Set-valued Model Predictive Control

Nathalie Risso, Berk Altin, Ricardo G. Sanfelice, and Jonathan Sprinkle

Abstract This chapter presents a formulation for Model Predictive Control (MPC) where system dynamics are represented in terms of sets. This formulation can be useful when dealing with systems that require optimal solutions subject to constraint satisfaction, in the presence of variability or uncertainty. This is the case of cyber-physical systems (CPS) where there is a need to account for computation or communication constraints in algorithm design while also considering model uncertainty. In our setting, the system is represented by a model where the state is set-valued and dynamics are defined by a set-valued map. Unlike other formulations for MPC, we consider a set-valued cost function, which associates a real-valued cost to each set-valued system trajectory. For this framework, we provide a notion of stability for systems with set-valued states which follows the classic notion of Lyapunov stability, which is later used to define the necessary conditions for the set-valued MPC to render a collection of sets stable. Throughout the chapter, examples illustrate the notions and results introduced.

Nathalie Risso
University of Arizona, 1235 James E. Rogers Way, Tucson, AZ 85719 e-mail: nrisso@arizona.edu

Berk Altin
University of California at Santa Cruz, 1156 High Street Santa Cruz, CA 95064 e-mail: berkaltin@ucsc.edu

Ricardo G. Sanfelice
University of California at Santa Cruz, 1156 High Street Santa Cruz, CA 95064 e-mail: ricardo@ucsc.edu

Jonathan Sprinkle
Vanderbilt University, 2201 West End Ave, Nashville, TN 37235 e-mail: jonathan.sprinkle@vanderbilt.edu

1 Introduction

1.1 Motivation

Model predictive control (MPC) has become a popular choice for the control of cyber-physical systems. This election is based mainly in the ability of MPC to explicitly incorporate system constraints along with online reasoning, which allows for a rich set of applications, such as those found in industrial processes, integration of intermittent energy sources, safety applications, and autonomous systems. An important challenge associated to the implementation of MPC for CPS is related to selecting models that may be accurate enough to represent the system dynamics, at the same time that they provide timely and safe responses. This is particularly important when dealing with the control of autonomous systems, where often safety requirements are encoded as constraints which may be obtained in real-time from perception systems. An example of this can be the control of an autonomous vehicle where a computer vision system identifies obstacles along a trajectory, which can be incorporated into the system dynamic constraints, to be avoided in the path selection. As the vehicle moves and the vision system output may be subject to uncertainty due, for instance, to sensor and road characteristics, safety requirements may need to be encoded as time-varying set-valued constraints. Finding accurate models to represent the interactions between the cyber and physical components, along with uncertainty can be a difficult task, and often approximations are required to represent inherent system variability which cannot be captured by one model. To deal with these challenges, the MPC formulation requires models suitable to capture this variability so it can be considered in decision-making, thus ensuring the generation of controls that can lead to feasible, safe, and computationally efficient trajectories.

The approach presented here considers a set-based representation to encode the system variability and constraints. Sets are a natural tool to encode systems constraints, as well as to represent variability, in particular when dealing with regions of the state space for which only partial information is available, or where the state space is obtained using data analysis techniques [1]. Additionally, several control tools are already presented in a set-based context [2, 3], often accompanied of computation and approximation methods that facilitate encoding and dealing with sets representation [4, 5]. In particular, we consider here a set-based representation that follows a dynamical systems perspective, namely, set dynamical systems, along with an MPC formulation which uses the sets framework on its definition.

1.2 Results in this chapter

This chapter introduces a representation for discrete-time systems with set-valued states. The solutions associated to these systems are given by sequences of sets or tubes, for which the following results are established:

1. A definition for stability of a collection of sets C, which follows the classic Lyapunov stability notion, is presented.
2. The formulation of a set-valued model predictive control framework, with a set-valued cost function, that allows to consider the cost associated to set-valued trajectories, which are used to represent system variability.
3. Conditions that yield the constrained optimal control problem associated to the set-based MPC formulation feasible and stable are presented, thus extending existing stability results for classic MPC to a set-based approach.

1.3 Related works

The idea of using sets to characterize system properties is not new, as several early works discussing *generalized systems* such as [6], [7], [8] present this notion. In addition, in the literature of control systems and classic MPC, set-based frameworks are used to study system properties and to encode system constraints as presented, for instance, in [2, 9, 3, 10] and [11]. In the case of robust model predictive control, tube-based approaches such as [2, 3, 12, 13] consider also set properties to represent the tubes along which the system trajectories are contained. In previous works (see [14], [15], [16], [17]) we study the properties of systems with set-valued states evolving in discrete time. An emphasis is given on extending notions and tools to characterize relevant properties defined for classic discrete time dynamical systems towards a representation based on sets. In this chapter we use the set dynamical systems framework to present a formulation for MPC which can be used to study the effects of system variability in predictive control. This approach shares ideas with tube-based MPC formulations, but unlike other approaches, it considers a set-valued cost functional which can be use to generalize approaches in tube-based MPC and Robust MPC formulations. We discuss also, some of the computational aspects of the proposed approach, particularly when dealing with CPS applications. As an illustration, we formalize the approach in [18] and provide some basic examples to help visualize how key properties of MPC can be generalized to a set-valued representation and thus can aid in the design of stabilizing and safe CPS controllers.

1.4 Chapter Organization

The remainder of this chapter is organized in the following manner. Section 2 presents a framework for the set dynamical systems considered here; Section 3 presents the formulation for the proposed set-valued predictive controller. Basic assumptions associated to this set-valued MPC are presented in Section 4, which are later used in Section 5 to establish conditions for feasibility and stability results. Connections to classic notions for discrete time MPC are also discussed here. Then, in Section 6 we discuss some implementation options for the proposed controller, and provide a numerical example related to the control of autonomous vehicles. General conclusions and future works associated are presented in Section 7.

Notation

The following notation is used throughout this chapter. The set of natural numbers including 0 is denoted as $\mathbb{N}$, i.e., $\mathbb{N} = \{0, 1, \ldots\}$. The set of real numbers is referred as $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ denotes the nonnegative real numbers and the n-dimensional Euclidean space is represented as $\mathbb{R}^n$. Given a vector $x \in \mathbb{R}^n$, $|x|_\sigma$ denotes the σ-norm, with $\sigma \in [1, \infty]$. Given a closed set $\mathcal{A} \subset \mathbb{R}^n$ and $x \in \mathbb{R}^n$, we define the distance $|x|_{\mathcal{A}} := \inf_{y \in \mathcal{A}} |x - y|$. Given a map V its domain of definition is denoted as $\operatorname{dom} V$. We also use the notation $\operatorname{dom}(X, U)$ to refer to the time domain of a solution-pair in Section 2. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class-$\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$. If α is also unbounded then it is said to be of class-$\mathcal{K}_\infty$. For a given pair of sets S_1, S_2, the notation $S_1 \subset S_2$ indicates that S_1 is a subset of S_2. We will refer to sets of subsets of $\mathbb{R}^n$ as collections (of sets). Given a set S, the notation $\mathcal{P}(S)$ denotes the collection all of nonempty subsets of S, namely $\mathcal{P}(S) = \{S_1, S_2, \ldots\}$, where for each i, S_i is a nonempty subset of S. The collection of all nonempty compact subsets of S is denoted as $\mathcal{P}_C(S)$. For a given pair of collections of sets $\mathcal{C}_1$, $\mathcal{C}_2$, the notation $\mathcal{C}_1 \subset \mathcal{C}_2$ indicates that $\mathcal{C}_1$ is a subset of the collection $\mathcal{C}_2$, namely, it indicates that every element of $\mathcal{C}_1$ is an element of $\mathcal{C}_2$. We denote the intersection between $\mathcal{C}_1$ and $\mathcal{C}_2$ as $\mathcal{C}_1 \cap \mathcal{C}_2$ which corresponds to a collection, namely, it indicates that all the elements in the collection $\mathcal{C}_1 \cap \mathcal{C}_2$ are both in $\mathcal{C}_1$ and in $\mathcal{C}_2$. Given a set C and a collection of sets $\mathcal{C}$, notation $C \in \mathcal{C}$ indicates that the set C is an element in the collection $\mathcal{C}$. In general we refer to collections of sets simply as collections. For a variable x evolving in discrete-time, we denote by x^+ the value of x after a discrete-time step. Discrete time is also denoted by $j \in \mathbb{N}$ and for a given function $j \to x(j)$ of discrete time $j \in \mathbb{N}$, we use the notation x_j to represent $x(j)$.

