Affine Abstraction of Nonlinear Systems

with Applications to Active Model Discrimination

by

Kanishka Raj Singh

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2018 by the
Graduate Supervisory Committee:

Sze Zheng Yong, Chair
Panagiotis Artemiadis
Spring Berman

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

This work considers the design of *separating* input signals in order to discriminate among a finite number of uncertain nonlinear models. Each nonlinear model corresponds to a system operating mode, unobserved intents of other drivers or robots, or to fault types or attack strategies, etc., and the separating inputs are designed such that the output trajectories of all the nonlinear models are guaranteed to be distinguishable from each other under any realization of uncertainties in the initial condition, model discrepancies or noise. I propose a two-step approach. First, using an optimization-based approach, we over-approximate nonlinear dynamics by uncertain affine models, as abstractions that preserve all its system behaviors such that any discrimination guarantees for the affine abstraction also hold for the original nonlinear system. Then, I propose a novel solution in the form of a mixed-integer linear program (MILP) to the active model discrimination problem for uncertain affine models, which includes the affine abstraction and thus, the nonlinear models. Finally, I demonstrate the effectiveness of our approach for identifying the intention of other vehicles in a highway lane changing scenario.

For the abstraction, I explore two approaches. In the first approach, I construct the bounding planes using a Mixed-Integer Nonlinear Problem (MINLP) formulation of the given system with appropriately designed constraints. For the second approach,I solve a linear programming (LP) problem that over-approximates the nonlinear function at only the grid points of a mesh with a given resolution and then accounting for the entire domain via an appropriate correction term. To achieve a desired approximation accuracy, we also iteratively subdivide the domain into subregions. This method applies to nonlinear functions with different degrees of smoothness, including Lipschitz continuous functions, and improves on existing approaches by enabling the use of tighter bounds. Finally, we compare the effectiveness of this approach with existing optimization-based methods in simulation and illustrate its applicability for estimator design.

ACKNOWLEDGEMENTS

I want to thank Dr. Yong for giving me an opportunity to work on this project and for being an excellent guide. It has been a great learning experience for me and has left me with a sense of achievement, learning and excitement about many new avenues to be explored. I want to thank Dr. Necmiye Ozay and Yuhao Ding from University of Michigan for their collaboration. I want to thank Dr. Artemiadis and Dr. Berman for being on my committee and being so prompt with any help I needed throughout this project. I also want to thank Arizona State University, especially Dr. Hyunglae Lee and Dr. Artemiadis for their excellence in teaching and for encouraging real learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

Recently, there is much public interest in the integration of smart systems into everyday lives. These systems that include smart homes, smart grids, intelligent transportation and smart cities, are essentially complex, integrated and interconnected engineered systems with multiple operating modes that are often not directly observed or measured; thus, they can be modeled as hidden mode hybrid systems. For example, autonomous vehicles/robots have no access to the intentions or decisions of other vehicles or humans [32, 38, 11], while smart infrastructures are prone to different fault types [17, 8] or attack modes [29, 39, 19]. In these scenarios, approaches for discriminating among these operating modes (or more generally, models of system behaviors) based on noisy observed measurements can have a significant impact on a broad range of applications in robotics, process control, medical devices, fault detection, etc. This is an important problem in statistics, machine learning and systems theory; thus, general techniques for model discrimination can have a significant impact on a broad range of applications. problems in robotics, process control, medical devices, fault detection, etc.

In particular, the transition to autonomy in passenger and commercial vehicles is happening at an incredible pace. Despite big technological breakthroughs in recent years, a major challenge in autonomous driving that remains is the capability of these autonomous vehicles to intelligently and safely identify and react to the behaviors of human-driven cars and other autonomous vehicles.

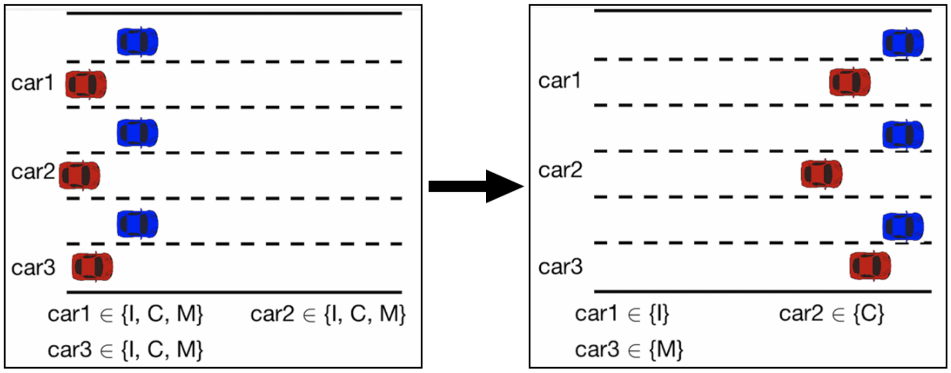

Figure 1.1: Illustration of active intention identification, where separating inputs help to distinguish Inattentive, Cautious and Malicious intentions [11].

The main focus of my thesis can be explained with Figure 1.1. In this scenario we have two cars - ego car (blue) and the other car (red) in a highway lane changing scenario. The ego car wants to move from its current lane to the other car's lane. My goal is to discern the intention of the other car (inattentive, cautious or malicious) by designing the ego car's input in such a way that at the end of a pre-determined time horizon, only one of the three intentions can be consistent with the other car's dynamics.

In the first part of Figure 1.1 the initial conditions - position, velocity etc. - for all three intentions is the same. In the second part we can see that the ego car moves in the same manner, but the other car reacts in a different way in all cases. Car 1 is not aware of the ego car and thus is coasting. Car 2 is cautious and yields the lane in order for the ego car to change its lane. Car 3 is Malicious and tries to match the position of the ego car as it comes into its lane.

Since the dynamics of these smart systems are almost always complex (nonlinear or hybrid), it is desirable to compute a simpler conservative approximation or abstraction of the system dynamics while preserving the dynamical characteristics of the original systems. Abstraction-based methods for analyzing and controlling smart systems

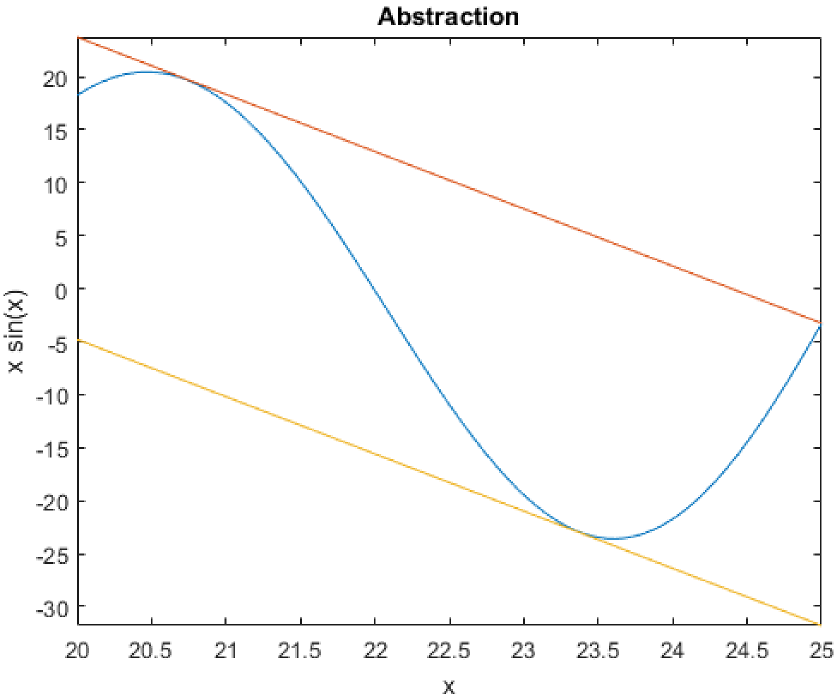

Figure 1.2: Illustration of Affine Abstraction/Overapproximation.

have recently attracted a great deal of interest [37].

Using these approximate systems, controllers that are correct-by-construction with respect to reachability and safety specifications can be synthesized efficiently, see e.g., [2, 13, 1], and similarly, guarantees for estimator designs also apply to the original complex systems [35].

Figure 1.2 is an illustration of abstraction in a 1-dimensional nonlinear system. Here, two lines are constructed in such a way that the original (nonlinear) system is completely contained within these lines for the concerned domain and the gap between the two lines is minimum. This is extended to higher dimensions for a n-dimensional nonlinear function by designing two n-dimensional affine hyperplanes that contain the original system. The active model discrimination algorithm is designed in such a way that it is applicable for everything within the bounds of the affine planes, thus being applicable to the original system as well.

1.2 Literature Review

The problem of discriminating among a set of models appears in a plethora of research areas such as fault detection, input-distinguishability and mode discernibility of hybrid systems, where the approaches in the literature can be grouped into passive and active methods. Passive discrimination techniques seek the separation of the models regardless of the input [25, 31, 38, 21], while active methods design a separating input such that the behaviors of different models are distinct.

Specifically in the area of input design for active model discrimination, many approaches have been proposed with the goal of finding a small excitation that has a minimal effect on the desired behavior of the system, while guaranteeing the isolation of different fault models [8, 34, 28, 33, 19, 11]. However, these methods are only applicable for known linear or affine models, and not for nonlinear or uncertain affine models that we consider in this work.

A passive fault detection and isolation approach first computes a time horizon length that is sufficient for guaranteeing the distinguishability of multiple faults represented by switched affine models (also known as the T-detectability problem). Then, a model invalidation approach is proposed to detect and isolate different fault models in real-time by only observing the output sequence for a finite time horizon without compromising detection/isolation guarantees. The problem of mode discernibility in switched autonomous linear models is also studied in [6], where discernibility of the linear models is considered for almost every initial condition.

Passive methods offer stronger conditions for distinguishability of the behaviors of multiple models in the sense that they guarantee separation of output trajectories for any input applied to the models. This, however, comes at a cost, which is the conservativeness introduced by these methods. In other words, the applicability of guaranteed passive methods is somewhat limited, because not all models are passively distinguishable. This is the motivation behind the introduction of active methods.

Active model discrimination is also considered in hybrid system community, where the concept of controlled-discernibility is introduced, which seeks for an input to discriminate among different mode sequences in switched linear models The concept of input-distinguishability for a pair of linear models is discussed in [25, 31], which is the passive version of distinguishability that is defined in [14]. The problem of model-based active fault detection is also extensively studied, where the goal is to find a small excitation that has a minimal effect on the desired behavior of the system, while guaranteeing the isolation of different fault models [8, 28, 33]. On the other hand, a computational method for passively discriminating among fault models is given in [18, 21]. The problem of mode discernibility in switched autonomous linear models in the passive setting is also studied in [6], and in addition, they introduced the concept of controlled-discernibility, which seeks for an input to discriminate among different mode sequences in switched linear models.

