Analysis of Control Barrier Function Framework for Safety-Critical Control of Connected Automated Vehicles

John Yu, Dr. Tamas Molnar, Anil Alan, Dr. Gabor Orosz

December 2022

1 Abstract

Vehicle-to-vehicle (V2V) connectivity has gained traction in the automated vehicle space for its potential to improve congestion mitigation, fuel economy, and vehicle safety. V2V connectivity refers to when automated vehicles exchange data with other nearby vehicles to inform their driving. In this way, the vehicles are now 'connected'. However, the current lack of a control framework with provable safety guarantees for V2V connected vehicles prevents this form of automation from being applicable outside of the academic setting. The objective of this project was to develop a V2V safety-critical controller via control barrier function (CBF) framework and apply this framework to models of increasing fidelity, from the 1 state to 4 state model case. Due to the complexity of real car dynamics, this paper approximates the vehicle with integrator, unicycle, and bicycle models and characterizes general controller behaviors for these models. Simulations of the CBF framework applied to these three models were conducted in MatLab and characteristic vehicle behaviors were analyzed for varying parameter values and initial conditions. Particular behaviors of interest were examined to find 1. initial (velocity) conditions that do not satisfy the safety condition 2. vehicle switching position between nominal and safety-critical control as a function of parameters, and 3. vehicle freezing as a function of parameters. These behaviors are useful in characterizing when the CBF framework controller is effective in delivering desirable safety-critical driving, giving a better understanding of how to engineer safety guarantees for V2V vehicles.

Keywords: V2V connectivity, autonomous vehicles, safety-critical control, control barrier function framework, extended barriers, integrator model, unicycle model, bicycle model, nominal to safety-critical switching, vehicle controller freezing, vehicle model parameters

2 Introduction

Vehicle-to-vehicle (V2V) connectivity has gained traction in the automated vehicle space for its potential to improve congestion mitigation, fuel economy, and vehicle safety. V2V connectivity refers to when automated vehicles exchange real-time data with other nearby vehicles to inform their driving. In this way, the vehicles are now 'connected'. This form of connectivity is powerful because it enables the vehicle to access information about the speed and position of nearby vehicles that it may not have otherwise, which can be used to detect crashes, dangerous traffic, and other driving phenomenon earlier and more readily than was previously possible. However, a significant problem in the V2V space is that there is currently lack of a control framework with provable safety guarantees for V2V connected vehicles. This limitation prevents this form of automation from being applicable outside of the academic setting.

Control barrier functions have been studied as a tool to engineer safety-critical control in applications including multi-agent systems [1], robotics [2], and automated vehicles [3]. Existing research on the creation of robust, tunable CBF frameworks [4],[5] and adaption of this framework for autonomous vehicles [3] makes this framework a natural candidate for exploration in the safety-critical V2V connected vehicle space. The objective of this project was to develop a V2V safety-critical controller via control barrier function (CBF) framework and apply this framework to models of increasing fidelity, from the 1 state to 4 state model case. Due to the complexity of real car dynamics, this paper approximates the vehicle with integrator, unicycle, and bicycle models and characterizes general controller behaviors for these models. This builds off of previous research done by T.G. Molnar et al, which shows that this CBF framework can be used to create application-agnostic safety-critical control of robotic systems without relying on the use of a high-fidelity dynamical model of the robot [6]. Simulations of the CBF framework applied to these three models were conducted in MatLab and characteristic vehicle behaviors were analyzed for varying parameter values and initial conditions.

This paper first demonstrates that a CBF framework applied to integrator, unicycle, and bicycle vehicle models can create desirable control with theoretical safety guarantees for V2V connected vehicles. Then, by simulating and evaluating the controller under varying safety critical obstacle cases, initial state conditions, and tuning and gain parameter conditions, limiting controller behaviors are identified that characterize when the framework delivers desirable control, or lack thereof. Particular behaviors of interest were examined to find 1. initial (velocity) conditions that do not satisfy the safety condition 2. vehicle switching position between nominal and safety-critical control as a function of parameters, and 3. vehicle freezing as a function of parameters. It is found that in the extended barrier case, the control framework safety goal may be violated if the vehicle is assigned high initial velocity. This limits the ability to simulate a more true to life vehicle case where there likely would be nonzero initial velocity. Vehicle switching is shown to be dependent on system parameters α and α_e in the unextended barrier case, as well as vehicle velocity in the

extended barrier case. However, more research is needed to characterize switching behavior for higher fidelity vehicle models. Finally, it is found that in the models where vehicle freezing occurs (unicycle and bicycle), this behavior as a function of parameters becomes more prominent with higher angular gain and is most prominent when lateral gain is between 0.1 and 0.2. However, more work is needed to understand mathematically why freezing occurs in the unicycle and bicycle, and why the aforementioned parameter conditions are particularly conducive to freezing.

The hope is that these findings serve as a base for further investigation of the application of control barrier functions on models of higher fidelity beyond the bicycle case and complex obstacle cases beyond the stationary single obstacle.

3 Methods

The general procedure for analyzing the impact of a control barrier function (CBF) framework on V2V connected vehicle safety-critical behavior consisted of three areas. First, a vehicle model was developed to characterize the vehicle's dynamics. Second, a safety function is defined and the CBF framework is applied to the model through inputs to create safety-critical control. Third, the resulting system is simulated via MatLab and the controller behavior with varying models and parameters is evaluated. This section will focus on discussing vehicle models that were explored and characterizing the procedure for applying the CBF framework.

3.1 General Representation of Vehicle Models

The first step is to develop a mathematical representation, or model, of the vehicle's dynamics. The most general expression of a vehicle model can be represented in state space form as (1).

$$\dot{x} = f(x) + g(x)u \tag{1}$$

Here, $\dot{x}$ represents the time derivative of relevant vehicle states, typical states being longitudinal and lateral positions, velocity, and heading angle. The u component represents the controller inputs to the system. These inputs are the only portion of the model that the controller has direct control over, and this component is where the CBF framework will be implemented to create safety-critical control. For example, some controller inputs that were used in this work were acceleration, which controls the velocity state, and front wheel steering angle, which controls the heading angle state for a bicycle model. $g(x)$ and $f(x)$ simply separate the portions of the model that u has and does not have direct control over respectively. A typical representation of a vehicle model with states is given in **Figure 1**.

