Is Your Commute Driving you Crazy? A Study of Misbehavior in Vehicular Platoons

Bruce DeBruhl, Sean Weerakkody, Bruno Sinopoli, and Patrick Tague
Carnegie Mellon University
{debruhl@, sweerakk@andrew., brunos@ece., tague@}cmu.edu

ABSTRACT

Traffic is not only a source of frustration but also a leading cause of death for people under 35 years of age. Recent research has focused on how driver assistance technologies can be used to mitigate traffic fatalities and create more enjoyable commutes. In this work, we consider cooperative adaptive cruise control (CACC) or platooning, a driver assistance technology that controls the speed of vehicles and inter-vehicle spacing. CACC equipped cars use radar to fine tune inter-vehicle spacing and dedicated short-range communication (DSRC) to collaboratively accelerate and decelerate. Platooning can reduce fuel consumption by over 5% and increases the density of cars on a highway. Previous work on platooning has focused on proving string stability, which guarantees that the error between cars does not grow with the length of a platoon, but little work has considered the impact an attacker can have on a platoon. To design safe distributed controllers and networks it is essential to understand the possible attacks that could be mounted against platoons.

In this work, we design a set of insider attacks and abnormal behaviors that occur in a platoon of cars. For example, we introduce the collision induction attack where an attacker exploits the platoon controller to cause a high-speed accident with the car following it. To mitigate these insider attacks we design a model-based detection scheme that leverages the broadcast nature of DSRC. Each car uses DSRC messages from other cars in the platoon to model the expected behavior of the car directly preceding it. If the expected behavior and actual behavior differ the monitoring vehicle switches to non-cooperative ACC, relying solely on radar, to mitigate the impact of the attack. We show that our detection scheme is able to detect many of our proposed insider attacks and when combined with a well designed ACC controller can avoid collisions. We propose combining our detection scheme with a global reputation scheme to detect when a car is malicious or needs maintenance.

WiSec'15, June 22–26, 2015, New York, NY, USA.

ACM 978-1-4503-3623-9/15/06 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766498.2766505.

Keywords

Vehicular Network Security; Model-Based Detection; Vehicle Platoons

1. INTRODUCTION

Traffic is a growing source of frustration in most urban areas. Traffic is also a major source of deaths due to driver errors and inclement road conditions. The percentage of vehicle related deaths is particularly startling for people under the age of 35; according to the CDC for people in the U.S. aged 5-34, traffic accidents are the leading cause of deaths [1]. Because of this, an ever increasing body of work [7,10,14] has explored the use of autonomous and semi-autonomous driving, allowing for a car to pilot itself while the passenger inside can focus on other tasks. Various technologies have enabled the rise in autonomous driving including the declining cost of mobile computing, declining cost of reliable radar, and the dependable emergence of vehicle to vehicle (V2V) communication.

As autonomous cars are being developed we have seen an increase in the ability of smart driving features like lane-keep assist, adaptive cruise control (ACC), and blind-spot warning systems. All of these technologies do not take the driver out of the loop but allows for safer driving by assisting the driver. One increasingly popular feature is adaptive cruise control which keeps a constant-headway following distance to the preceding car by using radar. The radar's intervention allows for the car to safely maintain a constant headway and thus reduce accidents caused by insufficient following distance. Insufficient following distance is a major concern since most people drive at headway under 1 second, and almost all motorists drive under the recommended 2 second headway for human drivers [2].

The performance of ACC is limited to the vehicle's operations, in particular brake lag. Brake lag is the time it takes for a car to start decelerating after a brake signal has been received. So if an ACC-equipped car follows at a headway less then the brake lag time then a collision may occur. With the advent of vehicle to vehicle communications (V2V) we can allow ACC equipped cars to further reduce their headway. This is enabled because the delay for the V2V communications is less then the brake lag. The decreased delay allows for lines of cars to cooperatively decelerate safely while using small headway times. This type of formation driving is either called cooperative ACC (CACC) or platooning. The benefits of platooning include increased density of cars on a highway and increased fuel efficiency of platooned vehicles [8].

To date, the work on platooning has largely focused on how to design a controller that is string stable. String stability in general is the idea that error does not grow along a platoon of vehicles [14]. While preliminary work has considered string stability in ideal systems (e.g. no networked communications [14] or perfect networks [20]), recent work has explored string stability in realistic networks including networks with delays [6], packet-based networks [11], and stochastic networks [16].

There has been a limited set of work that has explored the impact of attacks on platoon controllers and V2V communications [4]. In this work, we explore what happens when one of the cars in the platoon does not behave according to the control law. Such a vehicle could be malicious, greedy, or even a malfunctioning benign vehicle. We are particularly interested in an insider attack where the attacker either uses a malignant control law or misreports information about her behavior. We introduce a set of 5 different attacks and abnormal behaviors and briefly discuss their motivation and impact on the system. One particularly devastating attack is the collision induction attack where the attacker broadcasts that she is accelerating while in reality, the attacker jams on her brakes. This attack causes the preceding car to collide at high speeds and, with high probability, results in loss of life and assets.

Given the existence of misbehavior that can be mounted in a platoon of vehicles we propose using a model based detection scheme to detect and mitigate the impact of malicious behaviors. We propose each vehicle model the expected behavior of the vehicle proceeding them using DSRC information provided from cars farther up the platoon. Vehicles can use this model to calculate the error between the modeled states and the measured state of the proceeding car. The error calculations can then be used with a simple threshold to calculate whether an abnormality exists in the system or not. The technique for modeling, calculating error, and thresholding can all be chosen in the design of the system. We summarize this approach in Figure 1.

Once an attack is detected we propose that the vehicle changes to a non-cooperative ACC protocol with an increased headway distance to guarantee safe performance. We are able to detect most abnormalities and are able to avoid the collision that would be caused by the collision induction attack using this technique. This detection scheme could be combined with a global reputation system to keep track of whether certain vehicles are often problematic.

To summarize, in this paper we make the following contributions.

- We propose a set of insider attacks that can cause unexpected behavior in platoons and may cause fatal accidents.
- We develop a platoon detection method based on up stream DSRC communications to detect misbehavior.
- We design a two state operating mode for semi-autonomous cars to safely transition to a non-cooperative cruise control when attacks are being mounted.
- We simulate the above attacks, detection, and mitigation schemes to provide a proof-of-concept.

The rest of this paper is organized as follows. In Section 2 we introduce related work and in Section 3 we introduce our system models. In Section 4 we introduce misbehaviors along with their motivation and impact. In Section 5 we introduce our detection scheme and in Section 6 we provide simulation results. In Section 7 we discuss the trade-offs and possible improvements in the detection system design. In Section 8 we conclude the paper.

