AUTONOMOUS HIGHWAY SYSTEMS SAFETY AND SECURITY

by

Imran Sajjad

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Rajnikant Sharma, Ph.D. Major Professor	Don Cripps, Ph.D. Committee Member
Rees Fullmer, Ph.D. Committee Member	Mark R. McLellan, Ph.D. Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2017

ABSTRACT

Autonomous Highway Systems Safety and Security

by

Imran Sajjad, Master of Science

Utah State University, 2017

Major Professor: Rajnikant Sharma, Ph.D.
Department: Electrical and Computer Engineering

Automated vehicles are getting closer each day to large-scale deployment. It is expected that self-driving cars will be able to alleviate traffic congestion by safely operating at distances closer than human drivers are capable of and will overall improve traffic throughput. In these conditions, passenger safety and security is of utmost importance.

When multiple autonomous cars follow each other on a highway, they will form what is known as a cyber-physical system. In a general setting, there are tools to assess the level of influence a possible attacker can have on such a system, which then describes the level of safety and security. An attacker might attempt to counter the benefits of automation by causing collisions and/or decreasing highway throughput.

These strings (platoons) of automated vehicles will rely on control algorithms to maintain required distances from other cars and objects around them. The vehicle dynamics themselves and the controllers used will form the cyber-physical system and its response to an attacker can be assessed in the context of multiple interacting vehicles.

While the vehicle dynamics play a pivotal role in the security of this system, the choice of controller can also be leveraged to enhance the safety of such a system. After knowledge of some attacker capabilities, adversarial-aware controllers can be designed to react to the presence of an attacker, adding an extra level of security.

This work will attempt to address these issues in vehicular platooning. Firstly, a general analysis concerning the capabilities of possible attacks in terms of control system theory will be presented. Secondly, mitigation strategies to some of these attacks will be discussed. Finally, the results of an experimental validation of these mitigation strategies and their implications will be shown.

(89 pages)

PUBLIC ABSTRACT

Autonomous Highway Systems Safety and Security

Imran Sajjad

This thesis aims to address question of safety and security in autonomous highway systems. Chains of multiple self-driving vehicles (platoons) give rise to inherent vulnerabilities to attacks resulting from their control algorithms. These weaknesses can possibly result in collisions or decreased traffic flow.

This work first provides an overview of these vulnerabilities analyzed from a control systems perspective. The capabilities and extent of damage that an attack can cause are analyzed. Then some possible mitigation strategies are presented that can defend against these attacks.

Simulation results are provided to support the effectiveness of these controllers and an experimental validation is performed. It is concluded that by appropriate controller design, the objectives of platooning in a system under attack can be recovered by using a control scheme designed to withstand attacks.

ACKNOWLEDGMENTS

I am deeply grateful for all the help and support I received from my major professor Dr. Rajnikant Sharma, and the guidance of the Secure Automated Transportation Systems (SATS) group supervisor Dr. Ryan Gerdes.

This project would not have been possible without my friends and lab mates in the SATS group, Soudeh Dadras, Daniel D. Dunn, Samuel Mitchell, Ali A. Alhashimi and the members of the RISC lab, Anusna Chakraborty, Ishmaal Erekson, Spencer Maughan, Sohum Misra, Parwinder Mehrok and Abhishek Manjunath.

I would also like to acknowledge the numerous times I received help from Dr. Todd Moon, Dr. Rees Fullmer and Dr. Don Cripps.

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

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Importance and Safety of Autonomous Highway Systems

Highway congestion in the modern world is an ever growing concern. Vehicle population in the world has been exceeding predictions [3,4]. Coupled with this, the current technology employed in driving and traffic flow management leads to gross inefficiencies. A study by Schrank and Lomax suggests that a single American driver in 1982 would spend 14 hours per year stuck in traffic whereas in 2007, that number rose to 38 hours [5]. Furthermore, it has been found that around three billion gallons of fuel was wasted in 2014 due to traffic congestion [6]. This state of affairs has led to the recent interest in making the current highway infrastructure more efficient and less congestion prone.

