Set-Valued Model Predictive Control

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

Abstract— **Model predictive control (MPC) is a valuable tool to deal with systems that require optimal solutions and constraint satisfaction. In the case of systems with uncertainty, the formulation of predictive controllers requires models which are capable to capture system dynamics, constraints and also system uncertainty. In this work we present a formulation for a set-valued model predictive control (SVMPC) where uncertainty is represented in terms of sets. The approach presented here considers a model where the state is set-valued and dynamics are defined by a set-valued map. The cost function associated to the proposed MPC associates a real-valued cost to each set valued (or tube-based) trajectory. For this formulation, we study conditions that can yield the constrained optimal control problem associated to the set-valued MPC formulation feasible and stable, thus extending existing stability results from classic MPC to a set-based approach. Examples illustrate the results along the paper.**

I. Introduction

Model predictive control (MPC) represents a valuable tool to deal with systems that are required to satisfy physical constraints and to optimize a criterion, such as position error or fuel consumption. Applications with these requirements are common in the area of cyber-physical systems, in particular in autonomous vehicles, where a timely response is often also a requirement. An additional challenge associated to the implementation of autonomous systems control is the presence of uncertainty, which arises often from model error, and sensor or process uncertainty. To properly deal with this uncertainty, predictive controls require models which are able to capture system dynamics, constraints, and also uncertainty.

The problem of developing predictive controllers which can satisfy state and control constraints for all realizations of uncertainty, or Robust MPC (RMPC), has been studied extensively in the literature [1], [2], where main challenges are associated to accounting for the propagation of possible trajectories generated by uncertainty. To take into account uncertainty effects, often set-theoretical methods are employed [3], [4]. Although several approaches currently exist in the literature, representations based on tubes are the most common [4]. These Tube-based MPC (TMPC) approaches consider in general a setting with dynamics given by set-valued maps. However, a nominal (singleton) trajectory is considered, for which the predictive controller defines a strategy that keeps the state within a sequence of invariant sets or tubes [5] [6]. The cost is then characterized as a function with the nominal trajectory as argument.

In this work we propose a model predictive control structure which incorporates a set-based approach building on the works in [7], [8] and [9]. The systems considered for our MPC formulation have set-valued states which evolve in discrete time, with (possibly nonlinear) dynamics defined by set-valued maps. This leads to solutions being described as sequences of sets, or tubes, as in TMPC, but that are not necessarily associated to a nominal trajectory. This representation is useful since it allows to capture system variability and system constraints in a common framework. In this work we formalize the approach in [9] and provide a framework for set-based predictive control. In our proposed setting, since the state trajectory is set-valued, the cost functional uses set-to-points maps to characterize the cost associated to each (set valued) trajectory. For this formulation, we study conditions that can yield the constrained optimal control problem associated to the set-based MPC formulation feasible and stable, thus extending existing stability results from classic MPC formulations into a set-based approach.

This paper is structured as follows. Section II presents a framework for the set dynamical systems considered in this paper. Section III presents the formulation for proposed the set-valued predictive controller. Basic assumptions associated to this set valued MPC are presented in Section IV, which are later used in Section V to establish conditions for feasibility and stability of the optimal control problem. Section VI describes implementation options for the proposed controller. General conclusions and future works associated are presented in Section VII.

II. Preliminaries

A. Notation

The following notation is used throughout this paper. 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 denoted 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 $\operatorname{dom} V$. 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

N. Risso is with the Department of Electrical and Electronics Engineering, Universidad del Bío-Bío, Collao 1202, Concepción, Chile. nrisso@ubiobio.cl

B. Altin and R. G. Sanfelice are with the Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA berkaltin@ucsc.edu ricardo@ucsc.edu

J. Sprinkle is with the Department of Computer Science, Vanderbilt University, Nashville, TN 37212, USA sprinkle@acm.org

$\mathcal{P}(S) = \{S_1, S_2, \dots\}$, 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. Given a set C and a collection of sets $\mathcal{C}$, notation $C \in \mathcal{C}$ indicates that 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 $x(j)$.

B. Basic Definitions

Definition 2.1 (Hausdorff distance): Given two closed sets $\mathcal{A}_1$, $\mathcal{A}_2 \subset \mathbb{R}^n$ the Hausdorff distance 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 2.1, $d_H(\mathcal{A}_1, \mathcal{A}_2) = 0$ if and only if $\mathcal{A}_1 = \mathcal{A}_2$.

Definition 2.2 (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 $d(X, \mathcal{A}) = \inf_{A\subset\mathcal{A}} d_H(X, A)$.

