Formal Design of a Provably Safe Robotic Roundabout System

Jeffrey M. Duperret, Michael R. Hafner and D. Del Vecchio

***Abstract*— In this paper, we show how to design a provably safe robotic roundabout system comprised of three vehicles. This is accomplished by combining two-vehicle collision avoidance primitives, which are each computationally light given the natural partial order structure on which the system evolves. We show how to design the system parameters in order to prevent conflicts among the control primitives, and to thus ensure the safety and liveness of the system as a whole. We implement our design on a multi-vehicle test-bed involving three vehicles continuously running on three intersecting roundabouts, and provide experimental results demonstrating the system is collision free and live.**

I. Introduction

The development of Intelligent Transportation Systems (ITS) for cooperative active safety is a rapidly progressing area of research and joint initiatives in government, academia, and industry [1] [2] [3]. This research has been driven by recent advancements in vehicle sensor suits, vehicle-to-vehicle (V2V) wireless communication, vehicle-to-infrastructure (V2I) wireless communication, and on-board computational resources.

The employment of a formal hybrid modeling and control approach has been previously applied in the development of automated highway systems (AHS) by the California PATH project in the 90s. The objective of the AHS project was the employment of fully autonomous highway systems, mainly based on the concept of platooning, to increase traffic throughput, safety, and fuel efficiency [12], [13], [18], [22]. In the context of platooning, a number of papers and PATH reports have proposed a formal hybrid modeling and control approach based on the computation of the safe set of initial conditions (the complement of the static capture set), on optimal control, and on game theory [5]–[7], [10], [14]–[16]. There is a wealth of work on the safety control of hybrid systems (see for example [17], [19], [20], and the references therein). The safety control problem consists of preventing the state from entering a bad set, usually representing all possible collisions. This problem can be addressed by computing the set of states that flow into the bad set independently of an input choice. Then, a feedback signal is computed that guarantees that the state never enters such a set. As it appears from these previous works, computational constraints usually limit the system size to low dimensional state spaces.

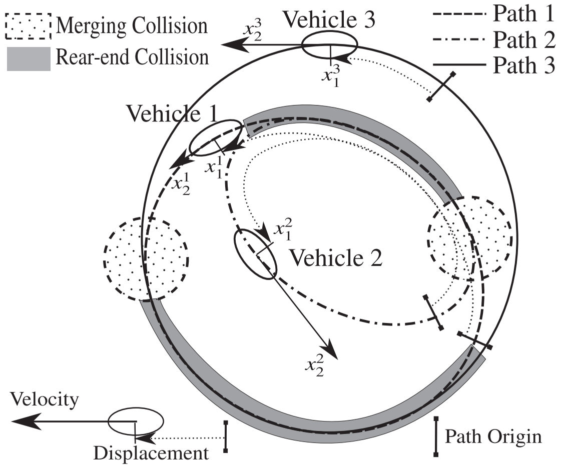

Fig. 1. Roundabout system with locations of potential collisions superimposed on the paths, and vehicle states shown along with path origins.

In this work, we solve the collision avoidance problem for a three-vehicle roundabout system (Figure 1). This problem is motivated by single lane modern roundabouts with multiple access points. Roundabouts are becoming common due to their impact on fuel economy and added safety benefits [4], [21]. To overcome the previously mentioned limitations in computational complexity, we exploit the fact that traffic systems can be well modeled by order preserving systems. Previous work involved exploiting the order preserving properties of the system dynamics to solve the two-vehicle collision avoidance problem with control primitives based on linear complexity algorithms [8], [11], [24], and implementing these primitives on a two-vehicle test-bed [9]. We combine a number of these control primitives by conjunction, where each primitive prevents two specific vehicles from colliding within a section of the roundabout without allowing any vehicle to come to a stop. We show the conjunction is non-conflicting and thus guarantees the three-vehicle roundabout system under consideration is collision free and live.

This paper is organized as follows. In Section II, we provide a description of the roundabout system, the models used for each vehicle, a formal description of the safety specification, and the characterization of the feedback maps used to guarantee that each safety specification is met. In Section III, we provide an experimental setup to implement the algorithms, the online control algorithm, and our experimental results.

Jeffrey M. Duperret and Michael R. Hafner are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. E-mail: jduperre,mikehaf@umich.edu

D. Del Vecchio is with the Department of Mechanical Engineering, MIT. E-mail: ddv@mit.edu

This work was supported in part by NSF-CAREER Award CNS-0642719 and in part by NSF-GOALI Award CMMI-0854907.

II. System Description and Safety Control Design

We first provide the physical model used to describe the dynamics of each vehicle. We next provide a mathematical abstraction of this model which allows us to state the inherent order preserving properties of the dynamics. The safety problem is defined for the specific roundabout system considered. Lastly, the control primitives used are provided.