In [6], controlled-discernibility is introduced, which seeks for an input to discriminate among different mode sequences in switched linear models. Moreover, in [33], an active fault detection approach is considered to isolate multiple linear time-varying models subject to uncertainties and noise. In this approach, a bi-level optimization formulation is posed as a Mixed-Integer Quadratic Program (MIQP) that is then used to search for optimal separating inputs.

Specifically in the area of intention identification, passive methods have been investigated in [38, 40, 27, 22] to estimate human behavior or intent, and the obtained intention estimates are then used for control. The problem of intention identification was also considered for inter-vehicle applications, where a partially observable Markov decision process (POMDP) framework was proposed to estimate the driver's intention

[23].

Another set of relevant literature pertains to abstractions of nonlinear systems as linear or affine models that over-approximates all possible original system behaviors, which is a common systematic approximation approach in the literature on hybridization [4, 2, 10, 1]. The abstraction process typically involves partitioning the state space of the original system into a finite number of regions and approximating its dynamics locally in each region by a simpler dynamics, which is possibly conservative affine or polynomial approximations of the analyzed system [13]. When the system state moves from one region to another, the dynamics of the approximate system also switches accordingly. That is to say, the approximate systems behaves like a hybrid system and thus, the abstraction is also referred to as a hybridization process [4]. In [3], abstraction of nonlinear systems as piecewise linear systems over a mesh with a fixed partition was studied. However, this evenly-sized partition of the domain of interest may not be computationally tractable as it requires a large number of discrete states/modes to make the partition size sufficiently fine. To reduce the number of subregions, the Lebesgue piecewise affine approximation was proposed for a class of nonlinear Lipschitz continuous functions in [5], where the partition of the state space depends on the variation of the vector field. On the other hand, on-the-fly abstraction is a dynamic method where the domain construction and the abstraction process are only carried out on states that are reachable [16, 2]. Although this method scales better into high dimensions, some drawbacks, such as error accumulation and the splitting of the currently-tracked set of states along multiple facets, still exist [7].

Moreover, abstractions for each partition are typically obtained by linear interpolation over a given region and adding the corresponding interpolation error to the simpler dynamics as bounded inputs [4, 30]. Hence, a set of relevant literature pertains to the analysis of interpolation error bounds. The size of the error bounds is important as it affects not only the approximation precision but also the computation time. In [36], optimal estimates for approximation errors in linear interpolation of functions with several degrees of smoothness were developed, while [10] presented a coordinate transformation to get a tighter interpolation error bound.

1.3 Contribution

We propose a novel two-step approach to active model discrimination among a set of uncertain nonlinear models, consisting of the affine abstraction/ over-approximation and the corresponding input design problem. This problem is relatively unexplored in the literature to the best of my knowledge.

In the first step, we propose optimization-based approaches to over-approximate nonlinear dynamics by uncertain affine models that are compactly described using interval-valued matrices/vectors, in contrast to only having a interval-valued affine vector. In particular, this uncertain affine model must preserve all the system behaviors of the original nonlinear dynamics such that any model discrimination guarantees for uncertain affine abstraction also hold for the original nonlinear models.

In addition, we develop a mesh-based method for piecewise affine abstraction, which over-approximates the nonlinear behaviors over an entire mesh as opposed to over each simplex/mesh element, thus our approach results in less complex abstractions that can simplify reachability analysis.

The novelty of our mesh-based approach lies in solving a linear programming (LP) optimization that over-approximates the nonlinear function at only the grid points of a mesh with a given resolution and then accounting for the entire domain in the interior of the mesh via an appropriate correction term. The proposed abstraction algorithm can also obtain an arbitrarily precise approximation of a nonlinear function at the price of increasing the mesh resolution, hence the size of the LP and its computational complexity.

Comparing with a recent abstraction method for Lipschitz continuous functions in [1], our method can apply to nonlinear functions with different degrees of smoothness including Lipschitz continuous functions. In addition, our analysis is based on mesh elements (in contrast to point-wise analysis in [1]) and this enables the use of tighter error bounds based on linear interpolation in [36, 10]. Therefore, the abstraction efficiency is improved and the number of subregions is reduced for the same desired approximation accuracy.

Next, in the second step, using the resulting set of uncertain affine models from the first step, we propose a novel solution to the active model discrimination problem for uncertain affine models, which includes the affine abstraction. We show that this problem can be casted as a mixed-integer linear program (MILP), for which off-the-shelf optimization tools are readily available. To my knowledge, this input design problem for uncertain affine models (an important problem on its own) has also not been considered in the literature.

Finally, using simulation examples, we demonstrate the advantages of the proposed approaches for affine abstraction over the existing optimization-based approach in [1] as well as illustrate the usefulness of the obtained abstraction for estimator design, specifically the active model discrimination problem in the context of identifying the intention of other vehicles in a lane changing scenario.

1.4 Notation and Definitions

Let $x \in \mathbb{R}^n$ denote a vector and $M \in \mathbb{R}^{n \times m}$ a matrix, with transpose $M^\intercal$ and $M \geq 0$ denotes element-wise non-negativity. The vector norm of x is denoted by $\|x\|_i$ with $i \in \{1, 2, \infty\}$, while $\mathbf{0}$, $\mathbb{1}$ and $\mathbb{I}$ represent the vector of zeros, the vector of ones and the identity matrix of appropriate dimensions.

The diag and vec operators are defined for a collection of matrices $M_i, i = 1, \ldots, n$ and matrix M as:

$$\begin{aligned}
&\mathrm{diag}_{i=1}^{n}\{M_i\} = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_n \end{bmatrix}, \ \mathrm{vec}_{i=1}^{n}\{M_i\} = \begin{bmatrix} M_1 \\ \vdots \\ M_n \end{bmatrix}, \\
&\mathrm{diag}_{i,j}\{M_k\} = \begin{bmatrix} M_i & \mathbf{0} \\ \mathbf{0} & M_j \end{bmatrix}, \qquad \mathrm{vec}_{i,j}\{M_k\} = \begin{bmatrix} M_i \\ M_j \end{bmatrix}, \\
&\mathrm{diag}_N\{M\} = \mathbb{I}_N \otimes M, \qquad \mathrm{vec}_N\{M\} = \mathbb{1}_N \otimes M,
\end{aligned}$$

where $\otimes$ is the Kronecker product.

The set of positive integers up to n is denoted by $\mathbb{Z}_n^+$, and the set of non-negative integers up to n is denoted by $\mathbb{Z}_n^0$. We will also make use of Special Ordered Set of degree 1 (SOS-1) constraints[1] in our optimization formulations, defined as:

Definition 1 (SOS-1 Constraints). *A special ordered set of degree 1 (SOS-1) constraint [e.g., [15]] is a set of integer, continuous or mixed-integer scalar variables for which at most one variable in the set may take a value other than zero, denoted as SOS-1: $\{v_1, \ldots, v_N\}$. For instance, if $v_i \neq 0$, then this constraint imposes that $v_j = 0$ for all $j \neq i$.*

[1] Off-the-shelf solvers such as Gurobi and CPLEX [15, 9] can readily handle these constraints, which can significantly reduce the search space for integer variables in branch and bound algorithms.

Chapter 2

AFFINE ABSTRACTION

2.1 Mesh-Based Affine Abstraction

There are two parts in solving the problem. In the first part, we consider the subproblem of abstracting a single pair of affine hyperplanes for the nonlinear dynamics in a single subregion $I_i \in \mathcal{I}$ using mesh-based affine abstraction. Unlike the recent paper [1] in which only Lipchitz continuous functions have been considered, we provide a novel analysis that considers mesh elements, as opposed to point-wise analysis, which enables us to exploit the tighter bounds from the literature on linear interpolation [36, 10] for several classes of continuous functions with different degrees of smoothness.

Then, in the second subproblem, we extend the abstraction method from a single subregion to multiple subregions, which constitute a cover of the state space of the nonlinear dynamics. Specifically, we will construct an ε_f*-accurate* cover that is composed of subregions with a pair of families of affine hyperplanes $(\overline{\mathcal{F}}, \underline{\mathcal{F}})$ such that the nonlinear dynamics f is over-approximated with desired accuracy ε_f holds in each subregion.

As will be demonstrated in Section 4.1, our abstraction method outperforms the algorithm in [1] for a given ε_f in terms of computation time and number of subregions required to over-approximate a function.

2.1.1 Mesh-Based Affine Abstraction of a Single Subregion

To solve the subproblem of mesh-based affine abstraction of a single subregion, we will rely on the following result on linear interpolation error bounds over simplices:

Proposition 1 ([36, Theorem 4.1 & Lemma 4.3])**.** *Let S be an $(n+m)$-dimensional simplex such that $S \subseteq \mathbb{R}^{n+m}$ with diameter δ . Let $f : S \to \mathbb{R}$ be a nonlinear function and let f_l be the linear interpolation of f at the vertices of the simplex S. Then, the approximation error bound σ defined as the maximum error between f and f_l on S:*

$$\sigma = \max_{s \in S}(|f(s) - f_l(s)|) \tag{2.1}$$

is upper-bounded by

(i) $\sigma \leq 2\lambda\delta_s$, *if* $f \in C^0$ *on* S,

(ii) $\sigma \leq \lambda\delta_s$, *if* f *is Lipschitz continuous on* S,

(iii) $\sigma \leq \delta_s \max_{s\in S} \|f'(s)\|_2$, *if* $f \in C^1$ *on* S,

(iv) $\sigma \leq \frac{1}{2}\delta_s^2 \max_{s\in S} \|f''(s)\|_2$, *if* $f \in C^2$ *on* S,

where λ is the Lipschitz constant, $f'(s)$ is the Jacobian of $f(s)$, $f''(s)$ is the Hessian of $f(s)$ and δ_s is simplex ball radius that satisfies

$$\delta_s \leq \sqrt{\frac{n+m}{2(n+m+1)}}\delta.$$

According to [36], all factors are the best possible, while [10] proposes a mapping of the original simplex to to an “isotropic” space to obtain a better bound for the simplex ball radius δ_s. On the other hand, the Lipschitz constant λ for f on S can be computed using well-known techniques, e.g., [26], while the constants for cases (iii) and (iv) above can be computed using any off-the-shelf optimization software.

