Attack Mitigation in Adversarial Platooning Using Detection-Based Sliding Mode Control

Imran Sajjad, Daniel D. Dunn, Rajnikant Sharma, Ryan Gerdes
Dept. of Electrical and Computer Engineering
Utah State University
Logan, UT, USA
{imran.sajjad, d.dunn}@aggiemail.usu.edu, {rajnikant.sharma, ryan.gerdes}@usu.edu

ABSTRACT

In this paper, we consider a mitigation strategy to prevent a vehicle controlled by an attacker from causing collisions in a vehicular platoon. An adversarial-aware control scheme, based on sliding mode control using only local sensor information and a decentralized attack detector, is shown to significantly reduce the number and severity of collisions, without the need for inter-vehicle or vehicle-to-infrastructure communication. Simulations demonstrate that collisions are eliminated (or significantly reduced) when the attacker and normal vehicles have same capabilities, and collisions are reduced even with more powerful attackers.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Reliability, availability, and serviceability; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*control theory*; I.2.9 [**Artificial Intelligence**]: Robotics—*autonomous vehicles*; K.4.1 [**Computers and Society**]: Public Policy Issues—*abuse and crime involving computers*

General Terms

Security, Reliability, Performance

Keywords

autonomous and automated vehicles; vehicle platoon; adaptive cruise control; cooperative adaptive cruise control; attack

1. INTRODUCTION

The automation of highway systems is an area of extensive and ongoing research. One method of developing an Automated Highway System (AHS) is to utilize vehicle platooning, in which vehicles on the highway follow each other with very small inter-vehicle separations, sensing the movements of other vehicles and reacting automatically according to some predefined law. Platooning has been shown to have environmental, safety, and passenger comfort benefits [1, 18]. They also help to alleviate traffic congestion on highways [17] and have shown to be more fuel efficient than manually operated vehicles [12].

The safety and security of these systems is essential. Platooning falls under the broad category of cyber-physical systems (CPS), where most security-centric work has focused on attack detection and not mitigation [8, 13, 14]. For a platooning CPS, it has been shown that a single attacker can disrupt normal operations simply and easily, and that such disruptions can cause catastrophic collisions [4]. Our work aims at ensuring that deviations from expected platoon behavior do not cause collisions or mitigate collision damage as much as possible.

The main contribution of this paper is a sliding mode controller coupled with an attack detection scheme that ensures that deviations from desired inter-vehicle separations remain low. Compared to existing control laws, our controller is able to almost completely eliminate collisions when the attacking vehicle is as strong as regular vehicles; even in the presence of a more powerful attackers the damage caused by collisions is greatly reduced. Our control law and attack detection scheme are decentralized and rely on only the local sensors the platooned vehicle is already equipped with for decision making and reaction purposes.

The remainder of this paper is organized as follows. In Section 2 some platooning preliminaries and a threat model are outlined. Our platoon model is presented in Section 3 and a sliding mode controller is derived around this model in Section 4. Simulation and results are presented in 5. Section 6 concludes this paper.

1.1 Related Work

A large body of work can be found regarding homogeneous platoons, where every car follows the same control low. Most of this work focuses on the stability and string stability of the system [3, 15, 20, 21]. Other work has highlighted some of the limitations of the bidirectional structure [2, 10]. Different inter-vehicle spacing policies have also been considered [20]. It has been previously shown that for the symmetric bidirectional linear controller, the bounds on the front and rear errors increase as the number of vehicles increases [2]. Our proposed controller can enter into asymmetric states and from our analysis, the error bounds are independent of the number of vehicles.

Sliding Mode Control has been used previously in many scenarios. Platooning strategies exist where sliding mode

CPS-SPC'15, October 16, 2015, Denver, Colorado, USA.

ACM 978-1-4503-3827-1/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2808705.2808713 .

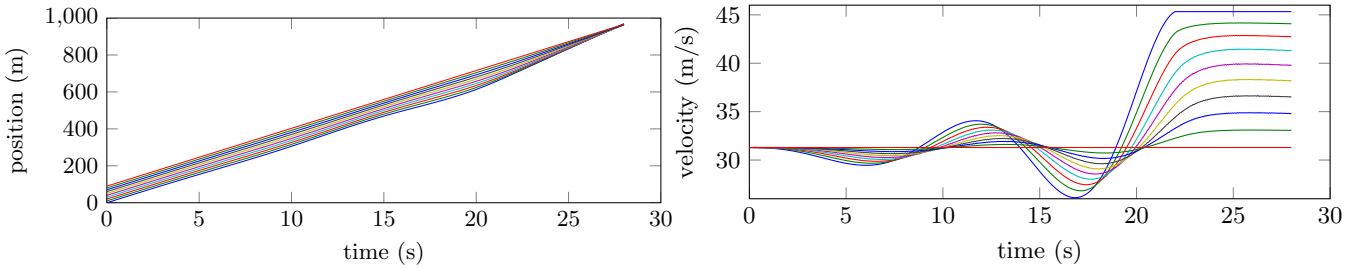


Figure 1: Oscillatory behavior brought on by an attacker, resulting in a high speed crash [4]. Each line represents the trajectory of a vehicle in a ten vehicle platoon with an attacker at the rear.

control has been used in a homogeneous platoon [6] under normal operation. Graph theoretic approaches similar to ours have been used before in platooning [5, 22], and in general problems of multiple vehicle target tracking in the presence of uncertainties [19].

Apart from platooning, much work has been done in interconnected dynamic cyber-physical systems. The security and robustness of these systems in the face of an attack or failure is crucial and is an active area of research [14]. Graph theory, information flow analysis have been used to analyze such systems as well [5, 22]. Much of this work is focused on ensuring suitable operating conditions for dynamic systems, mainly stability, controllability and observability.

Our approach builds upon the above works and tries to solve the safety problem in an adversarial environment. None of the above investigate platooning performance in the presence of an attacker. Our choice for sliding mode controller is follows naturally when some limitations on the attacker capabilities are known. We also incorporate maximum performance constraints and measure the efficacy of our approach by measuring the severity of the collisions that take place.

2. THREAT MODEL

For the purposes of this study, we consider a platoon of n members, each equipped with front- and rear-facing sensors that measure relative distance and velocity. Aside from the attacker, the vehicles adhere to the same control law and have the same capabilities, as described in the next section. The last member is indexed as 1 and the leader is at index n. We focus on the bidirectional platoon scheme [23] where every car gathers information about (e.g. range and relative velocity), and reacts to the movements of, both the vehicle preceding and following it. The leader tries to maintain a separation with its follower and has access to a reference trajectory. The last car only tracks the car immediately in front of it.