A. Longitudinal Dynamics Model

We consider the continuous evolution of three robotic vehicles constrained to forward motion along prescribed paths, as dictated by the roundabout system shown in Figure 1. We assume a low level controller keeps each vehicle on its path by actuating the steering angle. We can thus completely describe the state of the system by a displacement x_1 along its path, and a longitudinal velocity x_2. The longitudinal dynamics along the path can be written as $\dot{x}_2 = [\mathcal{R}^2/(J_w + \mathcal{M}\mathcal{R}^2)](f_w - f_{brake} - \frac{\rho_{air}}{2} C_D A_f(v)^2 - C_{rr}\mathcal{M}g)$, in which $\mathcal{R}$ is the tire radius, J_w is the wheel inertia, $\mathcal{M}$ is the mass of the vehicle, $f_w = \tau_w \mathcal{R}$ where τ_w is the drive shaft output torque, f_{brake} is the brake force, ρ_{air} is the air density, C_D is the drag coefficient, A_f is the frontal area of the vehicle, C_{rr} is the rolling resistance, and g is the gravitational constant. For more details of this model, the reader is referred to [23] and the references therein. Assuming that the air drag is negligible, we can re-write the longitudinal dynamics as $\dot{x}_2 = au + b$, where $u = f_w - f_{brake}$, $a = \mathcal{R}^2/(J_w + \mathcal{M}\mathcal{R}^2)$ and $b = -\mathcal{R}^2/(J_w + \mathcal{M}\mathcal{R}^2)C_{rr}\mathcal{M}g$.

B. Order Preserving System Description

We next introduce a mathematical abstraction of the longitudinal vehicle model needed to describe the order preserving properties of the system. For each vehicle $i \in \{1, 2, 3\}$, define the tuple $\Sigma^i := (X^i, U^i, f^i)$, in which X^i defines the state space, U^i defines the input set, and $f^i : X^i \times U^i \to X^i$ defines a piecewise continuous vector field with input.

The state space for each vehicle is given by $X^i := [0, D^i) \times [v^i_{min}, v^i_{max}]$, where $[0, D^i)$ represents longitudinal displacements along the respective path from the origin, and $[v^i_{min}, v^i_{max}]$ represents longitudinal velocity. For notation, let $X^i_1 = [0, D^i)$ and $X^i_2 = [v^i_{min}, v^i_{max}]$. We denote the state of the i^{th} vehicle as $x^i(t)$, and $x^i_j(t)$ as the j^{th} component of the state, where $j = 1$ corresponds to position and $j = 2$ corresponds to velocity. Since each vehicle can loop around, when $x^i_1(t) = D^i$, the state is reset back to $x^i_1(t) = 0$.

The input for vehicle i lies within the set $U^i := [u^i_{min}, u^i_{max}]$, where u^i_{min} is the maximum braking input and u^i_{max} is the maximum throttle input. We define the set of piecewise continuous signals on U as $S(U)$. We will commonly denote signals in boldface, that is, $\mathbf{u} \in S(U)$.

For each vehicle $i \in \{1, 2, 3\}$, the vector field is represented as a hybrid automaton, as shown in Figure 2, where a^i and b^i are vehicle parameters. The hybrid automaton is used to guarantee that $x^i_2 \geq v^i_{min} > 0$ and $x^i_2 \leq v^i_{max}$, thus enforcing the liveness condition and speed limit. The entire system is generated by taking the parallel composition of the three vehicles, that is, $\Sigma := \Sigma^1 \| \Sigma^2 \| \Sigma^3$ [8]. Let $\phi(t, x, \mathbf{u})$ denote the flow of the entire system at time t with initial condition x and input $\mathbf{u}$. We denote $\phi^i_j(t, x, \mathbf{u})$ as the j^{th} component of the i^{th} vehicle flow.

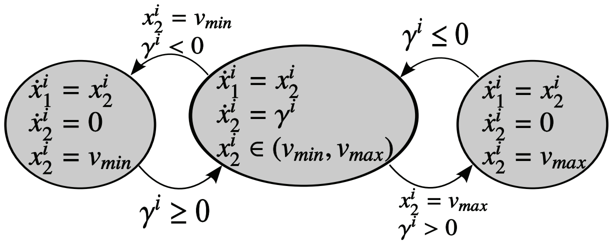

Fig. 2. Hybrid system modeling the vehicle dynamics. In the diagram, we have defined $\gamma^i := a^i\mathbf{u}^i + b^i$ for vehicle $i \in \{1, 2, 3\}$

We consider the partial order $(X, \leq)$ defined by component wise ordering, and the partial order $(S(U), \leq)$, where we say $\mathbf{u}_1 \leq \mathbf{u}_2$ provided $\mathbf{u}_1(t) \leq \mathbf{u}_2(t)$ for all $t \in \mathbb{R}_+$. It can be shown that the dynamics defined by Figure 2 are *order preserving* [8], that is, if $x \leq \tilde{x}$ and $\mathbf{u} \leq \tilde{\mathbf{u}}$ then $\phi(t, x, \mathbf{u}) \leq \phi(t, \tilde{x}, \tilde{\mathbf{u}})$ for all time. This, in practice, corresponds to a system where: (i) larger inputs and (ii) initial conditions with greater velocity and greater displacements, always generate flows that achieve a greater velocity and greater displacement along the path at any time.

C. Safety Control Problem

To mathematically define safety, we identify sets of vehicle states, called *bad* sets, that represent collisions. We distinguish these bad sets into specific types, the first are *merging* collisions between two vehicles, and the second are *rear-end* collisions between two vehicles. Merging collisions can occur where two paths first come together, and rear-end collisions can occur where two paths overlap. From Figure 1, we see that there can be merging collisions between vehicles 1 and 2, and vehicles 1 and 3, and rear-end collisions between vehicles 1 and 2, and vehicles 1 and 3.

