Development of Safety control for Hidden Mode Hybrid Systems and Verification in the Multi-vehicle lab

by

Rajeev Verma

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2011

Doctoral Committee:

Assistant Professor Domitilla Del Vecchio, Co-Chair (Massachusetts Institute of Technology, Cambridge)
Professor Jim Freudenberg, Co-Chair
Professor Jessy Grizzle
Professor Stephane Lafortune
Professor Jing Sun

I dedicate this thesis to my late mom for her love and sacrifice; to my father, sister, brother-in-law and Kapoor mausa for their encouragement and support and to my dear wife Kirti for her optimism and inspiration.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis advisor Prof. Domitilla Del Vecchio. It is impossible to put in words my gratitude and appreciation toward her. She has been a wonderful mentor and a great inspiration. I strive to copy her attitude towards life and work and I consider this effort a worthwhile goal for the rest of my life. I would like to thank my dissertation committee members, Prof. Jim Freudenberg, Prof. Jessy Grizzle, Prof. Stephane Lafortune and Prof. Jing Sun for their invaluable guidance, support and feedback. I also thank Becky Turanski, Beth Lawson and Ann Pace in the Systems office for their kind help which has always been accompanied with a big smile. I am grateful to my colleagues including Mike Hafner, Shridhar Jayanthi, Reza Ghaemi and Vishnu Desaraju for their friendship over the years. I owe special thanks to my dear friends, Devesh, Gayathri and Dipankar, for their devoted friendship which has been and will always be invaluable to me.

My family has been my greatest support and source of strength throughout this period. I have dedicated my thesis to my family and specially to my Mom, who passed away last year. I believe it is her blessing and sacrifice that saw me through my life's challenges. I also wish to thank my father, who has always encouraged me to pursue my ambitions. I would also like to express my gratitude towards my sister, Venu, and my brother-in-law, Ramit. Without them, it would have been impossible to start this endeavor and stay focused. I thank Kapoor Mausa for being a great inspiration to me since my childhood. I also thank my in-laws, Mom, Dad, Ashwani and Deepa for their blessings, love and support. Finally, I want to thank my wife, Kirti, for being there through all the ups and downs. Without her, I would be lost.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Development of Safety control for Hidden Mode Hybrid Systems and Verification in the Multi-vehicle lab

by

Rajeev Verma

Co-Chairs: Domitilla Del Vecchio and Jim Freudenberg

In this thesis, we consider the safety control problem for Hidden Mode Hybrid Systems (HMHS), which are a special class of hybrid automata in which the mode is not available for control. For these systems, safety control is a problem with imperfect state information. We tackle this problem by introducing the notion of non-deterministic discrete information state and by then translating the problem to one with perfect state information. The perfect state information control problem is obtained by constructing a new hybrid automaton, whose discrete state is an estimate of the HMHS mode and is thus available for control. This problem is solved by computing the capture set and the least restrictive control map for the new hybrid automaton. Sufficient conditions for the termination of the algorithm that computes the capture set are provided. We show that the solved perfect state information control problem is equivalent to the original problem with imperfect state information under suitable assumptions on the original HMHS.

A multi-vehicle roundabout test-bed is developed that employs scaled vehicles that are designed to have longitudinal dynamical response similar to a full scale vehicle. The application of the proposed formal hybrid control approach to the collision avoidance problem

between an autonomous vehicle and a human driven vehicle at a traffic intersection is experimentally illustrated in the multi-vehicle test-bed. We model the human driving behavior through a hybrid automaton, whose current mode is determined by the driver's decisions. On the autonomous vehicle, we implement formal methods for safety control, in which a mode estimator identifies in real time the current human driving behavior and uses this information to update a hybrid feedback map. The experimental results demonstrate that the solution proposed in this thesis is substantially less conservative than solutions employing worst-case design. Furthermore, they also demonstrate that, in structured tasks, human behavior can be reliably modeled and recognized for safety-critical closed loop control applications.

CHAPTER I

Introduction

Hidden Mode Hybrid Systems (HMHSs) are a special class of hybrid automata [52, 68], in which the mode is unknown and mode transitions are driven only by disturbance events. There are a large number of applications that can be well described by hybrid automata models, in which it is not realistic to assume knowledge of the mode. This is the case, for example, of intent-based conflict detection and avoidance for aircrafts, in which the intent of aircrafts in the environment is unknown and needs to be estimated (see [82] and the references therein). In robotic games such as RoboFlag [32, 27], the intents of non-team members are unknown and need to be identified to allow decisions toward keeping the home zone safe. Next generation warning and active safety systems for vehicle collision avoidance will have to guarantee safety in the presence of human drivers and pedestrians, whose intentions are unknown [1]. More generally, in a variety of multi-agent systems, for example assistive robotics, computer games, and robot-human interaction, the intentions of an observed agent are unknown and need to be identified for control [37].

There has been a wealth of research on safety control for hybrid systems in which the state is known [64, 88, 87, 68, 85, 43, 47, 11]. In [88, 87, 68, 85], the safety control problem is elegantly formulated in the context of optimal control and leads to the Hamilton-Jacobi-Bellman (HJB) equation. This equation implicitly determines the maximal controlled invariant set and the least restrictive feedback control map. Due to the complexity of exactly

solving the HJB equation, researchers have been investigating approximate algorithms for computing inner-approximations of the maximal controlled invariant set [88, 53, 54, 81]. Termination of the algorithm that computes the maximal controlled invariant set is often an issue and work has been focusing on determining special classes of systems that allow to prove termination [85, 83, 84]. The safety control problem for hybrid systems has also been investigated within a viability theory approach by a number of researchers [47, 11].

The safety control problem for hybrid systems when the mode is not available for feedback has been rarely addressed in the literature. The safety control problem in the case when the set of observations is a partition of the state space was discussed by [80]. The problem is first transformed into a game of perfect information and a controller with memory is derived. The proposed algorithm can deal with a system with finite number of states. It excludes important classes of systems such as timed and hybrid automata. A number of recent works have addressed the safety control problem for special classes of hybrid systems with imperfect state information [95, 31, 50, 33, 29]. In [95], a controller that relies on a state estimator is proposed for finite state systems. The results are then extended to control a class of rectangular hybrid automata with imperfect state information, which can be abstracted by a finite state system. In [31, 50, 33], linear complexity state estimation and control algorithms are proposed for special classes of hybrid systems with order preserving dynamics. In particular, discrete time models are considered in [31, 29] while continuous time models are considered in [50, 33]. In these works, the mode is assumed to be known and only continuous state uncertainty is considered.

Here, we consider the safety control problem for HMHSs, in which the mode is unknown and its transitions are driven only by uncontrollable and unobservable events. The continuous state is measured and the control input enters its dynamics, but these dynamics are driven also by the mode and a continuous disturbance input. For this class of systems, designing a controller to guarantee safety is a control problem with imperfect state information. Within this context, a controller is a *dynamic* feedback map from the system history

and as a consequence notions of maximal controlled invariant set and least restrictive feedback map need to be revisited. In the theory of games, control problems with imperfect state information have been elegantly addressed by translating them to problems with perfect state information [66, 63]. This transformation is obtained by introducing the notion of derived information state (non-deterministic or probabilistic), which, in the case of the non-deterministic information state, keeps track of the set of all possible current states compatible with the system history up to the current time. In the case in which a recursive update law can be constructed for the derived information state, the control problem can be described completely in terms of this new state. Since the derived information state is known, the problem becomes one with perfect state information. Among the difficulties of this approach is the construction of a recursive update law for the non-deterministic information state as this in general is not guaranteed to exist in any simple form suitable for analysis [66, 63].

We utilize the notion of non-deterministic discrete information state for a HMHS and formulate the safety control problem in terms of this derived information state. We translate this problem to one with perfect state information by introducing a new hybrid system called an estimator, which updates a discrete state estimate in the form of a set of possible discrete states. We only require that the discrete state estimate is correct, that is, that it contains the current mode of the original HMHS at any time, while we are not concerned with tightness or convergence guarantees [34]. This ensures that an estimator always exists and allows to *separate* the estimation problem from the control problem. Since the estimator state is measured, the original control problem becomes one with perfect state information.

We solve the new perfect state information control problem by providing an algorithm to determine the capture set (the complement of the maximal controlled invariant set) and the least restrictive control map. Then, we provide sufficient conditions for the termination of the algorithm that determines the capture set. We further illustrate how to construct an abstraction of the estimator for which the algorithm that determines the capture set always

terminates and has as fixed point the capture set of the estimator. Finally, we tackle the question of how the perfect state information problem that we have solved is related to the original problem with imperfect state information. Under a structural assumption and a mode distinguishability assumption on the original HMHS, we show that the two problems are equivalent, that is, their solution gives the same capture sets and control maps.

The problem considered has much in common with two-person repeated games of incomplete information, in which one player is informed about the environment state while the other is not [12, 49]. In these types of games, the informed player must take into account how his/her actions may reveal information that will affect future payoffs. The control of a HMHS can be viewed as a game between the controller (uninformed agent) and the disturbance (informed agent), in which the actions of the latter can reveal information on the current mode of the hybrid automaton. In turn, a given information/uncertainty on the current mode corresponds to the disturbance taking values in a restricted range of its domain. Therefore, even if the disturbance acts not to reveal information on the mode, it may still indirectly reveal information about the actions it is constrained to take. This information aids the controller as the controller now has to counteract only a restricted range of possible disturbance actions. The equivalence result implies that the best strategy for the disturbance is simply to keep the maximal uncertainty possible on the mode. In doing so, it will in fact not reveal useful information to the controller regarding its range of action.

The second part of this thesis focuses on the problem of building a laboratory-scale vehicle whose longitudinal dynamics are similar to those of a full-size High-Mobility Multipurpose Wheeled Vehicle (HMMWV). This laboratory-scale vehicle uses Hardware-in-the-Loop (HIL) simulation, in the sense of matching key dynamics of the full-size HMMWV (e.g., its inertia) physically, while using an onboard processor to simulate the rest. This is performed with the ultimate goal of developing a scaled experimental testbed. This testbed is used to validate safety control algorithms for HMHS, that form a part of the intelligent transportation systems (ITS) applications.

ITS include a wide range of systems from the basic cruise control system, to the more advanced adaptive cruise control system, to more complex systems that exploit embedded wireless communication technology. These systems include cooperative intersection collision avoidance systems, lateral collision avoidance systems, and longitudinal collision avoidance systems [16, 58]. In response to the highway incident statistics [72], several major automotive companies have established research programs focusing on cooperative safety systems [65]. These systems are conceived with three different levels of automation: advise or warn the driver, partially control the vehicle, and fully control the vehicle in emergency situations. Testing autonomous or partly autonomous algorithms directly on a full scale transportation system is challenging due to cost limitations and safety constraints. We thus develop a lab-scale testbed composed of 1/13 scale vehicles to validate decision and control algorithms for cooperative intersection collision avoidance systems. In such a testbed, the vehicles are equipped with wireless communication, with a positioning system emulating GPS, and with an on-board computer solving decision, control, and communication tasks. The vehicle's longitudinal dynamics play a central role in collision avoidance algorithms. For a meaningful algorithm validation, it is therefore crucial to design scaled vehicles whose dynamics are a faithful reproduction of the longitudinal dynamics of a full scale vehicle.

The scaled vehicle hardware is composed only of the chassis including wheels, tires, axis, and a DC motor with encoder. The unavailability of an exact scaled replica of engine or transmission makes it impossible to include a scaled physical drivetrain on the prototype. Therefore, a HIL setup is designed in which a microprocessor controlling the DC motor emulates the scaled drivetrain dynamics of a HMMWV including engine and transmission. The software coded in the microprocessor takes as input the throttle command by the on-board computer and applies voltage commands to the DC motor in order to obtain the desired drive torque at the wheels. The net result of such a HIL setup is that the system composed of the software on the microprocessor and the DC motor takes as input a throttle

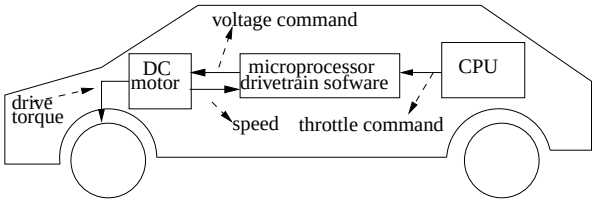


Figure 1.1: HIL setup. The hardware of the vehicle includes chassis, wheels, axis, and a DC motor with encoder. The scaled drivetrain dynamics is implemented on the microprocessor controlling the DC motor.

command and applies to the wheels the desired drive torque. This way, we are able to obtain a scaled vehicle that as a whole responds to throttle commands in a way similar to the full scale vehicle. The HIL setup is shown in Figure 1.1. Scaling of the drivetrain dynamics is performed by applying well known concepts from scaling theory, including the Buckingham π theorem and π groups. Scaling of active components such as engine and transmission is difficult to achieve in hardware. Thus we come up with a HIL setup and the scaling of these components is carried out in software.

Researchers have been studying scaled vehicles since 1930s for different reasons, such as trailer sway [40], vehicle dynamics [96, 15], performance on rough terrain and to determine vehicle turning radius [15], for automobile accident reconstruction [44]. More recently, work has been reported on vehicle dynamics and controls [21, 38, 18, 20, 19, 69, 22, 23], to study the lateral motion and design of steering controller [25, 56, 60], control prototyping of Anti-lock Braking System (ABS) [70, 76, 67, 75], and to study vehicle rollover [89, 94]. The work that exists in literature has focused mostly on lateral dynamics. The unique contribution of this work is the demonstration of longitudinal dynamics scaling of a vehicle with all the active powertrain subsystems present in it using HIL approach. In making this contribution, we build on well-established HMMWV powertrain and vehicle dynamics models, scaling techniques, and HIL simulation techniques to create a unique scaled testbed for ITS applications. The validation experiments confirm that the longitudinal response of the scaled vehicle matches the longitudinal response of a full scale

vehicle.

Similitude research has been used for over a century to study the behavior of systems that are difficult to analyze in their original size and normal operating environment. Typically, such research uses scaled models that are dynamically similar to a system much larger than the model. A historic account of development of similitude theory can be found in [23]. Many published studies on this topic are available [78, 22, 23, 77]. Using similitude theory, dynamics of a system can be studied in terms of dimensionless parameters. An important contributor in the development of this theory is Buckingham [24]. He formulated a theorem, called the π theorem, that can be used to study the scaling properties of any system. See the Appendix A.1 for more details.

In the third part of this thesis, we consider the application scenario of collision avoidance between two vehicles, one human driven and the other autonomous. We apply the safety control approach, developed in the first part of this work, to the semi-autonomous intersection system scenario of Figure 4.1, in which an autonomous vehicle and a human-driven vehicle are engaged in a conflict resolution problem at a traffic intersection. We model human-driving behavior in the proximity of an intersection through a hybrid automaton that can be in either of two modes: acceleration mode and braking mode. There is a rich literature about the classification through dynamical models of human behavior in structured tasks (see, for example [36, 35, 92] and the references therein). These works show that human behavior can be reliably recognized provided certain identifiability assumptions are satisfied, which depend on the system dynamical model and on how coarse/fine the behavior classification is. The human-driving behavior parameters learned through a process of supervised learning are then used by the dynamic feedback map implemented in the autonomous vehicle. A mode estimator, based on position measurements (as obtained, for example, from road-side sensors) determines the current driving mode of the human-driven vehicle. The automatic vehicle, on the basis of the current mode uncertainty, determines the control map that guarantees that the current system configuration is kept

outside of the current mode-dependent capture set. This results in safe throttle commands applied to the autonomous vehicle.

This thesis is organized as follows. In Chapter 2, the problem of safety control of HMHS is discussed. The solution to the problem is presented along with the application example of collision avoidance between a human driven and an autonomous vehicle at a traffic intersection. In Chapter 3, the development of a $\frac{1}{13}$ scale vehicle that has longitudinal dynamics similar to that of a HMMWV is presented. In Chapter 4, results from experimental validation of safety control algorithm of Chapter 2 are presented. These experiments are conducted in the Multi-vehicle lab that utilizes the in-scale vehicle hardware discussed in Chapter 3. Chapter 5 concludes the thesis and discusses possible extensions of the current work.

CHAPTER II

Safety Control of Hidden Mode Hybrid Systems

In this chapter, we define the problem of safety control of hybrid systems with hidden modes and propose a solution for the problem. In Section 2.1, we recall basic definitions and concepts. In Section 2.2, we introduce the HMHS model and its information structure. In Section 2.3, we introduce the control problem with imperfect state information (Problem 1) and its translation to a problem with perfect state information (Problem 2). We then provide the solution to Problem 2 in Section 2.4. We consider the problem of termination in Section 2.5. In Section 2.6, we show the equivalence of Problem 1 and Problem 2. In Section 2.7, we illustrate the application of the proposed control algorithms to a collision avoidance problem at a traffic intersection.

2.1 Basic notions and definitions

In this section, we introduce some basic notions and definitions. We employ basic notions from partial order theory [28]. A *partial order* is a set P with a partial order relation "$\leq$" and it is denoted by $(P, \leq)$. If any two elements in P have a unique supremum and a unique infimum in P, then P is a lattice. If $(P, \leq)$ is a lattice, we denote for any subset $S \subseteq P$ its supremum by $\bigvee S$. For a set X, we denote by 2^X the power set, that is, the set of all subsets of X. In this thesis, we consider the lattice given by 2^X with order established by set inclusion. This lattice is denoted by $(2^X, \subseteq)$. For any subset $S \subseteq 2^X$, the supremum

$\bigvee S$ is given by the union of all sets in S. Another partial order that is considered in this work is given by $\mathbb{R}^n$ with order established component-wise, that is, for $x = (x_1, ..., x_n) \in \mathbb{R}^n$ and $w = (w_1, ..., w_n) \in \mathbb{R}^n$, we say that $x \leq w$ provided $x_i \leq w_i$ for all $i \in \{1, ..., n\}$. We denote this partial order by $(\mathbb{R}^n, \leq)$. Let $(P, \leq)$ be a lattice, an interval in P is denoted by $[L, U] := \{p \in P \mid L \leq p \leq U\}$. For any vector $v \in \mathbb{R}^n$, we denote by v_i its ith component. Let $\mathbb{R}_+$ denote the set of non-negative real numbers and let $\mathbf{u} : \mathbb{R}_+ \to \mathbb{R}$ denote a signal with values in $\mathbb{R}$. Denote the set of all such signals by $\mathcal{S}(\mathbb{R})$. We define a partial order on this space of signals as follows. For any two signals $\mathbf{u}, \mathbf{w} \in \mathcal{S}(\mathbb{R})$, we say that $\mathbf{u} \leq \mathbf{w}$ provided $u(t) \leq w(t)$ for all $t \in \mathbb{R}$. Let $(P, \leq)$ and $(Q, \leq)$ be two partial orders and consider the map $f : P \to Q$. This map is said to be an *order preserving map* if for all $p_1, p_2 \in P$ such that $p_1 \leq p_2$, we have that $f(p_1) \leq f(p_2)$. It is said to be a *strongly order preserving map* if for all $p_1, p_2 \in P$ such that $p_1 < p_2$, we have that $f(p_1) < f(p_2)$. For any map $f : P \to Q$ and any subset $S \subseteq P$, we define $f(S) := \bigcup_{p \in S} f(p)$.

Notions from viability theory as found in [10] are here recalled. Let X be a normed space and let $S \subset X$ be nonempty. The *contingent cone* to S at $x \in S$ is the set given by $T_S(x) := \{v \in S \mid \liminf_{h \to 0+} \frac{d_S(x+h\,v)}{h} = 0\}$, in which $d_S(y)$ denotes the distance of y from set S, that is, $d_S(y) := \inf_{z \in K} \|y - z\|$. When S is an open set, the contingent cone to S at any point in S is always equal to the whole space.

A set valued map $F : X \to 2^X$ is said to be *Marchaud* provided (i) the graph and the domain of F are nonempty and closed; (ii) for all $x \in X$, $F(x)$ is compact, convex and nonempty; (iii) F has linear growth, that is, there exist $\alpha > 0$ such that for all $x \in X$ we have $\sup\{\|v\| \mid v \in F(x)\} \leq \alpha(\|x\| + 1)$.

A set valued map $F : X \to 2^X$ is said to be *Lipschitz* continuous on X if there is $\lambda > 0$ such that for all $x_1, x_2 \in X$ we have that $F(x_1) \subseteq F(x_2) + \lambda \|x_1 - x_2\| B_1(0)$, in which $B_1(0)$ is a ball in X of radius 1 centered at 0.

2.2 Hidden Mode Hybrid Systems

A hybrid system model with hidden modes is a hybrid automaton [68] in which the current mode of the system is unknown and mode transitions are driven by disturbance events only. This model is formally introduced by the following definitions.

Definition II.1. A *hybrid system with uncontrolled mode transitions* is a tuple $H = (Q, X, U, D, \Sigma, R, f)$, in which Q is a finite set of modes; $X \subseteq \mathbb{R}^n$ is a vector space; $U = [u_L, u_H] \subseteq \mathbb{R}$, with $u_L, u_H \in \mathbb{R}$ is a set of control inputs; $D = [d_L, d_H] \subseteq \mathbb{R}$, with $d_L, d_H \in \mathbb{R}$ is a bounded set of disturbance inputs; Σ is a finite set of disturbance events, which includes a silent event denoted ϵ; $R : Q \times \Sigma \to Q$ is the discrete state update map; $f : X \times Q \times U \times D \to X$ is the vector field, which is piecewise continuous on $X \times U \times D$.

The vector field f is allowed to be piecewise continuous in order to model switches in the dynamics determined by submanifolds in the space of states and inputs. We denote by $(q, x) \in Q \times X$ the hybrid state of the system. Similarly, we denote by $(u, d) \in U \times D$ the continuous inputs to the system and by $\sigma \in \Sigma$ the disturbance event. We define $R(q, \epsilon) := q$ for all $q \in Q$. Let $\{\tau'_i\}_{i\in I} \subset \mathbb{R}$ for $I = \{0, 1, 2, ...\}$ with $\tau'_i \leq \tau'_{i+1}$ be the sequence of times at which $\sigma(\tau'_i) \in \Sigma/\epsilon$ and $\sigma(t) = \epsilon$ for $t \notin \{\tau'_i\}_{i\in I}$. Let $\mathcal{T} := \bigcup_{i\in I}[\tau_i, \tau'_i)]$ in which $\tau_i \leq \tau'_i = \tau_{i+1}$ with $\tau_0 = 0$, and the ")]" parenthesis is closed ("]") if τ'_i is finite and open (")") if it is not finite. Then, we define the discrete and continuous trajectories of H, that is, $q(t)$ and $x(t)$ for $t \in \mathcal{T}$ as follows.

Definition II.2. Given initial conditions $(q_o, x_o) \in Q \times X$,

> *the discrete trajectory* $q(t)$ for $t \in \mathcal{T}$ is such that $q(\tau_{i+1}) = R(q(\tau'_i), \sigma(\tau'_i))$ and $q(t) = q(\tau_i)$ for $t \in [\tau_i, \tau'_i]$ if $\tau_i < \tau'_i$ with $q(\tau_0) = q_o$;
>
> *the continuous trajectory* $x(t)$ for $t \in \mathcal{T}$ is such that $\dot{x}(t) = f(x(t), q(t), u(t), d(t))$, $d(t) \in D$ for $t \in [\tau_i, \tau'_i]$ with $\tau_i < \tau'_i$ and $x(\tau_{i+1}) = x(\tau'_i)$ with $x(\tau_0) = x_o$.

Since we can have that $\tau_i' = \tau_{i+1}$, multiple discrete transitions can occur at one time. The value of x immediately before and immediately after a set of transitions occurring at the same time is unchanged. Basically, a hybrid system with uncontrolled mode transitions is a hybrid automaton [52] in which there is no continuous state reset and the discrete transitions cannot be controlled. The vector field f immediately after a set of transitions occurring at the same time t is evaluated on the value that q takes after the last transition occurred at time t. It is therefore useful to define also the discrete and continuous flows of H as follows. Let $\boldsymbol{\sigma} : \mathcal{T} \to \Sigma$, $\mathbf{u} : \mathcal{T} \to U$, and $\mathbf{d} : \mathcal{T} \to D$ be the disturbance event, the continuous control, and the continuous disturbance signals. We assume that the signal $\mathbf{u}$ is a piecewise continuous signal which is continuous from the right, that is, $\lim_{\delta \to 0^+} u(t+\delta) = u(t)$ for all $t \in \mathcal{T}$.

Definition II.3. For initial condition $(q_o, x_o) \in Q \times X$,

the discrete flow is defined as $\phi_q(t, q_o, \boldsymbol{\sigma}) := q(\sup_{\tau_i \leq t} \tau_i)$ for all $t \geq 0$;

the continuous flow is defined as $\phi_x(t, (q_o, x_o), \mathbf{u}, \mathbf{d}, \boldsymbol{\sigma}) := x(t)$ in which $\dot{x}(t) = f(x(t), \phi_q(t, q_o, \boldsymbol{\sigma}), u(t), d(t))$, $d(t) \in D$ for all $t \geq 0$.

Therefore, $\phi_q(t, q_o, \boldsymbol{\sigma})$ is a piecewise constant signal that at time t takes the value of q at the last transition that occurred before or at time t. When $\sigma(t) = \epsilon$ for all t, we denote the corresponding continuous flow by $\phi_x(t, (q_o, x_o), \mathbf{u}, \mathbf{d}, \epsilon)$.

Definition II.4. A *Hidden Mode Hybrid System* (HMHS) is a hybrid system with uncontrolled mode transitions in which $q(t)$ is not measured and q_o is only known to belong to a set $\bar{q}_o \subseteq Q$.

Therefore, in a HMHS only $x(t)$ is measured and its evolution is driven by hidden mode transitions. In the reminder of this thesis, H denotes a HMHS.

Definition II.5. Let $\bar{q} \subseteq Q$. The set of modes *reachable* from $\bar{q}$ under the trajectories of H is denoted $\text{Reach}(\bar{q}) \subseteq Q$ and is defined as $\text{Reach}(\bar{q}) := \bigcup_{q_o \in \bar{q}} \bigcup_{t \geq 0} \bigcup_{\boldsymbol{\sigma}} \phi_q(t, q_o, \boldsymbol{\sigma})$.

Note that since hybrid system H allows multiple transitions at the same time, any mode q can instantaneously transit to any element in its reachable set Reach(q).

2.2.1 The non-deterministic discrete information state

For a signal $\mathbf{s} : \mathbb{R}_+ \to S$, we define its truncation up to time t as $\mathbf{s}_t : [0,t] \to S$ and its truncation up to time t^- as $\mathbf{s}_{t^-} : [0,t) \to S$. At time t, the measured signals of H are given by $\mathbf{u}_{t^-}$ and $\mathbf{x}_t$, in which $\mathbf{x_0} := x_o$. Furthermore, the knowledge of the function $\mathbf{x}_t : [0,t] \to X$ implies that also the function $\dot{\mathbf{x}}_{t^-} : [0,t) \to X$ is known.

Definition II.6. The *history* of system H at time t for $t \geq 0$ is defined as $\boldsymbol{\eta}(t) := (\bar{q}_o, \mathbf{u}_{t^-}, \mathbf{x}_t, \dot{\mathbf{x}}_{t^-})$, in which for $\bar{q}_o \subseteq Q$ is the initial mode information.

The available *information* on the system mode at time t must be derived from the history signal $\boldsymbol{\eta}(t)$, in which $\eta(0) = (\bar{q}_o, \emptyset, x_o, \emptyset)$ contains information on the initial state of the system. We define the set of all possible current modes of the system compatible with the history. This set is called the non-deterministic discrete information state and is formally defined as follows in analogy to what is performed in the theory of games with imperfect information [66].

Definition II.7. The *non-deterministic discrete information state* at time $t \geq 0$ for system H is the set $\bar{q}(\boldsymbol{\eta}(t)) \subset Q$ defined as

$$\bar{q}(\boldsymbol{\eta}(t)) := \left\{ \begin{array}{r} q \in Q \mid \exists\ q_o \in \bar{q}_o,\ \boldsymbol{\sigma}\ s.t.\ q = \phi_q(t, q_o, \boldsymbol{\sigma}) \\ \text{and } \exists\ \mathbf{d}\ s.t. \dot{x}(\tau) = f(x(\tau), \phi_q(\tau, q_o, \boldsymbol{\sigma}), u(\tau), d(\tau)) \\ \text{for all } 0 \leq \tau < t \end{array} \right\}.$$

Hence, a mode q is possible at time t provided (a) there is a discrete state trajectory starting from a mode in $\bar{q}_o$ that reaches q at time t and (b) such a discrete state trajectory is consistent with the continuous state trajectory up to time t. It follows that $q(t) \in \bar{q}(\boldsymbol{\eta}(t))$ for all t and that $\bar{q}(\boldsymbol{\eta}(0)) = \text{Reach}(\bar{q}_o)$.

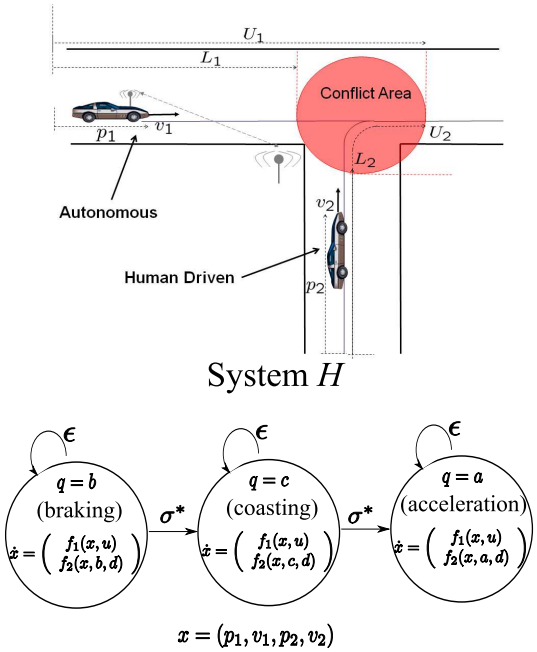


Figure 2.1: (Up) Two-vehicle Conflict Scenario. Vehicle 1 is equipped with a cooperative active safety system and communicates with the infrastructure wirelessly. Vehicle 2 does not communicate with the infrastructure. A collision occurs when both vehicles occupy the conflict area. We refer to vehicle 1 as the "autonomous vehicle" and to vehicle 2 as the "human driven vehicle". (Down) Hybrid automaton model H, in which f_1 and f_2 are given by equations (2.1-2.2).

2.3 Problem Formulation

In this section, we first employ the notion of non-deterministic discrete information state to formulate the safety control problem with imperfect state information. Then, we translate this problem to one with perfect state information by introducing a mode estimator.

2.3.1 Safety control problem with imperfect mode information

Let $Bad \subset X$ represent a set of unsafe continuous states. We consider the problem of determining the set of all initial informations $(\bar{q}_o, x_o)$ for which a *dynamic feedback* map does not exist that maintains the trajectory $x(t)$ outside *Bad* for all time. For this purpose, we first define the closed loop system H under a feedback map $\pi : 2^Q \times X \to U$.

Definition II.8. Consider a feedback map $\pi : 2^Q \times X \to U$. The *closed loop system* H^π is defined as system H, in which $u(t) = \pi(\bar{q}(\boldsymbol{\eta}(t)), x(t))$ for all $t \geq 0$. The continuous flow of H^π is denoted $\phi_x^\pi(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma})$.

The set of all initial informations $(\bar{q}_o, x_o)$ for which there is no feedback map π that maintains the trajectory $\phi_x^\pi(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma})$ outside *Bad* for all $q_o \in \bar{q}_o$, $\boldsymbol{\sigma}$, and $\mathbf{d}$ is called the *capture set* and is formally defined as follows.

Definition II.9. For $Bad \subseteq X$, the *capture set* for system H is defined as $C := \{(\bar{q}_o, x_o) \in 2^Q \times X \mid \forall\ \pi,\ \exists\ q_o \in \bar{q}_o,\ \boldsymbol{\sigma},\ \mathbf{d},\ t \geq 0,\ s.t.\ \phi_x^\pi(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma}) \in Bad\}$.

The following alternative expression of the capture set (obtained directly from the definition) is used in this thesis.

Proposition II.10. *For all $\bar{q} \in 2^Q$, let the mode-dependent capture set be defined as $C_{\bar{q}} := \{x_o \in X \mid \forall\ \pi,\ \exists\ q_o \in \bar{q},\ \boldsymbol{\sigma},\ \mathbf{d},\ t \geq 0,\ s.t.\ \phi_x^\pi(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma}) \in Bad\}$. Then, $C = \bigcup_{\bar{q} \in 2^Q} (\bar{q} \times C_{\bar{q}})$.*

Proposition II.11. *For all $\bar{q} \in 2^Q$, we have that $C_{\bar{q}} = C_{Reach(\bar{q})}$.*

Proof. We first show that $C_{\bar{q}} \subseteq C_{\text{Reach}(\bar{q})}$. Let $x_o \notin C_{\text{Reach}(\bar{q})}$. Then, there is a feedback map π^* such that for all $q_o \in \text{Reach}(\bar{q})$ and $t \geq 0$ we have that $\phi_x^{\pi^*}(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma}) \notin Bad$ for all $\mathbf{d}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\eta}$ with $\boldsymbol{\eta}(0)$ such that $\bar{q}(\boldsymbol{\eta}(0)) = \text{Reach}(\bar{q})$. In particular, such π^* is such that for all $q_o \in \bar{q}$ and $t \geq 0$, $\phi_x^{\pi^*}(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma}) \notin Bad$ for all $\mathbf{d}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\eta}$ with $\boldsymbol{\eta}(0)$ such that $\bar{q}(\boldsymbol{\eta}(0)) = \text{Reach}(\bar{q})$. This, in turn, implies that $x_o \notin C_{\bar{q}}$ from the definition of $C_{\bar{q}}$ and the fact that $\boldsymbol{\eta}(0) = (\bar{q}, \emptyset, x_o, \emptyset)$ implies $\bar{q}(\boldsymbol{\eta}(0)) = \text{Reach}(\bar{q})$.

We then show that $C_{\text{Reach}(\bar{q})} \subseteq C_{\bar{q}}$. Let $x_o \notin C_{\bar{q}}$. Then, there is π^* in which $\bar{q}(\boldsymbol{\eta}(0)) = \text{Reach}(\bar{q})$ such that for all $q_o \in \bar{q}$, $\boldsymbol{\sigma}$, $\mathbf{d}$, we have that $\phi_x^{\pi^*}(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma}) \notin Bad$ for all t. For all $q_j \in \text{Reach}(\bar{q})$, there is $\boldsymbol{\sigma}$ and $q_o \in \bar{q}$ such that $\phi_q(0, q_o, \boldsymbol{\sigma}) = q_j$. Therefore, for any piecewise continuous signal $\phi_q(t, q_o', \boldsymbol{\sigma}')$ with $q_o' \in \text{Reach}(\bar{q})$, we can find $\boldsymbol{\sigma}$ and $q_o \in \bar{q}$ such that $\phi_q(t, q_o, \boldsymbol{\sigma}) = \phi_q(t, q_o', \boldsymbol{\sigma}')$ for all $t \geq 0$. This implies that the feedback map π^* is such that $\phi_x^{\pi^*}(t, (q_o', x_o), \mathbf{d}, \boldsymbol{\sigma}') \notin Bad$ for all t, $\boldsymbol{\sigma}'$, and $q_o' \in \text{Reach}(\bar{q})$. Hence, $x_o \notin C_{\text{Reach}(\bar{q})}$. □

Problem II.12. (Safety Control with Imperfect State Information) Determine the capture set C and the set of feedback maps π such that if $(\bar{q}_o, x_o) \notin C$, then $(\bar{q}(\boldsymbol{\eta}(t)), \phi_x^\pi(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma})) \notin C$ for all $t \geq 0$, $\mathbf{d}$, $\boldsymbol{\sigma}$, and $q_o \in \bar{q}_o$.

2.3.2 Motivating example

In this section, we present an example in the context of cooperative active safety at traffic intersections [1], wherein a controlled vehicle has to prevent a collision with a non-controlled/non-communicating, possibly human-driven, vehicle (Figure 2.1). A possible approach to tackle this problem is to treat the non-communicating vehicle as a "disturbance" and employ available safety control techniques for hybrid systems with measured state. This approach, however, leads to conservative controllers, which are not acceptable as they result in warnings/control actions that the driver perceives as unnecessary. Therefore, in this application it is crucial to exploit all the available sensory information to reduce as much as possible the uncertainty on the non-communicating vehicle. For the controller on board the autonomous vehicle, the human-driven vehicle is a hybrid automaton with unknown state. A related but different application is the one in which a single vehicle can receive inputs from both a human driver and an on-board controller as considered, for example, by [73] in the context of a red-light violation problem. As opposed to our application, the resulting hybrid automaton to control in [73] has known state.

Since both vehicles are constrained to move along their lanes (see Figure 2.1), only the longitudinal dynamics of the vehicles along their respective paths are relevant. The longitudinal dynamics of vehicle 1 along its path are modeled by the equation $\ddot{p}_1 = k_1 u - k_2 v_1^2 - k_3$, in which p_1, v_1 are the longitudinal displacement and speed along the path, respectively, u represents throttle/braking, $k_3 > 0$ represents the static friction term, and $k_2 v_1^2$ with $k_2 > 0$ models air drag (see [90] for more details). The control input u ranges in the interval $[u_L, u_H]$ for given maximum braking action $u_L < 0$ and maximum throttle action $u_H > 0$. For vehicle 2, we assume a model given by $\ddot{p}_2 = \beta_q + d$, in which $d \in [-\bar{d}, \bar{d}]$ for some

$\bar{d} > 0$ and q represents the *unknown* driving mode that can be acceleration mode, denoted a, coasting mode, denoted c, and braking mode, denoted b. For each mode, β_q has a different value representing the nominal acceleration corresponding to that mode. For more details on modeling human (controlled) activities through non-deterministic hybrid systems, the reader is referred to [35, 36]. Vehicle 1 receives information about the position and speed of vehicle 2 from the infrastructure, which monitors speed and position of vehicles through road-side sensors. We assume that there are a lower bound v_{min} and an upper bound v_{max} on the achievable speed of the vehicles due, for example, to physical limitations (i.e., vehicles cannot go in reverse and have a finite maximum achievable speed).

The resulting HMHS $H = (Q, X, U, D, \Sigma, R, f)$ modeling the system is such that $Q = \{a, b, c\}$, $X = \mathbb{R}^4$, $U = [u_L, u_H]$, and $D = [-\bar{d}, \bar{d}]$. Denote $x = (x_1, x_2, x_3, x_4)$ with $x_1 = p_1, x_2 = v_1, x_3 = p_2, x_4 = v_2$. Let $\alpha := k_1 u - k_2 x_2^2 - k_3$. The vector field f is piecewise continuous and given by $f(x, q, u, d) = (f_1(x, u), f_2(x, q, d))$, with

$$f_1(x,u) = \begin{cases} (x_2, \alpha), & \text{if } x_2 \in (v_{min}, v_{max}) \\ (x_2, 0), & \text{if } x_2 \leq v_{min} \text{ and } \alpha < 0 \\ & \text{or } x_2 \geq v_{max} \text{ and } \alpha > 0 \end{cases} \tag{2.1}$$

$$f_2(x,q,d) = \begin{cases} (x_4, \beta_q + d), & \text{if } x_4 \in (v_{min}, v_{max}) \\ (x_4, 0), & \text{if } x_4 \leq v_{min} \text{ and } \beta_q + d < 0 \\ & \text{or } x_4 \geq v_{max} \text{ and } \beta_q + d > 0. \end{cases} \tag{2.2}$$

We assume that the human driven vehicle can transit from acceleration, to coasting, to braking [62]. This scenario can be modeled by $\Sigma = \{\epsilon, \sigma^*\}$ and $R : Q \times \Sigma \to Q$ such that $R(a, \sigma^*) = c$ and $R(c, \sigma^*) = b$. Here, we assume that $\beta_b < 0$, $\beta_c = 0$, and $\beta_a > 0$, with $\bar{d} < |\beta_q| < 2\bar{d}$ for $q \in \{a, b\}$. This system is a HMHS, in which $\bar{q}_o = \{a, b, c\}$ and it is pictorially represented in the right-side plot of Figure 2.1. Finally, the unsafe set is given by $Bad = \{x \mid (x_1, x_3) \in [L_1, U_1] \times [L_2, U_2]\}$ corresponding to both vehicles constrained to

their paths being in the conflict area of Figure 2.1.

2.3.3 Translation to a perfect state information control problem

In order to solve Problem II.12, it is necessary to compute the set $\overline{q}(\boldsymbol{\eta}(t))$. Computing this set from its definition is impractical as one would need to keep track of a growing history. Hence, it is customary to determine it recursively through a suitable update law [66]. A wealth of research on observer design and state estimation for hybrid systems has been concerned with determining such an update law and in particular with its properties for special classes of hybrid systems [34, 30, 32, 14, 91, 13, 41, 17]. Specifically, key properties, when considering discrete state estimation, are correctness, tightness, and convergence [34, 30]. Correctness requires that the estimated set of modes contains the true mode at any time; tightness requires that the estimated set of modes contains only modes compatible with the system history and dynamics; convergence requires that the estimated set converges to a singleton. In this thesis, we only require that the discrete state estimator has the correctness property. We are not concerned with tightness nor with convergence guarantees, which usually require observability assumptions. Hence, a discrete state estimator always exists as, for example, $\hat{q}(t) \equiv Q$ for all t is also an estimator. This allows us to *separate* the design of the estimator from that of the control map.