We assume only a single attacker in control of a car at an arbitrary position in the platoon. The attack car is possibly more powerful than the regular cars, i.e. it may have greater acceleration capabilities. The goal of the attacker is to cause multiple collisions in time. To accomplish this the attacker follows a modified control law that induces oscillations in the platoon (Fig. 1). It has been shown that an attacker can leverage oscillatory behavior to cause more accumulated damage, and more collisions over time, than one that simply accelerates in one direction and that this can be achieved simply by changing the some controller gains [4]. The attack always starts in a steady state configuration, when the cars are traveling at their desired separations, which is chosen to be one car length of separation in our tests.

3. RATIONALE AND SYSTEM OVERVIEW

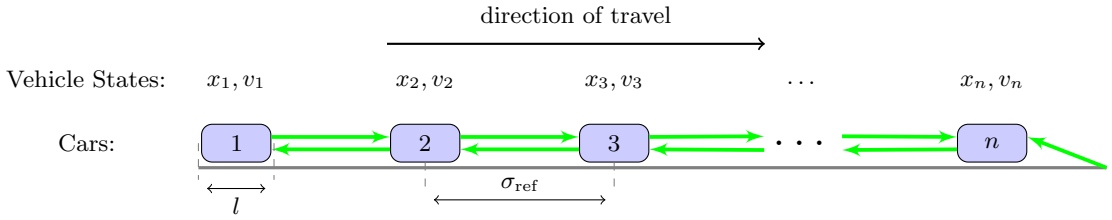
The bidirectional platoon structure (Fig. 2a) has two principle benefits over the unidirectional approach: 1) it offers the added safety advantage of avoiding collisions from the rear, and 2) also allows for constant spacing between vehicles, provided the size of the platoon is known and used to tune controller gains, without vehicle-to-vehicle communication [23]. We show that this structure is especially vulnerable to attack, but to retain its benefits we devise a scheme of altering the structure during times of attack that ensures that collisions are at least minimized, if not avoided altogether. Technologies such as cooperative adaptive cruise control (CACC) that use V2V communication are sensitive to jamming attacks and thus safety has to be guaranteed without reliable external information. Even these systems are vulnerable to instability attacks caused by attacker motion [4, 23].

3.1 The Vulnerability of Bidirectional Control

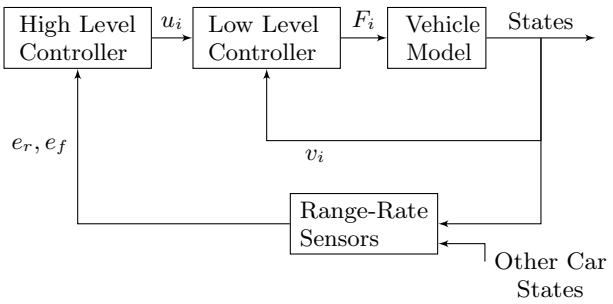
Both bidirectional and unidirectional platooning require consensus for proper operation. For example, a predecessor unidirectional law maintains the separation between vehicles by having the follower respond to the movements of the vehicle in front of it. In a three vehicle platoon there is consensus when the vehicle at the lead of the platoon slows down and so do the two followers. Consensus, however, cannot be guaranteed. If the car at the rear does not move back (or wants to accelerate into its predecessor), then the car in the middle will not be able to defend against it.

The bidirectional system, owing to the fact that the middle vehicle reacts to what is happening both in front and behind it, may seem to but, in fact, does not solve this problem. A symmetric bidirectional law does not control both the rear and front separations simultaneously. Rather, it tries to place the current car in the middle of the two neighboring cars. Assuming again the three vehicle platoon with an attacker in the rear who decides to accelerate, we can see that due to its length the middle vehicle that tries to place itself equidistant to the attacker and leader, when the space between is diminishing, would inevitably collide with the leader. In fact, we show (Appendix A) that the bidirectional system formed around a single car is locally uncontrollable and that at least one cooperating neighbor is required for stable operation.

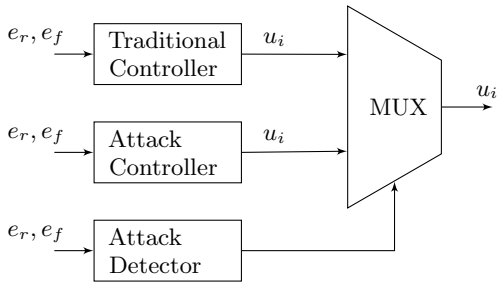
If we limit our discussion to a single attacker, we propose to use the consensus condition that is required in both cases



(a) Platoon with a bidirectional law under normal operation. Information flow in green. Each car is l meters long and the desired separation from center of one car to that of the other is σ_{ref}.



(b) Controllers of a single vehicle.



(c) Inside high level controller: attack detection and controller selection based on attack state.

Figure 2: Overview of Platoon. Each vehicle knows its own velocity and measures a relative distance and velocity from rear and front (e_r, e_f). These same measurements are used in the high level controller to switch between rear or front tracking if an attack is detected.

anyway. We recommend a secondary controller that tries to keep a constant distance from the more dangerous and uncooperative car (in front or behind) and relies on the other car to move and make room. Under normal circumstances a traditional bidirectional law is followed; however, upon detect of anomalous behavior, indicating the onset of an attack, this secondary controller is engaged to mitigate the attack (Fig. 2c). This approach is shown to allow a straightforward, ultimate boundedness analysis and simulation results show that it greatly reduces total damage compared to a bidirectional scheme.

3.2 Platooning Goals in Adversarial Conditions

An attacker vehicle cannot be assumed to be following the control scheme of the other vehicles. They have free reign to do whatever they want, and the other cars do not have any assurance of its cooperation. The possible combinations of such attacks are virtually limitless. To investigate operation in the presence of attackers, we define revised platooning goals in the presence of an attack that ensure safety at the expense of other desirable platooning properties:

1. The instantaneous and total mean square error from reference should be as tightly bounded as possible.

2. The instantaneous and total damage from collisions should be minimal.

From the point of view of the attacker(s), the aim is to defeat these goals. Both these goals are interlinked as well, in the sense that there has to be some error in relative positions before a collision takes place.

For the first goal, we have to investigate how errors propagate in the system in the presence of attackers. This largely depends on the number of attackers and what they are doing, but general statements can be made using concepts such as string stability and Lyapunov Stability [3, 11, 21]. Such an analysis would have to be global and the interaction between each member of the platoon with every other one would have to be investigated.

In this paper, we give priority to the second goal because that one seems more imperative if there will be human passengers in the platooning vehicles. Incidentally, it is easier to analyze as well, since the number of interactions is smaller, and the analysis is not entirely removed from that of the first goal.

For this purpose, a global analysis is not required as such but would be beneficial for a more complete understanding. We only have to ensure that each car does not collide with its neighbors. To achieve this, we design a decentralized controller in section 4 for a single car using a concept from Lyapunov Stability called *uniform ultimate boundedness*, which ensures that once an error is restricted to an interval, it will never leave that interval [11]. In other words, when the uniform ultimate boundedness property is ensured with an appropriate controller, the inter-vehicle distance between a car and its neighbor never deviates from the required separation enough to cause a collision.

