerably inferior number of regions than this same set obtained by algorithm 1, because of the exponential nature of $\gamma_{\mathbb{K}i^*}$. Consequently, the latter algorithm will be more computationally expensive. Furthermore, if the number of final convex regions is reduced by applying algorithms from [17] for both algorithms, algorithm 2 is even more efficient because in [17] the hull and complement of the final sets are needed. This sets have already been calculated in algorithm 2, but not in algorithm 1. The difference between both algorithms' cost is illustrated in examples of section 5.

4. Stability of MPC scheme with non-convex polyhedral constraints

One of the fundamental applications of maximal closed-loop admissible sets is its use as a terminal region in order to assure stability of model predictive control schemes.

Model predictive control solves regulation problems by defining a value function and solving the finite horizon optimal control problem (5) and later applying a receding horizon policy.

$$
\begin{aligned}
\mathcal{P}_N(x):\ \ & V_N^{OPT}(x) := \min V_N(\{x_k\}, \{u_k\}), \\
& \\
& \text{subject to:} \\
& x_{k+1} = f(x_k, u_k) \text{ for } k = 0, \dots, N-1, \\
& x_0 = x, \\
& u_k \in \mathbb{U} \text{ for } k = 0, \dots, N-1, \\
& x_k \in \mathbb{X} \text{ for } k = 1, \dots, N, \\
& x_N \in \mathbb{X}_f \subset \mathbb{X}, \\
& V_N(\{x_k\}, \{u_k\}) := F(x_N) + \textstyle\sum_{k=0}^{N-1} L(x_k, u_k)
\end{aligned}
\tag{5}
$$

where $L(x, u)$, $F(x)$ and $\mathbb{X}_f$ are, respectively, the per-stage weighting, the terminal state weighting and the terminal constraint set.

The minimizing solution to the optimization problem is a control sequence function of the current state x

$$\mathbf{u}^{opt} := \{u_0^{\text{opt}}, u_1^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}\}$$

and, by the receding horizon policy, the control applied to the plant is only the first element of the sequence

$$\mathcal{K}_N(x) = u_0^{\text{opt}} \tag{6}$$

In order to get a good system performance with a predictive controller, the value function is chosen with a quadratic per-stage weighting $L(x, u)$:

$$V_N(x) = F(x_N) + \frac{1}{2}\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k) \tag{7}$$

In this section, we aim to formulate a priori conditions that assure the stability of the predictive controller scheme with linear system (1), non-convex constraints (2) and value function (7).

4.1. Stability conditions

Stability of predictive control schemes such as (5)-(6) has been a widely studied topic. A good synthesis that covers the state of the art can be found in [1], where authors introduce stability conditions. Based on them, in [18] it is proposed the following theorem:

C1 The per-stage weighting $L(x, u)$ in (5) satisfies $L(0, 0) = 0$ and $L(x, u) \geq \gamma(||x||)$ for all x inside the set of feasible initial states ($x \in \mathbb{S}_N$), $u \in \mathbb{U}$, where $\gamma : [0, \infty) \to [0, \infty)$ is continuous, $\gamma(t) > 0$ for all $t > 0$, and $\lim_{t\to\infty} \gamma(t) = \infty$.

C2 The terminal state weighting $F(x)$ in (5) is a Lipschitz continuous function and satisfies $F(0) = 0$,$F(x) \geq 0$ for all $x \in \mathbb{X}_f$, and the following property: there exists a terminal control law $\mathcal{K}_f : \mathbb{X}_f \to \mathbb{U}$ such that $F(f(x, \mathcal{K}_f(x))) - F(x) \leq -L(x, \mathcal{K}_f(x))$ for all $x \in \mathbb{X}_f$.

C3 The set $\mathbb{X}_f$ is positively invariant for the system $x_{i+1} = f(x_i, u_i)$ under $\mathcal{K}_f(x)$, that is, $f(x, \mathcal{K}_f(x)) \in \mathbb{X}_f$ for all $x \in \mathbb{X}_f$.

C4 The terminal control law $\mathcal{K}_f(x)$ satisfies the control constraints in $\mathbb{X}_f$, that is, $\mathcal{K}_f(x) \in \mathbb{U}$ for all $x \in \mathbb{X}_f$.

C5 The sets $\mathbb{U}$ and $\mathbb{X}_f$ contain the origin of their respective spaces.

