



**HAL**  
open science

# Predictive and Symbolic Control: Performance and Safety for Non-linear Systems

Zakeye Azaki, Antoine Girard, Sorin Olaru

► **To cite this version:**

Zakeye Azaki, Antoine Girard, Sorin Olaru. Predictive and Symbolic Control: Performance and Safety for Non-linear Systems. CAO 2022 - 18th IFAC Workshop on Control Applications of Optimization, Jul 2022, Gif-sur-Yvette, France. pp.290-295, 10.1016/j.ifacol.2022.09.039 . hal-03708834

**HAL Id: hal-03708834**

**<https://hal.science/hal-03708834>**

Submitted on 29 Jun 2022

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

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

# Predictive and Symbolic Control: Performance and Safety for Non-linear Systems<sup>\*</sup>

Zakeye AZAKI<sup>\*</sup>, Antoine GIRARD<sup>\*\*</sup>, Sorin OLARU<sup>\*\*</sup>

<sup>\*</sup> Université Grenoble Alpes, CNRS, Grenoble-INP  
GIPSA-lab, F-38000, Grenoble, France  
zakeye.azaki@gipsa-lab.grenoble-inp.fr

<sup>\*\*</sup> Université Paris-Saclay, CNRS, CentraleSupélec,  
Laboratoire des signaux et systèmes, 91190 Gif-sur-Yvette, France  
antoine.girard@l2s.centralesupelec.fr, sorin.olaru@l2s.centralesupelec.fr

---

**Abstract:** Although Model Predictive Control (MPC) has proven its efficiency in the process operation and it is known for its high performance, it also suffers from design limitation. One of the fundamental issues for such an optimization-based technique is the difficulty to guarantee recursive feasibility in the absence of terminal constraints, or alternatively, the complexity of design when a certain basin of attraction or a controlled invariant set needs to be certified for the closed loop in the most general nonlinear setting. Based on symbolic control techniques, this paper proposes a simple and guaranteed solution for such problems. As a main result, a *Symbolically guided Model Predictive Control* scheme is developed. This controller is an improved version of the generic MPC approach such that the recursive feasibility is guaranteed through appending time-varying terminal constraints, carefully designed using the symbolic control approach, to the optimization problem of the original MPC formulation.

**Keywords:** Model predictive control; Symbolic control; Recursive feasibility; Controlled invariant set; Terminal constraints; Non-linear systems

---

## 1. INTRODUCTION

Model Predictive Control (MPC) is well established in the control engineering studies as a model-based methodology able to synthesize the control action based on the online solution of an open loop optimal-control problem subjected to physical constraints (Mayne (2014); Maciejowski (2002)). These constraints are defining safety regions in the state space or restrictions on control variables modelling the saturation of actuators or limitation in the available control energy. MPC becomes, nowadays, one of the most advanced control technique capable to control multivariate plants and it has proven its efficiency in industrial processes, mainly for systems with slow dynamics, as chemical processes (Qin and Badgwell (2003)).

Aside the practical computational problems related to the implementation of such an optimization-based control, one of the most important theoretical problems in MPC is the recursive feasibility, which deals with the fundamental guarantee that the nominal MPC closed-loop solutions have to remain within a safe set for an indefinite time. Unfortunately, although it is absolutely recommended from a theoretical point of view (Mayne and Falugi (2019)), it is not always easy to construct an MPC controller which has an a priori guarantee of recursive feasibility while maintaining the efficiency of MPC.

There are few approaches in the literature that suggest a solution for this problem (Mayne et al. (2000)). In Chmielewski and Manousiouthakis (1996), the approach focuses on solving the problem by computing a large enough prediction horizon  $N$  of the optimization problem while Sznaier and Damborg (1987)

investigates the relationship between the sub-optimality and the stability. More recent studies (Grüne and Pannek (2017)) have shown that choosing the length of the prediction horizon to ensure stability is not a trivial task. Moreover, enlarging the prediction horizon increases the complexity of the optimization problem for MPC and requires more computational power. Other suggested approaches (Löfberg (2012)) append the optimization problem by a maximal controlled invariant terminal constraint set. Unfortunately, the construction of this set is difficult and non-guaranteed to preserve the controlled invariance properties in a general nonlinear framework.

The approach proposed in the present paper is to build on the recent advances on symbolic control design (see e.g. Tabuada (2009); Belta et al. (2017)). Symbolic control is an approach to automatically synthesize certifiable controllers, for general nonlinear systems, that handle state and input constraints and various specifications such as safety, reachability or attractivity (Girard and Eqtami (2021)). However, to obtain good performance with symbolic control, one needs very fine partitions which are very costly from the computational viewpoint. In this paper, we develop a novel MPC scheme that we call *Symbolically guided Model Predictive Control* (SgMPC) where time-varying terminal constraints are appended to classical MPC formulation. The time-varying constraints are carefully designed using symbolic control techniques in such a way that recursive feasibility of the SgMPC controller is guaranteed. Time varying terminal constraints have already been considered in MPC, e.g in Limon et al. (2005); Kögel and Findeisen (2013); Lazar and Tetteroo (2018). In these works, the sequence of terminal constraint is fixed a priori and is chosen to be either constant after some time or periodic. In comparison,

---

<sup>\*</sup> This project has received funding from the H2020-EU.1.1. research and innovation programme(s) – ERC-2016-COG under grant agreement No 725144.

the sequence of terminal constraints in our approach is determined dynamically at run-time using a designed feedback mechanism. Finally, let us remark that while most of MPC approaches addresses both safety and stability requirements, in this paper we only deal with safety constraints and leave stability properties for future research.