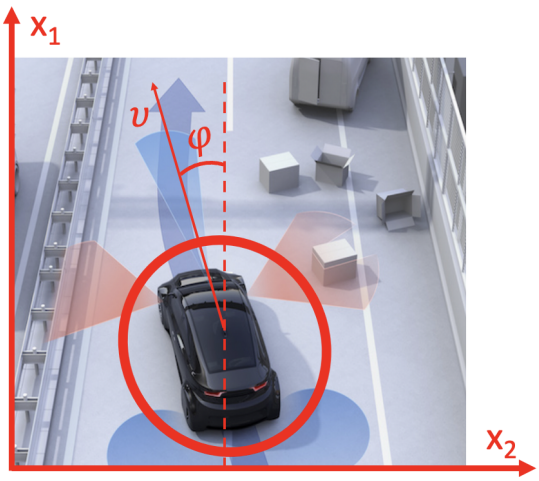

Figure 1: Representation of autonomous vehicle in a safety critical scenario, characterizing the typical vehicle states needed to fully define a model in this paper. As shown, x_1 $[m]$ is the longitudinal position (in the direction of the road) while x_2 $[m]$ is the lateral direction (transverse to the direction of the road), v $[m/s]$ is the vehicle velocity, and φ $[rad]$ is the vehicle heading angle relative to x_1.

Because the dynamics of a real-world car are complex, a model that fully characterizes the vehicle behavior is difficult to implement. However, simpler vehicle models can give a good representation of car behavior while making it easier to analyze relevant dynamics and behavior. The three models examined in this project were the integrator (point), unicycle, and bicycle. These three models, their mathematical representations, and their dynamics are elaborated in **4 Implementation of Various Vehicle Models**.

3.2 Implementation of Control Barrier Function Framework

The crux of the control barrier function (CBF) framework is to define a safety control barrier function $h(x)$, as shown in (2).

$$h(x) = d - r \tag{2}$$

$$d = \sqrt{(x_{o1} - x_1)^2 + (x_{o2} - x_2)^2} \tag{3}$$

This $h(x)$ $[m]$ is the difference between d $[m]$ (3), the distance from the car position $[x_1, x_2]$ to the center position of the obstacle $[x_{o1}, x_{o2}]$, and r $[m]$, the radius of the perceived obstacle. A visual representation of d and r can be seen in **Figure 2**.

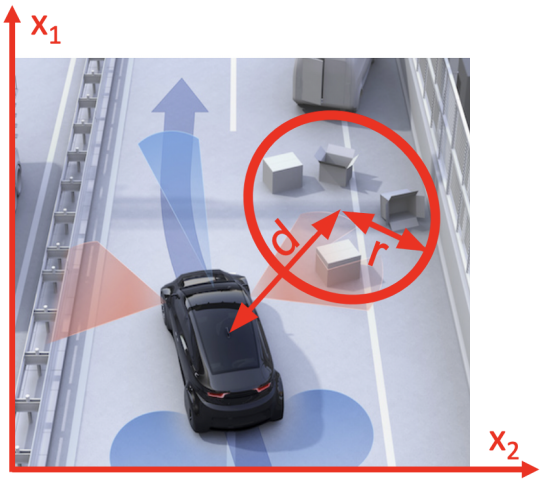

Figure 2: Representation of autonomous vehicle in a safety critical scenario. The objective of the car is to swerve around the boxes dropped on the road, then keep driving. x_1 is the longitudinal position (in the direction of the road) while x_2 is the lateral direction (transverse to the direction of the road). d is shown as the distance from the vehicle to the obstacle and r is the radius of the perceived obstacle.

If $h(x)$ is positive, there is a positive distance between the vehicle and the outer boundary of the obstacle, which is regarded as a circle in 2D space. This means that the vehicle is safe. A more formal way of writing this is by defining a safe set S, as shown in (4), that encompasses the total domain over which safety is guaranteed.

$$S = [x \in \mathbb{R}^n : h(x) \geq 0] \tag{4}$$

The objective of the CBF framework is to ensure that the controller remains within the safe set for all time, and the case where this is achieved can be written as the safety goal, (5). Finally, a safety condition can be written in terms of $h(x)$ for creating a controller that will guarantee safety (i.e. $h(x) \geq 0$ for all time), (6).

$$x(0) \in S \rightarrow x(t) \in S \quad \forall t \tag{5}$$

$$\frac{\partial h(x)}{\partial x} \cdot (f(x) + g(x) \cdot u) \geq -\alpha(h(x)) \tag{6}$$

In (6), α is a non-dimensional tuning parameter. The next step is to define a controller input that by definition satisfies (6). This can be written as shown in (7).

$$
\begin{aligned}
k_s(x) = &\text{argmin} ||u - k_n(x)||^2 \\
&u \in \mathbb{R}^m \\
&s.t. \quad \frac{\partial h(x)}{\partial x} \cdot (f(x) + g(x) \cdot u) \geq -\alpha(h(x))
\end{aligned} \tag{7}
$$

Here, $k_s(x)$ is the safety critical controller, and $k_n(x)$ is the desired (nominal) controller. The optimization problem translates as modifying the nominal input just enough so that the safety condition is fulfilled. To solve the system analytically, the safety condition can be rewritten as follows:

$$
c(x) \triangleq \frac{\partial h(x)}{\partial x} \cdot (f(x) + g(x) k_n(x)) + \alpha(h(x)) \tag{8}
$$

$$
b(x) \triangleq \frac{\partial h(x)}{\partial x} \cdot g(x) \tag{9}
$$

Finally, the analytical solution to $k_s(x)$ can be written as (10).

$$
k_s(x) = k_n(x) + \max\left(0, \frac{-c(x)}{||b(x)||^2}\right) b^T(x) \tag{10}
$$

We notice that the final general form of $k_s(x)$ consists of $k_n(x)$ and a max function dependent on $h(x)$ through $\alpha(h(x))$. In the case that the vehicle is very far from an obstacle $\alpha(h(x))$ becomes very large. If it is large enough to make $c(x)$ positive then the max function outputs zero, so $k_s(x) = k_n(x)$. Once the vehicle moves close enough to an obstacle, $\alpha(h(x))$ becomes small enough such that $c(x) < 0$, and the max function outputs the second term. When this happens we say the controller 'switches' to the safe controller, which modifies the nominal controller.

3.3 Extended CBF Case

An important note is that when working with models beyond the integrator (i.e. the unicycle and bicycle model), it does not make physical sense to control the vehicle's velocity. Rather, it makes more physical sense to control the vehicular acceleration, as this quantity can be directly controlled via force through Newton's Second Law. Therefore, $h(x)$ is in this case implemented as an input to the vehicle's acceleration. However, controlling the vehicle's acceleration in this way is not enough to guarantee safety critical behavior. Solving the system analytically, it is seen that this is because $b(x)$ in (10) will now always equal 0. When this occurs, the control input u has no effect on safety as defined with $h(x)$, so the optimal controller gives a controller with singularity. A way to resolve this is to"extend" the control barrier safety function. The extended safety function $h_e(x)$ $[m]$ is shown in (11).

$$
h_e(x) = \frac{\partial h(x)}{\partial x} \cdot f(x) + \alpha(h(x)) \tag{11}
$$

This definition comes from (6) - setting $b(x) = \frac{\partial h(x)}{\partial x} \cdot g(x) = 0$ and making the right side of the equation equal to 0 yields the exact expression for $h_e(x)$. The vehicle is now safe if both $h_e(x)$ and $h(x)$ are greater than 0. The extended barrier can be implemented into the vehicle model in the same way as the unextended barrier. Replacing $h(x)$ with $h_e(x)$, $k_s(x)$ for the extended case can be derived in the same way as with the unextended case. See **3.2 Implementation of Control Barrier Function Framework**.