More formally, let $\hat{H} = (\hat{Q}, X, U, D, Y, \hat{R}, \hat{f})$ be a hybrid system with uncontrolled mode transitions with state $(\hat{q}, \hat{x}) \in \hat{Q} \times X$, in which $\hat{Q} \subseteq 2^Q$, and disturbance events $y \in Y$. Let $\{\hat{\tau}'_i\}_{i\in\hat{I}} \subset \mathbb{R}$ for $\hat{I} = \{0, 1, 2, 3, ...\}$ with $\hat{\tau}'_i = \hat{\tau}_{i+1} \leq \hat{\tau}'_{i+1}$ be the sequence of times at which $y(\hat{\tau}'_i) \in Y/\epsilon$ and $y(t) = \epsilon$ for $t \notin \{\hat{\tau}'_i\}_{i\in\hat{I}}$. Denote $\hat{\mathcal{T}} := \bigcup_{i\in\hat{I}}[\hat{\tau}_i, \hat{\tau}'_i)]$ in which $\hat{\tau}_i \leq \hat{\tau}'_i = \hat{\tau}'_{i+1}$, and $\hat{\tau}_0 = \tau_0 = 0$. For all $\hat{q} \in \hat{Q}$, we define $\hat{R}(\hat{q}, \epsilon) := \hat{q}$. Let the initial state be $(\overline{q}_o, x_o) \in \hat{Q} \times X$. The trajectories of $\hat{H}$ are defined as in Definition II.2, in which the continuous state obeys the differential inclusion

$$\dot{\hat{x}}(t) \in \hat{f}(\hat{x}(t), \hat{q}(t), v(t), d(t)), \ d(t) \in D, \ \text{for } t \in [\hat{\tau}_i, \hat{\tau}'_i], \ \hat{\tau}_i < \hat{\tau}'_i,$$

in which $\hat{x}(\hat{\tau}_{i+1}) = x(\hat{\tau}'_i)$ and $\hat{x}(\tau_0) = x_o$. As performed for system H, we can define the flow of system $\hat{H}$. Specifically, the discrete flow of $\hat{H}$ is denoted $\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}) := \hat{q}(\sup_{\hat{\tau}_i \leq t} \hat{\tau}_i)$ and *any* continuous flow of $\hat{H}$ is denoted by $\phi_{\hat{x}}(t, (\overline{q}_o, x_o), \mathbf{v}, \mathbf{d}, \mathbf{y}) := \hat{x}(t)$ for all $t \geq 0$. When $\mathbf{y} = \epsilon$, it is useful to extend the definition of this flow to when $\overline{q}$ is any element in 2^Q, that is, $\phi_{\hat{x}}(t, (\overline{q}, x_o), \mathbf{v}, \mathbf{d}, \epsilon) := \hat{x}(t)$ with $\hat{x}(t)$ such that $\dot{\hat{x}}(t) \in \hat{f}(\hat{x}(t), \overline{q}, v(t), d(t))$ for all $t > 0$ and $x(0) = x_o$. Note that, however, this may not be realizable in $\hat{H}$ if $\overline{q} \notin \hat{Q}$. Also, for all $\overline{q}_o \in \hat{Q}$, we denote $\hat{\text{Reach}}(\overline{q}_o) \subseteq \hat{Q}$ the set of reachable modes from $\overline{q}_o$ and it is defined as $\hat{\text{Reach}}(\overline{q}_o) := \bigcup_{t \geq 0} \bigcup_{\mathbf{y}} \phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y})$. Then, we have the following definition of an estimator for H.

Definition II.13. The hybrid system with uncontrolled mode transitions $\hat{H}$ with initial state $(\overline{q}_o, x_o) \in \hat{Q} \times X$ is called an *estimator* for H provided

(i) for all input/output signals $(\mathbf{u}, \mathbf{x})$ of H and all initial mode informations $\overline{q}_o \in \hat{Q}$, there is an event signal $\mathbf{y}$ in $\hat{H}$ such that $\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}) \ni q(t)$ for all $t \in \mathcal{T}$;

(ii) for all $y \in Y$ and $\hat{q} \in \hat{Q}$, we have that $\hat{R}(\hat{q}, y) \subseteq \text{Reach}(\hat{q})$;

(iii) for all $(\hat{x}, \hat{q}, v, d) \in X \times \hat{Q} \times U \times D$, we have that $\hat{f}(\hat{x}, \hat{q}, v, d) = \bigcup_{q \in \hat{q}} f(\hat{x}, q, v, d)$.

The dynamics of $\hat{x}$ model for a suitable event signal $\mathbf{y}$ the set of all possible dynamics of x in system H compatible with the current mode estimate $\hat{q}(t)$. Note that in H we can have that $\tau'_0 = \tau_0$ with the mode $q(\tau'_0)$ taking any value in $\text{Reach}(\overline{q}_o)$. Since by (i) of the above definition $\overline{q}_o$ can be any element of $\hat{Q}$, we must have that for all $\hat{q} \in \hat{Q}$ there is $y \in Y$ such that $\hat{R}(\hat{q}, y) = \text{Reach}(\hat{q})$ to ensure that $\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}) \ni q(t)$. According to the above definition, an estimator always exists as one can choose, for example, $\hat{Q} = \{\overline{q}_o, \text{Reach}(\overline{q}_o)\}$, $Y = \{\epsilon, y_0\}$, $\hat{R}$ such that $\hat{R}(\overline{q}_o, y_0) = \text{Reach}(\overline{q}_o)$, $\hat{\tau}'_0 = \hat{\tau}_0$, and $y(\hat{\tau}'_0) = y_0$. This implies that $\hat{q}(\hat{\tau}_0) = \overline{q}_o$, that $\hat{q}(\hat{\tau}'_0) = \text{Reach}(\overline{q}_o)$, and that $\hat{q}(\hat{\tau}'_0) \equiv \text{Reach}(\overline{q}_o)$ for all $t \geq \hat{\tau}'_0$. Hence, $\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}) \equiv \text{Reach}(\overline{q}_o)$ always contains $q(t)$ for all $t \in \mathcal{T}$ as $q(t) \in \text{Reach}(\overline{q}_o)$ for all $t \in \mathcal{T}$. An example of how to construct a less trivial estimator is provided in the following paragraph.

Example II.14. Consider the HMHS $H = (Q, X, U, D, \Sigma, R, f)$, in which $X = \mathbb{R}^2$, $Q = \{a, b\}$, $U = \emptyset$, $D = [-\bar{d}, \bar{d}] \subset \mathbb{R}$ for $\bar{d} > 0$, $\Sigma = \{\epsilon\}$, and $f(x, d) = (x_2, \beta_q + d)$, in which β_q is a parameter whose value depends on the mode q. This system can model, for example, the non-communicating vehicle of the application example of Section 2.3.2, in which "a" is acceleration mode and "b" is braking mode. Let the initial information be $(\bar{q}_o, x_o)$, in which $\bar{q}_o = Q$. We let $\hat{Q} = \{\hat{q}_1, \hat{q}_2, \hat{q}_3\}$, in which $\hat{q}_1 = Q$, $\hat{q}_2 = \{a\}$, and $\hat{q}_3 = \{b\}$. The signal $\mathbf{y}$ determines how to transit among these modes on the basis of $x(t)$ so to guarantee that $\phi_{\hat{q}}(t, \bar{q}_o, \mathbf{y}) \ni q(t)$. Since R does not allow transitions between a and b, the only transitions allowed by $\hat{R}$ are from $\hat{q}_1$ to $\hat{q}_2$ and from $\hat{q}_1$ to $\hat{q}_3$ by property (ii) of Definition II.13. Then, let $Y = \{y_a, y_b, \epsilon\}$, in which y_a is such that $\hat{R}(\hat{q}_1, y_a) = \hat{q}_2$ and y_b is such that $\hat{R}(\hat{q}_1, y_b) = \hat{q}_3$. Let $\hat{\beta}(t) = \frac{1}{T}\int_{t-T}^{t} \dot{x}_2(\tau)d\tau,\ t > T$ and define $y(t)$ as $y(t) = y_a$ if $|\hat{\beta}(t) - \beta_b| > \bar{d}$, $y(t) = y_b$ if $|\hat{\beta}(t) - \beta_a| > \bar{d}$, and $y(t) = \epsilon$ otherwise.

Note that while the discrete state of system H is unknown, the discrete state of system $\hat{H}$ is known as its initial state is known and both $\hat{q}(t)$ and $\hat{x}(t)$ are measured. Hence, we define the closed loop system under a *static* feedback map as follows.

Definition II.15. Consider a feedback map $\hat{\pi} : \hat{Q} \times X \to U$. The *closed loop system* $\hat{H}^{\hat{\pi}}$ is defined as system $\hat{H}$, in which $v(t) = \hat{\pi}(\phi_{\hat{q}}(t, \bar{q}_o, \mathbf{y}), \hat{x}(t))$ for all $t \geq 0$. The flow of $\hat{H}^{\hat{\pi}}$ is denoted by $\hat{\phi}^{\hat{\pi}}(t, (\bar{q}_o, x_o), \mathbf{d}, \mathbf{y})$ and the continuous flow by $\phi^{\hat{\pi}}_{\hat{x}}(t, (\bar{q}_o, x_o), \mathbf{d}, \mathbf{y})$.

Definition II.16. The capture set for system $\hat{H}$ is denoted $\hat{C}$ and is given by $\hat{C} := \{(\bar{q}_o, x_o) \in \hat{Q} \times X \mid \forall\, \hat{\pi},\ \exists\, \mathbf{d},\ \mathbf{y},\ t \geq 0\ s.t.\ \text{some}\ \phi^{\hat{\pi}}_{\hat{x}}(t, (\bar{q}_o, x_o), \mathbf{d}, \mathbf{y}) \in Bad\}$.

Proposition II.17. *Let $\bar{q} \in \hat{Q}$ and define the mode-dependent capture set $\hat{C}_{\bar{q}} := \{x_o \in X \mid \forall\, \hat{\pi},\ \exists\, \mathbf{d},\ \mathbf{y},\ t \geq 0\ s.t.\ some\ \phi^{\hat{\pi}}_{\hat{x}}(t, (\bar{q}, x_o), \mathbf{d}, \mathbf{y}) \in Bad\}$. Then, we have that $\hat{C} = \bigcup_{\bar{q} \in \hat{Q}} \left(\bar{q} \times \hat{C}_{\bar{q}}\right)$.*

Problem II.18. (Safety Control with Perfect State Information) Let $\hat{H}$ be an estimator for H. Determine the capture set $\hat{C}$ and the set of feedback maps $\hat{\pi}$ such that if $(\bar{q}_o, x_o) \notin \hat{C}$, then all flows $(\phi_{\hat{q}}(t, \bar{q}_o, \mathbf{y}), \phi^{\hat{\pi}}_{\hat{x}}(t, (\bar{q}, x_o), \mathbf{d}, \mathbf{y})) \notin \hat{C}$ for all $t \geq 0$, $\mathbf{d}$, and $\mathbf{y}$.

Definition II.19. Consider the feedback map $\hat{\pi} : \hat{Q} \times X \to U$ and an estimator $\hat{H}$. The *estimator-based closed loop system* $H_e^{\hat{\pi}}$ is defined as system H, in which $u(t) = \hat{\pi}(\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}), x(t))$ for all $t \geq 0$.

Definition II.20. We say that system $\hat{H}^{\hat{\pi}}$ with initial state $(\overline{q}_o, x_o)$ is safe provided $(\overline{q}_o, x_o) \notin \hat{C}$ implies that $\hat{x}(t) \notin \mathit{Bad}$ for all t, $\mathbf{d}$, and $\mathbf{y}$. Similarly, we say that system $H_e^{\hat{\pi}}$ with initial information $(\overline{q}_o, x_o)$ is safe provided $(\overline{q}_o, x_o) \notin \hat{C}$ implies that $x(t) \notin \mathit{Bad}$ for all t, $\mathbf{d}$, and $\boldsymbol{\sigma}$.

Definition II.21. (Weak equivalence) We say that Problem II.12 and Problem II.18 are *weakly equivalent* provided that

(i) if $\hat{H}^{\hat{\pi}}$ with initial state $(\overline{q}_o, x_o)$ is safe then also $H_e^{\hat{\pi}}$ with initial information $(\overline{q}_o, x_o)$ is safe;

(ii) for all $\overline{q} \in \hat{Q}$, we have that $C_{\overline{q}} \subseteq \hat{C}_{\overline{q}}$.

Definition II.22. (Equivalence) We say that Problem II.12 and Problem II.18 are *equivalent* provided that

(i) they are weakly equivalent;

(ii) for all $\overline{q} \in \hat{Q}$, we have that $C_{\overline{q}} = \hat{C}_{\overline{q}}$.

Weak equivalence guarantees that any feedback map $\hat{\pi}$ that keeps $\hat{H}^{\hat{\pi}}$ safe keeps also system $H_e^{\hat{\pi}}$ safe. Equivalence guarantees that system $\hat{H}$ has the same mode-dependent capture sets as system H.

Proposition II.23. *Problem II.12 and Problem II.18 are weakly equivalent.*

Proof. (i) If $\hat{H}^{\hat{\pi}}$ is safe with initial state $(\overline{q}_o, x_o)$, we have that $(\overline{q}_o, x_o) \notin \hat{C}$ implies that $\hat{x}(t) \notin \mathit{Bad}$ for all t, $\mathbf{d}$, and $\mathbf{y}$. In particular, this is true for $\mathbf{y}$ such that $\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}) \ni q(t)$ for all t and hence for $\hat{x}^*(t)$ such that $\dot{\hat{x}}^*(t) = f(\hat{x}^*(t), q(t), \hat{\pi}(\phi_{\hat{q}}(t, \overline{q}_o, \mathbf{y}), \hat{x}^*(t)), d(t)),\ d(t) \in D$, and hence for $x(t)$ trajectory of $H_e^{\hat{\pi}}$.

(ii) We show that $C_{\bar{q}} \subseteq \hat{C}_{\bar{q}}$ for all $\bar{q} \in \hat{Q}$. Specifically, we show that if $x_o \notin \hat{C}_{\bar{q}}$ then $x_o \notin C_{\bar{q}}$. If $x_o \notin \hat{C}_{\bar{q}}$, there is a feedback map $\hat{\pi}$ such that for all $\mathbf{d}$, $\mathbf{y}$, $t \geq 0$ all flows $\phi^{\hat{\pi}}_{\hat{x}}(t, (\bar{q}, x_o), \mathbf{d}, \mathbf{y}) \notin \mathit{Bad}$. In particular, this is true for $\mathbf{y}'$ such that $\hat{\tau}_0 = \hat{\tau}'_0$, $\hat{R}(\bar{q}, y'(\hat{\tau}'_0)) = \text{Reach}(\bar{q})$, and $y'(t) = \epsilon$ for all $t > \hat{\tau}'_0$ (note that a y for which $\hat{R}(\bar{q}, y) = \text{Reach}(\bar{q})$ must always exist in Y by the definition of an estimator). This implies that $\phi_{\hat{q}}(t, \bar{q}, \mathbf{y}') = \phi_{\hat{q}}(0, \bar{q}, \mathbf{y}') = \text{Reach}(\bar{q})$ for all t. In such a case, $\pi'(\hat{x}) := \hat{\pi}(\text{Reach}(\bar{q}), \hat{x})$ is a map from the continuous state only as the first argument is always constant. Hence, the flow $\hat{x}(t) = \phi^{\pi'}_{\hat{x}}(t, (\bar{q}, x_o), \mathbf{d}, \mathbf{y}')$ satisfies $\dot{\hat{x}}(t) \in f(\hat{x}(t), \text{Reach}(\bar{q}), \pi'(\hat{x}(t)), d(t))$ for all t. In turn, any $\hat{x}(t)$ that satisfies this also satisfies $\dot{\hat{x}}(t) = f(x(t), \phi_q(t, q_o, \boldsymbol{\sigma}), \pi'(x(t)), d(t))$ for all $q_o \in \bar{q}$ and all $\boldsymbol{\sigma}$. As a consequence, π' is such that $\phi^{\pi'}_x(t, (q_o, x_o), \mathbf{d}, \boldsymbol{\sigma}) \notin \mathit{Bad}$ for all $t \geq 0$, all $\mathbf{d}$, all $\boldsymbol{\sigma}$, and all $q_o \in \bar{q}$. This, in turn, implies that $x_o \notin C_{\bar{q}}$. □

We first solve Problem II.18 and then address the question of when this problem is equivalent to Problem 1.

2.4 Solution to Problem 2

Since $\hat{H}$ is a hybrid system with uncontrolled mode transitions, it has more structure than the general class of hybrid automata. We exploit this structure to provide a specialized iterative algorithm for the computation of the capture set and of the feedback maps $\hat{\pi}$.

2.4.1 Computation of the capture set $\hat{C}$

In order to compute the set $\hat{C}$, we introduce the notion of uncontrollable predecessor operator.

Definition II.24. For a set $S \subset X$ and $\bar{q} \in \hat{Q}$ the *uncontrollable predecessor operator* for $\hat{H}$ is defined as $\text{Pre}(\bar{q}, S) := \{x_o \in X \mid \forall\, \hat{\pi}\, \exists\, \mathbf{d}, t \geq 0, \text{s.t. some } \phi^{\hat{\pi}}_{\hat{x}}(t, (x_o, \bar{q}), \mathbf{d}, \epsilon) \in S\}$.

This set represents the set of all states that are mapped to S when the mode estimate is constant and equal to $\bar{q}$. Note that the feedback map $\hat{\pi}$ is thus a simple feedback map from

x as $\hat{q}$ is constant during the entire evolution. The following properties of the Pre operator follow from the fact that it is an order preserving map in both of its arguments.

Proposition II.25. *The operator Pre* : $\hat{Q} \times 2^X \to 2^X$ *has the following properties for all* $\hat{q} \in \hat{Q}$ *and* $S \in 2^X$*:*

(i) $S \subseteq Pre(\hat{q}, S)$*;*

(ii) $Pre(\hat{q}, Pre(\hat{q}, S)) = Pre(\hat{q}, S)$*;*

(iii) $Pre(\hat{q}, S_1) \subseteq Pre(\hat{q}, S_2)$*, for all* $S_1 \subseteq S_2$*;*

(iv) $Pre(\hat{q}_1, S) \subseteq Pre(\hat{q}_2, S)$*, for all* $\hat{q}_1 \subseteq \hat{q}_2$*;*

(v) $Pre(\hat{q}_1, Pre(\hat{q}_2, S)) = Pre(\hat{q}_1, S)$*, for all* $\hat{q}_2 \subseteq \hat{q}_1$*;*

(vi) $Pre(\hat{q}_0, S_0 \cup Pre(\hat{q}_1, S_1) \cup \ldots \cup Pre(\hat{q}_n, S_n)) = Pre(\hat{q}_0, S_0 \cup S_1 \cup \ldots \cup S_n)$ *for* $\hat{q}_i \subseteq \hat{q}_0$ *for all* i*.*

Proof. Property (i) follows directly from the definition of Pre, in which $t = 0$. To show property (ii), let $x_o \in \text{Pre}(\hat{q}, \text{Pre}(\hat{q}, S))$. By the definition of Pre, we have that for all $\hat{\pi}$ there is $\mathbf{d}_1$ and a time t_1 such that some $\phi^{\hat{\pi}}_{\hat{x}}(t_1, (x_o, \hat{q}), \mathbf{d}_1, \epsilon) \in \text{Pre}(\hat{q}, S)$. Define $x'_o := \phi^{\hat{\pi}}_{\hat{x}}(t_1, (x_o, \hat{q}), \mathbf{d}_1, \epsilon)$. Since $x'_o \in \text{Pre}(\hat{q}, S)$, we have by the definition of Pre that for all $\hat{\pi}$ there is $\mathbf{d}_2$ and $t_2 > 0$ such that some $\phi^{\hat{\pi}}_{\hat{x}}(t_2, (x'_o, \hat{q}), \mathbf{d}_2, \epsilon) \in S$. Let $t = t_1 + t_2$ and define $\mathbf{d}$ such that $d(\tau) = d_1(\tau)$ for $\tau < t_1$ and $d(\tau) = d_2(\tau - t_1)$ for $\tau \geq t_1$. Then, we have that $\phi^{\hat{\pi}}_{\hat{x}}(t_2, (x'_o, \hat{q}), \mathbf{d}_2, \epsilon) = \phi^{\hat{\pi}}_{\hat{x}}(t, (x_o, \hat{q}), \mathbf{d}, \epsilon)$. Since for all $\hat{\pi}$ there is $\mathbf{d}$ such that $\phi^{\hat{\pi}}_{\hat{x}}(t, (x_o, \hat{q}), \mathbf{d}, \epsilon) \in S$, we also have that $x_o \in \text{Pre}(\hat{q}, S)$. Property (iii) is an immediate consequence of the definition of Pre. Property (iv) follows from the fact that if for all π a trajectory $\hat{x}(t)$ such that $\dot{\hat{x}}(t) \in \hat{f}(\hat{x}(t), \hat{q}_1, \hat{\pi}(\hat{q}_1, \hat{x}(t)), d(t))$ enters S, then also a trajectory such that $\dot{\hat{x}}(t) \in \hat{f}(\hat{x}(t), \hat{q}_2, \hat{\pi}(\hat{q}_2, \hat{x}(t)), d(t))$ with $\hat{q}_2 \supseteq \hat{q}_1$ enters S. Property (v) follows from the fact that (a) $\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_2, S)) \supseteq \text{Pre}(\hat{q}_1, S)$ by property (i) and (iii); and from the fact that (b) $\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_2, S)) \subseteq \text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_1, S))$ by properties (iv) and (iii); and

from the fact that (c) $\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_1, S)) = \text{Pre}(\hat{q}_1, S)$ by property (ii). Finally, we show property (vi). By property (i), we have that $S_1 \cup \ldots \cup S_n \subseteq \text{Pre}(\hat{q}_1, S_1) \cup \ldots \cup \text{Pre}(\hat{q}_n, S_n)$. Thus, applying property (iii), we have that $\text{Pre}(\hat{q}_0, S_0 \cup S_1 \cup \ldots \cup S_n) \subseteq \text{Pre}(\hat{q}_0, S_0 \cup \text{Pre}(\hat{q}_1, S_1) \cup \ldots \cup \text{Pre}(\hat{q}_n, S_n))$. Also, applying property (iv) and property (iii), we have that $\text{Pre}(\hat{q}_0, S_0 \cup \text{Pre}(\hat{q}_0, S_1) \cup \ldots \cup \text{Pre}(\hat{q}_0, S_n)) \supseteq \text{Pre}(\hat{q}_0, S_0 \cup \text{Pre}(\hat{q}_1, S_1) \cup \ldots \cup \text{Pre}(\hat{q}_n, S_n))$. However, $\text{Pre}(\hat{q}_0, S_0 \cup \text{Pre}(\hat{q}_0, S_1) \cup \ldots \cup \text{Pre}(\hat{q}_0, S_n)) = \text{Pre}(\hat{q}_0, S_0 \cup S_1 \cup \ldots \cup S_n)$ by the definition of Pre (using the same strategy as used for proving property (ii)). Hence $\text{Pre}(\hat{q}_0, S_0 \cup \text{Pre}(\hat{q}_1, S_1) \cup \ldots \cup \text{Pre}(\hat{q}_n, S_n)) = \text{Pre}(\hat{q}_0, S_0 \cup S_1 \cup \ldots \cup S_n)$ for $\hat{q}_i \subseteq \hat{q}_0$ for all i. □

We use for all $\hat{q} \in \hat{Q}$ the notation $\hat{R}(\hat{q}, Y) := \{\hat{q}' \in \hat{R}(\hat{q}, y) \mid y \in Y\}$, in which we set $\hat{R}(\hat{q}, y) := \emptyset$ if $\hat{R}(\hat{q}, y)$ is not defined for some $y \in Y$.

Proposition II.26. *The sets $\hat{C}_{\hat{q}_i}$ for all $\hat{q}_i \in \hat{Q}$ satisfy $\hat{C}_{\hat{q}_i} = Pre\left(\hat{q}_i, \bigcup_{\{\hat{q}_j \in \hat{R}(\hat{q}_i, Y)\}} \hat{C}_{\hat{q}_j} \cup Bad\right)$.*

Proof. Define $D := Bad \cup_{\{\hat{q}_j \in \hat{R}(\hat{q}_i, Y)\}} \hat{C}_{\hat{q}_j}$ and $A := \text{Pre}(\hat{q}_i, D)$. Take $x_o \in A$. This implies, by the definition of Pre that for all $\hat{\pi}$ there exists t and disturbance signal $\mathbf{d}$ such that some $\phi_{\hat{x}}^{\hat{\pi}}(t, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \in \bigcup_{\{\hat{q}_j \in \hat{R}(\hat{q}_i, Y)\}} \hat{C}_{\hat{q}_j} \cup Bad$. Since $Bad \subset \hat{C}_{\hat{q}_j}$ for all $\hat{q}_j$, this implies that for all $\hat{\pi}$ there exists time t_1, a signal $\mathbf{d}$, and $\hat{q}_j \in \hat{R}(\hat{q}_i, Y)$ such that $\hat{x}(t_1) = \phi_{\hat{x}}^{\hat{\pi}}(t_1, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \in \hat{C}_{\hat{q}_j}$. Let $\mathbf{y}$ be such that $\hat{q}(t_1) = \phi_{\hat{q}}(t_1, \hat{q}_i, \mathbf{y}) = \hat{q}_j$. Then, $(\hat{q}(t_1), \hat{x}(t_1)) \in \hat{C}$. Therefore for all $\hat{\pi}$, there exists t and signals $\mathbf{d}, \mathbf{y}$ such that some $\phi_{\hat{x}}^{\hat{\pi}}(t, (\hat{q}_i, x_o), \mathbf{d}, \mathbf{y}) \in Bad$. This in turn implies, by the definition of $\hat{C}_{\hat{q}}$, that $x_o \in \hat{C}_{\hat{q}_i}$.

Now consider $x_o \in \hat{C}_{\hat{q}_i}$. By definition of $\hat{C}$, we also have that $(\hat{q}_i, x_o) \in \hat{C}$. If $(\hat{q}_i, x_o) \in \hat{C}$, then for all $\hat{\pi}$ there are $\mathbf{d}, \mathbf{y}$ such that some $(\hat{q}(t), \phi_{\hat{x}}^{\hat{\pi}}(t, (\hat{q}_i, x_o), \mathbf{d}, \mathbf{y})) \in \hat{C}$, for all t, otherwise the definition of $\hat{C}$ would be contradicted. If $\mathbf{y}$ makes $\hat{q}_i$ switch to some $\hat{q}_j \in \hat{R}(\hat{q}_i, Y)$ at time t_1, it must be that $\phi_{\hat{x}}^{\hat{\pi}}(t_1, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \in \hat{C}_{\hat{q}_j}$. If instead $\mathbf{y}$ does not make $\hat{q}_i$ switch, then it must be that $\phi_{\hat{x}}^{\hat{\pi}}(t_2, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \in Bad$ for some t_2. Combining the last two statements, we obtain that for all $\hat{\pi}$, there exists t, $\mathbf{d}$, and $\mathbf{y}$ such that either $\phi_{\hat{x}}^{\hat{\pi}}(t, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \in \hat{C}_{\hat{q}_j}$ or $\phi_{\hat{x}}^{\hat{\pi}}(t, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \in Bad$, which implies $x_o \in A$. □

Definition II.27. A set $\hat{W} \subseteq \hat{Q} \times X$ is said a *controlled invariant set* for $\hat{H}$ if there is a feedback map $\hat{\pi}$ such that for all $(\overline{q}_o, x_o) \in \hat{W}$, we have that all flows $\hat{\phi}^{\hat{\pi}}(t, (\overline{q}_o, x_o), \mathbf{d}, \mathbf{y}) \in \hat{W}$ for all t, $\mathbf{d}$, and $\mathbf{y}$. A set $\hat{W} \subseteq \hat{Q} \times X$ is *the maximal controlled invariant set* for $\hat{H}$ provided it is a controlled invariant set for $\hat{H}$ and any other controlled invariant set for $\hat{H}$ is a subset of $\hat{W}$.

Proposition II.28. *The set $\hat{W} := (\hat{Q} \times X)/\hat{C}$ is the maximal controlled invariant set for $\hat{H}$ contained in $(\hat{Q} \times X)/(\hat{Q} \times Bad)$.*

Proof. Let $(\hat{q}_i, x_i) \in \hat{W}$. Then, by the definition of $\hat{C}$ we have that there is a feedback map $\hat{\pi}_i$ such that all $\hat{\phi}^{\hat{\pi}_i}(t, (\hat{q}_i, x_i), \mathbf{d}, \mathbf{y}) \in \hat{W}$ for all $\mathbf{d}, \mathbf{y}$ and $t \geq 0$. Define the set $\overline{W}_i := \bigcup_{\mathbf{d},\mathbf{y},t\geq 0} \hat{\phi}^{\hat{\pi}_i}(t, (\hat{q}_i, x_i), \mathbf{d}, \mathbf{y}) \subseteq \hat{W}$, which is controlled invariant with feedback map $\hat{\pi}_i$. Since the class of controlled invariant sets contained in $\hat{W}$ is closed under union (see the proof of Proposition 3 of [68]), there is a feedback map $\hat{\pi}$ that makes the union $\bigcup_{\{i \mid (\hat{q}_i, x_i) \in \hat{W}\}} \overline{W}_i \subseteq \hat{W}$ controlled invariant. Therefore $\hat{W}$ is also controlled invariant. It is the maximal controlled invariant set contained in $(\hat{Q} \times X)/(\hat{Q} \times Bad)$ because if $(\hat{q}, x) \notin \hat{W}$ then $(\hat{q}, x) \in \hat{C}$, which implies that for all maps $\hat{\pi}$ some flow $\hat{\phi}^{\hat{\pi}}(t, (\hat{q}, x), \mathbf{d}, \mathbf{y})$ enters $\hat{Q} \times Bad$ for some $\mathbf{d}, \mathbf{y}$, and $t \geq 0$. □

Let $\hat{Q} = \{\hat{q}_1, ..., \hat{q}_M\}$ with $\hat{q}_i \in 2^Q$ for $i \in \{1, \ldots, M\}$, $S_i \in 2^X$ for $i \in \{1, \ldots, M\}$, and define $S := (S_1, \ldots, S_M) \subseteq (2^X)^M$. We define the map $G : (2^X)^M \to (2^X)^M$ as

$$G(S) := \begin{bmatrix} \mathrm{Pre}\left(\hat{q}_1, \bigcup_{\{j \mid \hat{q}_j \in \hat{R}(\hat{q}_1, Y)\}} S_j \cup Bad\right) \\ \vdots \\ \mathrm{Pre}\left(\hat{q}_M, \bigcup_{\{j \mid \hat{q}_j \in \hat{R}(\hat{q}_M, Y)\}} S_j \cup Bad\right) \end{bmatrix}.$$

Proposition II.29. *Let $S := (S_1, ..., S_M)$ be a tuple of sets $S_i \subseteq X$ such that $S = G(S)$. Then, $(\hat{Q} \times X)/\bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$ is a controlled invariant set for $\hat{H}$.*

Proof. Let $(\hat{q}, x_o) \notin \bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$ for $\hat{q} = \hat{q}_i \in \hat{Q}$. Then $x_o \notin S_i$, where $S_i = \mathrm{Pre}(\hat{q}_i, \bigcup_{\{j \mid \hat{q}_j \in \hat{R}(\hat{q}_i, Y)\}} S_j \cup Bad)$. By the definition of Pre, this implies that there is a feedback map $\hat{\pi}$ such that $\phi^{\hat{\pi}}_{\hat{x}}(t, (\hat{q}_i, x_o), \mathbf{d}, \epsilon) \notin S_i$. Let t^* be such that $\hat{q}(t^*)$ switches to $\hat{q}_j \in \hat{R}(\hat{q}_i, Y)$.

At time t^*, we also have that any $\hat{x}(t^*) := \phi^{\hat{\pi}}_{\hat{x}}(t^*, (\hat{q}_i, x_o), \mathbf{d}, \epsilon)$ is not in S_i and thus $\hat{x}(t^*) \notin S_j$ which implies $(\hat{q}(t^*), \hat{x}(t^*)) \notin \bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$. Proceeding iteratively on the mode switch, we obtain that the flows of $\hat{H}$ starting from any $(\hat{q}, x_o) \notin \bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$ stay outside $\bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$ for a proper control map. Thus, the set $(\hat{Q} \times X)/\bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$ is a controlled invariant set. □

Let $Z := (2^X)^M$ represent the set of all M-tuples of subsets of X and define the partial order $(Z, \subseteq)$, where $\subseteq$ is defined component-wise. One can verify that $G : Z \to Z$ is an order preserving map (it follows from property (iii) of the Pre operator from Proposition II.25).

Algorithm II.30. $S^0 := (S^0_1, S^0_2, \ldots, S^0_M) := (\emptyset, \ldots, \emptyset)$,

$S^1 = G(S^0)$

while $S^{k-1} \neq S^k$

$S^{k+1} = G(S^k)$

end.

If Algorithm II.30 terminates, that is, if there is a K^* such that $S^{K^*} = (S^{K^*}_1, ..., S^{K^*}_M) = (S^{K^*+1}_1, ..., S^{K^*+1}_M) = S^{K^*+1}$, we denote the fixed point by S^*.

Theorem II.31. *If Algorithm 1 terminates, the fixed point S^* is such that $S^* = (\hat{C}_{\hat{q}_1}, ..., \hat{C}_{\hat{q}_M})$.*

Proof. If Algorithm 1 terminates, then there is $N^* > 0$ such that $G(\perp)^{N^*} = G(\perp)^{N^*+1} = S^*$, in which $\perp = \emptyset$. Thus, S^* is a fixed point of G. To show that it is the least fixed point, consider any other fixed point of G, called β. Since $\perp \leq \beta$ and G is an order preserving map, we have that $G(\perp) \leq G(\beta) = \beta$, $G^2(\perp) \leq G(\beta) = \beta$,..., $G^{N^*}(\perp) \leq \beta$. Since $G^{N^*}(\perp) = S^*$, we have that $S^* \leq \beta$. Thus S^* is the least fixed point of G.

Proposition II.26 indicates that the set $\hat{C} = \bigcup_{\hat{q}_i\in\hat{Q}}(\hat{q}_i \times \hat{C}_{\hat{q}_i})$ is such that the tuple of sets $(\hat{C}_{\hat{q}_1}, ..., \hat{C}_{\hat{q}_M})$ is a fixed point of G. Assume that such a tuple of sets is not the least fixed point of G. This implies that there are sets $S_i \subseteq \hat{C}_{\hat{q}_i}$ such that the tuple $(S_1, ..., S_M)$ is also a fixed point of G. Consider the sets $\hat{W} = (\hat{Q} \times X)/\bigcup_{\hat{q}_i\in\hat{Q}}(\hat{q}_i \times \hat{C}_{\hat{q}_i})$ and the new set $\hat{W}'$

defined as $\hat{W}' := (\hat{Q} \times X)/\bigcup_{i\in\{1,...,M\}}(\hat{q}_i \times S_i)$. By Proposition II.29, these two sets are both controlled invariant and are both contained in $(\hat{Q} \times X)/(\hat{Q} \times Bad)$. Since $\hat{W} \subset \hat{W}'$, we have that $\hat{W}$ is not the maximal controlled invariant set contained in the complement of $\hat{Q} \times Bad$. This contradicts Proposition II.28. Therefore, the tuple $(\hat{C}_{\hat{q}_1}, ..., \hat{C}_{\hat{q}_M})$ must be the least fixed point of G. Since the least fixed point of G equals S^* by the first part of the proof, it follows that $(\hat{C}_{\hat{q}_1}, ..., \hat{C}_{\hat{q}_M}) = S^*$. □

This result is based on the assumption that Algorithm 1 terminates and hence it is sufficient that the map G is an order preserving map. A stronger property for G, such as omega-continuity [59], is required for the result of Theorem II.31 to hold if termination of Algorithm 1 is not assumed. In Section 2.5, we address termination.

2.4.2 The control map

To determine the set of feedback maps that keep the complement of $\hat{C}$ invariant, we employ notions from viability theory.

Definition II.32. A set valued map $F : X \to 2^X$ is said *piecewise Lipschitz continuous* on X if it is Lipschitz continuous on a finite number of sets $X_i \subset X$ for $i = 1, ..., N$ that cover X, that is, $\bigcup_{i=1}^{N} X_i = X$, and $X_i \cap X_j = \emptyset$ for $i \neq j$.

The next result extends conditions for set invariance as found in [10] to the case of piecewise Lipschitz continuous set valued maps. This extension is required in our case because the vector field f is allowed to be piecewise continuous.

Proposition II.33. *Let $F : X \to 2^X$ be a set-valued Marchaud map. Assume that F is piecewise Lipschitz continuous on X. A closed set $S \subseteq X$ is invariant under F if and only if $F(x) \subseteq T_S(x)$ for all $x \in S$.*

Proof. We construct from F an impulse differential inclusion whose x trajectories are the same as the ones of the system $\dot{x} \in F(x)$ and then apply Theorem 3 from [11] to the resulting

impulse differential inclusion to conclude invariance of S. An impulse differential inclusion is a tuple $\bar{H} = (\bar{X}, \bar{F}, \bar{R}, \bar{J})$, in which $\bar{X}$ is a finite dimensional space, $\bar{F} : \bar{X} \to 2^{\bar{X}}$ is a set valued map regarded as a differential inclusion $\dot{\bar{x}} \in \bar{F}(\bar{x})$, $\bar{R} : \bar{X} \to 2^{\bar{X}}$ is a reset map, and $\bar{J} \subset \bar{X}$ is a forced discrete transition set. Since F is piecewise Lipschitz continuous on X, there are sets $X_i \subset X$ for $i = 1, ..., N$ that cover X on which F is Lipschitz. Define for each $i \in \{1, ..., N\}$ the maps $F_i : X \to 2^X$ such that $F_i(x) = F(x)$ for all $x \in X_i$ and for $x \notin X_i$ the map $F_i(x)$ is extended so that it is Lipschitz continuous on X. Then, $F_i : X \to 2^X$ is Marchaud and Lipschitz continuous. Let $z_i \in \{1, 0\}$ for $i \in \{1, ..., N\}$ and define $\bar{X} := X \times \{1, 0\}^N$. Let $z = (z_1, ..., z_N)$ and define the new map $\bar{F} : \bar{X} \to 2^{\bar{X}}$ as $\bar{F}(x, z) := \begin{pmatrix} z_1 F_1(x) + ... + z_N F_N(x) \\ 0_{N\times 1} \end{pmatrix}$, $\forall (x, z) \in \bar{X}$. Define a reset map $\bar{R} : \bar{X} \to \bar{X}$ by $\bar{R}(x, z) = (x, e_i)$, if $x \in X_i$. Define the set of forced transitions $\bar{J} \subset \bar{X}$ as $\bar{J} = \{(x, z) \in \bar{X} \mid x \in X_i \text{ and } z \neq e_i\}$. By construction, the x trajectories of $\bar{H}$ starting from initial conditions $z = e_i$ and $x \in X_i$ for all i coincide with the trajectories of $\dot{x} \in F(x)$ starting with the same $x \in X_i$.

Let $E := \{e_1, ..., e_N\} \subset \{1, 0\}^N$ and define the set $\bar{S} \subset \bar{X}$ as $\bar{S} = S \times E$. This is a closed set. Theorem 3 from [11] states that if $\bar{F}$ is Marchaud and Lipschitz and $\bar{J}$ is closed, then $\bar{S}$ is invariant under $\bar{H}$ if and only if (1) $\bar{R}(\bar{S}) \subseteq \bar{S}$ and (2) $\forall (x, z) \in \bar{S} \backslash \bar{J}$ we have $\bar{F}(x, z) \subseteq T_{\bar{S}}(x, z)$. Notice that $\bar{R}(\bar{S}) \subseteq \bar{S}$ by the way $\bar{R}$ is constructed. Let then $F(x) \subseteq T_S(x)$ for all $x \in S$. We show that this implies that also $\bar{F}(x, z) \subseteq T_{\bar{S}}(x, z)$ for all $(x, z) \in \bar{S} \backslash \bar{J}$. By the way $\bar{F}$, $\bar{S}$, and $\bar{J}$ have been defined, for all $(x, z) \in \bar{S} \backslash \bar{J}$ we have that $\bar{F}(x, z) = (F_i(x), 0_{N\times 1})$ with $x \in X_i$. Since also $x \in S$, we have $F_i(x) \subseteq T_S(x)$ because $x \in X_i$ and $F_i(x) = F(x)$ for $x \in X_i$. Since $z \in E$, we have that $T_E(z) = 0_{N\times 1}$. As a consequence, $\bar{F}(x, z) \subseteq T_S(x) \times T_E(z)$. Given that $T_{S\times E}(x, z) = T_S(x) \times T_E(z)$ [26], it follows that $\bar{F}(x, z) \subseteq T_{S\times E}(x, z)$ for all $(x, z) \in \bar{S} \backslash \bar{J}$. By Theorem 3 in [11], set $\bar{S}$ is invariant under $\bar{H}$, which implies that set S is invariant by F as the x trajectories of the first system starting in $(x_o, z_o) \in \bar{S}$ are the same as the x trajectories of the second system starting at $x_o \in S$.