Additionally, traffic collisions and fatalities are also a cause for concern. In 2015, the National Highway Traffic Safety Administration put the number of fatalities on U.S. roadways at 35,092 [7]. As such, there is an active interest in making cars safer and highways less dangerous.

This has spurred interest in the research of autonomous highway systems (AHS). The California PATH program has been researching automated vehicles and their interactions on highways since 1986 and has produced significant research [8]. The benefits in terms of fuel economy and safety have been thoroughly analyzed [9].

Since the arrival of automated cars is imminent, protecting them from malicious actors and other threats that seek to disrupt desired operation is of extreme importance. This thesis attempts a specific analysis of the vulnerabilities a string of automated vehicles (platoon) might exhibit and, building upon these results, provides a countermeasure for a specific attack scenario along with a generalized study of optimal attacks.

1.2 Literature Review

AHS and self-driving cars are getting ever closer to real-world implementation. Multiple automated vehicles following each other on a highway with technologies like adaptive cruise control naturally give rise to strings or platoons [10]. In this setting, the vehicles use some control law to adjust the distance between themselves and neighboring cars.

AHS is an area of extensive and ongoing research. In an AHS that uses platooning, 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 [11, 12]. It also helps to alleviate traffic congestion on highways and has shown to be more fuel efficient than manually operated vehicles [13, 14].

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 [15–17]. Chen et al. use optimal control to find an attack in constrained conditions [18]. Grimsman et al. attempt to find a generalized formulation of attacks based on system vulnerabilities [19]. Ramasubramanian et al. investigate a system's resilience to attacks based on its structure [20].

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 [1]. It is shown that a platoon is vulnerable to attacks if an attacker applies a destabilizing control input. This is achieved by the attacker modifying controller gains in order to cause instability. Only the gains in one vehicle need to be modified to achieve this. This sort of an attack is also presented by Dunn, where a group of colluding attackers cause string instability and increase traffic congestion [21].

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 [22–25]. The majority of prior research in platooning involves a specific control law that is used to either regulate distance from the front (unidirectional)

or from both front and back (bidirectional) [26]. Other work has highlighted some of the limitations of the bidirectional structure [27,28]. Different inter-vehicle spacing policies have also been considered [25]. It has been previously shown that the symmetric bidirectional linear controller causes the bounds on the front and rear errors increase as the number of vehicles increases [27]. There are also possibilities for inter-vehicle communication that add an additional layer of abstraction.

Sliding Mode Control has been used previously in many scenarios. Platooning strategies exist where sliding mode control has been used in a homogeneous platoon under normal operation [29]. Graph theoretic approaches similar to the one presented here have been used before in platooning [30,31]. They have also been used in general problems of multiple vehicle target tracking in the presence of uncertainties [32].

Apart from platooning, much work has been done in interconnected dynamic CPS. The security and robustness of these systems in the face of an attack or failure is crucial and an active area of research [16]. Graph theory, information flow analysis have been used to analyze such systems as well [30, 31]. Much of this work is focused on ensuring suitable operating conditions for dynamic systems, mainly stability, controllability and observability.

Game theory has been used in the field of control systems for some time now. There are classic problems in game theory such as pursuit evasion that have been adapted to modern control problems [33]. The area has also been used in the area of system design and disturbance rejection [34, 35]. Game theory is also used in communications and designing data networks [36].