Let the bad set $\mathbf{B}^M_{1j} \subset X$ represent a merging collision between vehicles 1 and $j \in \{2, 3\}$. Define this set as $\mathbf{B}^M_{1j} := \{x \in X \mid x^1_1 \in [L^{1,M}_{1j}, U^{1,M}_{1j}] \text{ and } x^j_1 \in [L^{j,M}_{1j}, U^{j,M}_{1j}]\}$, where the intervals represent the section of the respective path contained in the merging collision circles shown in Figure 1.

Let the bad set $\mathbf{B}^{RE}_{1j}$ represent a rear-end collision between vehicles 1 and $j \in \{2, 3\}$. Define this set as $\mathbf{B}^{RE}_{1j} := \{x \in X \mid x^1_1 \in [L^{1,RE}_{1j}, U^{1,RE}_{1j}],\ x^j_1 \in [L^{j,RE}_{1j}, U^{j,RE}_{1j}] \text{ and } |x^1_1 - x^j_1 + \rho^{RE}_{1j}| < l\}$, where each interval represents the section of the respective path contained in the rear-end collision region in Figure 1, l is the vehicle length, and $\rho^{RE}_{1j} := L^{j,RE}_{1j} - L^{1,RE}_{1j}$ is a relative translation. This set represents the set of states in which two vehicles are less than a car length apart and both inside the rear-end collision area seen in Figure 1.

Therefore, all the possible collision types are given by the sets $\mathbf{B}^M_{12}$, $\mathbf{B}^M_{13}$, $\mathbf{B}^{RE}_{12}$, and $\mathbf{B}^{RE}_{13}$. We define the total bad set $\mathbf{B} := \mathbf{B}^M_{12} \bigcup \mathbf{B}^M_{13} \bigcup \mathbf{B}^{RE}_{12} \bigcup \mathbf{B}^{RE}_{13}$, and say the system is safe if the flow never enters $\mathbf{B}$.

D. Control design

We seek to design a controller that prevents collisions between vehicles, that is, keeps the flow outside of $\mathbf{B}$, by regulating the input signal $\mathbf{u}$ of the entire system. This must be accomplished without stopping vehicles (liveliness) and without causing any vehicle to exceed a speed limit. We accomplish this by combining two-vehicle primitives that each prevent the flow from entering one bad set $\mathbf{B}_{1j}^{spec}$, where $spec \in \{M, RE\}$ and $j \in \{2,3\}$.

For the primitive used to avoid the bad sets $\mathbf{B}_{1j}^{M}$, we assume that both vehicles 1 and $j \in \{2,3\}$ can apply control input to avoid a collision. To construct the control primitive $g_{1j}^{M} : X \rightrightarrows U^1 \times U^j$, we look to compute the *capture* set, defined by

$$C_{1j}^{M} := \{x \in X \mid \forall\ \mathbf{u} \in S(U),\ \exists\ t \in \mathbb{R}_+ \text{ s.t. } \phi(t,x,\mathbf{u}) \in \mathbf{B}_{1j}^{M}\}.$$

This set corresponds to all initial conditions that generate a merging collision between vehicles 1 and $j \in \{2,3\}$ no matter what input is applied. If the flow is kept out of the capture set, then necessarily the flow never enters the bad set.

The order preserving properties of the system dynamics allow us to compute the capture set C_{1j}^{M} with the restricted capture set

$$C_{1j}^{M}(\mathcal{U}) := \left\{ x \in X \;\middle|\; \begin{array}{l} \forall \mathbf{u} \in S(\mathcal{U}), \exists\ t \in \mathbb{R}_+ \\ \text{s.t. } \phi(t,x,\mathbf{u}) \in \mathbf{B}_{1j}^{M} \end{array} \right\}, \tag{1}$$

where $\mathcal{U} \subset U$. This set corresponds to all initial conditions that generate a merging collision for any input signal $\mathbf{u}$ contained in $\mathcal{U}$. We introduce the input sets $\mathcal{U}_{LH}^{1j} := \{u \in U \mid u^1 = u_{min}^1,\ u^j = u_{max}^j\}$, and $\mathcal{U}_{HL}^{1j} := \{u \in U \mid u^1 = u_{max}^1,\ u^j = u_{min}^j\}$. Using these input sets, we compute C_{1j}^{M} with restricted capture sets according to

Theorem 1: $C_{1j}^{M} = C_{1j}^{M}(\mathcal{U}_{LH}^{1j}) \cap C_{1j}^{M}(\mathcal{U}_{HL}^{1j})$.

A proof of this theorem can be found in [8]. This result is significant because the restricted capture sets $C_{1j}^{M}(\mathcal{U})$ can be efficiently computed online, giving us C_{1j}^{M}.