Theorem 1. *Consider the system*

$$x_{i+1} = f(x_i, u_i) \quad \text{for } i \geq 0, \quad f(0,0) = 0,$$

controlled by the receding horizon algorithm (5)-(6) *and suppose that conditions* ***C1-C5*** *are satisfied. Then:*

1. *The set of feasible initial states $\mathbb{S}_N$ is positively invariant for the closed loop system.*
2. *The origin is globally attractive in $\mathbb{S}_N$ for the closed loop system.*
3. *If, in addition to* ***C1-C5****, $0 \in \mathrm{int}(\mathbb{S}_N)$ and the value function V_N^{OPT} is continuous on some neighborhood of the origin, then the origin is asymptotically stable in $\mathbb{S}_N$ for the closed loop system.*
4. *If, in addition to* ***C1-C5****, $0 \in \mathrm{int}(\mathbb{X}_f)$, $\mathbb{S}_N$ is compact, $\gamma(t) \geq at^\sigma$ in* ***C1****, $F(x) \leq b||x||^\sigma$ for all $x \in \mathbb{X}_f$ in* ***C2****, where $a > 0$, $b > 0$ and $\sigma > 0$ are some real constants, and the value function V_N^{OPT} is continuous in $\mathbb{S}_N$, then the origin is exponentially stable in $\mathbb{S}_N$ for the closed loop system.*

Next, we analyze conditions **C1-C5** for our particular problem (5)-(6) with linear system (1), non-convex polyhedral constraints (2) and value function (7):

C1: Holds by the choice of the quadratic per-stage weighting $L(x, u)$.

C2: Following the same arguments as in [18] for polyhedral constraints, condition holds by choosing appropriate control law and terminal state weighting from the solution of a Ricatti equation:

$$\begin{aligned} F(x_N) &= \tfrac{1}{2} x_N^T P x_N \\ \mathcal{K}_f(x) &= -Kx \end{aligned} \tag{8}$$

that satisfy

$$P = A^T P A + Q - K^T \bar{R} K \tag{9a}$$

$$K = \bar{R}^{-1} B^T P A \tag{9b}$$

$$\bar{R} = R + B^T P B \tag{9c}$$

C3-C4: Hold by definition when choosing as terminal region $\mathbb{X}_f$ the maximal closed-loop admissible set $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ as computed in section 3.

C5: Holds by assumption.

Once conditions **C1-C5** hold, Lyapunov stability of the MPC scheme is proved. However, in order to prove asymptotical stability, continuity of the value function on some neighbourhood of the origin must be assured. This property will hold as long as the origin is not on the boundary of any of the sets T_i, as we discuss next.

Considering the system equation (1) and all the possible combinations of constraints, problem (5) can be written with a single non-convex polyhedral constraint:

$$\begin{aligned} \mathcal{P}_{\mathbb{T}}(x): \quad & V_N^{OPT}(x) := \min \tfrac{1}{2}\mathbf{u}^T H \mathbf{u} + \mathbf{u}^T F x, \\ & \text{subject to:} \\ & (\mathbf{u}, x) \in \mathbb{T} = \textstyle\bigcup_{i=0}^{\gamma} T_i \end{aligned}$$

where $\mathbf{u}$ represents the future sequence of inputs.

This problem can be rewritten once again as:

$$\mathcal{P}_{\mathbb{T}}(x): \quad V_N^{OPT}(x) := \min \left\{ V_{iN}^{OPT}(x) \right\}$$

where:

$$\begin{aligned} & V_{iN}^{OPT}(x) := \min \tfrac{1}{2}\mathbf{u}^T H \mathbf{u} + \mathbf{u}^T F x, \\ & \text{subject to:} \\ & (\mathbf{u}, x) \in T_i \end{aligned} \tag{10}$$

Now, we take the T_i sets which contain the origin, and intersect them. As by assumption the origin is not in the boundary of any T_i, this intersection gives a convex set containing the origin in which the minimum of the value function is defined as the minimum of a finite set of functions: $V_N^{OPT}(x) = \min \left\{ V_{iN}^{OPT}(x) \right\}$.

It is well known that the optimal value function for each of the convex subproblems, $V_{iN}^{OPT}(x)$, is continuous [19]. Furthermore, the minimum of a finite number of continuous functions defined over the same convex set is also continuous on that set. Therefore, continuity of $V_N^{OPT}(x)$ in a neighbourhood of the origin holds.

As conditions for statement (3) of theorem 1 hold, the origin is asymptotically stable.