Moreover, we derive a useful lemma as follows:

Lemma 1. *Let f_1 and f_2 be affine hyperplanes on the same $(n+m)$-dimensional simplicial domain $S_k \subseteq \mathbb{R}^{n+m}$ with vertex set $\mathcal{V}_k = \{v_{k_1}, \ldots, v_{k_{n+m+1}}\}$. Suppose that*

$$f_1(v_{k_i}) \geq f_2(v_{k_i}), \ \forall \ i \in [n+m+1]. \tag{2.2}$$

Then, $f_1(s) \geq f_2(s), \ \forall \, s \in \ S$.

Proof. Since S is a simplex, any point $s \in S$ can be represented as $s = \sum_{i=1}^{n+m+1} \alpha_i v_{k_i}$, where $\alpha_i \geq 0$, $\sum_{i=1}^{n+m+1} \alpha_i = 1$. Moreover, we represent the affine hyperplanes as

$$f_1(s) = A_1 s + b_1 = \sum_{i=1}^{n+m+1} \alpha_i (A_1 v_{k_i} + b_1) = \sum_{i=1}^{n+m+1} \alpha_i f_1(v_{k_i}),$$
$$f_2(s) = A_2 s + b_2 = \sum_{i=1}^{n+m+1} \alpha_i (A_2 v_{k_i} + b_2) = \sum_{i=1}^{n+m+1} \alpha_i f_2(v_{k_i}).$$

Since (2.2) holds by assumption and $\alpha_i \geq 0$, the result follows directly from the above. □

Armed with the above interpolation error bounds and lemma, we can obtain the following lemma and theorem using a novel analysis that considers mesh elements for each subregion, as opposed to point-wise analysis in [1], resulting in tighter bounds and more effective abstraction.

Lemma 2. *Given a nonlinear function $f : I \to \mathbb{R}^n$ with a hyperrectangular domain $I \subset \mathbb{R}^{n+m}$ for any subregion $I \in \mathcal{I}$, let $\mathcal{V} = \{v_1, v_2, \ldots, v_l\}$ be a set of l grid points of a uniform mesh of the subregion I. Suppose that we have affine hyperplanes f_u and f_b such that:*

$$f_u(v_i) \geq f(v_i), \forall i \in [l], \tag{2.3}$$

$$f_b(v_i) \leq f(v_i), \forall i \in [l], \tag{2.4}$$

then, the affine hyperplanes $\overline{f}$ and $\underline{f}$ over-approximate the function f in the entire subregion I, i.e.,

$$\overline{f}(x,u) = f_u(x,u) + \sigma \geq f(x,u),\ \forall (x,u) \in I, \tag{2.5}$$

$$\underline{f}(x,u) = f_b(x,u) + \sigma \leq f(x,u),\ \forall (x,u) \in I, \tag{2.6}$$

where σ is a vector of the smallest possible error bounds based on the degrees of smoothness of each element of the vector-valued function f (cf. Proposition 1).

Proof. First, we note that the given hyperrectangular mesh can be considered to be comprised of simplices with the same set of vertices . Next, consider any $(n+m)$-dimensional simplex $S_k \subset I$ with vertex set $\mathcal{V}_k = \{v_{k_1}, \ldots, v_{k_{n+m+1}}\}$. By assumption, there exists an affine plane f_u that satisfies (2.3), and hence also at the vertices in $\mathcal{V}_k$, i.e.,

$$f_u(v_{k_i}) \geq f(v_{k_i}),\ \forall\ i \in [n+m+1], \tag{2.7}$$

since $\mathcal{V}_k \subseteq \mathcal{V}$. Moreover, the linear interpolation of the simplex vertices, $f_l(x,u)$, $\forall\,(x,u) \in S_k$ is a uniquely determined affine plane. Since f_u and f_l are both affine over the same domain, by Lemma 1, we have

$$\begin{aligned} & f_u(x,u) \geq f_l(x,u), && \forall (x,u) \in S_k, \\ \implies\ & \overline{f}(x,u) = f_u(x,u) + \sigma \geq f_l(x,u) + \sigma, && \forall (x,u) \in S_k. \end{aligned}$$

By Proposition 1, $f_l(x,u) + \sigma \geq f(x,u), \forall (x,u) \in S_k$, hence

$$\overline{f}(x,u) \geq f(x,u),\ \forall\ (x,u) \in S_k.$$

Since this result is applicable for all $S_k \subseteq I$ with the same f_u, we further have

$$\overline{f}(x,u) \geq f(x,u),\ \forall\,(x,u) \in I. \tag{2.8}$$

A similar proof can be derived to obtain (2.6). □

Theorem 1. *Given a nonlinear function $f : I \to \mathbb{R}^n$ with a hyperrectangular domain $I \subset \mathbb{R}^{n+m}$ for any subregion $I \in \mathcal{I}$, let $\mathcal{V} = \{v_1, v_2, \ldots, v_l\}$ be a set of l grid points of a uniform mesh of the subregion I and $\mathcal{C} = \{v_1^c, \ldots, v_{2(n+m)}^c\}$ be a set of the corner points of the hyperrectangular domain. The affine hyperplanes $\overline{f}$ and $\underline{f}$ that over-approximate/abstract f are given by:*

$$\overline{f} = f_u + \sigma, \quad \underline{f} = f_b - \sigma,$$

with σ as defined in Lemma 2, $f_u = \overline{A}\,x + \overline{B}\,u + h_u$, and $f_b = \underline{A}\,x + \underline{B}\,u + h_b$, where $\overline{A}, \underline{A}, \overline{B}, \underline{B}, h_u$ and h_b are obtained from the following linear programming (LP) problem:

$$\begin{aligned} \min_{\theta,\overline{A},\underline{A},\overline{B},\underline{B},h_u,h_b} \quad & \theta \\ \textit{subject to} \quad & \overline{A}\,x_i + \overline{B}\,u_i + h_u \geq f(x_i, u_i), && \text{(2.9a)} \\ & \underline{A}\,x_i + \underline{B}\,u_i + h_b \leq f(x_i, u_i), && \text{(2.9b)} \\ & (\overline{A} - \underline{A})\,x_j^c + (\overline{B} - \underline{B})\,u_j^c + h_u - h_b \leq \theta, && \text{(2.9c)} \\ & \forall i \in [l], \forall\, j \in [2^{(n+m)}], \end{aligned}$$

where (x_i, u_i) and (x_j^c, u_j^c) are the state-input values at the grid point v_i of the mesh and the vertex c_j of I, respectively.

Proof. The first two constraints (2.9a) and (2.9b) in the linear optimization problem can be interpreted as:

$$\overline{A}\,x_i + \overline{B}\,u_i + h_u = f_u(v_i) \geq f(v_i),\ \forall i \in [l],$$

$$\underline{A}\,x_i + \underline{B}\,u_i + h_b = f_b(v_i) \leq f(v_i),\ \forall i \in [l].$$

Based on Lemma 2, these inequalities imply that

$$\overline{f}(x, u) \geq f(x, u)\,, \forall (x, u)\ \in I,$$

$$\underline{f}(x, u) \leq f(x, u)\,, \forall (x, u)\ \in I,$$

which means that (2.9a) and (2.9b) always make sure that $\overline{f}$ and $\underline{f}$ are completely over and under f in I, as required by the definition of affine abstraction. Next, we wish to make f_u and f_b to be as close to each other as possible by minimizing θ, defined as:

$$\theta = \max_{(x,u)\in\mathcal{X}\times\mathcal{U}} \|f_u(x,u) - f_b(x,u)\|_\infty.$$

We now show this can be rewritten as a minimization problem with the objective function θ and the third constraint (2.9c). Consider any one dimension in $\mathbb{R}^{n+m}$ with the other dimensions arbitrarily fixed. Due to the linear nature of the difference between f_u and f_b, the difference can only be increasing or decreasing as the considered point in I moves in one direction. Because of this, the maximum difference would be at one of the ends. Since this argument applies to all dimensions, it follows that the maximum difference must be attained at one of the vertices of I. Hence, we only need to minimize the difference among the vertices of the $(n+m)$-dimensional hyperrectangle I, which leads to the third constraint (2.9c). □

2.1.2 Mesh-Based Affine Abstraction of Multiple Subregions

For multiple subregions, the mesh-based affine abstraction is provided in Algorithm 1, in which the abstraction method of a single subregion (cf. Theorem 1) is considered as the `abstraction` function. In Algorithm 1, the `epsCover` function is recursive in nature. First, the `abstraction` function is run in order to obtain $\overline{f}, \underline{f}$ and $e(\overline{f}, \underline{f})$. Then, the error $e(\overline{f}, \underline{f})$ is compared to the desired error ε_f. If it is smaller than ε_f, the information about the subregion boundary (`bound`) and the corresponding hyperplanes (with desired accuracy) is collected in a data structure called `cover`. Otherwise, the function `divBound` divides the state domain into a finer cover $\mathcal{I} = \{I_1, \ldots, I_{2^{n+m}}\}$ by partitioning each interval $[a_j, b_j], \forall j \in [n+m]$ into two

Algorithm 1: Creating a ε_f-accurate Cover

```
Data: $f$, bound $= \mathcal{X} \times \mathcal{U}$, resolution $r$, desired accuracy $\varepsilon_f$
function epsCover($f$, bound, $r$, $\varepsilon_f$)
    $(\overline{f}, \underline{f}, e(\overline{f}, \underline{f}))$ ← abstraction($f$, bound, $r$, $\varepsilon_f$)
    if $e(\overline{f}, \underline{f}) \leq \varepsilon_f$ then
        cover $= \{\overline{f}, \underline{f},$ bound$\}$
        return (cover)
    else
        $\mathcal{I}$ ← divBounds(bound)
        for $i = 1 : 2^{n+m}$ do
            cell$\{i\}$ = epsCover($f$, $I_i$, $r$, $\varepsilon_f$)
        end
        cover $= \bigoplus_{i=1}^{2^{n+m}} \{$cell$\{i\}\}$ ($\bigoplus$ = concatenation)
    end
    return (cover, $\mathcal{I}$)
function divBounds(bound)
    Refer to Section 2.1.2 for its description
    return (subBounds)
function abstraction($f$, bound, $r$, $\varepsilon_f$)
    Refer to Theorem 1 for its description
    return $(\overline{f}, \underline{f}, e(\overline{f}, \underline{f}))$
```

subintervals of width $(b_j - a_j)/2$. Thus, the region is divided into $2^{(n+m)}$ different subregions denoted by `subBounds`. Now, each of the subregions `subBounds` is recursively passed to `epsCover` in place of the original region until $e(\overline{f}, \underline{f})$ in each newly obtained subregion has an error that is less than ε_f. In each recursion, we keep tracking of the subregion boundaries and the corresponding affine-hyperplanes and store it in the data structure `cover`.