Since total or instantaneous damage is not formally defined, we propose to use a metric that depends on two things; whether an impact takes place and the relative velocity of the colliding vehicles. This choice of measuring damage is motivated by previous work done on automated vehicle and platooning safety [7, 9]. To measure the accumulation of damage, we assign the following as a rate of change to a state D:

$$\dot{D} = c^T v_{\text{rel}} \tag{1}$$

where c is an $n-1$ length vector whose entries are 0 normally, but 1 if there is a collision. v_{rel} is a vector containing the absolute values of $n-1$ relative velocities at time of collision.

3.3 Bidirectional Platooning Control

In keeping with the current literature [2, 16], each vehicle is analyzed as a double integrator system, where the control input is a desired acceleration. For an n-vehicle platoon, the state vector $x \in \mathbb{R}^{2n}$ is made up of positions and velocities and the input vector $u \in \mathbb{R}^n$ consists of control inputs. The state and input vectors can be expressed as

$$\begin{aligned} x &= \begin{bmatrix} x_1 \ x_2 \ \dots \ x_n \ v_1 \ v_2 \ \dots \ v_n \end{bmatrix}^T, \\ u &= \begin{bmatrix} u_1 \ u_2 \ \dots \ u_n \end{bmatrix}^T \end{aligned} \tag{2}$$

and the state space system becomes

$$\dot{x} = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} x + \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} u \tag{3}$$

where car i has position and velocity x_i, v_i respectively and control input u_i. These positions are measured from the center of mass of all the cars. In the bidirectional scheme

$$\begin{aligned} u_i = f_i(&x_{i-1} - x_i, v_{i-1} - v_i, \\ &x_{i+1} - x_i, v_{i+1} - v_i) \end{aligned} \tag{4}$$

which means each vehicle's control input can only use relative distance and velocity measurements from its immediate neighbors. This function f_i constitutes a high level controller that is meant to be independent of a vehicle's dynamics (Fig. 2b); as such the control input u_i serves as the vehicles desired acceleration.

As the rearmost and leader vehicle lack a follower and predecessor, respectively, they follow a slightly modified version of (4) wherein the rearmost car uses a unidirectional law, and the leader follows a reference trajectory while maintaining a follower separation

$$u_1 = f_1(x_{i+1} - x_i, v_{i+1} - v_i) \tag{5}$$

$$\begin{aligned} u_n = f_n(&x_{i-1} - x_i, v_{i-1} - v_i, \\ &x_{\text{ref}} - x_i, v_{\text{ref}} - v_i) \end{aligned} \tag{6}$$

3.4 Vehicle Model

The previous section assumes that we can achieve a desired acceleration and apply it directly to our system. A realistic model of a vehicle has a throttle input or some other type of actuator. The purpose of this section is to find how a desired acceleration can be achieved based on our knowledge of the vehicle. A model of the vehicle's dynamics is required in this case. This can be specific to different vehicles, but the general idea is to find an expression for the control input required for a desired acceleration. This constitutes the low level controller of Fig. 2b.

The vehicle model we use is a 2^{nd} order plant with a linear friction/drag coefficient. Such models are easy to analyze while capturing the major dynamics of the system. Similar models have been used in other control systems literature to analyze fundamental properties of single vehicles and platoons [10, 15, 21].

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= \alpha F_i - \beta v_i \end{aligned} \tag{7}$$

where $F_i \in [F^-, F^+]$ is a variable to set the actuator (throttle) and α, β are the model's parameters which can be chosen based on the vehicle's internal design values or through system modeling [16, 21].

For the high level controller described in (4) to work, we need to compensate for the internal dynamics of the vehicle. We use feedback linearization [16,21] to compensate for terms in the model described by (7) gives us

$$F_i = \frac{1}{\alpha}\left(u_i + \beta v_i\right) \tag{8}$$

Note that this controller does require a velocity measurement of vehicle i. A sensor which provides this reading will be required, but this is just car sensing its internal data and does not violate the decentralized condition.

The reason for including this model is to emphasize that there are bounds F^-, F^+ on F_i which lead to saturation. We simulate with these saturation limits in order to demonstrate our controller on a realistic system where the desired acceleration cannot always be achieved. A favorable consequence of this is that we cannot achieve infinite acceleration.

Substituting (8) into (7) gives us the required double integrator type system for each vehicle

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= u_i \end{aligned} \tag{9}$$

as long as the condition $\frac{1}{\alpha}\left(u_i + \beta v_i\right) \in [F^-, F^+]$ holds true. These saturation constraints apply to the attacker as well and ensure that we do not have a car with unrealistic capabilities.

We also add another constraint on this controller which prohibits reverse motion. This is to maintain relevance with the real application of AHS. It can be expressed as $F_i > F_i^-$ if $(v_i > 0)$ and $F_i > 0$ otherwise, which means that if a vehicle's speed is zero or below, it cannot apply negative actuator input.

4. ATTACK CONTROLLER

For our secondary controller that governs vehicle response when under attack, we propose the use of sliding mode control (SMC) because the nature of the problem lends itself naturally to SMC. Firstly, the demands of the system require the fewest collisions (preferably none at all) in the face of an attacker. Secondly, the attack is limited in what it can do by its (perhaps heightened) acceleration and velocity constraints. Sliding mode controller techniques can use these constraints directly and guarantee ultimate boundedness of the tracking error, which implies no collisions. Lastly, SMC is a robust, well-understood method of nonlinear control and straight-forward tools exist to design and analyze its performance.

We will incorporate uncertainties and bounds on acceleration and velocity based on our model in the previous section and derive a suitable sliding mode controller. Then we will design its continuous approximation that enables the defending cars to maintain the desired distance from the attacker within some error bound. Separate front and rear controllers, to control response to an attack originating to the front and rear of the vehicle, respectively, are designed and then combined later in this section to give a single, unified high level controller for platooning operations while under attack.

4.1 Mathematical Preliminaries

To demonstrate the efficacy of the sliding mode controller, we first need some mathematical preliminaries from control

systems theory. For a system with state $x \in \mathbb{R}^n$, the point $x = 0$ is stable if the quantity $\|x\|$ remains bounded for all future time, if it was bounded initially. It is asymptotically stable if $\|x\| \to 0$ as $t \to \infty$ (Def. 4.1, [11]).

The stability test involves finding a function $V(x)$ which has the some desirable properties (continuously differentiable, $V(0) = 0, V(x) \neq 0 \ \forall x \neq 0$) and demonstrating that its derivative $\dot{V}(x)$ is always negative (Thm. 4.1, [11]). Such functions are sometimes referred to as Lyapunov Candidate functions. Usually (and as in the next section) the control input u should appear in the expression for $\dot{V}$ (the derivative of V). If $\dot{V}$ is not negative, it can be forced to be negative by applying an appropriate u.