4 Implementation of Various Vehicle Models

This section will provide a more in-depth discourse on the each of the three vehicle models (integrator, unicycle, and bicycle) implemented and analyzed through this project. Each vehicle model will be characterized mathematically through the model-specific form of (1) and the nominal inputs, $k_n(x)$ used in each model will be discussed. This section will also outline important model parameters examined through the project and show that each model is capable of producing desirable safety-critical control.

4.1 Integrator Model: Unconstrained Movement

The simplest vehicle model is the integrator, or 'point'. This model approximates the vehicle as a point in space that is capable of omnidirectional movement. Therefore, the integrator model is the simplest way to model a vehicle - the point is unconstrained by any real world vehicle dynamics. As such, this model can be characterized fully by just longitudinal and lateral position. The general 2D system for this point model can be expressed as (12).

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_{s1} \\ k_{s2} \end{bmatrix} \tag{12}$$

Here, x_1 $[m]$ is the longitudinal position and x_2 $[m]$ is the lateral position. As the vehicle dynamics are fully controlled by model inputs over two states, the integrator system is relatively simple to control and to solve analytically.

A sample nominal controller that could be implemented for this model is given in (13).

$$k_n(x) = -k_p \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} x_{g1} \\ x_{g2} \end{bmatrix} \right) \tag{13}$$

Here, the x vector denotes the vehicle position in 2D space, and the x_g vector gives a goal position. The objective of the nominal controller is to reach the goal position - as shown through the equation, it will command a non-zero velocity towards the goal until it reaches the goal and velocity becomes 0. Because the controller is commanding velocity here, the units of $k_n(x)$ are $[m/s]$. The k_p is a dimensionless gain parameter that defines the convergence rate of the controller - i.e. larger k_p results in a higher vehicle velocity and faster convergence.

A visual representation of desired safe behavior for the safety-critical controller implemented through this model is shown in **Figure 3**.

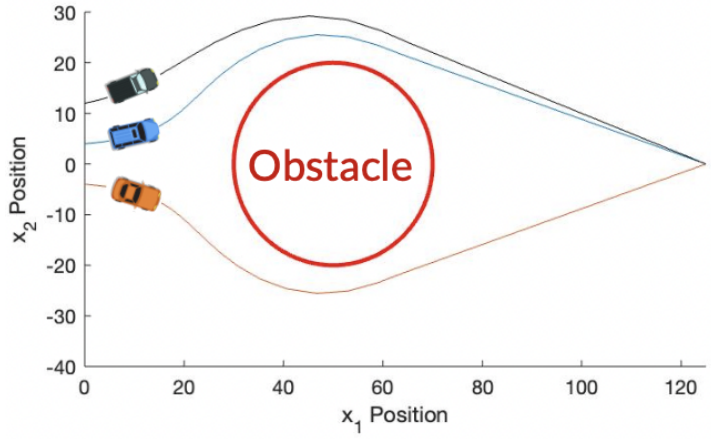

Figure 3: Simulation result from MatLab of integrator model 'vehicles' in a safety critical scenario. Obstacle of radius = 20 m is centered at (50,0). Vehicles start from $x_1 = 0$, $x_2 = -4, 4, 12$. Goal point is set at $(125, 0)$, $\alpha = 1$, and $k_p = 1$. The controller exhibits desirable behavior, with vehicles driving around the obstacle and then to the goal point.

4.2 Unicycle Model: Pivot in Place Constraint

A level of complexity higher than the integrator model is the unicycle model. In this case, the vehicle is approximated as a single wheel that is constrained in the sense that it can no longer move freely in the direction transverse to the wheel- instead, the vehicle must turn to change direction. However, because the unicycle is modeled as a single wheel, the vehicle can pivot in place. Therefore, a heading angle state, representing the angle the vehicle makes with the longitudinal axis, must be defined and controlled. A diagram of the model is shown in **figure 4**.

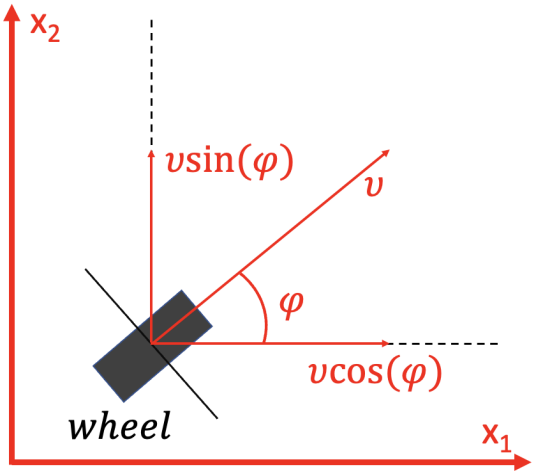

Figure 4: Diagram of the unicycle single wheel model case. As shown, the direction of the wheel can be represented via heading angle φ $[rad]$. The velocity v of the wheel can be separated into velocity in the longitudinal and lateral directions.

4.2.1 Introduction of Heading Angle and The Need for an Extended Barrier Model

As mentioned above, a heading angle state must be introduced into the unicycle model to define the unicycle's direction. Just as vehicle position is controlled by velocity, the heading angle can be controlled by angular velocity, ω $[rad/s]$. Therefore, a simplistic approach would be to define the system as (14).

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} cos(\varphi) & 0 \\ sin(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(x) \\ \omega(x) \end{bmatrix} \tag{14}$$

Here, (x_1, x_2) is the vehicle position and x_3 is the heading angle φ. The (x_1, x_2) states are controlled by the longitudinal and lateral components of v, velocity, respectively and the x_3 state is controlled by angular velocity ω. However, an outcome of this approach is that the x_3 component of $b(x)$ as defined in (9) will be 0. Therefore, the safety critical angular velocity input of the vehicle will be identical to the nominal angular velocity input of the vehicle. This results in a system that can stop at an obstacle, but cannot swerve to avoid an obstacle, as shown in **figure 5**.