2.2 MINLP-Based Affine Abstraction

Problem 1 (Affine Abstraction)**.** *Given a* nonlinear *n-dimensional vector field* $\mathfrak{f}(\vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}, w)$ *with (polytopic) domain* $\vec{\boldsymbol{x}} \in \boldsymbol{\mathcal{X}}, \vec{\boldsymbol{u}} \in \boldsymbol{\mathcal{U}}$*,* $w \in \mathcal{W}$*, find two n dimensional affine planes (i.e., upper and lower planes defined by matrices* $\overline{A}, \underline{A}, \overline{B}, \underline{B}, \overline{B}_w, \underline{B}_w$ *and vectors* $\overline{f}$*,* $\underline{f}$*) such that they "contain" (i.e., upper- and lower bound) the given vector field with minimum separation, as expressed by the following:*

$$\min_{\overline{A},\underline{A},\overline{B},\underline{B},\overline{B}_w,\underline{B}_w,\overline{f},\underline{f}\widetilde{A},\widetilde{B},\widetilde{B}_w,\widetilde{f}} \quad \widetilde{A} + \lambda_1 \widetilde{B} + \lambda_2 \widetilde{B}_w + \lambda_3 \widetilde{f}$$

subject to

$$\begin{aligned} &\underline{A}\,\vec{\boldsymbol{x}} + \underline{B}\,\vec{\boldsymbol{u}} + \underline{B}_w w + \underline{f} \le \mathfrak{f}(\vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}, w),\ \forall(\vec{\boldsymbol{x}} \in \boldsymbol{\mathcal{X}}, \vec{\boldsymbol{u}} \in \boldsymbol{\mathcal{U}}, \\ &\mathfrak{f}(\vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}, w) \le \overline{A}\,\vec{\boldsymbol{x}} + \overline{B}\,\vec{\boldsymbol{u}} + \overline{B}_w w + \overline{f}, \qquad w \in \mathcal{W}), \end{aligned} \tag{2.10a}$$

$$\begin{aligned} &\|\overline{A} - \underline{A}\| \le \widetilde{A}, \|\overline{B} - \underline{B}\| \le \widetilde{B}, \|\overline{B}_w - \underline{B}_w\| \le \widetilde{B}_w, \\ &\|\overline{f} - \underline{f}\| \le \widetilde{f}, \overline{A} \ge \underline{A}, \overline{B} \ge \underline{B}, \overline{B}_w \ge \underline{B}_w, \overline{f} \ge \underline{f}, \end{aligned} \tag{2.10b}$$

where λ_1*,* λ_2 *and* λ_3 *are tuning weights and suitable norms are also chosen based on the application at hand.*

In this section, we present an alternative approach to the previous mesh-based affine abstraction method by directly considering the mixed-integer nonlinear program (MINLP) as formulated in Problem 1. Since the robust formulation given in Problem 1 cannot be directly implemented using standard optimization packages, we will convert the problem into a more amenable form such that off-the-shelf optimization tools can be applied. For simplicity, we first describe our approach for a 1-dimensional vector field with interval domains, before describing how this can be extended to higher dimensional systems. Moreover, we will assume in the following discussion that there is at most one local optimum in the domain $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{U}}, \mathcal{W}$. This can also be extended to the case with multiple local optima.

Theorem 2 (Affine Abstraction)**.** *Given a* nonlinear 1-*dimensional differentiable vector field* $\mathfrak{f}(\vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}, w)$ *with interval domains* $\vec{\boldsymbol{x}} \in \boldsymbol{\mathcal{X}}, \vec{\boldsymbol{u}} \in \boldsymbol{\mathcal{U}}$*,* $w \in \mathcal{W}$*, two affine planes (i.e., upper and lower planes defined by* $\overline{A}, \underline{A}, \overline{B}, \underline{B}, \overline{B}_w, \underline{B}_w$ *and vectors* $\overline{f}$*,* $\underline{f}$*) that contain the given vector field with minimum separation are solutions to the following nonlinear optimization problem:*

$$\min_{\substack{\overline{A},\underline{A},\overline{B},\underline{B},\overline{B}_w,\underline{B}_w,\overline{f},\underline{f}\widetilde{A},\widetilde{B},\widetilde{B}_w,\widetilde{f},\\ \vec{\boldsymbol{x}}_u,\vec{\boldsymbol{u}}_u,w_u,\vec{\boldsymbol{x}}_b,\vec{\boldsymbol{u}}_b,w_b}} \quad \widetilde{A}+\lambda_1\widetilde{B}+\lambda_2\widetilde{B}_w+\lambda_3\widetilde{f}$$

subject to

$$\begin{aligned}&\underline{A}\,\vec{\boldsymbol{x}}+\underline{B}\,\vec{\boldsymbol{u}}+\underline{B}_w w+\underline{f}\le\mathfrak{f}(\vec{\boldsymbol{x}},\vec{\boldsymbol{u}},w),\ \forall(\vec{\boldsymbol{x}}\in\widetilde{\boldsymbol{\mathcal{X}}},\vec{\boldsymbol{u}}\in\widetilde{\boldsymbol{\mathcal{U}}},\\ &\mathfrak{f}(\vec{\boldsymbol{x}},\vec{\boldsymbol{u}},w)\le\overline{A}\,\vec{\boldsymbol{x}}+\overline{B}\,\vec{\boldsymbol{u}}+\overline{B}_w w+\overline{f},\qquad w\in\widetilde{\mathcal{W}}),\end{aligned} \tag{2.11a}$$

$$\begin{aligned}&\underline{A}-\nabla_x\,\mathfrak{f}(\vec{\boldsymbol{x}}_b,\vec{\boldsymbol{u}}_b,w_b)=0,\\ &\underline{B}-\nabla_u\,\mathfrak{f}(\vec{\boldsymbol{x}}_b,\vec{\boldsymbol{u}}_b,w_b)=0,\\ &\underline{B}_w-\nabla_w\,\mathfrak{f}(\vec{\boldsymbol{x}}_b,\vec{\boldsymbol{u}}_b,w_b)=0,\\ &((\vec{\boldsymbol{x}}_b\in\boldsymbol{\mathcal{X}}\ \wedge\ \vec{\boldsymbol{u}}_b\in\boldsymbol{\mathcal{U}}\ \wedge\ w_b\in\mathcal{W}\ \wedge(\underline{A}\,\vec{\boldsymbol{x}}_b+\underline{B}\,\vec{\boldsymbol{u}}_b\\ &\quad+\underline{B}_w w_b+\underline{f}\le\mathfrak{f}(\vec{\boldsymbol{x}}_b,\vec{\boldsymbol{u}}_b,w_b)))\\ &\quad\vee\vec{\boldsymbol{x}}_b\notin\boldsymbol{\mathcal{X}}\ \vee\ \vec{\boldsymbol{u}}_b\notin\boldsymbol{\mathcal{U}}\ \vee\ w_b\notin\mathcal{W}),\end{aligned} \tag{2.11b}$$

$$\begin{aligned}&\overline{A}-\nabla_x\,\mathfrak{f}(\vec{\boldsymbol{x}}_u,\vec{\boldsymbol{u}}_u,w_u)=0,\\ &\overline{B}-\nabla_u\,\mathfrak{f}(\vec{\boldsymbol{x}}_u,\vec{\boldsymbol{u}}_u,w_u)=0,\\ &\overline{B}_w-\nabla_w\,\mathfrak{f}(\vec{\boldsymbol{x}}_u,\vec{\boldsymbol{u}}_u,w_u)=0,\\ &((\vec{\boldsymbol{x}}_u\in\boldsymbol{\mathcal{X}}\ \wedge\ \vec{\boldsymbol{u}}_u\in\boldsymbol{\mathcal{U}}\ \wedge\ w_u\in\mathcal{W}\ \wedge(\underline{A}\,\vec{\boldsymbol{x}}_u+\underline{B}\,\vec{\boldsymbol{u}}_u\\ &\quad+\underline{B}_w w_u+\underline{f}\ge\mathfrak{f}(\vec{\boldsymbol{x}}_u,\vec{\boldsymbol{u}}_u,w_u)))\\ &\quad\vee\vec{\boldsymbol{x}}_u\notin\boldsymbol{\mathcal{X}}\ \vee\ \vec{\boldsymbol{u}}_u\notin\boldsymbol{\mathcal{U}}\ \vee\ w_u\notin\mathcal{W}),\end{aligned} \tag{2.11c}$$

$$\begin{aligned}&\|\overline{A}-\underline{A}\|\le\widetilde{A},\|\overline{B}-\underline{B}\|\le\widetilde{B},\|\overline{B}_w-\underline{B}_w\|\le\widetilde{B}_w,\\ &\|\overline{f}-\underline{f}\|\le\widetilde{f},\overline{A}\ge\underline{A},\overline{B}\ge\underline{B},\overline{B}_w\ge\underline{B}_w,\overline{f}\ge\underline{f},\end{aligned} \tag{2.11d}$$

where λ_1, λ_2, λ_3 are tuning weights (chosen based on the application), $\widetilde{\boldsymbol{\mathcal{X}}}$, $\widetilde{\boldsymbol{\mathcal{U}}}$, $\widetilde{\mathcal{W}}$ are the sets of endpoints of the interval domains $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{U}}$, $\mathcal{W}$, respectively, $\vec{\boldsymbol{x}}_u$, $\vec{\boldsymbol{u}}_u$ and w_u are the local optimum for the difference between the upper plane and function $\mathfrak{f}$, and similarly, $\vec{\boldsymbol{x}}_b$, $\vec{\boldsymbol{u}}_b$ and w_b for the bottom plane, while $\wedge$ and $\vee$ are logical `AND` *and* `OR` *operators*[1] *and* ∇ *denotes the derivative operator.*

Proof. The optimization formulation above is as in Problem 1 except that the semi-infinite constraints (2.10a) are replaced by readily implementable constraints (2.11a), (2.11c), (2.11b). This is possible because the maximum of a differentiable (thus, continuous) function over a closed domain is the maximum of its local optima (i.e., minima, maxima or saddle points) in its interior (i.e., (2.11b)) and the maximum over its boundaries (i.e., its endpoints; (2.11a)). The same holds for minimization over a closed domain, resulting in (2.11a) and (2.11c) for the lower plane.