Basic Definitions

Definition 1 (Hausdorff distance)

Given two closed sets $\mathcal{A}_1$, $\mathcal{A}_2 \subset \mathbb{R}^n$ the Hausdorff distance between them is given by

$$d_H(\mathcal{A}_1, \mathcal{A}_2) = \max\left\{ \sup_{x \in \mathcal{A}_1} |x|_{\mathcal{A}_2}, \sup_{z \in \mathcal{A}_2} |z|_{\mathcal{A}_1} \right\}$$

Given sets $\mathcal{A}_1$, $\mathcal{A}_2$ and d_H as in Definition 1, $d_H(\mathcal{A}_1, \mathcal{A}_2) = 0$ if and only if $\mathcal{A}_1 = \mathcal{A}_2$.

Definition 2 (distance from a set to a collection)

Given a set $X \in \mathcal{P}_C(\mathbb{R}^n)$ and a collection $\mathcal{A} \subset \mathcal{P}_C(\mathbb{R}^n)$, the distance from X to $\mathcal{A}$ is given by

$$d(X, \mathcal{A}) = \inf_{A \in \mathcal{A}} d_H(X, A)$$

The definition of d above extends the notion of distance from a point x to a set $\mathcal{A}$, to the case when the point x is replaced by a set X and the set $\mathcal{A}$ is replaced by a collection. Note also that the distance between a set X and a collection $\mathcal{A}$ is only equal to zero in the case where the set X coincides with an element of the collection $\mathcal{A}$, i.e., if $X \in \mathcal{A}$. The later is similar to the notion of distance from a point to a set presented in section 1.4.

Definition 3 (Set-valued maps)

Let G be a set-valued map, mapping sets in $\mathcal{P}(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m)$ to sets in $\mathcal{P}(\mathbb{R}^n)$, and denoted as $G : \mathcal{P}(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m) \rightrightarrows \mathcal{P}(\mathbb{R}^n)$. Given sets $X \in \mathcal{P}_C(\mathbb{R}^n)$, and $U \in \mathcal{P}_C(\mathbb{R}^m)$, $G(X, U)$ is defined as

$$\begin{aligned} G(X, U) &= \bigcup_{x \in X, u \in U} G(x, u) \\ &:= \{(x', u') \in G(x, u) : x \in X, u \in U\} \end{aligned}$$

In the remaining of this work we denote sequences of sets with boldface to distinguish them from the notation used to refer to a single set in the sequence. Hence, the sequence $\{T_i\}_{i=0}^{\infty}$ is represented as $\mathbf{T}$, and a set within this sequence is denoted by T_i.

For the next definition, we consider the definition of limit of a sequence of sets presented in [19].

Definition 4 (continuity of a set-valued map) [19, Definition 5.4] A set-valued map $S : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is *outer semicontinuous* at $\bar{x}$ if

$$\lim_{x \to \bar{x}} \sup S(x) \subset S(\bar{x})$$

and *inner semicontinuous* at $\bar{x}$ if

$$\lim_{x \to \bar{x}} \inf S(x) \supset S(\bar{x})$$

It is *continuous* at $\bar{x}$ if it is both outer semicontinuous and inner semicontinuous at $\bar{x}$.

2 Set dynamical systems

In this work, we propose a set-based predictive control scheme for discrete-time systems with solutions given by sequences of sets. This framework follows the ideas in [14] where the evolution of the state of a system is represented by a sequence of sets

$$X_0, X_1, X_2, \quad \ldots \quad X_j, \quad \ldots \quad \subset \quad \mathbb{R}^n \tag{1}$$

where $j \in \{0, 1, 2, \ldots\}$ and X_0 is the initial set. The sequence of sets in (1) defines a state trajectory. Such a trajectory defines the sequence of sets $\mathbf{X}$, indexed by $j \in [0, J]$, $J \in \mathbb{N}$. These solutions can be generated when incorporating uncertainty and the effects of several possible inputs in a "classical" dynamical system given by $x^+ = g(x, u)$, with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. We refer to these systems as *set dynamical systems*.

We consider set dynamical systems defined by

$$\begin{aligned} X^+ &= G(X, U) \\ (X, U) &\in \mathrm{d} \end{aligned} \tag{2}$$

where X is the set-valued state and U is the set-valued input, $G : \mathcal{P}(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m) \rightrightarrows \mathcal{P}(\mathbb{R}^n)$ is a set-valued map defining the evolution of the set-valued state, and the collection $\mathrm{d} = D_1 \times D_2$, with collections $D_1 \subset \mathcal{P}(\mathbb{R}^n)$ and $D_2 \subset \mathcal{P}(\mathbb{R}^m)$, defines constraints that the state and the inputs must satisfy. The collection d can be useful for instance to specify safety constraints, which can define regions in the state space where the system is safe to operate.

Note that the framework presented here follows the one described in [14], [15] and it seeks to extend the discrete-time representation used in model predictive control in [20, 11, 21]into a set-valued state space representation. Following this idea, constraints traditionally expressed in terms of sets, are presented here in terms of collections of sets, which define the admissible values for the system state and inputs.

The next definition formalizes the notion of solution pairs, which will be used when defining sequences of set-valued states generated by a sequence of inputs.

Definition 5 (Solution pair to a set dynamical system)

A solution pair for the set dynamical system in (2) is given by a sequence of compact nonempty sets $\mathbf{X}$ defining the state trajectory, and a sequence of closed nonempty sets $\mathbf{U}$ representing the input. Note that the input $\mathbf{U}$ generates the state trajectory $\mathbf{X}$ which define the solution pair. The first entry of the solution, X_0, is the initial set for the state. The sequence $(\mathbf{X}, \mathbf{U})$ is a solution to (2) if

$$\begin{aligned} X_{j+1} &= G(X_j, U_j) \\ (X_j, U_j) &\in \mathrm{d} \end{aligned}$$

for all $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$, where the domain of definition of the solution $\mathrm{dom}(\mathbf{X}, \mathbf{U})$ is given by the set $\{0, 1, 2, \ldots, J\} \cap \mathbb{N}$ with $J \in \mathbb{N} \cup \{\infty\}$[1]. A solution pair that has $J = 0$ is said to be *trivial*[2]. If the solution pair has $J > 0$ is *nontrivial*, and if it has $J = \infty$, it is *complete*. Given an initial set $X_0 \in D_1 \subset \mathcal{P}_C(\mathbb{R}^n)$, $\hat{S}(X_0)$ denotes the set of all possible solution pairs $(\mathbf{X}, \mathbf{U})$ with initial set X_0.