2. RELATED WORK

Two approaches are commonly used to analyze formation driving in automotive systems. In microscopic models each car is modeled as a point and their interaction is analyzed while in a macroscopic model the highway system is modeled as a set of pipes with the traffic modeled as a fluid [15]. Within the domain of platooning, or linear cooperative formation driving, the design criteria most often used is string stability. String stability is roughly defined as the error between vehicles not growing as the length of the platoon grows. In this paper we focus on linear formation driving using a microscopic model.

Various approaches have been suggested for practically implementing string stable platoons. Common design assumptions include the number of radars used and the necessary communication range. For platooning, an engineer can use the assumption from non-cooperative adaptive cruise controls which uses a single front-facing radar [12], implicitly trusting the following car. A design can also use a two radar approach, one rear-facing and one front-facing, for a controller [9] which balances the distance to both the preceding and following car. In this work we use a controller with a single forward facing radar unit with an implicit trust that the following car will not rear end us.

There are two common assumptions with respect to communication range in the literature. The first is that all cars are able to hear the leader [17], implying the range from the leader to the last car is limited. This assumption forces an artificial bound on platoon size, but this bound fits into many platooning frameworks [5]. The second approach assumes only local communication from the nearest neighbors [11] and allows for platoons of arbitrary lengths.

Considerable research has explored the impact of networking on the performance of a string stable platoon of cars. Heemels et al. [6] explore the tradeoff between network delay, transmission intervals, and performance on a string stable controller. Tabbara et al. [16] introduce string stability in a platoon of cars with stochastic communications. Zhao et al. [19] explore the effect of stochastic disturbances in the vehicle dynamics and how it impacts propogates in a string stable platoon of vehicles. Segata et al. [13] have recently explored the impact of communication performance (fading and transmit power levels) and how communication systems can be designed for platoons [6].

There has been limited research on attacks on platoon controllers or attacks on v2v networks used for platooning. In [3] an attack is designed that decreases the efficiency of a platoon of vehicles. They show they are able to leverage the controller to reduce the efficiency of cars around the attacker by 20-30%. Haas [4] explored an attack on the network of a platooned system showing that jamming, with only a 50% duty cycle, can can cause accidents and platoon deformation. In this work, we continue the exploration of platooning attacks by introducing new attacks using misinformation and malicious controllers.

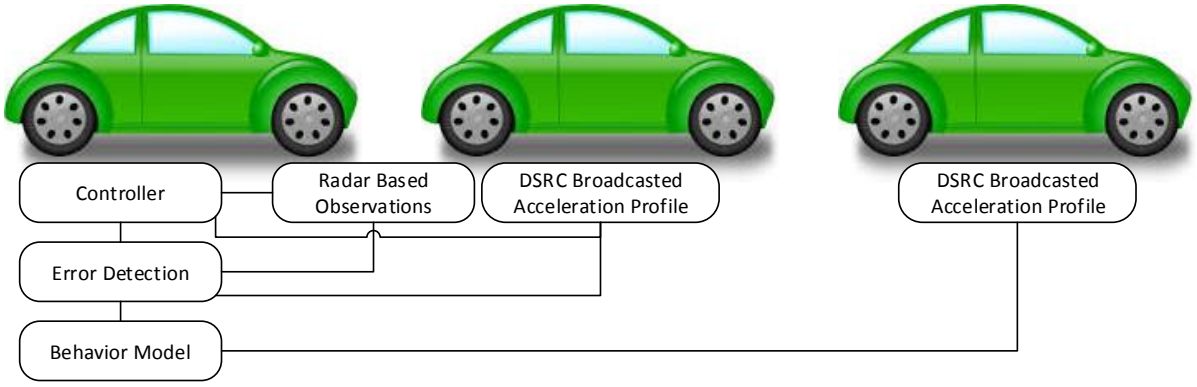

Figure 1: In this figure, we show our proposed detection scheme at a high-level. The car in the back of the platoon uses data sent via DSRC by the first car to model the expected behavior of car 2. The car then determines whether the two expected signals differ by an amount greater than a threshold.

3. SYSTEM MODEL

Our system consists of a platoon of K cars that that we number from 0 to $K-1$ with car 0 being the platoon's leader. We assume that the cars all drive in a single straight lane and that their order can not change. We indicate the spatial position, velocity, and acceleration of car i as q_i, v_i, and a_i respectively. We indicate the distance between the front bumper of car i and the rear bumper of car $i-1$ as $d_i = q_{i-1} - q_i$ with $d_0 = 0$ and the desired distance between car i and car $i-1$ as $d_{r,i}$ with $d_{r,0} = 0$. We define the error for car i as $e_i = d_i - d_{r,i}$.

The cars desire to follow a constant headway policy such that $d_{r,i} = h_{d,i}v_i + L_i$ where L_i is a constant distance offset and $h_{d,i}$ is the the desired headway of car i. We can substitute the constant headway policy into our error equations to get

$$e_i = q_{i-1} - q_i - h_{d,i}v_i - L_i. \tag{1}$$

We can set the distance $L_i = 0$ in (1) by assuming a change of basis to provide for the safe stopped distance such that $e_i = q_{i-1} - q_i - h_{d,i}v_i$.

We model the cars using a double integrator model with a lag constant of η_i for each car. Given a desired acceleration of u_i, car i has the following continuous time differential equations.

$$\dot{a}_i = -\eta_i^{-1}a_i + \eta_i^{-1}u_i \tag{2}$$
$$\dot{v}_i = a_i \tag{3}$$
$$\dot{q} = v_i \tag{4}$$
$$\dot{e}_i = v_{i-1} - v_i - h_{d,i}a_i. \tag{5}$$

3.1 Controller

In Figure 2 we show a vehicle that is equipped for cooperative adaptive cruise control. We assume the radar plus DSRC setup and use a controller that has been tested for this setup [6]. This controller uses a combination of a DSRC based feedforward input, $u_{ff,i}$, and a measurement based feedback input, $u_{fb,i}$, such that

$$u_i = u_{fb,i} + u_{ff,i}. \tag{6}$$

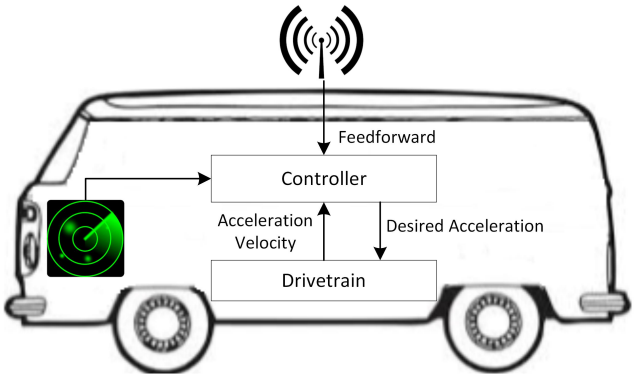

Figure 2: In this figure, we show our controller structure for a platooned vehicle. The vehicle uses radar to determine distance and error from the car in front of it, DSRC to get feedforward information from other cars, and powertrain measurements to determine its current state.