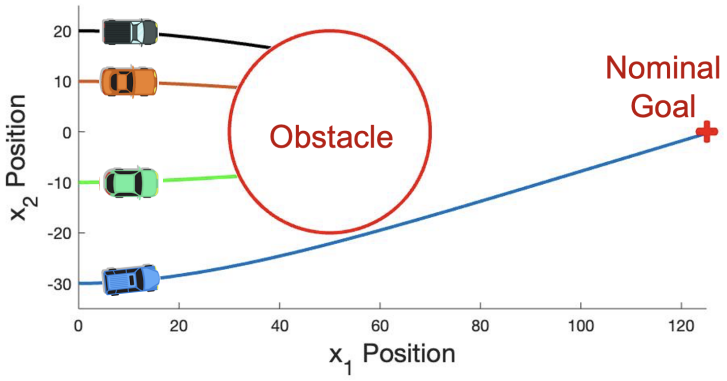

Figure 5: Diagram of the unicycle single wheel model with system defined in (14). As shown, vehicles move to the goal point as expected and break to prevent obstacle collision, but the vehicles do not change their direction to avoid the obstacle.

There are two shortcomings of this model. The first shortcoming is that the angular velocity input in this case does not have any effect on vehicle safety- i.e. the car is unable to safety steer. This problem can be resolved by extending the safety barrier function. The second is that the controller is directly commanding the vehicle velocity. This is not really achievable in the real world vehicle case. Acceleration (which can be controlled via force) can thus be used instead as a controller input to increase the accuracy of the model, as shown in (15).

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} x_3 \cos(\varphi) \\ x_3 \sin(\varphi) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(x) \\ \omega(x) \end{bmatrix} \tag{15}$$

In (15), x_3 is now the vehicle velocity and x_4 is the heading angle. Note that the states now under direct command of the controller are x_3 via $a(x)$ $[m/s^2]$, the vehicle's acceleration, and x_4 via $\omega(x)$, the vehicle's angular velocity. Note that in this new model case there is no direct control over the velocity, which is why velocity components are now in the model $f(x)$ matrix. Implementing an extended barrier with this model will allow the vehicle to both break and swerve when encountering an obstacle.

4.2.2 Shift from Controller 'Goal Point' to 'Goal State'

Another necessary change when transitioning from the integrator to the unicycle model was in the nominal controller: the shift from reaching to a 'goal point in 2D' to reaching to a 'goal velocity and lateral position'. This change was needed because unlike with the integrator, the unicycle is not capable of omnidirectional movement. Therefore, if the unicycle misses the goal point even slightly, it is very difficult to correct for this trajectory, which will cause the unicycle to have

some infinite periodic motion about the goal point. To address this problem, the unicycle nominal controller was developed as shown in (16).

$$\begin{bmatrix} k_{n1} \\ k_{n2} \end{bmatrix} = \begin{bmatrix} k_3(x_{3g} - x_3) \\ k_2(x_{2g} - x_2) - k_4 \sin(x_4) \end{bmatrix} \tag{16}$$

As shown, k_{n1} $[m/s^2]$ is the acceleration input. The input will change the vehicle velocity until the velocity, x_3, becomes the goal velocity, x_{3g}, at which point the input will become 0. Similarly, k_{n2} $[rad/s]$ is the angular velocity input. The input will change the vehicle lateral position, x_2, until it reaches the lateral goal, x_{2g}, and will change the heading angle until it reaches 0. The heading angle goal is defined to be 0, as anything else would conflict with the lateral goal. The parameters k_2, k_3, k_4 are dimensionless gain parameters that define the weight that the controller places in pursuing each goal. A higher gain means that the controller will pursue the corresponding goal more heavily.

A visual representation of desired safe behavior for the safety-critical controller implemented through this model is shown in **Figure 6**.

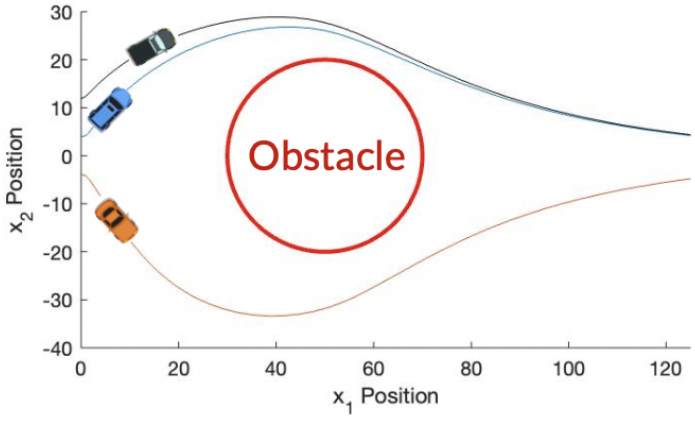

Figure 6: Simulation result from MatLab of unicycle model outlined in **eq.16,17** in a safety critical scenario. Obstacle of radius = 20 m is centered at (50,0). Vehicles start from $x_1 = 0$, $x_2 = -4, 4, 12$. Goal state is set to $x_{2g} = 0$ and $x_{3g} = 5$, $\alpha = \alpha_e = 0.2$, $k_2 = 0.01$, $k_3 = 1$, $k_4 = 0.5$. The controller exhibits desirable behavior, with vehicles driving around the obstacle and then moving towards the goal state.

4.3 Bicycle Model: Front Wheel Steer Constraint

The third vehicle model examined in this project was the bicycle model - i.e. two wheels joined by a beam - which is one level of fidelity higher than the unicycle model. Compared to the unicycle, the bicycle has the additional kinematic constraint that it cannot pivot in place. Instead, the front wheel of the bicycle

will steer by making some steering angle, γ $[rad]$, relative to rest of the vehicle. This front wheel steering along with a nonzero velocity is what changes the direction of the vehicle, which is still expressed as heading angle, φ, relative to the longitudinal direction. A diagram of the model is shown in **figure 7**.

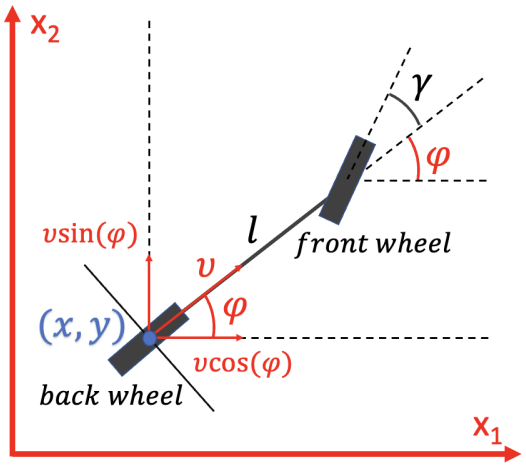

Figure 7: Diagram of the bicycle two wheel model case. The direction of the vehicle can be represented via heading angle φ while the direction of the front wheel relative to the rest of the vehicle can be represented via steering angle γ. The velocity v of the vehicle can be separated into velocity in the longitudinal and lateral directions. The length of the bicycle is l, and the position of the vehicle is measured from the center of the back wheel.