Conversely, if $F(x) \not\subseteq T_S(x)$ for some $x \in S$, then for some i such that $x \in X_i$ we have

that $F_i(x) \not\subseteq T_S(x)$. This in turn implies that for $(x,z) \in \bar{S}\setminus\bar{J}$ (that is, for $z = e_i$) we have $\bar{F}(x,z) \not\subseteq T_{\bar{S}}(x,z)$. By Theorem 3 in [11] set $\bar{S}$ is thus not invariant under $\bar{H}$. This implies that there is a time t at which either $x(t) \notin S$ or $z(t) \notin E$. However, if $z(0) \in E$ we must have that $z(t) \in E$ for all t as z can change its value only through $\bar{R}$, which always maps z back in E. Therefore, there must be a time t such that $x(t) \notin S$ for system $\bar{H}$. Since the x trajectories of $\bar{H}$ starting at $(x_o, z_o) \in \bar{S}$ are the same as those of $\dot{x} \in F(x)$ starting at $x_o \in S$, it must be that $x(t) \notin S$ also for system $\dot{x} \in F(x)$, implying that S cannot be invariant for F. □

For simplifying notation, for each mode $\hat{q} \in \hat{Q}$ define the set valued map $\bar{f} : X \times \hat{Q} \times U \to 2^X$ as $\bar{f}(\hat{x},\hat{q},u) = \{\hat{f}(\hat{x},\hat{q},u,d), d \in D\}$ for all $(\hat{x},\hat{q},u) \in X \times \hat{Q} \times U$. Define $L_{\hat{q}} := X\setminus\hat{C}_{\hat{q}}$ for all $\hat{q} \in \hat{Q}$ and consider the set valued map defined as

$$\Pi(\hat{q},\hat{x}) := \{u \in U \mid \bar{f}(\hat{x},\hat{q},u) \subset T_{L_{\hat{q}}}(\hat{x})\}. \tag{2.3}$$

Theorem II.34. *Assume that $\hat{\pi} : \hat{Q} \times X \to U$ is such that for all $\hat{q} \in \hat{Q}$ the set-valued map $F(\hat{x},\hat{q}) := \bar{f}(\hat{x},\hat{q},\hat{\pi}(\hat{x},\hat{q}))$ is Marchaud and piecewise Lipschitz continuous on X. Then, the set $(\hat{Q} \times X)\setminus\hat{C}$ is invariant for $\hat{H}^{\hat{\pi}}$ if and only if $\hat{\pi}(\hat{q},\hat{x}) \in \Pi(\hat{q},\hat{x})$.*

Proof. ($\Leftarrow$) Assume that $\hat{\pi}(\hat{q},\hat{x}) \in \Pi(\hat{q},\hat{x})$ and that $(\hat{q}(\hat{\tau}_0),\hat{x}(\hat{\tau}_0)) \notin \hat{C}$, we show that all $(\hat{q}(t),\hat{x}(t)) \notin \hat{C}$ for all $t \geq \hat{\tau}_0$. This is shown by induction argument on the transition times $\hat{\tau}'_i$. (Base case) By assumption we have that $(\hat{q}(\hat{\tau}_0),\hat{x}(\hat{\tau}_0)) \notin \hat{C}$. (Induction step) Assume that $(\hat{q}(\hat{\tau}_i),\hat{x}(\hat{\tau}_i)) \notin \hat{C}$. We show that this implies $(\hat{q}(t),\hat{x}(t)) \notin \hat{C}$ for all $t \in [\hat{\tau}_i, \hat{\tau}_{i+1}]$, in which $\hat{\tau}_{i+1} = \hat{\tau}'_i$. This in turn is equivalent to showing that $\hat{x}(t) \notin \hat{C}_{\hat{q}(\hat{\tau}_i)}$ for all $t \in [\hat{\tau}_i, \hat{\tau}'_i]$ and $\hat{x}(\hat{\tau}_{i+1}) \notin \hat{C}_{\hat{q}(\hat{\tau}_{i+1})}$. Since $\hat{C}_{\hat{q}(\hat{\tau}_{i+1})} \subseteq \hat{C}_{\hat{q}(\hat{\tau}_i)}$ by the properties of the Pre operator and by Proposition II.26, then if $\hat{x}(\hat{\tau}'_i) \notin \hat{C}_{\hat{q}_{\hat{\tau}_{i+1}}}$ also $\hat{x}(\hat{\tau}'_i) \notin \hat{C}_{\hat{q}(\hat{\tau}_{i+1})}$. Therefore, it is enough to show that $\hat{x}(t) \notin \hat{C}_{\hat{q}(\hat{\tau}_i)}$ for all $t \in [\hat{\tau}_i, \hat{\tau}'_i]$. If $\hat{\tau}'_i = \hat{\tau}_i$, then since $\hat{x}(\hat{\tau}'_i) = \hat{x}(\hat{\tau}_i)$ we have that $\hat{x}(\hat{\tau}_i) \notin \hat{C}_{\hat{q}(\hat{\tau}_i)}$. If $\hat{\tau}_i < \hat{\tau}'_i$, for $t \in [\hat{\tau}_i, \hat{\tau}'_i)$, the trajectory $\hat{x}(t)$ satisfies $\dot{\hat{x}}(t) \in \bar{f}(\hat{x}(t),\hat{q}(\hat{\tau}_i),\hat{\pi}(\hat{q}(\hat{\tau}_i))) = F(x,\hat{q}(\hat{\tau}_i))$. Since $\hat{\pi}(\hat{q},\hat{x}) \in \Pi(\hat{q},\hat{x})$, it follows that $F(\hat{x},\hat{q}(\hat{\tau}_i)) \subseteq T_{L_{\hat{q}(\hat{\tau}_i)}}(\hat{x})$. Proposition

II.33 thus implies that $L_{\hat{q}(\hat{\tau}_i)}$ is invariant by F. Therefore, we have that $\hat{x}(t) \in L_{\hat{q}(\hat{\tau}_i)}$ for all $t \in [\hat{\tau}_i, \hat{\tau}'_i]$. Thus, $\hat{x}(t) \notin \hat{C}_{\hat{q}(\hat{\tau}_i)}$ for all $t \in [\hat{\tau}_i, \hat{\tau}'_i]$.

($\Rightarrow$) The fact that if $\hat{\pi}(\hat{q}, \hat{x}) \notin \Pi(\hat{q}, \hat{x})$ the set $(\hat{Q} \times X)/\hat{C}$ is not invariant for $\hat{H}^{\pi}$ follows from Proposition II.33. □

Given the current mode estimate $\hat{q}$, a control map as given in Theorem II.34 is one that makes all the possible vector fields point outside the current mode-dependent capture set $\hat{C}_{\hat{q}}$. Once the mode estimate switches to $\hat{q}'$, the current mode-dependent capture set also switches to the new mode-dependent capture set $\hat{C}_{\hat{q}'}$, which is (by Algorithm 1) contained in the previous one $\hat{C}_{\hat{q}}$. At this point, the feedback map switches to one that makes all the possible vector fields originating from $\hat{q}'$ point outside the new current mode-dependent capture set $\hat{C}_{\hat{q}'}$. Note that control map (2.3) guarantees safety for any choice of an estimator. However, a coarser estimator leads to larger mode dependent capture sets to be avoided at any time and, as a consequence, the control actions are more conservative.

2.5 Termination of Algorithm 1

There are two main difficulties in the implementation of Algorithm 1. The first one is the exact computation of the Pre operator, which is known to be a hard problem for general classes of nonlinear and hybrid dynamics and general results are still lacking. Hence, research has been focusing on special classes of systems for which such an operator can be exactly computed [85, 83, 84]. The second difficulty lies in the fact that even when one can exactly compute the Pre operator, Algorithm 1 is not always guaranteed to terminate in a finite number of steps. In this section, we address the termination of Algorithm 1, that is, the existence of a *finite N* such that $S^N = S^{N+1}$. We then discuss the problem of the exact computation of the Pre operator.

For the termination problem, we first provide sufficient conditions on $\hat{H}$ for which Algorithm 1 terminates. Then, we show that one can construct an abstraction of $\hat{H}$ for which

Algorithm 1 always terminates and such that the fixed point gives the mode-dependent capture sets of $\hat{H}$. In order to proceed, we introduce the notion of kernel sets for $\hat{H}$.

Definition II.35. (Kernel set) The *kernel set* corresponding to a mode $\hat{q}^* \in \hat{Q}$ is defined as $ker(\hat{q}^*) := \{\hat{q} \in \hat{Q} \mid \hat{q} \in \widehat{\text{Reach}}(\hat{q}^*) \text{ and } \hat{q}^* \in \widehat{\text{Reach}}(\hat{q})\}$.

The kernel set for a mode $\hat{q}^*$ is thus the set of all modes that can be reached from $\hat{q}^*$ and from which $\hat{q}^*$ can be reached. One can verify that for all pairs of modes $\hat{q}_i, \hat{q}_j \in \hat{Q}$, we have that $\hat{q}_i \in \widehat{\text{Reach}}(\hat{q}_j)$ and $\hat{q}_j \in \widehat{\text{Reach}}(\hat{q}_i)$ if and only if $ker(\hat{q}_i) = ker(\hat{q}_j)$. The next result shows that any two modes of $\hat{H}$ in the same kernel set have the same mode-dependent capture set and hence the same set of safe feedback maps.

Proposition II.36. *For every kernel set $ker \subseteq \hat{Q}$ and for any two modes $\hat{q}, \hat{q}' \in ker$, we have that $\hat{C}_{\hat{q}} = \hat{C}_{\hat{q}'}$ and hence that $\Pi(\hat{q}, x) = \Pi(\hat{q}', x)$.*

Proof. Since $\hat{q}, \hat{q}' \in ker$, we have that $\hat{q}' \in \widehat{\text{Reach}}(\hat{q})$ and that $\hat{q} \in \widehat{\text{Reach}}(\hat{q}')$. By Proposition II.26, the first inclusion implies that $\hat{C}_{\hat{q}'} \subseteq \hat{C}_{\hat{q}}$, while the second inclusion implies that $\hat{C}_{\hat{q}} \subseteq \hat{C}_{\hat{q}'}$. Hence, we must have that $\hat{C}_{\hat{q}} = \hat{C}_{\hat{q}'}$. By equation (2.3), this in turn implies also that $\Pi(\hat{q}, x) = \Pi(\hat{q}', x)$. □

Definition II.37. (Type of a kernel set) We say that a kernel set $ker_1 \subseteq \hat{Q}$ *transits* to a kernel set $ker_2 \subseteq \hat{Q}$ if there is $\hat{q}_1 \in ker_1$, $\hat{q}_2 \in ker_2$, and $y \in Y$ such that $\hat{q}_2 = \hat{R}(\hat{q}_1, y)$. A kernel set is $type(1)$ if it does not transit to any other kernel set. A kernel set is $type(n)$ if it transits to $type(n-1)$ kernel sets and only to $type(n-1), \ldots, type(1)$ kernel sets.

Let $\mathcal{K} := \{ker(\hat{q}_1), \ldots, ker(\hat{q}_M)\}$. Let there be p distinct elements in $\mathcal{K}$ denoted $ker_1, \ldots, ker_p$. Note that $ker_i \cap ker_j = \emptyset$, for $i \neq j$. If each of the kernel sets is just one element in $\hat{Q}$, it means that there are no discrete transitions possible in $\hat{R}$ that bring a discrete state $\hat{q}$ back to itself. That is, there is no loop in any of the trajectories of $\hat{q}$. In this case, one can verify that Algorithm 1 terminates in a finite number of steps. If instead there are kernel sets composed of more than one element, it means that there are discrete transitions that bring a

discrete state back to itself, that is, there are loops in the trajectories of $\hat{q}$. In this situation, Algorithm 1 may not terminate. The next result shows that even when there are loops in the trajectories of $\hat{q}$, Algorithm 1 still terminates if each kernel set contains a maximal element.

Proposition II.38. *Let $\hat{q}_i$ for $i \in \{1, ..., M\}$ be in a type(1) kernel set. Then, Algorithm 1 is such that there is a $K^* \geq 0$ for which $S_i^{K^*} = S_i^{K^*+1}$.*

Proof. Consider a mode $\hat{q}_l \in ker_a$, in which ker_a is $type(1)$ and let $\hat{q}_o \in ker_a$ be the maximal element of ker_a. Since ker_a is a $type(1)$ kernel set, we have that $\hat{q}_o$ transits only to modes in ker_a. Let there be n_a elements in kernel set ker_a and let $n > n_a$. Then there exists k and a sequence $\{\hat{q}_{l_i}\}_{i \leq n}$ with $\hat{q}_{l_i} \in \hat{R}(\hat{q}_{l_{i-1}}, Y)$ and $\hat{q}_{l_0} = \hat{q}_l$ such that $\hat{q}_o = \hat{q}_{l_k}$. Thus, from Algorithm II.30 and property (i) of Proposition II.25, we have that $S_l^n \supseteq \text{Pre}\left(\hat{q}_o, \bigcup_{\{\hat{q}_{l_{k+1}} \in \hat{R}(\hat{q}_{l_k}, Y)\}} \text{Pre}\left(\hat{q}_{l_{k+1}}, \bigcup_{\{\hat{q}_{l_{k+2}} \in \hat{R}(\hat{q}_{l_{k+1}}, Y)\}} \text{Pre}\left(\hat{q}_{l_{k+2}}, \ldots \bigcup_{\{\hat{q}_{l_n} \in \hat{R}(\hat{q}_{l_{n-1}}, Y)\}} \text{Pre}(\hat{q}_{l_n}, Bad)...\right)\right)\right)$. Since $\hat{q}_o$ is the maximal element of kernel set ker_a, we have, by repeatedly applying property (vi) of Proposition II.25, that $S_l^n \supseteq \text{Pre}(\hat{q}_o, Bad)$. In turn, from Algorithm II.30 and property (iv) of Proposition II.25, we have that

$$S_l^n \subseteq \text{Pre}\left(\hat{q}_o, \bigcup_{\{\hat{q}_{l_1} \in \hat{R}(\hat{q}_l, Y)\}} \text{Pre}\left(\hat{q}_{l_1}, \bigcup_{\{\hat{q}_{l_2} \in \hat{R}(\hat{q}_{l_1}, Y)\}} \cdots \bigcup_{\{\hat{q}_{l_n} \in \hat{R}(\hat{q}_{l_{n-1}}, Y)\}} \text{Pre}(\hat{q}_{l_n}, Bad)...\right)\right).$$

By repeatedly applying property (vi) of Proposition II.25, we finally obtain that $S_l^n \subseteq \text{Pre}(\hat{q}_o, Bad)$. Hence, $S_l^n = \text{Pre}(\hat{q}_o, Bad)$. Since this holds for all $n > n_a$, Algorithm II.30 terminates for all modes in a $type(1)$ kernel set. □

Theorem II.39. *Algorithm II.30 terminates if all the kernel sets $ker_1, \ldots, ker_p$ have a maximal element with respect to the partial order $(\hat{Q}, \subseteq)$.*

Proof. We first show that Algorithm II.30 is such that if $\hat{q}_i$ is in $type(1)$ kernel set, then there is K_i^* such that $S_i^{K_i^*} = S_i^{K_i^*+1}$ (base case). We then use the induction argument to prove that if for $\hat{q}_i$ in $type(1), \ldots, type(n)$ kernel sets there is K_i^* such that $S_i^{K_i^*} = S_i^{K_i^*+1}$, then for $\hat{q}_j$ in $type(n+1)$ kernel sets there is K_j^* such that $S_j^{K_j^*} = S_j^{K_j^*+1}$ (induction step). Since n is arbitrary and $ker_1 \cup ... \cup ker_p = \{\hat{q}_1, ..., \hat{q}_M\}$, this leads to the conclusion that there is

a K^* such that for all $\hat{q}_i \in \{\hat{q}_1, ..., \hat{q}_M\}$ we have that $S_i^{K^*} = S_i^{K^*+1}$. This, in turn, implies termination of Algorithm II.30. Proposition II.38 shows the base case.

(Induction step.) By the induction assumption, Algorithm II.30 is such that for all $\hat{q}_i$ in $type(1), \ldots, type(n)$ kernel sets there is N^* such that $S_i^{N^*} = S_i^{N^*+1}$. Consider a mode $\hat{q}_l \in ker_a$ where ker_a is a $type(n+1)$ kernel set. Then from Algorithm II.30, we have that for all $J > 0$

$$S_l^J = \text{Pre}(\hat{q}_l, \bigcup_{\{l_1 | \hat{q}_{l_1} \in \hat{R}(\hat{q}_l, Y) \cap ker_a\}} S_{l_1}^{J-1} \bigcup_{\{l_1^* | \hat{q}_{l_1^*} \in \hat{R}(\hat{q}_l, Y) \setminus ker_a\}} S_{l_1^*}^{J-1}), \tag{2.4}$$

in which $\hat{q}_{l_1^*}$ belongs to $type(1), \ldots, type(n)$ kernel sets. By the induction assumption, we thus have that $S_{l_1^*}^N = S_{l_1^*}^{N+1} = S_{l_1^*}^{N^*}$ for all $N > N^*$. Let $\hat{q}_l$ transit to $\hat{q}_o$ in N_1 transitions. Starting from mode $\hat{q}_o$, there is an input signal $\mathbf{y}$ such that the discrete flow is taken back to $\hat{q}_o$ after visiting all the states of ker_a as $\hat{q}_o$ is in ker_a. Let N_2 be the number of transitions that the discrete flow makes in doing so. Let $i \geq N^* + N_1 + N_2$.

Let $\hat{q}_o$ be the maximal element of ker_a, then by employing property property (iv) of Proposition II.25 to the right hand side of 2.4 for $J = i$, we obtain

$$S_l^i \subseteq \text{Pre}\left(\hat{q}_o, \bigcup_{\{l_1 | \hat{q}_{l_1} \in \hat{R}(\hat{q}_l, Y) \cap ker_a\}} S_{l_1}^{i-1} \bigcup_{\{l_1^* | \hat{q}_{l_1^*} \in \hat{R}(\hat{q}_l, Y) \setminus ker_a\}} S_{l_1^*}^{i-1}\right). \tag{2.5}$$

Substituting $S_{l_1}^{i-1} = \text{Pre}(\hat{q}_{l_1}, \bigcup_{\{l_2 | \hat{q}_{l_2} \in \hat{R}(\hat{q}_{l_1}, Y) \cap ker_a\}} S_{l_2}^{i-2} \bigcup_{\{l_2^* | \hat{q}_{l_2^*} \in \hat{R}(\hat{q}_{l_1}, Y) \setminus ker_a\}} S_{l_2^*}^{i-2})$ in equation (2.5), we obtain

$$S_l^i \subseteq \text{Pre}\left(\hat{q}_o, \bigcup_{\{l_1^* | \hat{q}_{l_1^*} \in \hat{R}(\hat{q}_l, Y) \setminus ker_a\}} S_{l_1^*}^{i-1} \bigcup_{\{l_1 | \hat{q}_{l_1} \in \hat{R}(\hat{q}_l, Y) \cap ker_a\}} \text{Pre}\left(\hat{q}_{l_1}, \bigcup_{\{l_2 | \hat{q}_{l_2} \in \hat{R}(\hat{q}_{l_1}, Y) \cap ker_a\}} S_{l_2}^{i-2} \bigcup_{\{l_2^* | \hat{q}_{l_2^*} \in \hat{R}(\hat{q}_{l_1}, Y) \setminus ker_a\}} S_{l_2^*}^{i-2}\right)\right). \tag{2.6}$$

Employing property (vi) of Proposition II.25, we obtain

$$S_l^i \subseteq \mathrm{Pre}\Bigg(\hat{q}_o, \bigcup_{\{l_1^*|\hat{q}_{l_1^*}\in\hat{R}(\hat{q}_l,Y)\setminus ker_a\}} S_{l_1^*}^{i-1} \bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{l_2^*|\hat{q}_{l_2^*}\in\hat{R}(\hat{q}_{l_1},Y)\setminus ker_a\}} S_{l_2^*}^{i-2} \bigcup_{\{\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{l_2|\hat{q}_{l_2}\in\hat{R}(\hat{q}_{l_1},Y)\cap ker_a\}} S_{l_2}^{i-2}\Bigg), \tag{2.7}$$

which leads to $S_l^i \subseteq \mathrm{Pre}\Big(\hat{q}_o, \mathcal{S}_{l_1^*}^i \cup \mathcal{S}_{l_2^*}^i \bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{l_2|\hat{q}_{l_2}\in\hat{R}(\hat{q}_{l_1},Y)\cap ker_a\}} S_{l_2}^{i-2}\Big)$, in which $\mathcal{S}_{l_1^*}^i = \bigcup_{\{l_1^*|\hat{q}_{l_1^*}\in\hat{R}(\hat{q}_l,Y)\setminus ker_a\}} S_{l_1^*}^{i-1}$ and $\mathcal{S}_{l_2^*}^i = \bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{l_2^*|\hat{q}_{l_2^*}\in\hat{R}(\hat{q}_{l_1},Y)\setminus ker_a\}} S_{l_2^*}^{i-2}$.

Proceeding by repeatedly expanding $S_{l_m}^{i-m}$ for $m = 2, \ldots, i-1$ and employing property (vi) of Proposition II.25, we finally obtain that

$$S_l^i \subseteq \mathrm{Pre}\Bigg(\hat{q}_o, \mathcal{S}_{l_1^*} \cup \ldots \cup \mathcal{S}_{l_i^*} \bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{\hat{q}_{l_2}\in\hat{R}(\hat{q}_{l_1},Y)\cap ker_a\}} \ldots \bigcup_{\{l_i|\hat{q}_{l_i}\in\hat{R}(\hat{q}_{l_{i-1}},Y)\cap ker_a\}} \mathrm{Pre}(\hat{q}_{l_i}, Bad)\Bigg), \tag{2.8}$$

where $\mathcal{S}_{l_m^*}^i = \bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \cdots \bigcup_{\{l_{m-1}|\hat{q}_{l_{m-1}}\in\hat{R}(\hat{q}_{l_{m-2}},Y)\cap ker_a\}} \bigcup_{\{l_m^*|\hat{q}_{l_m^*}\in\hat{R}(\hat{q}_{l_{m-1}},Y)\setminus ker_a\}} S_{l_m^*}^{i-m}$ for $m \leq i$.

Since $\hat{q}_{l_m^*} \notin ker_a$, $\hat{q}_{l_m^*}$ belongs to a kernel set of type less than or equal to n. As a consequence, we have that $S_{l_m^*}^{i-m} = S_{l_m^*}^{j-m}$ for all $j \geq i$ if $i - m \geq N^*$, which in turn implies that $\mathcal{S}_{l_m^*}^i = \mathcal{S}_{l_m^*}^j := \mathcal{S}_{l_m^*}^*$ for all $j > i$ if $m \leq N_1 + N_2$.

Since $\{\bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \cdots \bigcup_{\{l_{m-1}|\hat{q}_{l_{m-1}}\in\hat{R}(\hat{q}_{l_{m-2}},Y)\cap ker_a\}} \bigcup_{\{l_m^*|\hat{q}_{l_m^*}\in\hat{R}(\hat{q}_{l_{m-1}},Y)\setminus ker_a\}} \hat{q}_{l_m^*}\}$ is the set of all modes not in ker_a that can be reached in one transition from modes in ker_a and $S_{l_m^*}^{j-m} \subseteq S_{l_m^*}^{i-m}$ for all $j \leq i$, we have that $\mathcal{S}_{l_{N_1+N_2+j}}^i \subseteq \bigcup_{k\in\{1,\ldots,N_1+N_2\}} \mathcal{S}_{l_k}^i$ for all $j > 0$. As a consequence, by applying property (iii) of Proposition II.25, we obtain that

$$S_l^i \subseteq \text{Pre}\Bigg(\hat{q}_o, \mathcal{S}^*_{l_1^*} \cup \ldots \cup \mathcal{S}^*_{l^*_{N_1+N_2}} \bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{\hat{q}_{l_2}\in\hat{R}(\hat{q}_{l_1},Y)\cap ker_a\}} \ldots$$
$$\bigcup_{\{l_i|\hat{q}_{l_i}\in\hat{R}(\hat{q}_{l_{i-1}},Y)\cap ker_a\}} \text{Pre}(\hat{q}_{l_i}, Bad)\Bigg). \tag{2.9}$$

Furthermore, according to our construction, starting from $\hat{q}_l$ the discrete flow reaches $\hat{q}_o$ in N_1 transitions and from $\hat{q}_o$ the discrete flow reaches $\hat{q}_o$ again in N_2 transitions after visiting all the modes in ker_a. Thus for $m \geq N_1 + N_2$ we have that

$$\{\bigcup_{\{l_1|\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \ldots \bigcup_{\{l_{m-1}|\hat{q}_{l_{m-1}}\in\hat{R}(\hat{q}_{l_{m-2}},Y)\cap ker_a\}} \hat{q}_{l_{m-1}}\} = ker_a$$

. As a consequence, $\bigcup_{\{\hat{q}_{l_1}\in\hat{R}(\hat{q}_l,Y)\cap ker_a\}} \bigcup_{\{\hat{q}_{l_2}\in\hat{R}(\hat{q}_{l_1},Y)\cap ker_a\}} \ldots \bigcup_{\{l_i|\hat{q}_{l_i}\in\hat{R}(\hat{q}_{l_{i-1}},Y)\cap ker_a\}} \text{Pre}(\hat{q}_{l_i}, Bad) = \bigcup_{\hat{q}\in ker_a} \text{Pre}(\hat{q}, Bad)$ so that equation (2.9) becomes

$$S_l^i \subseteq \text{Pre}\Bigg(\hat{q}_o, \mathcal{S}^*_{l_1^*} \cup \ldots \cup \mathcal{S}_{l^*_{N_1+N_2}} \bigcup_{\hat{q}\in ker_a} \text{Pre}(\hat{q}, Bad)\Bigg), \tag{2.10}$$

which further simplifies to $S_l^i \subseteq \text{Pre}(\hat{q}_o, \bigcup_{\{j\in\{1,\ldots,N_1+N_2\}} \mathcal{S}^*_{l_j^*})$ by property (vi) of Proposition II.25.

We next show that $S_l^i \supseteq \text{Pre}(\hat{q}_o, \bigcup_{\{j\in\{1,\ldots,N_1+N_2\}} \mathcal{S}^*_{l_j^*})$. To this end, note that $S_l^i \supseteq S_{l_{N_1}}^{i-N_1} = \text{Pre}(\hat{q}_{l_{N_1}}, \bigcup_{\{l_{N_1+1}|\hat{q}_{l_{N_1+1}}\in\hat{R}(\hat{q}_{l_{N_1}},Y)\cap ker_a\}} S_{l_{N_1+1}}^{i-N_1+1} \bigcup_{\{l^*_{N_1+1}|\hat{q}^*_{l_{N_1+1}}\in\hat{R}(\hat{q}_{l_{N_1}},Y)\setminus ker_a\}} S_{l^*_{N_1+1}}^{i-N_1+1})$, for $\hat{q}_{l_{N_1}}$ any mode reached from $\hat{q}_l$ in N_1 transitions. Since $\hat{q}_o$ is reachable from $\hat{q}_l$ in N_1 transitions, we have that

$$S_l^i \supseteq \text{Pre}\Bigg(\hat{q}_o, \bigcup_{\{\hat{q}_{l_{N_1-1}}\in\hat{R}(\hat{q}_{l_{N_1}},Y)\cap ker_a\}} S_{l_{N_1-1}}^{i-N_1-1} \bigcup_{\{\hat{q}_{l^*_{N_1-1}}\in\hat{R}(\hat{q}_{l_{N_1}},Y)\setminus ker_a\}} S_{l^*_{N_1-1}}^{i-N_1-1}\Bigg). \tag{2.11}$$

Simplifying the right hand side of equation 2.11 by repeatedly employing equation (2.4) in the argument of the Pre operator and using property (vi) of Proposition II.25, we finally obtain that $S_l^i \supseteq \text{Pre}(\hat{q}_o, \bigcup_{\{j\in\{1,\ldots,N_1+N_2\}} \mathcal{S}^*_{l_j^*}})$.

Thus $S_l^i = \text{Pre}(\hat{q}_o, \bigcup_{\{j\in\{1,\ldots,N_1+N_2\}} \mathcal{S}^*_{l_j^*}})$. Since this result holds for all $i \geq N^* + N_1 + N_2$, we have that $S_l^i = S_l^{i+1}$ for all $i \geq N^* + N_1 + N_2$, so that the proof of the induction step is concluded. □

This theorem provides an easily checkable sufficient condition for the termination of Algorithm 1 based on the structure of the map $\hat{R}$. Note that a corollary of this theorem is that if system $\hat{H}$ is such that all of its kernel sets are singletons in $\hat{Q}$, then Algorithm 1 terminates for $\hat{H}$. Here, we illustrate the logic of the proof and the concept of kernel set on a simple example.

Example II.40. Consider a simple instance of $(\hat{R}, \hat{Q}, Y)$ in which $\hat{Q} = \{\hat{q}_1, \hat{q}_2\}$, $Y = \{\epsilon, y^*\}$, $\hat{R}(\hat{q}_1, y^*) = \hat{q}_2$, and $\hat{R}(\hat{q}_2, y^*) = \hat{q}_1$. That is, we have one kernel set equal to $\{\hat{q}_1, \hat{q}_2\}$. Because of the loop between $\hat{q}_1$ and $\hat{q}_2$, Algorithm 1 may not terminate. Here, we show that if we assume that, for example, $\hat{q}_2 \subseteq \hat{q}_1$, then Algorithm 1 terminates in three steps. In this example, we have that $S = (S_1, S_2)$ and $G(S) = (\text{Pre}(\hat{q}_1, S_2 \cup Bad), \text{Pre}(\hat{q}_2, S_1 \cup Bad))$. Hence, $S^1 = G(\emptyset) = (\text{Pre}(\hat{q}_1, Bad), \text{Pre}(\hat{q}_2, Bad))$, and $S^2 = G(S^1) = (\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad)), \text{Pre}(\hat{q}_2, \text{Pre}(\hat{q}_1, Bad)))$. Consider S^2. On the one hand, we have that $\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad)) \subseteq \text{Pre}(\hat{q}_1, Bad)$ by properties (iv) and (ii) of Proposition II.25. On the other hand, we have that $\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad)) \supseteq \text{Pre}(\hat{q}_1, Bad)$ by property (iii) of Proposition II.25. Hence, we must have that $S_1^2 = \text{Pre}(\hat{q}_1, Bad)$. Similar reasonings lead to $S_2^2 = \text{Pre}(\hat{q}_1, Bad)$. This leads to $S^3 = G(S^2) = (\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_1, Bad)), \text{Pre}(\hat{q}_2, \text{Pre}(\hat{q}_1, Bad)))$, which, employing again the properties of the Pre operator, leads to $S^3 = (\text{Pre}(\hat{q}_1, Bad), \text{Pre}(\hat{q}_1, Bad))$. This set is, in turn, equal to S^2 and therefore Algorithm 1 terminates in three steps.

2.5.1 Proving termination through abstraction

When not all kernel sets have a maximal element, Theorem II.39 does not hold. However, for any estimator $\hat{H}$, one can construct an abstraction of $\hat{H}$, denoted $\hat{H}^a$, for which Algorithm 1 terminates and such that the fixed point gives the mode-dependent capture sets of $\hat{H}$. This abstraction is constructed by merging all the modes of $\hat{H}$ that belong to the same kernel set in a unique new mode as follows.

Definition II.41. Given hybrid system $\hat{H} = (\hat{Q}, X, U, D, Y, \hat{R}, \hat{f})$, the abstraction $\hat{H}^a = (\hat{Q}^a, X, U, D, Y^a, \hat{R}^a, \hat{f}^a)$ is a hybrid system with uncontrolled mode transitions such that

(i) $\hat{Q}^a = \{\hat{q}^a_1, ..., \hat{q}^a_p\}$, Y^a such that $\epsilon \in Y^a$ and $\hat{R}(\hat{q}^a, \epsilon) = \hat{q}^a$ for all $\hat{q}^a \in \hat{Q}^a$;

(ii) for all $i, j \in \{1, ..., p\}$ there is $y^a \in Y^a$ such that $\hat{q}^a_i = \hat{R}^a(\hat{q}^a_j, y^a)$ if and only if there are $\hat{q}' \in ker_i$, $\hat{q} \in ker_j$, and $y \in Y$ such that $\hat{q}' = \hat{R}(\hat{q}, y)$;

(iii) for all $i \in \{1, ..., p\}$, $x \in X$, $d \in D$, and $v \in U$, we have that $\hat{f}^a(x, \hat{q}^a_i, v, d) := \bigcup_{\hat{q} \in ker_i} \hat{f}(x, \hat{q}, v, d)$.

For a feedback map $\hat{\pi}^a : \hat{Q}^a \times X \to U$, initial states $x_o \in X$ and $\hat{q}^a_o \in \hat{Q}^a$, and signals $\mathbf{y^a}$, $\mathbf{d}$, we denote the flows of the closed loop system $\hat{H}^{a,\hat{\pi}^a}$ by $\phi_{\hat{q}^a}(t, \hat{q}^a_o, \mathbf{y^a})$ and $\phi^{\hat{\pi}^a}_{\hat{x}^a}(t, (\hat{q}^a_o, x_o), \mathbf{d}, \mathbf{y^a})$, in which $\hat{x}^a(t) := \phi^{\hat{\pi}^a}_{\hat{x}^a}(t, (\hat{q}^a_o, x_o), \mathbf{d}, \mathbf{y^a})$ satisfies $\dot{\hat{x}}^a(t) \in \hat{f}^a(\hat{x}^a(t), \phi_{\hat{q}^a}(t, \hat{q}^a_o, \mathbf{y^a}), \hat{\pi}^a(\phi_{\hat{q}^a}(t, \hat{q}^a_o, \mathbf{y^a}), \hat{x}^a), d(t))$. We also denote by $\hat{C}^a_{\hat{q}^a_i}$ for $i \in \{1, ..., p\}$ the mode-dependent capture sets of $\hat{H}^a$. For any $\hat{q}^a \in \hat{Q}^a$, we define $ker(\hat{q}^a) := ker_i$ provided $\hat{q}^a = \hat{q}^a_i$. Also, for all $\hat{q}^a \in \hat{Q}^a$, we denote the set of reachable modes from $\hat{q}^a$ as $\widehat{\text{Reach}}^a(\hat{q}^a) := \bigcup_{t \geq 0} \bigcup_{\mathbf{y^a}} \phi_{\hat{q}^a}(t, \hat{q}^a, \mathbf{y^a})$. In the sequel, we denote $\hat{R}^a(\hat{q}^a, Y^a) := \bigcup_{y^a \in Y^a} \hat{R}^a(\hat{q}^a, y^a)$, in which we set $\hat{R}^a(\hat{q}^a, y^a) := \hat{q}^a$ if $\hat{R}^a(\hat{q}^a, y^a)$ is not defined for some $y^a \in Y^a$. The following proposition is a direct consequence of Theorem II.39 and of the fact that all kernel sets of $\hat{H}^a$ are singletons.

Proposition II.42. *Algorithm 1 terminates for system $\hat{H}^a$.*

The next result shows that any piecewise continuous signal, which is continuous from

the right and contained in $ker(\phi_{\hat{q}^a}(t, \hat{q}_o^a, \mathbf{y^a}))$ is a possible discrete flow of $\hat{H}$ for suitable $\mathbf{y}$ starting from some $\hat{q}_o \in ker(\hat{q}_o^a)$.

Proposition II.43. *For any piecewise continuous signal $\boldsymbol{\alpha}$ that is continuous from the right and such that $\alpha(t) \in ker(\phi_{\hat{q}^a}(t, \hat{q}_o^a, \mathbf{y^a}))$, there are $\hat{q}_o \in ker(\hat{q}_o^a)$ and $\mathbf{y}$ such that $\alpha(t) = \phi_{\hat{q}}(t, \hat{q}_o, \mathbf{y})$ for all t.*

Proof. Since $\alpha(t) \in ker(\phi_{\hat{q}^a}(t, \hat{q}_o^a, \mathbf{y^a}))$ for all t, there are times $t_0, ..., t_N \leq t$ and a sequence $j_0, ..., j_N \in \{1, ..., p\}$ such that $\alpha(t) \in ker_{j_i}$ for all $t \in [t_i, t_{i+1})$. Since any mode in ker_{j_i} can transit to any other mode in ker_{j_i} instantaneously under the discrete transitions of $\hat{H}$, we have that there are $\hat{q}_{o,i} \in ker_{j_i}$ and $\mathbf{y}_i$ such that $\alpha(t) = \phi_{\hat{q}}(t - t_i, \hat{q}_{o,i}, \mathbf{y}_i)$ for all $t \in [t_i, t_{i+1})$. Also, for any two modes $\alpha_i \in ker_{j_i}$ and $\alpha_{i+1} \in ker_{j_{i+1}}$ we have that $\alpha_{i+1} \in \widehat{\text{Reach}}(\alpha_i)$. Hence, let $\alpha_i^- := \lim_{t \to t_{i+1}^-} \phi_{\hat{q}}(t - t_i, \hat{q}_{o,i}, \mathbf{y}_i)$ and $\alpha_i^+ := \lim_{t \to t_{i+1}^+} \phi_{\hat{q}}(t - t_{i+1}, \hat{q}_{o,i+1}, \mathbf{y}_{i+1})$. Then, since multiple transitions are possible in $\hat{H}$ at the same time, there is a signal $\mathbf{y}_{i,i+1}$ such that $\alpha_i^+ = \phi_{\hat{q}}(0, \alpha_i^-, \mathbf{y}_{i,i+1})$. Hence, there is a signal $\mathbf{y}$ such that $\alpha(t) = \phi_{\hat{q}}(t, \hat{q}_{o,0}, \mathbf{y})$ for all t. □

Theorem II.44. *For all kernel sets ker_i with $i \in \{1, ..., p\}$ and for all $\hat{q} \in ker_i$, we have that $\hat{C}_{\hat{q}} = \hat{C}^a_{\hat{q}_i^a}$.*

Proof. Let $\hat{q} \in ker_i$. We first show that $\hat{C}_{\hat{q}} \subseteq \hat{C}^a_{\hat{q}_i^a}$. Let $x_o \in \hat{C}_{\hat{q}}$, then for all $\hat{\pi} : \hat{Q} \times X \to U$, there are $\mathbf{y}$, $\mathbf{d}$, and $t > 0$ such that $\phi^{\hat{\pi}}_{\hat{x}}(t, (\hat{q}, x), \mathbf{d}, \mathbf{y}) \in Bad$. This is in particular true for all those feedback maps $\hat{\pi}$ such that $\hat{\pi}(\hat{q}, x) = \hat{\pi}(\hat{q}', x)$ whenever $\hat{q}, \hat{q}' \in ker_j$ for some $j \in \{1, ..., p\}$. Hence, we also have that for all $\hat{\pi}^a : \hat{Q}^a \times X \to U$, there are $\mathbf{y}$, $\mathbf{d}$, and $t > 0$ such that $\hat{x}(t) := \phi^{\hat{\pi}^a}_{\hat{x}}(t, (\hat{q}, x), \mathbf{d}, \mathbf{y}) \in Bad$, in which $\dot{\hat{x}} \in \hat{f}(\hat{x}(t), \phi_{\hat{q}}(t, \hat{q}, \mathbf{y}), \hat{\pi}^a(\alpha(t), x(t)), d(t))$ with $\alpha(t) := \hat{q}_j^a$ if $\phi_{\hat{q}}(t, \hat{q}, \mathbf{y}) \in ker_j$. Such a signal $\hat{x}(t)$ also satisfies $\dot{\hat{x}} \in \hat{f}^a(\hat{x}(t), \alpha(t), \hat{\pi}^a(\alpha(t), x(t)), d(t))$ by the definition of $\hat{f}^a$. By the definition of $\hat{R}^a$, there is $\mathbf{y^a}$ such that $\alpha(t) = \phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a})$ for all t. Hence, $\hat{x}(t)$ is also a continuous flow of $\hat{H}^a$ starting at $(\hat{q}_i^a, x_o)$ and therefore $x_o \in \hat{C}^a_{\hat{q}_i^a}$.

We now show that $\hat{C}^a_{\hat{q}_i^a} \subseteq \hat{C}_{\hat{q}}$. If $x_o \in \hat{C}^a_{\hat{q}_i^a}$, then for all feedback maps $\hat{\pi}^a : \hat{Q}^a \times X \to U$, there are $\mathbf{y^a}$, $\mathbf{d}$, and $t > 0$ such that $\hat{x}^a(t) := \phi^{\hat{\pi}^a}_{\hat{x}^a}(t, (\hat{q}_i^a, x_o), \mathbf{y^a}, \mathbf{d}) \in Bad$. Here, we have

that $\hat{x}^a(t)$ satisfies $\dot{\hat{x}}^a(t) \in \hat{f}^a(\hat{x}^a(t), \phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a}), \hat{\pi}^a(\phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a}), \hat{x}^a), d(t))$, which is equivalent (by the definition of $\hat{f}^a$) to $\dot{\hat{x}}^a(t) \in \hat{f}(\hat{x}^a(t), ker(\phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a})), \hat{\pi}^a(\phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a}), \hat{x}^a), d(t))$, which is equivalent to $\dot{\hat{x}}^a(t) = \hat{f}(\hat{x}^a(t), \alpha(t), \hat{\pi}^a(\phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a}), \hat{x}^a), d(t))$ for piecewise continuous signal $\boldsymbol{\alpha}$ (continuous from the right) such that $\alpha(t) \in ker(\phi_{\hat{q}^a}(t, \hat{q}_i^a, \mathbf{y^a}))$. By Proposition II.43, any such $\alpha(t)$ is such that there are $\mathbf{y}$ and $\hat{q}_o \in ker(\hat{q}_i^a)$ such that $\alpha(t) = \phi_{\hat{q}}(t, \hat{q}_o, \mathbf{y})$ for all t, that is, it is a discrete flow of system $\hat{H}$. Hence, for all $\pi' : \hat{Q} \times X \to U$ with $\hat{\pi}'(\hat{q}, x) = \hat{\pi}'(\hat{q}', x)$ for all $\hat{q}, \hat{q}' \in ker_j$ for all j, there are $\mathbf{y}$, $\mathbf{d}$, $\hat{q}_o \in ker_i$, such that $\phi_{\hat{x}}^{\hat{\pi}'}(t, (\hat{q}_o, x_o), \mathbf{y}, \mathbf{d}) \in Bad$. By Proposition II.36, this implies that for all $\pi : \hat{Q} \times X \to U$ there are $\mathbf{y}$, $\mathbf{d}$, $\hat{q}_o \in ker_i$, such that $\phi_{\hat{x}}^{\hat{\pi}}(t, (\hat{q}_o, x_o), \mathbf{y}, \mathbf{d}) \in Bad$. Hence, $x_o \in \hat{C}_{\hat{q}_o}$. □

The above theorem provides a useful result for the computation of the mode-dependent capture sets of $\hat{H}$. In particular, one constructs the abstraction $\hat{H}^a$ and applies Algorithm 1 to it. Algorithm 1 is in turn always guaranteed to terminate for system $\hat{H}^a$. The result (by Theorem II.44) provides the sets $\hat{C}_{\hat{q}}$. Hence, $\hat{H}^a$ can be considered only as a structural abstraction as it does not provide an over-approximation of the capture set of $\hat{H}$, but provides it exactly.