**Paper Organization:** This paper is organized as follows. In Section 2, we illustrate the main and general implemented principle to enforce MPC recursive feasibility by time-varying terminal constraints. The design of suitable terminal constraints using symbolic control techniques is discussed in Section 3. Section 4 presents possible refinements improving the approach presented in Section 3. Finally, Section 5 presents a series of numerical experiments to illustrate the performance and to validate the efficiency of the SgMPC strategy.

**Notations:**  $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{R}$  and  $\mathbb{R}_+$  denote the sets of natural, integer, real and non-negative real numbers, respectively. Given a set  $X$  we denote  $2^X$  to the set of subsets of  $X$ . The empty set is denoted by  $\emptyset$ . For  $x \in \mathbb{R}^n$ , we denote by  $x_i$  its  $i$ -th coordinate,  $i = 1, \dots, n$ . We define a partial order on  $\mathbb{R}^n$  as follows: for  $x, x' \in \mathbb{R}^n$   $x \preceq x'$  if and only if  $x_i \leq x'_i$ , for all  $i = 1, \dots, n$ . Similarly,  $x \prec x'$  if and only if  $x_i < x'_i$ , for all  $i = 1, \dots, n$ . We denote  $[x, x'] = \{y \in \mathbb{R}^n \mid x \preceq y \preceq x'\}$  and  $[x, x') = \{y \in \mathbb{R}^n \mid x \preceq y \prec x'\}$ . Then  $[x, x'] + [a, b] = [x + a, x' + b]$ .

## 2. TIME-VARYING TERMINAL CONSTRAINTS FOR RECURSIVE FEASIBILITY IN MPC

In this section, we revisit recursive feasibility in MPC and describe the mechanisms to enforce them using time-varying terminal constraints. We first present the problem under consideration. Then, we establish conditions on the design of time-varying terminal constraints, with the goal to provide theoretical guarantees of recursive feasibility.

### 2.1 System dynamics

Constrained control design is investigated in the current work for discrete-time nonlinear systems, assuming that all the system's states are measurable. The model of the system's dynamics is considered in the following form:

$$x_{t+1} = f(x_t, u_t) \quad (1)$$

with state  $x_t \in \mathbb{R}^d$  and control input  $u_t \in \mathbb{R}^m$ , for time  $t \in \mathbb{N}$ .  $f : \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}^d$  is a known and generally nonlinear map, such that  $f = (f_1, f_2, \dots, f_d)^T$ .

We employ the notation  $\mathbf{u} = (u_0, u_1, \dots)$ , with  $u_t \in \mathbb{R}^m$ , for a potentially infinite control sequence and the notation  $\phi(t, x_0, \mathbf{u})$  for the state reached at sampling instant  $t$  from the initial state  $x_0$  and under the control sequence  $\mathbf{u}$ . Given  $N \geq 1$  as prediction horizon, a control sequence computed at sampling instant  $t$  will be denoted as  $\mathbf{u}_t^N = (u_{0|t}, \dots, u_{N-1|t}) \in \mathbb{R}^{m \times N}$ . Given state  $x_t$  at sampling instant  $t$ , the associated predicted trajectory at  $t \in \mathbb{N}$  is denoted  $\mathbf{x}_t^N = (x_{0|t}, x_{1|t}, \dots, x_{N|t})$  where  $x_{k|t} = \phi(k, x_t, \mathbf{u}_t^N)$ , for  $k = 0, \dots, N$ .

### 2.2 Model predictive control

Considering the control system (1), the core of an MPC scheme relies on the resolution of an optimal control problem, at each

sampling instant  $t \in \mathbb{N}$ . This finite-horizon optimization, over the finite prediction horizon  $N$ , takes the following form:

$$\begin{aligned} \arg \min_{\mathbf{u}_t^N} J_N(x_t, \mathbf{u}_t^N) &= \sum_{k=0}^{N-1} \ell(x_{k|t}, u_{k|t}) + L(x_{N|t}) \quad (2) \\ \text{subject to } \begin{cases} x_{k|t} = \phi(k, x_t, \mathbf{u}_t^N), & k = 0, \dots, N; \\ x_{k|t} \in \mathbb{X}, u_{k|t} \in \mathbb{U}, & k = 0, \dots, N-1; \\ x_{N|t} \in \mathbb{X}_t. \end{cases} \quad (3) \end{aligned}$$

The control constraint set is  $\mathbb{U} \subseteq \mathbb{R}^m$ , while the state constraint set  $\mathbb{X} \subseteq \mathbb{R}^d$ . Let us emphasize that in (2)-(3), the terminal constraint set is time-varying and is described by the set  $\mathbb{X}_t \subseteq \mathbb{X}$ . The function  $\ell : \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}$  is called stage cost or running cost, the function  $L : \mathbb{X} \rightarrow \mathbb{R}$  is called terminal cost.