The inter-vehicle distance is measured using radar and used to calculate error which allows for PD feedback controller such that

$$u_{fb,i} = k_p e_i + k_d \dot{e}_i. \tag{7}$$

The feedforward controller is provided via DSRC using the update equation

$$\dot{u}_{ff,i} = -h_{d,i}^{-1}u_{ff,i} + h_{d,i}^{-1}\hat{u}_{i-1}, \tag{8}$$

where $\hat{u}_{i-1}$ is received via DSRC. In the case that all vehicles are behaving then $\hat{u}_{i-1} = u_{i-1}$ during update periods. However, in general, we assume this equation may not hold in order to account for malicious behavior.

It is important to note that car 0 has a unique control law such that $u_0 = u_r$ where u_r is a reference desired acceleration profile. It is assumed that car 0 is given u_r in real time so no non-casual predictions can be made. The proposed controller has been shown to be string stable in

continuous communication systems and has been tested in real networked platoons with delays and sampling [12].

3.2 System Description

We define the vector $x_i^T = [e_i, v_i, a_i, u_{ff,i}]$ for the state of car i. The update equation for a vehicle can be written as a linear system such that

$$\dot{x_i} = A_{i,i}x_i + A_{i,i-1}x_{i-1} + B_{s,i}u_i + B_{c,i}\hat{u}_{i-1},\ \forall\, i > 0 \quad (9)$$

and

$$\dot{x_0} = A_0 x_0 + B_{s,i}u_r \quad (10)$$

where

$$A_{i,i} = \begin{pmatrix} 0 & -1 & -h_{d,i} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\eta_i^{-1} & 0 \\ 0 & 0 & 0 & -h_{d,i}^{-1} \end{pmatrix}, \quad (11)$$

$$A_{i,i-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (12)$$

$$B_{s,i}^T = \begin{pmatrix} 0 & 0 & \eta_i^{-1} & 0 \end{pmatrix}, \quad (13)$$

$$B_{c,i}^T = \begin{pmatrix} 0 & 0 & 0 & h_{d,i}^{-1} \end{pmatrix}, \quad (14)$$

and

$$A_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\eta_0^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (15)$$

We define X as the state of the whole systems so that $X^T = [x_0^T, x_1^T, ...x_{K-1}^T]$. We define the inputs to the system as $U^T = [u_0, \hat{u}_0, u_1, \hat{u}_1, \ldots, u_{K-1}]$ where each vehicle chooses its input values u_i and $\hat{u}_i$. This allows us to write the linear equations for the whole system as

$$\dot{X} = AX + BU \quad (16)$$

where

$$A = \begin{pmatrix} A_0 & 0 & 0 & \ldots & 0 & 0 \\ A_{i,i-1} & A_{i,i} & 0 & \ldots & 0 & 0 \\ 0 & A_{i,i-1} & A_{i,i} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & A_{i,i-1} & A_{i,i} \end{pmatrix} \quad (17)$$

and

$$B = \begin{pmatrix} B_{s,i} & 0 & 0 & \ldots & 0 & 0 \\ 0 & B_{c,i} & B_{s,i} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & B_{c,i} & B_{s,i} \end{pmatrix}. \quad (18)$$

3.3 Discretization

We discretize the system since the controller is implemented on a digital computer using digital communications. We assume the radar has a sampling time of 1 ms and the communication system has a sampling time of 100 ms. We assume that the controller uses a sample and hold technique for the communication input variable. We thus use the update equations

$$X[k+1] = A_d X[k] + B_d U[k] \quad (19)$$

where A_d and B_d represent an exact discretized version of (16).

For cars that follow the control law we can similarly define the controller equations for car i in term of x_i and x_{i-1} as

$$u_i = k_1 x_i[k] + k_2 x_i[k-1] \quad (20)$$

where

$$k_1 = \begin{pmatrix} k_p + \frac{k_d}{.001} & 0 & 0 & 1 \end{pmatrix} \quad (21)$$

and

$$k_2 = \begin{pmatrix} -\frac{k_d}{.001} & 0 & 0 & 0 \end{pmatrix}. \quad (22)$$

The input $u_i[k]$ is updated every 1 ms while $\hat{u}_i[k]$ is updated every 100 ms and kept constant otherwise. We model this system for a platoon of 5 cars in Figure 3 where the platoon accelerates for 5 seconds, holds for 15 seconds and then decelerates for 5 seconds. The controller in this figure uses a CACC controller with a constant headway of .35 seconds and constant spacing of 1 m.

We assume homogeneous cars such that $A_{i,i-1}$, $A_{i,i}$, $B_{s,i}$, and $B_{c,i}$ are known and the same for all vehicles. This allows us to write our discrete matrices as

$$A_d = \begin{pmatrix} A_{d,0} & 0 & 0 & \ldots & 0 & 0 \\ A_{d,1} & A_{d,2} & 0 & \ldots & 0 & 0 \\ 0 & A_{d,1} & A_{d,2} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & A_{d,1} & A_{d,2} \end{pmatrix}$$

where $A_{d,0} \in \mathbb{R}^{4x4}$, $A_{d,1} \in \mathbb{R}^{4x4}$, and $A_{d,2} \in \mathbb{R}^{4x4}$. Likewise we define

$$B_d = \begin{pmatrix} B_{d,0} & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\ B_{d,1} & B_{d,2} & B_{d,3} & \ldots & 0 & 0 & 0 & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & 0 & 0 & \ldots & B_{d,1} & B_{d,2} & B_{d,3} & 0 \end{pmatrix}.$$

where $B_{d,0} \in \mathcal{R}^{4x1}$, $B_{d,1} \in \mathcal{R}^{4x1}$, $B_{d,2} \in \mathcal{R}^{4x1}$, and $B_{d,3} \in \mathcal{R}^{4x1}$.

4. ATTACK STRATEGIES

In this section, we introduce a set of attacks and abnormal behaviors that can occur in a platooned vehicular network. We discuss possible motivations for the attacks which range from rational to byzantine. This list is not comprehensive but provides a start to the discussion of what system level attacks may impact a formation of vehicles and how serious the effects of these attacks could be. For the convenience of the reader we summarize these attacks, their motivation, their potential impact, and their implementation in Table 1.

In the remainder of this section we refer to the attacker's control input and performance parameters using the letter 'a' in the subscript. Thus, u_a refers to the attacker's control signal during the attack and $\hat{u}_a$ refers to the attacker's broadcasted control signal. We assume that the attacker's signal is non-additive so the state update equation for the attacker become $x_a[k+1] = Ax_a[k] + Bu_a[k]$.