Eq. (2.11a) is required such that the upper and lower planes upper- and lower-bound each boundary/endpoint of the domains. On the other hand, Eqs. (2.11b) and (2.11c) are such that the local optima, which are given by their first order necessary condition (i.e., their first order derivative is set to zero), are also upper- and lower-bounded by the two planes, respectively. The constraints with logical operators are to be understood to be in conjunction with the other constraints, and has the interpretation that only the local optima in the given domain needs to be upper- or lower-bounded by the two planes. Finally, Eq. (2.11d) along with the objective function ensures that the upper plane remains above the lower plane and that the separation between the two planes is as small as possible. □

To extend the above formulation to higher dimensions, the key difference from the 1-dimensional case is that the boundaries of the domain are higher dimensional facets as opposed to line segments. To deal with this, the above procedure of replacing the maximization or minimization over a domain with the maximizing or minimizing over the local optima (using first order optimality condition for constrained optimization)

[1] Note that these logical operators can be directly implemented by off-the-shelf software such as YALMIP [24]. If needed, they can also be converted to mixed-integer constraints.

and its boundaries as in (2.11a), (2.11c), (2.11b) can be repeated to recursively reduce the dimension of the boundaries by one until we obtain line segments. This procedure may be tedious to implement for systems with high dimensions and domains with many facets, hence, in practice, we may start with only constraints on the vertices and iteratively add facets of the domain of interest when the resulting planes are found to intersect them.

Although this MINLP-based approach is exact, as above-mentioned, this approach may be tedious and moreover, a global optimum may not be found. Hence, the mesh-based approach in the previous section is more appealing in general.

Chapter 3

ACTIVE MODEL DISCRIMINATION

In this section, we extend and modify the optimization-based approach proposed in [11] to solve the nonlinear active model discrimination problem with given uncertain affine models from Chapter 2. This approach relies on formulating the problem as a bi-level optimization problem which can be further converted to a single level optimization problem using KKT conditions. The bi-level optimization can be then represented as MILP's for which off-the-shelf optimization softwares are readily available [15, 9]. For the sake of clarity, we will defer the definitions of certain matrices in the following results to the appendix. Moreover, for brevity, the proofs of this approach are omitted, as they follow similar steps to the proofs in [11].

Lemma 3 (Bi-level Optimization Formulation)**.** *Given a separability index ϵ, the active model discrimination problem is equivalent (up to ϵ) to a bi-level optimization problem with the following outer problem:*

$$\min_{u_T} J(u_T) \tag{P_{Outer}}$$

$$s.t. \quad \overline{Q}_u u_T \leq \overline{q}_u, \tag{3.1a}$$

$$\forall \iota \in \mathbb{Z}_{\mathrm{I}}^{+}: \ \delta^{\iota *}(u_T) \geq \epsilon, \tag{3.1b}$$

where $\delta^{\iota *}(u_T)$ is the solution to the inner problem:

$$\delta^{\iota *}(u_T) = \min_{\delta^\iota, \overline{x}^\iota} \ \delta^\iota \tag{P_{Inner}}$$

$$s.t. \quad R_1^\iota \overline{x}^\iota \leq r_1^\iota + S_1^\iota u_T, \tag{3.2a}$$

$$R_2^\iota \overline{x}^\iota \leq \mathbb{1}\delta^\iota + r_2^\iota + S_2^\iota u_T, \tag{3.2b}$$

$$H_{\overline{x}}^\iota \overline{x}^\iota \leq h_{\overline{x}}^\iota. \tag{3.2c}$$

Theorem 3 (Discriminating Input Design as an MILP). *Given a separability index ϵ, the active model discrimination problem (Problem 1.??) is equivalent (up to ϵ) to the following mixed-integer optimization problem:*

$$
\begin{aligned}
&\min_{u_T,\delta^\iota,,\bar{x}^\iota,\mu_1^\iota,,\mu_2^\iota,\mu_3^\iota} && J(u_T) && (P_{DID})\\
&s.t. && \bar{Q}_u u_T \leq \bar{q}_u, \\
&\forall \iota \in \mathbb{Z}_I^+ : && \delta^\iota(u_T) \geq \epsilon, \\
&\forall \iota \in \mathbb{Z}_I^+ : && 0 = \textstyle\sum_{i=1}^{i=\kappa} \mu_{1,i}^\iota H_{\bar{x}}^\iota(i,m) + \sum_{j=1}^{j=\xi} \mu_{2,j}^\iota R_1^\iota(j,m) \\
& && \quad + \textstyle\sum_{k=1}^{k=\rho} \mu_{3,k}^\iota R_2^\iota(k,m), \forall m = 1,\cdots,\eta, \\
& && 0 = 1 - {\mu_3^\iota}^T \mathbb{1}, \\
& && \widetilde{H}_{\bar{x},i}^\iota \bar{x}^\iota - h_{\bar{x},i}^\iota \leq 0, \forall i = 1,\ldots \kappa, \\
& && \widetilde{R}_{1,j}^\iota \bar{x}^\iota - r_{1,j}^\iota - S_{1,j}^\iota u_T \leq 0, \forall j = 1,\ldots \xi, \\
& && \widetilde{R}_{2,k}^\iota \bar{x}^\iota - \delta^\iota - r_{2,k}^\iota - S_{2,k}^\iota u_T \leq 0, \forall k = 1,\ldots \rho, \\
& && \mu_{1,i}^\iota \geq 0, \quad \forall i = 1,\ldots \kappa, \\
& && \mu_{2,j}^\iota \geq 0, \quad \forall j = 1,\ldots \xi, \\
& && \mu_{3,k}^\iota \geq 0, \quad \forall k = 1,\ldots \rho, \\
&\forall \iota \in \mathbb{Z}_I^+, \forall i \in \mathbb{Z}_\kappa^+ : && \textit{SOS-1} : \{\mu_{1,i}^\iota, \widetilde{H}_{\bar{x},i}^\iota \bar{x}^\iota - h_{\bar{x},i}^\iota\}, \\
&\forall \iota \in \mathbb{Z}_I^+, \forall j \in \mathbb{Z}_\xi^+ : && \textit{SOS-1} : \{\mu_{2,j}^\iota, \widetilde{R}_{1,j}^\iota \bar{x}^\iota - r_{1,j}^\iota - \tilde{S}_{1,j}^\iota u_T\}, \\
&\forall \iota \in \mathbb{Z}_I^+, \forall j \in \mathbb{Z}_\rho^\rho : && \textit{SOS-1} : \{\mu_{3,k}^\iota, \widetilde{R}_{2,k}^\iota \bar{x}^\iota - \delta^\iota - r_{2,k}^\iota - \tilde{S}_{2,k}^\iota u_T\},
\end{aligned}
$$

where $\mu_{1,i}^\iota$, $\mu_{2,j}^\iota$ and $\mu_{3,j}^\iota$ are dual variables, while $\widetilde{H}_{\bar{x},i}^\iota$ is the i-th row of $H_{\bar{x}}^\iota$, $\widetilde{R}_{1,j}^\iota$ and $\widetilde{S}_{1,j}^\iota$ are the j-th row of R_1^ι and S_1^ι, respectively, $\widetilde{R}_{1,k}^\iota$ and $\widetilde{S}_{2,k}^\iota$ are the k-th row of R_2^ι and S_2^ι, respectively, $\eta = IT(n + m_d + m_w + m_v)$ is the number of columns of $H_{\bar{x}}^\iota$, $\kappa = 2IT(c_0 + c_d + c_w + c_v)$ is the number of rows of $H_{\bar{x}}^\iota$, $\xi = 2IT(c_x + c_y)$ is the number of rows of R_1^ι and $\rho = 2ITp$ is the number of rows of R_2^ι.

Chapter 4

SIMULATION EXAMPLES—APPLICATION TO INTENTION IDENTIFICATION

In this section, we first compare the performance of our mesh-based affine abstraction approach with the state-of-the-art approach in [1]. Then, we apply both proposed affine abstraction approaches as well as the proposed active model discrimination technique to the problem of intention identification of other vehicles in a highway lane changing scenario.

In addition, we will investigate the effects of the choices of various parameters on the proposed mesh-based affine abstraction algorithm. In particular, we consider the impacts of the desired accuracy ε_f and approximation error bound σ in Section and the resolution vector r All the examples are implemented in MATLAB on a 2.9 GHz Intel Core i5 CPU.

4.1 One-Dimensional Affine Abstraction Example ($f(x, y) = x \cos y$)

In order to compare the effectiveness of our mesh-based affine abstraction approach with that in [1], we begin by applying our algorithm to the same one-dimensional nonlinear function $f(x, y) = x \cos y$, on the interval $[-2, 2] \times [0, 2\pi]$. Since this function is infinitely differentiable, all approximation error bounds σ from Lemma 2 apply and these bounds are used to obtain Table 4.1 for three different desired accuracies, $\varepsilon_f \in \{0.05, 0.1, 0.2\}$. The resulting number of subregions serve as a measure for quality of the abstraction procedure because a better approximation would naturally lead to fewer subregions that are required for obtaining a given desired accuracy ε_f (cf. Figure 4.4).

Table 4.1: Results of affine abstraction for the nonlinear function $x \cos y$ for varying desired accuracies ε_f and varying approximation error bounds σ (shown for the entire domain) corresponding to different degrees of smoothness.

	Desired Accuracy, ε_f	0.2	0.1	0.05
(i) C^0 function	No. of Subregions	784	1024	4096
($\sigma = 1.351$)	CPU Time (s)	169.97	212.48	765.22
(ii) Lipschitz function	No. of Subregions	256	976	3376
($\sigma = 0.676$)	CPU Time (s)	55.81	213.15	674.49
(iii) C^1 function	No. of Subregions	232	688	1024
($\sigma = 0.478$)	CPU Time (s)	50.84	149.83	212.89
(iv) C^2 function	No. of Subregions	64	232	256
($\sigma = 0.228$)	CPU Time (s)	14.22	50.77	56.83
[1][1] Lipschitz function	No. of Subregions	256	1024	4096
($\sigma = 1.170$)	Comp. Time (s)	57.18	214.40	786.88

Table 4.1 demonstrates that our proposed abstraction algorithm outperforms the approach in [1] because of the tighter bounds σ that we can obtain, with the exception of the case when we only assume continuity but not differentiability (i.e., $x \cos y$ is a C^0 function). Moreover, the computation (CPU) time is proportional to the resulting number of subregions.

As above-mentioned, the choice of desired accuracy ε_f impacts on the number of subregions, where a larger ε_f leads to fewer subregions, as shown in Figure 4.1. On the other hand, the choice of approximation error bound also impacts the number of subregions, where a tighter bound leads to less subregions, as illustrated in Figure 4.2.

