



HAL
open science

Safe Learning-Based Model Predictive Control Using The Compatible Models Approach

Anas Makdesi, Antoine Girard, Laurent Fribourg

► **To cite this version:**

Anas Makdesi, Antoine Girard, Laurent Fribourg. Safe Learning-Based Model Predictive Control Using The Compatible Models Approach. *European Journal of Control*, 2023, 74, pp.100849. 10.1016/j.ejcon.2023.100849 . hal-04134216

HAL Id: hal-04134216

<https://hal.science/hal-04134216>

Submitted on 20 Jun 2023

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.

Safe Learning-Based Model Predictive Control Using The Compatible Models Approach^{*}

Anas Makdesi^{a,*}, Antoine Girard^a and Laurent Fribourg^b

^aUniversité Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 3 Rue Joliot Curie, Gif-sur-Yvette, 91190, France

^bUniversité Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 4 avenue des Sciences, Gif-sur-Yvette, 91190, France

ARTICLE INFO

Keywords:

Model Predictive Control
Data-Driven Control
Safe Learning

ABSTRACT

In this paper, we introduce a novel approach to safe learning-based Model Predictive Control (MPC) for nonlinear systems. This approach, which we call the “compatible model approach”, relies on computing two models of the given unknown system using data generated from the system. The first model is a set-valued over-approximation guaranteed to contain the system’s dynamics. This model is used to find a set of provably safe controller actions at every state. The second model is a single-valued estimation of the system’s dynamics used to find a controller that minimises a cost function. If the two models are compatible, in the sense that the estimation is included in the over-approximation, we show that we can use the set of safe controller actions to constrain the minimisation problem and guarantee the feasibility and safety of the learning-based MPC controller at all times. We present a method to build an over-approximation for nonlinear systems with bounded derivatives on a partition of the states and inputs spaces. Then, we use piecewise multi-affine functions (defined on the same partition) to calculate a system’s dynamics estimation that is compatible with the previous over-approximation. Finally, we show the effectiveness of the approach by considering a path-planning problem with obstacle avoidance.

1. INTRODUCTION

As Model Predictive Control (MPC) has established its position as a vital tool for many control problems, research to incorporate data-driven methods into MPC has been surging recently [8, and references therein], with topics such as safety and robustness still remaining largely open topics [16, 11]. While some use input-output data to find optimal control policies directly, eliminating the need for building a model [7, 4], others use the data to find data-driven models and then apply the MPC scheme [1, 9]. A comparison between these direct and indirect approaches is provided in [12].

In this work, we use a data set sampled from a given system to build two models of this system. The first model is used to ensure strong safety requirements, whereas the second is used to find a controller minimising a cost function and thus fulfilling some performance requirements. The idea of decoupling safety and performance by using two models is introduced in [1], where one model is used to ensure safety using tube MPC, and the other model, updated online, is used to minimise a cost function. By contrast, both models are calculated offline in this work. Additionally, we do not impose any linear structure on the studied system.

To calculate the model enforcing the safety requirement, we use an algorithm introduced in [14] to build set-valued over-approximations of systems dynamics with bounded derivatives and subject to bounded disturbance. We first use

the upper and lower bounds on the derivatives to rewrite the data set into two new data sets that can be seen as generated from two monotone functions. Then, by relying on the monotonicity property, we can build the tightest interval-valued over-approximation of the two monotone functions on a given rectangular space partitioning. The two set-valued maps can then be used to find an interval-valued over-approximation of the original systems. We use the interval-valued over-approximation to calculate finite-state symbolic models of the system or what is called an abstraction. Using the scenario approach to find finite-state abstraction is investigated in [10, 13], but instead of calculating probabilistic guarantees for the relation between the true system and the finite-state representation, our method offers robust ones. Working with finite-state representation allows us to implement discrete controller synthesis techniques to automatically find at each state a safe set of inputs [15, 3]. The idea of calculating data-driven bounds on the system’s dynamics is utilised in [6] to compute an MPC controller; but in our case, the over-approximating model is used to find the safe inputs.

After finding a set of safe inputs, a single-valued estimation of the true system’s dynamics is calculated. This single-valued model is used to compute the MPC controller and achieve the desired performance. We show that if we choose a single-valued estimation which is included in the first over-approximation model, we can use the set of safe inputs as a constraint on the MPC optimisation problem to ensure safety while also optimising the performance. This notion of single-valued estimation being included in the over-approximation is referred to as model compatibility. We provide a way to find such a compatible estimation by calculating a piecewise multi-affine estimation of the system’s dynamics on the same space partition we used to

^{*}This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 725144).

*Corresponding author

✉ anas.makdesi@centralesupelec.fr (A. Makdesi)

ORCID(s): 0000-0001-8257-833X (A. Makdesi)

find the over-approximation. We make use of the fact that multi-affine functions on interval domains can be written as a linear combination of the values of the function on the vertices of the interval [2]. This will help us write the compatible estimation problem as a sparse quadratic optimisation problem to estimate the value of the function on the vertices of the partition.

This paper is organised as follows. In section 2, we introduce the compatible models approach for the safe learning-based MPC. Section 3 is dedicated to the calculation of the first set-valued over-approximation model, whereas Section 4 introduces the piecewise multi-affine estimation functions compatible with the over-approximation. Finally, Section 5 presents a numerical experiment to show the effectiveness of the proposed approach.

Notations $\overline{\mathbb{R}} = [-\infty, +\infty]$ is the set of *extended real numbers*. We use bold lowercase letters to represent vectors, e.g. $\mathbf{z} \in \mathbb{R}^n$; subscripts are used to differentiate between multiple vectors \mathbf{z}_i , whereas normal lowercase letters with superscripts z^i are used to denote the i^{th} component of a vector \mathbf{z} . Given two vectors $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n$, we define the partial order \leq on \mathbb{R}^n to be $\mathbf{z}_1 \leq \mathbf{z}_2$ if and only if $z_1^i \leq z_2^i$ for all $i = 1, \dots, n$. $[\mathbf{z}_1, \mathbf{z}_2] = \{\mathbf{z} \in \mathbb{R}^n \mid \mathbf{z}_1 \leq \mathbf{z} \leq \mathbf{z}_2\}$ defines a closed interval of \mathbb{R}^n . $\|\cdot\|$ denotes the Euclidean norm. $\mathbf{e}_k \in \mathbb{R}^n$ is the vector whose k^{th} component is 1 and all others are 0.