Note that depending on the input sequence $\mathbf{U}$ we can have different solutions $\mathbf{X}$ from the same X_0.

[1] Note that $\mathrm{dom}(\mathbf{X}, \mathbf{U})$ is equal to $\mathrm{dom}\,\mathbf{X} = \mathrm{dom}\,\mathbf{U} := \mathrm{dom}(\mathbf{X}, \mathbf{U})$

[2] Necessarily, in such case, $X_0 \notin D_1$.

Example 1 Consider a ground vehicle represented by the Dubins model. An exact discretization for this system with step size T is given in [18] by

$$x^+ = g(x,u) = \begin{bmatrix} q_1 + u_1 \frac{2\cos(\theta+u_2)\sin(u_2)}{\omega} \\ q_2 + u_1 \frac{2\sin(\theta+u_2)\sin(u_2)}{\omega} \\ \theta + 2u_2 \end{bmatrix} \tag{3}$$

where the state is given by $x := (q_1, q_2, \theta)^\top$, with (q_1, q_2) being the vehicle Cartesian coordinates, θ is the heading angle, angular velocity associated to heading given by $\omega = \dot{\theta}$, ϕ is the front wheels angle, and $u = (u_1, u_2)^\top = (v, T\omega/2)^\top$ is the input, where v represents the speed. A diagram with the associated variables is presented in Fig. 1. For this system, consider the case where there is uncertainty in the vehicle position (q_1, q_2). We capture such uncertainty by defining the initial set X_0 as the set of all possible vehicle positions for the initial time. We can represent the dynamics of this system by defining a system with data given by

$$\begin{aligned} X^+ &= G(X,U) \\ (X,U) &\in \mathrm{d} \end{aligned} \tag{4}$$

where $G(X,U) = \bigcup_{x\in X, u\in U} g(x,u)$, $\mathrm{d} = \mathcal{P}(\mathbb{R}^3) \times \mathcal{P}(\mathbb{R}^2)$. For a given input $u \in U$, the state trajectory for this system is given by a sequence of sets $\mathbf{X}$. The state trajectory for of this system from $X_0 = \{(q_1, q_2, \theta) \in \mathcal{P}(\mathbb{R}^3) : \sigma_1^{\min} \leq q_1 \leq \sigma_1^{\max}, \sigma_2^{\min} \leq q_2 \leq \sigma_2^{\max}, \theta = 0\}$, with an applied singleton input sequence $\mathbf{U}$ is depicted in Fig. 2 up to time $J = 9$.

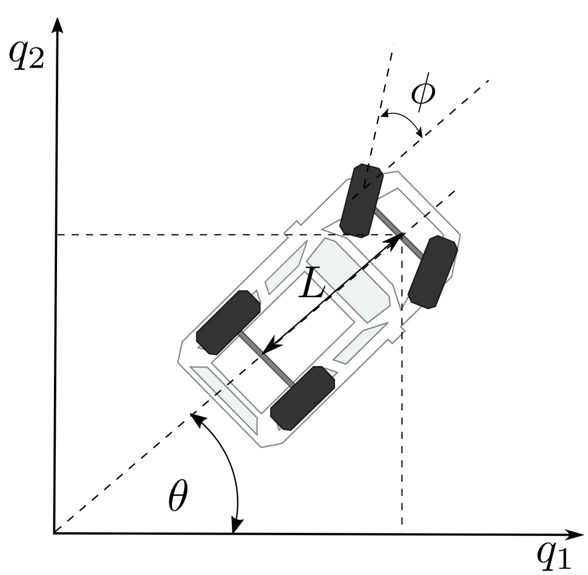

Fig. 1 Variables and parameters in Dubin's representation in Example 1

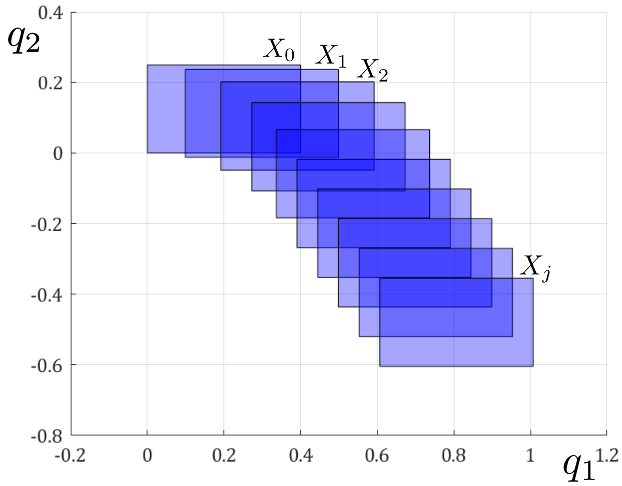

Fig. 2 Set-valued trajectory for the system in Example 1 from $X_0 = \{(q_1, q_2, \theta) \in \mathcal{P}(\mathbb{R}^3) : 0 \le q_1 \le 0.4, 0 \le q_2 \le 0.25, \theta = 0\}$, for $J = 9$, $u_1 = (2,2,2,2,2,2,2,2,2)^\top$, and $u_2 = (-\frac{\pi}{4}, -\frac{\pi}{4}, -\frac{\pi}{4}, -\frac{\pi}{4}, 0, 0, 0, 0)^\top$
.

2.1 Set dynamical systems under Static State-Feedback

Given the map $\kappa : \mathcal{P}_C(\mathbb{R}^n) \rightrightarrows \mathcal{P}(\mathbb{R}^m)$, let

$$\begin{aligned} X^+ &= G_\kappa(X) = G(X, \kappa(X)) \\ (X, \kappa(X)) &\in \mathrm{d} \end{aligned} \tag{5}$$

A solution pair $(\mathbf{X}, \mathbf{U}) = (\mathbf{X}, \kappa(\mathbf{X}))$ is said to be generated by the feedback κ. For the system in (5), we define the following notion of invariance

Definition 6 (forward and backward invariance for (5)**)** A collection $\mathcal{M} \subset \mathcal{P}(\mathbb{R}^n)$ is said to be forward invariant for the set-valued system in (5) if for every set $T \in \mathcal{M} \cap D_1$, we have $G_\kappa(T) \in \mathcal{M}$ with T such that $G_\kappa(T)$ is nonempty and it satisfies the constraints in (5). A collection $\mathcal{M} \subset \mathcal{P}(\mathbb{R}^n)$ is said to be backward invariant for (5) if for every set $T' \in \mathcal{M} \cap D_1$ for which there exists a set T with the property $T' = G_\kappa(T)$, we have $T \in \mathcal{M}$ for every such set T. A collection $\mathcal{M} \subset \mathcal{P}(\mathbb{R}^n)$ is said to be invariant if it is both forward and backward invariant.