4.2. Set of feasible initial states

An important feature of any MPC scheme is the subset of the space state from which the closed-loop system is guaranteed to be stable, $\mathbb{S}_N$. As can be seen from (10), our particular problem can be formulated as finding the minimum cost given by the solutions of several convex optimization problems. Therefore, for a given x_0, if any of these problems has a feasible solution, the global problem will have too. So, we can define the set of feasible initial states for the global problem as the union of that same set for every subproblem, $\mathbb{S}_{Ni}$. To calculate these sets, te procedure from [16] can be used.

5. Examples

5.1. Example 1

Consider the 2 × 2 linear system:

$$G(s) = \begin{bmatrix} \frac{5}{s^2+5} & \frac{-1}{s+2} \\ 0 & \frac{2}{s+2} \end{bmatrix}$$

which is discretized with a sample time $T_s = 0.2$.

The system is subject to constraints in the states, $x(t) \in \mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2$ and in the inputs, $u(t) \in \mathbb{U} = \mathbb{U}_1 \cup \mathbb{U}_2$ where:

$$\mathbb{U}_1 := \left\{ \begin{array}{c} -2 \leq u_1 \leq 1 \\ -1 \leq u_2 \leq 2 \end{array} \right\} \quad \mathbb{U}_2 := \left\{ \begin{array}{c} 1 \leq u_1 \leq 3 \\ -2 \leq u_2 \leq 3 \end{array} \right\}$$

$$\mathbb{X}_1 := \left\{ \begin{array}{c} -6 \leq x_1 \leq 30 \\ -30 \leq x_2 \leq 30 \\ -30 \leq x_3 \leq 30 \end{array} \right\} \quad \mathbb{X}_2 := \left\{ \begin{array}{c} -30 \leq x_1 \leq -6 \\ -30 \leq x_2 \leq 30 \\ -21 \leq x_3 \leq 30 \end{array} \right\}$$

As can be seen in figures 1 and 2, both constraint sets are non-convex polyhedra.

We aim to find the maximal closed-loop admissible set $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ for the LQR controller (8) with weighting matrices $Q = C^T C$ and $R = 5I$:$\mathcal{K}_f(x) = -Kx$.

To do so, we follow algorithm 2 and start by calculating the convex hulls of $\mathbb{U}$ and $\mathbb{X}$ and finding the complement of these constraint sets in them. As shown in figures 3 and 4, these complements are formed by, respectively, two and one polyhedra ($\mathbb{U}_1^*$ and $\mathbb{U}_2^*$ and $\mathbb{X}_1^*$).

Then, we calculate the maximal input admissible set with constraints defined as the hulls, $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}})$, shown in figure 5.

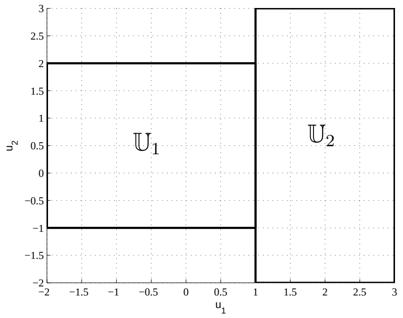

Figure 1: Input constraints $\mathbb{U}$.

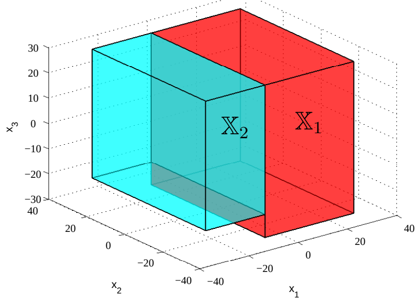

Figure 2: State constraints $\mathbb{X}$.

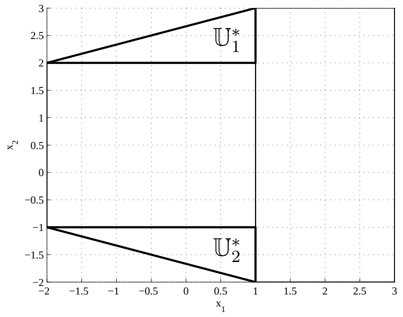

Figure 3: Complement of input constraints $\mathbb{U}^*$.

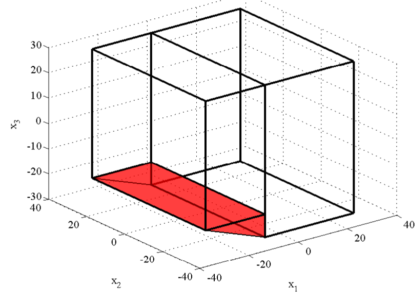

Figure 4: Complement of state constraints $\mathbb{X}^*$.