2. Safe Learning-Based MPC

In this section, we study how to implement a data-driven MPC strategy under strict safety requirements. The proposed solution, which we call the ‘‘compatible models approach’’, consists of finding two models for the system. The first one is a set-valued over-approximation of the system’s dynamics which will be used to enforce the safety requirement, whereas the second one is a single-valued estimation of the system’s dynamics. From the single-valued estimation, we can find the controller that minimizes a given cost function.

We start by defining the setup we have. Given $X \subseteq \mathbb{R}^{n_x}$, $U \subseteq \mathbb{R}^{n_u}$, $W \subseteq \mathbb{R}^{n_x}$, let us consider a discrete-time nonlinear system of the form:

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t) \quad (1)$$

where $\mathbf{x} \in X$, $\mathbf{u} \in U$, $\mathbf{w} \in W$ are the state, input, and disturbance. $f : X \times U \rightarrow X$ is an unknown nonlinear function. Throughout this paper, we will make the following working assumptions:

Assumption 1. *The unknown function f has bounded derivatives i.e. for all $\mathbf{x} \in X$, $\mathbf{u} \in U$:*

$$\begin{aligned} \frac{\partial f^i}{\partial x^j}(\mathbf{x}, \mathbf{u}) &\in [\underline{\alpha}_{ij}, \overline{\alpha}_{ij}], \quad i, j \in \{1, \dots, n_x\}, \\ \frac{\partial f^i}{\partial u^j}(\mathbf{x}, \mathbf{u}) &\in [\underline{\beta}_{ij}, \overline{\beta}_{ij}], \quad i \in \{1, \dots, n_x\}, \quad j \in \{1, \dots, n_u\}, \end{aligned}$$

where the bounds $\underline{\alpha}_{ij}, \overline{\alpha}_{ij}, \underline{\beta}_{ij}, \overline{\beta}_{ij} \in \mathbb{R}$ are assumed to be known. The set of disturbances $W = [\underline{\mathbf{w}}, \overline{\mathbf{w}}]$, is a bounded

interval with known bounds $\underline{\mathbf{w}}, \overline{\mathbf{w}} \in \mathbb{R}^{n_x}$ and such that $0 \in W$.

Assumption 2. *We are given a set of data generated from the dynamic system (1):*

$$D = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}'_k) \mid \mathbf{x}'_k \in f(\mathbf{x}_k, \mathbf{u}_k) + W, k \in \mathbb{K}\}$$

where \mathbb{K} is a finite set of indices.

Collecting the data set D can be done in various fashion. One can for instance sample the dynamics of the system randomly using independent samples. This can be done easily if one can use a black box model of (1) to generate independent simulations. However, our approach does not require to use samples that are independent. Actually, they can be collected by recording the evolution of the true system on a given period of time. In that case, we would have $\mathbf{x}'_k = \mathbf{x}_{k+1}$.

Remark 1. An algorithm to determine the bounds on the derivatives of the function f and the bounds on the disturbance W was introduced in [14]. The algorithm uses the set of data D , and the resulting bounds are valid with probabilistic guarantees that depend on the number of sampled points. However, in that case, the samples are required to be independent identically distributed.

We want to implement a learning-based MPC scheme while enforcing a strong safety requirement such that the system’s state always stays safe $\mathbf{x}(t) \in X_s, \forall t \in \mathbb{N}$, where $X_s \subseteq X$ is a safe set.

First, we find a set of safe inputs using a data-driven over-approximation of the system’s dynamics

Definition 1. *An over-approximation of the dynamics defined in (1) is a set-valued map $F : X \times U \rightrightarrows X$ that satisfies*

$$f(\mathbf{x}, \mathbf{u}) + W \subseteq F(\mathbf{x}, \mathbf{u}), \forall \mathbf{x} \in X, \forall \mathbf{u} \in U. \quad (2)$$

Section 3 is dedicated to showing how we can compute such an over-approximation. This set-valued over-approximation can then be used to find a safety controller C_F using either set-theoretic methods [5] or symbolic control [15].

Definition 2. *A safety controller C_F for the safe set X_s and the map F is a set-valued map $C_F : X \rightrightarrows U$ satisfying*

- $\text{dom}(C_F) \subseteq X_s$,
- $\forall \mathbf{x} \in \text{dom}(C_F), \forall \mathbf{u} \in C_F(\mathbf{x}), F(\mathbf{x}, \mathbf{u}) \subseteq \text{dom}(C_F)$,

where $\text{dom}(C_F) = \{\mathbf{x} \in X \mid C_F(\mathbf{x}) \neq \emptyset\}$ is the domain of C_F .

As it can be seen from the definition, safety controllers, calculated using the over-approximation, can attribute to each state $\mathbf{x} \in \text{dom}(C_F)$ a set of allowed inputs. To find the one input (out of several safe ones) that minimizes a receding horizon cost function, we build a single-valued estimation of the true function

Definition 3. An estimation $\hat{f} : X \times U \rightarrow X$ of the true function f is said to be compatible with the over-approximation F if

$$\hat{f}(\mathbf{x}, \mathbf{u}) + W \subseteq F(\mathbf{x}, \mathbf{u}), \forall \mathbf{x} \in X, \forall \mathbf{u} \in U. \quad (3)$$

Section 4 deals with finding this compatible estimation.

Now, we have everything we need to introduce the safe learning-based MPC scheme. First, we find the data-driven over-approximation F . Then, we use it to find the safe controller C_F . After that, we build an estimation of the dynamics \hat{f} compatible with the over-approximation. The following theorem shows how to use those models to implement a learning-based MPC program to meet the strict safety requirements while enforcing a soft performance optimization by minimizing the estimation of a cost function J using an estimate \hat{f} .

Theorem 1. Given a stage costs $J_k : X \times U \rightarrow \mathbb{R}$, $k \in \{1, \dots, N-1\}$ and a terminal cost $J_N : X \rightarrow \mathbb{R}$, starting from $\mathbf{x}(0) \in \text{dom}(C_F)$, consider the trajectory of (1) with $\mathbf{u}(t) = \mathbf{u}(0|t)$ where $\mathbf{u}(0|t)$ is obtained by solving the optimisation problem below:

$$\begin{aligned} & \min_{\mathbf{u}(0|t), \dots, \mathbf{u}(N-1|t)} \sum_{i=0}^{N-1} J_i(\mathbf{x}(i|t), \mathbf{u}(i|t)) + J_N(\mathbf{x}(N|t)) \\ & \text{subject to} \quad \mathbf{x}(i+1|t) = \hat{f}(\mathbf{x}(i|t), \mathbf{u}(i|t)), \\ & \quad \quad \quad \forall i \in \{0, \dots, N-1\} \\ & \quad \quad \quad \mathbf{x}(i|t) \in X_s, \forall i \in \{0, \dots, N\} \\ & \quad \quad \quad \mathbf{u}(0|t) \in C_F(\mathbf{x}(t)) \\ & \quad \quad \quad \mathbf{x}(0|t) = \mathbf{x}(t) \end{aligned} \quad (4)$$

Then, for all $t \in \mathbb{N}$, $\mathbf{x}(t) \in X_s$ and (4) admits a feasible solution, i.e. the closed-loop system is safe and well-posed.