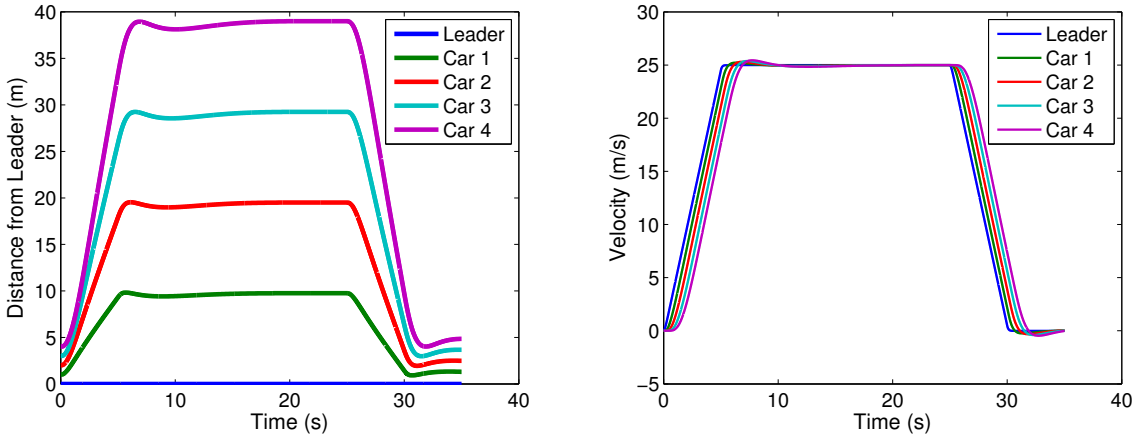

Figure 3: In this figure, we show a simulation of our system without attacks. On the left, we show a plot the distance for each car behind the lead car. On the right, we show the velocity for each of the cars.

Attack	Impact	Motivation	Method
Reduced Headway Attack	Decreased String Stability	Decreased fuel consumption Increased density	Misbehavior
Joining Without Radar	Decreased String Stability Danger in wireless congestion	Decreased cost over radar equipped car	Misbehavior
Mis-report Attack	Decreased Performance	Mistrust of the system	Misinformation
Collision Induction Attack	Collision Loss of Life Property Damage	Maliciousness Terror	Misbehavior & Misinformation
Non-Attack Abnormalities	Decreased Performance Decreased String Stability	Improper Maintenance	Misbehavior

Table 1: In this table, we summarize the system level attacks that we propose including their impact, method, and motivations.

4.1 Reduced Headway Attack

In the current highway system the majority of motorists do not follow the recommended 2 second headway speed, with many studies showing the average speed on freeways being under 1 second [2]. This attack models a similar greedy behavior where a car ignores the recommended headway speed that guarantees string stability and follows closer. A driver might, for example, follow at a headway of 0.125 second speed when the vehicles in the platoon are only string stable at headway distance greater than or equal to a 0.25 second. This attack would likely be implemented by a driver who wants to increase fuel savings by decreasing draft or a driver who manually drives with extremely small headways.

To implement this attack we change the attacker's headway parameter to $h_{d,a} < h_{d,min}$ where $h_{d,min}$ is the recommended minimum headway speed.

4.2 Joining Without Radar

This is another greedy behavior where a car attempts to become part of a platoon without having the necessary radar, or other distancing equipment. This is motivated by a driver who does not want to buy a new vehicle but retrofits a car with DSRC which, unlike radar, does not require per vehicle tuning. This attack causes the reaction of the car to be based only on the feedforward information which is dangerous if wireless congestion prevents the cars from communicating properly. This also eliminates the guarantees that are provided by string stability to the platoon of cars, increasing the risk of an accident.

This attack is implemented by changing the attacker's control law to $u_a = u_{ff,a}$ and ignoring the feedback portion of the control law.

4.3 Mis-report Attack

This is an attack that could be mounted for various reasons including not trusting the cooperative adaptive cruise control system. The attacker misinforms the vehicle that is following to increase the following car's headway or to cause a change in the following car's behavior. The attacker mounting this attack could either follow the prescribed control law or choose an alternative control law. We will assume in this work that the attacker follows the prescribed control law and only misreports its behavior so $u_a = u_i$. This attack is motivated by wanting to increase the following distance of the preceding car.

The attacker defines a mis-report percentage $\beta \in [0, 1]$ and then implements the attack by reporting $\hat{u}_a = (1 - \beta)u_a$ if $u_a > 0$ and $\hat{u}_a = (1 + \beta)u_a$ if $u_a < 0$.

4.4 Collision Induction Attack

In this attack, the attacker broadcasts an acceleration profile indicating that they are speeding up which causes the following vehicle to accelerate. The attacker actually starts

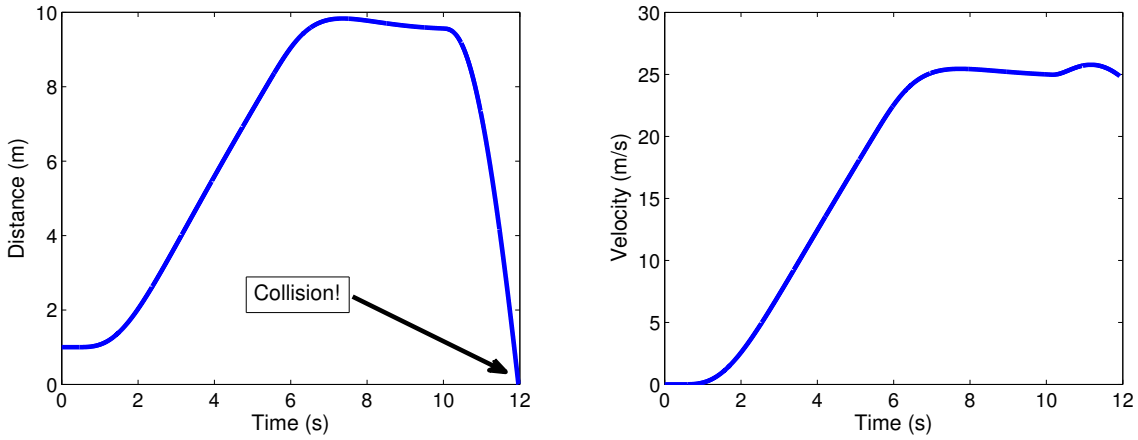

Figure 4: In this figure, we show the effect of a collision induction attack that is started at 10 seconds. On the left is a plot of the distance between the attacker and the car under attack and on the right is a graph of the car under attacks velocity. In under 2 seconds the attacking car is hit by the following car going at a speed of over 55 miles per hour.

to aggressively brake which causes the error between the attacker and following car to quickly increase. This is very likely to cause an accident at high speed which makes this attack extremely dangerous.