The next two technical propositions provide a characterization of the Pre operator computed for system $\hat{H}^a$ and the relationship between $\hat{R}^a$ and R. Specifically, denote the predecessor operator for system $\hat{H}^a$ by $\mathrm{Pre}^a(\hat{q}^a, S)$ for some $S \subseteq X$ as $\mathrm{Pre}^a(\hat{q}^a, S) := \{x_o \in X \mid \forall\, \hat{\pi}^a \;\exists\, t, \mathbf{d},$ s.t. $\phi_{\hat{x}^a}^{\hat{\pi}^a}(t, (\hat{q}^a, x_o), \mathbf{d}, \epsilon) \in S\}$.

Proposition II.45. *For all $\hat{q}^a \in \hat{Q}^a$ and $S \subseteq X$, we have that $Pre^a(\hat{q}^a, S) = Pre(\bigvee ker(\hat{q}^a), S)$.*

Proof. From the definition of $\mathrm{Pre}^a(\hat{q}^a, S)$, we have that $x_o \in \mathrm{Pre}^a(\hat{q}^a, S)$ if and only if for all $\hat{\pi}^a$, there are $t, \mathbf{d}$ such that $\hat{x}^a(t) = \phi_{\hat{x}^a}^{\hat{\pi}^a}(t, (\hat{q}^a, x_o), \mathbf{d}, \epsilon) \in S$, in which $\dot{\hat{x}}^a(t) \in \hat{f}^a(\hat{x}^a(t), \hat{q}^a, \hat{\pi}^a(\hat{x}^a(t)), d(t))$, which, by the definition of $\hat{f}^a$ and of $\hat{f}$ is equivalent to $\dot{\hat{x}}^a(t) \in f(\hat{x}^a(t), \bigcup_{\hat{q} \in ker(\hat{q}^a)} \bigcup_{q \in \hat{q}} q, \hat{\pi}^a(\hat{x}^a(t)), d(t)) = f(\hat{x}^a(t), \bigvee ker(\hat{q}^a), \hat{\pi}^a(\hat{x}^a(t)), d(t))$. Hence, by the definition of Pre, we have that $x_o \in \mathrm{Pre}^a(\hat{q}^a, S)$ if and only if $x_o \in \mathrm{Pre}(\bigvee ker(\hat{q}^a), S)$. □

Proposition II.46. *Let $\hat{q}_{j_1}^a, \hat{q}_{j_0}^a \in \hat{Q}^a$. If $\hat{q}_{j_1}^a \in \hat{R}^a(\hat{q}_{j_0}^a, Y^a)$ then $\bigvee ker(\hat{q}_{j_1}^a) \subseteq Reach(\bigvee ker(\hat{q}_{j_0}^a))$.*

Proof. If $\hat{q}^a_{j_1} \in \hat{R}^a(\hat{q}^a_{j_0}, Y^a)$, then by the definition of $\hat{R}^a$ there are $\hat{q} \in ker(\hat{q}^a_{j_0})$ and $\hat{q}' \in ker(\hat{q}^a_{j_1})$ such that $\hat{q}' = \hat{R}(\hat{q}, y)$ for some $y \in Y$. By the definition of a kernel set, this also implies that for all $\hat{q} \in ker(\hat{q}^a_{j_0})$ and $\hat{q}' \in ker(\hat{q}^a_{j_1})$, there is a sequence of events $y_1, ..., y_k$ and of modes $\hat{q}_{j_0}, ..., \hat{q}_{j_k} \in \hat{Q}$ such that $\hat{q}_{j_0} = \hat{q}$, $\hat{q}_{j_k} = \hat{q}'$ and $\hat{q}_{j_{i+1}} = \hat{R}(\hat{q}_{j_i}, y_{i+1})$ for $i \in \{0, ..., k-1\}$. Since $\hat{R}(\hat{q}, y) \subseteq \text{Reach}(\hat{q})$ for all $y \in Y$ and $\hat{q} \in \hat{Q}$, this in turn implies that $\hat{q}_{j_{i+1}} \subseteq \text{Reach}(\hat{q}_{j_i})$ for $i \in \{0, ..., k-1\}$. This leads to $\hat{q}' \subseteq \text{Reach}(\hat{q})$ for all $\hat{q} \in ker(\hat{q}^a_{j_0})$ and $\hat{q}' \in ker(\hat{q}^a_{j_1})$. This also implies that $\hat{q}' \subseteq \text{Reach}(\bigvee ker(\hat{q}^a_{j_0}))$ and hence (since this holds for all $\hat{q}' \in ker(\hat{q}^a_{j_1})$) to $\bigvee ker(\hat{q}^a_{j_1}) \subseteq \text{Reach}(\bigvee ker(\hat{q}^a_{j_0}))$. □

Lemma II.47. For all $\overline{q} \in \hat{Q}$, we have that $\hat{C}_{\overline{q}} = \text{Pre}(\text{Reach}(\overline{q}), Bad)$.

Proof. First, we show that $\hat{C}_{\overline{q}} \subseteq \text{Pre}(\text{Reach}(\overline{q}), Bad)$. Since Algorithm II.30 terminates in a finite number n of steps for $\hat{H}^a$, we have that

$$\hat{C}^a_{\hat{q}^a} = \text{Pre}^a\left(\hat{q}^a, \bigcup_{\hat{q}^a_{j_1} \in \hat{R}^a(\hat{q}^a, Y^a)} \text{Pre}^a\left(\hat{q}^a_{j_1}, \bigcup_{\hat{q}^a_{j_2} \in \hat{R}^a(\hat{q}^a_{j_1}, Y^a)} \text{Pre}^a\left(\hat{q}^a_{j_2}, ... \bigcup_{\hat{q}^a_{j_{n-1}} \in \hat{R}^a(\hat{q}^a_{j_{n-2}}, Y^a)} \text{Pre}^a(\hat{q}^a_{j_{n-1}}, Bad)...\right)\right)\right).$$

By Proposition II.45, we also have that $\hat{C}^a_{\hat{q}^a} = \text{Pre}\Big(\bigvee ker(\hat{q}^a), \bigcup_{\hat{q}^a_{j_1} \in \hat{R}^a(\hat{q}^a, Y^a)} \text{Pre}\Big(\bigvee ker(\hat{q}^a_{j_1}), \bigcup_{\hat{q}^a_{j_2} \in \hat{R}^a(\hat{q}^a_{j_1}, Y^a)} \text{Pre}\Big(\bigvee ker(\hat{q}^a_{j_2}), ... \bigcup_{\hat{q}^a_{j_{n-1}} \in \hat{R}^a(\hat{q}^a_{j_{n-2}}, Y^a)} \text{Pre}(\bigvee ker(\hat{q}^a_{j_{n-1}}), Bad)...\Big)\Big)\Big)$. We have from Proposition II.46 that $\bigvee ker(\hat{q}^a_{j_1}) \subseteq \text{Reach}(\bigvee ker(\hat{q}^a))$ and, for $i < n$, that $\bigvee ker(\hat{q}^a_{j_{i+1}}) \subseteq \text{Reach}(\bigvee ker(\hat{q}^a_{j_i}))$. Since the Pre operator and Reach preserve the inclusion relation in the first argument, these imply that $\hat{C}^a_{\hat{q}^a} \subseteq \text{Pre}(\text{Reach}(\bigvee ker(\hat{q}^a)), Bad)$. Since for all $\overline{q}_1, \overline{q}_2 \in ker(\hat{q}^a)$ we have that $\text{Reach}(\overline{q}_1) = \text{Reach}(\overline{q}_2)$, we also have that $\text{Reach}(\overline{q}) = \text{Reach}(\bigvee ker(\hat{q}^a))$ for all $\overline{q} \in ker(\hat{q}^a)$. Hence, $\hat{C}^a_{\hat{q}^a} \subseteq \text{Pre}(\text{Reach}(\overline{q}), Bad)$ for all $\overline{q} \in ker(\hat{q}^a)$. This along with Theorem II.44 finally imply that for all $\overline{q} \in ker(\hat{q}^a)$ we have $\hat{C}_{\overline{q}} \subseteq \text{Pre}(\text{Reach}(\overline{q}), Bad)$.

To show that $\hat{C}_{\overline{q}} \supseteq \text{Pre}(\text{Reach}(\overline{q}), Bad)$, we employ the properties of the Pre operator and Proposition II.26. By such a proposition, by the fact that (since $\hat{H}$ is an estimator for

H) for all $\bar{q} \in \hat{Q}$ there is $y \in Y$ such that $\hat{R}(\bar{q}, y) = \text{Reach}(\bar{q})$, and by property (iii) of Proposition II.25, it follows that $\hat{C}_{\bar{q}} \supseteq \text{Pre}(\bar{q}, \hat{C}_{\text{Reach}(\bar{q})})$. In turn we have that $\hat{C}_{\text{Reach}(\bar{q})} \supseteq \text{Pre}(\text{Reach}(\bar{q}), \mathit{Bad})$ by Proposition II.26 and property (iii) of Proposition II.25. Hence, we have that $\hat{C}_{\bar{q}} \supseteq \text{Pre}(\bar{q}, \text{Pre}(\text{Reach}(\bar{q}), \mathit{Bad}))$, which by property (i) of Proposition II.26 leads to $\hat{C}_{\bar{q}} \supseteq \text{Pre}(\text{Reach}(\bar{q}), \mathit{Bad})$. □

This result shows that the mode-dependent capture set $\hat{C}_{\bar{q}}$ can be computed by computing the Pre operator only once as opposed to being determined through a (finite, by Theorem II.44 and Proposition II.42) iteration of Pre operator computations. Exact computation of Pre for general dynamics is not always possible. However, there are a number of works that have focused on the exact computation of uncontrollable predecessor operators for restricted classes of systems. For example, the work of [83] shows that Pre can be exactly computed for special classes of linear systems; [84] further extends this result to linear hybrid systems; [85] shows that Pre is exactly computable also for triangular hybrid systems. Finally, [50, 33] show that Pre is computable with a linear complexity algorithm for classes of order preserving systems. Based on these results and on Lemma II.47, we conclude that Problem II.18 is *decidable* when for each mode $\bar{q} \in \hat{Q}$ the continuous dynamics $\dot{x} \in f(x, \bar{q}, u, d),\ d \in D$ belong to one of the above cited classes of systems. Since the application example falls in the class of systems described in [50, 33], we summarize the main result here. For this sake, we restrict the structure of H and *Bad* to that of a two-agent game.

Definition II.48. The pair (H, Bad) has the form of a two-agent game provided $H = H^1 \parallel H^2$ with $H^i = (Q^i, X^i, U^i, D^i, \Sigma^i, R^i, f^i)$ for $i \in \{1, 2\}$ with $Q^1 = \emptyset$, $D^1 = \emptyset$, $\Sigma^1 = \emptyset$, $U^2 = \emptyset$, and $\mathit{Bad} = B^1 \times B^2$ with $B^i \subseteq X^i$.

The next result provides a class of systems for which $x \in \text{Pre}(\hat{q}, \mathit{Bad})$ for $\hat{q} \in \hat{Q}$ can be exactly determined in finite time. This result is taken from [50, 33] and is here stated as a proposition.

Proposition II.49. *Let* (H, Bad) *be in the form of a two-agent game. Assume that*

(i) $U^1 = [u_L, u_H] \subseteq \mathbb{R}$*; the flow of* H^1 *denoted* $\phi^1(t, \cdot, \cdot) : X \times \mathcal{S}(U) \to X$ *is an order preserving function in both arguments; there is* $\zeta > 0$ *such that* $f_1^1(x^1, u) \geq \zeta$*;* $B^1 = B_1^1 \times \mathbb{R}^{n_1 - 1}$*;*

(ii) For $\hat{q} \in \hat{Q}$ *there are* $\theta_L, \theta_U \in \mathbb{R}$ *and a function* $\bar{f} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ *such that* $\{f^2(x^2, \hat{q}, d) \mid d \in D^2\} = \{\bar{f}(x^2, \theta) \mid \theta \in [\theta_L, \theta_U]\}$*; the flow of* $\dot{x}^2 = \bar{f}(x^2, \theta)$*, that is,* $\phi^2(t, \cdot, \cdot) : X \times \mathcal{S}([\theta_L, \theta_U]) \to X$*, is an order preserving map in both arguments; there is* $\zeta > 0$ *such that* $\bar{f}_1(x^2, u) \geq \zeta$*;* $B^2 = B_1^2 \times \mathbb{R}^{n_2 - 1}$*.*

Then, $Pre(\hat{q}, Bad) = Pre(\hat{q}, Bad)_L \cap Pre(\hat{q}, Bad)_H$*, in which* $Pre(\hat{q}, Bad)_L = \{x_o \in X \mid \exists\, t,\ \mathbf{d}$ *s.t. some* $\phi_{\hat{x}}(t, (\hat{q}, x_o), \mathbf{d}, u_L, \epsilon) \in Bad\}$ *and* $Pre(\hat{q}, Bad)_H = \{x_o \in X \mid \exists\ t,\ \mathbf{d}$ *s.t. some* $\phi_{\hat{x}}(t, (x_o, \hat{q}), \mathbf{d}, u_H, \epsilon) \in Bad\}$*. A feedback map* $\hat{\pi}(\hat{q}, x) \in \Pi(\hat{q}, x)$ *is given by*

$$\hat{\pi}(\hat{q}, x) := \begin{cases} u_L & if\ x \in Pre(\hat{q}, Bad)_H \wedge x \in \partial Pre(\hat{q}, Bad)_L \\ u_H & if\ x \in Pre(\hat{q}, Bad)_L \wedge x \in \partial Pre(\hat{q}, Bad)_H \\ u_L & if\ x \in \partial Pre(\hat{q}, Bad)_L \wedge \partial Pre(\hat{q}, Bad)_L \\ * & otherwise. \end{cases} \tag{2.12}$$

By virtue of this result, one can avoid computing the set $\text{Pre}(\hat{q}, Bad)$, which requires optimization over the space of control inputs. One can instead compute the sets $\text{Pre}(\hat{q}, Bad)_L$ and $\text{Pre}(\hat{q}, Bad)_H$, which, since the control input is fixed and the flow preserves the ordering, can be computed by linear complexity algorithms. The structure of the set *Bad* well models collision configurations between agents sharing a common space as illustrated in the application examples of Section 2.7. We omit the details of the algorithms, which can be found elsewhere [33, 50] and instead present in Section 2.7 their application to a concrete example.

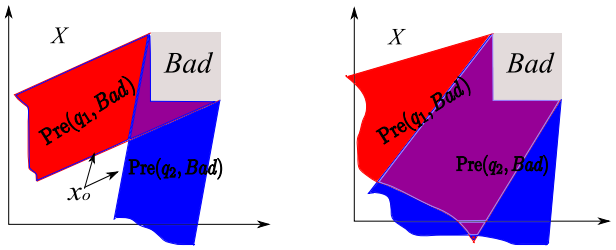


Figure 2.2: (Left) Example II.50, in which the continuous dynamics are given by equations (2.13). (Right) Example II.50, in which the continuous dynamics are given by equations (2.14). The set $\text{Pre}(q_1, Bad)$ is in red while the set $\text{Pre}(q_2, Bad)$ is in blue. Both sets extend to $-\infty$.

2.6 Equivalence between Problem II.12 and Problem II.18

Showing that Problem 1 is equivalent to Problem 2 is based on showing that for all $\bar{q} \in \hat{Q}$ we have that $\hat{C}_{\bar{q}} = C_{\bar{q}}$. In general, the set of possible continuous trajectories of system $\hat{H}$ for every mode $\bar{q} \subseteq Q$ contains but is not equal to the set of continuous trajectories possible in H. This is due to the fact that in H not all transitions may be possible among the modes in $\bar{q}$ due to the structure of R. This information was lost in the construction of $\hat{H}$ in order to obtain a hybrid system with uncontrolled mode transitions and *known* discrete/continuous state. In order to illustrate this point, consider the following example.

Example II.50. Consider system H with two modes q_1 and q_2 between which there is no transition and let the continuous dynamics for each mode be given, for $x \in \mathbb{R}^2$, by

$$\dot{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} u, \text{ for } q = q_1 \text{ and } \dot{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} u, \text{ for } q = q_2, \tag{2.13}$$

in which $u \in [0, 1]$ and $\bar{q}_o = \{q_1, q_2\}$. Let $Bad = [1, 2] \times [1, 2]$. In order to determine $C_{\bar{q}_o}$, refer to the left plot of Figure 2.2, in which we depict the sets $\text{Pre}(q_1, Bad)$ and $\text{Pre}(q_2, Bad)$. Any point $x_o \notin \text{Pre}(q_1, Bad) \cup \text{Pre}(q_2, Bad)$ admits a control that keeps x_o outside Bad for every initial mode. This is due to the fact that the mode of H does not switch and hence a continuous trajectory starting at x_o will follow either of the two directions depicted, none

of which takes the flow inside *Bad*. Hence, we have that $C_{\bar{q}_o} = \text{Pre}(q_1, Bad) \cup \text{Pre}(q_2, Bad)$. By contrast, we have that $\hat{C}_{\bar{q}_o} = \text{Pre}(\bar{q}_o, Bad)$, which includes point x_o in Figure 2.2 as this can be taken to *Bad* by, for example, first flowing under q_1 and then under q_2. Hence, in this case we have that $\hat{C}_{\bar{q}_o}$ is strictly larger than $C_{\bar{q}_o}$.

If we instead had that $\text{Pre}(\bar{q}_o, Bad) = \text{Pre}(q_1, Bad) \cup \text{Pre}(q_2, Bad)$, we would also have that $\hat{C}_{\bar{q}_o} = C_{\bar{q}_o}$. In order to illustrate how we can obtain this equality, we modify system (2.13) to

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 2 \\ 1 \end{pmatrix} u + \begin{pmatrix} 1 \\ 1 \end{pmatrix} d, \ d \in [0,1], \ \text{when } q = q_1 \\ \dot{x} &= \begin{pmatrix} 1 \\ 2 \end{pmatrix} u + \begin{pmatrix} 1 \\ 1 \end{pmatrix} d, \ d \in [0,1], \ \text{when } q = q_2. \end{aligned} \tag{2.14}$$

In this case, the sets $\text{Pre}(q_1, Bad)$ and $\text{Pre}(q_2, Bad)$ are larger than before and are depicted in the right side plot of Figure 2.2. One can check that in this case we still have that $C_{\bar{q}_o} = \text{Pre}(q_1, Bad) \cup \text{Pre}(q_2, Bad)$ and that $\hat{C}_{\bar{q}_o} = \text{Pre}(\bar{q}_o, Bad)$. But, as opposed to before, we also have that $\text{Pre}(\bar{q}_o, Bad) = \text{Pre}(q_1, Bad) \cup \text{Pre}(q_2, Bad)$ so that the two capture sets are the same, that is, $\hat{C}_{\bar{q}_o} = C_{\bar{q}_o}$.

This example illustrates an instance of a system in which $C_{\bar{q}} \neq \hat{C}_{\bar{q}}$ due to $\text{Pre}(\bar{q}, Bad)$ not being equal to $\bigcup_{q_i \in \bar{q}} \text{Pre}(q_i, Bad)$. It also illustrates how requiring that $\text{Pre}(\bar{q}, Bad) \subseteq \bigcup_{q_i \in \bar{q}} \text{Pre}(q_i, Bad)$ (note that $\bigcup_{q_i \in \bar{q}} \text{Pre}(q_i, Bad) \supseteq \text{Pre}(\bar{q}, Bad)$ derives from the definition of Pre) is sufficient to have $C_{\bar{q}} = \hat{C}_{\bar{q}}$. We thus pose the following assumption.

Assumption II.51. For all $\bar{q} \in \hat{Q}$ we have that $\text{Pre}(\bar{q}, Bad) \subseteq \bigcup_{q_i \in \bar{q}} \text{Pre}(q_i, Bad)$.

This assumption requires that if an initial state x_o is taken to *Bad* by an arbitrary sequence of modes in $\bar{q}$, then there is a disturbance signal for which it is also taken to *Bad* by at least one mode $q_i \in \bar{q}$. We provide at the end of this section classes of systems for which this assumption is satisfied.

Since by Lemma II.47, $\text{Pre}(q_i, Bad) \subseteq \hat{C}_{\bar{q}}$ for all $q_i \in \bar{q}$, in order to obtain equivalence, we should at least have that $\text{Pre}(q_i, Bad)$ is also a subset of $C_{\bar{q}}$, which is not the case in general. In fact, an element x_o is in $\text{Pre}(q_i, Bad)$ if and only if there is no feedback map $\pi'(x)$ that prevents the flow starting from this element to end-up in *Bad*. Nevertheless, for such an element x_o there could still be a feedback map $\pi(\bar{q}(\boldsymbol{\eta}(t)), x)$ that prevents the flow originating from it to enter *Bad*. Hence, x_o may not be in $C_{\bar{q}}$. However, if $x(t) = \phi_x(t, (x_o, q_i), \mathbf{u}, \mathbf{d}, \epsilon)$ implies that $\bar{q}(\boldsymbol{\eta}(t))$ is equal to a constant for all $t > 0$, then the map $\pi(\bar{q}(\boldsymbol{\eta}(t)), x)$ that prevents the flow from entering *Bad* becomes a simple feedback map $\pi'(x)$. In this case, if x_o is in $\text{Pre}(q_i, Bad)$, it must also be in $C_{\bar{q}}$. The next assumption and proposition provide conditions for when this is the case.

Definition II.52. A mode $q_i \in Q$ is called *weakly distinguishable* provided

(i) there is a set of modes $I_{q_i} \subseteq Q$ such that $f(x, q_i, u, D) \subseteq f(x, q, u, D)$ for all $q \in I_{q_i}$ and for all $(x, u) \in X \times U$;

(ii) for all $(x, u) \in X \times U$ there is $d \in D$ such that $f(x, q_i, u, d) \notin f(x, q, u, D)$ for all $q \notin I_{q_i}$.

The set I_{q_i} is called the *indistinguishable set* for q_i.

Note that in the case in which the indistinguishable set for q_i is q_i itself, the mode q_i is distinguishable from any other mode, that is, for all (x, u) there is d such that $f(x, q_i, u, d) \notin f(x, q_j, u, D)$ for all $q_j \neq q_i$. Weak distinguishability allows for q_i to generate the same vector fields as those generated by the modes in the set I_{q_i}.

Assumption II.53. System H is such that all modes in Q are weakly distinguishable.

Proposition II.54. *Let $q_i \in \bar{q}_o$, and $x(t) = \phi_x(t, (q_i, x_o), \mathbf{u}, \mathbf{d}, \epsilon)$. Then, Assumption II.53 implies that there is $d(0)$ such that $\bar{q}(\boldsymbol{\eta}(t)) = Reach(Reach(\bar{q}_o) \cap I_{q_i})$ for all $t > 0$.*

Proof. Assumption II.53 implies that for all $(x(0), u(0))$, there is a $d(0)$ such that $f(x(0), q_i, u(0), d(0)) = f(x(0), q_j, u(0), \bar{d}(0))$ for some $\bar{d}(0) \in D$ implies that $q_j \in I_{q_i}$. Hence, $\bar{q}(\boldsymbol{\eta}(t))$

can be re-written as

$$\bar{q}(\boldsymbol{\eta}(t)) = \left\{ \begin{array}{r} q \in Q \mid \exists\, q_o \in \bar{q}_o,\ \boldsymbol{\sigma}\ s.t.\ q = \phi_q(t, q_o, \boldsymbol{\sigma}), \\ \phi_q(0, q_o, \boldsymbol{\sigma}) \in I_{q_i},\ \text{and } \exists\, \bar{\mathbf{d}}\ s.t. \\ \dot{x}(\tau) = f(x(\tau), \phi_q(\tau, q_o, \boldsymbol{\sigma}), u(\tau), d(\tau)) \text{ for all } \tau < t \end{array} \right\}.$$

This, in turn, implies that $\bar{q}(\boldsymbol{\eta}(t)) \subseteq \text{Reach}(\text{Reach}(\bar{q}_o) \cap I_{q_i})$ for all $t > 0$.

Let $q^* \in \text{Reach}(\text{Reach}(\bar{q}_o) \cap I_{q_i})$. Then, for all $t > 0$ there are $\boldsymbol{\sigma}$ and $q_o \in \bar{q}_o$ such that $q^* = \phi_q(t, q_o, \boldsymbol{\sigma})$ and $\phi_q(\tau, q_o, \boldsymbol{\sigma}) \in \text{Reach}(\bar{q}_o) \cap I_{q_i}$ for all $\tau < t$. This, in turn, implies that $\phi_q(0, q_o, \boldsymbol{\sigma}) \in I_{q_i}$. Since for all $\mathbf{d}$ we have that $\dot{x}(\tau) = f(x(\tau), q_i, u(\tau), d(\tau)) \in f(x(\tau), q, u(\tau), D)$ for all $q \in I_{q_i}$, there must be a disturbance signal $\mathbf{d}^*$ such that $\dot{x}(\tau) = f(x(\tau), \phi_q(\tau, q_o, \boldsymbol{\sigma}), u(\tau), d^*(\tau))$ for all $\tau < t$. Hence, we also have that $q^* \in \bar{q}(\boldsymbol{\eta}(t))$ for all $t > 0$. □

Lemma II.55. Let Assumption II.53 hold. Then, we have that $\text{Pre}(q_i, \mathit{Bad}) \subseteq C_{\bar{q}}$ for all $q_i \in \bar{q}$.

Proof. Let $x_o \notin C_{\bar{q}}$, then there is a feedback map π such that for all $q \in \bar{q}$, $\boldsymbol{\sigma}$, $\mathbf{d}$, it guarantees that $\phi_x^\pi(t, (q, x_o), \mathbf{d}, \boldsymbol{\sigma}) \notin \mathit{Bad}$ for all $t \geq 0$. This holds in particular for $q = q_i$, $\boldsymbol{\sigma} = \epsilon$ and $\mathbf{d}$ such that $d(0)$ leads to $\bar{q}(\boldsymbol{\eta}(t)) = \text{Reach}(\text{Reach}(\bar{q}) \cap I_{q_i})$ for all $t > 0$, which exists by Proposition II.54. In this case, $\pi(\bar{q}(\boldsymbol{\eta}(t)), x) = \pi(\text{Reach}(\text{Reach}(\bar{q}) \cap I_{q_i}), x) =: \pi'(x)$ is a simple feedback from x for all $t > 0$. Since $x(0^+) = x(0) = x_o$, we thus have that π' is also such that $\phi_x^{\pi'}(t, (q_i, x_o), \mathbf{d}, \epsilon) \notin \mathit{Bad}$ for all $\mathbf{d}$. Hence, $x_o \notin \text{Pre}(q_i, \mathit{Bad})$. □

Theorem II.56. *Under Assumptions II.51 and II.53, Problem II.12 and Problem II.18 are equivalent.*

Proof. Proposition II.23 proves that $C_{\bar{q}} \subseteq \hat{C}_{\bar{q}}$. We next prove the reverse inclusion. Specifically, by Lemma II.47 and Assumption II.51 we have that $\hat{C}_{\bar{q}} \subseteq \bigcup_{q \in \text{Reach}(\bar{q})} \text{Pre}(q, \mathit{Bad})$, in which by Lemma II.55 we have that $\text{Pre}(q, \mathit{Bad}) \subseteq C_{\text{Reach}(\bar{q})}$, in which $C_{\text{Reach}(\bar{q})} = C_{\bar{q}}$ by Proposition II.11. This proves equivalence. □

2.6.1 Systems that satisfy Assumption 1 and Assumption 2

Assumption II.51 can be difficult to check for general hybrid systems. We thus provide two classes of systems for which such an assumption is satisfied and illustrate in the next section how one of these classes well models the application example. We first introduce two intermediate results.

Proposition II.57. *Let $x \in \mathbb{R}^n$, $\theta \in \Theta \subseteq \mathbb{R}^p$ with $(\Theta, \leq)$ a lattice, and consider the system $\dot{x} = \bar{f}(x, \theta)$, in which $\theta \in \cup_{k\in\{1,...,N\}}[\theta_L^k, \theta_U^k]$. Assume that*

(i) the flow of the system $\phi(t, x_o, \circ) : \mathcal{S}(\Theta) \to \mathbb{R}^n$ is a continuous and order preserving map for all $x_o \in \mathbb{R}^n$ and $t \in \mathbb{R}_+$;

(ii) we have that $[\theta_L^k, \theta_U^k] \cap [\theta_L^{k+1}, \theta_U^{k+1}] \neq \emptyset$, $\theta_L^1 \leq \theta_L^k$, and $\theta_U^N \geq \theta_U^k$ for all $k \in \{1, ..., N-1\}$.

Then, for all x_o, $T > 0$, $i \in \{1, ..., n\}$, and $\bar{x_i}$ such that there is $\boldsymbol{\theta}$ with $\theta(t) \in \cup_{k\in\{1,...,N\}}[\theta_L^k, \theta_U^k]$ for $t < T$ and with $\phi_i(T, x_o, \boldsymbol{\theta}) = \bar{x_i}$, there are $k \in \{1, ..., N\}$ and $\boldsymbol{\theta}'$ with $\theta'(t) \in [\theta_L^k, \theta_U^k]$ for $t < T$ such that $\phi_i(T, x_o, \boldsymbol{\theta}') = \bar{x_i}$.

Proof. Let $\bar{x_i} = \phi_i(T, x_o, \boldsymbol{\theta})$ for $\theta(t) \in \cup_{k\in\{1,...,N\}}[\theta_L^k, \theta_U^k]$ for $t < T$. By property (i) and property (ii), we have that $[\phi_i(T, x_o, \theta_L^j), \phi_i(T, x_o, \theta_U^j)] \cap [\phi_i(T, x_o, \theta_L^{j+1}), \phi_i(T, x_o, \theta_U^{j+1})] \neq \emptyset$ for all $j \in \{1, ..., N-1\}$. Hence, it follows that $\bigcup_{k\in\{1,...,N\}}[\phi_i(T, x_o, \theta_L^k), \phi_i(T, x_o, \theta_U^k)] = [\phi_i(T, x_o, \theta_L^1), \phi_i(T, x_o, \theta_U^N)]$. Since $\bar{x_i} \in [\phi_i(T, x_o, \theta_L^1), \phi_i(T, x_o, \theta_U^N)]$, this implies that there is $k \in \{1, ..., N\}$ such that $\bar{x_i} \in [\phi_i(T, x_o, \theta_L^k), \phi_i(T, x_o, \theta_U^k)]$. Since ϕ is a continuous map from the space of input signals to $\mathbb{R}^n$, it maps the connected set $\mathcal{S}([\theta_L^k, \theta_U^k])$ for all k to the connected set $\phi_i(T, x_o, \mathcal{S}([\theta_L^k, \theta_U^k]))$. Since all connected sets in $\mathbb{R}$ are intervals, we have that $\phi_i(T, x_o, \mathcal{S}([\theta_L^k, \theta_U^k])) = [\phi_i(T, x_o, \theta_L^k), \phi_i(T, x_o, \theta_U^k)]$. Hence, $\bar{x_i} \in \phi_i(T, x_o, \mathcal{S}([\theta_L^k, \theta_U^k]))$, which implies that there is $\boldsymbol{\theta}'$ with $\theta'(t) \in [\theta_L^k, \theta_U^k]$ for $t < T$ such that $\phi_i(T, x_o, \boldsymbol{\theta}') = \bar{x_i}$. □

This proposition states that for a system defined on partial orders whose flow preserves the order and whose set of inputs is a connected union of intervals, any point reachable by

a coordinate of the flow through an arbitrary input signal can also be reached by an input signal that takes values in one only of the possible intervals.

Proposition II.58. *Let $x, L^k, U^k \in \mathbb{R}^n$ for $k \in \{1, ..., N\}$ and consider a differential inclusion of the form $\dot{x} \in [L^1, U^1] \cup ... \cup [L^N, U^N]$. Assume that there are $L, U \in \mathbb{R}^n$ such that $[L^1, U^1] \cup ... \cup [L^N, U^N] = [L, U]$. Then, for all $x_o, \overline{x} \in \mathbb{R}^n$ and $T > 0$ such that $x_o + \int_0^T \dot{x}(t)dt = \overline{x}$, there is $k \in \{1, ..., N\}$ such that $x_o + \int_0^T \dot{x}(t)dt = \overline{x}$ with $\dot{x}(t) \in [L^k, U^k]$ for $t < T$.*

Proof. Let $\overline{x} = x_o + \int_0^T \dot{x}(t)dt$ for $\dot{x}(t) \in [L, N]$ for all $t \leq T$. Re-writing this equality component-wise, we have that for all $i \in \{1, ..., n\}$ $\overline{x}_i - x_{0i} = \int_0^T \dot{x}_i(t)dt$ for $\dot{x}_i(t) \in [L_i, U_i]$ for all $t \leq T$. Then, there is $c_i \in [L_i, U_i]$ such that $\int_0^T \dot{x}_i(t)dt = c_i T$ and hence such that $\overline{x}_i - x_{0i} = c_i T$. The constant vector $c := (c_1, ..., c_n)'$ is thus such that $\overline{x} - \overline{x}_o = cT$, in which $c \in [L, U]$. Since $[L, U] = [L^1, U^1] \cup ... \cup [L^N, U^N]$, there is $k \in \{1, ..., N\}$ such that $c \in [L^k, U^k]$. Hence, there is $k \in \{1, ..., N\}$ such that $\overline{x} - \overline{x}_o = \int_0^T \dot{x}(t)dt$ for $\dot{x}(t) \in [L^k, N^k]$ for all $t \leq T$. □

This proposition states that any point that can be reached under a rectangular differential inclusion in the form of a union of "smaller" rectangular differential inclusions can also be reached under at least one of these smaller rectangular differential inclusions.

Proposition II.59. *Let (H, Bad) be in the form of a two-agent game. Assumption 1 is satisfied if for all $\overline{q} \in \hat{Q}$ with $\overline{q} = \{q_1, ..., q_N\}$ either one of the two following properties are satisfied by H^2:*

(i) for all $q_k \in \overline{q}$ there are $L^k, U^k \in \mathbb{R}^n$ such that $\{f^2(x^2, q_k, d) \mid d \in D^2\} = [L^k, U^k]$, there are $L, U \in \mathbb{R}^n$ such that $\{f^2(x^2, \overline{q}, d) \mid d \in D^2\} = [L, U]$, and $[L^1, U^1] \cup ... \cup [L^N, U^N] = [L, U]$;

(ii) for all $q_k \in \overline{q}$ there are $\theta_L^k, \theta_U^k \in \Theta$ with $(\Theta, \leq)$ a lattice and a function $\overline{f} : \mathbb{R}^n \times \Theta \to \mathbb{R}^n$ such that $\{f^2(x^2, q_k, d) \mid d \in D^2\} = \{\overline{f}(x^2, \theta) \mid \theta \in [\theta_L^k, \theta_U^k]\}$ and $\{f^2(x^2, \overline{q}, d) \mid d \in D^2\} = \{\overline{f}(x^2, \theta) \mid \theta \in \cup_{k \in \{1,...,N\}} [\theta_L^k, \theta_U^k]\}$, $\dot{x} = \overline{f}(x, \theta)$ with $\theta \in \cup_{k \in \{1,...,N\}} [\theta_L^k, \theta_U^k]$ satisfies (i) and (ii) of Proposition II.57, and $B^2 = B_1^2 \times \mathbb{R}^n$.

Proof. Let $(x_0^1, x_0^2) \in \text{Pre}(\bar{q}, Bad)$, we show that when either (i) or (ii) is satisfied there is $q_k \in \bar{q}$ such that $(x_0^1, x_0^2) \in \text{Pre}(q_k, Bad)$. We consider first case (i). Then, for all feedback maps π there is a $T > 0$ such that $\phi^{\pi}_{x^1}(T, x_0^1) \in B^1$ and $x_0^2 + \int_0^T \dot{x}^2(t) = x^2(T) \in B^2$ for $\dot{x}^2(t) \in [L, U]$ for all $t < T$. Let $\vec{x^2} := x^2(T)$, then by Proposition II.58 there is $k \in \{1, ..., N\}$ such that $x_0^2 + \int_0^T \dot{x}^2(t)dt = \vec{x^2} \in B^2$ with $\dot{x}(t) \in [L^k, U^k]$ for $t < T$. Hence, $(x_0^1, x_0^2) \in \text{Pre}(q_k, Bad)$.

Consider now case (ii). We have that for all feedback maps π there are $T > 0$ and $\boldsymbol{\theta}$ with $\theta(t) \in \cup_{k\in\{1,...,N\}}[\theta_L^k, \theta_U^k]$ for all $t < T$ such that $\phi^{\pi}_{x^1}(T, x^1) \in B^1$ and $\phi_{x_1^2}(T, x^2, \boldsymbol{\theta}) \in B_1^2$. Let $\vec{x_1^2} := \phi_{x_1^2}(T, x^2, \boldsymbol{\theta})$, then by Proposition II.57 there are also $k \in \{1, ..., N\}$ and $\boldsymbol{\theta'}$ with $\theta'(t) \in [\theta_L^k, \theta_U^k]$ for all $t < T$ such that $\vec{x_1^2} := \phi_{x_1^2}(T, x^2, \boldsymbol{\theta'})$. Hence, $(x^1, x^2) \in \text{Pre}(q_k, Bad)$. □

This proposition states that if (H, Bad) is in the form of a two-agent game and the continuous dynamics of H^2 (the uncontrolled agent) have either the order preserving properties established by the assumptions of Proposition II.57 or can be modeled by a family of differential inclusions according to Proposition II.58, then Assumption II.51 is satisfied. In turn, the assumptions of Propositions II.57 and II.58 are simple to check. Note that modeling the uncontrolled agent by a family of switching differential inclusions is often a practical approach when an accurate dynamical model of such an agent is missing. In this case, rectangular differential inclusions can be effectively employed to approximate the agent dynamics for safety control purposes. Similarly, systems whose dynamics have order preserving properties are found in several application domains, including biological networks [9, 8] and networks of agents evolving on pre-specified paths such as trains on rails [74, 55], aircrafts on their routes [79, 57], and vehicles in their lanes [39, 42].

Assumption II.53 requires that for all values (x, u), the possible vector fields generated by any given mode q_i cannot be all generated by modes that do not belong to the indistinguishable set for q_i. In the case in which $f(x, q_i, u, d)$ is affine in the disturbance d, that is, $f(x, q_i, u, d) = h(x, q_i, u) + g(x, u)d$, in which $h(x, q_i, u)$ can be regarded as the "nominal" dynamics, a sufficient condition for weak distinguishability of mode i is given, for example, when the nominal dynamics of mode q_i are not possible dynamics in any other mode.

This can, in turn, be ensured if $\|h(x,q_i,u)-h(x,q_j,u)\| > \sup_{d\in D}\|g(x,u)d\|$. As an example, consider f in the form of a chain of integrators, that is, $f(x,q_i,u,d)=(x_2,...,x_n,\beta_i+u+d)$. Letting $d\in[-\bar{d},\bar{d}]$ for some $\bar{d}>0$, one can verify that any mode q_i is weakly distinguishable if $|\beta_i-\beta_j|>\bar{d}$ for all $j\neq i$. For the special case in which f is linear, one can obtain the following general sufficient condition for weak distinguishability.