PROOF. Let $t \in \mathbb{N}$, and let us assume that $\mathbf{x}(t) \in \text{dom}(C_F)$.

Let us first show that the optimisation problem (4) has a feasible solution. At each prediction stage $i \in \{0, \dots, N-1\}$, let us choose an input $\mathbf{u}(i|t) \in C_F(\mathbf{x}(i|t))$ then from (3),

$$\mathbf{x}(i+1|t) = \hat{f}(\mathbf{x}(i|t), \mathbf{u}(i|t)) \in F(\mathbf{x}(i|t), \mathbf{u}(i|t)).$$

Since $\mathbf{x}(0|t) = \mathbf{x}(t) \in \text{dom}(C_F)$, we have, according to the second item of Definition 2, that for all $i \in \{0, \dots, N-1\}$, $\mathbf{x}(i+1|t) \in \text{dom}(C_F)$. Then, it follows from the first item of Definition 2, that for all $i \in \{0, \dots, N\}$, $\mathbf{x}(i|t) \in X_s$ and the optimisation problem is feasible.

Then, by (4), we have that $\mathbf{u}(t) = \mathbf{u}(0|t) \in C_F(\mathbf{x}(t))$. From (2), we get that

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t) \in F(\mathbf{x}(t), \mathbf{u}(t)).$$

From the second item of Definition 2, it follows that $\mathbf{x}(t+1) \in \text{dom}(C_F)$. Then, starting from $\mathbf{x}(0) \in \text{dom}(C_F)$, we have by induction that for all $t \in \mathbb{N}$, $\mathbf{x}(t) \in \text{dom}(C_F)$ and (4) admits a feasible solution. Moreover, from the first item of Definition 2, we get that for all $t \in \mathbb{N}$, $\mathbf{x}(t) \in X_s$. \square

3. Data-Driven Over-Approximation of Unknown Functions

In this section, we introduce an algorithm to find an over-approximation of an unknown function starting from a data set of samples of this function. The algorithm finds an over-approximation under the assumption that the unknown function's derivatives and the disturbances added to the measurements are bounded. This algorithm was first introduced in our previous work [14] and is briefly presented here with the goal of making the paper self-contained. The algorithm relies on a method to over-approximate monotone functions on a fixed partition of the states and inputs spaces. Using some mathematical adjustments, we can transfer the problem at hand to a version where we have monotone functions. By doing that, we can use the aforementioned algorithm.

Given the system (1) and Assumptions 1 and 2, let us define the auxiliary matrices $A^-, A^+ \in \mathbb{R}^{n_x \times n_x}$ and $B^-, B^+ \in \mathbb{R}^{n_x \times n_u}$, where for all $i, j \in \{1, \dots, n_x\}$

$$A_{ij}^- = \begin{cases} \underline{\alpha}_{ij} & \text{if } \underline{\alpha}_{ij} < 0, \\ 0 & \text{otherwise,} \end{cases} \quad A_{ij}^+ = \begin{cases} \bar{\alpha}_{ij} & \text{if } \bar{\alpha}_{ij} > 0, \\ 0 & \text{otherwise,} \end{cases}$$

and for all $i \in \{1, \dots, n_x\}$, for all $j \in \{1, \dots, n_u\}$

$$B_{ij}^- = \begin{cases} \underline{\beta}_{ij} & \text{if } \underline{\beta}_{ij} < 0, \\ 0 & \text{otherwise,} \end{cases} \quad B_{ij}^+ = \begin{cases} \bar{\beta}_{ij} & \text{if } \bar{\beta}_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then, let the functions $f^-, f^+ : X \times U \rightarrow \mathbb{R}^{n_x}$ be defined for all $\mathbf{x} \in X$, $\mathbf{u} \in U$, by:

$$\begin{aligned} f^-(\mathbf{x}, \mathbf{u}) &= f(\mathbf{x}, \mathbf{u}) - A^- \mathbf{x} - B^- \mathbf{u}, \\ f^+(\mathbf{x}, \mathbf{u}) &= A^+ \mathbf{x} + B^+ \mathbf{u} - f(\mathbf{x}, \mathbf{u}). \end{aligned}$$

Functions f^-, f^+ are unknown but monotone since all their partial derivatives are non-negative.

Now let us define the two auxiliary data sets:

$$\begin{aligned} D^- &= \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k^-) \mid \mathbf{y}_k^- = \mathbf{x}'_k - A^- \mathbf{x}_k - B^- \mathbf{u}_k, k \in \mathbb{K}\}, \\ D^+ &= \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k^+) \mid \mathbf{y}_k^+ = A^+ \mathbf{x}_k + B^+ \mathbf{u}_k - \mathbf{x}'_k, k \in \mathbb{K}\}. \end{aligned}$$

The two auxiliary sets can be seen as if they were generated using the two monotone maps¹

$$\mathbf{y}^- \in f^-(\mathbf{x}, \mathbf{u}) + W, \quad (5)$$

$$\mathbf{y}^+ \in f^+(\mathbf{x}, \mathbf{u}) + W. \quad (6)$$

Now, we introduce the algorithm to over-approximate monotone maps. Without a loss of generality, let us assume that the sets of states and inputs are intervals; $X = [\underline{\mathbf{x}}, \bar{\mathbf{x}}]$ and $U = [\underline{\mathbf{u}}, \bar{\mathbf{u}}]$. For each coordinate $i \in \{1, \dots, n_x\}$, let be given a finite partition $(X_{q^i})_{q^i \in Q^i}$ of the interval $[\underline{x}^i, \bar{x}^i]$ where $Q^i = \{0, \dots, K^i\}$ and

$$\begin{cases} X_0^i &= [\underline{x}^i, \alpha_1^i], \\ X_{q^i}^i &= [\alpha_{q^i}^i, \alpha_{q^i+1}^i], \quad q^i = 1, \dots, K^i - 1, \\ X_{K^i}^i &= [\alpha_{K^i}^i, \bar{x}^i], \end{cases}$$