In the case of sliding mode control, we choose u in such a way that u is only dependent on one variable $s = \sum k_i x_i$. This sliding manifold s is chosen to be stable and the controller is used to drive the system onto this manifold. The variable s can be controlled by a bang-bang type of controller, but a continuous controller is desired in most real-life situations to avoid chattering. A function $\text{sat}(s/\epsilon)$[1] can be used with the variable ϵ chosen for a given ultimate bound (Thm. 14.1, [11]). Thus we can choose a bound on the maximum deviation of position error such that it is less than the distance to the next car. This will ensure no collisions.

The process we follow in the next section is to design a single sliding mode controller that ensures constant spacing for one direction, and then combine two of these for rear and front separation to mitigate attacks from either direction.

4.2 Single Controller Design

The design of a front error controller follows. Note that the next car's acceleration is not assumed to be known. This car could very well be an attacker so we only use bounds on this quantity to derive our controller. We begin by defining error coordinates in the frame of car i with the desired separation σ_{ref} in meters

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} x_{i+1} - x_i - \sigma_{\text{ref}} \\ v_{i+1} - v_i \end{bmatrix} \tag{10}$$

where e_1 and e_2 are the front separation error and relative velocity error respectively. If σ_{ref} does not change with time, or changes slowly enough, we can formulate the following state space model.

$$\begin{bmatrix} \dot{e_1} \\ e_2 \end{bmatrix} = \begin{bmatrix} e_2 \\ \ddot{x}_{i+1} - u_i \end{bmatrix} \tag{11}$$

If we define a sliding manifold $s = k_1 e_1 + e_2$, $s = 0$ is naturally stable if $k_1 > 0$, which is to say that so long as $e_2 = -k_1 e_1$, both e_1 and e_2 go to zero. To show this mathematically,

$$V_s = \frac{1}{2} e_1^2 \tag{12}$$

$$\dot{V}_s = e_1 \dot{e}_1 = e_1 e_2 = -k_1 e_1^2 \tag{13}$$

which implies asymptotic stability (Thm. 4.1 [11]). Outside of this manifold, we can use the Lyapunov Candidate $V = \frac{1}{2} s^2$ to check if our system reaches the line $s = 0$, and

$$\begin{aligned} \dot{V} &= s\dot{s} \\ &= s(k_1 e_2 + \ddot{x}_{i+1} - u_i) \\ &\leq \|s\| \left(k_1 \|e_2\| + \|\ddot{x}_{i+1}\|\right) - s(u_i) \end{aligned}$$

[1] $\text{sat}(x) = x$ if $\|x\| < 1$ and $\text{sgn}(x)$ otherwise

Now we are ready to include our attacker's constraints in our controller. With $\|e_2\| \leq 2v_{\max}$ and $\ddot{x}_{i+1} \leq a_{\max}$ [2] we use the controller:

$$u_i = \text{sat}\left(\frac{s}{\epsilon}\right) [2k_1 v_{\max} + a_{\max} + \epsilon] \tag{14}$$

For $\|s\| > \epsilon > 0$, we have:

$$\begin{aligned} \dot{V} &\leq \|s\| \left[k_1 \|e_2\| + \|\ddot{x}_{i+1}\|\right] - s(\text{sgn}(s) \left[2k_1 v_{\max} + a_{\max} + \epsilon\right]) \\ &= \|s\|[k1(\|e_2\| - 2v_{\max}) + (\|\ddot{x}_{i+1}\| - a_{\max}) - \epsilon] \\ &\leq -\|s\|\epsilon \end{aligned}$$

Hence choosing ϵ will give us an ultimate bound on the error (Thm. 14.1, [11]). Given a choice of k_1 and the requirement that $\|e_1\| < (\sigma_{\text{ref}} - l)$, where l is the length of a car and $e_2 = 0$, we choose ϵ such that

$$\begin{aligned} \|s\| &> \epsilon \\ k_1 \|e_1\| + \|e_2\| &> \epsilon \\ k_1(\sigma_{\text{ref}} - l) &> \epsilon \end{aligned}$$

Thus, we have a controller with two parameters (k_1, ϵ) that have a range of acceptable values. The separate controllers for the rear systems can be derived in a similar manner and are combined in the next section. Combining (14) and (8) appropriately will give us our full controller. Also, n does not appear in any of these expressions. If the controller is applied as is (in only one direction), our error bounds will essentially be independent of the number of vehicles.

4.3 Unified Attack Controller

The front and rear controllers are combined in the graph theoretic manner presented in [19]. Let $G = (V, E)$ be the directed graph representing the interconnectivity of the system. V is the set of nodes (same as the number cars) present in our graph and E the set of directed edges. An edge $(i, j) \in E$ (drawn from j to i in our figures) means that car i can sense information about car j. If a car can sense information about another car (directed edge exists), it is said to be its neighbor. A useful way to denote this is the adjacency matrix.

The adjacency matrix of a directed graph G is denoted $A_{\text{adj}} \in \mathbb{R}^{n \times n}$ where $a_{ii} = 0$, $a_{ij} = 0$ if there is no edge (i, j) and $a_{ij} = c$ where $c > 0$ represents the weight of the edge (i, j). For the bidirectional system this becomes:

$$A_{\text{adj}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{15}$$

The controllers combined for the front and rear are the sum of the controllers weighted with the rows of the adjacency matrix,

$$\begin{aligned} u_i &= \sum_{j=1}^{n} a_{ij} u_{i,j} \\ &= u_{i,r} + u_{i,f} \end{aligned}$$

[2] The maximum velocities and accelerations can be easily derived from the given model in the previous section and saturation levels on the input F_i.

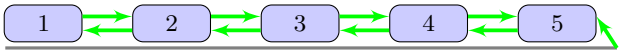


Figure 3: Interaction of a 5-member platoon. The leader also follows a reference. The arrows denote information flow; an arrow from 3 to 4 means 4 senses some information about 3, for example relative distance.

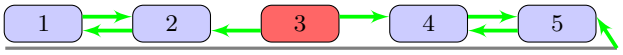


Figure 4: Interaction of a five member platoon with attacker at position three. Note that attacker is assumed to be indifferent.

which is simply the linear combination of the two high level controllers from front and back error systems. The leader's front controller is based on the reference trajectory.

This structure is shown in Fig. 3. If there is consensus in the system, all the cars will try to maintain the same inter-vehicle spacing and the platooning goals will be met. However if some car is not cooperating, platooning goals might not be met, as in the case of an attacker. Furthermore, this controller combining can be seen as an external disturbance which one of the two sliding mode controllers is not meant to deal with. $u_{i,r}$ can be seen as an external disturbance to $u_{i,f}$ and vice versa. Hence the ultimate boundedness analysis might not hold.