Proposition II.60. *Let $f(x,q_i,u,d)=A_ix+B_iu+M_id$ with $u\in U\subseteq\mathbb{R}^m$ and $d\in D\subseteq\mathbb{R}^p$ for all $q_i\in Q$. Then, mode q_i is weakly distinguishable if ColSpan$\{M_i\}\cap$ ColSpan$\{A_i-A_j \mid B_i-B_j \mid M_j\}=0$ for all $j\neq i$.*

Proof. If ColSpan$\{M_i\}\cap$ ColSpan$\{A_i-A_j \mid B_i-B_j \mid M_j\}=0$ for all $j\neq i$, then for all d,d^*,u,x with $M_id\neq 0$ we have that $M_id\neq(A_i-A_j)x+(B_i-B_j)u+M_jd^*$, which is equivalent to having $M_id+A_ix+B_iu\neq M_jd^*+A_jx+B_ju$. This, in turn, is equivalent to having that there is d such that $f(x,q_i,u,d)\neq f(x,q_j,u,d^*)$ for all x,u,d^*, which implies weak distinguishability. □

Finally, consider the class of systems introduced in Proposition II.49, in which for all $\hat{q}=q_k\in Q$ we have $\theta\in[\theta_L^k,\theta_U^k]$. If for every k we have that $[\theta_L^k,\theta_U^k]\not\subseteq\bigcup_{j\neq k}[\theta_L^j,\theta_U^j]$ and the map $f^2:X\times\Theta\to X$ is strongly order preserving with respect to the second argument, then Assumption II.53 is satisfied. Similarly, consider case (i) of Proposition II.59. If for all k such that $q_k\in Q$ we have that $[L^k,U^k]\not\subseteq\bigcup_{j\neq k}[L^j,U^j]$, then Assumption II.53 is satisfied.

2.7 Application Example: Control Design

Consider the application example described in Section 2.3.2 and depicted in Figure 2.1. Here, we construct an estimator, calculate the mode-dependent capture sets, and determine the feedback map. An estimator $\hat{H}=(\hat{Q},X,U,D,Y,\hat{R},\hat{f})$ is uniquely determined by $\hat{Q},\hat{R}$, and Y. We set $\hat{Q}=\{\hat{q}_1,\hat{q}_2,\hat{q}_3\}$, in which $\hat{q}_1=\{a,b,c\}$, $\hat{q}_2=\{c,b\}$, and $\hat{q}_3=\{b\}$. To determine $\hat{R}$ and Y, consider the estimate $\hat{\beta}(t)=\frac{1}{T}\int_{t-T}^{t}\dot{v}_2(\tau)d\tau,\ t>T$. For each possible

value of $q(t)$, we compute the interval in which $\hat{\beta}(t)$ must lie. Thus, we have to consider three cases: (1) $q(t) = a$; (2) $q(t) = c$; (3) $q(t) = b$.

Case (1): $q(t) = a$. Then, in the interval of time $[t - T, t]$, the mode $q(t)$ can only have been equal to a. Since it is still possible that $\dot{v}_2(t) = 0$ when v_{max} is exceeded, we have that $\dot{v}_2(\tau) = \beta_a + \tilde{d}(\tau)$ with $|\tilde{d}(\tau)| \leq \beta_a$ for $\tau \in [t - T, t]$. This, in turn, leads to having $|\hat{\beta}(t) - \beta_a| \leq \beta_a$.

Case (2): $q(t) = c$. Then, in the interval of time $[t - T, t]$, the mode $q(t)$ can be c for all time or be first equal to a and then be equal to c. In this case, we have that $\dot{v}_2(\tau) = \frac{\beta_a}{2} + \tilde{d}(\tau)$ for some $\tilde{d}(\tau)$ such that $|\tilde{d}(\tau)| \leq \frac{\beta_a}{2} + \bar{d}$. As a consequence, we have that $\hat{\beta}(t) \in [-\bar{d}, \beta_a + \bar{d}]$.

Case (3): $q(t) = b$. Then, in the interval of time $[t - T, t]$, the mode $q(t)$ can be in b for all time, or also in c for some time, or also in a and then c for some time. It is easy to verify that this implies that $\hat{\beta}(t) \in [-|\beta_b| - \bar{d}, \beta_a + \bar{d}]$, that is, $\hat{\beta}(t)$ can be anywhere.

Hence, we have that if $\hat{\beta}(t) \in [-|\beta_b| - \bar{d}, -\bar{d}]$ then necessarily $q(t) = b$. Similarly, if $\hat{\beta}(t) \in [-\bar{d}, 0]$ then, a is not currently possible and thus we must have that $q(t) \in \{c, b\}$. As a consequence, we let $Y = \{y_{cb}, y_b, \epsilon\}$ and define for $t > T$ $y(t) = y_{cb}$ if $\hat{\beta}(t) \in [-\bar{d}, 0]$, $y(t) = y_b$ if $\hat{\beta}(t) \in [-|\beta_b| - \bar{d}, -\bar{d}]$, and $y(t) = \epsilon$ otherwise. Thus, $\hat{R}$ is such that $\hat{R}(\hat{q}_1, y_{cb}) = \hat{q}_2$, $\hat{R}(\hat{q}_1, y_b) = \hat{q}_3$, and $\hat{R}(\hat{q}_2, y_b) = \hat{q}_3$. System $\hat{H}$ is represented in the top left diagram of Figure 2.3. The properties of an estimator are satisfied as when a or $\{a, c\}$ are ruled out, the structure of R guarantees that $q(t)$ cannot take again those values. By Theorem II.39, Algorithm 1 terminates and by Lemma II.47 we have that $\hat{C}_{\hat{q}_1} = \text{Pre}(\hat{q}_1, Bad)$, $\hat{C}_{\hat{q}_2} = \text{Pre}(\hat{q}_2, Bad)$, and $\hat{C}_{\hat{q}_3} = \text{Pre}(\hat{q}_3, Bad)$. Since for all $\hat{q} \in \hat{Q}$, the assumptions of Proposition II.49 are satisfied, we employ such a proposition to determine whether $x \in \text{Pre}(\hat{q}_i, Bad)$ for all $i \in \{1, 2, 3\}$ and to determine the feedback map $\hat{\pi}$. Assumption II.51 is satisfied and Assumption II.53 is also satisfied for $x_4 \in (v_{min}, v_{max})$. Simulation results are shown in panels (a)-(e) of Figure 2.3.

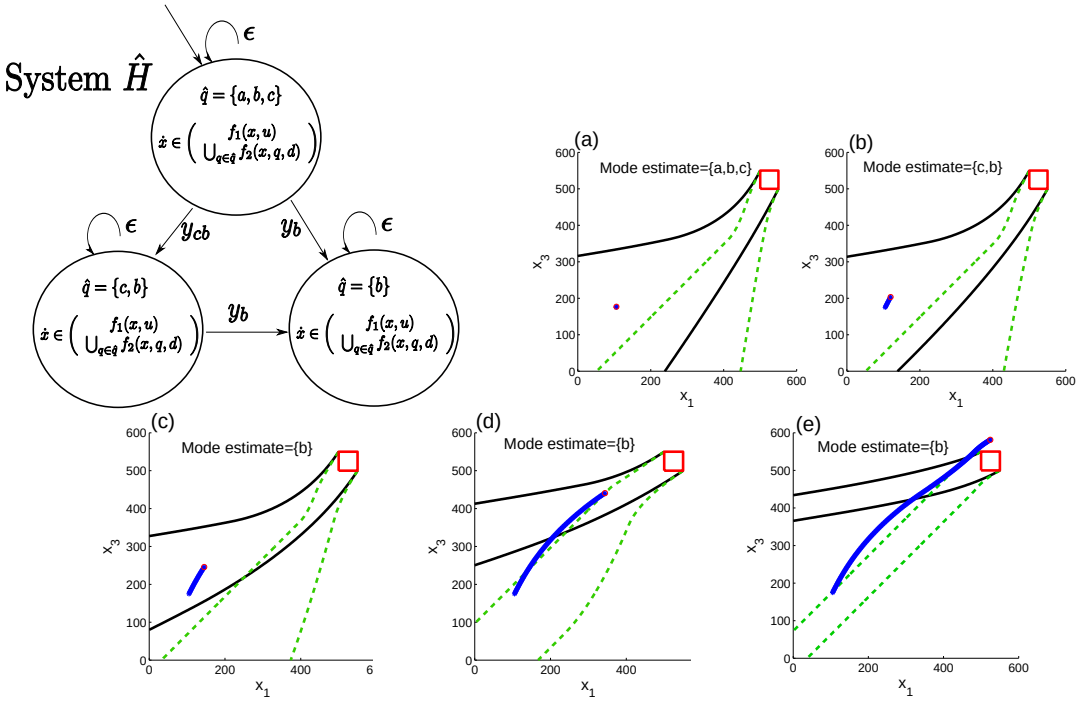


Figure 2.3: (Top Left) Diagram representing $\hat{H}$. In each of the plots (a)–(e), the red box represents $[L_1, U_1] \times [L_2, U_2]$. In the simulation, we have $L_1 = L_2 = 500$, $U_1 = U_2 = 550$, $U = [-1, 1]$, $D = [-0.4, 0.4]$, $\beta_a = 0.6, \beta_c = 0$, and $\beta_b = -0.6$. The black solid lines delimit the slice of the set $\text{Pre}(\hat{q}, Bad)_H$ for the current speeds values (x_2, x_4). Similarly, the green dashed lines delimit the slice of the set $\text{Pre}(\hat{q}, Bad)_L$ for the same current speeds values (x_2, x_4). The intersection of these two slices delimits the slice of the current mode dependent capture set $\hat{C}_{\hat{q}}$ for the same current speeds values (x_2, x_4). The red circle denotes the pair of current longitudinal displacements x_1, x_3, while the blue trace represents the trajectory of this pair. The initial (unknown) driving mode of the human driver is acceleration a and it stays constant for the first 1 second, then from 1 to 3 seconds, the driving mode is coasting c, and finally after 3 seconds the mode is braking b. Plot (a) shows the pair of initial longitudinal displacements. Here, the current mode estimate is $\hat{q} = \{a, b, c\}$ and the current mode dependent capture set is $\hat{C}_{\hat{q}_1}$. Plot (b) shows the mode estimate switching to $\hat{q} = \{c, b\}$ and the current mode dependent capture set shrinks to $\hat{C}_{\hat{q}_2}$. Plot (c) shows the time at which the mode estimate becomes $\hat{q} = \{b\}$, so that the current mode dependent capture set further shrinks to $\hat{C}_{\hat{q}_3}$. Plot (d) shows when the continuous state hits the boundary of $\hat{C}_{\hat{q}_3}$ and thus control is applied. Plot (e) shows the vehicles passing the intersection.

2.8 Application example: Human driving model with two modes

In this case, we assume that there are two modes only, that is, acceleration a and braking b and that in the proximity of the intersection the driver makes the decision of either braking or accelerating. This scenario can be modeled by the HSHM $H = (Q, X, U, D, \Sigma, R, f)$, in

which $Q = \{a, b\}, X = \mathbb{R}^4, U = [u_L, u_H], D = [-\bar{d}, \bar{d}], \Sigma = \{\epsilon\}$, and $R : Q \times \Sigma \to Q$ is represented by the top left diagram of Figure 2.4. The vector field f is piecewise continuous and given by $f(x, q, u, d) = (f_1(x, u), f_2(x, q, d))$, with

$$f_1(x,u) = \begin{cases} (x_2, k_1 u - k_2 x_2^2 - k_3), & \text{if } x_2 \in (v_{min}, v_{max}) \\ (x_2, 0), & \text{if } x_2 \leq v_{min} \text{ and } k_1 u - k_2 x_2^2 - k_3 < 0 \\ & \text{or } x_2 \geq v_{max} \text{ and } k_1 u - k_2 x_2^2 - k_3 > 0, \end{cases} \tag{2.15}$$

$$f_2(x,q,d) = \begin{cases} (x_4, \beta_q + d), & \text{if } x_4 \in (v_{min}, v_{max}) \\ (x_4, 0), & \text{if } x_4 \leq v_{min} \text{ and } \beta_q + d < 0 \\ & \text{or } x_4 \geq v_{max} \text{ and } \beta_q + d > 0. \end{cases} \tag{2.16}$$

We assume that $\beta_b < 0$, $\beta_a > 0$, and $|\beta_q| < \bar{d}$ for $q \in \{a, b\}$. Therefore, we have that $0 \in [\beta_q - \bar{d}, \beta_q + \bar{d}]$ for $q \in \{a, b\}$, which implies a possible confusion between the two modes [35]. The estimator $\hat{H}$, in which we have $2^Q = \{\hat{q}_1, \hat{q}_2, \hat{q}_3\}$ with $\hat{q}_1 = \{a, b\}$, $\hat{q}_2 = \{a\}$, $\hat{q}_3 = \{b\}$, and $\hat{q}(0) = \hat{q}_1$, is uniquely defined once the set Y and map $\hat{R}$ are defined. For this sake, consider the estimate $\hat{\beta}(t) = \frac{1}{T}\int_{t-T}^{t} \dot{v}_2(\tau) d\tau,\ t \geq T$.[1] If the mode is q, then necessarily we have that $|\hat{\beta}(t) - \beta_q| \leq \bar{d}$. In fact, while the mode is q, we have that $\dot{v}_2(\tau) = \beta_q + d(\tau)$ or $\dot{v}_2(\tau) = 0$ for $\tau \in [t - T, t]$. Since $0 \in [\beta_q - \bar{d}, \beta_q + \bar{d}]$ and $|d(t)| \leq \bar{d}$, for all $\tau \in [t - T, t]$ there is $\tilde{d}(\tau)$ with $|\tilde{d}(\tau)| < \bar{d}$ such that $\dot{v}_2(\tau) = \beta_q + \tilde{d}(\tau)$ for $\tau \in [t - T, t]$. Thus $|\hat{\beta}(t) - \beta_q| \leq \frac{1}{T}\int_{t-T}^{t} |\tilde{d}(\tau)| d\tau \leq \bar{d}$. Define thus for $t > T$

$$y(t) = \begin{cases} y_a & \text{if } |\hat{\beta}(t) - \beta_b| > \bar{d} \text{ and } \hat{q}(t^-) = \hat{q}_1 \\ y_b & \text{if } |\hat{\beta}(t) - \beta_a| > \bar{d} \text{ and } \hat{q}(t^-) = \hat{q}_1 \\ \epsilon & \text{otherwise} \end{cases}.$$

Thus, $\hat{R}$ is represented by the bottom left diagram of Figure 2.4. The properties of a mode estimator are satisfied. In fact (i) is satisfied as if q is currently possible (i.e., $|\hat{\beta}(t) - \beta_q| \leq \bar{d}$),

[1] Note that in practice, we will not require measurement of acceleration as we will consider discrete time models where derivative is replaced by time anticipation.

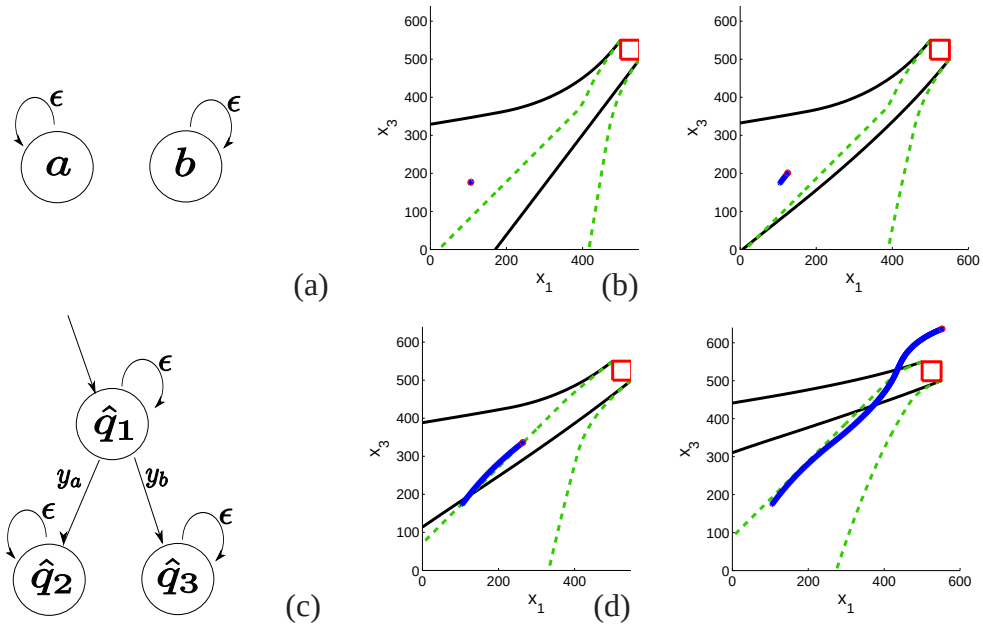


Figure 2.4: **Example 1.** (Left) Diagram representing map R (top) and diagram representing map $\hat{R}$ (bottom). In each of the plots (a)–(d), the red box represents $[L_1, U_1] \times [L_2, U_2]$. Since the sets $\text{Pre}(\hat{q}, Bad)_H$ and $\text{Pre}(\hat{q}, Bad)_L$ are four dimensional, we plot slices of these sets in the (x_1, x_3) position plane corresponding to the current speed values (x_2, x_4). The black solid lines delimit the slice of the set $\text{Pre}(\hat{q}, Bad)_H$ in position plane for the current speeds values (x_2, x_4). Similarly, the green dashed lines delimit the slice of the set $\text{Pre}(\hat{q}, Bad)_L$ in position plane for the same current speeds values (x_2, x_4). The intersection of these two slices delimits the slice of the current mode dependent capture set $\hat{C}_{\hat{q}}$ for the same current speeds values (x_2, x_4). The red circle denotes the current position x_1, x_3, while the blue trace represents the projection in the position plane of the continuous trajectory of H. Plot (a) shows the initial configuration in the position plane. Here, the current mode is $\hat{q} = \{a, b\}$. Plot (b) shows the time at which the mode estimator identifies that H is in braking mode, so that $\hat{q} = \{b\}$ and the current mode-dependent capture set shrinks. In plot (c) the system trajectory hits the boundary of the current mode-dependent capture set and control is applied. Plot (d) shows that both vehicles cross the intersection safely.

it cannot be discarded starting from $\hat{q}_1$. Similarly, once mode q is discarded, since R does not allow transitions, q cannot be possible even when $|\hat{\beta}(t) - \beta_q| \leq \bar{d}$.

By Theorem II.39, Algorithm 1 terminates and by Lemma II.47 we have that $\hat{C}_{\hat{q}_1} = \text{Pre}(\hat{q}_1, Bad)$, $\hat{C}_{\hat{q}_2} = \text{Pre}(\hat{q}_2, Bad)$, $\hat{C}_{\hat{q}_3} = \text{Pre}(\hat{q}_3, Bad)$. Since for all $\hat{q} \in \hat{Q}$, the assumptions of Proposition II.49 are satisfied, we employ such a proposition to determine whether $x \in \text{Pre}(\hat{q}_i, Bad)$ for all $i \in \{1, 2, 3\}$ and to determine the feedback map $\hat{\pi}$. Assumption II.51 is satisfied and Assumption II.53 is also satisfied for $x_4 \in (v_{min}, v_{max})$.

Simulation results. In the simulations, the bad set coordinates are taken as $L_1 = L_2 = 500$, $U_1 = U_2 = 550$, the continuous control input lies in the interval $U = [-1, 1]$, the continuous disturbance input lies in the interval $D = [-0.6, 0.6]$, $\beta_a = 0.4$ and $\beta_b = -0.4$. The simulation results are shown in Figure 2.4. The initial mode estimate is $\{a, b\}$ (Figure 2.4(a)). The driver drives in braking mode but initially choses a disturbance input compatible with both acceleration and braking mode. As a consequence, the estimate of the mode remains $\{a, b\}$ until the driver applies a disturbance input that makes the estimator classify the current mode as braking (Figure 2.4(b)). As a consequence, the current mode-dependent capture set $\hat{C}_{\hat{q}}$ shrinks. In plot (c) of Figure 2.4, the system continuous trajectory hits the boundary of the current mode-dependent capture set. Thus, control $u = u_L$ is applied according to the control law of equation (2.12) in such a way that the vector field points outside the current mode-dependent capture set. Plot (d) of Figure 2.4 shows the vehicles successfully avoiding the collision.

CHAPTER III

Development of a scaled vehcile with Longitudinal dynamics of a HMMWV for Multi-vehicle lab

This chapter presents the development of a scaled vehicle that has dynamical similitude to a High Mobility Multipurpose Wheeled Vehicle (HMMWV) for use in the Multi-vehicle lab for validating the safety control algorithms developed in the previous chapter. The symbols used in the chapter are listed in Section 3.1. In Section 3.2, we describe the drivetrain model that we consider. In Section 3.3, we perform the computation of the π groups and simulate the scaled model to show the match with the full scale model. In Section 3.4, we implement the scaled dynamics on the microprocessor. In Section 3.5, we show experimental results and validate the obtained data against the simulation data of the scaled model.

3.1 NOMENCLATURE

Table 3.1 lists all the symbols used in this chapter.

Table 3.1: Nomenclature.

Symbol	Meaning	Unit
ρ_{air}	Air density	Kg/m^3
g	Acceleration due to gravity	m/s^2
θ_{CS}	Angular displacement of flywheel	$radian$

θ_i	Angular displacement of turbine	*radian*
θ_t	Angular displacement of transmission	*radian*
θ_p	Angular displacement of propeller shaft	*radian*
ρ	Average density of vehicle material	Kg/m^3
τ_{brake}	Brake torque	Nm
B	Damping coefficient of transmission	Kg/s
R_{dcm}	DC motor armature resistance	*ohm*
K_τ	DC motor torque coefficient	*constant*
K_B	DC motor back EMF coefficient	*constant*
I	DC motor current	*ampere*
L_{dcm}	DC motor armature inductance	*henry*
θ	DC motor angular displacement	*radian*
C_D	Drag coefficient	*constant*
τ_w	Drive shaft output torque	Nm
J_e	Flywheel moment of inertia	Kgm^2
i_t	Gear ratio	*ratio*
τ_i	Impeller torque	Nm
U	Longitudinal speed of the vehicle	m/s
τ_d	Output torque produced by final drive	Nm
θ_f	Output angular displacement of final drive	*radian*
θ_w	Output angular displacement of drive shaft	*radian*
A_f	Projected front area of the vehicle	m^2
τ_p	Propeller shaft input torque	Nm
τ_f	Propeller shaft output torque	Nm
V_{PWM}	PWM voltage signal applied to DC motor	*volt*
C_{rr}	Rolling resistance coefficient	*constant*
θ_{road}	Road gradient	*radian*
K	Stiffness of transmission	Kg/s^2
R	Tire radius	m
K_{fc}	Torque converter capacity factor	$Kg^{-0.5}m^{-1}$
T_{ratio}	Torque converter torque ratio	*ratio*
N_{ratio}	Torque converter speed ratio	*ratio*
τ_t	Torque output of the torque converter	Nm
τ_m	Torque produced by DC motor	Nm
τ_e	Torque produced by the engine	Nm
I_t	Transmission inertia	Kgm^2
τ_t	Turbine torque	Nm
m	Vehicle mass	Kg
l	Vehicle track length	m
J_w	Wheel inertia	Kgm^2

3.2 Drivetrain model

The literature presents physics-based models of the longitudinal and powertrain dynamics of the HMMWV, as well as explanations of the assumptions underlying these models

[45, 46, 61, 48]. This work adopts these models, as described briefly below. Figure 3.1 shows the schematic of a vehicle drivetrain. We consider a 4 speed vehicle with automatic transmission and rear wheel drive.

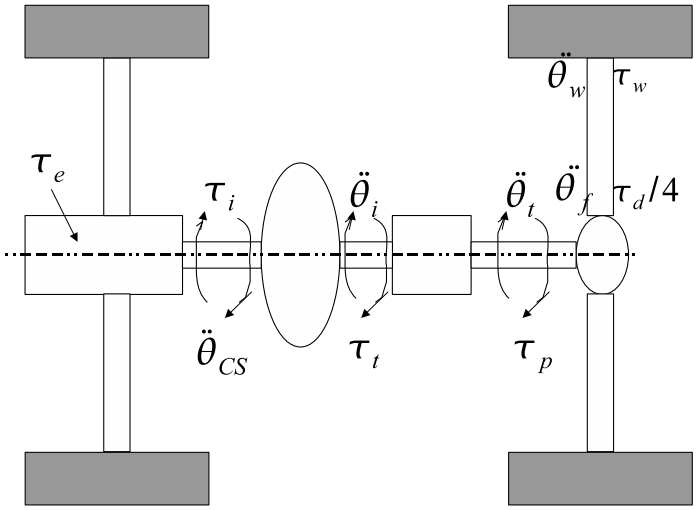


Figure 3.1: Drivetrain.

3.2.1 Engine

The engine produces torque resulting from the combustion process. The engine is modeled as a map (Figure 3.2[45]), which takes throttle command and engine speed as input and calculates torque generated by the engine, τ_e. For the low frequency dynamics we are interested in, a map based engine model can be used. Engine acceleration is calculated from equation (3.1), which takes as input engine torque, τ_e, and load torque from torque converter, τ_i. Flywheel is modeled as inertia. The governing equation for the engine and flywheel is

$$J_e\ddot{\theta}_{CS} = \tau_e - \tau_i, \tag{3.1}$$

where J_e is the engine and flywheel moment of inertia, $\ddot{\theta}_{CS}$ is the acceleration of the flywheel, τ_e is the torque produced by the engine and τ_i is the impeller torque.

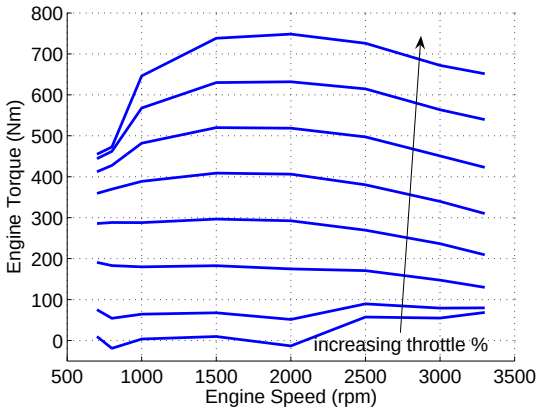


Figure 3.2: Engine map. [45]

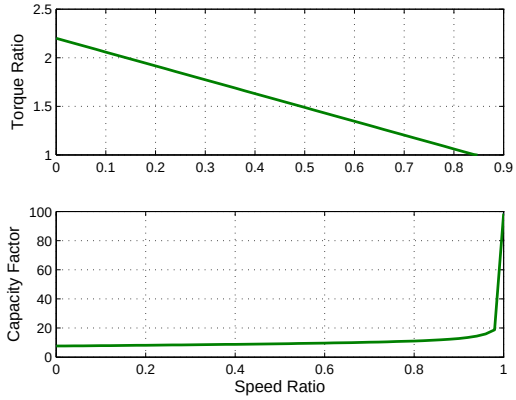


Figure 3.3: Torque converter characteristics [45].

3.2.2 Torque converter

The torque converter model is a tabular relationship between the impeller torque, τ_i, the turbine torque, τ_t, the impeller speed, which is assumed to be equal to $\dot{\theta_{CS}}$, and the turbine speed, $\dot{\theta}_i$. The inputs to this model are speed ratio, $N_{ratio} = \frac{\dot{\theta}_i}{\dot{\theta_{CS}}}$, and impeller speed. The capacity factor, K_{fc}, and the torque ratio, T_{ratio}, are determined by the map shown in Figure 3.3. The impeller torque and turbine torque are calculated as:

$$\tau_i = \frac{\dot{\theta_{CS}}^2}{{K_{fc}}^2}, \tag{3.2}$$

$$\tau_t = T_{ratio}\tau_i. \tag{3.3}$$

3.2.3 Transmission

The transmission is modeled as a variable gear ratio transformer. To derive transmission dynamics, we treat it as a mass-spring-damper system. This system takes as input the torque output of the Torque converter, τ_t, and the gear ratio, i_t. It produces propeller shaft input torque, τ_p. The model is given by

$$\begin{bmatrix} I_t\ddot{\theta}_i \\ \dot{\theta}_i - \dot{\theta}_t i_t \end{bmatrix} = \begin{bmatrix} \frac{-B}{I_t} & -K \\ \frac{1}{I_t} & 0 \end{bmatrix} \begin{bmatrix} I_t\dot{\theta}_i \\ \theta_i - \theta_t i_t \end{bmatrix} + \begin{bmatrix} 1 & B \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \tau_t \\ \dot{\theta}_t i_t \end{bmatrix} \tag{3.4}$$

$$\tau_p = \begin{bmatrix} \frac{B}{I_t} & K \end{bmatrix} \begin{bmatrix} I_t\dot{\theta}_i \\ \theta_i - \theta_t i_t \end{bmatrix} \begin{bmatrix} 0 & B \end{bmatrix} \begin{bmatrix} \tau_t \\ \dot{\theta}_t i_t \end{bmatrix}, \tag{3.5}$$

in which B is the damping coefficient, I_t is the inertia, and K is the stiffness of the transmission.

3.2.4 Shift Logic

Gear shift is modeled as a shift map (Figure 3.4), which takes propeller shaft speed and throttle position commanded by the driver as the input and determines the instantaneous gear ratio as the output [45]. Torque and speed variations during the gear shift are captured by incorporating a blending function into the model. The blending function (Figure 3.5) gives the variation of torque ratio and speed ratio during the gearshift and captures important dynamics observed during a gearshift [45, 61].

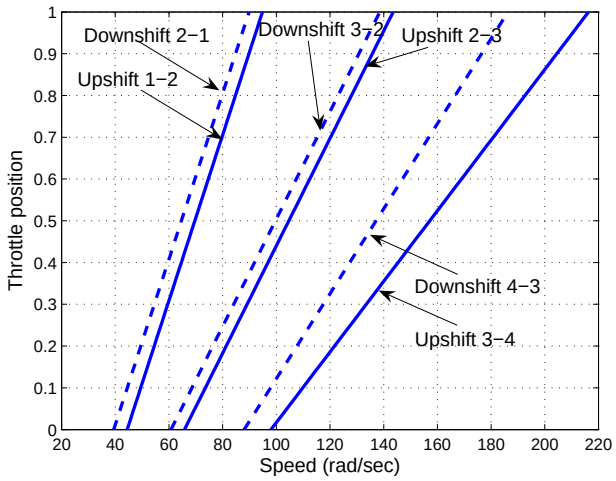


Figure 3.4: Shiftmap (taken from [45]).

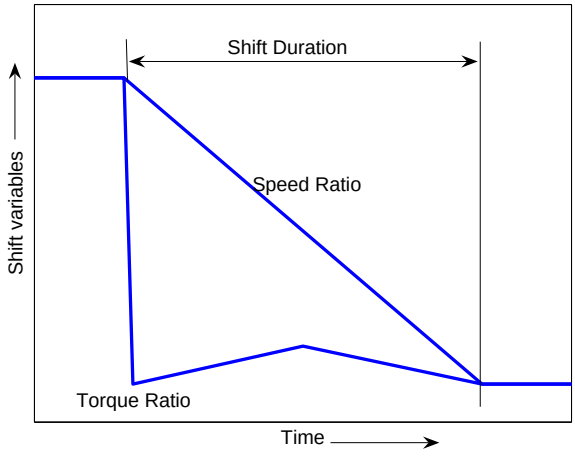


Figure 3.5: Blending function (taken from [45]).

3.2.5 Propeller shaft

The propeller shaft dynamics are taken into account in the transmission model, and thus the propeller shaft input torque, τ_p, and speed, $\dot{\theta}_t$, are equal to the output torque, τ_f, and speed, $\dot{\theta}_p$:

$$\tau_f = \tau_p \tag{3.6}$$

$$\dot{\theta}_t = \dot{\theta}_p. \tag{3.7}$$

3.2.6 Final drive

Final drive is modeled as a ratio, i_f, which reduces the input speed, $\dot{\theta}_p$, and increases the input torque, τ_f, to produce the output speed, $\dot{\theta}_f$, and torque, τ_d, respectively:

$$\tau_d = \tau_f i_f \tag{3.8}$$

$$\theta_p = \theta_f i_f. \tag{3.9}$$

3.2.7 Drive shaft

The drive shaft dynamics are taken into account in the transmission model, and thus the drive shaft input torque, τ_d, and speed, $\dot{\theta}_f$, are equal to the output torque, τ_w, and speed, $\dot{\theta}_w$:

$$\tau_w = \tau_d \tag{3.10}$$

$$\dot{\theta}_w = \dot{\theta}_f. \tag{3.11}$$

3.2.8 Vehicle Model

Point mass vehicle model is considered here, in which we consider only longitudinal vehicle dynamics. We do not address the lateral vehicle dynamics in this work. The longi-

tudinal motion of the vehicle is defined by:

$$(J_w + mR^2)\ddot{\theta}_w = \tau_w - \tau_{brake} - \frac{\rho_{air}}{2} C_D A_f U^2 R - C_{rr} mg - Rmgsin(\theta_{road}), \tag{3.12}$$

where J_w is the wheel inertia, m is the mass of the vehicle, τ_{brake} is the brake torque, U is the longitudinal vehicle velocity, ρ_{air} is the air density, C_D is the drag coefficient, A_f is the projected front area of the vehicle, $C_r r$ is the rolling resistance cofficient, R is the tire radius, and θ_{road} is the road gradient, assumed 0 here.

The system, equations (3.1) to (3.12), is simulated using Simulink. The main components modeled are engine, automatic transmission, gear shift logic, shafts and vehicle. This constitutes a point mass longitudinal dynamics model that does not account for roll and pitch. The model considered serves well the purpose of predicting the behavior of a HMMWV in longitudinal maneuvers in the frequency range relevant for control (including the lowest resonance modes of the driveline), and is simple enough to be programmable on the motioncontroller, given its processing and memory constraints.

3.3 Scaling

To apply Buckingham's π theorem to the system described in equations (3.1) to (3.12), the governing dynamical equations are examined and parameters and variables associated with the system that are used in this study are listed in Table 3.2.

The fundamental quantities (basic units) chosen for the formulation of nondimensional groups (π groups) are M, T, L. Similitude is achieved by grouping the parameters into $(n - m)$ independent nondimensional groups, where n is the number of parameters and m is the number of fundamental quantities. The parameters listed in Table 3.2, which are important to design the scaled vehicle, can be written in terms of fundamental quantities as illustrated in Table 3.3.

Engine	$Throttle, J_e, \theta_{CS}, \tau_e, \tau_i$
Torque Converter	$\tau_t, K_{fc}, T_{ratio}, N_{ratio}$
Transmission	$\theta_i, \theta_t, I_t, \tau_p, i_t, B, K$
Propeller shaft, Final drive and Drive shaft	$\theta_p, \theta_f, \theta_w, \tau_f, \tau_d, \tau_w, i_f$
Vehicle model	$Brake, R, m, \tau_{brake}, \tau_w$ $U, l, J_w, \rho_{air}, C_D, A_f, C_{rr}$

Table 3.2: Parameters and variables associated with the vehicle.

$N_{ratio}, T_{ratio}, i_t, i_f$	$ratio$	$[M^0L^0T^0]$
$\theta_{CS}, \theta_i, \theta_t, \theta_p, \theta_f, \theta_w$	$radians$	$[M^0L^0T^0]$
$Throttle, Brake$	$percentage$	$[M^0L^0T^0]$
$\tau_e, \tau_i, \tau_t, \tau_p, \tau_f, \tau_d, \tau_w$ τ_{brake}	Nm	$[M^1L^2T^{-2}]$
J_e, I_t, J_w	kgm^2	$[M^1L^2T^0]$
m	kg	$[M^1L^0T^0]$
R, l	m	$[M^0L^1T^0]$
U	m/s	$[M^1L^0T^{-1}]$
ρ, ρ_{air}	kgm^{-3}	$[M^1L^{-3}T^0]$
K_{fc}	$kg^{-0.5}s/m$	$[M^{-0.5}L^{-1}T^1]$
A_f	m^2	$[M^0L^2T^0]$
B	kg^1/s	$[M^1L^0T^{-1}]$
C_D, C_{rr}	$dimensionless$	$[M^0L^0T^0]$

Table 3.3: Parameters associated with the vehicle in terms of fundamental quantities.

All of the unitless parameters, such as angles and percentages form their own π group. Now, we have 3 fundamental dimensions and 34 parameters (Table 3.3). Out of these, if we choose m, U and l as repeating parameters (parameters that can appear in some or all of the π groups), the remaining parameters will form 31 dimensionless π groups.

A list of all the π groups is given in Table 3.4.

π group	Description
$\pi_1 = Throttle, \pi_2 = Brake$	Non-dimensional driver inputs
$\pi_3 = i_t, \pi_4 = i_f, \pi_5 = N_{ratio}, \pi_6 = T_{ratio}$	Non-dimensional transmission and final drive gear ratio
$\pi_7 = \frac{J_e}{mL^2}, \pi_8 = \frac{I_t}{mL^2}, \pi_9 = \frac{J_w}{mL^2}$	Non-dimensional engine, transmission and wheel inertia
$\pi_{10} = \theta_{CS}, \pi_{11} = \theta_i, \pi_{12} = \theta_t, \pi_{13} = \theta_p, \pi_{14} = \theta_f, \pi_{15} = \theta_w$	Non-dimensional angular displacements
$\pi_{16} = \frac{\tau_e}{mU^2}, \pi_{17} = \frac{\tau_i}{mU^2}, \pi_{18} = \frac{\tau_t}{mU^2}, \pi_{19} = \frac{\tau_p}{mU^2}, \pi_{20} = \frac{\tau_f}{mU^2}, \pi_{21} = \frac{\tau_d}{mU^2}, \pi_{22} = \frac{\tau_w}{mU^2}, \pi_{23} = \frac{\tau_{brake}}{mU^2}$	Non-dimensional torques
$\pi_{24} = K_{fc}\sqrt{mU^2}$	Non-dimensional capacity factor
$\pi_{25} = \frac{R}{l}$	Non-dimensional wheel radius
$\pi_{26} = \frac{\rho L^3}{m}, \pi_{27} = \frac{\rho_{air} L^3}{m}$	Non-dimensional vehicle and air density
$\pi_{28} = \frac{A_f}{l^2}$	Non-dimensional projected front area of vehicle
$\pi_{29} = \frac{BU}{ml}$	Non-dimensional damping
$\pi_{30} = C_D, \pi_{31} = C_{rr}$	Non-dimensional drag and rolling resistance coefficient

Table 3.4: π groups associated with the system.

3.3.1 Design of the scaled vehicle

It follows from Buckingham π theorem that if two dynamical systems are described by the same differential equations, then the solution to these differential equations will be the

scale-invariant if the π groups are the same. To design the scaled vehicle, we thus start with analyzing the π groups given in Table 3.4. For the scaled vehicle to be dynamically similar to the actual vehicle, the value of these π groups should be the same for both systems. Based on this concept, we can derive the parameter values of the scaled vehicle or of the actual vehicle.

3.3.1.1 Calculation of parameter values for the scaled vehicle

The track length of the full scale vehicle and of the scaled vehicle are fixed. The tire size of the scaled vehicle is calculated by equating the π group corresponding to tire size of the scaled vehicle to the full size vehicle as follows (Table 3.4, row 5):

$$\left(\frac{R}{l}\right)_{Full} = \left(\frac{R}{l}\right)_{Scaled}$$
$$\left(\frac{0.4412}{3.302}\right) = \left(\frac{R}{0.257}\right)$$
$$D_{scaled} = 0.0343m. \tag{3.13}$$

The actual tire diameter of the RC car is 0.033m. This error is compensated by using a feedforward control loop as discussed in detail in Section 3.4.

3.3.1.2 Mass of the full scale vehicle

The mass of the scaled vehicle is 3.15 Kg. Using the π groups corresponding to the vehicle density (Table 3.4-row 6), we calculate the mass of actual vehicle as follows:

$$\left(\frac{\rho l^3}{m}\right)_{Scaled} = \left(\frac{\rho l^3}{m}\right)_{Actual}$$
$$(\rho)_{Scaled} = (\rho)_{Actual}\,(assumed)$$
$$\left(\frac{l^3}{m}\right)_{Scaled} = \left(\frac{l^3}{m}\right)_{Actual}$$
$$\left(\frac{3.302^3}{m}\right)_{Actual} = \left(\frac{0.257^3}{3.15}\right)_{Scaled}$$
$$m_{Actual} = 6681Kg. \tag{3.14}$$

Note that the Gross Vehicle Weight of the full scale vehicle is 5112 kg [45]. It is assumed that the full scale vehicle is carrying a payload of 1569 kg.

3.3.1.3 Velocity of the scaled vehicle

To find the ratio of velocity that the scaled vehicle should maintain with respect to the full scale vehicle in response to the same input, first observe that time is not being scaled. Thus, we can consider Ut/l to form another π group. Now we can write:

$$\left(\frac{Ut}{l}\right)_{Scaled} = \left(\frac{Ut}{l}\right)_{Actual}$$
$$\frac{U_{full}}{U_{Scale}} = 3.302/0.257 = 12.84. \tag{3.15}$$

Thus, the full scale vehicle velocity should be 12.84 times the velocity of the scaled vehicle when the same maneuver is performed on both systems.

3.3.1.4 Moment of inertia of the scaled engine

The moment of inertia of the engine in the full scale HMMWV is 0.5 $kg\ m^2$ [46]. We proceed as follows to scale the moment of inertia:

$$\left(\frac{J_e}{ml^2}\right)_{Scaled} = \left(\frac{J_e}{ml^2}\right)_{Scaled}$$
$$\left(\frac{J_e}{3.15(0.257^2)}\right)_{Scaled} = \left(\frac{.5}{6681(3.302^2)}\right)_{Actual}$$
$$J_{eScaled} = 1.51 * 10^{-6}, \tag{3.16}$$

which gives the moment of inertia of the scaled engine.

3.3.1.5 Engine torque scaling

To determine the ratio of torque produced by the engine of the scaled vehicle to the full scale vehicle, the π groups corresponding to engine torque are equated (Table 3.4 - row 2). We thus obtain the following relation:

$$\left(\frac{\tau_e}{mU^2}\right)_{Scaled} = \left(\frac{\tau_e}{mU^2}\right)_{Actual}$$
$$(\tau_e)_{Scaled} = 2.855 * 10^{-6}\,(\tau_e)_{Actual} \tag{3.17}$$

This torque scaling is used to scale the engine torque map (Figure 3.2).