¹A definition of monotone set-valued maps can be found in [14]

where $\underline{x}^i < \alpha_1^i < \dots < \alpha_{K^i}^i < \bar{x}^i$. We define $Q = Q^1 \times \dots \times Q^{n_x}$, and let the finite rectangular partition $(X_{\mathbf{q}})_{\mathbf{q} \in Q}$ of X be given for $\mathbf{q} = (q^1, \dots, q^{n_x})$ by $X_{\mathbf{q}} = X_{q^1}^1 \times \dots \times X_{q^{n_x}}^{n_x}$. Similarly, we define the finite rectangular partition $(U_{\mathbf{p}})_{\mathbf{p} \in P}$ of U . We denote by $\underline{\mathbf{x}}_{\mathbf{q}}, \bar{\mathbf{x}}_{\mathbf{q}}, \underline{\mathbf{u}}_{\mathbf{p}}, \bar{\mathbf{u}}_{\mathbf{p}}$ the lower and upper bounds of the intervals $X_{\mathbf{q}}$ and $U_{\mathbf{p}}$.

We also consider a quantization function $\phi : X \times U \rightarrow Q \times P$ associated to the finite partitions $(X_{\mathbf{q}})_{\mathbf{q} \in Q}, (U_{\mathbf{p}})_{\mathbf{p} \in P}$ and defined as

$$\begin{aligned} \forall \mathbf{x} \in X, \forall \mathbf{u} \in U, \forall \mathbf{q} \in Q, \forall \mathbf{p} \in P, \\ \phi(\mathbf{x}, \mathbf{u}) = (\mathbf{q}, \mathbf{p}) \iff \mathbf{x} \in X_{\mathbf{q}}, \mathbf{u} \in U_{\mathbf{p}}. \end{aligned} \quad (7)$$

In what follows, we will build the over-approximation of the map defined by (5). We can build an over-approximation of (6) with similar steps. First, let us define the map $\sigma^- : Q \times P \rightrightarrows \bar{\mathbb{R}}^{n_x}$ given for all $\mathbf{q} \in Q, \mathbf{p} \in P$ by

$$\begin{aligned} \sigma^-(\mathbf{q}, \mathbf{p}) = & \left(\bigcap_{k \in \mathbb{K}^-(\underline{\mathbf{x}}_{\mathbf{q}}, \underline{\mathbf{u}}_{\mathbf{p}})} \{ \mathbf{y} \in \bar{\mathbb{R}}^{n_x} \mid \mathbf{y}_k^- + \underline{\mathbf{w}} - \bar{\mathbf{w}} \leq \mathbf{y} \} \right) \\ & \cap \left(\bigcap_{k \in \mathbb{K}^+(\bar{\mathbf{x}}_{\mathbf{q}}, \bar{\mathbf{u}}_{\mathbf{p}})} \{ \mathbf{y} \in \bar{\mathbb{R}}^{n_x} \mid \mathbf{y} \leq \mathbf{y}_k^- + \bar{\mathbf{w}} - \underline{\mathbf{w}} \} \right) \end{aligned} \quad (8)$$

where

$$\begin{aligned} \mathbb{K}^-(\mathbf{x}, \mathbf{u}) &= \{ k \in \mathbb{K} \mid \mathbf{x}_k \leq \mathbf{x}, \mathbf{u}_k \leq \mathbf{u} \}, \\ \mathbb{K}^+(\mathbf{x}, \mathbf{u}) &= \{ k \in \mathbb{K} \mid \mathbf{x} \leq \mathbf{x}_k, \mathbf{u} \leq \mathbf{u}_k \}. \end{aligned}$$

we have that σ^- is an interval-valued map; for all $\mathbf{q} \in Q, \mathbf{p} \in P, \sigma^-(\mathbf{q}, \mathbf{p}) = [\underline{\sigma}^-(\mathbf{q}, \mathbf{p}), \bar{\sigma}^-(\mathbf{q}, \mathbf{p})]$.

We denote $F^- = \sigma^- \circ \phi$. The following proposition establishes that the map F^- over-approximates the map in (5).

Proposition 1 (see [14, Theorem 2]). *Let σ and ϕ be given by (8) and (7), then:*

$$f^-(\mathbf{x}, \mathbf{u}) + W \subseteq F^-(\mathbf{x}, \mathbf{u}), \forall \mathbf{x} \in X, \forall \mathbf{u} \in U.$$

In our previous work, we established the minimality of F^- in the sense that any other interval-valued map σ' such that $f^-(\mathbf{x}, \mathbf{u}) + W \subseteq \sigma' \circ \phi(\mathbf{x}, \mathbf{u})$, for all $\mathbf{x} \in X$, for all $\mathbf{u} \in U$ will also include F^- , i.e. $F^-(\mathbf{x}, \mathbf{u}) \subseteq \sigma' \circ \phi(\mathbf{x}, \mathbf{u})$, for all $\mathbf{x} \in X$, for all $\mathbf{u} \in U$.

We build the maps σ^+, F^+ that over-approximate the monotone map (6) in the same fashion as σ^-, F^- . The following proposition shows how we use those maps to build the over-approximation of the true dynamics.

Proposition 2 (see [14, Proposition 8]). *Let $\sigma : Q \times P \rightrightarrows \bar{\mathbb{R}}^{n_x}$ be given for all $\mathbf{q} \in Q, \mathbf{p} \in P$ by:*

$$\begin{aligned} \sigma(\mathbf{q}, \mathbf{p}) = & [A^- \bar{\mathbf{x}}_{\mathbf{q}} + B^- \bar{\mathbf{u}}_{\mathbf{p}} + \underline{\sigma}^-(\mathbf{q}, \mathbf{p}), A^- \underline{\mathbf{x}}_{\mathbf{q}} + B^- \underline{\mathbf{u}}_{\mathbf{p}} + \bar{\sigma}^-(\mathbf{q}, \mathbf{p})] \\ & \cap [A^+ \underline{\mathbf{x}}_{\mathbf{q}} + B^+ \underline{\mathbf{u}}_{\mathbf{p}} - \bar{\sigma}^+(\mathbf{q}, \mathbf{p}), A^+ \bar{\mathbf{x}}_{\mathbf{q}} + B^+ \bar{\mathbf{u}}_{\mathbf{p}} - \underline{\sigma}^+(\mathbf{q}, \mathbf{p})]. \end{aligned}$$

Then, the function $F : X \times U \rightrightarrows \bar{\mathbb{R}}^{n_x}$ defined as $F = \sigma \circ \phi$ satisfies

$$\forall \mathbf{x} \in X, \mathbf{u} \in U, f(\mathbf{x}, \mathbf{u}) + W \subseteq F(\mathbf{x}, \mathbf{u}).$$

In terms of computational complexity, it has been shown in [14] that the over-approximation map F can be computed in linear time with respect to the number of data points, which makes it possible to deal with large data sets. We also refer to [14] for an experimental evaluation of the quality of the over-approximation as function of the number of data points.