**Definition 2.1.** (*Feasible control sequence*) Given a state  $x_t \in \mathbb{X}$  and a terminal constraint set  $\mathbb{X}_t \subseteq \mathbb{X}$ , a finite control sequence  $\mathbf{u}_t^N = (u_{0|t}, \dots, u_{N-1|t})$ , with its associated trajectory  $\mathbf{x}_t^N = (x_{0|t}, \dots, x_{N|t})$ , is said to be feasible if:

$$\begin{aligned} x_{k|t} \in \mathbb{X}, u_{k|t} \in \mathbb{U}, \quad \forall k = 0, \dots, N-1; \\ x_{N|t} \in \mathbb{X}_t. \end{aligned}$$

We denote by  $U_f(x_t, \mathbb{X}_t)$  the set of all feasible control sequences  $\mathbf{u}_t^N$  for state  $x_t \in \mathbb{X}$  and terminal constraint set  $\mathbb{X}_t \subseteq \mathbb{X}$ .

We assume in the following that  $\mathbb{U}$ ,  $\mathbb{X}$  and  $\mathbb{X}_t$  are compact sets and that  $f$ ,  $\ell$  and  $L$  are continuous functions. Then, whenever  $U_f(x_t, \mathbb{X}_t) \neq \emptyset$ , the optimization problem (2)-(3) admits at least one minimizer, which is denoted by the optimal control sequence

$$\mathbf{u}_t^{N*}(x_t, \mathbb{X}_t) = (u_{0|t}^*, \dots, u_{N-1|t}^*).$$

Then, the MPC control action fed to the system can be described as a feedback law

$$\mu_N(x_t, \mathbb{X}_t) = u_{0|t}^*.$$

The resolution of the optimization problem (2)-(3) is repeated online, at each sampling instant leading to the following closed-loop behavior

$$x_{t+1} = f(x_t, \mu_N(x_t, \mathbb{X}_t)). \quad (4)$$

A fundamental problem in MPC is that of *recursive feasibility*. The optimization problem (2)-(3) is recursively feasible if it can be shown that  $U_f(x_t, \mathbb{X}_t) \neq \emptyset$  at each sampling instant, resulting in a well-posed closed loop-system (4) satisfying both state and input constraints at all time  $t \in \mathbb{N}$ .

A classical way (see e.g. Mayne et al. (2000)) to enforce recursive feasibility is to use a controlled invariant set  $\mathbb{X}_I \subseteq \mathbb{X}$  as terminal constraint set in (2)-(3), i.e.,  $\mathbb{X}_t = \mathbb{X}_I$ , for all  $t \in \mathbb{N}$ . The larger  $\mathbb{X}_I$  (i.e. the closer to the maximal controlled invariant set) the better since it provides more freedom in (2)-(3) to optimize the performance cost. However, for nonlinear systems and possibly non-convex state and input constraints, large controlled invariant sets may be very complex and typically non-convex, rendering the optimization problem (2)-(3) difficult to solve online.

In the following, we propose an alternative approach that takes advantage of the fact that time-varying terminal constraint sets can be used in the MPC design in (2)-(3). Essentially, we use a simple (e.g. convex) subset  $\mathbb{X}_t \subseteq \mathbb{X}_I$  as terminal constraint. However, since  $\mathbb{X}_t$  is generally not control invariant itself, it needs to be adapted at each time step in order to ensure recursive feasibility. In the following section, we present an

approach to design a suitable sequence of terminal constraint sets ensuring recursive feasibility.

### 2.3 Construction principle for time-varying terminal constraints

Let us assume the existence of a possibly complex (i.e. non-convex) controlled invariant set  $\mathbb{X}_I \subseteq \mathbb{X}$ . Furthermore, assume the existence of a set-valued map  $T : \mathbb{X}_I \rightarrow 2^{\mathbb{X}_I}$  and of a controller  $C : \mathbb{X}_I \rightarrow \mathbb{U}$  such that

$$\forall x \in \mathbb{X}_I, f(x, C(x)) \in T(x). \quad (5)$$

Let us remark that a trivial choice for  $T$  and  $C$  is given by  $T(x) = \mathbb{X}_I$  with  $C$  being a controller guaranteeing the closed-loop invariance properties of  $\mathbb{X}_I$ . However, in the following, we will be interested in maps  $T$  such that  $T(x)$  are simple, typically convex, sets.

We now describe the construction of the time-varying terminal constraint sets for the optimization problem (2)-(3):

- At sampling instant  $t = 0$ , with initial state  $x_0 \in \mathbb{X}_I$ , let

$$\bar{\mathbf{u}}_0^N = (C(\bar{x}_{0|0}), \dots, C(\bar{x}_{N-1|0})) \quad (6)$$

where  $\bar{x}_{0|0} = x_0$  and

$$\bar{x}_{k+1|0} = f(\bar{x}_{k|0}, C(\bar{x}_{k|0})), \quad k = 0, \dots, N-1.$$

Let the terminal constraint set at time  $t = 0$  be defined as:

$$\mathbb{X}_0 = T(\bar{x}_{N-1|0}). \quad (7)$$

- For  $t \in \mathbb{N}$ , consider the optimal control sequence at sampling time  $t$ ,

$$\mathbf{u}_t^{N*}(x_t, \mathbb{X}_t) = (u_{0|t}^*, \dots, u_{N-1|t}^*)$$

and the associated predicted trajectory

$$\mathbf{x}_t^{N*} = (\bar{x}_{0|t}^*, \dots, \bar{x}_{N|t}^*).$$

Let

$$\bar{\mathbf{u}}_{t+1}^N = (u_{1|t}^*, \dots, u_{N-1|t}^*, C(x_{N|t}^*)) \quad (8)$$

and let the terminal constraint set at time  $t+1$  be defined as:

$$\mathbb{X}_{t+1} = T(x_{N|t}^*). \quad (9)$$

For the sequence of terminal constraint sets constructed above, we can show the following result:

**Theorem 2.1.** *Let us assume that there exist a set-valued map  $T : \mathbb{X}_I \rightarrow 2^{\mathbb{X}_I}$  and a controller  $C : \mathbb{X}_I \rightarrow \mathbb{U}$  satisfying the conditions presented in (5) with  $\mathbb{X}_I \subseteq \mathbb{X}$ . Then, for any initial state  $x_0 \in \mathbb{X}_I$ , with the time-varying terminal constraint sets  $\mathbb{X}_t$  selected according to (7) and (9):*

- MPC optimization problem (2)-(3) is recursively feasible;
- the control sequences  $\bar{\mathbf{u}}_t^N$  constructed based on (6) and (8) are feasible at all sampling instants.

*Proof.* Let us consider an initial state  $x_0 \in \mathbb{X}_I$ . Let the control sequence  $\bar{\mathbf{u}}_0^N$  be given by (6) with its associated predicted trajectory  $\bar{\mathbf{x}}_0^N = (\bar{x}_{0|0}, \dots, \bar{x}_{N|0})$ . Then, according to (5),

$$\bar{x}_{k+1|0} \in T(\bar{x}_{k|0}) \subseteq \mathbb{X}_I \subseteq \mathbb{X}, \quad \forall k = 0, \dots, N-1.$$

Moreover, the terminal constraint given by (7) is satisfied:  $\bar{x}_{N|0} \in T(\bar{x}_{N-1|0}) = \mathbb{X}_0$ . Therefore, it follows that the control sequence  $\bar{\mathbf{u}}_0^N \in U_f(x_0, \mathbb{X}_0)$ .

Now, let  $t \in \mathbb{N}$ , let us consider the control sequence  $\bar{\mathbf{u}}_{t+1}^N$  given by (8) with its associated predicted trajectory  $\bar{\mathbf{x}}_{t+1}^N = (\bar{x}_{0|t+1}, \dots, \bar{x}_{N|t+1})$ . Obviously,  $\bar{x}_{k|t+1} = x_{k+1|t}^*$ , for all  $k =$

$0, \dots, N-1$ . Therefore, from (2)-(3),  $\bar{x}_{k|t+1} \in \mathbb{X}$  and  $\bar{x}_{N-1|t+1} \in \mathbb{X}_t \subseteq \mathbb{X}_I$ . Then, based on (5) it holds

$$\bar{x}_{N|t+1} = f(\bar{x}_{N-1|t+1}, C(\bar{x}_{N-1|t+1})) \in T(\bar{x}_{N-1|t+1}).$$

Remarking that from (9),

$$T(\bar{x}_{N-1|t+1}) = T(x_{N|t}^*) = \mathbb{X}_{t+1},$$

we obtain that  $\bar{\mathbf{u}}_{t+1}^N \in U_f(x_{t+1}, \mathbb{X}_{t+1})$ .

From above, we see that the set of feasible control sequences is non-empty at any time. Hence, recursive feasibility is guaranteed for the MPC optimization problem.  $\square$

In this section, we proposed a framework for MPC with time-varying terminal constraints and derived design principles ensuring recursive feasibility. The main problem considered in the following sections is that of constructing the map  $T : \mathbb{X}_I \rightarrow 2^{\mathbb{X}_I}$  and controller  $C : \mathbb{X}_I \rightarrow \mathbb{U}$  satisfying (5). This is done using symbolic control techniques.

## 3. SYNTHESIS OF TERMINAL CONSTRAINT SETS BASED ON SYMBOLIC CONTROL

Symbolic control (see e.g. Tabuada (2009); Belta et al. (2017)) is a computational technique for synthesizing controllers for nonlinear systems with state and input constraints. The approach is based on the use of *symbolic models*, which are finite state / finite input dynamical systems over-approximating the dynamics of the system we want to control. Compared to MPC, most of the computation is carried offline. Symbolic control is particularly helpful for the computation of controlled invariant. In this section, we show how one can use symbolic models to design the map  $T$  and controller  $C$  satisfying (5).

### 3.1 Computation of symbolic models

We briefly recall an approach to compute a symbolic model of (1) similar to that presented in Reissig et al. (2016).

*Discretization of states and inputs:* Since  $\mathbb{X}$  is assumed to be compact there exists an interval  $R_{\mathbb{X}} = [\underline{x}, \bar{x}]$  such that  $\mathbb{X} \subseteq R_{\mathbb{X}}$ . We first start by considering a partition  $S_0$  of  $R_{\mathbb{X}}$  such that each element  $s \in S_0$  is an interval of the form  $s = [\underline{s}, \bar{s}]$ . Then, we obtain a partition  $S$  of  $\mathbb{R}^d$  defined as follows  $S = S_0 \cup \{\mathbb{R}^d \setminus R_{\mathbb{X}}\}$ . The elements of  $S$  are referred to as *symbolic states*. We also define the quantizer  $\sigma : \mathbb{R}^d \rightarrow S$  defined by

$$\forall x \in \mathbb{R}^d, \sigma(x) = s \iff x \in s.$$

The symbolic state constraint set  $\mathbb{S} \subseteq S_0$  is defined as follows

$$\mathbb{S} = \{s \in S_0 \mid s \subseteq \mathbb{X}\}. \quad (10)$$

We also consider a finite subset of inputs  $U \subseteq \mathbb{U}$ . The elements of  $U$  are referred to as *symbolic inputs*.