It is difficult to measure parameters such as $I_t, J_w, A_f, B, C_D, C_{rr}$ for the scaled vehicle. The difference in these parameters is compensated by simulating it in the HIL setup and will be discussed in Section 3.4.1.

3.3.2 Validation of scaled model

The validation of the derived π groups and scaled vehicle design based on these groups is done in two steps. A simulation of scaled model is carried out as a first step and is

discussed in this section. This is followed by experimental tests with the scaled vehicle hardware. These are discussed in Section 3.5.

This section presents the simulation results of the scaled vehicle compared to the full scale vehicle. All parameters of the scaled model are derived as illustrated in Section 3.3.1. The full scale and scaled vehicle simulation are carried out for the same input commands. It is found that the longitudinal velocity of the full scale vehicle is 12.84 time the velocity of the scaled vehicle (Figure 3.6). The gearshift in both full scale and scaled vehicle occur at the same time as shown in Figure 3.7.

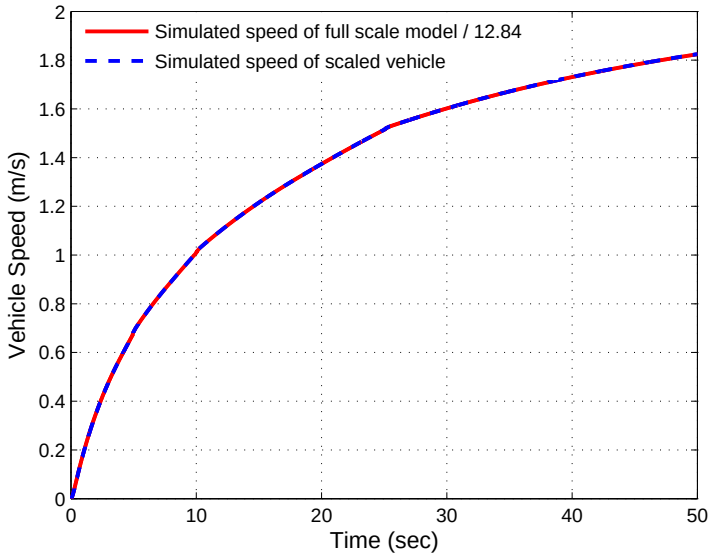


Figure 3.6: Scaled vehicle velocity vs full scale vehicle velocity.

3.4 Implementation on scaled RC car

A Tamiya scaled RC car chassis [2] is used as the hardware platform to implement the scaled dynamics and validate the simulation results. The vehicle originally had four wheel drive. A quadrature encoder is used to sense the tire speed. The front axle of the vehicle is modified to fit the encoder and it no longer drives the vehicle. Thus the vehicle has rear wheel drive and front wheel steering.

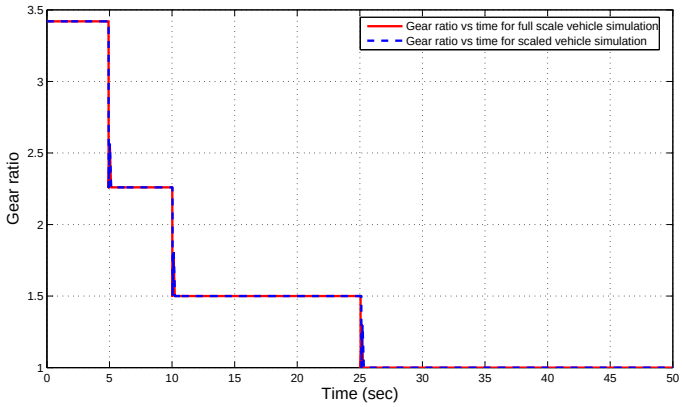


Figure 3.7: Scaled vehicle gear ratio vs full scale vehicle gear ratio.

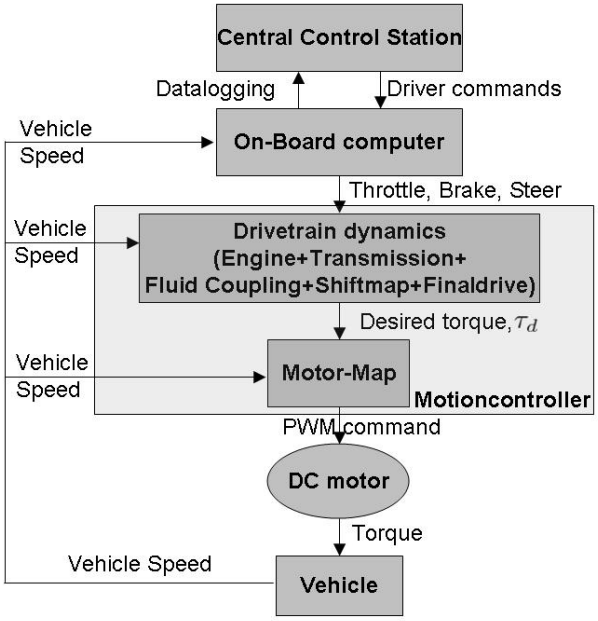


Figure 3.8: Scaled vehicle command flow.

Figure 3.8 shows the system architecture. In the present configuration, a human driver issues throttle, brake and steer commands through a central control station. These commands are transmitted to the on-board computer (Mini ITX) through a wireless connection.

These commands act as input to the driveline dynamics which are programmed on the motioncontroller. The driveline dynamics consist of the engine, fluid coupling, transmission and gearshift logic, as explained in Section 3.2. Shift logic is programed in the form of a shift map. The output of this program is the drive torque, τ_d.

The drive torque is the torque that should be applied to the wheels. Since we have a DC motor, it is difficult to measure or control such a torque because of the absence of current measurement. To overcome this problem, a set of experiments were performed to identify the relationship between drive torque and motor voltage for any given wheel speed. This is discussed in the next section. Vehicle speed is measured using an optical encoder and is used for calculations in drivetrain and motor map blocks. This speed is sent to on-board computer and is transmitted to central control station through a wireless connection, where it is recorded.

3.4.1 DC motor system identification

The dynamics of the electromechanical system comprising a car being run by the DC motor includes three parts: (1) A dynamic mechanical subsystem, which is the scaled vehicle, (2) a dynamic electrical subsystem, which includes all of the motor's electrical effects, and (3) a static relationship which represents the conversion of electrical quantities into mechanical torque. Assuming very high torsional stiffness of drivetrain components transmitting torque, the mechanical subsystem dynamics of the vehicle run by a permanent magnet brush DC motor are assumed to be of the form

$$M\ddot{\theta} + B\dot{\theta} = \tau_m, \tag{3.18}$$

in which

$$M = J_w + mR^2,$$

and

$$\tau_m = K_\tau I.$$

The motor armature current, $I(t)$, is given by the electrical subsystem dynamics for the permanent magnet brush DC motor, which is assumed to be of the form:

$$L_{dcm}\dot{I} = V_{PWM} - R_{dcm}I - K_B\dot{\theta}. \tag{3.19}$$

in which, J_w is the wheel moment of inertia, m is the vehicle mass, B is the damping in the drivetrain, K is the stiffness in drivetrain, $\theta(t)$ is the angular motor position, K_τ is the coefficient which characterizes the electromechanical conversion of armature current to torque, L_{dcm} is the armature inductance, R_{dcm} is the armature resistance, K_B is the back-emf coefficient (which is equal to K_τ), and V_{PWM} is the Pulse Width Modulation voltage signal supplied to the DC motor. For the above model, the states θ and $\dot{\theta}$ are easy to measure while I is difficult to measure. Because of the inability to measure motor current I, the

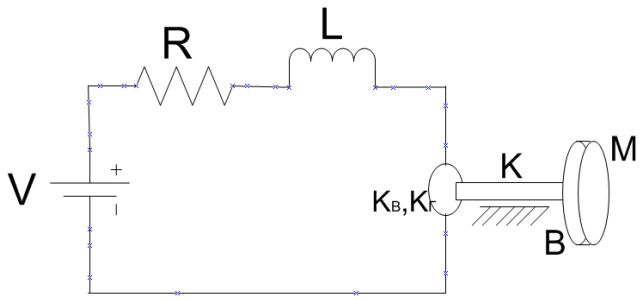


Figure 3.9: Electromechanical system.

control of the torque produced by the motor is hard. This difficulty is overcome by noticing that in this mechatronic systems, the time constant of the electrical subsystem is faster than the mechanical subsystem. This means that we can assume the current and voltage to be statically related. This allows us to perform system identification of the electromechanical

system to obtain V_{PWM} vs speed ($\dot{\theta}$) vs total torque (τ_{total}) map. Assuming L_{dcm} to be negligible, we can write equation (3.19) as:

$$I = \frac{V_{PWM}}{R_{dcm}} - \frac{K_B\dot{\theta}}{R_{dcm}}. \tag{3.20}$$

From equation (3.18), we have

$$M\ddot{\theta} + \left(B + K_\tau \frac{K_B}{R_{dcm}}\right)\dot{\theta} - K_\tau \frac{V_{PWM}}{R_{dcm}} = 0. \tag{3.21}$$

Figure 3.10: Vehicle speed vs time.

$M\ddot{\theta}$ is the torque that accelerates the vehicle. We call it the total torque, τ_{total}. It is equal to the torque produced by the motor minus the torque lost in damping of the scaled vehicle. Our objective is to be able to control the torque generated by the DC motor, τ_m, and make it proportional at all time to the torque generated by the engine, τ_e, that is programmed on the motioncontroller. As stated earlier, this is a hard problem in the absence of current measurement. Though τ_m cannot be measured, τ_{total} can be determined experimentally. We shift the problem from trying to make τ_e proportional to τ_m to making $\tau_d = \tau_{total}$.

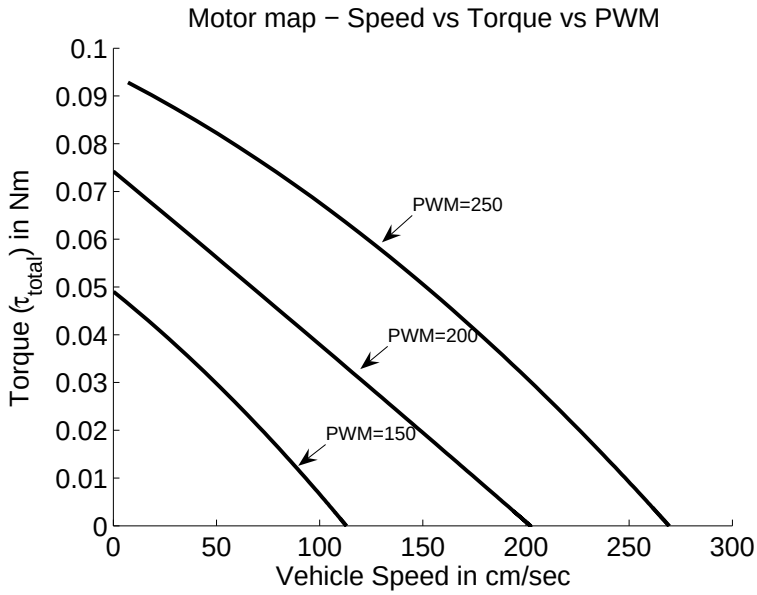


Figure 3.11: Motor map.

The torque τ_{total} corresponds to the torque generated by the drivetrain that accelerates the vehicle, i.e τ_d. In order to solve this problem, we identify the coefficients of $\dot{\theta}$ and of V_{PWM} in equation(3.21) by running experiments.

Experiments are performed, in which a constant PWM signal (V_{PWM}) is applied to the DC motor and the vehicle response (vehicle velocity vs time) is recorded (Figure 3.10 shows an example). The data is logged at a frequency of 7.7 Hz. Vehicle acceleration is obtained by differentiating vehicle velocity. As vehicle velocity is noisy, a polynomial fit of the third order to the vehicle velocity vs time curve is used before differentiation to calculate the acceleration a. The value of $\ddot{\theta}$ is calculated from this acceleration as follows:

$$\ddot{\theta}_{wheel} = \frac{a}{R} \tag{3.22}$$

$$\ddot{\theta} = 7.21\ddot{\theta}_w \tag{3.23}$$

where, $\ddot{\theta}_w$ is the wheel angular acceleration and 7.21 is the gear ratio of the scaled model. Using equation (3.21), τ_{total} for a given vehicle velocity and PWM signal can be calculated.

This data can be plotted as τ_{total} vs vehicle velocity at a constant PWM. A number of such experiments are performed, for a particular PWM, to check the repeatability of the experiment. PWM signals are chosen to cover the whole range of operation of the DC motor. A motor map is obtained by plotting τ_{total} vs vehicle velocity at a constant PWM value as shown in Figure 3.11. In calculating M, we consider J_w negligible when compared to mR^2.

To use this map on a running vehicle at any instant of time, the drivetrain block (Figure 3.8) calculates the torque that is to be applied to the scaled vehicle. The PWM that has to be supplied to the DC motor to generate this torque, given the velocity of the scaled vehicle and required τ_d is given by:

$$V_{PWM} = k_1\tau_d + k_2 v, \tag{3.24}$$

where v is the vehicle velocity, k_1 and k_2 are constants that are determined by performing system identification on motor map. The value calculated were $k_1 = 2503$ and $k_2 = 1$. However, during experimentation, we found that the relation that produced the best response was

$$V_{PWM} = 70 + 2800\tau_d + 0.72v.$$

The experiments were conducted as a constant throttle performance for 30 %, 40 % and 50 % throttle. The values of k_1 and k_2 are found to be different for these 3 cases, though not significantly. The values of k_1 and k_2 is thus kept constant corresponding to 30 % throttle for the rest of this study.

3.4.2 Description of the scaled vehicle hardware

In this section, specifications of the scaled vehicle hardware are provided. The scaled vehicle specifications are given in Table 3.5.

Overall length	375mm, Width: 185mm, Wheelbase: 257mm, Weight: 890g
Transmission type	Longitudinally mounted motor with shaft-driven 4WD
Gear ratio	7.21:1
Suspension	Double wishbone (front and rear)
Damper	CVA oil dampers (front and rear
Tire Width	27mm, Diameter: 67mm
Tread Pattern	Racing slick

Table 3.5: Basic specifications of the scaled model.

3.4.2.1 Motor specifications

The model uses GT-Tuned-Motor (25T), which is a replaceable brush standard type electric motor. The motor specifications are in Table 3.6.

Turns	25
Voltage	7.2 Volt
Speed	Unloaded RPM 19000
Maximum Torque	500 gf/cm (Gram Force per Centimeter) @ 7.2V

Table 3.6: Basic specifications of the DC motor.

3.4.2.2 Electronic architecture

Each vehicle (Figure 3.12) is equipped with a motioncontroller (BrainStem Module) implementing the scaled driveline dynamics of a HMMWV (Section 3.2). The on board computer (running Linux, Fedora core) communicates with the motioncontroller by means of a serial connection. The computer is also equipped with wireless communication capability. The computer handles the high level control functions by commanding steering braking and throttle to the motioncontroller. The motioncontroller offers 2 channels of high-resolution motion control. These channels offer flexible PWM or PID control of motors with various types of feedback including encoders, quadrature encoders, analog input,

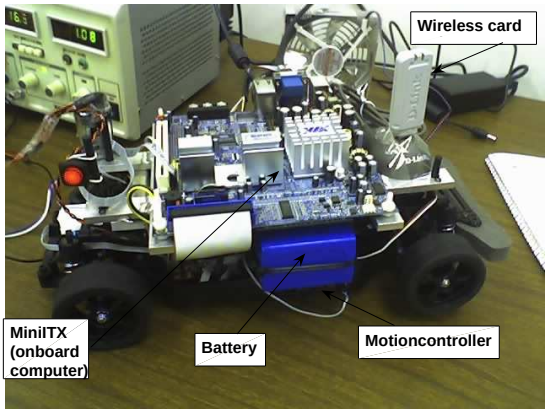


Figure 3.12: Scaled vehicle.

and Back-EMF speed control. The motioncontroller can handle a variety of motion control needs. This processor can run concurrent TEA programs, reflexes, and handle slave commands from a host PC, all simultaneously. The motioncontroller board can accept two 3 Amp H-Bridge with back EMF control. Access to the module is through a Console Application and C language. For more information, refer to [4].

3.4.3 Description of the scaled vehicle software

The high level control algorithms are programmed on the onboard computer. A data-logging module is programed on the on board computer which can read vehicle speed from the motioncontroller at a frequency of 10 samples/sec. The drivetrain components, including Engine, Fluid coupling, Transmission, Gear shift logic and Final drive are programmed on the motioncontroller. The motioncontroller issues control signals to the steering servo and controls the PWM signal to the DC motor. The programming on the motioncontroller is performed in TEA language, described in the next section.

3.4.3.1 Brainstem TEA language

The Tiny Embedded Application (TEA) [3] language is a subset of the C programming language. TEA has integer math operations, simple looping constructs, conditional statements, and parameterized subroutines. It is ideally suited for simple control loops, sequencing behaviors in robotics, and other tasks that can be offloaded from the main controller of a complex system. TEA is precompiled, enabling conditional compilation, macros, and inclusion of other files. Programs are typically very small and have no memory allocation, structures, or objects. All variables are stack-based; the stack can be very small on some environments so minimal recursion is possible. Program compilation is done through the Console application. The compiler translates the TEA language file into the virtual machine's specific instructions (opcodes).

3.5 Experiments

A number of experiments were performed to ascertain the behavior of the scaled vehicle and its dynamic similitude to a HMMWV. The following sections discuss the experimental setup and results.

3.5.1 Experimental Setup

The scaled vehicle can take throttle, steering and braking commands from the human driver at a central control station. The driving maneuver that is considered for verifying the longitudinal response of the scaled vehicle vis a vis an actual vehicle is constant throttle performance test. In this test, a constant throttle input is given to the scaled vehicle and the resulting velocity and gear shift response are logged. This test is repeated for several throttle values ($30\%, 40\%, 50\%$). The 4th floor corridor in the Electrical Engineering and Computer Science building is utilized for testing because the testbed dimension (6.6 meter by 5.6 meter) is not sufficient to run the vehicle for significant length and duration. All

communication to and from the scaled vehicle is through a local wireless network from the lab. The maximum length that is avialable with wireless coverage is limited to 42.67 meters. This also restricts the length of a test run.

Throttle inputs more than 50% cannot be applied because of the difficulty in controlling the vehicle at high speed while conducting the experiment in the corridor. During the experiments, the voltage supply to the scaled vehicle has to be kept constant at 15.4 volts. Therefore, for this test, we power the vehicle through a power supply method instead of using the on-board batteries. One person carrying the powersupply has to follow the vehicle while it is running. Due to the limited length of the corridor, the steady state speed of the vehicle is never achieved. This issue is not relevant for this research because it is not expected to experience high speed of the scaled vehicle in the testbed, which is only 6.6 by 5.6 meters. Since the purpose of this testbed is to test algorithms for multiagent traffic intersection, we do not expect to attain high velocities or steady state response of the vehicle.

It is observed during the experiments that the scaled vehicle does not start as soon as the throttle command is applied by the driver. There is a delay between the application of throttle command and start of the vehicle. An external excitation is required to set the vehicle into motion. This can be attributed to the friction in the scaled hardware drivetrain.

3.5.2 Experimental Results

Figure 3.13 shows the speed response of the scaled vehicle vis-a-vis simulation. The delay in start of vehicle can be clearly see in Figure 3.13.

The results are presented for a constant input of 30 %, 40 % and 50 % throttle to the vehicle. Figures 3.14, 3.16 and 3.18 show the speed response of the scaled vehicle vis-a-vis simulation. It is seen that the response of the scaled vehicle closely follows the simulated response. The average root mean square (RMS) error in speed for 30 % throttle is 0.0525m/s, 40 % throttle is 0.0809m/s and 50 % throttle is 0.1099m/s. There seem to be

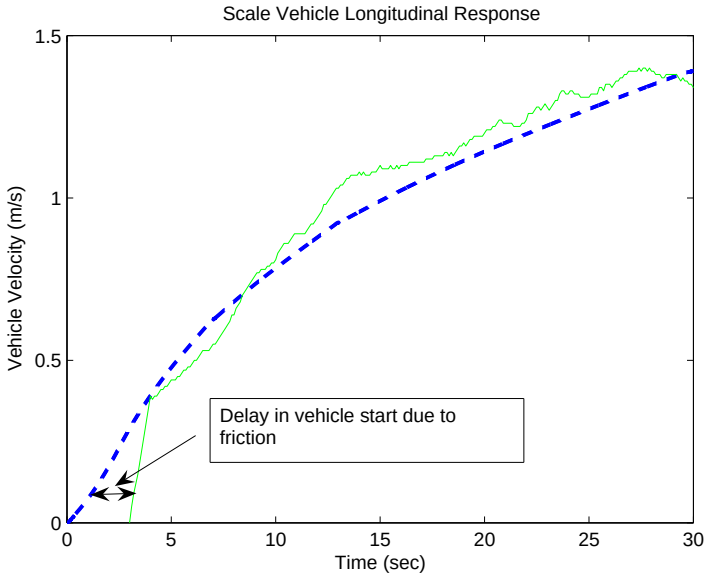


Figure 3.13: Scale Vehicle speed vs time for 40 percent throttle.

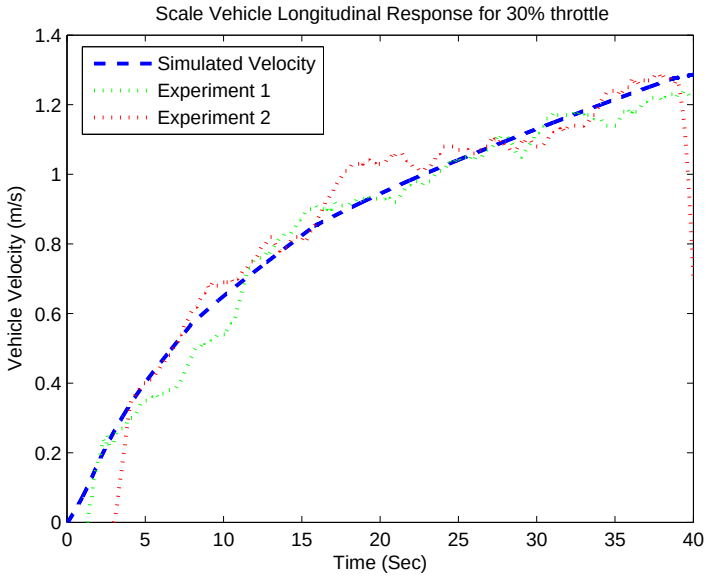


Figure 3.14: Vehicle speed vs time for Scaled vehicle model and Scaled vehicle simulation.

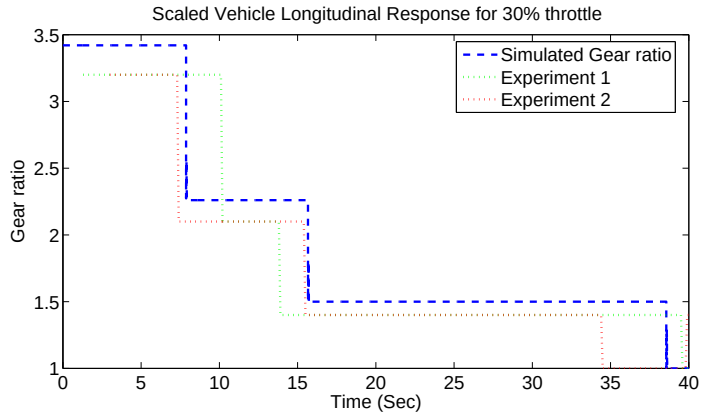


Figure 3.15: Gear ratio vs time for Scaled vehicle model and Scaled vehicle simulation.

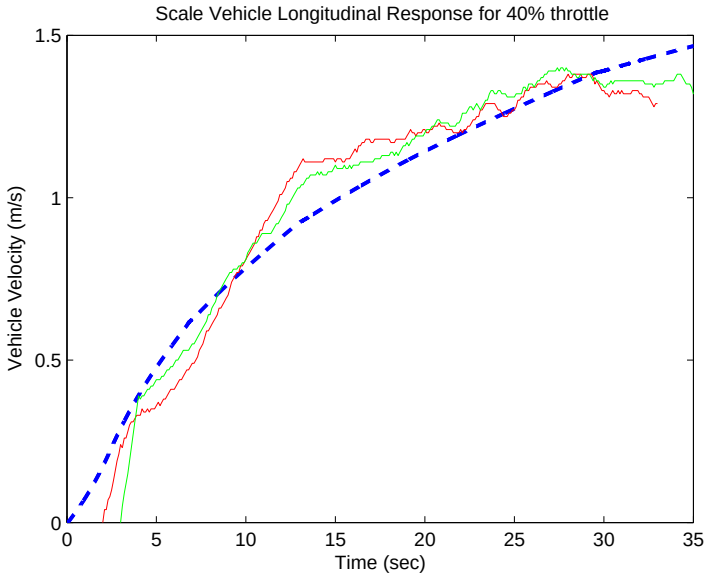


Figure 3.16: Vehicle speed vs time for Scaled vehicle model and Scaled vehicle simulation.

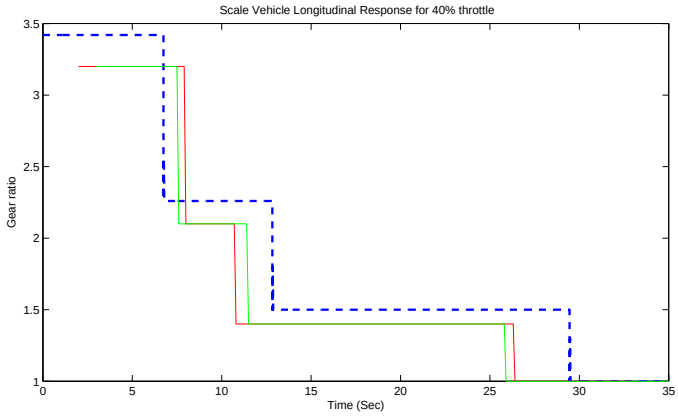


Figure 3.17: Gear ratio vs time for Scaled vehicle model and Scaled vehicle simulation.

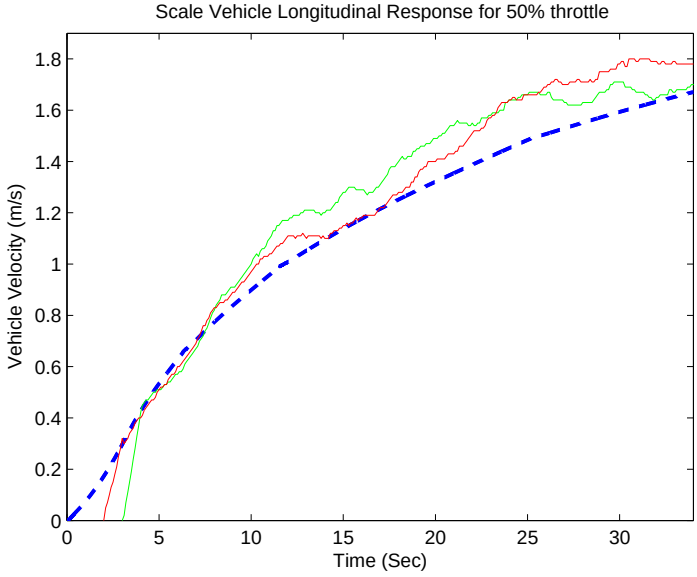


Figure 3.18: Vehicle speed vs time for Scaled vehicle model and Scaled vehicle simulation.

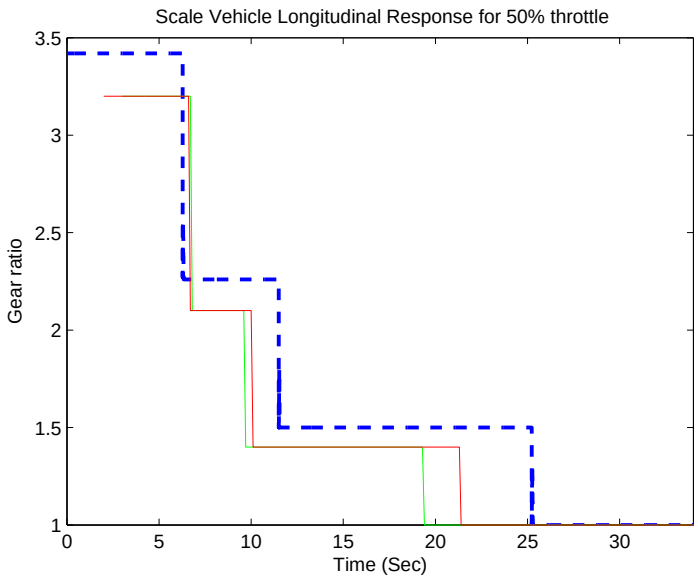


Figure 3.19: Gear ratio vs time for Scaled vehicle model and Scaled vehicle simulation.

an increasing trend in the RMS error. This, in part, can be attributed to the higher speeds attained by the vehicle with increasing throttle because of which we normalize RMS error with maximum speed. Normalized RMS error attained by the vehicle with 30 %, 40 % and 50 % is 0.4375, 0.559 and 0.605 respectively. The lower error for the 30 % throttle can be attributed to the fact that the motor map parameters were chosen so as to obtain the best results for the 30 % throttle (as explained in Section 3.4.1).

Gear ratio of the scaled vehicle as compared to the simulation is shown in Figures 3.15, 3.17 and 3.19. We see that the gearshift occurs before it is expected to occur, as indicated by the simulation, if the actual speed is more than simulated speed. Similarly, the gearshift occurs after it is expected to occur, as indicated by the simulation, if the actual speed is less than simulated speed. This agrees with the observed data. Table 3.7 presents the error in longitudinal response of prototype vs the scaled vehicle simulation.

Overall, it is observed that the scaled vehicle response matches the simulated response of the scaled vehicle both in terms of speed as well as gear shift vs time. The scaled vehicle simulation was shown to be dynamically similar to a HMMWV in Section 3.3.2. Thus, the

RMS error in speed for 30 % throttle	0.0525m/s
RMS error in speed for 40 % throttle	0.0809m/s
RMS error in speed for 50 % throttle	0.1099m/s
1st to 2nd gearshift time difference for 30 % throttle	1.25 sec
2nd to 3rd gearshift time difference for 30 % throttle	1 sec
3rd to 4th gearshift time difference for 30 % throttle	2.5 sec
1st to 2nd gearshift time difference for 40 % throttle	1 sec
2nd to 3rd gearshift time difference for 40 % throttle	1.8 sec
3rd to 4th gearshift time difference for 40 % throttle	3.3 sec
1st to 2nd gearshift time difference for 50 % throttle	.45 sec
2nd to 3rd gearshift time difference for 50 % throttle	1.6 sec
3rd to 4th gearshift time difference for 50 % throttle	4.5 sec

Table 3.7: Error in experimental results.

match of the experimental vehicle longitudinal response to that of the simulated vehicle is sufficient to prove dynamic similitude of the scaled vehicle to a HMMWV.

3.6 Acknowledgements

I am grateful to Jeff Lovell, Vishnu Desaraju and Jeff Powell who helped me with constructing the hardware, putting the software framework together and the experimentation. This work could not be complete without their help.

CHAPTER IV

Validation of safety control algorithms

In this chapter, we present the experimental validation of the theory proposed in Chapter 2. In particular, in Section 4.1, we describe the application scenario. Section 4.2 describes the experimental setup and Section 4.3 presents the experimental results.

4.1 Application example

In this section, we show how the formulation of the safety control problem for hidden mode hybrid systems and its solution, as proposed in Chapter 2, applies to the semi-autonomous intersection system. In doing so, we make use of various definitions and concepts formalized in Chapter 2. We introduce the application under study, that is, the two-vehicle conflict scenario depicted in Figure 4.1. Vehicle 1 is autonomous and communicates with the infrastructure, while vehicle 2 is human-driven and does not communicate its intents to the infrastructure nor to the other vehicle. We assume that the infrastructure measures the position and speed of vehicle 2 through road-side sensors such as cameras and magnetic-induction loops and that it transmits this information to the on-board controller of vehicle 1. Vehicle 1 has to use this information to avoid a collision.

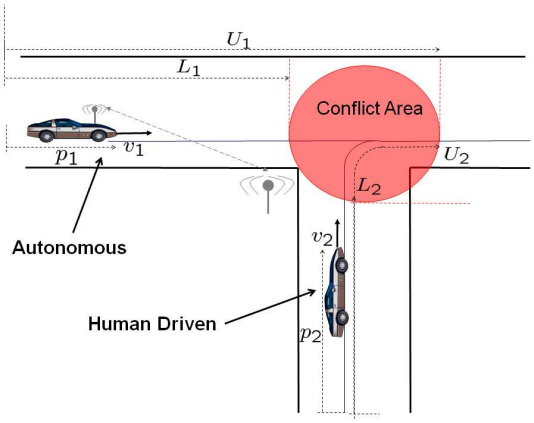


Figure 4.1: **Two-vehicle Conflict Scenario**. Vehicle 1, whose longitudinal displacement and speed are denoted p_1 and v_1, respectively, is autonomous and communicates with the infrastructure via wireless. Vehicle 2, whose longitudinal displacement and speed are denoted p_2 and v_2, respectively, is human-driven and does not communicate with the infrastructure. A collision occurs when more than one vehicle occupies the conflict area at the same time.

Vehicle 1 longitudinal dynamics along its path is given by the second order system

$$\begin{aligned} \dot{p}_1 &= v_1 \\ \dot{v}_1 &= a\,u + b - cv_1^2, \end{aligned} \tag{4.1}$$

in which p_1 is the longitudinal displacement of the vehicle along its path and v_1 is the longitudinal speed (see Figure 4.1), u represents the input command (positive when the vehicle accelerates and negative when the vehicle brakes), $b < 0$ represents the static friction term, and $c > 0$ with the cv_1^2 term modeling air drag (see Chapter 3 for more details on the model). The control input to vehicle 1 is the throttle or braking action u, which can range in the interval $[u_L, u_H]$ for given maximum braking action $u_L < 0$ and maximum throttle action $u_H > 0$.

Vehicle 2 is controlled by the driver decisions. There has been a wealth of work on the modeling of human driving behavior for vehicle safety applications, in which driving behavior is modeled through hybrid system models, wherein each mode corresponds to a

primitive behavior such as braking, acceleration, steering, run-out, lane change maneuver, etc. [6, 86]. Each mode is typically modeled through a (most often linear) dynamical system, whose parameters are in one-one correspondence with the mode. Depending on the number of modes, on the amount of "separation" between the nominal parameter values of each mode, and on the error about these nominal values, the accuracy of the mode classification problem changes [36, 35]. In this work, we model human driving behavior in the proximity of an intersection through a simple hybrid system with two modes: a braking mode and an acceleration mode, that is,

$$\begin{aligned} \dot{p}_2 &= v_2 \\ \dot{v}_2 &= \beta_q + \gamma_q d, \text{ with } q \in \{A, B\},\ d \in [-\bar{d}, \bar{d}], \end{aligned} \tag{4.2}$$

in which p_2 is the longitudinal displacement of the vehicle along its path and v_2 is the longitudinal speed (see Figure 4.1), $\bar{d} > 0$, q is the mode with $q = B$ corresponding to braking mode and $q = A$ corresponding to acceleration mode, and $\gamma_q > 0$. The value of β_q corresponds to the nominal dynamics of mode q and thus we have that $\beta_B < 0$ and that $\beta_A > 0$. The disturbance d models the error with respect to the nominal model. This implies that if $\dot{v}_2 \in \beta_q + \gamma_q[-\bar{d}, \bar{d}]$, the current mode can be mode q. This allowed error in each mode captures the fact that there are several ways in which mode A or mode B can be realized (for example, having harder braking or softer braking, harder acceleration or softer acceleration). It also captures (as we shall see in the experimental section) variability among drivers. We finally assume that there is confusion between the modes, that is, $\{\beta_B + \gamma_B[-\bar{d}, \bar{d}]\} \cap \{\beta_A + \gamma_A[-\bar{d}, \bar{d}]\} \neq \emptyset$, which leads to having $\beta_B + \gamma_B\bar{d} \geq \beta_A - \gamma_A\bar{d}$. As we will see in the experimental section, this is the case in practice due to subject variability and measurement errors.

Since the vehicles do not go in reverse, there is a lower non-negative speed limit, denoted v_{min}. Note that a strictly positive v_{min} also guarantees the liveness of the system

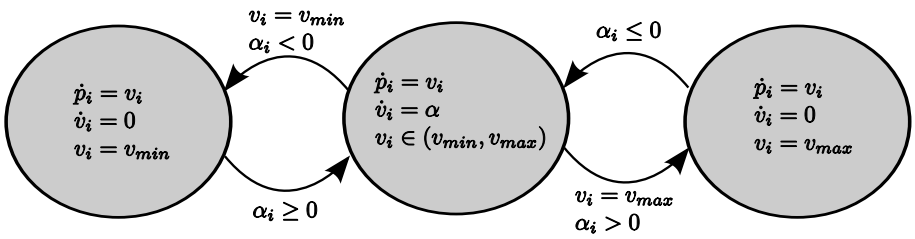


Figure 4.2: Hybrid model for vehicles 1 and 2. The switches are designed to enforce lower v_{min} and upper v_{max} bounds on the speed. In the diagram, we have denoted $\alpha_1 := a\,u + b - cv_1^2$ for vehicle 1 and $\alpha_2 = \beta_q + d$ for vehicle 2.

preventing vehicles to stop. Similarly, we allow an upper speed limit (which could be infinity), denoted v_{max}, to respect speed limitation regulations in the proximity of the intersection. This results in the hybrid system model for each vehicle depicted in Figure 4.2.

The safety control problem is to determine the least restrictive control actions for vehicle 1 to prevent a collision with vehicle 2 at the intersection. Given the coordinate system as depicted in Figure 4.1, collision configurations are those for which $(p_1, p_2) \in [L_1, U_1] \times [L_2, U_2]$.

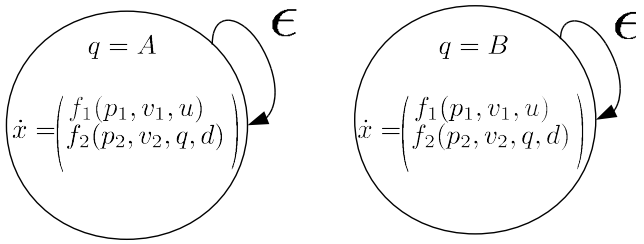


Figure 4.3: Hybrid automaton H. The continuous state is $x = (p_1, v_1, p_2, v_2)$, A refers to acceleration mode and B refers to braking mode for the human-driven vehicle.

The intersection system is a hybrid automaton with uncontrolled mode transitions H, shown in Figure 4.3 (see definition 2.2), in which $Q = \{A, B\}$; $X = \mathbb{R}^4$ and $x \in X$ is such

that $x = (p_1, v_1, p_2, v_2)$; $U = [u_L, u_H] \subset \mathbb{R}$; $D = [-\bar{d}, \bar{d}] \subset \mathbb{R}$; $\Sigma = \{\epsilon\}$ as there is no transition allowed between the modes; $R : Q \times \Sigma \to Q$ is the mode update map, which is trivial as $\Sigma = \{\epsilon\}$, that is, the mode can start in either A or B and no transitions ever occur between these two modes and $f : X \times Q \times U \times D \to X$ is the vector field, which is piecewise continuous and it is given by $f(x, q, u, d) = (f_1(p_1, v_1, u), f_2(p_2, v_2, q, d))$ in which

$$f_1(p_1, v_1, u) = \begin{pmatrix} v_1 \\ \begin{cases} 0 & \text{if } (v_1 = v_{min} \text{ and } \alpha_1 < 0) \text{ or } (v_1 = v_{max} \text{ and } \alpha_1 > 0) \\ \alpha_1 & \text{otherwise} \end{cases} \end{pmatrix}, \quad (4.3)$$

with $\alpha_1 = au + b - cv_1^2$ and

$$f_2(p_2, v_2, q, d) = \begin{pmatrix} v_2 \\ \begin{cases} 0 & \text{if } (v_2 = v_{min} \text{ and } \alpha_2 < 0) \text{ or } (v_1 = v_{max} \text{ and } \alpha_2 > 0) \\ \alpha_2 & \text{otherwise} \end{cases} \end{pmatrix}, \quad (4.4)$$

with $\alpha_2 = \beta_q + \gamma_q d$. The hybrid automaton H can also be viewed as the parallel composition of two hybrid automata H_1 and H_2 modeling vehicle 1 and vehicle 2, respectively. That is, $H = H_1 || H_2$, in which $H_1 = (Q_1, X_1, U_1, D_1, \Sigma_1, R_1, f_1)$, in which $Q_1 = \{0\}$, $X_1 = \mathbb{R}^2$ with $x_1 = (p_1, v_1)$, $U_1 = U = [u_L, u_H]$, $D_1 = \emptyset$, $\Sigma_1 = \emptyset$, $f_1(p_1, v_1, q_1, u_1, d_1) = f_1(p_1, v_1, u)$ is as given in equation (4.3). Similarly, $H_2 = (Q_2, X_2, U_2, D_2, \Sigma_2, R_2, f_2)$ in which $Q_2 = Q = \{A, B\}$, $X_2 = \mathbb{R}^2$ with $x_2 = (p_2, v_2)$, $U_2 = \emptyset$, $D_2 = D = [-\bar{d}, \bar{d}]$, $\Sigma_2 = \{\epsilon\}$, $f_2(p_2, v_2, q_2, u_2, d_2) = f_2(p_2, v_2, q, d)$ is as given in equation (4.4). Referring to Figure 4.1, the set of bad states for system H models collision configurations and it is given by $Bad := \{(p_1, v_1, p_2, v_2) \in \mathbb{R}^4 \mid (p_1, p_2) \in [L_1, U_1] \times [L_2, U_2]\}$.