This is very similar to attacks that could be mounted in the current highway system. If a driver was to jam on their breaks during rush hour while being tailgated, the vehicle would likely be rear ended. If there are many cars that are all tailgating this could even result in a pile up. In Figure 4 we show this attack implemented starting at ten seconds. In under two seconds the car behind the attacking car collides with the attacker at speeds over 25 meters per second, or approximately 56 miles per hour.

Assuming that cars have a range on their inputs defined as $u_i \in [u_{min}, u_{max}]$ we can implement this attack by setting the attackers control parameters to $u_a = u_{min}$ and $\hat{u}_a = u_{max}$.

4.5 Non-attack abnormalities

Our detection method is also able to detect non-malicious abnormal behaviors in the system. For example, our detection scheme would detect if the acceleration or breaking parameters of a vehicle were to change due to normal wear on the system. This could be used in conjunction with a global monitoring system to help alert drivers when their vehicle might need maintenance.

To model abnormal driving in our system we vary the value of η_a for a vehicle that we call the attacker even though their intent may not be malicious.

5. MODEL BASED ATTACK DETECTION

In Figure 5, we show our proposed approach to detecting abnormal behavior in a platoon of cars. Our approach has every car model the expected behavior of the vehicle directly in front of them. The vehicles then compare the calculated expected behavior with the observed behavior. Using these comparisons the car is then able to detect both malicious and benign abnormalities. The ability to detect malicious as well as benign but dangerous behavior is one of the greatest strengths of our approach.

Once abnormal behavior is detected, the car switches from operating in a cooperative platoon framework to a radar only based adaptive cruise control framework where it is safe even if the preceding car is mounting an attack. We choose this very aggressive response to a potential attack for a multiple reasons. First, the potential impact of a malicious car is a high-speed traffic accident which, in the worst case, results in loss of life, and, in the best case, results in high-value property damage. This technique can also be combined with regional reputation systems to detect vehicles that frequently behave abnormally. We discuss each portion of our detection and response scheme in detail below.

5.1 Modeling Techniques

In this section, we design an algorithm for car i to model the expected behavior of car $i-1$ given the data packets from car $i-j$. We define $x_{m,i-1}$ as the modeled state of car $i-1$. We can then define the state of all the cars in the model as $X_m = [x_{m,i-j}, x_{m,i-j+1}, \ldots, x_{m,i-1}]$. Likewise, we define the feedback inputs and feedforward inputs of car $i-1$ as $u_{m,i-1}$ and $\hat{u}_{m,i-1}$ respectively. This allows us to define the inputs at each time as.

$$U_m = [u_{m,i-j}, \hat{u}_{m,i-j}, u_{m,i-j+1}, \hat{u}_{m,i-j+1}, \ldots, u_{m,i-1},]^T.$$

We can write the system update equation for the model as

$$X_m[k+1] = A_m X_m[k] + B_m U_m[k] \tag{23}$$

where

$$A_m = \begin{pmatrix} A_{d,0} & 0 & 0 & \ldots & 0 & 0 \\ A_{d,1} & A_{d,2} & 0 & \ldots & 0 & 0 \\ 0 & A_{d,1} & A_{d,2} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & A_{d,1} & A_{d,2} \end{pmatrix} \tag{24}$$

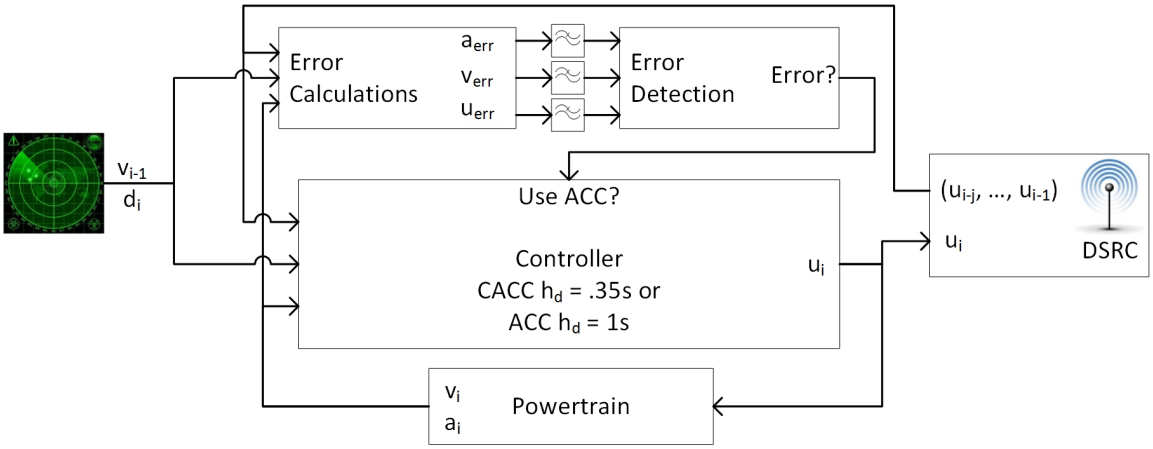

Figure 5: In this figure, we show a detailed diagram of our proposed detection scheme. A model of the expected behavior of the car in front of the monitoring car is made from the broadcasted upstream control information. This is compared to the measured behavior of the car in front of the monitoring car. If the error is larger than expected, the monitoring car switches to a non-cooperative ACC algorithm.

and

$$B_m = \begin{pmatrix} B_{d,0} & 0 & 0 & \dots & 0 & 0 & 0 \\ B_{d,1} & B_{d,2} & B_{d,3} & \dots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \dots & B_{d,1} & B_{d,2} & B_{d,3} \end{pmatrix}.$$

We assume that the cars in the model behave according the control law given in (20) with 1ms radar update times and 100ms state broadcast times.

During an update period we can use (20) to define

$$U_m[k] = \phi_1 X_m[k] + \phi_2 X_m[k-1] + \phi_3 \hat{u}_{i-j}[k] \qquad (25)$$

where

$$\phi_1 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_1 & 0 & \dots & 0 \\ 0 & k_1 & 0 & \dots & 0 \\ 0 & 0 & k_1 & \dots & 0 \\ 0 & 0 & k_1 & \dots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & \dots & k_1 \\ 0 & 0 & 0 & \dots & k_1 \end{pmatrix}, \phi_2 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \dots & 0 \\ 0 & 0 & k_2 & \dots & 0 \\ 0 & 0 & k_2 & \dots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & \dots & k_2 \\ 0 & 0 & 0 & \dots & k_2 \end{pmatrix}$$

and

$$\phi_3 = \left(1, 1, 0, \dots, 0\right)^T. \qquad (26)$$

Likewise, during a non-update period we can define our update input as

$$U_m[k] = \phi_4 X_m[k] + \phi_5 X_m[k-1] + \phi_6 U_m[k-1] \qquad (27)$$

where

$$\phi_4 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & k_1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & k_1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \phi_5 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & k_2 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & k_2 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

and

$$\phi_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}. \qquad (28)$$

We used the linear double integrator for modeling the vehicles in the system but in an actual implementation more advanced models could be used. The techniques used for modeling could be as complex as desired considering tradeoffs in accuracy, calculation cost, and time for calculation. Improvements that could be considered in the modeling include capturing non-linear behavior of the vehicles drivetrain and using terrain mapping to predict variations.