4.4 Adjusting the Graph in Case of an Attacker

Consider the case where an attacker is at position three in a five member platoon. The attacker cannot be assumed to be looking at any other members (possible worst case) so row three is zero. The adjacency matrix is then

$$A_{\text{adj}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and this scheme is shown in Fig. 4. From simulation results, it is possible that cars around the attacker, while trying to maintain their distance from the other cars, fail to keep their spacing from the attacker, as the controllers are given equal weight. Our proposed solution to this problem is to use an attack detection method, and change the weights in the adjacency graph so that the controller in the direction of the attack is prioritized.

Since a car can only change its own rows of A_{adj}, the detection and adjustment scheme has to be decentralized and without inter-vehicle communication as well. Attack detection filters are implemented in the following subsection as discussed in [14]. Such filters, which each car is equipped with, have two outputs, one for an attack somewhere in front, the other for anywhere behind. This is a detection scheme and not an identification method. But, as we demonstrate, even this helps greatly with damage mitigation.

A rule for adjusting the weights of the controller is used. This rule, in practice, could be a continuous mapping of the attack detection output, or perhaps just a decision rule. A simple scheme for this is outlined as follows (Fig. 5): The rear and front attack detection filters give outputs r_r, r_f respectively. These values should be zero if there is no attack and more and more positive if there is one. The threshold ϵ_r can be chosen to ignore false positives due to sensor noise. Additionally, it can also be set to achieve a tolerance level; cars might have to deviate a certain amount before they are detected as attackers by their neighbors.

input: (r_r, r_f) %results from attack detection (rear and front)
if $\|r_f - r_r\| < \epsilon_r$ % epsilon_r is some threshold
$a_{\text{adj},i,i-1} \leftarrow 0.5$; $a_{\text{adj},i,i+1} \leftarrow 0.5$ % look front and back
else
if $r_f - r_r > 0$
$a_{\text{adj},i,i-1} \leftarrow 0$; $a_{\text{adj},i,i+1} \leftarrow 1$ % only look front
if $r_f - r_r < 0$
$a_{\text{adj},i,i-1} \leftarrow 1$; $a_{\text{adj},i,i+1} \leftarrow 0$ % only look back

Figure 5: Decision rule for adjusting adjacency matrix. Each car adjusts only its own row, based on local information.

If implemented correctly, we should obtain the structure outlined in Fig. 6 and the following adjacency matrix

$$A_{\text{adj}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In this setting our ultimate boundedness analysis should hold, as there is only one control objective for each car. It should be noted that, say for this case, car number two moves back to avoid car three and can only 'hope' that car one moves back as well[3]. Additionally, the error bounds are independent of the number of vehicles if the controller is in this state.

Thus our detection scheme switches the bidirectional controller to a unidirectional one in certain cases, with the direction (rear or front), dependent on the position of the attacker. The leader follows a different version of this rule because in front of it, there is a reference trajectory and not another car. A threshold is only applied to the rear attack detection filter output to decide whether to ignore the reference or not. The last car does not follow this rule at all since it has only one control objective.

4.5 Attack Detection Filter Design

The attack detection filters used here are essentially low pass filters that act on measurement residuals. Low pass filtering is essential because an attack detection filter should not change its result with the same frequency at which the attacker is oscillating. Another convenience is that the measurement residual depends only on the error coordinates $e = [e_1\ e_2]^T$, which are used for the controller in section 4.2. The error coordinates are then passed through a squaring function with gains l_1, l_2 and then low pass filtered. The

[3]This fact should not be surprising, since even under normal platooning, consensus is required for operation. For example, no car can arbitrarily assign front and rear desired distances for itself if either of the other cars does not want that spacing.

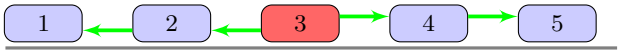


Figure 6: Interaction of a 5-member platoon with attacker at position 3 with adjacency adjusted.

Table 1: Simulation Parameters

Vehicle Dynamics		Controller	Detection Filter
normal	attacker		
$F_i^+ = 1$	$F_i^+ = 1$	$k_1 = 0.1$	$l_1 = 200$
$\beta = 0.1$	$\beta = 0.1$	$\epsilon = 0.025$	$l_2 = 600$
$\alpha = 5$	$\alpha_{\text{att}} \geq 5$		$f_{\text{cutoff}} = 0.01$ Hz

time and frequency domain representations are given as

$$r(t) = h_{\text{lp}}(t) * e(t)^T \begin{bmatrix} l_1 & 0 \\ 0 & l_2 \end{bmatrix} e(t) \tag{16}$$

where $h_{\text{lp}}(t)$ is the impulse response of the low pass filter and for our simulation, this is chosen to be a 2^{nd} order Butterworth filter with cutoff frequency f_{cutoff}. Standard filter design techniques can be used to choose the parameters when certain characteristics are desired from the response, such as rise time and damping.

In general, the filter parameters can be chosen to be very high for a quick response, but there is a trade off between speed and accuracy that is mostly set by the cutoff frequency. One of the caveats of this type of filter is that it 'undetects' an attack as quickly as it detects it. But it is also important to note that the choice of filter does not play an essential role in the global picture and that parameters can be chosen with some degree of freedom in searching for optimal performance.

Because we know that errors propagate in interconnected platoons [2, 10, 15, 21], these filters will be able to detect an attack even if the attacking vehicle is far down or up in the platoon. In other words if the attacker is at position three and car two reacts accordingly, then it too will deviate from the desired spacing. Car one will sense this deviation from car two and will then react accordingly and so on. We emphasize again that we have an attack detection scheme, not an identification scheme.

As mentioned before, there are two of these filters, one for the rear error system, and one for the front. The results of these two r_r, r_f are compared to figure out the change in connectivity. Since they work on information already available, they do not require any extra sensors or communication. Our system is still completely decentralized.

5. SIMULATION AND RESULTS

To demonstrate the effectiveness of our approach, we consider a five vehicle platoon with the attacker at position three. The attacker vehicle follows a square-wave acceleration pattern, where the attacker applies maximum control effort and then minimum with a given frequency f_{att}. Our platooning goals stipulate $\sigma_{\text{ref}} = 9$ m and $v_{\text{ref}} = 25$ ms^{-1}, where each car length $l = 4.5$ m (one car length of separation between cars). The parameters in the dynamic model of the cars, controller and detection filter are given in Table 1. In order to increase the attacker power, α_{att} is chosen to be greater than α. This is equivalent to having a more powerful engine. Consequently the maximum acceleration and velocity of the attacker will be equal or higher than the normal vehicles.

5.1 Evaluating Attack Efficacy

To calculate the effect of an attack we assign a damage state to the platoon along the lines of 1. This damage state starts with a value of zero and all the collisions' relative velocities are accumulated as the simulation progresses and cars collide.

We also define a 'collision line' as follows:

Definition 1. Given an attacking and a defending vehicle along with some initial conditions, with both applying maximum actuator effort, it is possible to find the time they collide (t_{col}) using the solution to $x_{i+1}(t) - x_i(t) = 0$. Then $f_{\text{col}} = \frac{1}{2t_{\text{col}}}$ is a function of relative attacker power and initial conditions.