3 Set-valued Model Predictive Control

In this section we propose a set-valued model predictive control (MPC) scheme for discrete-time systems with solutions given by sequences of sets. Given a dynamical system where variability can be captured by the representation in (2), the predictive controller is implemented by measuring the set-valued state of the plant in (5) and finding a solution pair which minimizes a cost functional, subject to constraints. As with classic moving horizon implementation for MPC, at each measurement instant, the algorithm computes an optimal control sequence of sets, from which commands are applied to the plant until the next measurement is available. Unlike other formulations for robust MPC, such as tube-based approaches [22], where the optimal control problem is designed to constraint singleton trajectories to sequences of sets or tubes, but cost is evaluated in terms of a nominal (classic) state trajectory, the cost function considered here assigns a real-valued cost to each set-valued solution pair. Considering a set-valued cost function allows to evaluate solution-pairs where state variability and also potential input variability is captured by the set-valued representation. This approach results interesting in systems where there is intrinsic variability in process variables, such as industrial applications [23] and control applications for intermittent renewable energy sources [24, 25], where targets are defined as desirable zones or *operation regions* of the state space.

Next, we describe the formulation of the set-valued MPC, where, as in the case of classic MPC strategies, the controller considers a prediction horizon $N = J \geq 1$, a control horizon $1 \leq M \leq J$, a terminal constraint collection of sets $X_V \subset \mathcal{P}_C(\mathbb{R}^n)$, a stage cost ℓ, and a terminal cost V_f.

3.1 Finite Horizon Set-valued Optimal Control

In this section we present the main elements in the formulation of the proposed set-valued predictive controller. In order to relate the formulation and properties of the set-valued MPC to the classic ones, the notation considered here resembles the ones in [11] and [21].

3.1.1 The Cost Functional

Given a solution pair $(\mathbf{X}, \mathbf{U})$ of (5) with terminal time J, a stage cost ℓ, and a terminal cost V_f, we define the cost $\mathcal{J}$ associated to the solution pair as

$$\mathcal{J}(\mathbf{X}, \mathbf{U}) = \sum_{j=0}^{J-1} \ell(X_j, U_j) + V_f(X_J) \tag{6}$$

where $\ell : \mathcal{P}_C(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m) \to \mathbb{R}_{\geq 0}$ and $V_f : \mathcal{P}_C(\mathbb{R}^n) \to \mathbb{R}_{\geq 0}$. Note that the maps ℓ and V_f assign a cost to every nonempty closed subset in $\mathcal{P}_C(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m)$ and $\mathcal{P}_C(\mathbb{R}^n)$, respectively.

3.1.2 The Constrained Optimal Control Problem

The optimal control problem to be considered is defined next.

Problem 1 Given the prediction horizon $J \geq 1$, stage cost ℓ, terminal cost V_f, terminal constraint collection X_V, constraints defined by the collection of sets d, dynamics described by the map G as in Definition 5, and initial state X_0

$$\begin{aligned} &\min_{(\mathbf{X},\mathbf{U}) \in \hat{S}(X_0)} \mathcal{J}(\mathbf{X}, \mathbf{U}) \\ &\text{subject to } X_J \in X_V \end{aligned} \tag{7}$$

For this problem the optimization is performed over solution pairs of (2), with initial condition X_0, and terminal state X_J belonging to the terminal constraint collection X_V. Here, the decision variable associated to the optimization corresponds to the sequence of inputs $\mathbf{U}$. Note that having a set-valued input allows to capture actuator variability or a range of control inputs that can lead the system to the desired performance. State-input constraints associated to (2) along with typical MPC constraints can be captured by d.

A solution pair is said to be *feasible* if it satisfies the constraints of (7). We also refer to a given sequence of inputs $\mathbf{U}$ as feasible if along with its associated $\mathbf{X}$, they correspond to a feasible pair. We define the feasible collection $\mathcal{X}$ as the collection of all sets X_0 such that there exists a feasible pair $(\mathbf{X}, \mathbf{U}) \in \hat{S}(X_0)$.

The *value function* $\mathcal{J}^* : X \to \mathbb{R}_{\geq 0}$ is defined as

$$\mathcal{J}^*(X_0) := \inf_{\substack{(\mathbf{X},\mathbf{U}) \in \hat{S}(X_0) \\ X_J \in X_V}} \mathcal{J}(\mathbf{X}, \mathbf{U}) \quad \forall X_0 \in \mathcal{X} \tag{8}$$

If the infimum is attained by a feasible $(\mathbf{X}, \mathbf{U}) \in \hat{S}(X_0)$, then the pair $(\mathbf{X}, \mathbf{U})$ is said to be optimal over the prediction interval and it is denoted $(\mathbf{X}^*, \mathbf{U}^*)$.

Note that in general, solutions to this problem may not always exist and may not be simple to compute numerically. We focus first on the properties of the resulting predictive control algorithm, and we discuss later possible computationally feasible implementations for this controller.

3.2 Set-valued MPC algorithm

Given a prediction horizon J and a control horizon M, the set-valued MPC algorithm operates by measuring the initial (set-valued) state, solving the optimal control problem described in Problem 1 to find a solution pair $(\mathbf{X}^*, \mathbf{U}^*)$. The optimal control sequence $\mathbf{U}^* = \{U_0^*, U_1^*, \dots, U_{M-1}^*\}$ is then applied to the system in (5) until time step M at which point the process in repeated for a new initial condition given by the current state measure. Note that this process defines an implicit control law given as a function of the initial state X_0, which we will denote with $\kappa_c(X)$. This process is summarized in Algorithm 1. Note that here in the last expression, the state X corresponds to the state which was used as a starting point of the optimization.

Note that by the execution of Algorithm 1, the resulting trajectories generated by the set-valued MPC correspond to concatenations of truncated optimal solutions. This notion is formalized in the next definition.

Definition 7 (solution pair generated by SVMPC)

A solution pair $(\mathbf{X}, \mathbf{U})$ is said to be generated by the set-valued MPC algorithm if it is the concatenation of a sequence of solution pairs $(\tilde{\mathbf{X}}, \tilde{\mathbf{U}})$ where for each $j \in \mathrm{dom}(\tilde{\mathbf{X}}, \tilde{\mathbf{U}})$, $(\tilde{\mathbf{X}}, \tilde{\mathbf{U}})$ in the sequence of sets is the truncation of an optimal solution pair $(\mathbf{X}^*, \mathbf{U}^*)$.

An illustration of the solution pair notion provided in the previous definition is presented in Fig. 3.