*Computation of the symbolic transition relation:* The symbolic model is a dynamical system with finite sets of states and inputs  $S$  and  $U$ , respectively. For an input  $u \in U$  and an interval  $s \in S_0$ , let us denote by  $\hat{f}(s, u)$  an interval such that  $f(s, u) \subseteq \hat{f}(s, u)$ . Such interval over-approximations can be computed using interval reachability analysis (see e.g. the various techniques described in Meyer et al. (2021)).

The dynamics of the symbolic model is given by the *transition relation*, which is a set-valued map  $F : S \times U \rightarrow 2^S$  defined for all  $s \in S_0, u \in U$  by

$$F(s, u) = \sigma(\hat{f}(s, u)), \quad (11)$$

and for  $s = \mathbb{R}^d \setminus R_{\mathbb{X}}, u \in U$ , by  $F(s, u) = S$ .

Then, the symbolic model is the dynamical system given by the difference inclusion

$$s_{t+1} \in F(s_t, u_t) \quad (12)$$

with  $s_t \in S, u_t \in U$ , for time  $t \in \mathbb{N}$ .

### 3.2 Symbolic design of terminal constraint sets

Let  $\mathbb{S}_I$  be the maximal robust controlled invariant set of (12) contained in the symbolic state constraint set  $\mathbb{S}$ , i.e. the largest subset of  $\mathbb{S}$  such that

$$\forall s \in \mathbb{S}_I, \exists u \in U, \text{ such that } F(s, u) \subseteq \mathbb{S}_I.$$

Since  $S$  and  $U$  are finite sets, there exist algorithms to compute  $\mathbb{S}_I$  in a finite number of iterations (see e.g. Tabuada (2009); Girard and Eqtami (2021)). Then, let us consider a symbolic controller  $K : \mathbb{S}_I \rightarrow U$  guaranteeing the closed-loop robust invariance property of  $\mathbb{S}_I$ :

$$\forall s \in \mathbb{S}_I, F(s, K(s)) \subseteq \mathbb{S}_I. \quad (13)$$

Let us now define the set

$$\mathbb{X}_I = \sigma^{-1}(\mathbb{S}_I), \quad (14)$$

and for all  $x \in \mathbb{X}_I$ ,

$$\begin{aligned} T(x) &= \hat{f}(\sigma(x), K(\sigma(x))), \\ C(x) &= K(\sigma(x)). \end{aligned} \quad (15)$$

Then, the following result holds:

**Theorem 3.1.** *Let  $\mathbb{X}_I, T$  and  $C$  be given by (14) and (15). Then,  $\mathbb{X}_I$  is a controlled invariant set included in  $\mathbb{X}$ . Moreover, for all  $x \in \mathbb{X}_I, T(x) \subseteq \mathbb{X}_I, C(x) \in U$  and (5) holds.*

*Proof.* Let  $x \in \mathbb{X}_I$ , then  $C(x) = K(\sigma(x)) \in U \subseteq \mathbb{U}$  by construction. From (15), (11), (13) and (14)

$$\begin{aligned} T(x) &= \hat{f}(\sigma(x), K(\sigma(x))) \\ &\subseteq \sigma^{-1}(F(\sigma(x), K(\sigma(x)))) \\ &\subseteq \sigma^{-1}(\mathbb{S}_I) = \mathbb{X}_I. \end{aligned}$$

Moreover, by (15)

$$\begin{aligned} f(x, C(x)) &= f(x, K(\sigma(x))) \in f(\sigma(x), K(\sigma(x))) \\ f(\sigma(x), K(\sigma(x))) &\subseteq \hat{f}(\sigma(x), K(\sigma(x))) = T(x). \end{aligned}$$

Thus, (5) holds. From above, we get that for all  $x \in \mathbb{X}_I$

$$f(x, C(x)) \in T(x) \subseteq \mathbb{X}_I.$$

Hence,  $\mathbb{X}_I$  is a controlled invariant set. Finally, since  $\mathbb{S}_I \subseteq \mathbb{S}$  and by (10), we get that

$$\mathbb{X}_I = \sigma^{-1}(\mathbb{S}_I) \subseteq \sigma^{-1}(\mathbb{S}) \subseteq \mathbb{X}. \quad \square$$

Hence, it follows from Theorem 3.1, that  $T$  and  $C$  given by (15) and synthesized using symbolic control techniques are suitable for the design of time-varying terminal constraints in the MPC problem (2)-(3).

Let us remark that one could also use the control invariant  $\mathbb{X}_I$  in (14) as terminal constraint to ensure recursive feasibility

of (2)-(3). However,  $\mathbb{X}_I$  is defined as a union of (possibly many) intervals. Then, this introduces some combinatorial features to (2)-(3), which makes it very hard to solve online. In comparison, the time-varying constraints given by the map  $T$  in (15) are simple intervals and can be easily handled by off-the-shelf optimization software.