The bicycle model can be expressed mathematically as (18).

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} x_3 \cos(\varphi) \\ x_3 \sin(\varphi) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \frac{x_3}{l} \end{bmatrix} \begin{bmatrix} a(x) \\ \tan(\gamma) \end{bmatrix} \tag{17}$$

Where x_1 and x_2 denote the position of the rear wheel, x_3 is the vehicle velocity, x_4 is the heading angle, γ is the front wheel steering angle relative to the rest of the vehicle, and l is the vehicle wheelbase. Note that the bicycle model in (17) is identical to the unicycle in (16) except in the definition of x_4. This difference is to account for the fact that the vehicle can no longer pivot in place, and turning will now be dependent on vehicle speed, length, and steering angle.

Similarly, the nominal control implemented for the bicycle model is identical to the nominal control implemented for the unicycle model given in (16), substituting $\tan(\gamma)$ $[rad]$ for $\omega(x)$.

A visual representation of desired safe behavior for the safety-critical controller implemented through this model is shown in **Figure 8**.

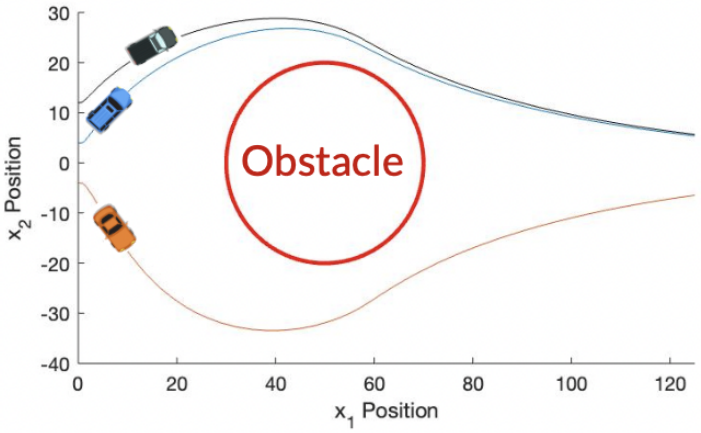

Figure 8: Simulation result from MatLab of the bicycle model outlined in (17),(16) in a safety critical scenario. Obstacle of radius = 20 m is centered at (50,0). Vehicles have a wheelbase of $l = 2.5$ m and start from $x_1 = 0$, $x_2 = -4, 4, 12$. Goal state is set to $x_2 = 0$ and $x_3 = 5$ $[m/s]$, $\alpha = \alpha_e = 0.2$, $k_2 = 0.01$, $k_3 = 1$, $k_4 = 0.5$. The controller exhibits desirable behavior, very similar to the analogous case for the unicycle model.

5 Characteristic Limiting Behaviors of Controller

An important result of this project is the identification of general limiting behaviors of the controller, in order to characterize when the controller both is and is not effective in delivering desirable safety-critical driving. In particular, three behaviors of interest were of interest. First, initial conditions that do not satisfy the safety goal are examined to identify what conditions can cause unsafe behavior in the controller. Second, controller switching position between nominal and safety control as a function of parameters is examined to characterize what parameter conditions are suitable for real world scenarios. Third, vehicle freezing as a function of parameters is examined to identify what conditions cause this safe but undesirable behavior.

5.1 Initial Conditions That do not Satisfy the Safety Goal

Recall that the controller will guarantee safety if x remains within the safe set S as defined in (4) for all time, and this is true if $x(0) \triangleq x_0$ is within the safe set. Simply put, this means that there must be $h(x_0)$ and $h_e(x_0) \geq 0$. If this condition is not satisfied, the controller will misjudge the obstacle, and there will be a collision. A sample scenario that shows simulated vehicle trajectories when the condition is not satisfied is given in **figure 9**.

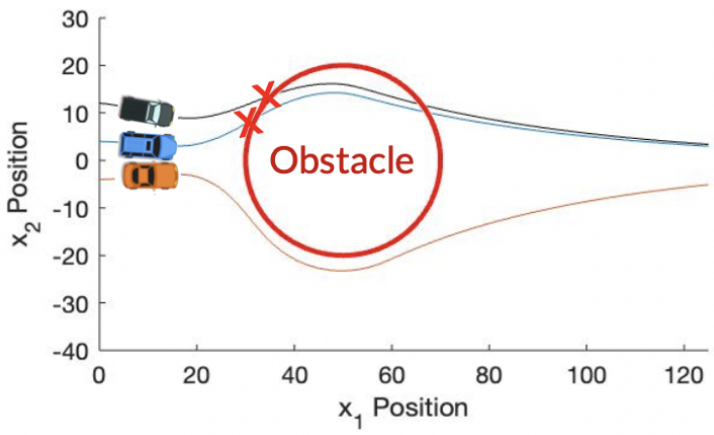

Figure 9: Simulation result from MatLab showing unsafe vehicle trajectories of vehicles under bicycle model. The projected trajectories show that the controller misjudges the obstacle and the top two vehicles would collide with the obstacle. This situation is identical to the desirable case in **figure 8**, except for vehicle initial velocity, v_0. Unlike in **figure 8** where $v_0 = 0$, in this figure v_0 is 15 m/s, which is an initial condition that violates $h_e(x_0) \geq 0$.

Notice that this limiting behavior should never be a problem in the unextended barrier case. This is because $h(x_0)$ will only ever be negative if the vehicle starts inside of the obstacle, and this should not happen as it would mean that the vehicle is already in a collision. The more relevant case is with the extended barrier case, $h_e(x)$. Note from (11) that the $\alpha(h(x))$ portion of $h_e(x)$ should always be positive, as α is a positive constant. Therefore, the portion of the equation of concern is $\frac{\partial h(x)}{\partial x} \cdot f(x)$. Solving analytically, this expression can be written as (18) for the unicycle and bicycle models.

$$\frac{\partial h(x)}{\partial x} \cdot f(x) = \frac{x_3(\cos(x_4)(x_1 - x_{o1}) + \sin(x_4)(x_2 - x_{o2}))}{d} \tag{18}$$

Here, d $[m]$ is the distance from the vehicle to the obstacle as defined in (3). Based on the goal state for the unicycle and bicycle model, it makes sense that the relevant safety-critical case is when the vehicle approaches an obstacle in the x_1 direction and x_4 $[rad]$ is closer to 0 than $\frac{\pi}{2}$. Therefore, the sum of the cosine and sine term in the parentheses should be negative at initial time. Finally, from this it can be seen that if vehicle initial velocity is sufficiently large, this can cause $h_e(x_0) < 0$.