The map F can be used to find a discrete state space representation of the system (symbolic abstraction), which in turn can be used to find a safety controller using an iterative algorithm.

4. Piecewise Multi-Affine Compatible Estimation

In this section, we demonstrate how to build a single-valued piecewise compatible estimation of the system's dynamics using the class of multi-affine functions. This class of functions was studied on n -dimensional intervals in [2]. We make use of this study to build a piecewise multi-affine estimation compatible with the over-approximation introduced in the previous section. The piecewise estimation is defined on the same partition used to build the over-approximation.

4.1. Piecewise multi-affine functions

Now, we introduce the class of piecewise multi-affine functions that we will use to estimate the unknown function. In this section and to simplify the notations, we will note $Z = X \times U$ and $\mathbf{z} = (\mathbf{x}, \mathbf{u})$.

Definition 4. *A multi-affine function $g : Z \rightarrow \mathbb{R}^m, Z \subseteq \mathbb{R}^n$ is a function of the form*

$$g(z^1, \dots, z^n) = \sum_{i_1, \dots, i_n \in \{0,1\}} c_{i_1, \dots, i_n} (z^1)^{i_1} \dots (z^n)^{i_n} \quad (9)$$

where $c_{i_1, \dots, i_n} \in \mathbb{R}^m$ for all $i_1, \dots, i_n \in \{0, 1\}$

In the case where Z is an interval; $Z = [\underline{\mathbf{z}}, \bar{\mathbf{z}}], \underline{\mathbf{z}}, \bar{\mathbf{z}} \in \mathbb{R}^n$, we denote the set of vertices of this interval by

$$V = \prod_{i=1}^n \{ \underline{z}^i, \bar{z}^i \}.$$

Let $\xi_i : \{ \underline{z}^i, \bar{z}^i \} \rightarrow \{0, 1\}$ for all $i \in \{1, \dots, n\}$ denote the indicator function

$$\xi_i(\underline{z}^i) = 0 \quad \xi_i(\bar{z}^i) = 1 \quad \forall i \in \{1, \dots, n\}.$$

The following proposition states that any multi-affine function defined on an interval can be written as a linear combination of the values of the function on the vertices of this interval.

Proposition 3 (see [2, Proposition 1]). *Let Z be an n -dimensional interval, $g : Z \rightarrow \mathbb{R}^m$ a multi-affine function such that, for all $\mathbf{v} = (v^1, \dots, v^n) \in V$ we have $g(v^1, \dots, v^n) = \mathbf{y}_v$. Then, for all $\mathbf{z} = (z^1, \dots, z^n) \in Z$ the function g is uniquely given by*

$$g(\mathbf{z}) = \sum_{\mathbf{v} \in V} \prod_{i=1}^n \left(\frac{z^i - \underline{z}^i}{\bar{z}^i - \underline{z}^i} \right)^{\xi_i(v^i)} \left(\frac{\bar{z}^i - z^i}{\bar{z}^i - \underline{z}^i} \right)^{1-\xi_i(v^i)} \mathbf{y}_v. \quad (10)$$

As a consequence of Proposition 3, we can estimate a multi-affine function on a given interval by estimating the function's values on the vertices of the interval.

Lemma 2 (see [2, Lemma 2]). *Let $s \in \mathbb{R}^m$ and $d \in \mathbb{R}$. Then, $s^T g(\mathbf{z}) \bowtie d$ for all $\mathbf{z} \in Z$ if and only if $s^T g(\mathbf{v}) \bowtie d$, for all $\mathbf{v} \in V$, where \bowtie stands for any of $<, \leq, =, \geq, >$.*

Given a partition $(Z_{\mathbf{r}})_{\mathbf{r} \in R}$ of the interval $Z \subseteq \mathbb{R}^n$, we denote the vertices of an interval $Z_{\mathbf{r}}$ by $V_{\mathbf{r}}$. A function $g : Z \rightarrow \mathbb{R}^m$ is piecewise multi-affine if for all $\mathbf{r} \in R$ the function is multi-affine on $Z_{\mathbf{r}}$.

The following proposition establishes that if a piecewise multi-affine function is continuous on the grid points of the partition, then it is continuous everywhere

Proposition 4. *If a piecewise multi-affine function $g : Z \rightarrow \mathbb{R}^m$ is continuous on the grid points of the partition $(Z_{\mathbf{r}})_{\mathbf{r} \in R}$:*

$$\lim_{\mathbf{z} \rightarrow \mathbf{v}} g(\mathbf{z}) = g(\mathbf{v}) \quad \forall \mathbf{r} \in R, \mathbf{v} \in V_{\mathbf{r}}$$

then g is continuous for all $\mathbf{z} \in Z$.

PROOF. To establish the continuity of piecewise multi-affine functions, it is sufficient to study them on the shared facets of neighbouring cells in the partition because they are defined as multi-affine functions on the interior of those cells; and multi-affine functions are continuous. The facets of an interval $Z_{\mathbf{r}} = [\underline{\mathbf{z}}_{\mathbf{r}}, \bar{\mathbf{z}}_{\mathbf{r}}]$, $\underline{\mathbf{z}}_{\mathbf{r}}, \bar{\mathbf{z}}_{\mathbf{r}} \in \mathbb{R}^n$ are given by

$$E_{Z_{\mathbf{r}}, w^j} = Z_{\mathbf{r}} \cap \{ \mathbf{z} \in \mathbb{R}^n \mid z^j = w^j \}$$

for all $w^j \in \{ \underline{z}^j, \bar{z}^j \}$, $j \in \{1, \dots, n\}$ which implies according to (9) that $g(z^1, \dots, w^j, \dots, z^n)$ is also a multi-affine function.