Finally, it is important to remark that all computations described in this section can be carried out offline and that  $T$  and  $C$  can be easily stored as lookup tables. In the following, we refer to this approach by *Symbolically guided Model Predictive Control* (SgMPC).

## 4. FURTHER IMPROVEMENTS ON SgMPC

In this section, we report two improvements of the technique described above. The first one allows us to design larger terminal constraint sets giving more freedom to optimize the behavior. The second one makes it possible to use the symbolic model to optimize the controller  $C$  in view of the initialization of the MPC problem (2)-(3).

### 4.1 Enlarged terminal constraint sets

We first show how to redesign the map  $T$  given by (15) in order to provide more degrees of freedom for optimization. Consider  $\mathbb{X}_I$  as in (14) and  $C$  as in (15). For all  $s \in \mathbb{S}_I$ , let

$$\rho(s) = \max \left\{ \rho \geq 0 \mid \hat{f}(s, K(s)) + \rho[-\mathbf{1}, \mathbf{1}] \subseteq \mathbb{X}_I \right\}.$$

Then, let us define the map, given for all  $x \in \mathbb{X}_I$  by

$$T'(x) = T(x) + \rho(\sigma(x))[-\mathbf{1}, \mathbf{1}]. \quad (16)$$

An illustration of the map  $T'$  can be seen in Fig.1. Let us remark that by design, we have for all  $x \in \mathbb{X}_I$

$$T(x) \subseteq T'(x) \subseteq \mathbb{X}_I.$$

The above inclusions, together with Theorem 3.1 lead to the following result:

**Corollary 4.1.** *Let  $\mathbb{X}_I, T'$  and  $C$  be given by (14), (16) and (15). Then, for all  $x \in \mathbb{X}_I, T'(x) \subseteq \mathbb{X}_I$  and (5) holds with  $T'$  and  $C$ .*

Hence, using  $T'$  instead of  $T$  makes it possible to use more relaxed terminal constraints, also under the form of intervals, while still guaranteeing recursive feasibility. Moreover, similar to  $T$ , the map  $T'$  can be computed offline and stored in a lookup table.

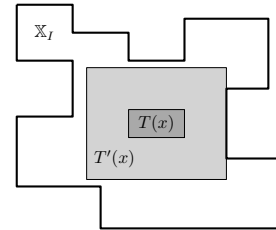


Figure 1. Illustration of the enlarged terminal constraint set construction

### 4.2 Optimized symbolic controller

There is a lot of freedom in the design of the symbolic controller  $K$ . Indeed, there are often many controllers satisfying (13), i.e. enforcing the closed-loop robust invariance of  $\mathbb{S}_I$ . So it is natural to consider optimizing over the set of possible symbolic controllers.

In particular, symbolic control makes it possible to synthesize controllers rendering  $\mathbb{S}_I$  robustly invariant, while minimizing some receding horizon cost function such as the following:

$$\sum_{k=0}^{N-1} \hat{\ell}(s_{k|t}, u_{k|t}) + \hat{L}(s_{N|t})$$

where  $\hat{\ell}(s, u) = \max_{x \in s} \ell(x, u)$  and  $\hat{L}(s) = \max_{x \in s} L(x)$ .

This optimization problem is solved in Meyer et al. (2015). Let us remark that this problem can be seen as the symbolic counterpart of (2)-(3). One major difference, though, is that all the computations can be done offline for our symbolic model (12). Then, we can use the optimized controller  $K$  to design the state dependent terminal constraint and the initial control sequence guess, that are utilized in SgMPC. All the preceding results are still valid and the performance of SgMPC is improved when it is guided by an optimized symbolic controller (OSC).

## 5. NUMERICAL EXAMPLE

Let us consider a discrete-time non-linear system with the state vector  $x \in \mathbb{R}^3$  and control input vector  $u \in \mathbb{R}^2$ , such that  $x = (x_1, x_2, x_3)^T$  and  $u = (u_1, u_2)^T$  and the dynamics governed by the following equations:

$$\begin{aligned} x_1(t+1) &= x_1(t) + u_1(t) \cos(x_3(t)) \\ x_2(t+1) &= x_2(t) + u_1(t) \sin(x_3(t)) \\ x_3(t+1) &= x_3(t) + u_2(t) \pmod{2\pi} \end{aligned} \quad (17)$$

The trajectories represent the behaviour of a mobile cart with  $(x_1, x_2)$  representing the 2D Cartesian coordinates in meter.  $x_3$  represents the angular orientation of the velocity vector around the  $x$ -direction in radians,  $u_1$  is the linear velocity in  $m/s$  and  $u_2$  is the angular velocity in  $rd/s$ . Note that, by convention we consider that the angle  $x_3(t) \in [-\pi, \pi)$ . The system is subjected to state and control input constraints, such that the state constraint set is

$$\mathbb{X} = \left\{ (x_1, x_2)^T \in \mathbb{R}^2 \mid \begin{array}{l} x_1^2 - x_2^2 \leq 4 \\ 4x_2^2 - x_1^2 \leq 16 \end{array} \right\} \quad (18)$$

and the control input constraint set is

$$\mathbb{U} = [0.2, 2] \times [-1, 1]. \quad (19)$$

For the optimization problem (2)-(3), the prediction horizon was chosen to be  $N=20$  and the stage cost