With this characterization of the capture set, the primitive g_{1j}^{M} can be found as

$$g_{1j}^{M}(x) := \begin{cases} \mathcal{U}_{LH}^{1j} & \text{if } x \in \partial C_{1j}^{M}(\mathcal{U}_{LH}^{1j}) \cap C_{1j}^{M}(\mathcal{U}_{HL}^{1j}) \\ \mathcal{U}_{HL}^{1j} & \text{if } x \in C_{1j}^{M}(\mathcal{U}_{LH}^{1j}) \cap \partial C_{1j}^{M}(\mathcal{U}_{HL}^{1j}) \\ \mathcal{U}_{LH}^{1j} \cup \mathcal{U}_{HL}^{1j} & \text{if } x \in \partial C_{1j}^{M}(\mathcal{U}_{LH}^{1j}) \cap \partial C_{1j}^{M}(\mathcal{U}_{HL}^{1j}) \\ U & \text{otherwise.} \end{cases}$$

Under this primitive, control action is only applied when the flow is on the boundary of the capture set C_{1j}^{M} and any initial condition starting outside of C_{1j}^{M} will generate a flow that remains outside, thus guaranteeing safety.

For the primitives used to avoid $\mathbf{B}_{1j}^{RE}$, we assume only vehicle $j \in \{2,3\}$ applies control. This assumption is made to prevent modules from conflicting while vehicle 1 is simultaneously avoiding a rear-end collision with one vehicle and a merging collision with the other. This assumption can be formally stated by modeling the input of vehicle 1 as a disturbance, that is, we suppose vehicle 1 is trying to push the flow into the bad set $\mathbf{B}_{1j}^{RE}$.

To construct the control primitive $g_{1j}^{RE} : X \rightrightarrows U^j$, we look to compute the *capture* set, defined by

$$C_{1j}^{RE} := \left\{ x \in X \;\middle|\; \begin{array}{l} \forall\ (\mathbf{u}^2, \mathbf{u}^3) \in S(U^2 \times U^3),\ \exists\ t \in \mathbb{R}_+, \\ \exists\ \mathbf{u}^1 \in S(U^1) \text{ s.t. } \phi(t,x,\mathbf{u}) \in \mathbf{B}_{1j}^{RE} \end{array} \right\}.$$

This set corresponds to all initial conditions that generate a rear-end collision for some input $\mathbf{u}^1 \in S(U^1)$ regardless of the input $(\mathbf{u}^2, \mathbf{u}^3) \in S(U^2 \times U^3)$. The order preserving properties of the system dynamics allow us to compute the capture set C_{1j}^{RE} with the restricted capture set

$$C_{1j}^{RE}(\mathcal{U}) := \left\{ x \in X \;\middle|\; \begin{array}{l} \forall (\mathbf{u}^2, \mathbf{u}^3) \in S(\mathcal{U}^2 \times \mathcal{U}^3),\ \exists\ t \in \mathbb{R}_+ \\ \exists \mathbf{u}^1 \in S(\mathcal{U}^1) \text{ s.t. } \phi(t,x,\mathbf{u}) \in \mathbf{B}_{1j}^{RE} \end{array} \right\}.$$

This set corresponds to all initial conditions that generate a rear-end collision for some input $\mathbf{u}^1 \in S(U^1)$ regardless of the input $(\mathbf{u}^2, \mathbf{u}^3) \in S(\mathcal{U}^2 \times \mathcal{U}^3)$. We introduce the input sets $\mathcal{U}_H^j := \{u \in U \mid u^j = u_{max}^j\}$, $\mathcal{U}_L^j := \{u \in U \mid u^j = u_{min}^j\}$. Using these input sets, we compute C_{1j}^{RE} with restricted capture sets according to

Theorem 2: $C_{1j}^{RE} = C_{1j}^{RE}(\mathcal{U}_L^j) \cap C_{1j}^{RE}(\mathcal{U}_H^j)$.

A proof of this theorem can be found in [11]. This result is significant because the restricted capture sets $C_{1j}^{RE}(\mathcal{U})$ can be efficiently computed online, giving us C_{1j}^{RE}.

With this characterization of the capture set, the module g_{1j}^{RE} can be found as

$$g_{1j}^{RE}(x) := \begin{cases} \mathcal{U}_L^j & \text{if } x \in \partial C_{1j}^{RE}(\mathcal{U}_L^j) \text{ and } (x_1 > x_j) \\ \mathcal{U}_H^j & \text{if } x \in \partial C_{1j}^{RE}(\mathcal{U}_H^j) \text{ and } (x_1 < x_j) \\ U & \text{otherwise.} \end{cases}$$

Under this primitive, control action is only applied when the flow is on the boundary of the capture set C_{1j}^{RE} and any initial condition starting outside of C_{1j}^{RE} remains outside, thus guaranteeing safety.

The control primitives are combined under conjunction, giving the complete feedback $g : X \rightrightarrows U$ as $g(x) := g_{12}^{M}(x) \cap g_{13}^{M}(x) \cap g_{12}^{RE}(x) \cap g_{13}^{RE}(x)$. We say $g(x)$ is non-conflicting if $g(x) \neq \emptyset$, that is, no primitives generate conflicting control inputs.

III. Experiments

A. Experimental Setup

The experimental test-bed consists of a 6x6 meter arena, custom dynamic vehicles equipped with central processing units (CPUs), a local 802.11b wireless network, and an overhead camera positioning system. The custom dynamic vehicles are shown in Figure 3.