We also have that the facets of the n -dimensional interval are $(n-1)$ -dimensional intervals, which implies that the expression of the function g on a given facet is of the form (10). Therefore, as the shared facets of two neighbouring intervals in the partition have the same vertices, then the limit of the function g from the two neighbouring intervals on the shared facet will be the same. \square

Based on this result, we can see that to estimate a continuous piecewise multi-affine function on a given partition, we only need to estimate its values on the vertices of the partition.

4.2. Compatible estimation

Now we will describe how we build a piecewise multi-affine estimation function \hat{f} of the system's dynamics compatible with the over-approximation calculated in the previous section.

Starting from the given finite rectangular partitions $(X_{\mathbf{q}})_{\mathbf{q} \in Q}$, $(U_{\mathbf{p}})_{\mathbf{p} \in P}$ of X and U , and Assumption 2, each transition triple $(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}'_k)$, $k \in \mathbb{K}$ allows us to write the following equation

$$\mathbf{x}'_k = \hat{f}_{\mathbf{q}, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{e}(\mathbf{x}_k, \mathbf{u}_k)$$

where $(\mathbf{q}, \mathbf{p}) = \phi(\mathbf{x}_k, \mathbf{u}_k)$. The function $\hat{f}_{\mathbf{q}, \mathbf{p}} : X_{\mathbf{q}} \times U_{\mathbf{p}} \rightarrow X$ is the multi-affine estimation of the true function on the interval $X_{\mathbf{q}} \times U_{\mathbf{p}}$, and the vector $\mathbf{e}(\mathbf{x}_k, \mathbf{u}_k)$ represents the residuals of the estimation.

According to (10), we can rewrite \hat{f} as a linear combination of the estimated function on the vertices of the interval $\hat{\mathbf{y}}_{\mathbf{v}_1}, \dots, \hat{\mathbf{y}}_{\mathbf{v}_{2^{n_x+n_u}}}$, where $\mathbf{v}_1, \dots, \mathbf{v}_{2^{n_x+n_u}} \in V_{\mathbf{q}, \mathbf{p}}$ are the vertices of the interval $X_{\mathbf{q}} \times U_{\mathbf{p}}$

$$\hat{f}_{\mathbf{q}, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k) = \sum_{\mathbf{v} \in V_{\mathbf{q}, \mathbf{p}}} \gamma_{\mathbf{v}}(\mathbf{x}_k, \mathbf{u}_k) \hat{\mathbf{y}}_{\mathbf{v}}$$

where $\gamma_{\mathbf{v}}$ represent the coefficients of the linear combination given by (10). Therefore, we can write the estimation problem to calculate the piecewise multi-affine estimation function \hat{f} from the data set in a matrix form.

We first denote the set of all vertices of the partition

$$\mathcal{V} = \bigcup_{\mathbf{q} \in Q, \mathbf{p} \in P} V_{\mathbf{q}, \mathbf{p}}$$

The set \mathcal{V} is finite and thus can be numbered $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_K\}$, where K is the number of grid's points. For every $j \in \{1, \dots, n_x\}$, We aggregate all variables representing the values of \hat{f}^j on the grid's points of the partition ($\hat{\mathbf{y}}_{\mathbf{v}}^j$, for all $\mathbf{v} \in \mathcal{V}$) in a single vector $\Phi_j \in \mathbb{R}^K$. Then, the regression problem for every component \hat{f}^j is

$$\chi_j = A \cdot \Phi_j + \mathbf{E}_j \quad (11)$$

where $\chi_j \in \mathbb{R}^{|\mathbb{K}|}$ is a vector aggregating the j components of the data points' transitions $\chi_j = (x_1^j, \dots, x_{|\mathbb{K}|}^j)$, $\mathbf{E}_j \in \mathbb{R}^{|\mathbb{K}|}$ the vector of residuals, and $A \in \mathbb{R}^{|\mathbb{K}| \times K}$ is the sparse coefficients matrix. Each row of this matrix is built using a data point and contains $2^{n_x+n_u}$ entries which is the number of vertices of the interval $X_{\mathbf{q}} \times U_{\mathbf{p}}$ to which the data point belongs. The values of the entries at each row of the matrix are the coefficients of the multi-affine function defined on this interval as seen in (10).

We use the least squares estimator to find the values of Φ_j . The cost function, which is the sum of the squares of residuals, can be written as

$$\begin{aligned} S(\Phi_j) &= \mathbf{E}_j^T \mathbf{E}_j = (\chi_j - A \cdot \Phi_j)^T (\chi_j - A \cdot \Phi_j) \\ &= \chi_j^T \chi_j - 2 \Phi_j^T A^T \chi_j + \Phi_j^T A^T A \Phi_j \end{aligned}$$

Hence, the estimation problem can be expressed as a sparse quadratic optimisation problem.

Finally, let us define the two vectors $\underline{\Phi}_j, \overline{\Phi}_j \in \mathbb{R}^K$. For all the component $i \in \{1, \dots, K\}$, we define $\underline{\Phi}_j^i, \overline{\Phi}_j^i$

$$\begin{aligned}\underline{\Phi}_j^i &= \max_{\mathbf{q}', \mathbf{p}'} \{ \underline{\sigma}^j(\mathbf{q}', \mathbf{p}') \mid \mathbf{v}_i \in V_{\mathbf{q}', \mathbf{p}'} \} - \underline{w}^j \\ \overline{\Phi}_j^i &= \min_{\mathbf{q}', \mathbf{p}'} \{ \overline{\sigma}^j(\mathbf{q}', \mathbf{p}') \mid \mathbf{v}_i \in V_{\mathbf{q}', \mathbf{p}'} \} - \overline{w}^j\end{aligned}$$

which resemble the minimum of over-approximation's upper bound and the maximum of over-approximation's lower bound for all the cells that \mathbf{v}_i is a vertex of.

The following proposition gives the sufficient condition so that the estimated piecewise multi-affine function is compatible with the over-approximation built using the same partition and assumptions.

Proposition 5. *Given the finite rectangular partitions $(X_q)_{q \in Q}$, $(U_p)_{p \in P}$ of X and U , and under Assumption 2, the piecewise multi-affine estimation function \hat{f} whose components are calculated using the following optimisation problem, for every $j \in \{1, \dots, n_x\}$,*

$$\begin{aligned}\min_{\Phi_j} \quad & \Phi_j^T A^T A \Phi_j - 2 \Phi_j^T A^T \chi_j \\ \text{subject to} \quad & \underline{\Phi}_j \leq \Phi_j \leq \overline{\Phi}_j\end{aligned}\tag{12}$$

is compatible with the over-approximation F , i.e. $\hat{f}(\mathbf{x}, \mathbf{u}) + W \subseteq F(\mathbf{x}, \mathbf{u})$, for all $\mathbf{x} \in X$, for all $\mathbf{u} \in U$.