The definition of d above extends the notion of distance from a point x to a set $\mathcal{A}$, which is denoted $|x|_{\mathcal{A}}$ in Section II-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 $X \in \mathcal{A}$.

Definition 2.3 (Set-valued maps): [7] Let G be a set-valued map, mapping sets in $\mathcal{P}(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^m)$ to sets in $\mathcal{P}(\mathbb{R}^n)$. Given sets $X \in \mathcal{P}_C(\mathbb{R}^n)$, and $\mathcal{U} \in \mathcal{P}_C(\mathbb{R}^m)$, $G(X,\mathcal{U})$ is defined as

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

Definition 2.4 (inner and outer limit): [10] For a sequence of sets $\{T_i\}_{i=0}^{\infty}$ in $\mathbb{R}^n$:

- The inner limit of the sequence $\{T_i\}_{i=0}^{\infty}$, denoted $\liminf_{i\to\infty} T_i$, is the set of all $x \in \mathbb{R}^n$ for which there exist points $x_i \in T_i$, $i \in \mathbb{N}$, such that $\lim_{i\to\infty} x_i = x$.
- The outer limit of the sequence $\{T_i\}_{i=0}^{\infty}$, denoted $\limsup_{i\to\infty} T_i$, is the set of all $x \in \mathbb{R}^n$ for which there exist a subsequence $\{T_{i_k}\}_{k=0}^{\infty}$ of $\{T_i\}_{i=0}^{\infty}$ and points $x_k \in T_{i_k}$, $k \in \mathbb{N}$, such that $\lim_{k\to\infty} x_k = x$.

The *limit* of the sequence exists if the outer and the inner limit sets are equal, namely $\lim_{i\to\infty} T_i = \liminf_{i\to\infty} T_i = \limsup_{i\to\infty} T_i$

The inner and outer limit of a sequence of sets always exist and are closed, although the limit itself might not exist. When the limit of the sequence $\{T_i\}_{i=0}^{\infty}$ exists in the sense of Definition 2.4, and is equal to T, the sequence of sets $\{T_i\}_{i=0}^{\infty}$ is said to converge to the set T. 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.

Definition 2.5 (continuity of a set-valued map): [10] 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 $\lim_{x\to\bar{x}} \sup S(x) \supset S(\bar{x})$. It is *continuous* at $\bar{x}$ if it is both outer semicontinuous and inner semicontinuous at $\bar{x}$.

C. 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 [7], [8], and [11] where the evolution of the state of a system is represented by a sequence of sets

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

where $j \in \{0, 1, 2, \dots\}$ and X_0 is the initial set. The sequence of sets in (1) defines a state trajectory (or tube-based 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

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

where X is the set-valued state and $\mathcal{U}$ 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 $\mathcal{D} = D_1 \times D_2$, 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 $\mathcal{D}$ can be useful for instance to specify safety constraints, which can define regions in the state space where the system is safe to operate.

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 2.6 (Solution pair to a set dynamical system): [11] 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. 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

$$\begin{aligned} X_{j+1} &= G(X_j, \mathcal{U}_j) \\ (X_j, \mathcal{U}_j) &\in \mathcal{D} \end{aligned}$$

for all $j \in \operatorname{dom}(\mathbf{X}, \mathbf{U})$, where the domain of definition of the solution $\operatorname{dom}(\mathbf{X}, \mathbf{U})$ is given by the set $\{0, 1, 2, \dots, J\} \cap \mathbb{N}$ with $J \in \mathbb{N} \cup \{\infty\}$. A solution pair that has $J = 0$ is said