$$\ell(x, u) = 100 \|(x_1, x_2)^T - x^r\|^2 + \|u\|^2$$

where  $x^r \in \mathbb{R}^2$  is a reference position. The terminal cost is set to  $L(x) = 100 \|(x_1, x_2)^T - x^r\|^2$ . In order to construct the symbolic abstraction we have chosen  $R_{\mathbb{X}} = [\underline{x}, \bar{x}]$  with

$$\underline{x} = (-3.5, -2.6, -\pi)^T, \quad \bar{x} = (3.5, 2.6, \pi)^T.$$

The following subsections compare the performance of OSC (Optimized Symbolic Control), MPC, and SgMPC with enlarged terminal constraints guided by OSC. For these simulations, we used a symbolic model with 109200 symbolic states and 40 symbolic inputs. Using symbolic control techniques, a control invariant was computed and is shown in Figure 2.

In our first experiment, we consider a reference  $x^r = (0.5, 0.5)$  inside the constraint set. Figure 3 shows that the three behaviors are quite acceptable. The reference is tracked correctly by all controllers and all state and input constraints are satisfied. OSC is slightly less accurate than MPC and SgMPC in

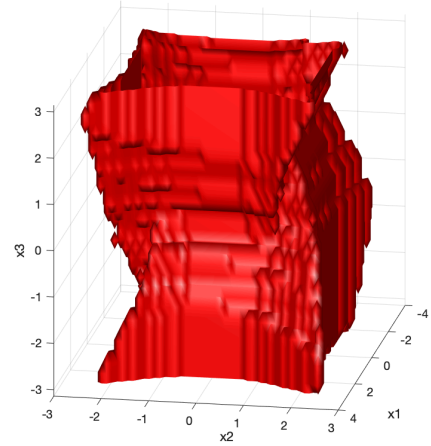


Figure 2. Controlled invariant  $\mathbb{X}_I$  computed using symbolic control techniques.

tracking. The asymptotic performances of MPC and SgMPC are equivalent. The transitory phase of MPC is shorter than that of SgMPC.

In our second experiment, we consider a reference

$$x^r = (\sqrt{32/3}, \sqrt{20/3})$$

on the boundary of the constraint set. The simulation results are shown in Figure 4. First, it should be noticed that the simulation of the system controlled by MPC stopped at  $t = 13$  because the optimization became infeasible. In comparison, both OSC and SgMPC are able to enforce the constraints at all time. As for tracking the reference, one can see that the trajectory controlled by OSC remains quite far from the reference  $x^r$  which is located in the top right corner of the domain. In comparison, SgMPC is much more successful in driving the system close to the reference. Hence, it appears that SgMPC is an interesting alternative to MPC and OSC since it makes it possible to combine high performances with safety constraints. State and inputs signals are for SgMPC are represented on Figure 5. One can check that the control signals are periodic after some time.

## 6. CONCLUSION

The recursive feasibility of NMPC has been revisited in the light of the design enhancements which can be obtained through the symbolic control design. As a main theoretic contribution, we propose the use of time-varying terminal constraints to guarantee the controlled invariance of the feasible set. From the methodological point of view, it is shown that such time-varying constraints can be obtained from a symbolic control. The proposed Symbolically guided Model Predictive Control scheme proves to provide a simple and guaranteed solution for the complex control of nonlinear constrained dynamical systems. In the future, it is planned to extend this approach to provide performance and stability guarantees.

## REFERENCES

- Belta, C., Yordanov, B., and Gol, E.A. (2017). *Formal methods for discrete-time dynamical systems*, volume 15. Springer.
- Chmielewski, D. and Manousiouthakis, V. (1996). On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29(3), 121–129.

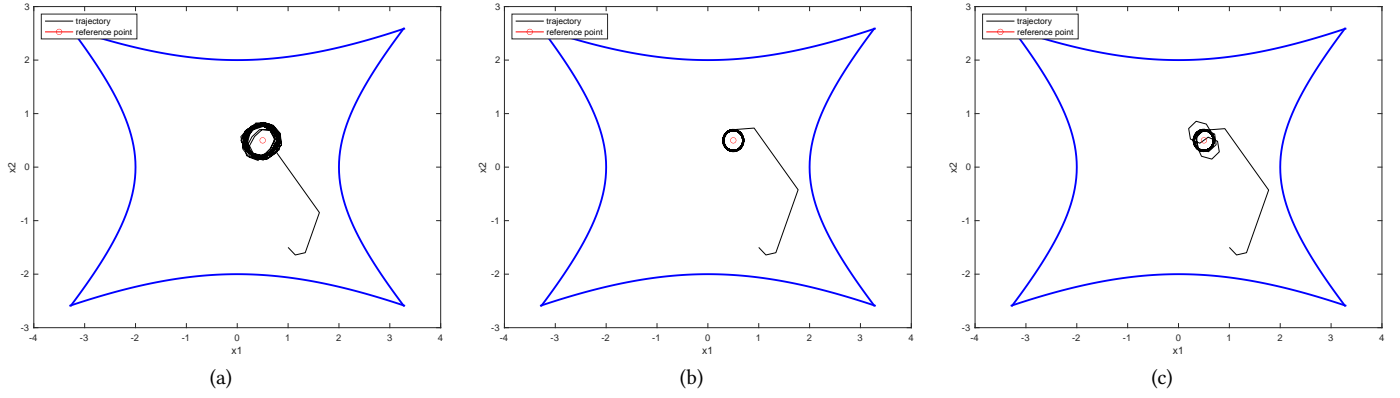


Figure 3. For reference position  $x^r = (0.5, 0.5)$ , closed-loop trajectories with OSC (a), MPC (b), SgMPC (c).