4 Basic Assumptions for Set-valued MPC

In this section we present assumptions associated to Problem 1 to ensure feasibility and stability properties. These assumptions resemble the stabilizing conditions for constrained problems in classic MPC formulations, such as the ones summarized in

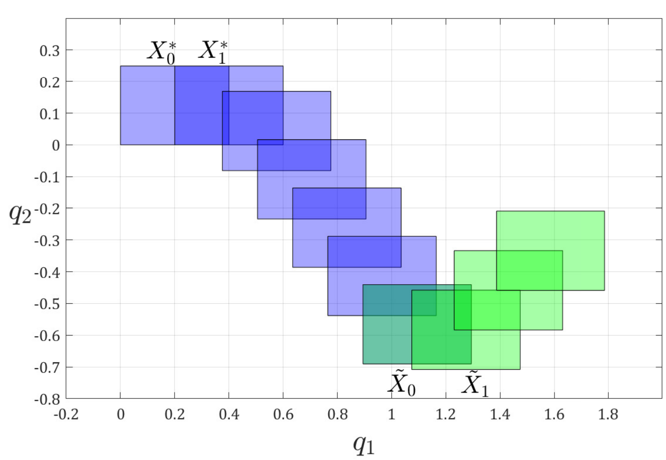

Fig. 3 Set-valued trajectory for the system in Example 2

Algorithm 1 Set-valued predictive control

Obtain initial state X
Set $X_0 = X$, $i = 0$.
while True **do**
    Solve Problem 1, obtain $(\mathbf{X}^*, \mathbf{U}^*)$
    Set $j = 0$
    for $j \leq M - 1$ **do**
        $X_{i+1} = X^*_{j+1} = G(X^*_j, U^*_j)$
        $i = i + 1$, $j = j + 1$
    end for
    Set $X_0 = X^*_M$
end while

[11]. We start by considering a basic assumption associated to the existence of an optimal solution, which is needed for feasibility.

Assumption 1 For each $X_0 \in \mathcal{X}$, there exists an optimal solution pair $(\mathbf{X}^*, \mathbf{U}^*) \in \hat{S}(X_0)$. □

Assumption 2 Given a collection $\mathcal{A} \subset X_V \subset \mathcal{P}_C(\mathbb{R}^n)$, and a stage cost $\ell : \mathcal{P}_C(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m) \to \mathbb{R}_{\geq 0}$, there exists a class-$\mathcal{K}_\infty$ function α such that $\ell(X, U) \geq \alpha(d(X, \mathcal{A}))$ for every $(X, U) \in$ d. □

Assumption 2 considers that a lower bound on the stage cost associated to the set-valued MPC formulation exists, which generalizes a similar condition for classic MPC presented in Chapter 2 in [11]. The next assumptions impose an upper bound on the terminal cost and forward invariance of the terminal collection of sets, thus representing a generalization of the "so called" basic stabilizing assumptions 2.12 and 2.13 in [11].

Assumption 3 Given a terminal cost V_f, there exists $\epsilon > 0$ such that the following hold:

(B0) There exist class-$\mathcal{K}_\infty$ functions α_1 and α_2 such that $\alpha_1(d(X, \mathcal{A})) \leq V_f(X) \leq \alpha_2(d(X, \mathcal{A}))$ for all $X \in X_V \cap \mathcal{A}_\epsilon$, where the collection $\mathcal{A}_\epsilon$ is defined as $\mathcal{A}_\epsilon := \{X \in \mathcal{P}_C(\mathbb{R}^n) : d(X, \mathcal{A}) \leq \epsilon\}$.

(B1) The inclusion $\mathcal{A}_\epsilon \cap D_1 \subset X_V$ holds. □

Assumption 4 There is a state feedback κ such that the terminal constraint collection of sets X_V is forward invariant for the system (5). Moreover, κ satisfies $V_f(G_\kappa(X)) - V_f(X) \leq -\ell(X, \kappa(X))$ for all $X \in X_V$ such that $(X, \kappa(X)) \in$ d. □

5 Properties of the Optimal Control Problem

In this section, the basic assumptions defined before are used to characterize properties of the optimal control problem formulated in Section 3. These properties are then used to develop a stability result with conditions for asymptotic stability of the system resulting from using the proposed set-based MPC.

Proposition 1 *Suppose Assumptions 2 and 4 hold. Then*

$$\ell(X, \kappa(X)) = 0 \tag{9}$$

for all $(X, \kappa(X)) \in \mathrm{d}$ *such that* $X \in \mathcal{A}$.

Proof The property follows directly from Assumptions 2 and 4. $\square$

Proposition 2 *Let* $(\mathbf{X}, \mathbf{U})$ *be a feasible solution pair to the set dynamical system in* (5)*. Suppose the terminal constraint collection* X_V *is forward invariant for the system* (5)*. Then, for any* $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$*, there exists a feasible pair* $(\mathbf{X}', \mathbf{U}') \in \hat{S}(X_j)$*; i.e.,* $X_j \in \mathcal{X}$ *for all* $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$.

Proof We prove this by constructing the feasible solution pair $(\mathbf{X}', \mathbf{U}')$. Since $(\mathbf{X}, \mathbf{U})$ is a feasible solution pair, we can obtain a feasible solution from X_1. The sequences of states $\{X_1, X_2, X_3, \ldots, G(X_J, \kappa(X_J))\}$ and the control sequence $\{U_1, U_2, U_3, \ldots, \kappa(X_J)\}$ are feasible as they satisfy all the constraints associated to Problem 1 as by Assumption 4, the collection X_V is forward invariant for the system (5). Hence, X_1 belongs to the feasible collection $\mathcal{X}$. By induction, it follows that we can define a feasible solution pair $(\mathbf{X}', \mathbf{U}') \in \hat{S}(X_j)$ for all j. $\square$

The next results present properties analogous to the ones used in classic MPC literature to establish the value function as a candidate Lyapunov function.

Lemma 1 *Suppose Assumptions 2, 3 and 4 hold. Then,* $\mathcal{J}^*(X) = 0$ *for all* $X \in \mathcal{A} \cap X_V$.

Proof Given the compact set $X_0 \in \mathcal{A} \cap X_V$, let $(\mathbf{X}^1, \mathbf{U}^1) \in \hat{S}(X_0)$ be a solution pair generated by the feedback κ. By Assumption 4, $X_j^1 \in X_V$ for all $j \in \mathrm{dom}(\mathbf{X}^1, \mathbf{U}^1)$, as $X_0 \in \mathcal{A} \cap X_V$, and X_V is forward invariant for the system (5). By Proposition 1, $\ell(X_0, \kappa(X_0)) = 0$, as $X_0 \in \mathcal{A} \cap X_V$. By Assumption 3 and the definition of α_1 and α_2, $V_f(X_0) = 0$, as $X_0 \in \mathcal{A}$. Combining these properties, by Assumption 4, $V_f(G_\kappa(X_0)) - V_f(X_0) \leq -\ell(X_0, \kappa(X_0))$, and, consequently, we have that $V_f(G_\kappa(X_0)) = 0$. Then, by its definition, it follows that $\mathcal{J}(\mathbf{X}^1, \mathbf{U}^1) = 0$ and consequently, $\mathcal{J}^*(X_0) = 0$. Therefore, $\mathcal{J}^*(X) = 0$ for all $X \in \mathcal{A} \cap X_V$. $\square$

Lemma 2 *Suppose Assumption 2 holds. Then, there exists a class-*$\mathcal{K}_\infty$ *function* α *such that the value function satisfies* $\mathcal{J}^*(X) \geq \alpha(d(X, \mathcal{A}))$ *for all* $X \in \mathcal{X}$.