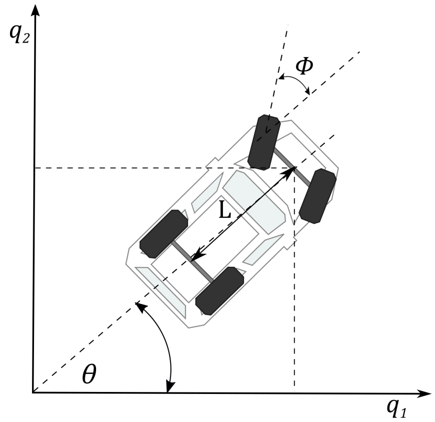

Fig. 1. Variables and parameters in Dubin's representation in Example 2.1

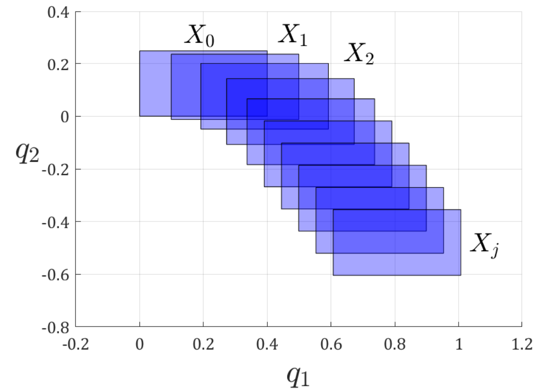

Fig. 2. Set-valued trajectory for the system in Example 2.1 from $X_0 = \{(q_1, q_2, \theta) \in \mathcal{P}(\mathbb{R}^3) : 0 \leq q_1 \leq 0.4, 0 \leq q_2 \leq 0.25, \theta = 0\}$

to be *trivial*. 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. The idea of a control input set can be useful when analyzing reachability for a given set of possible inputs.

Example 2.1: Consider a ground vehicle represented by the Dubins model. An exact discretization for this system with step size T is given in [9] by

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

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 $\omega = \dot{\theta}$, 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 Figure 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 such as (2), where $G(X, \mathcal{U}) = \bigcup_{x \in X, u \in \mathcal{U}} g(x, u)$, $\mathcal{D} = \mathcal{P}(\mathbb{R}^3) \times \mathcal{P}(\mathbb{R}^2)$. For a given input $u \in \mathcal{U}$, 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 Figure 2 up to time $J = 9$.

D. Set dynamical systems under Static State-Feedback

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

$$\begin{aligned} X^+ = G_\kappa(X) &= G(X, \kappa(X)) \\ (X, \kappa(X)) &\in \mathcal{D} \end{aligned} \quad (4)$$

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 (4), we define the following notion of invariance

Definition 2.7 (forward and backward invariance for (4)*):* A collection $\mathcal{M} \subset \mathcal{P}(\mathbb{R}^n)$ is said to be forward invariant for (4) 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 (4). A collection $\mathcal{M} \subset \mathcal{P}(\mathbb{R}^n)$ is said to be backward invariant for (4) 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.

III. 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 (4) 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 [2], 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.

Next, we describe the formulation of set-valued MPC, where, as in the case of classic MPC strategies, the controller considers a prediction horizon $N \geq 1$, a control horizon $1 \leq M \leq N$, 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.

A. Finite Horizon Set-valued Optimal Control

In this section we present the main elements in the formulation of the proposed set-valued predictive controller.

1) The Cost Functional: Given a solution pair $(\mathbf{X}, \mathbf{U})$ of (4) with terminal time N, a stage cost ℓ, and a terminal cost V_f, we define the cost $\mathcal{J}$ associated to the solution pair as

$$\mathcal{J}(\mathbf{X}, \mathbf{U}) = \sum_{j=0}^{N-1} \ell(X_j, \mathcal{U}_j) + V_f(X_N) \tag{5}$$

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.

2) The Constrained Optimal Control Problem: The optimal control problem to be solved is defined next.

Problem 1: Given the prediction horizon $N \geq 1$, stage cost ℓ, terminal cost V_f, terminal constraint collection X_V, constraints defined by the collection of sets $\mathcal{D}$, dynamics described by the map G, and initial state X_0

$$\min_{(\mathbf{X}, \mathbf{U}) \in \hat{S}(X_0)} \mathcal{J}(\mathbf{X}, \mathbf{U}) \tag{6}$$
$$\text{subject to } X_N \in X_V$$

For this problem the optimization is performed over solution pairs of (2), with initial condition X_0, and terminal state X_N belonging to the terminal constraint collection X_V. Note that the decision variables are the input sequences, which are sets. State-input constraints associated to (2) along with typical MPC constraints can be captured by $\mathcal{D}$. Note that the system dynamics is also a constraint in Problem 1.

A solution pair is said to be *feasible* if it satisfies the constraints of (6) for some X_0. We also refer to a given sequence of inputs $\mathbf{U}$ as feasible if along with its associated state trajectory $\mathbf{X}$, 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

$$\mathcal{J}^*(X_0) := \inf_{\substack{(\mathbf{X}, \mathbf{U}) \in \hat{S}(X_0) \\ X_N \in X_V}} \mathcal{J}(\mathbf{X}, \mathbf{U}) \quad \forall X_0 \in \mathcal{X} \tag{7}$$

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 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.