1) Longitudinal dynamics: The longitudinal dynamics of the vehicles are a scaled version of the dynamics of a full-size high-mobility multipurpose wheeled vehicle (HMMWV). Each vehicle is 0.17 meters wide and 0.38 meters long. The torque command τ_d is issued as a percentage from -100 to 100 percent, with 100 percent corresponding to a torque value of 0.9 Nm. The longitudinal dynamics of each vehicle,

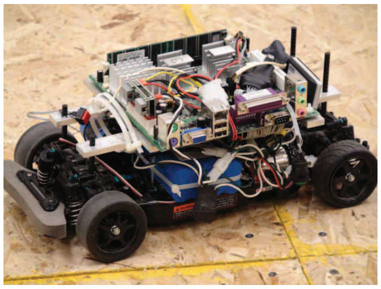

Fig. 3. Vehicle Hardware.

as described in Section II-A, were found through standard least squares techniques to be:

$$\begin{array}{l} \text{Vehicle 1: } \dot{x}_2^1{=}3.77u^1 - 55 \\ \text{Vehicle 2: } \dot{x}_2^2{=}5.07u^2 - 12 \\ \text{Vehicle 3: } \dot{x}_2^3{=}6.43u^3 - 133, \end{array} \tag{2}$$

where $\dot{x}_2$ is in mm/s^2, u is in $kg\ mm/s^2$, parameter a is in kg^{-1}, and parameter b is in mm/s^2. The acceleration is limited to be between $\dot{x}_{2,min}$ = -250 mm/s^2 and $\dot{x}_{2,max}$ = 250 mm/s^2. A speed limiter uses a PI controller to bound the velocity between v_{min} = 350 mm/s and v_{max} = 850 mm/s. Therefore, the vehicle dynamics conform with the hybrid automaton in Figure 2.

2) CPU: The CPU used on the vehicles is a VIA EPIA Mini-ITX with a 600 MHz processor, 512 MB of RAM, and a 40 GB hard drive. The control and communication algorithms are written in C and run on the Mini-ITX using a Linux Fedora Core 5 operating system. The resulting torque commands from the control algorithms are transmitted to a BrainStem Moto 1.0 motor controller via a serial connection, where the scaled HMMWV dynamics are emulated.

3) Communication and Positioning System: Communication between the lab computers and the vehicles is achieved through a local 802.11b wireless network. The wireless network is used to emulate V2V and V2I communication. The overhead positioning system consists of four overhead 640 x 480 BW Firefly MV cameras linked to two desktop computers via a FireWire cable. The computers use template matching to locate patterns affixed to the top of the vehicles. The computers then calculate and send the position and orientation of each vehicle over the wireless network.

4) Path following: A steering controller on the vehicles uses a basic bicycle model to follow the paths by means of actuating the steering angle. The vehicles map their position from the overhead positioning system to a longitudinal displacement along their path for use in the control algorithms. The vehicle velocity is assumed to be the same as the velocity along the path because the paths are followed accurately.

B. Algorithm implementation

This section examines how the control primitives are implemented to guarantee safety using the controller presented in Section II-D.

1) Algorithm Parameters: The system design parameters that must be selected to guarantee the non-conflicting composition of the control primitives, are the allowed velocity ranges, the size and location of the bad sets, and the lengths of the paths that the vehicles follow.

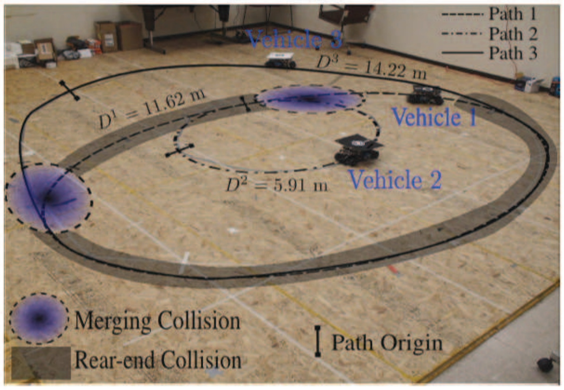

Fig. 4. Image of the experimental test-bed, with path lengths, path origins and collision locations denoted.

The velocity ranges were all chosen to be X_2^i = $[.35, .85]m/s$. The dimensions of the merging collision bad sets were chosen to extend .9 m along each path. The vehicle lengths that define the rear-end bad set were measured to be .38 m. The lengths of each path section are shown in Figure 4.

2) System Discretization: Since the algorithms are run online with a digital computer, the control primitives use a discrete time implementation of the dynamics, with a time step of Δt = 0.1 seconds. Using a forward Euler approximation of the vector field $f^i(x^i, u^i)$ (Section II-B), the discrete time equivalent of the system is given as $x_1^i[n+1] = x_1^i[n] + x_2^i[n] * \Delta t$, and $x_2^i[n+1] = \bar{F}(x_2^i[n], \mathbf{u}^i[n])$, where $\bar{F}(x_2^i[n], \mathbf{u}^i[n]) := x_2^i[n] + \Delta t f_2^i(u[n]^i, x_2^i[n])$.