PROOF. First, let us show that the optimisation problem (12) has a feasible solution. According to Proposition 2 we have that $\forall \mathbf{x} \in X, \mathbf{u} \in U, f(\mathbf{x}, \mathbf{u}) + W \subseteq F(\mathbf{x}, \mathbf{u})$. Therefore, for every $j \in \{1, \dots, n_x\}$ the values of the f^j on the grid's points $y_v^j = f^j(\mathbf{v})$, for all $\mathbf{v} \in V_{\mathbf{q}, \mathbf{p}}$ for all $\mathbf{q} \in Q, \mathbf{p} \in P$ are a feasible solution for the optimisation problem (12).

From (12), we have that for a given cell $X_q \times U_p$, all the resulting values of the estimation are included in the over-approximation map, i.e. $y_v^{*j} \leq \overline{\sigma}^j(\mathbf{q}, \mathbf{p}) - \overline{w}^j$ and $y_v^{*j} \geq \underline{\sigma}^j(\mathbf{q}, \mathbf{p}) - \underline{w}^j$ for all $\mathbf{v} \in V_{\mathbf{q}, \mathbf{p}}$ for all $\mathbf{q} \in Q, \mathbf{p} \in P$. This implies according to Lemma 2 (by choosing $\mathbf{s} = \mathbf{e}_j$) that $\hat{f}^j(\mathbf{x}, \mathbf{u}) \leq \overline{\sigma}^j(\mathbf{q}, \mathbf{p}) - \overline{w}^j$ and $\hat{f}^j(\mathbf{x}, \mathbf{u}) \geq \underline{\sigma}^j(\mathbf{q}, \mathbf{p}) - \underline{w}^j$ for all $\mathbf{x} \in [\underline{x}_q, \overline{x}_q]$ for all $\mathbf{u} \in [\underline{u}_p, \overline{u}_p]$. Hence, $\hat{f}(\mathbf{x}, \mathbf{u}) + W \subseteq F(\mathbf{x}, \mathbf{u})$, for all $\mathbf{x} \in X$, for all $\mathbf{u} \in U$. \square

Let us remark that $A^T A \in \mathbb{R}^{K \times K}$ and $A^T \chi_j \in \mathbb{R}^K$, which makes the size of the quadratic program (12) independent of the number of data points. These products of matrices can be computed in linear time with respect to the number of data points. Moreover, due to the sparsity of $A \in \mathbb{R}^{|\mathbb{K}| \times K}$, these computations can be done efficiently. It follows that large data sets can be handled in practice.

Remark 2. The estimated function \hat{f} is continuous and differentiable almost everywhere, making the MPC optimisation problem in (4) solvable using subgradient descent methods.

5. Case Study

To showcase the validity of our approach, we present in this section a path planning problem. We learn the dynamics of a unicycle model by sampling data transitions from it. Then, we use the learned model to drive the vehicle in an environment containing obstacles. To do this, we use the two-model approach described in the paper. We show how the approach can offer a robust safety guarantee while moving optimally towards a predefined goal.

We consider the unicycle models defined by the following equations

$$\begin{aligned}\dot{x} &= v \cdot \cos(\theta) + w_1 \\ \dot{y} &= v \cdot \sin(\theta) + w_2 \\ \dot{\theta} &= \frac{v}{L} \tan(\delta) + w_3\end{aligned}\tag{13}$$

where x, y are the coordinates of the vehicle, θ is the heading angle, as can be seen in Figure 1, L is the length of the vehicle, and we chose the value $L = 0.1$ m. The velocity v and the steering angle δ are considered the input. The goal of this experiment is to drive the vehicle from a starting position to a target position in an environment, as shown in Figure 2. The vehicle should manoeuvre around an obstacle to reach its target. We consider a 2×5 m area, a 1×0.5 m obstacle positioned at 2.5 m away from the left side of the room.

To build the over-approximation of the dynamics in (13), we sampled $|\mathbb{K}| = 10^6$ data points using a black box simulator of (13). We chose for the states, inputs, and disturbance intervals the following $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\delta \in [-1, 1]$, $v \in [0.025, 0.5]$, $\mathbf{w} \in [-0.05, 0.05] \times [-0.05, 0.05] \times [-0.05, 0.05]$, and partitioned those intervals uniformly into 30 cells each. We only use θ, δ , and v to study the dynamics of the unicycle model as the function representing this model clearly does not depend on the states x and y . The resulting

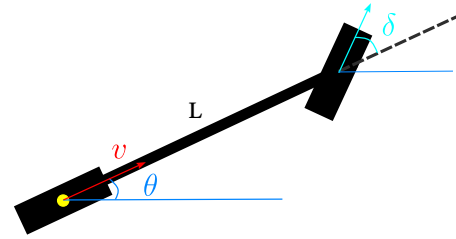


Figure 1: The unicycle model. The reference point is at the center of the rear axle.

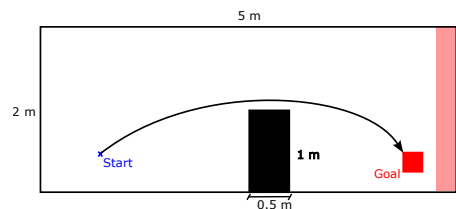


Figure 2: The environment where the vehicle should manoeuvre an obstacle to reach a target position

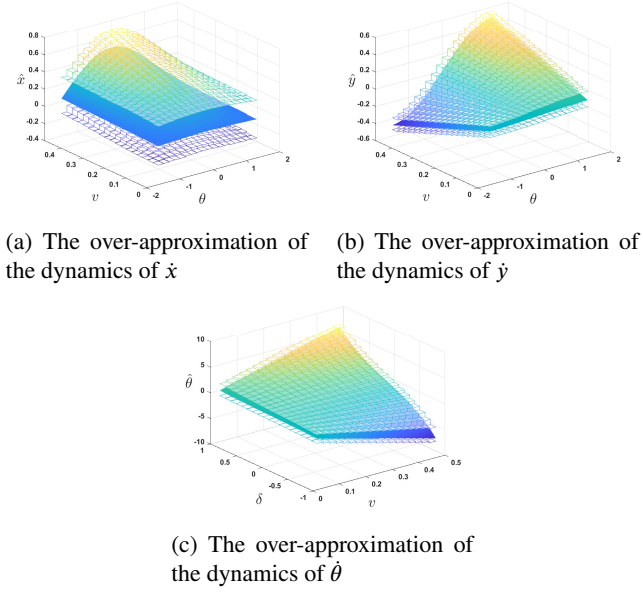


Figure 3: Over-approximation of the bicycle dynamics

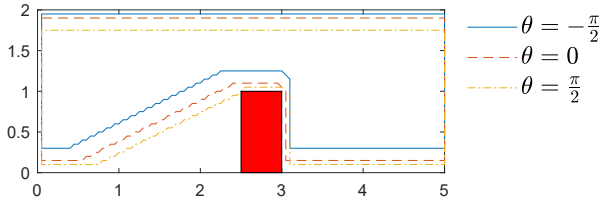


Figure 4: The maximal controlled invariant for three values of θ .

over-approximation can be seen in Figure 3. The execution time to find the over-approximation is $t_{ov} = 2.78$ s.