B. Set-valued MPC algorithm

Given a prediction horizon N and a control horizon M, the set-valued MPC algorithm operates by measuring the initial state, solving the optimal control problem described in Problem 1 to find a solution pair $(\mathbf{X}^*, \mathbf{U}^*)$. The optimal control sequence $\mathbf{U}^* = \{\mathcal{U}_0^*, \mathcal{U}_1^*, \ldots, \mathcal{U}_{M-1}^*\}$ is then applied to the system in (4) 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. This process is summarized in Algorithm 1. Note that in Algorithm 1 i tracks time and j is associated to the application of the optimal control. Additionally, in line 10, 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 3.1 (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 $j \in \operatorname{dom}(\tilde{\mathbf{X}}, \tilde{\mathbf{U}})$, $(\tilde{\mathbf{X}}, \tilde{\mathbf{U}})$ in the sequence of sets is the truncation of an optimal solution pair $(\mathbf{X}^*, \mathbf{U}^*)$.

Algorithm 1 Set-valued predictive control

```
1:  Obtain initial state X
2:  Set X_0 = X, i = 0, N, M.
3:  while True do
4:      Solve Problem 1, obtain (X*, U*)
5:      Set j = 0
6:      for j ≤ M − 1 do
7:          X_{i+1} = X*_{j+1} = G(X*_j, U*_j)
8:          i = i + 1, j = j + 1
9:      end for
10:     Set X_0 = X*_M
11: end while
```

IV. 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 [12].

Assumption 4.1: For each $X_0 \in \mathcal{X}$, there exists an optimal solution pair $(\mathbf{X}^*, \mathbf{U}^*) \in \hat{S}(X_0)$.

Assumption 4.2: 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 $\ell(X, \mathcal{U}) \geq \alpha(d(X, \mathcal{A}))$ for every $(X, \mathcal{U}) \in \mathcal{D}$.

Assumption 4.3: 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 4.4: There is a state feedback $\kappa : \mathcal{P}_C(\mathbb{R}^n) \rightrightarrows \mathcal{P}(\mathbb{R}^m)$ such that the terminal constraint collection of sets X_V is forward invariant for the system (4). Moreover, κ satisfies $V_f(G_\kappa(X)) - V_f(X) \leq -\ell(X, \kappa(X))$ for all states $X \in X_V$ such that $(X, \kappa(X)) \in \mathcal{D}$.

V. 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 III.

Proposition 5.1: Suppose Assumptions 4.2 and 4.4 hold. Then, $\ell(X, \kappa(X)) = 0$ for all $(X, \kappa(X)) \in \mathcal{D}$ such that $X \in \mathcal{A}$.

Proposition 5.2: Let $(\mathbf{X}, \mathbf{U})$ be a feasible solution pair to the set dynamical system in (4). Suppose the terminal constraint collection X_V is forward invariant for the system (4). 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 $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$.

The next results present properties analogous to the obtained for classic MPC to establish the value function as a candidate Lyapunov function.

Lemma 5.1: Suppose Assumptions 4.2, 4.3 and 4.4 hold. Then, $\mathcal{J}^*(X) = 0$ for all $X \in \mathcal{A} \cap X_V$.

Lemma 5.2: Suppose Assumption 4.2 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 $X \in \mathcal{X}$.

Lemma 5.3: Suppose Assumption 4.4 holds and $X_V \subset \mathcal{X}$. Then, $\mathcal{J}^*(X_0) \leq V_f(X_0)$ for all $X_0 \in X_V$.

Lemma 5.4: Suppose Assumptions 4.2 and 4.4 hold. Let $(\mathbf{X}^*, \mathbf{U}^*) \in \hat{S}(X_0)$ be an optimal solution pair to Problem 1. Then, for any $j \in \mathrm{dom}(\mathbf{X}^*, \mathbf{U}^*)$, $\mathcal{J}^*(X_j) \leq \mathcal{J}^*(X_0) - \sum_{i=0}^{j-1} \ell(X_i, \mathcal{U}_i)$.

A. Asymptotic Stability of Set-valued MPC

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

Definition 5.1 (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:

1) 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.
2) 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 $j \in \mathrm{dom}(\mathbf{X}, \mathbf{U})$.
3) If, in addition to 1) and 2), 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.

Theorem 5.1: Suppose Assumptions 4.1, 4.2, 4.3, and 4.4 hold. Then, the set-valued MPC algorithm renders the collection of sets $\mathcal{A}$ asymptotically stable for the system (2).

VI. 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 [9], 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 [13].
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.