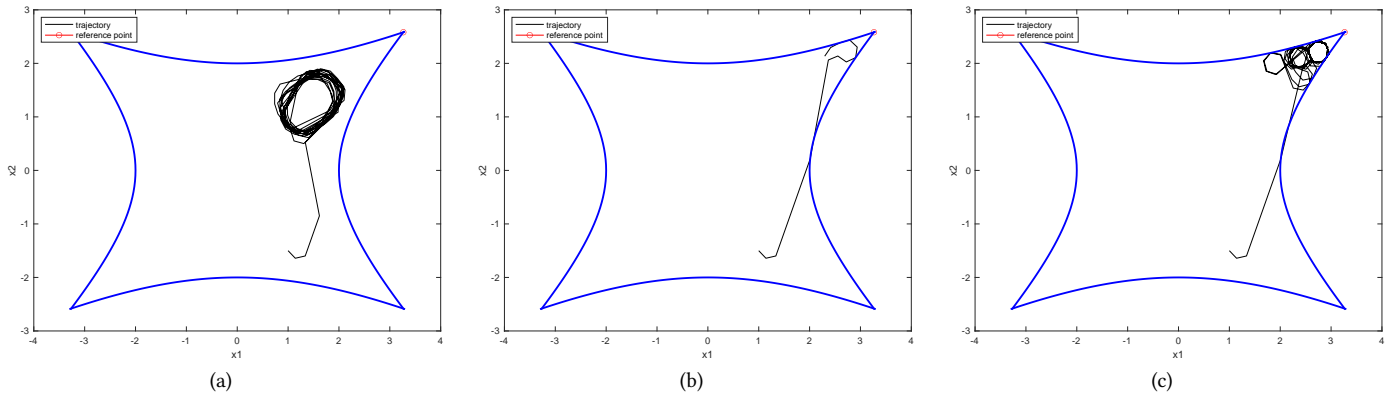


Figure 4. For reference position  $x^r = (\sqrt{32/3}, \sqrt{20/3})$ , closed-loop trajectories with OSC (a), MPC (b), SgMPC (c).

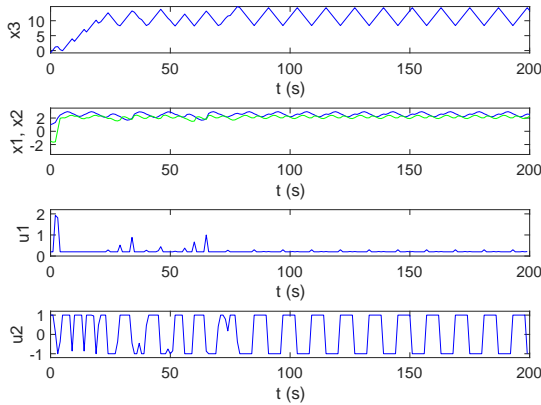


Figure 5. For reference position  $x^r = (\sqrt{32/3}, \sqrt{20/3})$ , state and input signals for SgMPC.

Girard, A. and Eqtami, A. (2021). Least-violating symbolic controller synthesis for safety, reachability and attractivity specifications. *Automatica*, 127, 109543.

Grüne, L. and Pannek, J. (2017). Nonlinear model predictive control. In *Nonlinear model predictive control*, 45–69. Springer.

Kögel, M. and Findeisen, R. (2013). Stability of nmpc with cyclic horizons. In *9th IFAC Symposium on Nonlinear Control Systems*, 46.

Lazar, M. and Tetteroo, M. (2018). Computation of terminal costs and sets for discrete-time nonlinear mpc. *IFAC-PapersOnLine*, 51(20), 141–146.

Limon, D., Alamo, T., and Camacho, E. (2005). Enlarging the domain of attraction of mpc controllers. *Automatica*, 41,

629–635.

Löfberg, J. (2012). Oops! i cannot do it again: Testing for recursive feasibility in mpc. *Automatica*, 48(3), 550–555.

Maciejowski, J.M. (2002). *Predictive control: with constraints*. Pearson education.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.

Mayne, D.Q. and Falugi, P. (2019). Stabilizing conditions for model predictive control. *International Journal of Robust and Nonlinear Control*, 29(4), 894–903.

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.

Meyer, P.J., Devonport, A., and Arca, M. (2021). *Interval Reachability Analysis: Bounding Trajectories of Uncertain Systems with Boxes for Control and Verification*. Springer.

Meyer, P.J., Girard, A., and Witrant, E. (2015). Safety control with performance guarantees of cooperative systems using compositional abstractions. *IFAC-PapersOnLine*, 48(27), 317–322.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, 11(7), 733–764.

Reissig, G., Weber, A., and Rungger, M. (2016). Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4), 1781–1796.

Sznaier, M. and Damborg, M.J. (1987). Suboptimal control of linear systems with state and control inequality constraints. In *26th IEEE Conference on Decision and Control*, 761–762.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media.