For the application scenario, system $\hat{H} = (\hat{Q}, X, U, D, Y, \hat{R}, f)$, in which $\hat{Q} = \{\hat{q}_1, \hat{q}_2, \hat{q}_3\}$ with $\hat{q}_1 = \{A, B\}$, $\hat{q}_2 = \{A\}$, $\hat{q}_3 = \{B\}$, and $\hat{q}(0) = \hat{q}_1$, is uniquely defined once the set Y and map $\hat{R}$ are defined (see definition II.13). We define $Y = \{y_A, y_B, \epsilon\}$. Starting in $\hat{q}_1$, event y_A occurs as soon as B is not currently possible given the measurement x and event y_B occurs

as soon as A is not currently possible given the measurement x. This results in the map $\hat{R}$ defined as $\hat{R}(\hat{q}_1, y_A) := \hat{q}_2$ and $\hat{R}(\hat{q}_2, y_B) := \hat{q}_3$, which leads to the automaton of Figure 4.4.

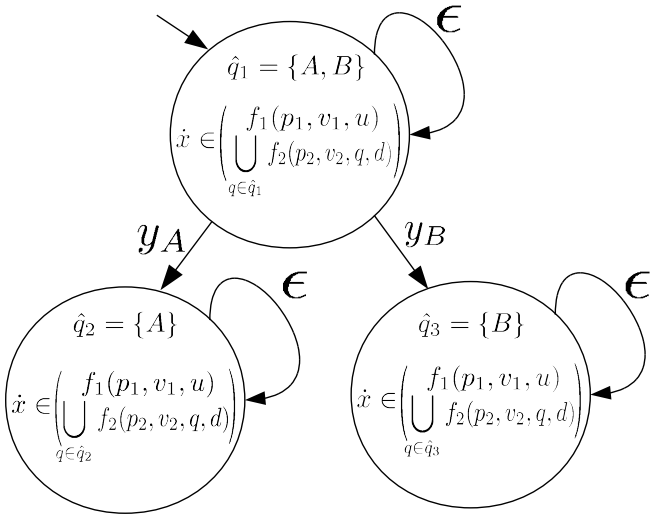


Figure 4.4: Hybrid automaton $\hat{H}$.

In order to establish when A or B are ruled out given the measurement of x, we consider the following estimate

$$\hat{\beta}(t) = \frac{1}{t} \int_0^t \dot{v}_2(\tau) d\tau, \; t \geq T, ^1$$

where $T > 0$ is a time window. If the mode is q, then necessarily we have that $|\hat{\beta}(t) - \beta_q| \leq \gamma_q \bar{d}$. Thus, for $t > T$, define

$$y(t) = \begin{cases} y_A & \text{if } |\hat{\beta}(t) - \beta_B| > \gamma_B \bar{d} \\ y_B & \text{if } |\hat{\beta}(t) - \beta_A| > \gamma_A \bar{d} \\ \epsilon & \text{otherwise} \end{cases} .$$

The properties of a mode estimator are satisfied. In fact (i) is satisfied as if q is currently possible (i.e., $|\hat{\beta} - \beta_q| \leq \gamma_q \bar{d}$), it cannot be discarded starting from $\hat{q}_1$. Similarly, once mode q is discarded, since R does not allow transitions, q cannot be possible even when

[1] Note that in practice, we will not require measurement of acceleration as we will consider discrete time models where derivative is replaced by time anticipation.

$|\hat{\beta} - \beta_q| \leq \gamma_q \bar{d}$. Condition (ii) is satisfied as $\hat{q}_2 \subseteq \mathcal{R}(\hat{q}_1)$ and $\hat{q}_3 \subseteq \mathcal{R}(\hat{q}_1)$.

Let us now use Algorithm II.30 for calculation of the Capture set. The system $\hat{H}$ is such that $\hat{Q} = \{\hat{q}_1, \hat{q}_2, \hat{q}_3\}$ with $\hat{q}_1 = \{A, B\}$, $\hat{q}_2 = \{A\}$, and $\hat{q}_3 = \{B\}$. As a consequence, we have that

$$G(S) := \begin{bmatrix} \text{Pre}\,(\hat{q}_1, S_2 \cup S_3 \cup Bad) \\ \text{Pre}\,(\hat{q}_2, Bad) \\ \text{Pre}\,(\hat{q}_3, Bad) \end{bmatrix},$$

so that

$$S^1 := \begin{bmatrix} \text{Pre}\,(\hat{q}_1, Bad) \\ \text{Pre}\,(\hat{q}_2, Bad) \\ \text{Pre}\,(\hat{q}_3, Bad) \end{bmatrix} \text{ and } S^2 := \begin{bmatrix} \text{Pre}\,(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad) \cup \text{Pre}(\hat{q}_3, Bad) \cup Bad) \\ \text{Pre}\,(\hat{q}_2, Bad) \\ \text{Pre}\,(\hat{q}_3, Bad) \end{bmatrix}.$$

By the properties of the Pre operator (see Proposition II.25), we have $B \subseteq \text{Pre}(\hat{q}_1, Bad)$, $\text{Pre}(\hat{q}_2, Bad) \subseteq \text{Pre}(\hat{q}_1, Bad)$, and $\text{Pre}(\hat{q}_3, Bad) \subseteq \text{Pre}(\hat{q}_1, Bad)$. Therefore $\text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad) \cup \text{Pre}(\hat{q}_3, Bad) \cup Bad) \subseteq \text{Pre}(\hat{q}_1, \text{Pre}(\hat{q}_1, Bad)) = \text{Pre}(\hat{q}_1, Bad)$. Also, we have that $\text{Pre}(\hat{q}_1, Bad) \subseteq \text{Pre}\,(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad) \cup \text{Pre}(\hat{q}_3, Bad) \cup Bad)$. Therefore we finally obtain that

$$\text{Pre}\,(\hat{q}_1, \text{Pre}(\hat{q}_2, Bad) \cup \text{Pre}(\hat{q}_3, Bad) \cup Bad) = \text{Pre}(\hat{q}_1, Bad),$$

so that Algorithm 1 terminates at the second step. Therefore, we have that $\hat{C}_{\hat{q}_1} = \text{Pre}(\hat{q}_1, Bad)$, $\hat{C}_{\hat{q}_2} = \text{Pre}(\hat{q}_2, Bad)$ and $\hat{C}_{\hat{q}_3} = \text{Pre}(\hat{q}_3, Bad)$.

4.1.1 Computational tools

The sets $\text{Pre}(\hat{q}, Bad)$ can be efficiently computed for the application under study. This is because for every mode estimate $\hat{q}$ the continuous dynamics are the parallel composition of two order preserving systems [51, 33]. Specifically, for the application example, define

the restricted Pre operators for $i \in \{1, 2, 3\}$

$$\text{Pre}(\hat{q}_i, Bad)_{u_L} := \{x \in X \mid \exists\, \mathbf{d},\ t \geq 0 \text{ s.t. some } \phi_{\hat{x}}(t, (\hat{q}_i, x), u_L, \mathbf{d}, \epsilon) \in Bad\}$$

and

$$\text{Pre}(\hat{q}_i, Bad)_{u_H} := \{x \in X \mid \exists\, \mathbf{d},\ t \geq 0 \text{ s.t. some } \phi_{\hat{x}}(t, (\hat{q}_i, x), u_H, \mathbf{d}, \epsilon) \in Bad\}.$$

Then, we have that (refer to [51])

$$\text{Pre}(\hat{q}_i, Bad) = \text{Pre}(\hat{q}_i, Bad)_{u_L} \cap \text{Pre}(\hat{q}_i, Bad)_{u_H} \text{ for } i \in \{1, 2, 3\}. \tag{4.5}$$

Each of the sets $\text{Pre}(\hat{q}_i, Bad)_{u_L}$ and $\text{Pre}(\hat{q}_i, Bad)_{u_H}$ can be computed by linear complexity discrete time algorithms. This is because the control input u is fixed so that only plain backward integration is needed. Backward integration is in turn very efficient because of the order preserving property of the dynamics (see Section 4.2.6).

For each mode $\hat{q}_i$ for $i \in \{1, 2, 3\}$, a safe control map $\hat{\pi}(\hat{q}_i, x)$ acts in such a way to maintain the state outside the current mode-dependent capture set $\hat{C}_{\hat{q}}$. This results in a map $\hat{\pi}(\hat{q}_i, x)$ that makes the vector field point outside set $\hat{C}_{\hat{q}_i}$ when x is on the boundary of $\hat{C}_{\hat{q}_i}$. For $\hat{H}$, we have $\hat{C}_{\hat{q}_i} = \text{Pre}(\hat{q}_i, Bad)$ for all $i \in \{1, 2, 3\}$, in which the sets $\text{Pre}(\hat{q}_i, Bad)$ for every $i \in \{1, 2, 3\}$ satisfy relation (4.5). Because of this relation, one can show (refer to [51]) that a control map $\hat{\pi}(\hat{q}_i, x)$ that maintains the state x outside $\text{Pre}(\hat{q}_i, Bad)$ is given by

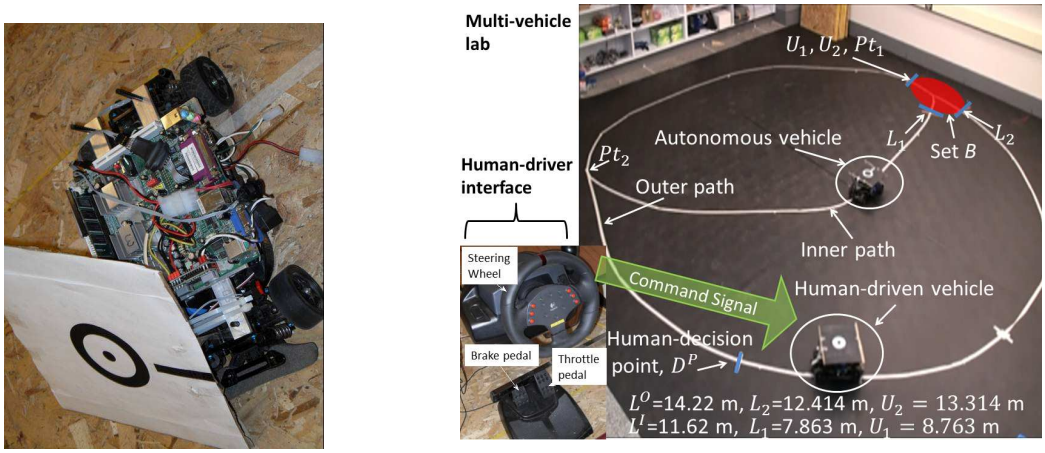
$$\hat{\pi}(\hat{q}_i, x) := \begin{cases} u_H & \text{if } x \in \text{Pre}(\hat{q}_i, Bad)_{u_L} \text{ and } x \in \partial\text{Pre}(\hat{q}_i, Bad)_{u_H} \\ u_L & \text{if } x \in \text{Pre}(\hat{q}_i, Bad)_{u_H} \text{ and } x \in \partial\text{Pre}(\hat{q}_i, Bad)_{u_L} \\ \{u_H, u_L\} & \text{if } x \in \partial\text{Pre}(\hat{q}_i, Bad)_{u_H} \text{ and } x \in \partial\text{Pre}(\hat{q}_i, Bad)_{u_L} \\ U & \text{otherwise.} \end{cases}$$

Since we have that $\text{Pre}(\hat{q}_i, Bad) \subseteq \text{Pre}(\hat{q}_1, Bad)$ for $i \in \{2, 3\}$, when the mode switches from

$\hat{q}_1$ to $\hat{q}_2$ or from $\hat{q}_1$ to $\hat{q}_3$ the continuous state x being outside Pre($\hat{q}_1$, *Bad*) implies that it is also outside Pre($\hat{q}_2$, *Bad*) or Pre($\hat{q}_3$, *Bad*). Therefore, the feedback map above guarantees that the state never enters the capture set.

4.2 Experimental setup

The two-vehicle conflict scenario of Figure 4.1 was implemented in an in-scale multi-vehicle lab. In this section, we briefly describe the hardware and software setup and implementation details pertaining to the lab. The lab is equipped with a over-head camera-based positioning system, a control station, a human-driver interface, the roundabout system and six scaled vehicles. Various software modules are developed for each of these components, notably the software on-board the vehicle that implements various vehicle subroutines, the control algorithms and communication subroutines, the software on the control station and the software for the positioning system. These components are briefly described in this section.



(a) The scaled vehicle.

(b) The roundabout system.

Figure 4.5: Sub-figure (a) shows the scaled vehicle with its label, sub-figure (b) the human-driver interface and the roundabout system, L^O is the length of the outer path while L^I is the length of the inner path.

4.2.1 Scaled vehicle and human-driver interface

A car chassis (length 0.375 m, width 0.185 m and wheelbase 0.257 m) [2] is used as the hardware platform for the scaled vehicle. The vehicle, as shown in Figure 4.5 (a), is equipped with an on-board computer (Mini ITX) and a motion controller. The longitudinal dynamics of this vehicle are dynamically similar to that of a high mobility multipurpose wheeled vehicle (HMMWV). The in-scale vehicle acts as a hardware-in-the-loop setup that implements the scaled engine, fluid coupling, transmission and gearshift logic of a HMMWV. See Chapter 3 for more details on this implementation.

One of the scaled vehicles is configured to be an autonomous vehicle that can follow a predefined path and control its throttle/brake input while another acts as a human-driven vehicle that can be driven by a human-driver using a human-driver interface. The human-driver interface comprises a steering wheel and two pedals for throttle and brake commands (see Figure 4.5 (b)). The hardware used is a Logitech MOMO force feedback racing wheel and pedal set [2]. The steering wheel has a range of -100^0 to 100^0. The hardware gives an output in the range of -1000 to 1000 for both brake and throttle pedal, where -1000 corresponds to pedal not being pressed and 1000 corresponds to pedal being fully pressed. These values are scaled to a range of 0 to 100, where 0 corresponds to pedal not being pressed and 100 corresponds to the pedal being fully pressed. The hardware is connected to the control station via a USB cable and the input command from the hardware is transmitted to the vehicle controlled by the human via the wireless connection.

The high level control algorithms are programmed on the on-board computer. A data-logging module is programed on the on board computer which can read vehicle speed from the motioncontroller at a frequency of 10 samples/sec. The drivetrain components, including engine, fluid coupling, transmission, gear shift logic and final drive are programmed on the motioncontroller. The motioncontroller issues control signals to the steering servo and controls the PWM signal to the DC motor. The programming on the motioncontroller is

[2] See $http://www.logitech.com/index.cfm/gaming/wheels/devices/320\&cl=us,en$ for specifications

performed in Tiny Embedded Application (TEA) [3] language which is a subset of the C programming language.

4.2.2 Roundabout system

The roundabout system is designed to replicate a collision situation at a road intersection where two vehicles merge. Figure 4.5 (b) shows the roundabout system. There are two circular paths that share a common section on a 6 m by 6 m arena. The human-driven vehicle follows the outer path while the autonomous vehicle follows the inner path. Both of the vehicles travel in an anti-clockwise direction. A collision is possible at the intersection when both vehicles are in the area shaded red, in Figure 4.5 (b), at the same time. This area corresponds to the set, $B = \{(p_1, p_2) | (p_1, p_2) \in [L_1, U_1] \times [L_2, U_2]\}$, with $L_1 = 7.863$ m, $L_2 = 12.414$ m, $U_1 = 8.763$ m and $U_2 = 13.314$ m. The length of the outer path is 14.22 m and the inner path is 11.62 m. The human-driver controls the vehicle from the human-driver interface and has the full view of the roundabout system at any time. Point D^P is referred to as the human-decision point, this is the point where the human-driver has to decide if he/she wants to break or accelerate in order to force both vehicles to enter the bad set at the same time. We have assumed that point D^P is located 6 m before the point L_2 on the outer path.

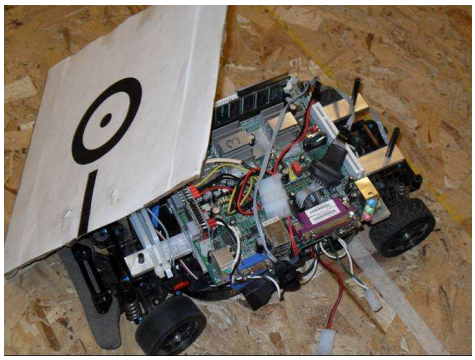
The maximum allowable speed that can be achieved by a vehicle in the roundabout system is 1100 mm/sec and the minimum speed is 350 mm/sec. A software module is present on all the vehicles that maintains the speed at minimum or maximum if the speed drops below the minimum or exceeds the maximum speed limit. When the two vehicles are simultaneously present in the shared path (between points Pt_1 and Pt_2), another software module prevents rear end collision by appropriately accelerating or decelerating the autonomous vehicle when the two vehicles are too close. The maintain speed and rear end collision prevention modules are based on a simple PID control scheme.

The lab is equipped with an overhead camera based positioning system. The positioning

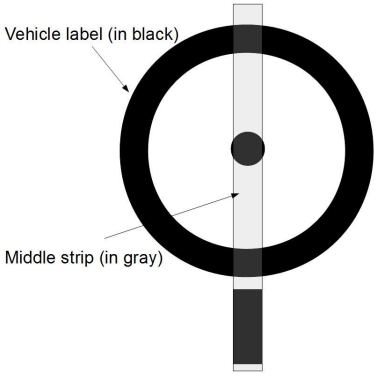
system can simultaneously monitor 6 vehicles and provide the position coordinates with an accuracy of 50 mm. The system consists of six cameras mounted on the ceiling in a way to cover the whole of the lab area. Each vehicle is mounted with distinguishable label that can be tracked (Figure 4.5 (a)). The labels are concentric white or black circles on a white or black background. The cameras are connected via FireWire to three dedicated desktop computers which implement the tracking software. Each computer receives input from two cameras and runs image processing and tracking algorithms developed in the lab [3]. These algorithms are implemented in C++. Each camera runs on its own PCI card. This allows running two cameras per computer without any loss in frame rate. The tracking algorithms are run independently on each computer rather than having a central computer do all the processing. The lab is divided into three zones. Each zone corresponds to the area viewed by two cameras attached to a particular computer. The computer tracks the vehicle labels present in its zone at any time and transmits the positioning information to the vehicles over the wireless network. Once a vehicle label is about to exit its current zone, a message is sent to the computer corresponding to the next zone it is entering. This computer now takes over the responsibility of tracking the vehicle label. Tracking is performed by doing a pixel-by-pixel comparison of the image captured by the camera with the images of the vehicle label references.

For initialization, the tracking software performs an exhaustive search of all region to look for vehicle labels. Once it has initialized the position of vehicles, tracking of vehicle position is performed by conducting a localized search around the vehicle. Computational load is reduced by searching for middle strip of the label and not the whole label. This also reduces the effect of image distortion, that can occur if the label is not mounted correctly on the vehicle. The vehicle position is converted to a global coordinate system and is transmitted to the vehicles via the lab network.

[3] http://wiki.eecs.umich.edu/delvecchio/



(a) Scaled vehicle with label



(b) Label with middle strip

Figure 4.6: The vehicle with a label.

4.2.3 Learning of human driving model

As mentioned in Section 4.1, we model the human-driven vehicle using a hybrid automaton whose discrete state models the intention of the human-driver. In this work, we assume that the human either decides to brake or accelerate near the intersection, depending on the situation. A set of experiments are performed in which human subjects drive a vehicle on the outer path in the roundabout system (Figure 4.5 (b)). The driving task is designed to generate a model of the human behavior near the intersection of the inner and outer path (collision point) when the human decides to either brake or accelerate. In these experiments, since we intend to characterize the human driving model, the subjects drive

the vehicle in the outer path and are directed to either brake or accelerate at the human-decision point in Figure 4.5 (b), while also avoiding a moving target on the inner path. The data collected in these braking and acceleration trials is then analyzed to estimate the parameters β_q and γ_q as described in Section 4.1.

The experiment is started by first allowing the subject to get accustomed to driving the vehicle using the human-driver interface. Once the subject is confident, he/she is asked to run one trial each of braking/acceleration, in which the vehicle is run on the outer path and the subject decelerates/accelerates the vehicle once the decision point is crossed. In the trials, the vehicle is started 2 m after the collision point (see Figure 4.5(b)) at a random velocity. The steering is locked at a value such that the vehicle approximately follows the outer path. At the decision point (4 m before the collision point) the driver is asked randomly to either pass through the intersection before the moving target on the inner path or let the moving target pass the intersection before the human-driven vehicle. The former case is considered acceleration trial while the latter case is a braking trial. The subject is allowed to take control of the vehicle 3 to 4 m before the collision point.We used 5 different subjects to run 10 acceleration and 10 braking trials each. For each of these trials, the position of the vehicle and time at which this measurement was taken are recorded. The data is analyzed starting 3 m before the collision point. We denote the position measurement at time step k by $p(k)$ and the time lapsed between two consecutive steps is dT =0.1 sec. The acceleration/deceleration at time step k is denoted $a(k)$ and can be calculated as

$$a(k) = \frac{p(k) - 2p(k-1) + p(k-2)}{dT^2}.$$

The average acceleration/deceleration is calculated for the trial as $\bar{a} = \frac{1}{N-1}\sum_{k=2}^{N} a(k)$. A total of 99 trial runs are obtained from 5 subjects.

These trials are divided into the training set, comprising 79 trials with 40 braking and 39 acceleration trials, and the test set comprising of the remaining trials. The model depicting the driver behavior is created by fitting two Gaussian distributions to the training data for

braking and acceleration trials and then using the test data to verify the model. In order to obtain the best model, more than 1000 randomly chosen training and testing sets are considered. The average training and testing errors are .56% and .96% respectively. For use as the final model, we chose a model with no training and testing error. The confusion matrices and parameters associated with the model are shown in Tables 4.1, 4.2, and 4.3. The associated Gaussian distribution is shown in Figure 4.7. From these results, we have that the mean of the acceleration mode is 350.7 mm/sec^2 and that of braking mode is -282.7 mm/sec^2. We thus take the value of parameters in equation (4.2) as β_B = -282.7 mm/sec^2 and β_A=350.5 mm/sec^2. The values of γ_B and γ_A are given by γ_A =139.6 mm/sec^2 and γ_B =106.6 mm/sec^2. The value of $\bar{d}$ is set to $\bar{d}$ = 3 and corresponds to three standard deviations. This also results in an overlap of human input range in braking and acceleration modes.

Confusion Matrix: Training set	Brake	Acceleration
Brake	40	0
Acceleration	0	39

Table 4.1: Training set confusion matrix.

Confusion Matrix: Test set	Brake	Acceleration
Brake	10	0
Acceleration	0	10

Table 4.2: Test set confusion matrix.

4.2.4 Trials experimental conditions

This section presents the experimental conditions. A total of eight human subjects participated in the study. This set of subjects is different from the set used to generate the human driving model. This confirms that the human driving model can generalize and is able to identify the intent of human subjects not present during training.

Errors	Training	Test
Error (%)	0	0
Decision Boundary	-2.18 mm/sec^2	
Mean (β_B, β_A)	(-282.7,350.5) mm/sec^2	
Standard Deviation (γ_B, γ_A)	(106.6,139.6) mm/sec^2	
Gaussian Fit Error	1.00 %	
Samples	79	20

Table 4.3: The parameters of the final model.

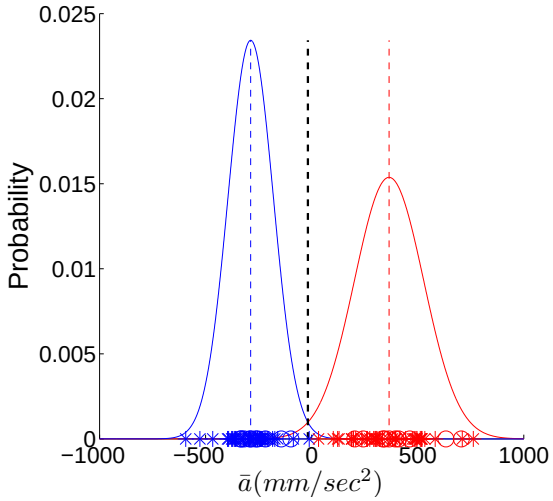


Figure 4.7: The Gaussian distribution for braking and acceleration trials. The mean of braking mode is β_B = -282.7 mm/sec^2 and standard deviation is γ_B = 106.6 mm/sec^2. The mean of acceleration mode is β_A = 350.5 mm/sec^2 and standard deviation is γ_A = 139.6 mm/sec^2.

To start the experiment, the subjects are given an introduction about the setup. This is followed by a practice session in which the subject drives the human controlled vehicle on the outer path. Once the subject is comfortable with maneuvering the vehicle, we proceed to the next step of the experiment. The autonomous vehicle is run on the inner path at a constant speed of 500 mm/sec. The maximum and minimum speeds that can be achieved by both the human and the autonomous vehicle are 350 mm/sec and 1100 mm/sec, respectively. The subjects are instructed to drive the human controlled vehicle on the outer path. They are free to drive the human-driven vehicle at any speed between the points Pt_1 and Pt_2. Between point Pt_2 and D^P, the maintain speed module keeps the vehicle speed at 600 mm/sec. This is performed to ensure that the human-driven vehicle does not cross the decision point with minimum or maximum speed. If the vehicle speed is v_{min} and the subject decides to apply brake at the decision point, no discernible change in the vehicle speed will be noticeable. This will result in the mode being identified as both braking and acceleration resulting in the largest capture set. Since we are interested in scenarios where the mode can be distinguished, the subjects are instructed to not apply any control between point Pt_2 and D^P. A collision situation is one in which the current position of the vehicles is in the set B ($B = \{(p_1, p_2) | (p_1, p_2) \in [L_1, U_1] \times [L_2, U_2]\}$, as shown in Figure 4.5 (b)) at the same time. Thus, we instruct the human subjects to either accelerate or decelerate as soon as they cross the decision point, indicated D^P on Figure 4.5 (b), in order to hit the autonomous vehicle or to force the two vehicles in the bad set at the same time.

During the trial, data relevant to the internal state of the autonomous and the human-driven vehicle are recorded at a frequency of 10 samples per second, on the computer on-board each vehicle. The recorded signals include the position coordinates (x, y) of both vehicles, the derived velocities of both vehicles, the derived acceleration of both vehicles, the estimated discrete state of the human-driven vehicle, the control signals of the autonomous vehicle and the data relevant to collision avoidance algorithm implementation.

After the conclusion of the trial , this data can be downloaded on to a computer via

the wireless connection between the vehicle's on-board computer and the computer where data is being downloaded. This data can then be analyzed to check whether the controller performed as desired during the experiment. We next define the mode estimator used for estimating the set of current possible modes for the human driven vehicle.

4.2.5 Mode estimator

In the application scenario of Section 4.1, the human-driven vehicle is modeled as a hybrid automaton with two modes: braking and acceleration. However, the system mode at any time is not measured and so it is unknown. In the absence of exact current system mode information, an estimator is designed that provides a set of modes in which the current mode must lie. One such estimator is proposed in Section 4.1. For implementation in the current experimental setup, we use the discrete time form of this estimator. Since the driver decides to switch the mode to brake or accelerate once the human-driven vehicle crosses D^P on the roundabout system, the mode estimator running on the autonomous vehicle uses the continuous state measurements of human-driven vehicle after it crosses D^P.

The instance $n = 0$ corresponds to the time step when the human-driven vehicle crosses the decision point. The position measurement of the human-driven vehicle at k^{th} time step is $p(k)$ and dT =0.1 sec is the time step. We take a time window of $N = 20$ for the above calculation. Consider $n > N$. At the n^{th} time step after the human-driven vehicle crosses the human-decision point, the estimate is calculated by using the formula:

$$\hat{\beta}(n) = \frac{1}{n-1}\Sigma_{k=2}^{n} a(k)$$

where, $a(k) = \frac{p(k)-2p(k-1)+p(k-2)}{dT^2}$ is the acceleration at k^{th} time step. Then, n time step after the human-driven vehicle crosses the decision point, $y(n)$ is given by

$$y(n) = \begin{cases} y_A & \text{if } |\hat{\beta}(n) - \beta_B| > \gamma_B \bar{d} \\ y_B & \text{if } |\hat{\beta}(n) - \beta_A| > \gamma_A \bar{d} \\ \epsilon & \text{otherwise} \end{cases}.$$

4.2.6 Control map implementation

We introduce the following discretization of system H given in equations (4.3)-(4.4) (employing forward Euler approximation) with step size $dT > 0$, $i \in \{1, 2\}$, and index j

$$\begin{aligned} p_i[j+1] &= p_i[j] + F_1^i(v_i[j], \alpha_i[j]) \\ v_i[j+1] &= \bar{F^i}(v_i[j], \alpha_i[j]) \end{aligned}$$

where, for the application example, $F_1^i = dT\, v_i[j]$, $\bar{F^i}(v_i[j], \alpha_i[j]) = v_i[j] + dT\, \gamma(v_i[j], \alpha_i[j])$, $\gamma(v, \alpha) := \alpha$ if $v + \alpha dT < v_{max}$ and $v + \alpha dT > v_{min}$, $\gamma(v, \alpha) := v_{max} - v$ if $v + \alpha dT > v_{max}$, and $\gamma(v, \alpha) := v_{min} - v$ if $v + \alpha dT < v_{min}$. With these definitions, one can check that $v_i + dT\gamma(v_i, \alpha_i)$ is an order preserving function in both v_i and α_i. We define the notation for a sequence of constant inputs α_i for $i \in \{1, 2\}$

$$\begin{aligned} \bar{F^{i,0}}(v_i, \alpha_i) &:= v_i \\ \bar{F^{i,k+1}}(v_i, \alpha_i) &:= \bar{F^i}(\bar{F^{i,k}}(v_i, \alpha_i), \alpha_i), \end{aligned}$$

with, $k \in \mathbb{N}$. The value of $p_i[k]$ starting from initial conditions (p_i, v_i) can be calculated as

$$p_i[k] = p_i + \sum_{j=0}^{k-1} F_1^i \left(\bar{F^{i,j}}(v_i, \alpha_i), \alpha_i \right).$$

Since $B = [L_1, U_1] \times [L_2, U_2]$ and $Bad = [L_1, U_1] \times \mathbb{R} \times [L_2, U_2] \times \mathbb{R}$, define for $i \in \{1, 2\}$ the sequences

$$\begin{aligned}
L_1^k(v_1, \alpha_1) &:= L_1 - \sum_{j=0}^{k-1} F_1^i\left(\bar{F}^{i,j}(v_1, \alpha_1), \alpha_1\right) \\
U_1^k(v_1, \alpha_1) &:= U_1 - \sum_{j=0}^{k-1} F_1^i\left(\bar{F}^{i,j}(v_1, \alpha_1), \alpha_1\right) \\
L_2^k(v_2, \max(\alpha_2)) &:= L_2 - \sum_{j=0}^{k-1} F_1^i\left(\bar{F}^{i,j}(v_2, \max(\alpha_2)), \max(\alpha_2)\right) \\
U_2^k(v_2, \min(\alpha_2)) &:= U_2 - \sum_{j=0}^{k-1} F_1^i\left(\bar{F}^{i,j}(v_2, \min(\alpha_2)), \min(\alpha_2)\right),
\end{aligned}$$

in which $\max(\alpha_2) = \beta_q + \gamma_q \bar{d}$ and $\min(\alpha_2) = \beta_q - \gamma_q \bar{d}$ when $\hat{q} = q$, while $\max(\alpha_2) = \beta_A + \gamma_A \bar{d}$ and $\min(\alpha_2) = \beta_B - \gamma_B \bar{d}$ when $\hat{q} = \{A, B\}$. Then, one can show that

$$\begin{aligned}
\text{Pre}(\hat{q}, Bad)_u = \Big\{ x \in X \mid \exists\, k \geq 0 \text{ s. t. } & L_1^k(v_1, \alpha_1) < p_1 < U_1^k(v_1, \alpha_1) \text{ and} \\
& L_2^k(v_2, \max(\alpha_2)) < p_2 < U_2^k(v_2, \min(\alpha_2)) \Big\}.
\end{aligned}$$

Hence, given mode estimate $\hat{q}$, $\text{Pre}(\hat{q}, Bad)_{u_L}$ and $\text{Pre}(\hat{q}, Bad)_{u_H}$ are computed for the given pair of speeds (v_1, v_2) as a union of rectangles in the position plane. Checking whether a point $x = (p_1, v_1, p_2, v_2)$ is in $\text{Pre}(\hat{q}, Bad)_{u_L} \cap \text{Pre}(\hat{q}, Bad)_{u_H}$ is performed by comparing (p_1, p_2) against the upper and lower bounds L_1^k, U_1^k, L_2^k and U_2^k. Moreover, to check whether $p_1 \in [L_1^k, U_1^k]$, it is enough to compute such intervals only while $U_1^k > p_1$, since the sequences $\{L_1^k\}_{k\geq 0}$, $\{U_1^k\}_{k\geq 0}$, $\{L_2^k\}_{k\geq 0}$ and $\{U_2^k\}_{k\geq 0}$ are strictly decreasing [51]. Thus, we only need to make a finite number of computations.

To implement the feedback map $\hat{\pi}(\hat{q}, x)$ of Section 4.1.1, we need to track when the continuous flow hits the boundary of the relevant set Pre(.,.). In discrete time, we accomplish this by projecting the continuous state forward in time. Specifically, we consider the continuous state to be on the boundary of Pre(.,.) when it is outside it while its projection

forward in time is inside it. To make this procedure robust to both communication and actuator delays, we consider $n_p = 10$ forward projections in time instead of one only.

4.3 Experimental results

We now present experimental results for the semi-autonomous collision avoidance system. A total of eight subjects took part in the experiments. The duration of each trial depends on the time each vehicle can operate on a single battery charge. The vehicle battery, when fully charged, yields an operating time of around 10 to 15 minutes. This operation time is divided into the driver training time and the actual experimentation time. Some subjects learn to drive the vehicle and follow the outer path closely in a reasonable time (less than 5 minutes), while others are unable to follow the path closely even after the training for more than half of the total operation time. The variation in subjects results in the variation of trial length. The shortest trial length that we obtained is 230 seconds while the longest is 600 seconds.

The cumulative time for which the trials are conducted is 3479 seconds resulting in a total of 97 instances of collision avoidance in which the human-driver tried to force a collision and the autonomous vehicle applied control in order to avoid the collision. In doing so, the autonomous vehicle entered the capture set in 3 such instances and resulted in a collision in 1 such instance resulting in an overall success rate of 96.9 %. The total number of times that the mode is locked is 161 times. Out of 161 times, the total number of times the mode is locked at A (acceleration) is 102, at B (braking) is 45 times and at $\{A, B\}$ (acceleration or braking) is 10 times. These results are presented in table 4.4 and 4.5.

On our data set, the flow entered the capture set only 3% of the times. These failures are due mainly to the fact that communication delays between the vehicle and the workstation are not formally accounted for by the algorithm. These delays, when significant, cause the calculated capture set to be different from the actual one and hence may cause to enforce control too late. Other possible causes for such failure can be violation of speed limits

by the vehicle and the actuator delay. It is assumed that the speed of the vehicle remains between v_{min} and v_{max}. However, the vehicle speed can violate these limits for a few time steps causing a failure. The actuators do not repond instantaneously to the control signal and may cause a failure.

More complex models of human decisions in the proximity of an intersection, such as those studied in [92], and the incorporation of additional details such as weather conditions and road geometry, offer the potential for reducing the conservatism of safe control actions even further. Therefore, further work will explore the refinement of these models by incorporating additional domain-specific knowledge. The models here considered are deterministic because most of the tools currently available to perform safety control have assumed deterministic models, wherein uncertainty is bounded. However, human decision models are more naturally captured by stochastic frameworks, in which uncertainty due to variability in both subjects and realizations of the same decision is probabilistic (see [71] for a review on the topic). As results in stochastic safety verification and design become available [5, 7], it will be important to extend the proposed techniques of this paper to safety control of stochastic hybrid automata, in which the mode estimate is constructed probabilistically as opposed to being constructed deterministically.

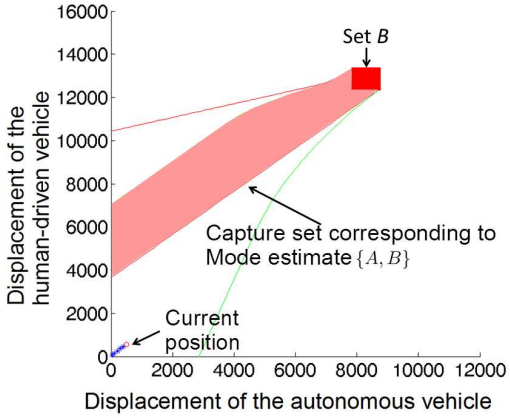
Figures 4.8, 4.9 and 4.10 show collision avoidance instances when the human-driven vehicle mode is identified as A, B and $\{A, B\}$, respectively.

Subject number	Total time (sec)	Number of mode detections	Mode A	Mode B	Mode $\{A, B\}$
1	374.8	16	9	6	1
2	265	13	8	5	0
3	258	9	5	3	1
4	670	28	18	6	2
5	560	29	17	7	3
6	230	13	11	2	0
7	522	27	16	10	0
8	600	26	18	6	2

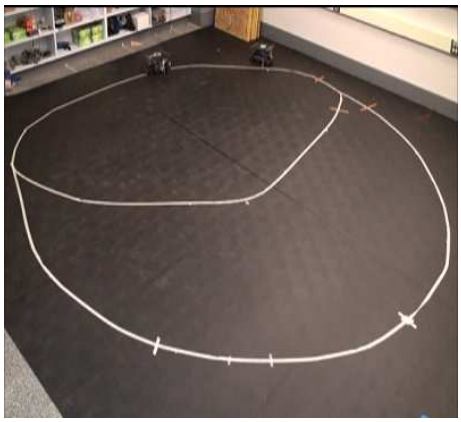
Table 4.4: Mode detection for various subjects. The first column shows the subject number, the second column presents the total time the trial lasted, the third column presents the number of mode detections that occurred during the given trial, the fourth column shows the number of times the mode is locked at acceleration $\{A\}$, the fifth column shows the number of times the mode is locked at braking $\{B\}$, and the sixth column shows the number of times mode could not be locked and remained at $\{A, B\}$.

Subject number	Number of CA instances	Times entered $\hat{C}$	Times entered *Bad*
1	14	1	0
2	8	1	0
3	5	1	1
4	19	0	0
5	6	0	0
6	7	0	0
7	16	0	0
8	22	0	0

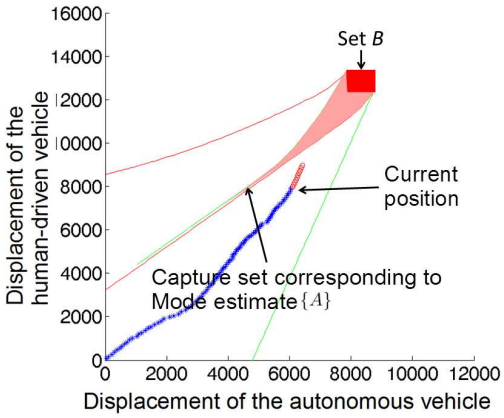
Table 4.5: Collision avoidance instances for various subjects. ”CA” stands for collision avoidance and $\hat{C}$ is the capture set. The first column is the subject number, the second column shows the number of collision avoidance instances generated by the subject. The third column shows the times the flow enters the capture set while the control is on. The last column shows the number of times the flow entered the bad set, *Bad*.



(a) Unknown mode, i.e., $\hat{q} = \{A, B\}$, time = 81.9 seconds.

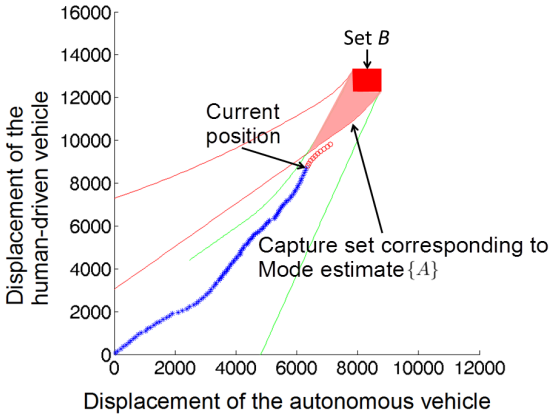


(b) Lab snapshot corresponding to (a).



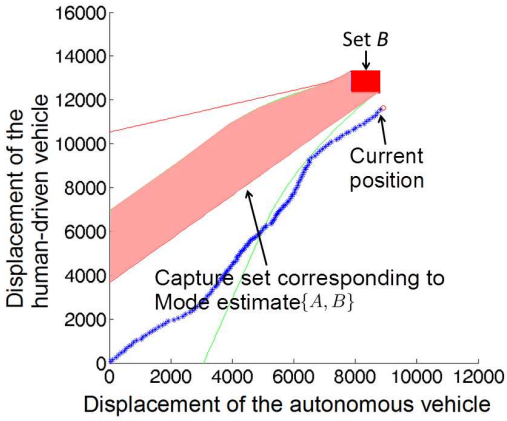
(c) Mode identified as acceleration, i.e., $\hat{q} = \{A\}$, time = 91.3 seconds.

(d) Lab snapshot corresponding to (c).