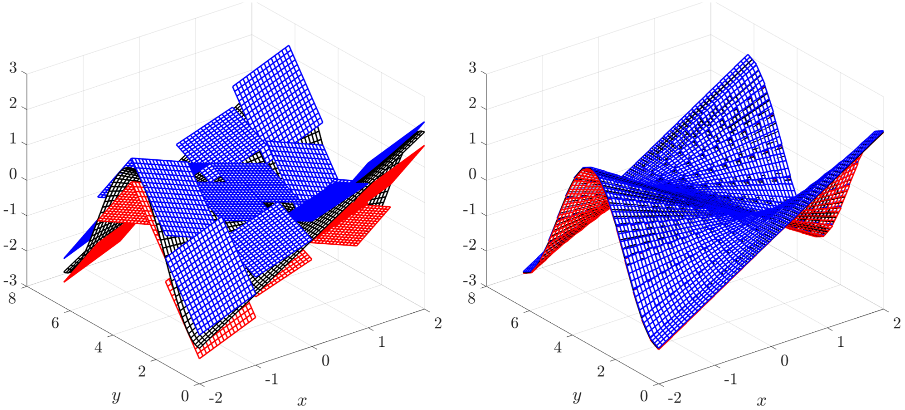

Figure 4.1: Affine abstraction of $x \cos y$ using an approximation error bound $\sigma = 0.228$ and desired accuracies, $\varepsilon_f = 1$ (**left**) and $\varepsilon_f = 0.05$ (**right**), resulting in 16 and 256 subregions.

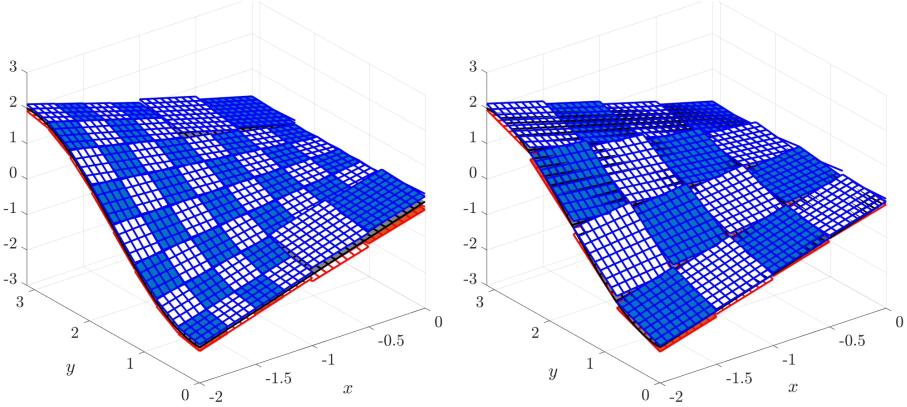

Figure 4.2: Affine abstraction of $x \cos y$ using a desired accuracy $\varepsilon_f = 0.4$ and approximation error bounds $\sigma = 1.170$ (**left**, [1]) and $\sigma = 0.228$ (**right**); zoomed in $[-2, 0] \times [0, \pi]$ with added emphasis (colored) on different subregions.

4.2 Application Example: Active Intention Identification in Lane Change Scenario

In this example, we apply the active model discrimination approach proposed in the previous sections to design a separating control input that, in conjunction with a modified model invalidation algorithm (e.g., [17, 21, 20]), can be used for active intention identification in a highway lane change scenario (cf. Figure 4.3).

The dynamics of the ego vehicle (blue) is of the Dubins car [12] with acceleration

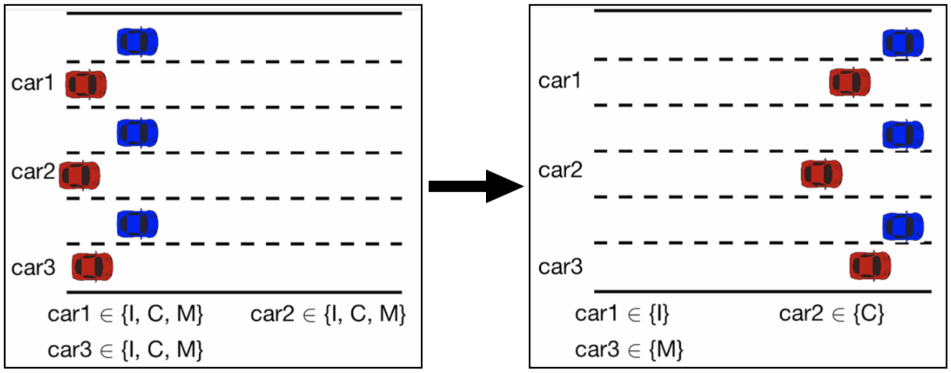

Figure 4.3: Illustration of active intention identification, where separating inputs help to distinguish Inattentive, Cautious and Malicious intentions [11].

input:

$$\dot{x}_e = v_e \cos\phi_e,\ \dot{y}_e = v_e \sin\phi_e,\ \dot{v}_e = u_1 + w_1,\ \dot{\phi}_e = u_2 + w_2, \tag{4.1}$$

while the other car dynamics (red) is described by:

$$\dot{x}_o = v_o,\ \dot{v}_o = d_i + w_3,$$

where x_e and y_e are the longitudinal and lateral coordinates of the ego car, v_e is its speed and ϕ_e is its heading angle, while x_o is the longitudinal coordinate of the other car with its speed given by v_o (no lateral movement), with process noise signals $w_j, j \in \{1,2,3\}$. $u_1 \in \mathcal{U}_a \triangleq [-7.848, 3.968]$ and $u_2 \in \mathcal{U}_s \triangleq [-0.44, 0.44]$ are the acceleration and steering inputs of the ego car, whereas d_i is the (uncontrolled) acceleration input of the other car for each intention $i \in \{I, C, M\}$, corresponding to an Inattentive, Cautious or Malicious driver.

The **Inattentive** driver is unaware of the ego car and tries to maintain his speed using an acceleration input which lies in a small range $d_I \in \mathcal{D}_I \triangleq [-0.392, 0.198]$. On the other hand, the **Cautious** driver tends to yield the lane to the ego car with the input equal to $d_C \triangleq -K_{d,C}(v_e - v_o) + L_{p,C}\phi_e + L_{d,C}\dot{\phi}_e + \tilde{d}_C$, where $\dot{\phi}_e = u_2 + w_2$, $K_{d,C} = 1$, $L_{p,C} = 12$ and $L_{d,C} = 14$ are PD controller parameters and the input uncertainty is $\tilde{d}_C \in \mathcal{D}_C = \mathcal{D}_I$. Finally, the **Malicious** driver does not want to yield the lane and attempts to cause a collision with input equal to $d_M \triangleq K_{d,M}(v_e - v_o) - L_{p,M}\phi_e - L_{d,M}\dot{\phi}_e + \tilde{d}_M(k)$, if provoked, where $\dot{\phi}_e = u_2 + w_2$, $K_{d,C} = 0.9$, $L_{p,C} = 12$ and $L_{d,C} = 14$ are PD controller parameters and the input uncertainty satisfies $\tilde{d}_M \in \mathcal{D}_M = \mathcal{D}_I$.

Without loss of generality, we assume that the initial position and heading angle of the ego car are 0, while the initial velocities match typical speed limits of the highway. Moreover, both cars are close to the center of their lanes that are 3.2 m wide. Thus, the initial conditions are:

$$\begin{aligned} v_e(0) &\in [24, 26]\tfrac{m}{s}, \quad & y_e(0) &\in [1.5, 1.7]m, \\ v_o(0) &\in [24, 26]\tfrac{m}{s}, \quad & x_o(0) &\in [10, 12]m. \end{aligned}$$

Further, the velocity of the ego vehicle is constrained to be between $[20, 30]\frac{m}{s}$ at all times (in order to obey the speed limit of a highway), its heading angle is between $[0, 0.44]rad$ and its lateral position is constrained between $[0.3, 2.5]m$. Process and measurement noise signals are bounded with a range of $[-0.01, 0.01]$ and the separability threshold is set to $\epsilon = 0.5\frac{m}{s}$. Moreover, we assume the extreme scenario where only noisy observation of other car's velocity is observed $z = v_o + v$.

4.2.1 Affine Abstraction fo Dubins Dynamics

The Dubins vehicle and intention models above are nonlinear. Hence, we resort to the approach in Chapter 2 to obtain affine abstractions of the models. Since the nonlinearity only affects the speed v_e and the heading angle ϕ_e, we first define a suitable domain that is appropriate for the lane changing scenario. Specifically, we consider the speed of the ego car to be between 20 m/s to 30 m/s (72 to 108 km/h) and a heading angle range of −25° to 25° ($[-0.44, 0.44]rad$).

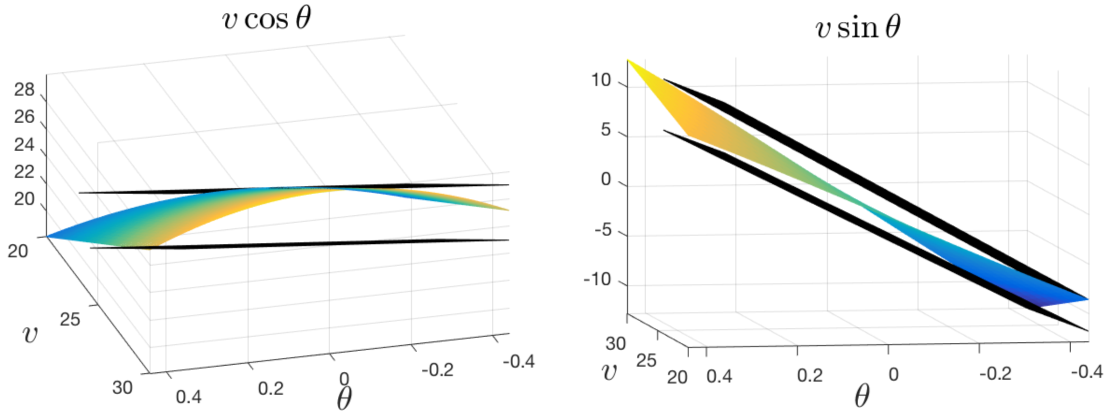

Figure 4.4: Affine abstraction/over-approximation of Dubins vehicle dynamics such that the true nonlinear system behavior is contained/enveloped by the abstraction.