5.2 Controller Switching Positions

Recall that the controller is engineered such that the model will operate under nominal control when the vehicle is not in a safety critical scenario and will switch into safety-critical control when it approaches an obstacle. This switching

position is useful because it characterizes the aggressiveness of the controller and helps define what parameter conditions are appropriate for a safety situation. The switching position will also put bounds on what can be a feasible controller input for real world vehicles. Note from (10) that the switch will occur when $\frac{-c(x)}{||b(x)||^2} = 0$. Contours of controller switching position can be characterized by solving the inequality and plotting in MatLab.

A simple situation can be used to characterize the dependency of controller switching position on parameters and vehicle states. Consider a 1D version of the case of the integrator model defined in **eq.13,14**, where there is only driving in the x_1 direction. Assume that the nominal controller wants to drive the vehicle to $x_{1g} = 10\ [m]$ but there is a wall at $x_1 = 5\ [m]$, so safety-critical control must take over to prevent the vehicle from colliding with the wall. The vehicle switching position can be characterized by **figure 10**.

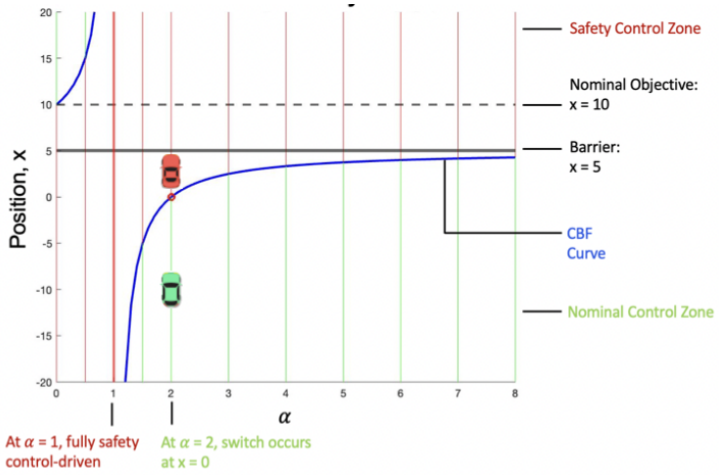

Figure 10: Pictured is the switching position of the integrator with respect to tuning parameter α. The general result is that as α increases, the controller switches to safety-critical control later. This places bounds on what α can be, as it takes time for a real vehicle to break. When $\alpha < 1$, the vehicle will operate under safety-critical control for all relevant space ($x_1 < 5$).

Compared to the unextended barrier case in **figure 10** where the only state that switching position is dependent on is position, extending the barrier will place an additional dependency on the velocity state. This is shown in **figure 11**, which examines the same simple situation as described at the top of this subsection.

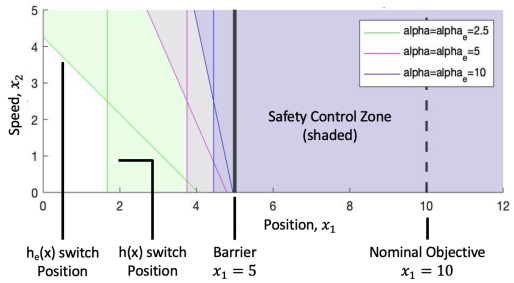

Figure 11: The figure shows that the switching position for the unextended case ($h(x)$) is only dependent on α parameter values. In the extended case ($h_e(x)$), the switching position is dependent on both α parameter values and vehicle velocity. This result is general to any of the controllers analyzed in this paper.

The switching case for 2D models is more complex but is dependent on the same parameters and states as the analogous 1D case. In the 2D case, there will be zones in 2D space where the safety control is active. Similar to the 1D case, these zones will decrease in size as α increases, meaning that the controller will be more aggressive. A characterization of switching in the 2D integrator model case is shown in figure 12.

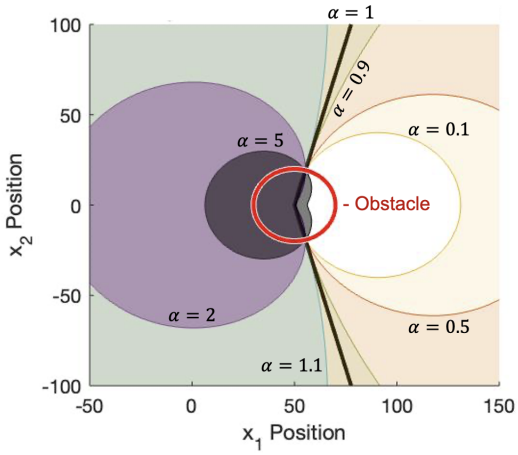

Figure 12: Switching position for the unextended 2D integrator case. Similar to the 1D case, switching position for the 2D case is dependent only on α, with higher α resulting in more aggressive control. This result is general to all 2D controllers analyzed. At some α (1 in this case), there will be a transition from most of 2D space being under nominal control to being under safety control. This transition seems general to the 2D integrator case, but it has not been determined whether this also extends to the unicycle and bicycle model case.

The switching contours for the 2D unicycle and bicycle case are not characterized in this project because it is difficult to find an analytical solution. However, **figure 13** gives a general idea of these contours by solving the inequality $\frac{-c(x)}{||b(x)||^2} = 0$ for the unicycle and bicycle model numerically in MatLab for various initial conditions.

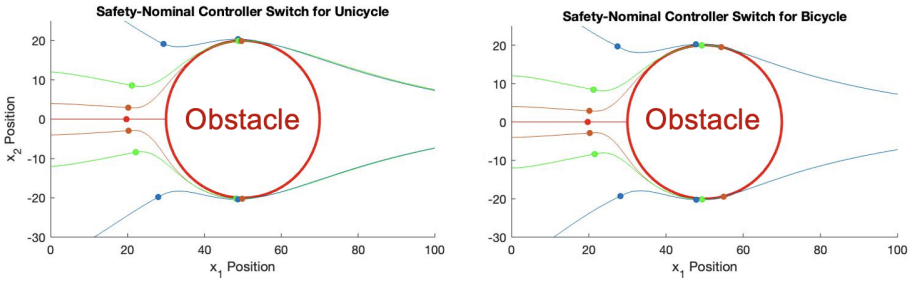

Figure 13: The figure shows switching position for the extended unicycle and bicycle cases for multiple initial conditions under the same parameter conditions. As shown, the switching positions for the unicycle and bicycle are very similar and seem to match the switching curve given in **figure 12**. More switching points or an analytical solution would be needed to fully characterize these curves.

5.3 Vehicle Freezing Cases

Vehicle freezing describes the limiting behavior whereby the controller causes the vehicle to drive to the obstacle and then freeze, instead of driving around the obstacle. This behavior is still safe because the vehicle and obstacle do not collide, but it is not desirable, as the goal is for the vehicle to be able to avoid obstacle and then continue driving.