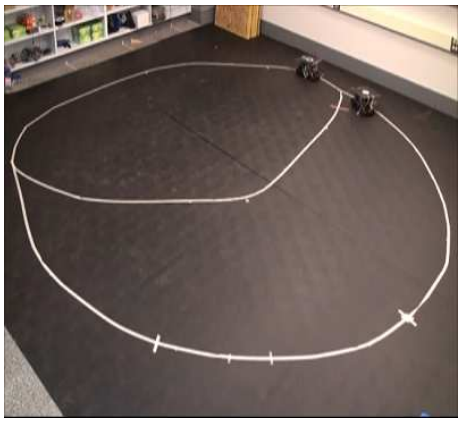


(e) Autonomous vehicle applies control, time = 91.9 seconds.

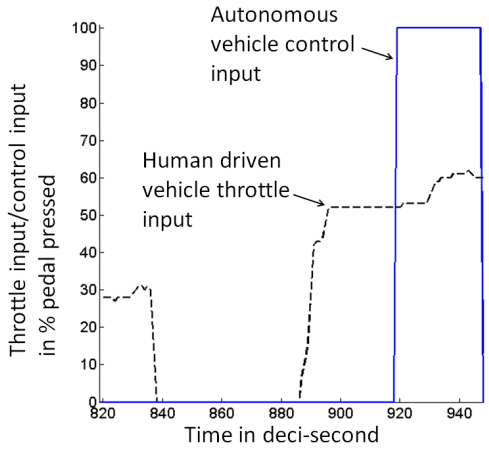
(f) Lab snapshot corresponding to (e).



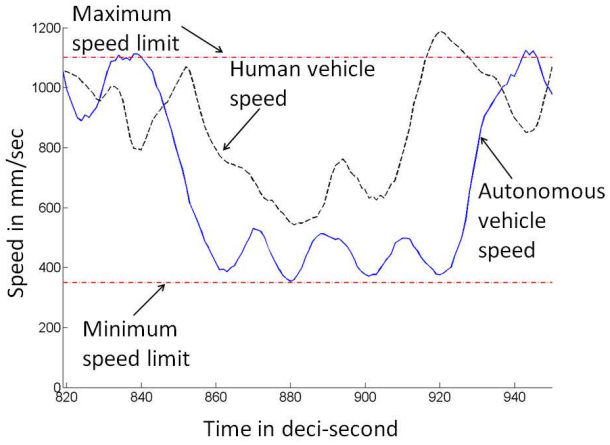
(g) Collision averted, time = 94.8 seconds.



(h) Lab snapshot corresponding to (g).

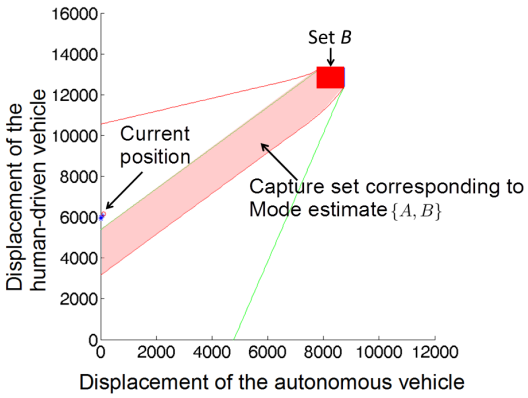


(i) Throttle input of human and autonomous vehicle.

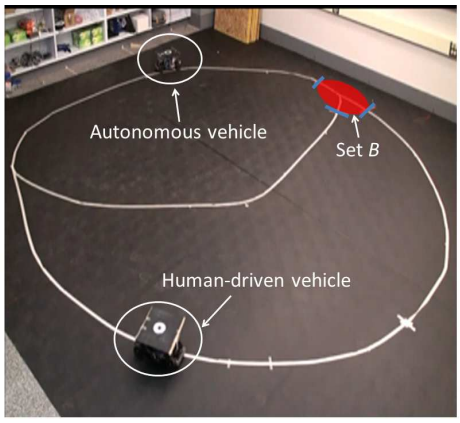


(j) Human and autonomous vehicle speed.

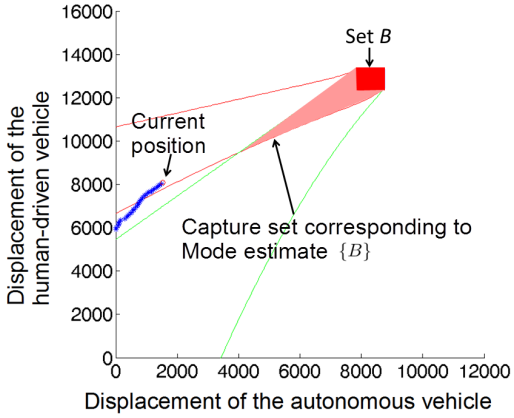
Figure 4.8: Sub-figures (a), (c), (e) and (g) show the displacement of autonomous and human-driven vehicles along their paths on the x-axis and y-axis respectively (sub-figures (b), (d), (f) and (h) show the corresponding snapshots from the experiment). The slice of the current mode-dependent capture set, corresponding to the current velocity of the two vehicles, is shown as the area shaded in red. In the case when the hidden mode is not known, both braking and acceleration are taken as possible modes resulting in a larger capture set (sub-figure (a)). With more data, the estimator is able to identify that the human-driven vehicle is in braking mode and thus the capture set shrinks (sub-figure (c)). The control input is applied in sub-figure (e) since the predicted state (denoted by red circles) enters the capture set. The applied control keeps the two vehicles from entering the bad set at the same time and thus prevents collision, as shown in sub-figure (g). Sub-figures (i) and (j) show the control input and velocity for the two vehicles, respectively.



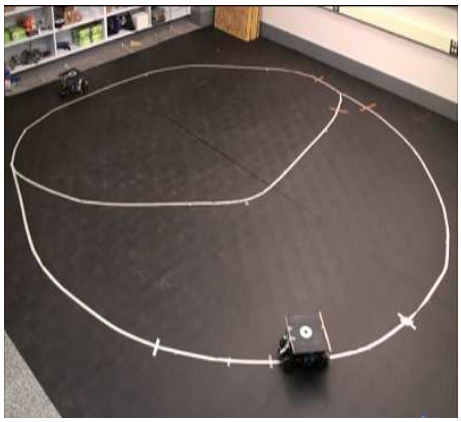
(a) Unknown mode, i.e., $\hat{q} = \{A, B\}$, time = 237.8 seconds.



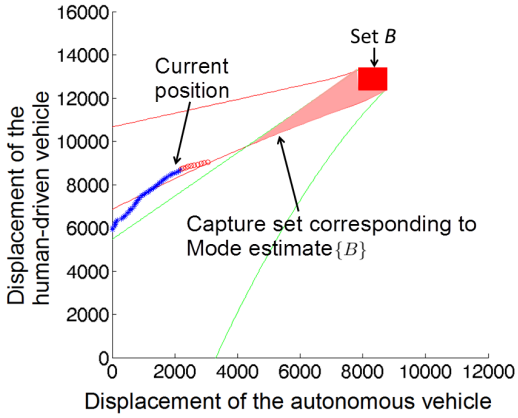
(b) Lab snapshot corresponding to (a).



(c) Mode identified as braking, i.e., $\hat{q} = \{B\}$, time = 241.1 seconds.

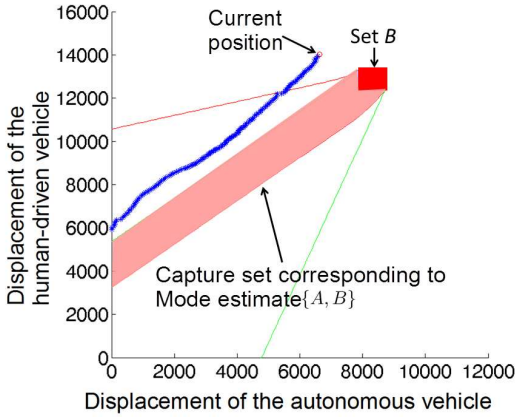


(d) Lab snapshot corresponding to (c).

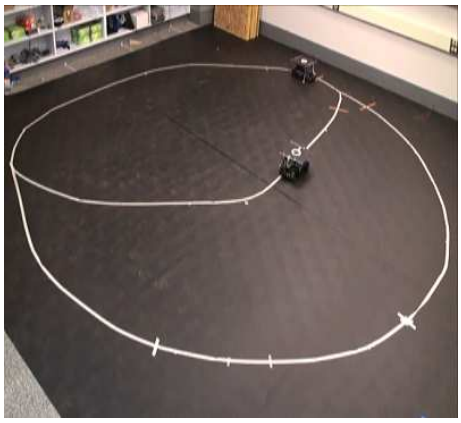


(e) Autonomous vehicle applies control, time = 242.4 seconds.

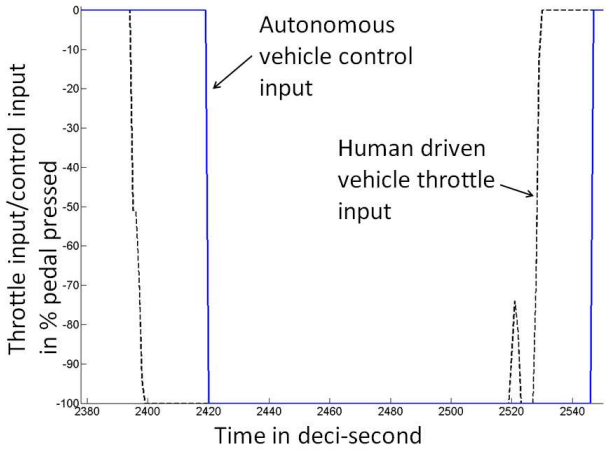
(f) Lab snapshot corresponding to (e).



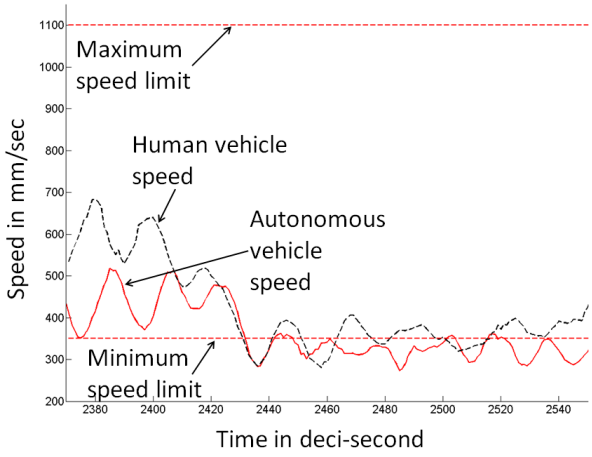
(g) Collision situation averted, time = 254.6 seconds.



(h) Lab snapshot corresponding to (g).

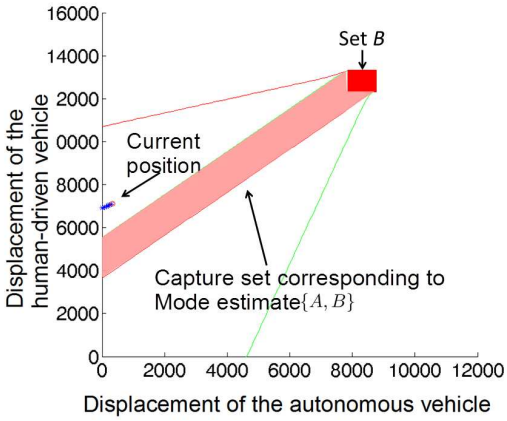


(i) Throttle input of human and autonomous vehicle.

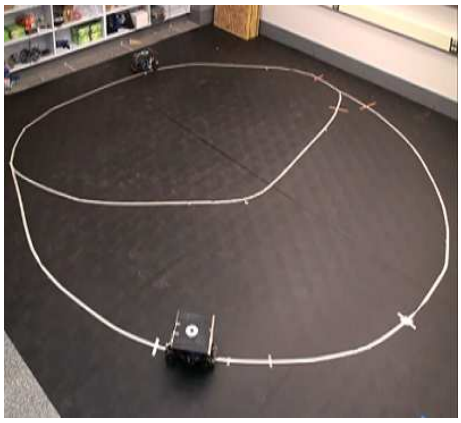


(j) Human and autonomous vehicle speed.

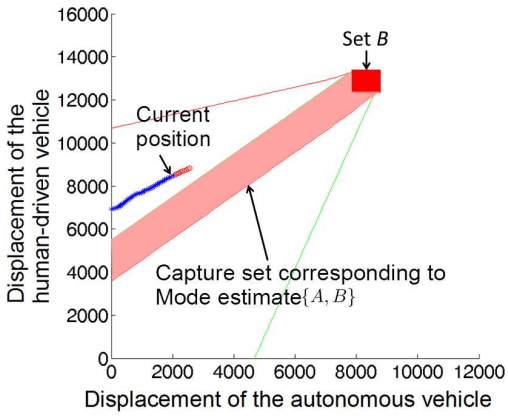
Figure 4.9: Both braking and acceleration are initially taken as possible modes resulting in a larger capture set (sub-figure (a)). With more data, the estimator is able to identify that the human-driven vehicle is in acceleration mode and thus the capture set shrinks (sub-figure (c)). The control input is applied in sub-figure (e) since the predicted state (denoted by red circles) enters the capture set. The applied control keeps the two vehicles from entering the bad set at the same time and thus prevents collision, as shown in sub-figure (g). Sub-figures (i) and (j) shows the velocity and control input for the two vehicles.



(a) Unknown mode, i.e., $\hat{q} = \{A, B\}$, time = 141.4 seconds.

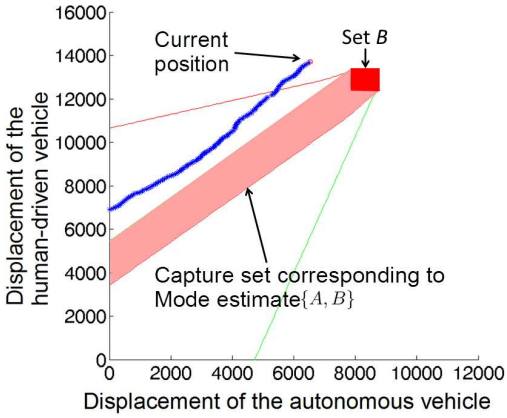


(b) Lab snapshot corresponding to (a).

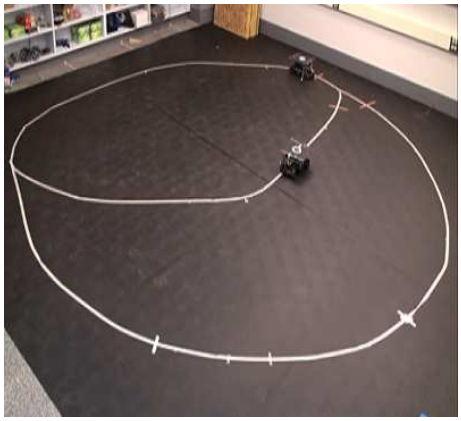


(c) Autonomous vehicle applies control with mode still unknown, i.e., $\hat{q} = \{A, B\}$, time = 145.1 seconds.

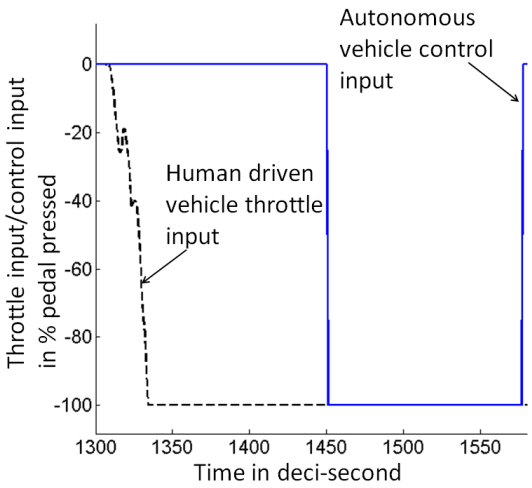
(d) Lab snapshot corresponding to (c).



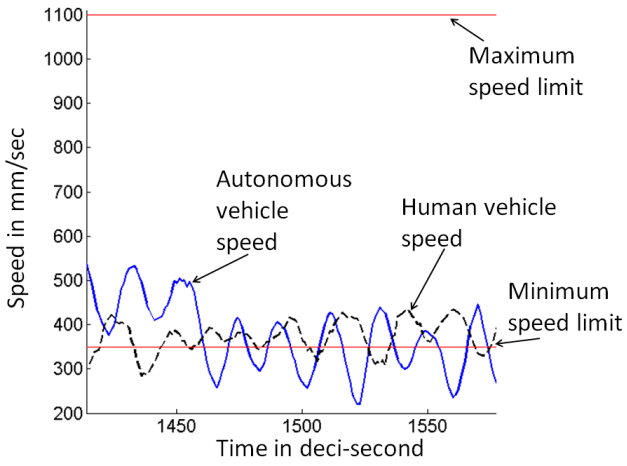
(e) Collision averted, time = 157.1 seconds.



(f) Lab snapshot corresponding to (e).



(g) Throttle input of human and autonomous vehicle.



(h) Human and autonomous vehicle speed.

Figure 4.10: Both braking and acceleration are initially taken as possible modes resulting in a larger capture set (sub-figure (a)). However, in this case even with more data, the estimator is unable to identify the mode and the capture set does not shrink (sub-figure (c)). The control input is applied in sub-figure (c) since the predicted state (denoted by red circles) enters the capture set. The applied control keeps the two vehicles from entering the bad set at the same time and thus prevents collision, as shown in sub-figure (e). Sub-figures (g) and (h) shows the velocity and control input for the two vehicles.

4.4 Acknowledgements

I am grateful to Vishnu Desaraju, Matt McCullough, Dan David Clark and many other undergraduate students who helped me with constructing the hardware, putting the software framework together and the experimentation. I also thank the human subjects for their time and patience while performing the experiments. This work could not be complete without their help.

CHAPTER V

Conclusions and future work

This thesis focuses on the safety control problem for hybrid systems in which the mode is not available for control (HMHS). There are three main parts of this thesis. In the first part (Chapter 2), we addressed the above problem by adopting an approach inspired by the theory of games with imperfect information. Specifically, we introduced the notion of non-deterministic discrete information state and formulated the control problem on its basis (Problem 1). We defined the notion of an estimator and we formulated a control problem with perfect state information on a new hybrid automaton $\hat{H}$ (Problem 2). We provided an algorithm for the computation of the capture set for $\hat{H}$ and for the least restrictive control map. We provided conditions for the termination of the iterative algorithm that computes the capture set. We also showed how to construct an abstraction of $\hat{H}$ for which the algorithm always terminates and has as fixed point the capture set of $\hat{H}$. We showed that Problem 2 is equivalent to Problem 1 under suitable assumptions and provided classes of systems for which these assumptions are satisfied. Accordingly, an application example in the context of cooperative active safety systems has been presented. Future research will include removing Assumptions 1 and 2 by employing a dynamic feedback design that does not impose separation between estimation and control.

The second part (Chapter 3) focuses on the development of an in-scaled vehicle which is dynamically similar to a HMMWV. We introduced the models of various subsystems of

the full scale vehicle and carried out the scaled vehicle design. Implementation on a scaled RC car is performed. Finally, we conducted experiments that demonstrate the dynamic similitude of the scaled vehicle to full scale vehicle.

In the third part (Chapter 4) of this thesis, we illustrated the application of the formal hybrid control approach, developed in the first part of this thesis, to design semi-autonomous multi-vehicle systems that are guaranteed to be safe. Specifically, we illustrated our results on a roundabout vehicle test-bed involving an autonomous and a human-driven in-scale vehicle. The autonomous vehicle estimates the driving mode of the human-driven vehicle and selects estimate-dependent collision avoidance maneuvers to ensure safety. Our experimental results illustrate that in a structured task, such as driving, simple human decision models can be effectively learned and employed in a feedback control system that enforces a safety specification. They also highlight how the incorporation of these models in a safety control system makes the control actions required for safety less conservative. In fact, by virtue of the mode estimate, the current (mode dependent) capture set to avoid to guarantee safety is considerably smaller than the capture set to be avoided when the mode estimate is not available. This is essential for the practical applicability of cooperative active safety systems. On our data set, the flow entered the capture set only 3% of the times. These failures are due mainly to the fact that communication delays between the vehicle and the workstation are not formally accounted for by the algorithm. These delays, when significant, cause the calculated capture set to be different from the actual one and hence may cause to enforce control too late. These delays, in future work, should be formally accounted for in the models and in the safety control algorithm.

We assumed that the decision point D^P is a fixed distance away from the intersection. However, the point D^P may be selected based on studying human-driver acceleration and braking behavior near the intersection. This will be studied in the future work.

More complex models of human decisions in the proximity of an intersection, such as those studied in [92], and the incorporation of additional details such as weather condi-

tions and road geometry, offer the potential for reducing the conservatism of safe control actions even further. Therefore, further work will explore the refinement of these models by incorporating additional domain-specific knowledge. The models here considered are deterministic because most of the tools currently available to perform safety control have assumed deterministic models, wherein uncertainty is bounded. However, human decision models are more naturally captured by stochastic frameworks, in which uncertainty due to variability in both subjects and realizations of the same decision is probabilistic (see [71] for a review on the topic). As results in stochastic safety verification and design become available [5, 7], it will be important to extend the proposed techniques of this paper to safety control of stochastic hybrid automata, in which the mode estimate is constructed probabilistically as opposed to being constructed deterministically.

Finally, in any real-life implementation of cooperative active safety systems, the algorithms implemented by the autonomous vehicle should be capable of interacting with a human driver. That is, they should first warn the driver and suggest actions, and take control of the vehicle only when the driver is incapable of preventing a collision. Hence, future work will consider the incorporation of human response time to warnings in the algorithms and the problem of establishing when it is absolutely necessary to override a human driver for maintaining safety.

APPENDICES

APPENDIX A

Buckingham π theorem and an example

A.1 Buckingham π Theorem

Theorem A.1 (F. M. White Fluid Mechanics Sections 5.3)**.** (Buckingham π theorem) *If a physical process satisfies the principle of dimensional homogeneity and involves n dimensional variables, it can be reduced to a relation between only k dimensionless variables or π's. The reduction $j = n - k$ equals the maximum number of variables which do not form a π among themselves and is always less than or equal to the number of dimensions describing the variables.*

Let $q_1, q_2, ...q_n$ be n dimensional variables that are physically relevant in a given problem and that are inter-related by an (unknown or known) dimensionally homogeneous set of equations. These can be expressed via a functional relationship of the form:

$$F(q_1, q_2, ...q_n) = 0$$

If k is the number of fundamental quantities required to describe the n variables, then there will be k primary variables and the remaining $j = (n - k)$ variables can be expressed as (n –

k) dimensionless and independent quantities or 'Pi groups', $\pi_1, \pi_2, ...\pi_{n-k}$. The functional relationship can thus be reduced to the much more compact form:

$$\Phi(\pi_1, \pi_2, ...\pi_{n-k}) = 0$$

Note that this set of non-dimensional parameters is not unique. They are however independent [93]. Here, independence means that one π group can be varied while keeping other groups constant. The fundamental quantities of a system consist of the minimum number of unit dimensions needed to describe each parameter. For example, the units of measure for acceleration are length unit/(time unit)^2. The fundamental quantities most often used are Mass, Length, Time, Temperature, Current, Amount of substance, Luminous intensity. Two differently-sized physical systems, with different dimensional parameters, can be reduced to the same dimensionless description if the corresponding π parameters have the same numerical values. An example with application of π theorem is given in the next section.

A.1.1 An Example: The Simple Pendulum

Consider a simple pendulum, which is a mass m on the end of a massless rod of length l. We wish to investigate what quantities may affect the period τ of this pendulum. In addition to the mass and length of the pendulum, the acceleration due to gravity g, the initial angle θ_0 and the tension in the rod T may have an effect on the period. We ignore elasticity in the rod for this simple example. Thus we have,

$$\tau = F(m, l, g, \theta_0).$$

The next step is to identify for each quantity its dimensions in terms of the fundamental dimensions appropriate for the problem. Each quantity Z will have its dimensions, $[Z]$,

written as a product of powers of the fundamental dimensions, $L_1, L_2, ..., L_k$:

$$[Z] = {L_1}^{\theta_1}{L_2}^{\theta_2}...{L_k}^{\theta_k}.$$

In the case of the simple pendulum, the fundamental dimensions are length $L_1 = L$, mass $L_2 = M$, and time $L_3 = T$. In terms of these fundamental dimensions, each quantity has the following dimensions:

$$[\tau] = [M^0L^0T^1], [m] = [M^1L^0T^0], [l] = [M^0L^1T^0]$$

$$[g] = [M^0L^1T^{-2}], [\theta_0] = [M^0L^0T^0].$$

The number of fundamental quantities is 3, $n = 5$ and $m = 3$. Thus we have 3 fundamental quantities and 2 dimensionless, independent π groups. The angle θ_0 is a dimensionless quantity and its dimension can be denoted by 1. The first π group is $\pi_1 = \theta_0$. π groups are not unique. Different π groups can result based on the selection of repeating variables. Repeating variables appear in more than one π group. Let us consider m, l, g as the repeating variables.

$$[\tau][m^a][l^b][g^c] = [MLT]^0.$$

This means that

$$[T]^1[M]^a[L]^b([L][T]^{-2})^c = [MLT]^0.$$

The dimensions of T, L and M should match on both sides of this relationship. Equating exponents on both sides leads to the following set of linear equations in the three unknowns a, b, and c:

$$a = 0$$

$$b + c = 0$$

$$2c = 1.$$

Solving the above linear equations, we obtain $a = 0$, $b = -1/2$, and $c = 1/2$ Thus, the second π group is given by, $\pi_2 = \tau\sqrt{\frac{g}{l}}$. The functional relationship of this system can be reduced to the form:

$$\Phi(\pi_1, \pi_2) = 0.$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] U.S. DOT Joint Program Office ITS. http://www.its.dot.gov.

[2] webpage : http://www.tamiyausa.com/.

[3] webpage : www.acroname.com/brainstem/tea/tea1.html.

[4] webpage : www.acroname.com/robotics/parts/s10-moto-brd.html.

[5] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16:624641, December 2010.

[6] T. Akita, S. Inagaki, T. Suzuki, S. Hayakawa, and N. Tsuchida. Hybrid system modeling of human driver in the vehicle following task. In *SICE, 2007 Annual Conference*, pages 1122 –1127, 2007.

[7] Saurabh Amin, Ro Abate, Maria Pr, John Lygeros, and Shankar Sastry. Reachability analysis for controlled discrete time stochastic hybrid systems. In *in Hybrid Systems: Computation and Control - HSCC 2006*, pages 49–63. Springer Verlag, 2006.

[8] D. Angeli and E.D. Sontag. Interconnections of monotone systems with steady-state characteristics. *Optimal control, stabilization and nonsmooth analysis. Lecture Notes in Control and Inform. Sci. Springer*, 301:135–154, 2004.

[9] D. Angeli and E.D. Sontag. Oscillations in I/O monotone systems. *IEEE Transactions on Circuits and Systems*, 55:166–176, 2008.

[10] J. Aubin. *Viability Theory*. Birkhäuser, 1991.

[11] J. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube. Impulse differential inclusions: A viability approach to hybrid systems. *IEEE Transactions on Automatic Control*, 47:2–20, 2002.

[12] R. J. Aumann and M. Maschler. *Repeated Games with Incomplete Information*. MIT Press, 1995.

[13] M. Baglietto, G. Battistelli, and L. Scardovi. Active mode observability of switching linear systems. *Automatica*, 43:1442–1449, 2007.

[14] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, and A. Sangiovanni-Vincentelli. Design of observers for hybrid systems. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science vol. 2289, C. J. Tomlin and M. R. Greensreet (Eds.), Springer Verlag, pages 76–89, 2002.

[15] M. G. Bekker. *Introduction to TerrainVehicle Systems*. The University of Michigan Press, Ann Arbor, 1959.

[16] R. Bishop. *Intelligent Vehicle Technology And Trends*. Artech House, Inc., Norwood, MA, 2005.

[17] L. Blackmore, S. Rajamanoharan, and B. C. Williams. Active estimation for switching linear dynamic systems. In *Conf. on Decision and Control*, pages 137–144, 2006.

[18] S. Brennan and A. Alleyne. Driver assisted yaw rate control. In *American Control Conference*, pages 1697–1704, 1999.

[19] S. Brennan and A. Alleyne. A scaled testbed for vehicle control: the IRS. In *International Conference on Control Applications*, pages 327 – 332, 1999.

[20] S. Brennan and A. Alleyne. The Illinois roadway simulator: A mechatronic testbed for vehicle dynamics and control. *IEEE/ASME Trans. on Mech. Sys.*, 5(4):349–359, 2000.

[21] S. Brennan, A. Alleyne, and M. DePoorter. The Illinois roadway simulator - a hardware-in-the-loop testbed for vehicle dynamics and control. In *American Control Conference*, pages 493 – 497, 1998.

[22] S. N. Brennan. Modeling and control issues associated with scaled vehicles. Master's thesis, University of Illinois at Urbana Champaign, 1999.

[23] S. N. Brennan. *On size and control: the use of dimensional analysis in controller design*. PhD thesis, University of Illinois at Urbana Champaign, 2002.

[24] E. Buckingham. On physically similar systems: Illustration of the use of dimensional equations. *Physics Review*, (4):345 – 376, 1914.

[25] S. R. Burns, Jr. R. T. O'Brien, and J. A. Piepmeier. Steering controller design using scale-model vehicles. *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory*, pages 476–478, 2002.

[26] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley, New York, 1983.

[27] R. D'Andrea, R. M. Murray, J. A. Adams, A. T. Hayes, M. Campbell, and A. Chaudry. The RoboFlag Game. In *American Control Conference*, pages 661–666, 2003.

[28] B. A. Davey and H. A. Priesteley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.

[29] D. Del Vecchio. A partial order approach to discrete dynamic feedback in a class of hybrid systems. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, vol. 4416, A. Bemporad, A. Bicchi, and G. Buttazzo (Eds.), Springer Verlag, pages 159–173, Pisa, Italy, 2007.

[30] D. Del Vecchio. Cascade estimators for systems on a partial order. *Systems and Control Letters*, 57(10):842–850, 2008.

[31] D. Del Vecchio. Observer-based control of block triangular discrete time hybrid automata on a partial order. *International Journal of Robust and Nonlinear Control*, 19(14):1581–1602, 2009.

[32] D. Del Vecchio and E. Klavins. Observation of guarded command programs. In *Conf. on Decision and Control*, pages 3353–3359, 2003.

[33] D. Del Vecchio, M. Malisoff, and R. Verma. A separation principle for a class of hybrid automata on a partial order. In *American Control Conference*, 2009.

[34] D. Del Vecchio, R. M. Murray, and E. Klavins. Discrete state estimators for systems on a lattice. *Automatica*, 42(2):271–285, 2006.

[35] D. Del Vecchio, R. M. Murray, and P. Perona. Primitives for human motion: A dynamical approach. In *IFAC World Congress*, Barcelona, 2002.

[36] D. Del Vecchio, R. M. Murray, and P. Perona. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. *Automatica*, 39(12):2085–2098, 2003.

[37] Y. Demiris. Prediction of intent in robotics and multi-agent systems. *Cognitive Processes*, 8:151–158, 2007.

[38] M. DePoorter, S. Brennan, and A. Alleyne. Driver assisted control strategies: Theory and experiment. In *Proc. ASME Int. Mech. Eng. Congr. Expo.*, pages 721–728, 1998.

[39] V. Desaraju, M H. C. Ro, E. Tay Yang, S. Roth, and D. Del Vecchio. Partial order techniques for vehicle collision avoidance: Application to an autonomous roundabout test-bed. In *Proc. of International Conference on Robotics and Automation*, pages 82–87, 2009.

[40] L. Huberand O. Dietz. Pendelbewegungen von lastkraftwagenanhangern und iner vereidung (snaking of truck trailers). *VDI-zeit schrift*, 81(16):459–463, 1937.

[41] E. A. Domlan, J. Ragot, and D. Maquin. Active mode estimation for switching systems. In *acc*, 2007.

[42] J. Duperret, M. Hafner, and D. Del Vecchio. Formal design of a provably safe robotic roundabout system. In *Proc. of International Conference on Intelligent Robots and Systems*, pages 2006–2011, 2010.

[43] O. Maler E. Asarin and A.Pnueli. Symbolic controller synthesis for discrete and timed systems. In *Hybrid Systems II*, Lecture Notes in Computer Science, vol. 999, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry (Eds.), Springer Verlag, pages 1–20, 1995.

[44] R. I. Emori. Automobile accident reconstruction. Technical report, UCLA Motor Vehicle Safety Contract Final Report, UCLA Schools of Engineering and Medicine, 1969.

[45] H. Fathy, R. Ahlawat, and J. L. Stein. Proper powertrain modeling for engine-in-the-loop-simulation. In *Proceedings of the ASME International Engineering Congress and Exposition, number IMECE2005-81592, 2006*, volume 74, pages 1195–1201, New York.

[46] Z. Filipi, H. Fathy, J. R. Hagena, A. Knafl, R. Ahlawat, J. Liu are D. Jung, D. N. Assanis, H. Peng, and J. Stein. Engine-in-the-loop testing for evaluating hybrid propulsion concepts and transient emissions - HMMWV case study. In *SAE 2006 World Congress and Exhibition, Detroit, MI, USA*, 2006.

[47] Y. Gao, J. Lygeros, and M. Quincampoix. The reachability problem for uncertain hybrid systems revisited: A viability theory perspective. In *Lecture Notes in Computer Science LNCS, no. 3927*, pages 242–256, 2006.

[48] T. D. Gillespie. *Fundamentals of Vehicle Dynamics*. SAE International, March 1992.

[49] A. Gilpin and T. Sandholm. Solving two-person zero-sum repeated games of incomplete information. In *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems*, 2008.

[50] M. Hafner and D. Del Vecchio. Computation of safety control for uncertain piecewise continuous systems on a partial order. In *Conference on Decision and Control*, 2009.

[51] M. Hafner and D. Del Vecchio. Computation of safety control for uncertain piecewise continuous systems on a partial order. In *Conference on Decision and Control*, 2009.

[52] T. A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE press, 1996.

[53] T. A. Henzinger, P. H. Ho, and H. Wong-Toi. A user guide to HyTech. In *TACAS 95: Tools and Algorithms for the construction and analysis of systems*, Lecture Notes in Computer Science, vol. 1019, E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen (Eds.), Springer-Verlag, pages 41–71, 1995.

[54] T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HyTech: Hybrid systems analysis using interval numerical methods. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, vol. 1790, B. Krogh and N. Lynch (Eds.), Springer Verlag, pages 130–144, 2000.

[55] R. J. Hill. Electric railway traction tutorial, part 1: Electric traction and dc traction motor drives. *Power Engineering Journal*, (1):47–56, 1994.

[56] P. Hoblet, Jr. R. T. OBrien, and J. A. Piepmeier. Scale-model vehicle analysis for the design of a steering controller. In *the 35th Southeastern Symposium on System Theory*, pages 201–205, 2003.

[57] J. Hu, M. Prandini, and S. Sastry. Aircraft conflict prediction in the presence of a spatially correlated wind field. *IEEE Transactions on Intelligent Transportation Systems*, (3):326–340, 2005.

[58] Intelligent Vehicle Initiative. Final report. *Available at http://www.itsdocs.fhwa.dot.gov/JPODOCS/ REPTS_PR/14153_files/ivi.pdf*, 2005.

[59] S. Istrail. Generalization of the Ginsburg-Rice Schützenberger fixed-point theorem for context-sensitive and recursive-enumerable languages. *Theoretical Computer Science*, 18:333–341, 1982.

[60] R.T. O'Brien Jr., J.A. Piepmeier, P.C. Hoblet, S.R. Burns, and C.E. George. Scale-model vehicle analysis using an off-the-shelf scale-model testing apparatus. In *American Control Conference*, pages 3387–3392, 2004.

[61] U. Kiencke and L. Nielsen. *Automotive Control Systems, For Engine, Driveline, and Vehicle*. Springer Verlag, 2nd edition, 2005.

[62] J.-H. Kim, Y.-W. Kim, and D.-H. Hwang. Modeling of human driving behavior based on piecewise linear model. *AUTOMATIKA*, 46:29–37, 2005.

[63] H. W. Kuhn. Extensive games and the problem of information. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*. Princeton University Press, pages 196–216, 1953.

[64] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for hybrid dynamics: the reachability problem. In *New Directions and Applications in Control Theory*, Lecture Notes in Control and Information Sciences, vol 321, W.P. Dayawansa, A. Lindquist, and Y. Zhou (Eds.), pages 193–205, 2005.

[65] K. Laberteaux, L. Caminiti, D. Caveney, and H. Hada. Pervasive vehicular networks for safety. *IEEE Pervasive Computing, Spotlight*, pages 60–62, 2006.

[66] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 1st edition, 2006.

[67] R. G. Longoria, A. Al-Sharif, and C. Patil. Scaled vehicle system dynamics and control: A case study on anti-lock braking. *International Journal of Vehicle Autonomous Systems*, 2(1-2):18–39, 2004.

[68] J. Lygeros, C. J. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[69] D. Lynch and A. Alleyne. Velocity scheduled driver assisted control. *Int. J. Veh. Des.*, 29:1–22, Aug 2003.

[70] A. Mathews. Implementation of a fuzzy rule based algorithm for adaptive antilock braking in a scaled car. Master's thesis, University of Texas at Austin, 2002.

[71] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[72] US Department of Transportation Federal Highway Administration. Road safet fact sheet. *http://safety.fhwa.dot.gov/facts/road_factsheet.htm*, January 2007.

[73] M. Oishi, I. Mitchell, A. Bayen, and C. Tomlin. Invariancepreserving abstractions of hybrid systems: Application to user interface design. *IEEE Transactions on Control Systems Technology*, 16(2):229–244, 2008.

[74] J. Pachl. *Railway operation and control*. VTD Rail Publishing, 2002.

[75] C. B. Patil, R. G. Longoria, and J. Limroth. Control prototyping for an anti-lock braking control system on a scaled vehicle. In *Conference on Decision and Control*, pages 4962–4967, 2003.

[76] Chinmaya Patil. Anti-lock brake system re-design and control prototyping using a one-fifth scale vehicle experimental test-bed. Master's thesis, University of Texas at Austin, 2003.

[77] M. Polley, A. Alleyne, and E. De Vries. Scaled vehicle tire characteristics: dimensionless analysis. *Vehicle Systems Dynamics*, 44(2):87–105, February 2006.

[78] Matthew Polley. Size effects on steady state pneumatic tire behavior: An experimental study. Master's thesis, University of Illinois at Urbana Champaign, 2001.

[79] R. Raffard, S. Waslander, A. Bayen, and C. Tomlin. A cooperative distributed approach to multi-agent eulerian network control: Application to air traffic management. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[80] J. H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer System Sciences*, 29(2):274–301, 1984.

[81] E. De Santis, M. D. Di Benedetto, and L. Berardi. Computation of maximal safe sets for switching systems. *IEEE Trans. Automatic Control*, 49(2):184–195, 2004.

[82] C.-E. Seah and I. Hwang. Terminal-area aircraft tracking by hybrid estimation. *AIAA Journal of Guidance, Control and Dynamics*, 32(3):836–849, 2009.

[83] O. Shakernia, G. J. Pappas, and S. Sastry. Decidable controller synthesis for classes of linear systems. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, vol. 1790, Springer Verlag, 2000.

[84] O. Shakernia, G. J. Pappas, and S. Sastry. Semidecidable controller synthesis for classes of linear hybrid systems. In *Proc. of Conf. on Decision and Control*, 2000.

[85] O. Shakernia, G. J. Pappas, and Shankar Sastry. Semi-decidable synthesis for triangular hybrid systems. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, vol. 2034, M. D. Di Benedetto and A. Sangiovanni-Vincentelli (Eds.), Springer Verlag, 2001.

[86] T. Suzuki. Advanced motion as a hybrid system. In *Electronics and Communications in Japan*, 2010.

[87] C. J. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, 2000.

[88] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

[89] W. E. Travis, R. J. Whitehead, D. M. Bevly, and G. T. Flowers. Using scaled vehicles to investigate the influence of various properties on rollover propensity. In *American Control Conference*, pages 3381–3386, 2004.

[90] R. Verma, D. Del Vecchio, and H. Fathy. Development of a scaled vehicle with longitudinal dynamics of a HMMWV for an ITS testbed. *IEEE/ASME Transactions on Mechatronics*, 13:46–57, 2008.

[91] R. Vidal, A. Chiuso, and S. Soatto. Observability and identifiability of jump linear systems. In *Conf. on Decision and Control*, pages 3614 – 3619, 2002.

[92] C. Walton, D. Del Vecchio, and R. M. Murray. Identification of decision rules in a human-controlled system: Vehicles at a traffic intersection. In *International Conference on Robotics and Automation*, pages 1173–1178, New Orleans, 2004.

[93] Frank M. White. *Fluid Mechanics*. McGraw-Hill Book Company, 1989.

[94] R. J. Whitehead, D. M. Bevly, and B. Clark. Esc effectiveness during property variations on scaled vehicles. In *ASME IMECE*, pages 233–242, November 2005.

[95] M. De Wulf, L. Doyen, and J.-F. Raskin. A lattice theory for solving games of imperfect information. *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, vol. 3927, J. Hespanha and A. Tiwari (Eds.), Springer-Verlag, pages 153–168, 2006.

[96] Ya. H. Zakin. Pritchine vozniknovenya pritsepov (the cause of instability of trailers). *Avtom. Promyshlennost (Russian Automotive Industry Journal)*, 11, 1959.