3) Calculation of Control Primitives: The control primitives, as described in Section II D, require the computation of restricted capture sets. By exploiting the order preserving properties of the system dynamics, the restricted capture set can be recursively computed with linear complexity algorithms. For a constant input $\mathbf{u}^i[n] = u^i$ for all $n \in \mathbb{N}$, we define $\bar{F}^0(x_2^i, u^i) := x_2^i$ and the recursive relation $\bar{F}^{k+1}(x_2^i, u^i) := \bar{F}(\bar{F}^k(x_2^i, u^i), u)$, where $k \in \mathbb{N}$. For the constants L^0 and U^0, let

$$\begin{array}{ll} L^{i,k}(x_2^i, u^i) & := L^{i,0} - \sum_{n=0}^{k-1} \bar{F}^n(x_2^i, u^i) \\ U^{i,k}(x_2^i, u^i) & := U^{i,0} - \sum_{n=0}^{k-1} \bar{F}^n(x_2^i, u^i). \end{array} \tag{3}$$

The restricted capture set $C_{1j}^M(\mathcal{U})$ can be computed as follows

Claim 1:

$$C_{1j}^M(\mathcal{U}) = \left\{ x \in X \;\middle|\; \begin{array}{l} \exists\, k \geq 0,\ \exists\, u \in \mathcal{U} \text{ s.t.} \\ L^{1,k}(x_2^1, u^1) < x_1^1 < U^{1,k}(x_2^1, u^1) \\ L^{j,k}(x_2^j, u^j) < x_1^j < U^{j,k}(x_2^j, u^j) \end{array} \right\},$$

where $L^{1,0} = L_{1j}^{1,M}$, $U^{1,0} = U_{1j}^{1,M}$,$L^{j,0} = L_{1j}^{j,M}$, and $U^{j,0} = U_{1j}^{j,M}$. A proof of this result can be found in [8].

Rather than explicitly computing the set $C_{1j}^{RE}(\mathcal{U})$ based on the set $\mathbf{B}_{1j}^{RE}$ and the control input set $\mathcal{U}$, we represent $\mathbf{B}_{1j}^{RE}$ as a union of interval sets, where $y \in \mathbb{R}$ and $\mathcal{B}_{1j}^{RE}(y)$ is defined as

$$\mathcal{B}^{RE}_{1j}(y) := \{x \in X \mid x_1^1, x_1^j \in [y-l, y+l]\},$$

which can be used to redefine $\mathbf{B}^{RE}_{1j}$ as

$$\mathbf{B}^{RE}_{1j} = \bigcup_{y\in[L^{RE}_{1j}+l,\, U^{RE}_{1j}-l]} \mathcal{B}^{ACC}_{1j}(y). \tag{4}$$

We now define the restricted capture set for the parameter $y \in \mathbb{R}$ and input $(\mathbf{u}^2, \mathbf{u}^3) \in S(\mathcal{U}^2 \times \mathcal{U}^3)$ as

$$\widetilde{C}^{RE}_{1j}(\mathcal{U}, y) := \left\{ x \in X \;\middle|\; \begin{array}{l} \forall(\mathbf{u}^2,\mathbf{u}^3)\in S(\mathcal{U}^2\times\mathcal{U}^3),\\ \exists\, t\in\mathbb{R}_+,\ \exists\, \mathbf{u}^1\in S(\mathcal{U}^1)\\ \text{s.t. } \phi(t,x,\mathbf{u})\in \mathcal{B}^{RE}_{1j}(y)\end{array}\right\}.$$

As shown in [11], each interval set $\mathcal{B}^{RE}_{1j}(y)$ generates a restricted capture set $\widetilde{C}^{RE}_{1j}(\mathcal{U}, y)$ given by

$$\widetilde{C}^{RE}_{1j}(\mathcal{U}, y) = \left\{ \begin{array}{l} x\in X \mid \exists\, k\ge 0\ \ \exists\, u\in\mathcal{U}\text{ s.t.}\\ L^{1,k}(x_2^1,u^1_{max}) < x_1^1 < U^{1,k}(x_2^1,u^1_{min})\\ L^{j,k}(x_2^j,u^j) < x_1^j < U^{j,k}(x_2^j,u^i)\end{array}\right\},$$

where we have taken $L^{1,0} = y - l$ and $L^{j,0} = y - l$ to be the lower bounds, and $U^{1,0} = y + l$ and $U^{j,0} = y + l$ the upper bounds of the interval sets used in the definition of $\mathcal{B}^{RE}_{1j}(y)$. The dynamics are independent of the state x_1, therefore, we take the union of all of these sets to obtain the restricted capture set $C^{RE}_{1j}(\mathcal{U})$, that is,

$$C^{RE}_{1j}(\mathcal{U}) = \bigcup_{y\in[L^{RE}_{1j}+l,\, U^{RE}_{1j}-l]} \widetilde{C}^{ACC}_{1j}(\mathcal{U}, y).$$

.

4) Criteria for Non-conflicting Primitive Conjunction: Here, we provide several conditions such that the conjunction of all primitives is non-conflicting, that is, $g(x) \neq \emptyset$. Define the canonical projection $\tau_{1j} : X \to X_1^1 \times X_1^j$, which can be naturally extended to sets. We define the lower bound $\alpha^M_{1j} := \inf \tau_{1j}(C^M_{1j})$. Therefore, if $x \in C^M_{1j}$ then $\alpha^M_{1j} \leq (x_1^1, x_1^j) \leq \sup \tau_{1j}(\mathbf{B}^M_{1j})$ by the definition of α^M_{1j} and the order preserving property of the dynamics with respect to state. The following conditions are sufficient to guarantee the conjunction of the modules is non-conflicting:

(i) The capture set C^M_{1j} does not extend the entire length of a vehicle's path, namely $(\sup \tau_{1j}(B^M_{1j}) - \alpha^M_{1j}) < (D^1, D^j)$. This prevents a capture set from overlapping itself in X_1^1 or X_1^j.