Proof By Assumption 2 there exists a class-$\mathcal{K}_\infty$ function α such that $\ell(X, U) \geq \alpha(d(X, \mathcal{A}))$ for every solution pair $(\mathbf{X}, \mathbf{U}) \in \mathrm{d}$. By definition $\mathcal{J}$ is given by $\mathcal{J}(\mathbf{X}, \mathbf{U}) = \sum_{j=0}^{J-1} \ell(X_j, U_j) + V_f(X_J)$. This leads to $\mathcal{J}(\mathbf{X}, \mathbf{U}) \geq \alpha(d(X_j, \mathcal{A}))$ for every $(X, U) \in \mathrm{d}$, as $V_f(X_J) \geq 0$. Since this is valid for all $(X, U) \in \mathrm{d}$, so it is for optimal solutions X^*, which, by definition satisfy $X^* \in \mathcal{X}$, hence $\mathcal{J}^*(X) \geq \alpha(d(X, \mathcal{A}))$ for all $X \in \mathcal{X}$. $\square$

Lemma 3 *Suppose Assumption 4 holds and $X_V \subset \mathcal{X}$. Then, $\mathcal{J}^*(X_0) \leq V_f(X_0)$ for all $X_0 \in X_V$.*

Proof Consider any feasible, not necessarily optimal solution pair $(\mathbf{X}, \mathbf{U})$ generated by the feedback κ with terminal time J. Then,

$$\mathcal{J}(\mathbf{X}, \mathbf{U}) = \sum_{j=0}^{J-1} \ell(X_j, \kappa(X_j)) + V_f(X_J)$$

Using Assumption 4,

$$\mathcal{J}(\mathbf{X}, \mathbf{U}) \leq -\sum_{j=0}^{J-1} (V_f(G_\kappa(X)) - V_f(X)) + V_f(X_J)$$

$$= -(V_f(X_J) - V_f(X_0)) + V_f(X_J) = V_f(X_0)$$

Leading to $\mathcal{J}(\mathbf{X}, \mathbf{U}) \leq V_f(X_0)$, which, by definition of the value function shows that $\mathcal{J}^*(X_0) \leq V_f(X_0)$. □

Lemma 4 *Suppose Assumptions 2 and 4 hold. Let $(\mathbf{X}^*, \mathbf{U}^*) \in \hat{S}(X_0)$ be an optimal solution pair to Problem 1. Then, for any $j \in \operatorname{dom}(\mathbf{X}^*, \mathbf{U}^*)$, $\mathcal{J}^*(X_j) \leq \mathcal{J}^*(X_0) - \sum_{i=0}^{j-1} \ell(X_i, U_i)$.*

Proof Let $(\mathbf{X}, \mathbf{U})$ be an optimal solution pair with terminal time J and $X_J \in X_V$. Let $(\mathbf{X}^0, \mathbf{U}^0)$ be a solution pair such that $\operatorname{dom}(\mathbf{X}^0, \mathbf{U}^0) = \{i \in \mathbb{N} : (i+j) \in \operatorname{dom}(\mathbf{X}^*, \mathbf{U}^*)\}$. By Assumption 4, there is a solution pair $(\mathbf{X}^1, \mathbf{U}^1) \in \hat{S}(X^0(J_0))$, with terminal time J_1 generated by the feedback κ that satisfies $X^1_j \in X_V$, for all $j \in \operatorname{dom}(\mathbf{X}^1, \mathbf{U}^1)$, as X_V is forward invariant. Consider now the solution $(\mathbf{X}', \mathbf{U}')$, with terminal time J' obtained from concatenating $(\mathbf{X}, \mathbf{U})$ and $(\mathbf{X}^1, \mathbf{U}^1)$ and truncating it by removing the first $j - 1$ terms. The cost of this solution is given by

$$\mathcal{J}(\mathbf{X}', \mathbf{U}') = \mathcal{J}^*(X_0) - \sum_{i=0}^{j-1} \ell(X_i, U_i) - V_f(X^1_0)$$

$$+V_f(X^1_{J'-J_1}) + \sum_{i=0}^{J'-J_1-1} \ell(X^1_i, \kappa(X^1_i))$$

Using Assumption 4, we have that

$$-V_f(X^1_0) + V_f(X^1_{J'-J_1}) \leq -\sum_{i=0}^{J'-J_1-1} \ell(X^1_i, \kappa(X^1_i))$$

which leads to

$$\mathcal{J}(\mathbf{X}', \mathbf{U}') \leq \mathcal{J}^*(X_0) - \sum_{i=0}^{j-1} \ell(X_i, U_i)$$

Since the truncated solution starts at X_j it follows that $\mathcal{J}^*(X_j) \leq \mathcal{J}^*(X_0) - \sum_{i=0}^{j-1} \ell(X_i, U_i)$. □

5.1 Asymptotic Stability of Set-valued Model Predictive Control

In this section we use the properties defined in the previous section for the Optimal control problem to find conditions which guarantee stability for the set-valued MPC approach. We start by providing a definition of stability for a collection of sets.

Definition 8 (stability of a collection)

The set-valued MPC algorithm is said to render the collection $\mathcal{A} \subset \mathcal{P}_C(\mathbb{R}^n)$ stable for the set dynamical system in 2 if the following hold:

- There exists $\delta > 0$ such that for every $X_0 \in D_1$ satisfying $d(X_0, \mathcal{A}) \leq \delta$, there exists a solution pair $(\mathbf{X}, \mathbf{U})$ generated by the set-valued MPC algorithm originating from X_0.
- For every $\epsilon > 0$, there exists $\delta > 0$ such that given a solution pair $(\mathbf{X}, \mathbf{U})$ generated by the set-valued MPC algorithm, $d(X_0, \mathcal{A}) \leq \delta$ implies $d(X_j, \mathcal{A}) \leq \epsilon$ for all $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$.
- If, additionally, every solution pair $(\mathbf{X}, \mathbf{U})$ generated by the set-valued MPC algorithm satisfies
$$\lim_{j \to \infty} d(X_j, \mathcal{A}) = 0$$
then the set-valued MPC algorithm renders the collection $\mathcal{A}$ *asymptotically* stable.

The previous definition of stability for a collection of sets generalizes the classical notions of stability and asymptotic stability employed in classic MPC towards the set-valued state space representation considered here. Note that if the sets are replaced by points and the collections are replaced by sets in Definition 8, the classical definition of stability in [11] is recovered.

Theorem 1 *Suppose Assumptions 1, 2, 3, and 4 hold. Then, the set-valued MPC algorithm renders the collection of sets $\mathcal{A}$ asymptotically stable for the system 2.*

Proof Consider a solution pair $(\mathbf{X}, \mathbf{U}) \in \hat{S}(X_0)$ generated by the set valued MPC described in Section 3, with $X_0 \in \mathcal{X}$. $X_V \in \mathcal{X}$, by definition of the feasible set and by Assumption 3. By the previous statement and the result in Lemma 3, there is an $\epsilon' > 0$ and a class-$\mathcal{K}$ infinity function α_2 such that $\mathcal{J}^*(X_0) \leq \alpha_2(d(X_0, \mathcal{A}))$, for all $X_0 \in X_V \cap \mathcal{A}_{\epsilon'}$. By Lemma 2, there exists a class-$\mathcal{K}$ infinity function α_1 such that $\mathcal{J}^*(X_0) \geq \alpha_1(d(X, \mathcal{A}))$ for all $X_0 \in \mathcal{X}$. Given an initial set $X_0 \in \mathcal{X}$, suppose that $d(X_0, \mathcal{A}) \leq \delta$, with $\delta > 0$. By Lemma 3, we can have that $\delta \leq \alpha_2^{-1}(\alpha_1(\min(\epsilon', \epsilon)))$, such that $X_0 \in \mathcal{A}_{\epsilon'} \cap D_1 \cap X_V$. By Lemma 4, optimal solutions pairs satisfy $\mathcal{J}^*(X_j) \leq \mathcal{J}^*(X_0)$. From this and Lemmas 2 and 3 we have that

$$\alpha_1(d(X_j, \mathcal{A})) \leq \mathcal{J}^*(X_j) \leq \mathcal{J}^*(X_0) \leq \alpha_2(d(X_0, \mathcal{A}))$$