The value of f_{col} gives us a cutoff frequency for each value of relative attacker power. Below this frequency ($T/2 > t_{\text{col}}$, enough time to collide in any case), there will be unavoidable collisions. Above this frequency we can avoid collisions if a suitable control scheme is adopted.

In other words, for Figures. 8, 9 and 10, all damage that outside the green line should be avoidable. The attacker is oscillating too fast to have enough position deviation to hit the other vehicle that is moving away from it.

5.2 Results Comparison

In all of the plots against time presented below, the attacker is of equal power ($\alpha_{\text{att}} = \alpha$) as the other vehicles. Total simulation time was 120 seconds.

From a comparison at a single frequency of attack (Fig. 7a, Fig. 7b), we find that damage is reduced significantly by applying the attack detection approach. Below are accumulated damage comparisons across a range of frequencies and a range of relative attacker power (Fig. 9, Fig. 10). The numbers on the y-axis correspond to the ratio of attacker power over normal vehicle power. For a reference, we also include a total damage measurement using a linear bidirectional control law in Fig. 8. The high level controller for this was

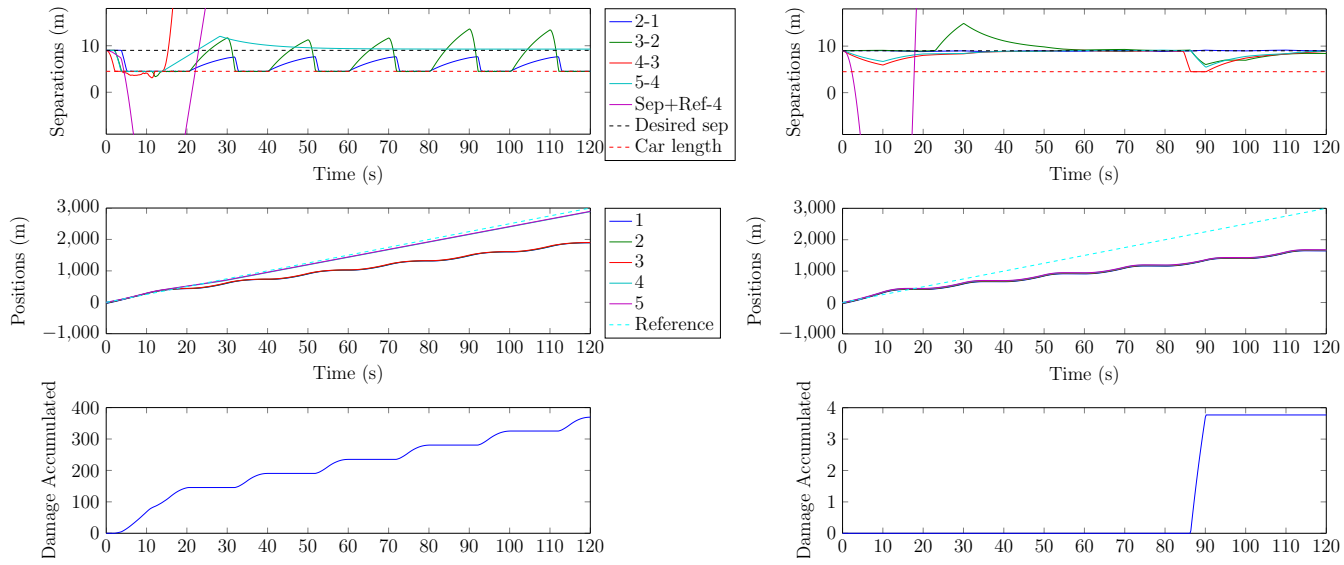
$$\begin{aligned} u_i = k_p(x_{i+1} - x_i - \sigma_{\text{ref}}) + k_p(x_{i-1} - x_i + \sigma_{\text{ref}}) + \\ k_d(v_{i+1} - v_i) + k_d(v_{i-1} - v_i) \end{aligned} \tag{17}$$

with $k_p = 1$ and $k_d = 3$. The attacker breaks the platoon into two sections and for five vehicles, these gains are string stable [4].

The collision line bounds a region in which whatever controller we design, there will be collisions based on the saturation limits of the defending vehicles with respect to those of the attacker. Outside of this region, collisions are avoidable. We see that with detection, we can have close to zero collisions outside this region, whereas we still see damage with normal bidirectional control.

The little damage that occurs in the low frequency region with relative attacker power of 1 is small. We surmise that our attack detection filters may at times register false negatives, where the error goes down enough that the detection filters unregister an attack. After that, it might take time for the filters to detect the attack again. This drawback should be negated using a more robust or intelligent filter design. Even if there are false positives, they should be short lived and the attack detection should correct itself before any significant damage has taken place, as is the case with our controller.

Across this landscape, we find that damage is greatly reduced using attack detection in many cases, most notably



(a) Sliding mode controller without detection. **(b) Sliding mode controller with detection.**

Figure 7: Separations, positions and damage data, single attacker at 3. With detection, we sometimes see a few collisions where our filters detect a false negative and are not quick enough to register the attack again.

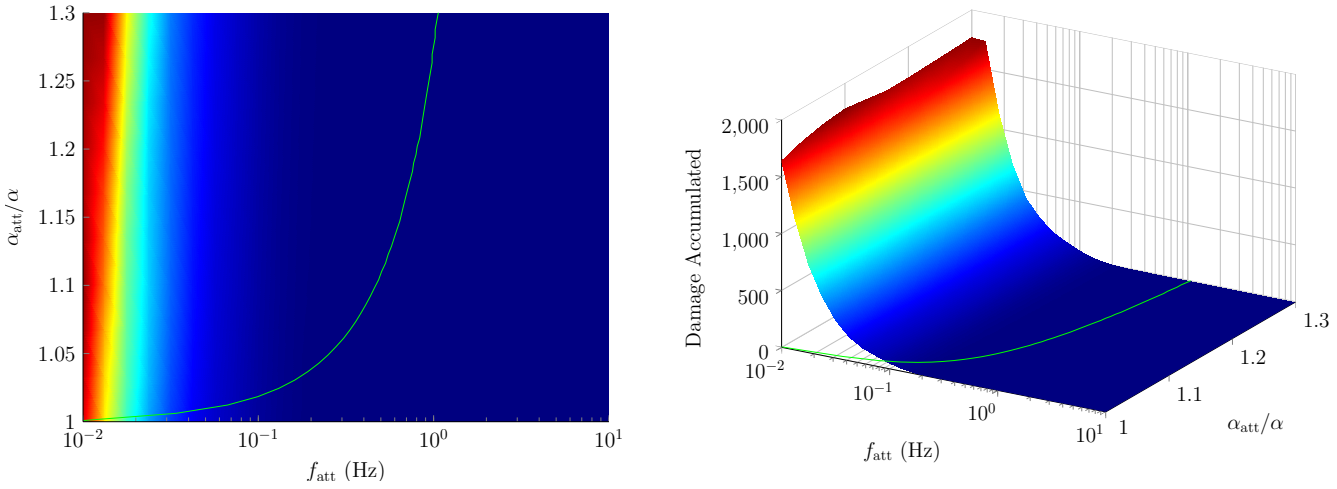


Figure 8: Linear Controller without attack detection. Total damage across relative attacker power and frequencies. Collision line in green.

low frequency attacks with attacker power equal to normal vehicle power.