(ii) The capture sets C^M_{12} and C^M_{13} do not overlap each other in X_1^1. This ensures that vehicle 1 is avoiding at most one merging capture set at a time.

(iii) We require that the vehicles make a safe transition from avoiding a merging collision to avoiding a rear-end collision. This is accomplished if the capture set C^{RE}_{1j} entirely enters the capture set B^M_{1j} for $j \in \{2, 3\}$. If this holds, then the order preserving properties of the system guarantee that at the moment the vehicles pass B^M_{1j} the flow cannot be contained within C^{RE}_{1j}, thus guaranteeing the two primitives never conflict.

The maximum length in both X^j and X^1 of the capture set C^M_{1j}, equal to $\sup \tau_{1j}(B^M_{1j}) - \alpha^M_{1j}$, was calculated to be 1.96 m for j=2 and j=3. Condition (iii) was shown to hold in a MATLAB numerical analysis.

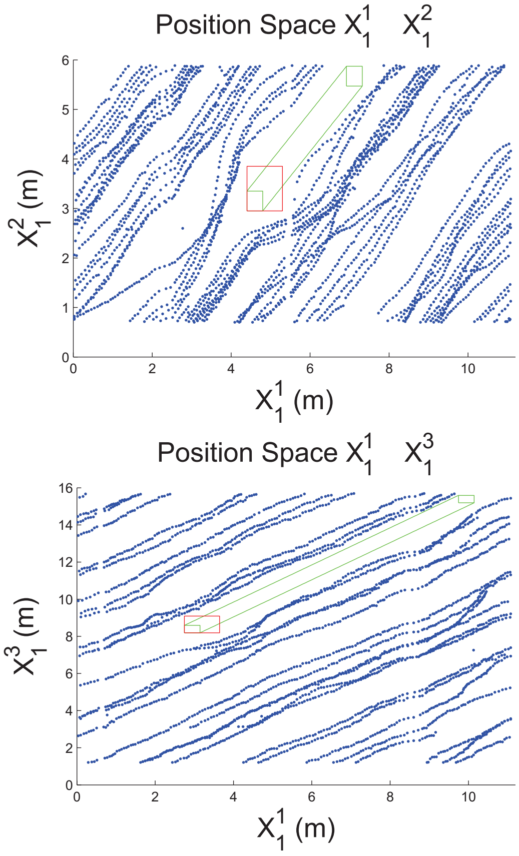

Fig. 5. The blue points represent vehicle positions during the experiment. The red boxes are B^M and the green boxes are B^{RE}. The blue points show the position of the vehicles over time. Note that the coordinate system for vehicle 1 has been shifted in the second figure so that the bad sets are easily shown.

C. Experimental Results

The algorithms presented in Section III-B were run on the experimental setup outlined in Section III-A for 6 minutes and 4 seconds. The merging control module applied 16 instances of control, and the rear-end control module applied 9 instances of control.

The flow over the entire system history is projected into the position spaces $X_1^{1,2}$ and $X_1^{1,3}$ in Figure 5. The vehicle positions never entered the bad set, so no collisions occurred during the experiment. There were cases where the flow came close to the bad sets, thus demonstrating control is applied without being conservative. To illustrate a specific instance of collision avoidance, Figure 6 shows an example of the prevention of a merging collision, and Figure 7 shows an example of the prevention of a rear-end collision.

IV. Conclusion

In this paper, we have presented a formal controller design methodology for a three-vehicle safety control problem on a roundabout system. By exploiting the order preserving properties of the system, we have designed a controller through

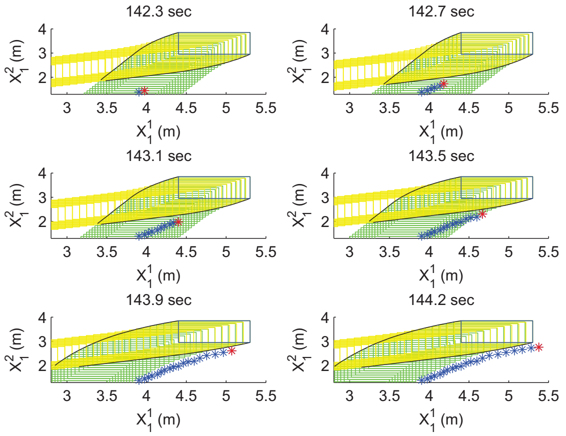

Fig. 6. Experiment data showing vehicles 1 and 3 applying merging collision avoidance control. The blue box is B_{12}^M, the red point current position of the vehicles, and the blue points are the previous positions. The yellow set is a slice of $C_{12}^M(\mathcal{U}_{HL}^{12})$ corresponding to the current velocity and the green set is a slice of $C_{12}^M(\mathcal{U}_{LH}^{12})$ corresponding to the current velocity. The intersection of these sets is the capture set slice C_{12}^M. Merging control is applied from $t = 142.7$ *sec* to $t = 149.3$ *sec*, after which time no collision is predicted.