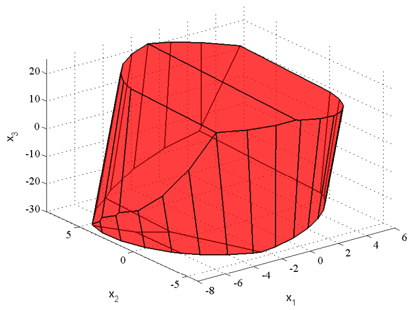

Figure 5: $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}})$.

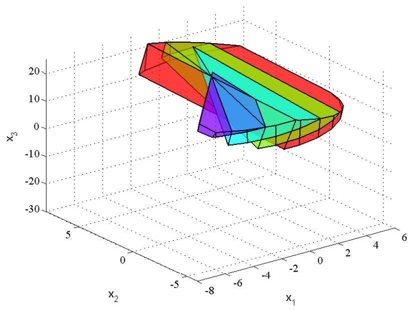

Figure 6: $\mathcal{T}_1 \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_1)$.

Next, we calculate the set of states $\mathcal{T}_1$ for which the controller $\mathcal{K}_f$ gives an input inside $\mathbb{U}_1^*$ (and therefore not inside $\mathbb{U}$). Furthermore, we also calculate the set of states that, in any number of steps, drives the future trajectory of the closed-loop system to the previous set, $\mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_1)$. The union of these sets is shown in figure 6. The set $\mathcal{T}_2 \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_2)$ is calculated the same way and shown in figure 7.

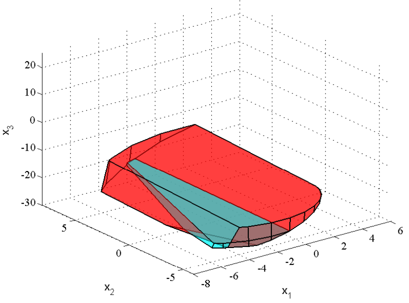

Figure 7: $\mathcal{T}_2 \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{T}_2)$.

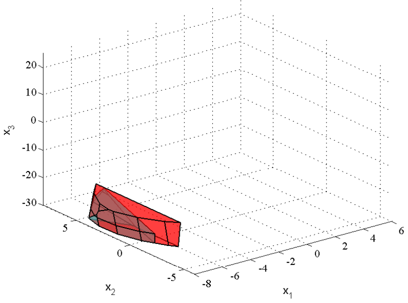

Figure 8: $\mathcal{D}_1 \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{D}_1)$.

Then, we obtain the set of states that drive the closed-loop system, in any number of steps, to $\mathbb{X}_1^*$ (which is outside of $\mathbb{X}$): $\mathcal{D}_1 \cup \mathcal{Q}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}}, \mathcal{D}_1)$. This set is represented in figure 8.

Lastly, we calculate the maximal closed loop invariant set for the non-convex constraints $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ by removing from $\mathcal{O}_\infty^{\mathcal{K}_f}(\bar{\mathbb{U}}, \bar{\mathbb{X}})$ all the previously calculated sets. As can be seen in figure 9, $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ is formed by the union of 8 convex polyhedra.

Figure 10 represents this same region calculated as $\mathbb{K}_\infty(\mathbb{U}, \Omega^{\mathcal{K}_f}, \Omega^{\mathcal{K}_f})$ using algorithm 1. It can be seen that the non-convex polyhedral region is the same in both cases although when algorithm 1 is used, the number of convex regions obtained is 278.

For this example, we compare the implementation in *MATLAB* with the *MPT toolbox* [20] of both algorithms in terms of number of LP solved and CPU time in seconds on a 2.66 GHz Pentium D with 1 GB of RAM. With algorithm 2, 1165 LPs are solved, with a total computation time of 1.9 seconds. On the other hand, with algorithm 1, 256725 LP problems are solved, with a total computation time of 140.3 seconds.

Furthermore, if $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ is calculated to be used as the terminal constraint set for a predictive control problem, it is crucial to have it defined

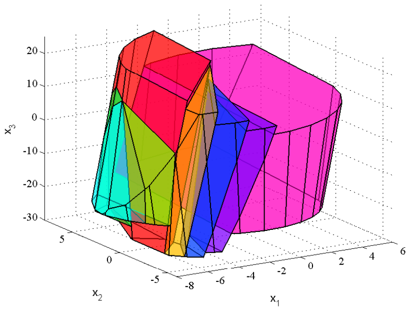

Figure 9: $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$ by algorithm 2 .