5.2 Thresholding Techniques

Once we have a model of $\bar{x}_{i-1}|\hat{u}_{i-j}$ we can then predict whether the model error is acceptable or not. We indicate the measured values for $\hat{x}_{i-1,m}$ and assume we can measure acceleration and velocity. It is not possible to measure the error since we do not have a line of site to car $i-2$.

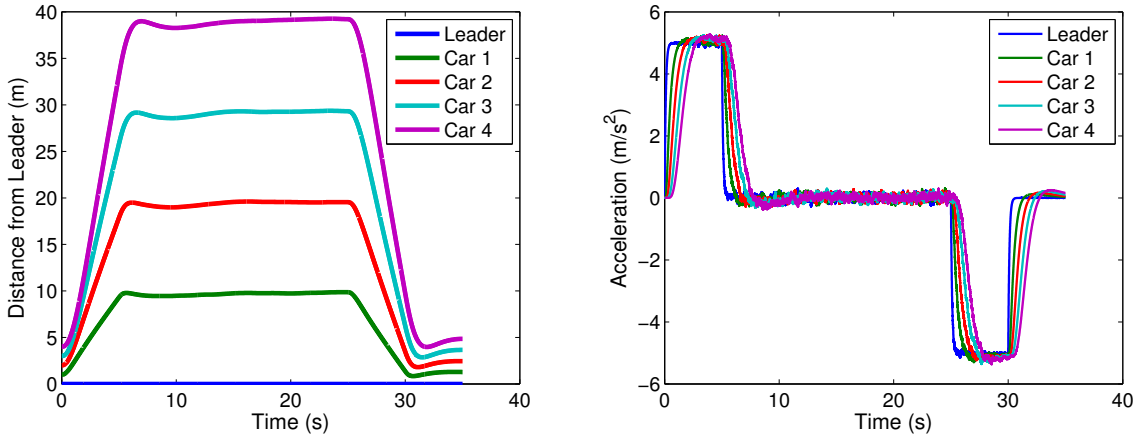

Figure 6: In this figure, we show the system operating under noisy conditions with the variance in noise set to .001 of the vehicles velocity. On the left we show the distance from the leader and on the right we show the acceleration for each of the vehicles.

We propose using the model error normalized to acceleration such that

$$(a_{err}|\hat{u}_{i-j}) = \left(\frac{(a_{m,i-1}|\hat{u}_{i-j}) - a_{i-1,}}{a_{i-1}}\right)^2 \quad (29)$$

$$(v_{err}|\hat{u}_{i-j}) = \left(\frac{(v_{m,i-1}|\hat{u}_{i-j}) - v_{i-1}}{a_{i-1}}\right)^2 \quad (30)$$

$$(\hat{u}_{err}|\hat{u}_{i-j}) = \left(\frac{(\hat{u}_{m,i-1}|\hat{u}_{i-j}) - \hat{u}_{i-1}}{a_{i-1}}\right)^2 \quad (31)$$

We use a hand-tuned constant delay between the model and measured data since the model trails the real system.

5.3 Attack Mitigation

We propose a two-state operating condition for the monitoring vehicle. The vehicle will use the CACC controller proposed in Section 3.1. When an attack is detected the control law changes to a non-adaptive cruise control law such that $u_i = u_{fb,i} = k_d\dot{e}_i + k_p e_i$ where the error is now calculated with a larger headway constant, for example 1 second. In Section 6, we show that this controller is effective at mitigating the impact of the collision induction attack, avoiding the loss of life or assets. This controller would likely cause other cars in the platoon to flag the detecting car as an attacker and result in the loss of the platoon formation.

As a design decision other reactions could be implemented to mitigate the impact of abnormal behavior. For example a control law could be designed where the headway is proportional to the amount of error in the model and actual system. We leave the design of more response techniques as an implementation decision.

5.3.1 Global reputation system

The ability to detect malicious and abnormal behavior can be combined with a global system to keep a reputation for vehicles. If a car continually gets flagged then it would be flagged for investigation by authorities. Similarly, a car that is often flagged could run a diagnostics routine to determine if it has a system failure as proposed in various works on reputation systems [18].

6. SIMULATIONS

We simulate our attacks as well as the detection scheme to provide a proof-of-concept in a five car platoon. We use the following parameters for our platoon: $\eta = 0.1$, $h_d = 0.35s$, $k_p = 0.2$, $k_d = 0.7$, and $K = 5$ cars. We set the sampling time for the radar at $T_s = 0.001s$ and assume that the update for the feedforward information occurs every $100ms$. For each trial we assume that the lead car in the platoon accelerates from standstill for 5 seconds at a constant rate, maintains the maximum speed for 20 seconds, decelerates at a constant rate for 5 seconds, and remains at rest for 5 seconds. We do not make assumptions on the acceleration rate used in the test. We assume a model delay of $250ms$, which was determined by empirical tuning.

We assume that the 4^{th} car in the platoon mounts the attack so $a = 3$. We assume that the 5^{th} car is monitoring for the attack and can react if an attack occurs. The monitoring car receives DSRC communications from the 2^{nd} and 3^{rd} advertising their respective acceleration profiles. Thus car 5 has a model

$$err = \begin{pmatrix} a_{err}|\hat{u}_1 \\ v_{err}|\hat{u}_1 \\ \hat{u}_{err}|\hat{u}_1 \\ a_{err}|\hat{u}_2 \\ v_{err}|\hat{u}_2 \\ \hat{u}_{err}|\hat{u}_2 \end{pmatrix} \quad (32)$$

to base its detection results on. We hand tune our detection parameters to $\delta = [0.23, 0.48, 0.9, 0.46, 0.9, 0.9]^T$. We assume that if any element of $err > \delta$ the system is under attack and instantly switch to an ACC.

6.1 False Positives

We consider the impact of noise on our detection scheme by adding Gaussian noise to the acceleration. We assume that the variance in system noise is proportional to current velocity of the car. In Figure 6 we show the system during noisy operation with the variance for the car i set to $\sigma_i = .0005v_i[k]$. This results in the acceleration update equation being modified to $a_i[k+1] = a_i[k] + v_i[k] + \mathcal{N}(0, .0005v_i[k])$.