Freezing can occur in three cases. The first case is when the vehicle drives exactly towards the center of the obstacle. In this case, it is not preferable to swerve one way or the other, so the vehicle will simply freeze because the controller does not know which way to turn. Mathematically, this happens because the $b^T(x)$ vector from (10) will point in the exact opposite direction to the vehicle's velocity. Therefore, this vector will cancel out the vehicle's velocity instead of telling the vehicle to change direction.

The second case where freezing is prominent is when the obstacle case is severe. This can either be when the vehicle's initial position is very close to the obstacle or when the obstacle is very large. In the first instance, a very large turn would be needed to swerve around the obstacle (significant divergence from heading angle goal). In the second instance, the vehicle would need to travel a large lateral distance (significant divergence from lateral goal).

The third case where freezing is prominent is when the controller is engineered with specific gain parameter conditions. Particularly, this project explored freezing as a function of angular gain k_4 and lateral gain k_2. Velocity

gain k_3 is held constant at 1, as this is representative of what is typical for a real world vehicle and changing k_3 has complex consequences on vehicle dynamics. Generally speaking, increasing a gain value increases the weight that the model puts into attaining the corresponding state goal and makes the model less willing to deviate from that goal state. This third case is interesting because it shows that whether freezing occurs is dependent on controllable parameters. An instance of freezing caused by parameter conditions is shown in **figure 14**.

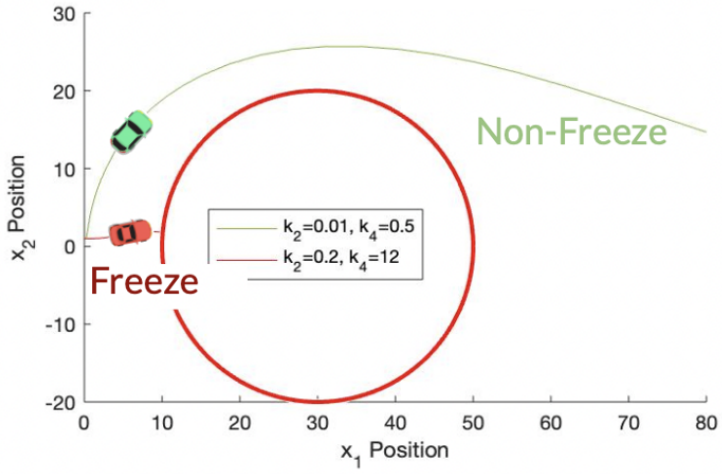

Figure 14: Bicycle model with vehicle starting at (0,1) and obstacle of radius = 20 m centered at (30,0). For both trajectories shown, $\alpha = \alpha_e = 0.2$, $l =$ 2.5 m, and $k_3 = 1$. The only difference between the two trajectory cases is the tuning of k_2 and k_4 as stated in the figure. As shown, the red car exhibits undesirable freezing at the obstacle while the green car exhibits desirable safety critical driving.

Simulating the scenario in **figure 14** for a range of k_2 and k_4 conditions results in the freezing map shown in **figure 15**.

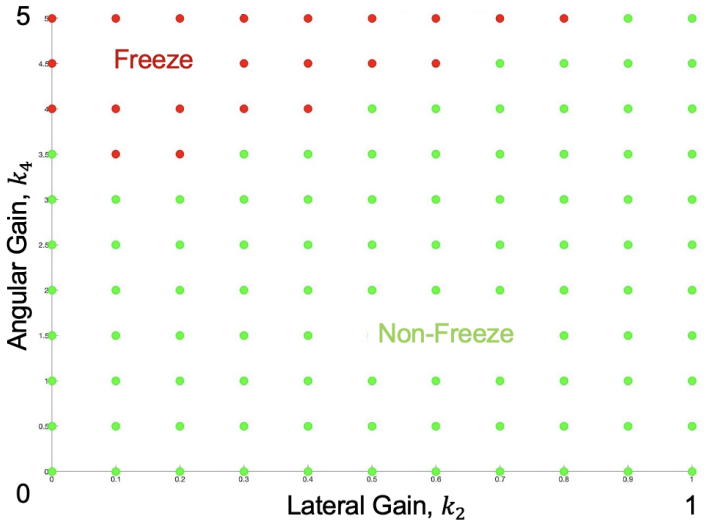

Figure 15: This map corresponds to the same model and scenario as described in **figure 14**. Vehicle freezing or lack thereof was recorded for k_2 from 0 to 1 and k_4 from 0 to 5. These ranges are meant to reflect reasonable bounds for a real world vehicle. Freezing was more prominent for greater k_4 and was most prominent for k_2 from 0.1 to 0.2. Freezing in the unicycle model followed these same trends, but was less prominent due to the unicycle's ability to pivot in place. There was no freezing observed in the integrator model, perhaps because the integrator is capable of omnidirectional movement.

The result that freezing becomes more prominent as k_4 increases is intuitive - a greater k_4 means that the vehicle is less willing to turn. However, it was less intuitive that freezing is most prominent when k_2 is between a certain value range. The shape of the freezing boundary in **figure 15** was general to all cases analyzed. As stated, freezing in the unicycle followed the same trends, but was less prominent in comparison to the analogous bicycle case. This is shown in **figure 16**.

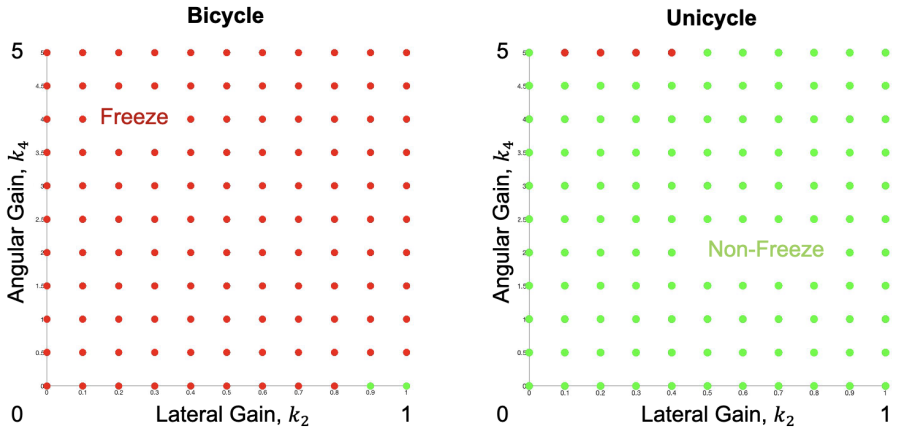

Figure 16: These maps compare freezing between the bicycle and unicycle for an analogous case between the two models. As in **figure 15**, the vehicle started at (0,1), with $k_3 = 1$ and $\alpha = \alpha_e = 0.2$. The wheelbase was 2.5 $[m]$ for the bicycle. However, unlike the case in **figure 15**, the obstacle of radius 20 $[m]$ was centered at (24,0). As shown, this is a very severe obstacle case for the bicycle, and freezing occurs a majority of the time under these conditions. Conversely, this is not a severe obstacle case for the unicycle and freezing occurs a small fraction of the time.