Leading to $d(X_j, \mathcal{A}) \leq \alpha_1^{-1}(\mathcal{J}^*(X_j)) \leq \alpha_1^{-1}(\alpha_2(d(X_0, \mathcal{A}))) = \epsilon$. From this we have that we can have $d(X_0, \mathcal{A}) \leq \delta$, implies that a solution starting from X_0 will satisfy $d(X_j, \mathcal{A}) \leq \epsilon$, thus showing stability in the sense of definition 8. By Lemma 4 and Assumption 2, we have that

$$\limsup_{j\to\infty} \mathcal{J}^*(X_j) \leq \lim_{j\to\infty} \mathcal{J}^*(X_0) - \sum_{j=0}^{\infty} \alpha(d(X_j, \mathcal{A})) \tag{10}$$

Let $c := \limsup_{j\to\infty} \mathcal{J}^*(X_j)$, and assume $c > 0$. By Lemma 2, $\mathcal{J}^*(X_j) \geq \alpha(d(X_j, \mathcal{A}))$, for all $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$, thus leading to $\limsup_{j\to\infty} \mathcal{J}^*(X_j) \geq \lim_{j\to\infty} \alpha(d(X_j, \mathcal{A}))$. As c is strictly positive and α is a class-$\mathcal{K}_\infty$, this means the argument of α in the last expression is nonzero, say $d(X_j, \mathcal{A}) \geq r$, with $r > 0$. If we now apply this to the expression (10), the summation term on the right side of the expression will become negative eventually as $\mathcal{J}^*(X_0)$ is finite, thus leading to a contradiction. Thus $c = 0$, leading to $\limsup_{j\to\infty} \mathcal{J}^*(X_j) = 0$ and consequently to $\lim_{j\to\infty} d(X_j, \mathcal{A}) = 0$. □

6 Implementation

The set-valued predictive control proposed in the previous sections presents several challenges for its implementation, given the need to properly generate and represent sets, and to solve online the constrained optimization formulated in Problem 1. These challenges, as discussed in [18], can be summarized as below.

1. A suitable and computationally efficient representation for the sets characterizing the dynamics must be found.
2. A solution for Problem 1 must be obtained, which may be difficult given the presence of state and inputs defined as sets, along with constraints formulated as collections of sets.
3. The computational burden associated to the numerical solution of Problem 1 may become intractable, similar to the case of some robust formulations for MPC [21].
4. Presence of delays, perturbations on the set dynamical system or unmodeled dynamics, can severely affect the performance of the described set-valued MPC implementation.

The challenges listed above are not uncommon with classic MPC implementations, such as the need for accurate, fast optimization [26] and the need to propagate and evaluate set-based trajectories, also found in reachability problems [27]. Approaches to these issues often consider over- or under-approximation of the dynamics, in order to provide computationally tractable solutions. These include the use of polytopes, zonotopes and support functions, among others, as a means to represent sets and to maintain desirable computation properties [21]. We illustrate next an implementation approach for the set-valued MPC based on a approximations using polytopes, which allows for the proposed controller to be computationally efficient. In general the implementation of the set-valued model predictive controller described here requires

1. *Selection of a proper set-valued map evolving in discrete time to represent the system dynamics*. This map can be described using Definition 3 by using classic dynamic representation.
2. *Constraints selection*. The set-valued representation allows to incorporate allowable regions defined in terms of physical construction (such as actuator range) or safety requirements (such as safe regions of the state space).
3. *Cost function and terminal constraint selection*. The selected cost function must satisfy conditions identified in Section 5, and the terminal constraint collection should contain the target set-valued state for the system.
4. *Sets representation*. The selection of a suitable representation, or approximation if needed, of the sets to be used in the system characterization requires an approach consistent with real-time requirements and computation cost. Here approaches such as the use of polytopes or zonotopes can yield faster computation times, thus providing timely and safe trajectories for the system to be controlled.

Example 2 (Autonomous vehicle control)

Consider the problem of controlling an autonomous vehicle towards a given target location described by the collection $\mathcal{X}_T = \mathcal{P}(X_T)$, while satisfying system constraints. For instance, $\mathcal{X}_T$ may represent a parking space as the terminal state. Recalling the coordinates in Fig. 1, we assume that there exists bounded uncertainty in the vehicle coordinates (q_1, q_2) due to sensor noise, while θ may be determined more exactly due to visual feedback of parking space lines: this motivates the set-valued framework. With this system, dynamics will be represented using an over approximation, i.e. the dynamics will be contained in a set, where the map G will be defined such that $G(X, U)$ is a compact convex polytope. Similar to the approach in [18] we consider a selection of constraints for the system such that the area of the set X given by its $q_1 - q_2$ projection remains constant. We present next the selection of a representation and parameters to implement the set-valued MPC for this problem.

1. Dynamics representation. We consider the system dynamics as in (3), where the state satisfies $x \in [z_1, z_2] \times [z_3, z_4] \times [z_5]$ with $z_i \in \mathbb{R}, i = 1, \ldots, 5$. With this, as the dynamics of q_1 and q_2 are decoupled, the system can be described in terms of the new variable $z = [z_1, z_2, z_3, z_4, z_5]$ by

$$z^+ = g(z, u) = \begin{bmatrix} z_1 + Tu_1 \frac{2\cos(z_5+u_2)\sin(u_2)}{u_2} \\ z_2 + Tu_1 \frac{2\cos(z_5+u_2)\sin(u_2)}{u_2} \\ z_3 + Tu_1 \frac{2\sin(z_5+u_2)\sin(u_2)}{u_2} \\ z_4 + Tu_1 \frac{2\sin(z_5+u_2)\sin(u_2)}{u_2} \\ z_5 + 2u_2 \end{bmatrix}$$

   For consistency with real actuator commands, we will consider the decision variable ($\mathbf{U}^*$) to be chosen from subsets of $\mathbb{R}^2$ consisting of a single element.