Then, we use the over-approximation model to build a symbolic abstraction of the behaviour of the vehicle in the environment as follows: We partitioned the x -axis into 100 sections and the y -axis into 40 sections uniformly. Then, we found the reachable set of the vehicle starting from each cell in the partition using the calculated over-approximation. We use the Euler method for discretizing. The symbolic abstraction is then used to find the maximal control invariant inside the safe region and the set of safe actions at each cell. The speed is chosen to be strictly positive, and hence the vehicle cannot stop. All the walls of the area are considered obstacles except the right wall because the vehicle cannot stop. Figure 4 shows the regions where we can find a safety controller for certain values of θ . The execution time to find the maximal controlled invariant is $t_{inv} = 871.88$ s.

After that, we calculated the piecewise multi-affine estimation of the dynamics according to the algorithm introduced in Section 4. Figure 5 shows the result of estimation for the first component of the dynamical model and the difference between the true function and the estimation on the grid's points. The execution time to find the piecewise multi-affine estimation is $t_{est} = 31.48$ s.

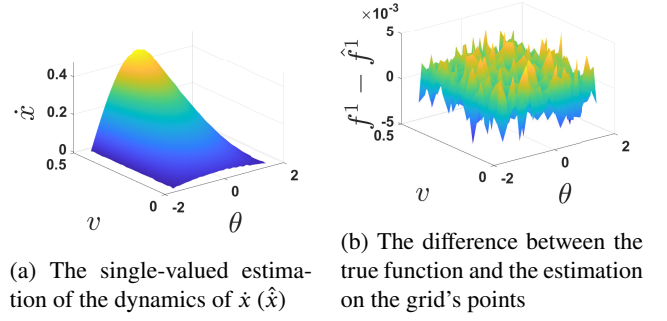


Figure 5: Estimation with piecewise multi-affine functions

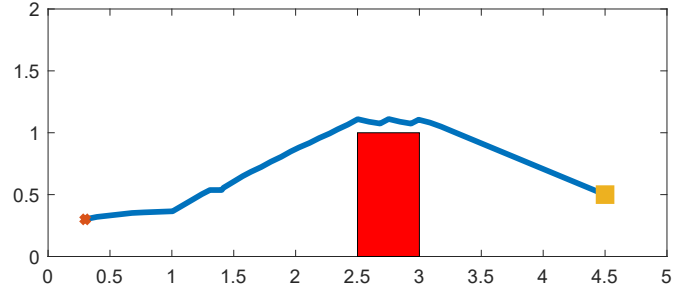


Figure 6: Trajectory of the vehicle

Finally, the calculated estimation and safety controller are used to find a feasible trajectory to reach the target position while avoiding the obstacle. We chose a starting position $\mathbf{x}_0 = (0.3, 0.3)$ and the target $\mathbf{x}_f = (4.5, 0.5)$. The cost function

$$J = \sum_{i=1}^N \|\mathbf{x}(i|t) - \mathbf{x}_f\| \quad (14)$$

was chosen to drive the vehicle to the target. The trajectory is shown in Figure 6, where it can be seen that the vehicle reached the target position while avoiding the obstacle. Although we chose a simple sum-of-distances cost function (14), we were able to reach the goal while avoiding the obstacle.

6. Conclusion

In this paper, a novel approach to safe learning-based MPC was introduced. The approach depends on finding two models representing the system. The first is an over-approximation of the system's dynamics and is used to find the set of safe inputs at each state. The second is a piecewise multi-affine estimation used to ensure the performance requirements. A compatibility condition between models ensures safety and well-posedness. A case study was explored to show how we can use the proposed approach to navigate a vehicle in an environment while avoiding obstacles. In future, a prospect of this work could be to investigate more complex scenario and the possibility of updating the learned models online.

References

- [1] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.
- [2] C. Belta and L. C. G. J. M. Habets. Controlling a class of nonlinear systems on rectangles. *IEEE Transactions on Automatic Control*, 51(11):1749–1759, Nov. 2006.
- [3] C. Belta, B. Yordanov, and E. A. Gol. *Formal methods for discrete-time dynamical systems*. Springer, 2017.
- [4] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4):1702–1717, 2020.
- [5] F. Blanchini and S. Miani. *Set-theoretic methods in control*. Springer, 2008.
- [6] M. Canale, L. Fagiano, and M. Signorile. Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control*, 24(1):123–139, 2014.
- [7] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the deepc. In *European Control Conference*, pages 307–312, 2019.
- [8] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.
- [9] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370, 2019.
- [10] M. Kazemi, R. Majumdar, M. Salamati, S. Soudjani, and B. Wooding. Data-driven abstraction-based control synthesis. *arXiv preprint arXiv:2206.08069*, 2022.
- [11] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. Learning-based model predictive control for safe exploration. In *IEEE Conference on Decision and Control*, pages 6059–6066, 2018.
- [12] V. Krishnan and F. Pasqualetti. On direct vs indirect data-driven predictive control. In *IEEE Conference on Decision and Control*, pages 736–741, 2021.
- [13] A. Lavaei and E. Frazzoli. Data-driven synthesis of symbolic abstractions with guaranteed confidence. *IEEE Control Systems Letters*, 7:253–258, 2022.
- [14] A. Makdesi, A. Girard, and L. Fribourg. *Data-Driven Models of Monotone Systems*. 2022.
- [15] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
- [16] X. Zhang, J. Liu, X. Xu, S. Yu, and H. Chen. Robust learning-based predictive control for discrete-time nonlinear systems with unknown dynamics and state constraints. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(12):7314–7327, 2022.