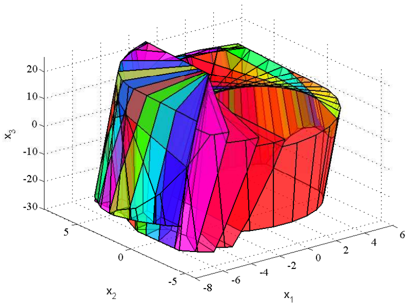

Figure 10: $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$ by algorithm 1.

as the union of a minimum number of convex regions, to keep the optimization problem as simple as possible. To do so, algorithms for the union of regions as the ones proposed in [17], should be used. If we perform one of these algorithms for the example after algorithm 2, we reduce the number of convex polyhedra to 7, but the total number of LP increases to 4278 and the computation time to 53.9 seconds. For algorithm 1, after the procedure for the union of regions the set is defined by 23 polyhedra, with a total cost of 484453 LP and 909 seconds.

Finally, if we use weighting matrices Q, R and P as calculated in (9), and a terminal region $\mathbb{X}_f = \mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U},\mathbb{X})$, an MPC scheme (5)-(7) will be asymptotically stable.

5.2. Example 2

Consider the simplified model of the open-loop unstable dynamics of an AFTI-F16 aircraft [21, 22]. The manipulated variables are the elevator and flaperon angles, while the attack and pitch angles are the measured outputs to be regulated. The model discretized with $T_s = 0.5s$ is defined by the following matrices:

$$A = \begin{bmatrix} 0.9993 & -3.008 & -0.1131 & -1.608 \\ -0.4703 \cdot 10^{-5} & 0.9862 & 0.0478 & 0.385 \cdot 10^{-5} \\ 0.3703 \cdot 10^{-5} & 2.083 & 1.009 & -0.4362 \cdot 10^{-5} \\ 0 & 0.05258 & 0.04979 & 1 \end{bmatrix} \quad B = \begin{bmatrix} -0.0804 & -0.6347 \\ -0.0291 & -0.0143 \\ -0.8679 & -0.0917 \\ -0.0216 & -0.0022 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

An MPC scheme for this system can be found as a case study in the *MPC toolbox* for *MATLAB* [23] with the following saturation in the actuators and output constraints:

$$\bar{\mathbb{U}} := \left\{ \begin{array}{c} -25 \leq u_1 \leq 25 \\ -25 \leq u_2 \leq 25 \end{array} \right\}$$
$$\bar{\mathbb{X}} := \left\{ \begin{array}{c} -0.5 \leq y_1 \leq 0.5 \\ -100 \leq y_2 \leq 100 \end{array} \right\}$$

We consider here a case in which, as well as satisfying the previous constraints, the measured outputs have to be kept outside the region:

$$\mathbb{X}^* := \left\{ \begin{array}{c} 0.1 \leq y_1 \leq 0.5 \\ 0 \leq y_2 \leq 100 \end{array} \right\}$$

Once again, we aim to find the maximal closed-loop admissible set $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ for an LQR controller (8) for such a system with weighting matrices $Q = C^T C$ and $R = 0.1I$.

In order to apply algorithm 1, the non-convex constraints sets have to be calculated:

$$\mathbb{X}_1 := \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 0.5 \\ -0.1 \\ 0 \\ 100 \end{bmatrix} \right\} \quad \mathbb{X}_2 := \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 0.1 \\ 0.5 \\ 100 \\ 100 \end{bmatrix} \right\}$$

If we apply algorithm 2, we obtain $\mathcal{O}_\infty^{\mathcal{K}_f}(\mathbb{U}, \mathbb{X})$ defined by 28 convex polyhedra. The algorithm needs to solve a number of 10539 LP, needing a total CPU time of 38.6 seconds. On the other hand, when applying algorithm 1, the region is defined by 773 polyhedra, the number of LP solved is 2619070, and the total CPU time is 1866 seconds.

When the procedure for the union of regions is applied after algorithm 2 the number of convex polyhedra is reduced to 15 after solving 61456 LP during a total time of 845 seconds. If the same procedure is applied after algorithm 1, the final region is defined by 17 polyhedra, after solving 6769237 LP during 20717 seconds.

Once again, the use of the maximal closed loop admissible set as terminal set, allows to prove stability of an MPC scheme for the proposed problem.