These challenges are not uncommon in classic MPC, such as the need for accurate, fast optimization [14] and the need to propagate and evaluate set-based trajectories, also found in reachability problems [15]. 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 means to represent sets and to maintain desirable computation properties [13]. 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.

Example 6.1 (Autonomous vehicle control): Consider the problem of controlling an autonomous vehicle towards a given target location $\mathcal{X}_T = \mathcal{P}(X_T)$, while satisfying system constraints. Here, $\mathcal{X}_T$ 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 the 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 $G(X, \mathcal{U})$ is a compact convex polytope. Similar to the approach in [9] 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) 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.

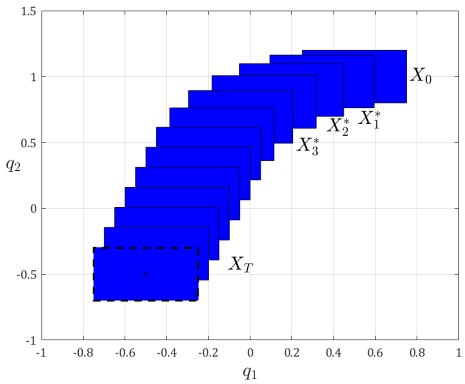

Fig. 3. Set-valued trajectory for the system in Example 6.1

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 $D_u = \{(u_1, u_2) : 0 \le u_1 \le u_{\text{max}}, \frac{-T}{2}\phi_{\text{car}} \le u_2 \le \frac{T}{2}\phi_{\text{car}}\}$, where u_{max}, ϕ_{car} represent the autonomous vehicle allowable maximum speed and steering, respectively, and where T is the sampling time.
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 $i = 1, \dots, 4$. 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. In order to steer the system towards the selected target, we define $\ell(X, \mathcal{U}) = \sum_{i=1}^{p} |x_k|_{X_T}$, where x_k, with $k = 1, \dots, p$, 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 $\lambda \in \mathbb{R}_{\ge 0}$ a weight factor as in classic MPC.

Numerical simulation result associated is presented in Figure 3 where the selected parameters for the set-valued MPC are: $N = 6$, $M = 1$, $\lambda = 1$, target location X_T defined as $[-0.75, -0.25] \times [-0.7, -0.3] \times \mathbb{R}$, and system parameters $u_{\text{max}} = 0.8$, $\phi_{\text{car}} = \frac{\pi}{6}$, vehicle length and width of 0.5m and 0.4m respectively, with sampling time $T = 0.2s$.

VII. Conclusions and Future Work

This paper presented a formulation for a set-valued model predictive controller where the state trajectory is represented as a sequence of sets. This framework can be useful to incorporate the effects of uncertainty or variability in the MPC formulation. In the proposed setting, the cost associated to state trajectories assigns a real-valued cost to solutions given by sequences of sets. For the resulting optimal control problem properties were presented and used to develop recursive feasibility and stability results associated to the set-valued MPC formulation. Even though the implementation of the proposed controller may be complex or require high computational costs, as it is the case with other optimal control formulations, successful implementation can be accomplished in particular cases, using computationally efficient sets representations, such as polytopes. Future work includes the development of practical applications where data generated from multiple vehicle trajectories can be used to obtain the characterization of the set-valued dynamics considered in this approach.

Acknowledgment

This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621, Grant no. CNS-1544396, CNS-1544395 and Grant no. CNS-2039054, by the Air Force Office of Scientific Research under Grant no. FA9550-19-1-0053, Grant no. FA9550-19-1-0169, and Grant no. FA9550-20-1-0238, by the Army Research Office under Grant no. W911NF-20-1-0253 and by Universidad del Bío-Bío Project 194810 GI/VC and Corfo grant 14ENI-26886: Ingeniería de Clase Mundial.

References

[1] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer, “Robust model predictive control for nonlinear discrete-time systems,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 13, no. 3-4, pp. 229–246, 2003.

[2] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[3] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer Science & Business Media, 2007.

[4] S. V. Raković, “Set theoretic methods in model predictive control,” in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 41–54.

[5] 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.

[6] 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.

[7] 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.

[8] N. Risso and R. G. Sanfelice, “Detectability and invariance properties for set dynamical systems,” *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 1030–1035, 2016.

[9] 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.

[10] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[11] N. Risso and R. G. Sanfelice, “Sufficient conditions for asymptotic stability and feedback control of set dynamical systems,” in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 1923–1928.

[12] 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

[13] 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.

[14] 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.

[15] M. Althoff, G. Frehse, and A. Girard, “Set propagation techniques for reachability analysis,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2020.