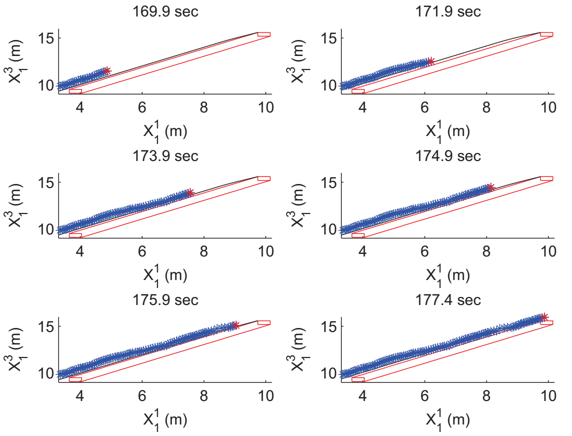

Fig. 7. Experiment data showing vehicles 1 and 2 applying rear-end collision avoidance control. The red lines are the boundaries of B_{12}^{RE}, the red point current position of the vehicles, and the blue points are the previous positions. The black line represents the upper boundary of the capture set for the current velocity slice of C_{12}^{RE}. Rear-end control is applied from $t = 171.9$ *sec* to $t = 175.9$ *sec*, after which time a collision is no longer predicted.

the composition of control primitives, each of which have linear complexity with respect to the number of continuous states. We have shown that if certain design constraints are met, then the conjunction of the modules is non-conflicting, thereby guaranteeing the system is safe.

We have implemented the proposed algorithm on the multi-vehicle testbed at the and provided experimental results showing that the system maintains safety when running the algorithms online.

Future work involves extending the roundabout system to include more conflict points, increasing the number of vehicles on the roundabout drill, investigating how communication and actuator delays affect safety guarantees, and understanding how to compose control primitives in more general traffic network systems.

References

[1] Car 2 Car Communication Consortium. http://www.car-to-car.org.

[2] Crash Avoidance Metrics Partnership (CAMP). http://www.camp-ivi.com.

[3] Vehicle Infrastructure Integration (VII). http://www.its.dot.gov/vii.

[4] National roundabout conference, trb. http://144.171.11.107/Main/Public/Blurbs/156622.aspx, 2005.

[5] L. Alvarez and R. Horowitz. Safe platooning in automated highway systems. *California Partners for Advanced Transit and Highways (PATH). Research Reports: Paper UCB-ITS-PRR-97-46.*

[6] L. Alvarez and R. Horowitz. Analysis and verification of the PATH AHS coordination-regulation layers hybrid system. In *American Control Conference*, pages 2460–2461, Albuquerque, New Mexico, 1997.

[7] L. Alvarez and R. Horowitz. Hybrid controller design for safe maneuvering in the PATH AHS architecture. In *American Control Conference*, pages 2454–2459, Albuquerque, New Mexico, 1997.

[8] 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.

[9] V. Desaraju, H. C. Ro, M. Yang, E. Tay, S. Roth, and D. Del Vecchio. Partial order techniques for vehicle collision avoidance: Application to an autonomous roundabout testbed. In *International Conference on Robotics and Automation*, 2009.

[10] J. A. Haddon, D. N. Godbole, A. Deshpande, and J. Lygeros. Verification of hybrid systems: Monotonicity in the AHS control system. In *Hybrid Systems III.* Lecture Notes in Computer Science, vol. 1066. Springer, 1996.

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

[12] J. K. Hedrick, Y. Chen, and S. Mahal. Optimized vehicle control/communication interaction in an automated highway system. *California Partners for Advanced Transit and Highways (PATH). Research Reports: Paper UCB-ITS-PRR-2001-29.*, 2001.

[13] R. Horowitz and P. Varaiya. Control design of an automated highway system. *Proceedings of the IEEE*, 88(7):913–925, Jul 2000.

[14] R. Horowitz and P. Varaiya. Control design of an automated highway system. *Proceedings of the IEEE*, 88(7):913–925, 2000.

[15] J. Lygeros, D. N. Godbole, and S. Sastry. A verified hybrid controller for automated vehicles. In *Conf. on Decision and Control*, pages 2289–2294, Kobe, Japan, 1996.

[16] J. Lygeros and N. Lynch. Strings of vehicles: Modeling and safety conditions. pages 273–288, 1998.

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

[18] A. Puri and P. Varaiya. Driving safely in smart cars. In *American Control Conference*, pages 3597–3599, Seattle, WA, 1995.

[19] 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.

[20] 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.

[21] U.S. DOT Turner-Fairbank Highway Research Administration (TFHRA). Roundabouts: An informational guide. http://www.tfhrc.gov/safety/00068.htm, 2000.

[22] P. Varaiya. Smart cars on smart roads. *IEEE Transactions on Automatic Control*, 38(2):195–207, Feb 1993.

[23] 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(1):46–57, 2008.

[24] R. Verma and D. Del Vecchio. Continuous control of hybrid automata with imperfect mode information assuming separation between state estimation and control. *Proc. of Conference on Decision and Control*, 2009.