There are also characteristic similarities between the two curves, namely that there is a frequency above which there is no damage for every possible attacker power level. This is expected given the attacking vehicle has some constraints from saturation.

One more thing to note is that the leader gives up the reference trajectory when it detects an attack behind it. This is equivalent to giving up platooning and following the attacker. Thus control of all the vehicles is given to the attacker, which acts like a new reference. This undesirable effect might be avoidable in the future by using a less aggressive adjustment law, where control in one direction is not fully turned off. It might be possible to start a different platooning protocol after a certain amount of time spent defending this way, such as to increase separations or to disband the platoon. Further investigation does seem warranted in this direction.

6. CONCLUSION AND FUTURE WORK

In this paper, we examined a sliding mode control scheme's effectiveness at stopping collisions in adversarial platooning environment. Two independent sliding mode controllers, to thwart attacks coming from the front and rear of the vehicle, were devised and then combined using an adjacency matrix. While some of the assumptions of sliding-mode control are not met when controllers are combined in this way, with certain detection measures these deficiencies are negated and the amount of damage taken reduced by switching the interconnection of the system. We tested our approach on a

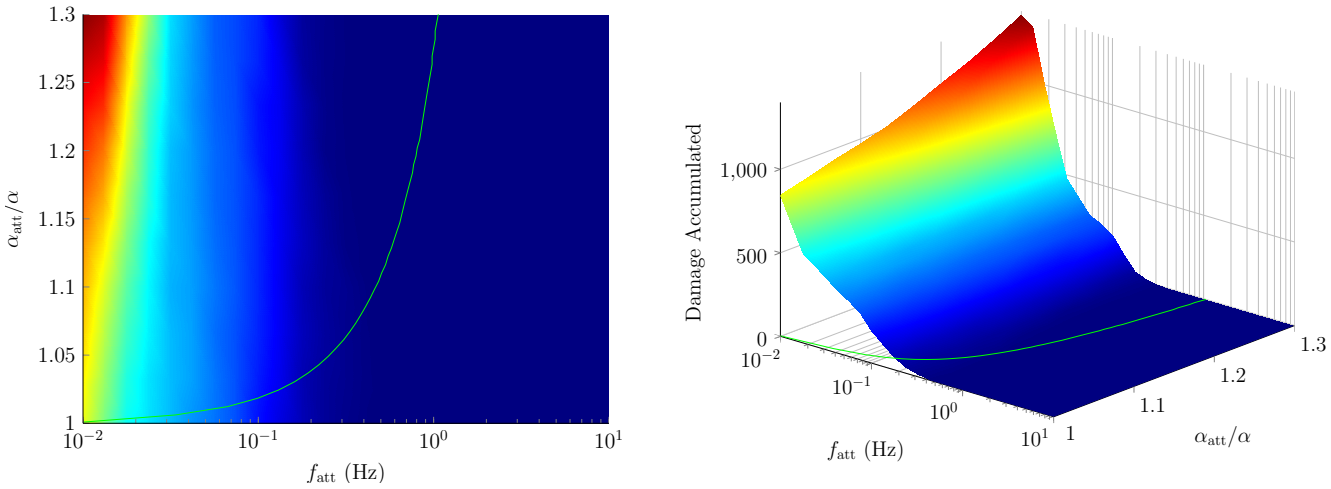


Figure 9: Sliding Mode Controller without attack detection. Total damage across relative attacker power and frequencies. Collision line in green. Note that outside the collision line, we still have collisions, especially when attacker is as strong as normal vehicles.

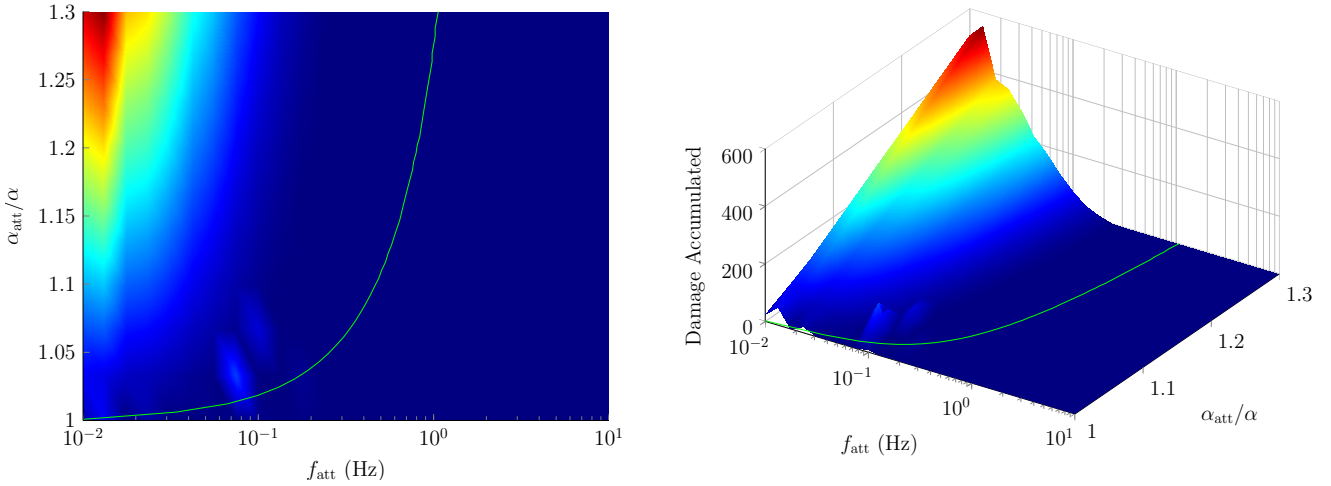


Figure 10: Sliding Mode Controller with attack detection. Total damage across relative attacker power and frequencies. Collision line in green. Note that outside the collision line, there is very little damage with detection. This means avoidable collisions are being avoided.

realistic model of a vehicle and presented the methodology for developing a controller based on this model. Through simulation results, we observed that damage is greatly reduced when our controller and detection method are employed, and that most, if not all, avoidable collisions are protected against.

The primary goal of this work was to preserve the safety of platoons at the expense of other platooning goals (e.g. string stability). Consequently, every car follows the actions of the attacker to ensure that no collisions result from their actions. Future work will consider how a hybrid approach, which takes both safety and string stability into account, may be developed. Secondly, we did not observe a drastic change between the linear and sliding mode controllers in the absence of adjusting the adjacency matrix to accommodate the direction from which the attack originated; i.e. a pure SMC approach for bidirectional platooning would not provide inherent protection. Finally, the case of multiple attackers remains to be investigated. An analysis on controllability and consensus will have to be carried out with more than one attacking car and tests will have to include parameters like attacker positions, level of collusion and attack observability will have to be included.

7. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. 1410000.

8. REFERENCES

[1] "The SARTRE project". `www.sartre-project.net`, 2002. [Online; accessed 15-June-2015].

[2] P. Barooah and J. Hespanha. Error amplification and disturbance propagation in vehicle strings with decentralized linear control. In *Decision and Control,*

2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on, pages 4964–4969, Dec 2005.

[3] R. Caudill and W. Garrard. Vehicle-follower longitudinal control for automated transit vehicles. *Journal of Dynamic Systems, Measurement, and Control*, 99(4):241–248, 1977.

[4] S. Dadras, R. M. Gerdes, and R. Sharma. Vehicular platooning in an adversarial environment. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 167–178. ACM, 2015.

[5] J. Fax and R. Murray. Information flow and cooperative control of vehicle formations. *Automatic Control, IEEE Transactions on*, 49(9):1465–1476, Sept 2004.

[6] A. Ferrara and C. Vecchio. Sliding mode control for automatic driving of a platoon of vehicles. In *Variable Structure Systems, 2006. VSS'06. International Workshop on*, pages 262–267, June 2006.

[7] J. Fishelson. Platooning Safety and Capacity in Automated Electric Transportation. Master's thesis, Utah State University, 2013.

[8] 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.

[9] S. S. Jackson. Safety Aware Platooning of Automated Electric Transport Vehicles. Master's thesis, Utah State University, 2013.

[10] M. Jovanovic and B. Bamieh. On the ill-posedness of certain vehicular platoon control problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 3780–3785 Vol.4, Dec 2004.

[11] H. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[12] K.-Y. Liang, J. Martensson, and K. Johansson. Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1061–1068, June 2014.

[13] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5967–5972. IEEE, 2010.

[14] F. Pasqualetti, R. Carli, A. Bicchi, and F. Bullo. Identifying cyber attacks via local model information. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5961–5966, Dec 2010.

[15] L. Peppard. String stability of relative-motion pid vehicle control systems. *Automatic Control, IEEE Transactions on*, 19(5):579–581, Oct 1974.

[16] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer, 2011.

[17] W. Ren and D. Green. Continuous platooning: a new evolutionary operating concept for automated highway systems. In *American Control Conference, 1994*, volume 1, pages 21–25 vol.1, June 1994.

[18] T. Robinson, E. Chan, and E. Coelingh. Operating platoons on public motorways: An introduction to the sartre platooning programme. In *17th world congress on intelligent transport systems*, volume 1, page 12, 2010.

[19] R. Sharma, M. Kothari, C. Taylor, and I. Postlethwaite. Cooperative target-capturing with inaccurate target information. In *American Control Conference (ACC), 2010*, pages 5520–5525, June 2010.

[20] S. Sheikholeslam and C. Desoer. Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study. *Vehicular Technology, IEEE Transactions on*, 42(4):546–554, Nov 1993.

[21] D. Swaroop and J. Hedrick. String stability of interconnected systems. *Automatic Control, IEEE Transactions on*, 41(3):349–357, Mar 1996.

[22] H. Tanner. On the controllability of nearest neighbor interconnections. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 2467–2472 Vol.3, Dec 2004.

[23] D. Yanakiev and I. Kanellakopoulos. A simplified framework for string stability analysis in ahs. In *Proceedings of the 13th IFAC World Congress*, pages 177–182, 1996.

APPENDIX

A. CONSENSUS REQUIREMENT IN A BIDIRECTIONAL SYSTEM

We can express the complete bidirectional system as follows. Allow e be the error states:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} x_{i-1} - x_i + \sigma_{\text{ref}} \\ x_{i+1} - x_i - \sigma_{\text{ref}} \\ \dot{x}_{i-1} - \dot{x}_i \\ \dot{x}_{i+1} - \dot{x}_i \end{bmatrix}$$

and for each car using the high-low level controller, we have a computed acceleration as an input ($\ddot{x}_i = a_i$). Then

$$\begin{bmatrix} \dot{e}_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} e_3 \\ e_4 \\ \ddot{x}_{i-1} - u_i \\ \ddot{x}_{i+1} - u_i \end{bmatrix} \tag{18}$$

which can be written in matrix form as:

$$\begin{bmatrix} \dot{e}_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1 \end{bmatrix} u_i + \begin{bmatrix} 0 \\ 0 \\ \ddot{x}_{i-1} \\ \ddot{x}_{i+1} \end{bmatrix} \tag{19}$$

which can then be rewritten as:

$$\dot{e} = Ae + Bu_i + [0\ 0\ \ddot{x}_{i-1}\ 0]^T + [0\ 0\ 0\ \ddot{x}_{i+1}]^T \tag{20}$$

The terms on the right could include inputs from the attackers, which we cannot change. They can be regarded as external disturbances. We only have access to a_i.

Controllability is independent of the feedback controller (or lack thereof) applied. A necessary and sufficient condition for controllability for a linear n-dimensional system $\dot{x} = Ax + Bu$ is

$$\text{rank}([B\ AB\ A^2B\ \dots\ A^{n-1}B]) = n \tag{21}$$

For our 4-dimensional system 20, the rank is only 2. Thus the system is not fully controllable. Only two linear combinations of the four possible states are controllable. Using

controllability staircase form, it can be shown that these controllable modes are

$$\begin{bmatrix} e_1 + e_2 \\ e_3 + e_4 \end{bmatrix} \tag{22}$$

which is the difference between the front and rear separations and its corresponding relative velocity (e_1 is in opposite direction to e_2 and so are e_3 and e_4). Thus, a car can only place itself anywhere in between the two neighboring cars using its own controller.

The system is stable only if the uncontrollable part of the system follows the desired trajectory without control effort. If only one of $(\ddot{x}_{i-1}, \ddot{x}_{i+1})$ is also following the platooning protocol, then the system is stabilizable.

If this is the case, then one of these terms can be considered an input to the system and B becomes:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 1 \\ -1 & 0 \end{bmatrix}$$

and $\text{rank}([B \ AB \ A^2B \ \dots \ A^{n-1}B]) = 4$. Hence, in the bidirectional system, each car (except for the last one) relies on at least one good neighbor to ensure platooning. For the leader, the reference cannot be considered working to stabilize the system. This consensus condition is required in all regular platooning scenarios.

Thus, even the bidirectional structure relies on the other cars cooperating, just as in the unidirectional case. And the actual quantity being controlled, with a symmetric controller, is just the position and velocity in between the two neighboring cars, which is driven to the mid point within that space. It should be noted that the unidirectional case is also a special case of bidirectional system where we choose the rear controller to be zero.