6. Conclusions

In this work, we have proposed an efficient algorithm for the computation of the maximal closed-loop admissible set for linear systems with non-convex polyhedral constraints based on removing subsets of the maximal closed-loop invariant set for the convex hull of the original constraints. We have compared this new algorithm with a previous generic one, showing that, in a worst-case scenario, the set obtained by our procedure is defined by a lower number of regions than when it is obtained by the existing algorithm.

Next, we have proposed the use of this set as a terminal constraint for the model predictive control of linear systems with non-convex polyhedral constraints, showing that it allows to establish the well known *a priori* stability conditions.

Finally, two examples for which the proposed algorithm outperforms the previous generic algorithm have been presented.

Acknowledgment

This work was partially funded by projects DPI2008-02133 and DPI2008-06731/DPI Ministerio de Ciencia e Innovación - Spanish Government.

References

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

[2] A. B. Kurzhanski, Dynamic optimization for nonlinear target control synthesis, Nonlinear Control Systems 2004 (2005) 21.

[3] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research 5 (1) (1986) 90.

[4] J. K. Kuchar, L. C. Yang, A review of conflict detection and resolution modeling methods, IEEE Transactions on Intelligent Transportation Systems 1 (4) (2000) 179–189.

[5] K. P. Fruzzetti, A. Palazoglu, K. A. McDonald, Nonlinear model predictive control using hammerstein models, Journal of Process Control 7 (1) (1997) 31–41.

[6] R. S. Patwardhan, S. Lakshminarayanan, S. L. Shah, Constrained nonlinear MPC using Hammerstein and Wiener models: PLS framework, AICHE Journal 44 (7) (1998) 1611–1622.

[7] J. J. E. Slotine, W. Li, Applied nonlinear control, Prentice-Hall Englewood Cliffs, NJ, 1991.

[8] G. Oriolo, A. D. Luca, M. Vendittelli, WMR control via dynamic feedback linearization: design, implementation, and experimental validation, IEEE Transactions on Control Systems Technology 10 (6) (2002) 835–852.

[9] G. J. Pappas, J. Lygeros, D. N. Godbole, Stabilization and tracking of feedback linearizable systems under input constraints, in: Proceedings of the IEEE Conference on Decision and Control, Citeseer, 1995, pp. 596–601.

[10] W. Heemels, B. D. Schutter, A. Bemporad, Equivalence of hybrid dynamical models, Automatica 37 (7) (2001) 1085–1091.

[11] F. Borrelli, Constrained optimal control of linear and hybrid systems, Springer Verlag, 2003.

[12] F. Borrelli, P. Falcone, J. Pekar, G. Stewart, Reference governor for constrained piecewise affine systems, Journal of Process Control 19 (8) (2009) 1229–1237.

[13] M. Lazar, W. Heemels, S. Weiland, A. Bemporad, Stabilizing model predictive control of hybrid systems, IEEE Transactions on Automatic Control 51 (11) (2006) 1813.

[14] S. V. Rakovic, P. Grieder, M. Kvasnica, D. Q. Mayne, M. Morari, Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances, in: IEEE Conference on Decision and Control, 2004, pp. 1418–1423.

[15] E. G. Gilbert, K. T. Tan, Linear systems with state and control constraints: the theory andapplication of maximal output admissible sets, IEEE Transactions on Automatic Control 36 (9) (1991) 1008–1020.

[16] E. C. Kerrigan, Robust constraint satisfaction: Invariant sets and predictive control, Ph.D. thesis, University of Cambridge, UK (2000).

[17] T. Geyer, F. D. Torrisi, M. Morari, Optimal complexity reduction of polyhedral piecewise affine systems, Automatica 44 (7) (2008) 1728–1740.

[18] G. C. Goodwin, M. M. Seron, Constrained Control and Estimation: An Optimisation Approach, Springer, 2005.

[19] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (1) (2002) 3–20.

[20] M. Kvasnica, P. Grieder, M. Baotic, M. Morari, Multi-parametric toolbox (MPT), Hybrid Systems: Computation and Control (2004) 121–124.

[21] A. Bemporad, A. Casavola, E. Mosca, Nonlinear control of constrained linear systems via predictive reference management, IEEE Transactions on Automatic Control 42 (3) (1997) 340–349.

[22] P. K. M. A. G. Stein, Design of feedback control systems for unstable plants with saturating actuators, in: IFAC Symp. Nonlinear Control Syst. Design, 1989, p. 302307.

[23] M. Morari, N. L. Ricker, Model predictive control toolbox, Matlab User's Guide, Version 1.