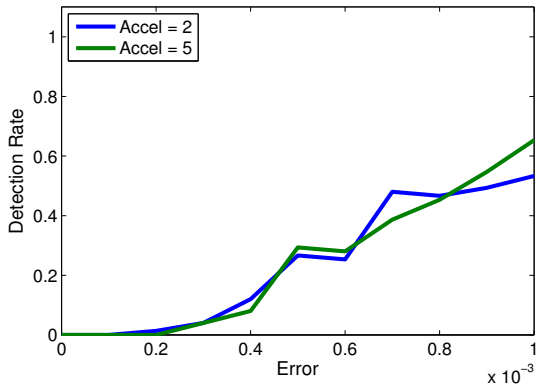

Figure 7: In this figure, we show the false positive rate for different levels of noise with velocity relative variance.

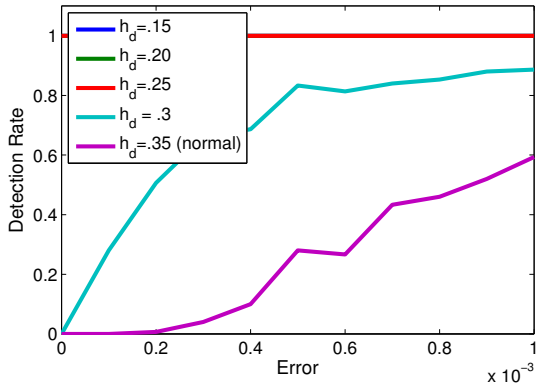

Figure 8: In this figure, we show the headway attack detection results, we calculate the false positive rate across 75 trials with an acceleration rate of $2\frac{m}{s^2}$ and 75 trials with an acceleration rate of $5\frac{m}{s^2}$.

We simulate the system without an attacker present to explore the impact of noise on the false positive rate of our detection scheme. In Figure 7 we model two acceleration rates ($2\frac{m}{s^2}$ and $5\frac{m}{s^2}$) at various noise levels. For each noise level and acceleration profile we run 75 trials and calculate the percentage of time that an attack was detected. In this figure, the false positive rate is acceptable when the noise variance is under .0004 of the velocity. When the noise increases beyond this the false positive rate is extremely high.

We can mitigate the high false positive rate by using filtering to limit the impact of Gaussian noise.

6.2 Attack Detection Results

We again simulate the attacks at various noise levels and, where applicable, with various attack parameters. Similarly, with mis-report percentages $\beta \in [0.05, 0.1, 0.15, 0.2, 0.3]$ we detect 100% of mis-report attacks. Additionally, collision induction attacks are detected 100% of the time. In the next section we explore whether they are detected fast enough to mitigate the attack's effect.

In Figure 8 we show the result for detecting headway reduction with our detection scheme. For headways under 0.3 seconds the detection scheme works 100% of the time. when the headway is close to the expected value of 0.35 seconds, i.e. at 0.3 seconds the detection scheme is not effective.

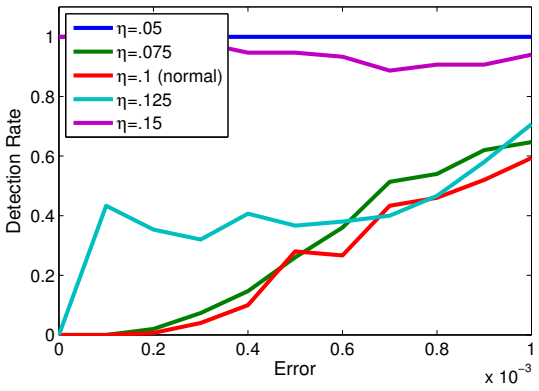

Figure 9: In this figure, we show the abnormal behavior detection results, we calculate the detection rate across 75 trials with an acceleration rate of $2\frac{m}{s^2}$ and 75 trials with an acceleration rate of $5\frac{m}{s^2}$.

In Figure 9 we show the results for detecting abnormalities in the lag using our detection scheme. We again calculate the detection rate based on 75 trials with $2\frac{m}{s^2}$ acceleration and 75 trials with $5\frac{m}{s^2}$ acceleration. When $\eta = 0.075$ and $\eta = 0.125$ which is close to the expected value of $\eta = .1$ the attack is not detected. This is desired behavior since the lag parameters would be defined by a range in real system. When the values of η are 50% greater or less than the expected value the detection rate is over 90%.

Regardless of noise level our detection algorithm does not detect when a car attempts to join without a radar. This is expected as the vehicle that joins without a radar uses a model based acceleration profile, matching the model-based acceleration that we compare against extremely well.

6.3 Attack Avoidance Results

Lastly, we explore whether a collision induction attack can be detected in time to mitigate the collision. In Figure 10 we show the performance of a car using our detection scheme when a collision induction attack is mounted at 10.001 seconds with no system noise. The blue line shows the behavior when no detection and mitigation scheme is used. It is clear that just after 12 seconds an accident occurs at $25\frac{m}{s}$. The red line shows the same attack when the detection and mitigation scheme is used. It takes less then $100ms$ to detect the attack and as can be seen no collision occurs, meaning the detection coupled with the non-cooperative ACC controller successfully mitigated the attack.

7. DISCUSSION

In this section, we discuss design considerations and shortcomings of our misbehavior detection and mitigation scheme. First, we highlight design consideration including tuning the detection threshold and selecting the response mechanism. We then highlight the shortcomings of our system and possible improvements.

It is important to choose the detection threshold for the system carefully balancing false negatives and false positives. Given that a false negative can have a catastrophic outcome, either loss of life or loss of personal assets, the designer likely

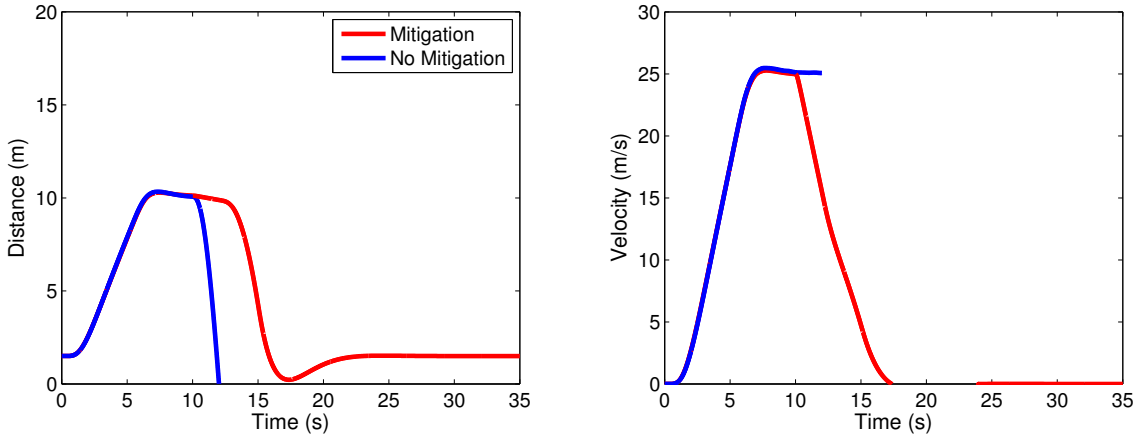

Figure 10: In this figure, we show a car using our attack detection technique avoiding an accident during a collision induction technique. We plot distance between the attacker and monitoring on the left and the velocity of the monitoring vehicle on the right.

will want to minimize them. The cost of false positives on the other hand is relatively low, 20-70 % decreased headway and 5% decreased fuel efficiency, making the costs of a false positive relatively low. Given the danger of false negatives and relatively low cost of false positives a designer will likely choose a conservative model.