Table 1 Computation time

Discretization time (s)	Min runtime (s)	Max runtime (s)	Average runtime (s)
0.2	0.0223	0.3035	0.0403
0.1	0.0213	0.1783	0.0451
0.05	0.0232	0.2046	0.0571
0.01	0.0304	0.2296	0.0490

2. Constraint selection. Bounds associated to the state and commanded inputs are governed by physical parameters of the vehicle and sensors. In particular we consider here: $D_1 = \mathcal{P}(\mathbb{R}^2 \times \mathbb{R})$ and $D_2 = \mathcal{P}(D_u)$, with $D_u = \{(u_1, u_2) : 0 \leq u_1 \leq u_{\max}, \frac{-T}{2}\phi_{\text{car}} \leq u_2 \leq \frac{T}{2}\phi_{\text{car}}\}$, where $u_{\max}$, ϕ_{car} represent the autonomous vehicle allowable maximum speed and steering, respectively, and where T is the sampling time associated to the discrete time representation.
3. Cost Function and terminal constraint selection. We can represent the target collection as $\mathcal{X}_T = \mathcal{P}(X_T)$, where X_T can be defined by the physical dimensions of the target location. In particular here we consider $X_T = [d_1, d_2] \times [d_3, d_4] \times \mathbb{R}$, where $d_i \in \mathbb{R}$, with $i = 1, \ldots, 4$. We define the terminal constraint set $X_V \subset \mathcal{P}(\mathbb{R}^n)$ to be such that $X_V \cap \mathcal{P}(X_T)$ is nonempty.
4. Set representation. In order to steer the system towards the selected target, we define $\ell(X, U) = \sum_{i=1}^{p} |x_k|_{X_T}$, where x_k, with $k = 1, \ldots, p$, represent the vertices of the set-valued state X, which is considered to be a polytope. The terminal cost is also defined in terms of the target as $V_f(X) = \lambda \sum_{i=1}^{p} |x_k|_{X_T}$, with $\lambda \in \mathbb{R}_{\geq 0}$ a weight factor as in classic MPC.

A numerical simulation result for these settings is presented in Fig. 4 where the selected parameters for the set-valued MPC are as follows: prediction horizon $N = 6$, control horizon $M = 1$, $\lambda = 1$, target location X_T defined as $[-0.75, -0.25] \times [-0.7, -0.3] \times \mathbb{R}$, and system parameters $u_{\max} = 0.8$, $\phi_{\text{car}} = \frac{\pi}{6}$, vehicle length and width of 0.5m and 0.4m respectively, with sampling time $T = 0.2s$. Table 1 presents a summary of the computation time associated to the solution of the set-valued MPC for different discretization times.

7 Final Remarks

This chapter introduced a formulation for set-valued model predictive control. Unlike other approaches in the literature, systems with set-valued states along with a set-valued cost function for the MPC formulation are considered. Necessary conditions for the formulated set-valued predictive controllers are presented and illustrated here. Currently the solution of the proposed set-valued MPC may require high computational costs, as it is the case of other MPC formulations; however in certain cases or if the system admits a suitable approximation, efficient routines can be implemented as it is the case of polytopic sets approximations.

Acknowledgements This work is supported by the National Science Foundation under awards 1544395 and 1544396. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

1. S. Chen, Z. Wu, and P. D. Christofides, "Machine-learning-based construction of barrier functions and models for safe model predictive control," *AIChE Journal*, vol. 68, no. 6, p. e17456, 2022.
2. F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer Science & Business Media, 2007.
3. S. V. Raković, "Set theoretic methods in model predictive control," in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 41–54.
4. F. Scibilia, S. Olaru, and M. Hovd, "On feasible sets for mpc and their approximations," *Automatica*, vol. 47, no. 1, pp. 133–139, 2011.
5. C. N. Jones and M. Morari, "Polytopic approximation of explicit model predictive controllers," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2542–2553, 2010.
6. P. Habets and K. Peiffer, "Classification of stability-like concepts and their study using vector lyapunov functions," *Journal of Mathematical Analysis and Applications*, vol. 43, no. 2, pp. 537–570, 1973.
7. A. Pelczar, "Stability in generalized pseudo-dynamical systems," *Zeszyty Naukowe UJ, Prace Matematyczne*, vol. 19, pp. 33–38, 1977.
8. M. Sobański, "Limit sets in generalized pseudo-dynamical systems," *Zeszyty Naukowe III, Prace Mat*, vol. 20, pp. 145–159, 1978.
9. F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008, vol. 78.
10. S. V. Raković, "Minkowski–lyapunov functions: Alternative characterization and implicit representation," *Automatica*, vol. 139, p. 110081, 2022.

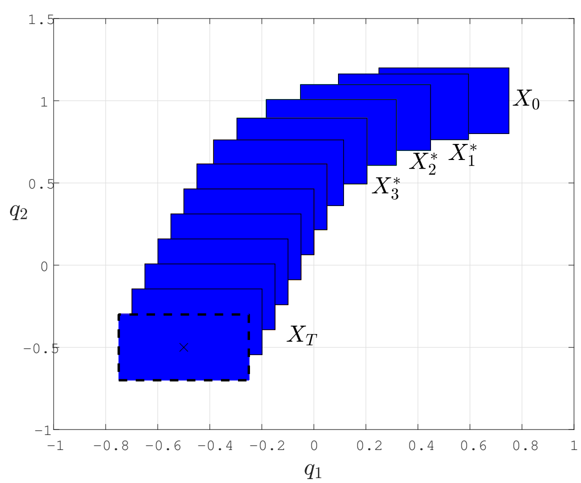

Fig. 4 Set-valued trajectory for the system in Example 2

11. J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2012. [Online]. Available: https://books.google.cl/books?id=3_rfQQAACAAJ
12. S. V. Raković and Q. Cheng, "Homothetic tube mpc for constrained linear difference inclusions," in *2013 25th Chinese Control and Decision Conference (CCDC)*. IEEE, 2013, pp. 754–761.
13. M. S. Ghasemi and A. A. Afzalian, "Robust tube-based mpc of constrained piecewise affine systems with bounded additive disturbances," *Nonlinear Analysis: Hybrid Systems*, vol. 26, pp. 86–100, 2017.
14. R. G. Sanfelice, "Asymptotic properties of solutions to set dynamical systems," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2287–2292.
15. N. Risso and R. G. Sanfelice, "Detectability and invariance properties for set dynamical systems," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 1030–1035, 2016.
16. ——, "Sufficient conditions for asymptotic stability and feedback control of set dynamical systems," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 1923–1928.
17. J. Sprinkle, N. Risso, B. Altin, and R. Sanfelice, "Challenges in set-valued model-predictive control," in *Proceedings of the Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems*, 2021, pp. 13–14.
18. J. Crowley, Y. Zeleke, B. Altm, and R. G. Sanfelice, "Set-based predictive control for collision detection and evasion," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 541–546.
19. R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.
20. D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
21. W. S. Levine, L. Grüne, R. Goebel, S. V. Rakovic, A. Mesbah, I. Kolmanovsky, S. Di Cairano, D. A. Allan, J. B. Rawlings, M. A. Sehr *et al.*, "Handbook of model predictive control," 2018.
22. D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
23. Z. Wu, A. Tran, D. Rincon, and P. D. Christofides, "Machine learning-based predictive control of nonlinear processes. part i: theory," *AIChE Journal*, vol. 65, no. 11, p. e16729, 2019.
24. E. Tofighi, D. Schlipf, and C. M. Kellett, "Nonlinear model predictive controller design for extreme load mitigation in transition operation region in wind turbines," in *2015 IEEE Conference on Control Applications (CCA)*. IEEE, 2015, pp. 1167–1172.
25. D. Schlipf, D. J. Schlipf, and M. Kühn, "Nonlinear model predictive control of wind turbines using lidar," *Wind energy*, vol. 16, no. 7, pp. 1107–1129, 2013.
26. K. Zhang, J. Sprinkle, and R. G. Sanfelice, "Computationally aware switching criteria for hybrid model predictive control of cyber-physical systems," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 479–490, 2016.
27. M. Althoff, G. Frehse, and A. Girard, "Set propagation techniques for reachability analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2020.