6 Discussion of Findings

This paper has demonstrated that a CBF framework applied to integrator, unicycle, and bicycle vehicle models can create desirable control with theoretical safety guarantees for V2V connected vehicles. By simulating and evaluating the controller under varying safety critical obstacle cases, initial state conditions, and tuning and gain parameter conditions, limiting controller behaviors are identified that characterize when the framework delivers desirable control, or lack thereof. The hope is that these findings serve as a base for further investigation of the application of control barrier functions on models of higher fidelity beyond the bicycle case and complex obstacle cases beyond the stationary single obstacle.

6.1 Limitations to Desirable Safety Critical Control

While the vehicle will by definition be safe if the system remains within the safe set S, the safety goal defined in (5) may be violated if the system's initial conditions result in $h_e(x_0) < 0$. Due to the dependence of $h_e(x)$ on velocity, this can occur if the vehicle is given a sufficiently large initial velocity. This limitation can theoretically be offset by increasing α because the term $\alpha(h(x))$ in $h_e(x)$ is always positive, but this is not realistic as increasing α beyond

a certain value range will make the controller too aggressive for a real world vehicle. As mentioned, this is a limitation to simulating a more true to life vehicle case.

Characterizing the vehicle's switching position between nominal and safety critical control via a simple 1D integrator case provides general insights on how switching position is dependent on controller parameters and states. Within the relevant control space, the controller will switch to safety control closer to the obstacle given higher α and α_e values. While switching position in the unextended barrier case is only dependent on α, switching in the extended barrier case is also dependent on the vehicle velocity. Higher velocity results in earlier switching position. Results from this paper seem to suggest that the switching zones for the unicycle and bicycle model are similar to those for the integrator model, but more complete results for the unicycle and bicycle models would be needed to verify this. Finding an analytical solution to the switching position for these models would be needed to fully characterize this behavior. Understanding switching behavior is important for engineering an appropriate controller based on how aggressive or conservative the controller should be. Further, while there is no theoretical upper bound for α, there are upper bounds for a real world vehicle as a real vehicle cannot break or swerve over a very short distance and thus may not react in time if given a large α.

Vehicle freezing is a behavior that is safe but not desirable. In the unicycle and bicycle models, this behavior becomes more prominent when the controller is engineered with specific parameter conditions. Particularly, freezing becomes more prominent with higher angular gain and is most prominent when lateral gain is between 0.1 to 0.2. No freezing behavior was observed in the integrator model for the ranges of angular and lateral gains tested. This may be because the integrator is capable of omnidirectional movement unconstrained by real vehicle dynamics. More work is needed to understand mathematically why freezing occurs in the unicycle and bicycle, and why the aforementioned parameter conditions are particularly conducive to freezing. An analytical solution to describe freezing boundaries such as the one seen in **figure 14** would be useful for fully defining this behavior.

6.2 Conditions for Desirable Safety Critical Control

The integrator model delivered theoretically desirable safety critical driving for all cases tested. Freezing was not observed in the integrator model, even when the vehicle was placed along the edge of a very large obstacle. This may be because the integrator model's motion is unconstrained by real vehicle dynamics. In both the unicycle and bicycle case, the control barrier was extended to enable safety critical control to command vehicle steering. Conditions in the unicycle and bicycle case were engineered to be representative of a real world car - namely, $\alpha = \alpha_e = 0.2$, $k_2 = 0.01$, $k_3 = 1$, and $k_4 = 0.5$. The goal velocity is 15 m/s and the bicycle is defined as 2.5 m in length. Though a real car would likely have nonzero initial velocity, the controller is given a zero initial velocity because a sufficient initial velocity (beyond 5 m/s in this case) would cause ini-

tial conditions to violate the safety goal. With these conditions, the controllers delivered theoretically desirable safety critical driving where the vehicle successfully swerved around the obstacle for all cases tested except when vehicle initial position was very close to the obstacle, which caused freezing. The behavior of the bicycle and unicycle were very similar for analogous situations, except that the bicycle was more prone to freezing because the bicycle is unable to pivot in place.

7 Conclusions

This paper has examined the application of a control barrier function framework to engineer safety-critical control for V2V connected automated vehicles. Particularly, a V2V safety-critical controller was developed via a CBF framework and applied to models of increasing fidelity, namely an integrator, unicycle, and bicycle vehicle model. This paper first demonstrated that a CBF framework applied to these three models can create desirable control with theoretical safety guarantees for V2V connected vehicles. Then by simulating and evaluating the controller under varying safety critical obstacle cases, initial state conditions, and tuning and gain parameter conditions, several limiting controller behaviors were identified that characterize when the application of this framework does and does not deliver desirable control. It is shown that these limiting controller behaviors place bounds on the scope in which the current CBF framework is effective and have implications on what is achievable for a real world car. However, more work would be needed to better characterize these behaviors - namely freezing and switching behaviors for the unicycle and bicycle model. This paper is intended to serve as a stepping stone for future investigation into the implementation of control barrier functions in vehicle models of high fidelity under more complex obstacle cases - such as multiple and dynamic obstacles.

8 Acknowledgements

Thank you to Dr. Gabor Orosz, Dr. Tamas Molnar, and Anil Alan for making this project possible through your mentorship throughout the semester and investment in this work. Thank you to Rachel Armstrong-Ceron, Dr. Jason McCormick, Dr. Eric Johnsen, and the rest of the Engineering Honors faculty and RISE faculty for the opportunity to distribute this work.

9 References

[1] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier func- tions with applications to multi-robot systems," IEEE Control Syst. Lett., vol. 1, no. 2, pp. 310–315, Oct. 2017.

[2] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in Proc. Robot. Sci. Syst., 2017.

[3] A. Ames, J. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in Proc. 53rd Conf. Decis. Control, 2014, pp. 6271–6278.

[4] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames. Safe controller synthesis with tunable input-to-state safe control barrier functions. IEEE Control Systems Letters, 6:908-913, 2022.

[5] Q. Nguyen and K. Sreenath, "Optimal robust safety-critical control for dynamic robotics," 2020. [Online]. Available: arXiv:2005.07284.

[6] T. G. Molnar, R. Cosner, A. Singletary, W. Ubellacker, and A. D. Ames, "Model-Free Safety-Critical Control for Robotic Systems", IEEE Robotics and Automation Letters, Vol. 7, No. 2, 2022

[7] A. D. Ames, and P. Tabuada. Lectures on Nonlinear Dynamics and Control. 2022