Another design decision is how to model the vehicles given DSRC packets. In the real system this involves modeling the dynamics of the car and its interaction with the environment; for example, a car performs differently going up hill. The cars have to have a trusted way to choose parameters for each car which could either be through a cloud-based service or through a trusted broadcast scheme. Modeling the dynamics with the environment could easily be supplemented with GPS data to estimate environmental impact.

Another design decision is how to respond when an attack is detected. In this work we assume that all attacks are equally bad and when anomalous behavior is detected switch to ACC but many approaches could be used. For example, a designer could use an adaptive response where the headway and feedforward controller weight are adjusted based on the anomalous behavior measurement.

One major shortcoming of our approach is the inability of our scheme to detect when the platoon leader is misbehaving. To mitigate this attack we would have to consider global schemes with trusted and secured cars that would mitigate risk. Our system is also susceptible to noise. To mitigate this shortcoming we can use many noise reduction techniques. For example, we could mitigate the impact of Gaussian noise in our observations by using a Kalman filter for optimal estimation.

Another shortcomings of our detection scheme is its inability to detect when a vehicle joins without a radar. This behavior is expected because when a vehicle joins without radar it uses a model based approach to determine its acceleration. Thus the model based approach to the non-radar equipped vehicle performs similarly to the system model based on future communications. In normal scenarios this performs acceptably but becomes dangerous when the DSRC is unavailable. One approach to detect a car without radar is to occasionally introduce a small amount of noise at a given car's velocity. The car directly behind it should respond according to the feedback controller if it has a radar. After a short period the first noise inducing car can communicate how it inserted noise and it can be tested.

It should also be noted that our scheme would likely cause the platoon to disassemble around the detecting car. This opens up another attack similar to the efficiency attacks [3] allowing a car to lessen the efficiency of the platoon.

8. CONCLUSION

Platooning improves the use of current highway systems by safely allowing cars to drive closer together. The decreased headway distances can reduce drag forces on cars allowing platooned cars to reduce fuel consumption by over 5%. Safety in platooning is guaranteed via string stability given that all the vehicles behave according to the prescribed control laws and within the expected performance bounds.

In this work we introduce various ways that a car may misbehave in a platoon, both in a benign or malicious fashion. To mitigate misbehavior in platoons we propose an model-based detection scheme that is used to trigger a non-cooperative ACC mode. In the detection scheme each car makes a model of the expected behavior of the preceding car based on upstream communications. It can uses this modeled expected behavior and measurements to determine if their is anomalous error. If the monitoring vehicle detects anomalous behavior it switches to ACC and we show this is effectively able to mitigate when an attacker attempts to cause a collision.

9. ACKNOWLEDGMENTS

This material is based upon work partially supported by the National Science Foundation under Grants No 0955111, No 1329936, and CNS-1149582. S. Weerakkody and B. DeBruhl are supported in part by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of CMU, NSF, or the U.S. Government or any of its agencies.

10. REFERENCES

[1] Injury prevention & control: Key data and statistics. Center for Disease Control. `http://www.cdc.gov/injury/overview/data.html`, Accessed: 2015-01-18.

[2] Chen. Estimation of car following safety: Application to the design of intelligent cruise control. *1996, PhD Dissertation.*

[3] R. M. Gerdes, C. Winstead, and K. Heaslip. Cps: an efficiency-motivated attack against autonomous vehicular transportation. In *Proceedings of the 29th Annual Computer Security Applications Conference*, pages 99–108. ACM, 2013.

[4] J. J. Haas. The effects of wireless jamming on vehicle platooning, 2009.

[5] R. Hall and C. Chin. Vehicle sorting for platoon formation: impacts on highway entry and throughput. *Transportation Research Part C: Emerging Technologies*, 13(5):405–420, 2005.

[6] W. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nesic. Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *Automatic Control, IEEE Transactions on*, 55(8):1781–1796, 2010.

[7] A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, 25(3):727–745, 2014.

[8] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson. Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass. Technical report, SAE Technical Paper, 2014.

[9] F. Lin, M. Fardad, and M. R. Jovanovic. Optimal control of vehicular formations with nearest neighbor interactions. *Automatic Control, IEEE Transactions on*, 57(9):2203–2218, 2012.

[10] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1484–1497, 2012.

[11] D. Nesic and A. R. Teel. Input-output stability properties of networked control systems. *Automatic Control, IEEE Transactions on*, 49(10):1650–1667, 2004.

[12] J. Ploeg, B. T. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *Intelligent Transportation Systems (ITSC), 14th International IEEE Conference on*, pages 260–265. IEEE, 2011.

[13] M. Segata and R. Lo Cigno. Automatic emergency braking: Realistic analysis of car dynamics and network performance. *Vehicular Technology, IEEE Transactions on 62.9: 4150-4161.* 2013.

[14] D. Swaroop. String stability of interconnected systems: An application to platooning in automated highway systems. *California Partners for Advanced Transit and Highways (PATH)*, 1997.

[15] D. Swaroop, J. Hedrick, C. Chien, and P. Ioannou. A comparision of spacing and headway control laws for automatically controlled vehicles1. *Vehicle System Dynamics*, 23(1):597–625, 1994.

[16] M. Tabbara and D. Nesic. Input–output stability of networked control systems with stochastic protocols and channels. *Automatic Control, IEEE Transactions on*, 53(5):1160–1175, 2008.

[17] R. Teo, D. Stipanovic, and C. Tomlin. Decentralized spacing control of a string of multiple vehicles over lossy datalinks. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 682–687. IEEE, 2003.

[18] J. Zhang. A survey on trust management for vanets. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pages 105–112. IEEE, 2011.

[19] Y. Zhao, P. Minero, and V. Gupta. Disturbance propagation in strings of vehicles with limited leader information. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 757–762. IEEE, 2012.

[20] Y. Zhao, P. Minero, and V. Gupta. On disturbance propagation in vehicle platoon control systems. In *American Control Conference (ACC), 2012*, pages 6041–6046. IEEE, 2012.