MINLP-Based Affine Abstraction

Using the above domain, we can obtain an affine abstraction for the reduced 2-dimensional system in (5.1) with $\lambda_1 = 1000, \lambda_2 = \lambda_3 = 0$ and ∞-norms, as illustrated by Figure 4.4, and using the compact interval matrix representation, the abstracted open-loop model is given by:

$$\begin{aligned} \mathcal{A} &= [\underline{A}, \overline{A}] \\ &= \begin{bmatrix} 0 & 0 & [0.9947, 0.9956] & 2.3821 & 0 & 0 \\ 0 & 0 & [-0.1028, -0.0928] & [23.0325, 25.2673] & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

$$\mathcal{B} = \mathcal{B}_w = B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathcal{F} = [\underline{f}, \overline{f}] = \begin{bmatrix} [-3.7471, 0.2859] \\ [0.4413, 4.4754] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Mesh-Based Affine Abstraction

As in the mixed-integer nonlinear optimization (MINLP) approach above, we consider only one region (i.e., without subdividing into subregions) with the speed between $20m/s$ and $30m/s$ (72 to 108 km/h) and the heading angle between $-25°$ to $25°$ ($[-0.44, 0.44]$ rad). Moreover, we consider an objective function that minimizes $\gamma_A \|\overline{A} - \underline{A}\|_\infty + \gamma_h \|\overline{h} - \underline{h}\|_\infty$, where γ_A and γ_h are chosen as 0.5 and 5, respectively.

For very small resolution r, e.g., $r = 25$, the mesh-based approach performed worse than the MINLP approach (independent of resolution) in terms of the obtained optimal value, however the optimal value decreases rapidly as the resolution is increased, as shown in Figure 4.5. On the other hand, the computation (CPU) time of the mesh-based approach increases with increasing resolution but is still generally faster than the MINLP approach up until the resolution of over $r = 3000$.

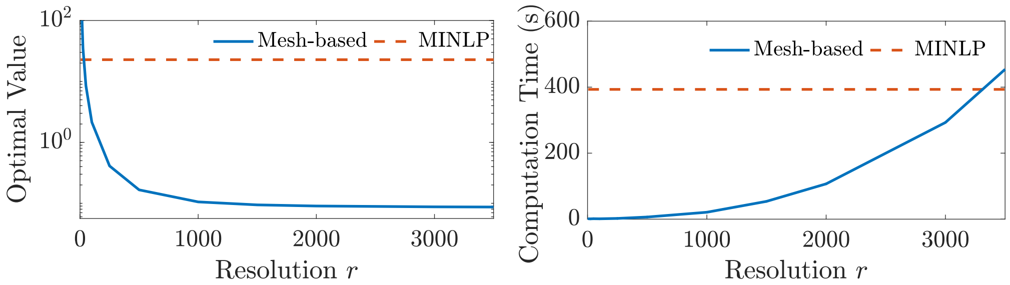

Figure 4.5: Decreasing optimal value (**left**) and increasing computation (CPU) time (**right**) as the resolution r is increased for the mesh-based affine abstraction approach, in comparison with the values obtained from the MINLP-based approach.

Similarly, we can obtain the abstracted open-loop model as:

$$\begin{aligned}\mathcal{A} &= [\underline{A}, \overline{A}] \\ &= \begin{bmatrix} 0 & 0 & [0.90474, 1] & [0, -0.07512] & 0 & 0 \\ 0 & 0 & [-0.085188, 0.085188] & 23.2331 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\end{aligned}$$

$$\mathcal{B} = \mathcal{B}_w = B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathcal{F} = [\underline{f}, \overline{f}] = \begin{bmatrix} [0.000233309, 0.00023858] \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

4.2.2 Active Nonlinear Model Discrimination

Combining the abstracted model with the intention models and considering a time-discretization with sampling time $\delta t = 0.3$ s, we obtain the following closed-loop intention models:

Inattentive Driver $(i = I)$:

$$\underline{A}_I = I + \delta t \underline{A},\ \overline{A}_I = I + \delta t \overline{A},$$

$$\underline{B}_I = \overline{B}_I = \underline{B}_{w,I} = \overline{B}_{w,I} = \delta t B,$$

$$C_I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},\ D_I = 0,\ D_{v,I} = 1,$$

$$\underline{f}_I = \delta t \underline{f},\ \overline{f}_I = \delta t \overline{f}.$$

Cautious Driver $(i = C)$:

$$\tilde{A}_C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -K_{d,C} & L_{d,C} & 0 & K_{d,C} \end{bmatrix},\ \tilde{B}_C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & L_{p,C} & 0 \end{bmatrix},$$

$$\underline{A}_C = I + \delta t(\underline{A} + \tilde{A}_C),\ \overline{A}_C = I + \delta t(\overline{A} + \tilde{A}_C),$$

$$\underline{B}_C = \overline{B}_C = \underline{B}_{w,C} = \overline{B}_{w,C} = \delta t(B + \tilde{B}_C),$$

$$C_C = C_I,\ D_C = 0,\ D_{v,C} = 1,$$

$$\underline{f}_C = \delta t \underline{f},\ \overline{f}_C = \delta t \overline{f}.$$

Malicious Driver $(i = M)$:

$$\tilde{A}_M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -K_{d,M} & L_{d,M} & 0 & K_{d,M} \end{bmatrix}, \tilde{B}_M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & L_{p,M} & 0 \end{bmatrix},$$

$$\underline{A}_M = I + \delta t(\underline{A} - \tilde{A}_M),\ \overline{A}_M = I + \delta t(\overline{A} - \tilde{A}_M),$$

$$\underline{B}_M = \overline{B}_M = \underline{B}_{w,M} = \overline{B}_{w,M} = \delta t(B - \tilde{B}_M),$$

$$C_M = C_I,\ D_M = 0,\ D_{v,M} = 1,$$

$$\underline{f}_M = \delta t \underline{f},\ \overline{f}_M = \delta t \overline{f}.$$

Next, we apply the active model discrimination approach we developed in Chapter 3 to the above uncertain affine models for the three intentions of the other driver in a highway lane changing scenario. Figure 4.6 shows the active inputs of the ego car to discern the other car's intention based on its response when using various objective functions when using MINLP-based approach. When comparing the solutions with different norms as the objective functions, we observe that $\|u\|_1$ reduces the sum of absolute values, thus keeping all the data points as close to zero as possible, $\|u\|_2$ minimizes energy and may be desirable to reduce fuel consumption, and $\|u\|_\infty$ ensures comfort by minimizing the maximum absolute input values.

Moreover, in all three solutions, the ego car accelerates and turns towards the other car's lane, and then either maintains its speed or decelerates while re-aligning to its lane, although now laterally shifted toward the other car's lane. Throughout the maneuver, the inattentive car would coast with small accelerations, the cautious car would slow down when it observes the ego car turning to its lane and the malicious car would try and match the ego car's position and cause a collision. The obtained optimal separating input is then applied to the ego vehicle in real-time. After measurements are recorded for $T = 3$, the passive model discrimination approach based on model invalidation of each intention model (e.g., [17, 20]) can be applied to identify the intention of the other vehicle since the observed dynamics of the other car will be consistent with only one of the three intention models by design (see above definition of separating input).

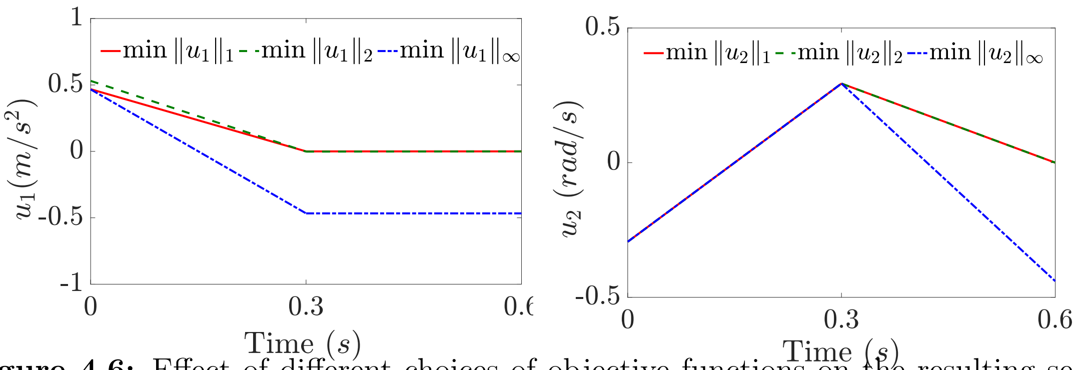

Figure 4.6: Effect of different choices of objective functions on the resulting separating inputs.

In addition, we compare the active model discrimination solutions when using the MINLP- and mesh-based affine abstraction approaches. Table 4.2 shows that the optimal values ($\{1, 2, \infty\}$-norms of the excitation input u_T) that are obtained for the active model discrimination problem based on mesh-based affine abstraction are lower than when the MINLP-based abstraction is used. However, this improvement comes at the cost of higher computation times.

Table 4.2: Optimal values and computation (CPU) times for active model discrimination when using affine abstractions of Dubins dynamics from MINLP- and mesh-based approaches.

		$\|u_T\|_1$	$\|u_T\|_\infty$	$\|u_T\|_2$
MINLP-based [35]	Optimal Value	1.0819	0.4930	0.4171
	CPU Time (s)	10.2590	8.6219	277.7508
Mesh-based	Optimal Value	0.4341	0.2937	0.1746
$(r = 3500)$	CPU Time (s)	9.4357	27.4894	1600.5861

Chapter 5

CONCLUSION AND FUTURE WORK

This work considered the novel design of separating input signals in order to discriminate among a finite number of uncertain nonlinear models, using a two-step approach. First, we developed two methods to over-approximate nonlinear dynamics. In the first method, we use an MINLP-based approach and in the second method, we define a mesh on the function domain of interest and use the properties of the function and to determine bounds on the function interpolation in each mesh element, thus over-approximating the nonlinear function.Then, we proposed one of the first active model discrimination algorithms for uncertain affine models, which includes the affine abstraction, hence, the nonlinear models. Finally, we demonstrated our approach on an example of intent estimation/identification in a lane changing scenario on a highway.

We also extend the mesh based abstraction method to find a piecewise affine abstraction. We divide the domain of interest into smaller subregions that form a cover of the domain with a desired approximation accuracy for each suberegion. On each subregion, the nonlinear dynamics is conservatively approximated by a pair of piecewise affine functions, which brackets the original nonlinear dynamics. Our novel analysis allows for the use of tighter interpolation bounds, thus the proposed abstraction method achieves better time efficiency and requires less subregions for the same desired approximation accuracy when compared to existing approaches. Our method also applies to nonlinear functions with different degree of smoothness. We demonstrated the advantages of our approach in simulation and illustrated its applicability for the problem of active model discrimination. Future works will explore partitioning the domain of interest into subregions with a non-uniform, non-rectangular mesh, e.g., simplicial mesh, with the objective of improving the approximation quality and accuracy.

REFERENCES

[1] Vadim Alimguzhin, Federico Mari, Igor Melatti, Ivano Salvo, and Enrico Tronci. Linearizing discrete-time hybrid systems. *IEEE Transactions on Automatic Control*, 62(10):5357–5364, 2017.

[2] Matthias Althoff, Olaf Stursberg, and Martin Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[3] Eugene Asarin, Thao Dang, and Antoine Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Int. Workshop on Hybrid Systems: Computation and Control*, pages 20–35. Springer, 2003.

[4] Eugene Asarin, Thao Dang, and Antoine Girard. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, 43(7):451–476, 2007.

[5] Shun-ichi Azuma, Imura Jun-ichi, and Sugie Toshiharu. Lebesgue piecewise affine approximation of nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 4(1):92–102, 2010.

[6] M. Babaali and M. Egerstedt. Observability of switched linear systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 48–63. Springer, 2004.

[7] Stabley Bak, Sergiy Bogomolov, Thomas A. Henzinger, Taylor T. Johnson, and Prakash Pradyot. Scalable static hybridization methods for analysis of nonlinear systems. In *the 19th International Conference on Hybrid Systems: Computation and Control*, pages 155–164. Springer, 2016.

[8] S. Cheong and I.R. Manchester. Input design for discrimination between classes of LTI models. *Automatica*, 53:103–110, 2015.

[9] IBM ILOG CPLEX. V12. 1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.

[10] Thao Dang, Oded Maler, and Romain Testylier. Accurate hybridization of nonlinear systems. In *ACM International Conference on Hybrid Systems: Computation and Control*, pages 11–20, 2010.

[11] Y. Ding, F. Harirchi, S. Z. Yong, E. Jacobsen, and N. Ozay. Optimal input design for affine model discrimination with applications in intention-aware vehicles. In *ACM/IEEE International Conference on Cyber-Physical Systems*, 2018. Available from: arXiv:1702.01112.

[12] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[13] Antoine Girard and Samuel Martin. Synthesis for constrained nonlinear systems using hybridization and robust controller on symplices. *IEEE Transactions on Automatic Control*, 57(4):1046–1051, 2012.

[14] M. Grewal and K. Glover. Identifiability of linear and nonlinear dynamical systems. *IEEE Trans. Aut. Cont.*, 21(6):833–837, 1976.

[15] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015.

[16] Zhi Han and Bruce H. Krogh. Reachability analysis of nonlinear systems using trajectory piecewise linearized models. In *the American Control Conference*, pages 1505–1510, 2006.

[17] F. Harirchi and N. Ozay. Model invalidation for switched affine systems with applications to fault and anomaly detection. *IFAC-PapersOnLine*, 48(27):260–266, 2015.

[18] F. Harirchi and N. Ozay. Guaranteed model-based fault detection in cyber-physical systems: A model invalidation approach. `arXiv:1609.05921 [math.OC]`, 2016.

[19] F. Harirchi, S. Z. Yong, E. Jacobsen, and N. Ozay. Active model discrimination with applications to fraud detection in smart buildings. In *IFAC World Congress, Toulouse, France*, 2017.

[20] F. Harirchi, S. Z. Yong, and O. Necmiye. *Passive Diagnosis of Hidden-Mode Switched Affine Models with Detection Guarantees via Model Invalidation.* Springer, 2018. To appear.

[21] F. Harirchi, S. Z. Yong, and N. Ozay. Guaranteed fault detection and isolation for switched affine models. In *IEEE CDC*, 2017.

[22] P. Kumar, M. Perrollaz, S. Lefvre, and C. Laugier. Learning-based approach for online lane change intention prediction. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 797–802, June 2013.

[23] C.P. Lam and S. S. Sastry. A POMDP framework for human-in-the-loop system. In *IEEE CDC*, pages 6031–6036, 2014.

[24] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *CACSD*, Taipei, Taiwan, 2004.

[25] H. Lou and P. Si. The distinguishability of linear control systems. *Nonlinear Analysis: Hybrid Systems*, 3(1):21–38, 2009.

[26] Regina Hunter Mladineo. An algorithm for finding the global maximum of a multimodal, multivariate function. *Mathematical Programming*, 34(2):188–200, 1986.

[27] T.H. Nguyen, D. Hsu, W.S. Lee, T.Y. Leong, L.P. Kaelbling, T. Lozano-Perez, and A.H. Grant. Capir: Collaborative action planning with intention recognition. *arXiv preprint arXiv:1206.5928*, 2012.

[28] R. Nikoukhah and S. Campbell. Auxiliary signal design for active failure detection in uncertain linear systems with a priori information. *Automatica*, 42(2):219–228, 2006.

[29] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Trans. on Aut. Cont.*, 58(11):2715–2729, November 2013.

[30] Nacim Ramdani, Nacim Meslem, and Yves Candau. A hybrid bounding method for computing an over-approximation for the reachable set of uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 54(10):2352–52364, 2009.

[31] P. Rosa and C. Silvestre. On the distinguishability of discrete linear time-invariant dynamic systems. In *IEEE CDC-ECC*, pages 3356–3361, 2011.

[32] D. Sadigh, S.S. Sastry, S. Seshia, and A. Dragan. Information gather-ing actions over human internal state. In *IEEE/RSJ IROS*, Oct. 2016.

[33] J.K. Scott, R. Findeisen, R. D Braatz, and D.M. Raimondo. Input design for guaranteed fault diagnosis using zonotopes. *Automatica*, 50(6):1580–1589, 2014.

[34] M. Šimandl and I. Punčochář. Active fault detection and control: Unified formulation and optimal design. *Automatica*, 45(9):2052–2059, 2009.

[35] Kanishka Singh, Yuhao Ding, Necmiye Ozay, and Sze Zheng Yong. Input design for nonlinear model discrimination via affine abstraction. In *IFAC Conference on Analysis and Design of Hybrid Systems*, pages 1–8, 2018. accepted.

[36] Martin Stämpfle. Optimal estimates for the linear interpolation error for simplices. *Journal of Approximation Theory*, 103:78–90, 2000.

[37] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer, 2009.

[38] S. Z. Yong, M. Zhu, and E. Frazzoli. Generalized innovation and inference algorithms for hidden mode switched linear stochastic systems with unknown inputs. In *IEEE CDC*, pages 3388–3394, Dec. 2014.

[39] S. Z. Yong, M. Zhu, and E. Frazzoli. Resilient state estimation against switching attacks on stochastic cyber-physical systems. In *IEEE CDC*, pages 5162–5169, Dec. 2015.

[40] B.D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J.A. Bagnell, M. Hebert, A.K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *IEEE/RSJ IROS*, pages 3931–3936, 2009.

APPENDIX

Time-Concatenated Matrices and Vectors

$$M_i = \begin{bmatrix} \underline{A}_i & -I & 0 & 0 & \cdots & 0 \\ -\overline{A}_i & I & 0 & 0 & \cdots & 0 \\ 0 & \underline{A}_i & -I & 0 & \cdots & 0 \\ 0 & -\overline{A}_i & I & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \underline{A}_i & -I \\ 0 & 0 & \cdots & \cdots & -\overline{A}_i & I \end{bmatrix},\ F_i = \begin{bmatrix} \underline{f}_i \\ -\overline{f}_i \\ \underline{f}_i \\ -\overline{f}_i \\ \vdots \\ \underline{f}_i \\ -\overline{f}_i \end{bmatrix},$$

$E_i = \underset{T}{\operatorname{diag}}\{C_i\}, G_i = \underset{T}{\operatorname{vec}}\{g_i\}.$

For $\star = \{d, w, u\}$:

$$\Gamma_{\star,i} = \begin{bmatrix} \underline{B}_{\star,i} & 0 & 0 & \cdots & 0 \\ -\overline{B}_{\star,i} & 0 & 0 & \cdots & 0 \\ 0 & \underline{B}_{\star,i} & 0 & \cdots & 0 \\ 0 & -\overline{B}_{\star,i} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \underline{B}_{\star,i} \\ 0 & 0 & \cdots & \cdots & -\overline{B}_{\star,i} \end{bmatrix}, F_{\star,i} = \underset{T}{\operatorname{diag}}\{D_{\star,i}\}.$$

Pair-Concatenated Matrices and Vectors

$$M^\iota = \underset{i,j}{\operatorname{diag}}\{M_i\}, \Gamma^\iota_u = \underset{i,j}{\operatorname{vec}}\{\Gamma_{u,i}\}, \Gamma^\iota_d = \underset{i,j}{\operatorname{diag}}\{\Gamma_{d,i}\},$$
$$\Gamma^\iota_w = \underset{i,j}{\operatorname{diag}}\{\Gamma_{w,i}\}, F^\iota_T = \underset{i,j}{\operatorname{vec}}\{F_i\}, E^\iota = \underset{i,j}{\operatorname{diag}}\{E_i\},$$
$$F^\iota_u = \underset{i,j}{\operatorname{vec}}\{F_{u,i}\}, F^\iota_d = \underset{i,j}{\operatorname{diag}}\{F_{d,i}\},$$
$$F^\iota_v = \underset{i,j}{\operatorname{diag}}\{F_{v,i}\}, G^\iota = \underset{i,j}{\operatorname{vec}}\{G_i\}.$$

Matrices and Vectors in Theorem 3

$$R_1^\iota = \begin{bmatrix} M_i & 0 & \Gamma_{d,i} & 0 & \Gamma_{w,i} & 0 & 0 & 0 \\ 0 & M_j & 0 & \Gamma_{d,j} & 0 & \Gamma_{w,j} & 0 & 0 \end{bmatrix},$$

$$r_1^\iota = \begin{bmatrix} -F_i \\ -F_j \end{bmatrix},\ S_1^\iota = \begin{bmatrix} -\Gamma_{u,i} \\ -\Gamma_{u,j} \end{bmatrix},$$

$$R_2^\iota = \begin{bmatrix} E_i & -E_j & F_{d,i} & -F_{d,j} & 0 & 0 & F_{v,i} & -F_{v,i} \\ -E_i & E_j & -F_{d,i} & F_{d,j} & 0 & 0 & -F_{v,i} & F_{v,i} \end{bmatrix},$$

$$r_2^\iota = \begin{bmatrix} G_j - G_i \\ G_i - G_j \end{bmatrix},\ S_2^\iota = \begin{bmatrix} F_{u,j} - F_{u,i} \\ F_{u,i} - F_{u,j} \end{bmatrix}.$$