1.3 Physical Description

For the purposes of this thesis, the following system description will be used unless otherwise stated [37]. Consider the platoon of n vehicles shown in Fig 1.1. The state vector is

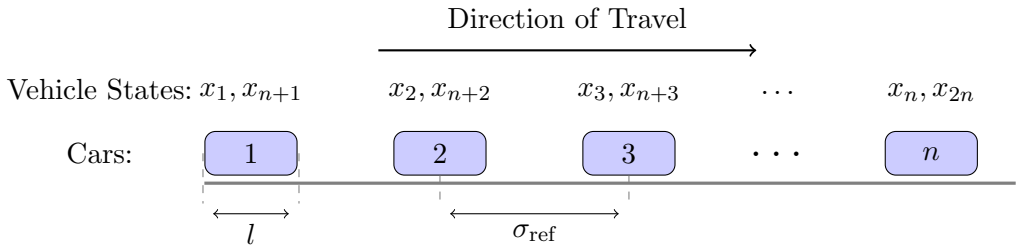


Fig. 1.1: A Platoon of n Vehicles. Each Car is l Meters Long and the Desired Separation from Center of One Car to that of the Other is σ_{ref}. Car n is the Leader.

$$\begin{aligned} x &= \begin{bmatrix} x_1 \; x_2 \; \ldots \; x_n \; x_{n+1} \; x_{n+2} \; \ldots \; x_{2n} \end{bmatrix}^T, \\ u &= \begin{bmatrix} u_1 \; u_2 \; \ldots \; u_n \end{bmatrix}^T \end{aligned} \tag{1.1}$$

and the complete linear time-invariant (LTI) system is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ A &= \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} \quad B = \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} \end{aligned} \tag{1.2}$$

where car i has position and velocity x_i, x_{n+i} respectively and control input u_i. In chapter 3, the x_{n+i} are replaced by v_i. These positions are measured from the center of mass of all the cars. Each car is essentially a double integrator in this setting.

With this description, the dynamics of the system itself can be expressed in a general manner. Adding the controller closes the loop on the system by finding an expression for u, and the system becomes autonomous [38].

The following chapter uses this description and a bidirectional controller to see what sort of attack capabilities are achieved by one malicious car.

CHAPTER 2
REACHABLE SET FOR AN ATTACKER

2.1 Introduction

Using the most general idea for an attack, the question of the capabilities of an attacker can be asked. One such question, which this chapter attempts to answer, asks what state configurations are achievable by an attacker in finite time with a bounded control input.

It should be noted that this question has been asked before for dynamical systems in general. Pontryagin proved the maximum principle which results in a set of necessary conditions for a optimality [39]. Bellman arrived at a similar result using a dynamic programming approach [40]. Both these methods can be extended to provide a sufficient condition for reachability, but finding this solution in the case of constraints such as bounded control can be difficult[1].

Note that the system in (1.2) does not have a control architecture. For the purposes of this chapter, the bidirectional controller given by Yanakiev and Kanellakopoulos will be used [26]. It is restated here for clarity

$$\begin{aligned} u_i = k_p(x_{i+1} - x_i - \sigma_{\text{ref}}) + k_p(x_{i-1} - x_i + \sigma_{\text{ref}}) + \\ k_d(v_{i+1} - v_i) + k_d(v_{i-1} - v_i) \end{aligned} \tag{2.1}$$

This imposes a structure on A that can be expressed as follows

[1] Under a well informed but somewhat idealized choice of functions, both these methods result in the linear-quadratic regulator (LQR) [41, 42].

$$A = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ -k_p L_{n\times n} & -k_d L_{n\times n} \end{bmatrix} \text{ where } \quad L_{n\times n} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix} \tag{2.2}$$

As is noted by Barooah and Hespanha, this matrix structure is similar to the Laplacian of the system from graph theory [27]. The last entry in L is different because the leader has access to a reference trajectory.

But supposing that there is an attacker, the system can still be analyzed with the remaining cars forming an autonomous system and the attacker has one control input. In other words, all cars except the attacker have their respective u_i set according to (2.1), and the attacker has one control input which is denoted simply u. Without loss of generality, the attacker can be at position j.

Thus, for the purposes of this chapter, the system can now be written as

$$\dot{x} = Ax + Bu \tag{2.3}$$
$$A = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ -k_p L_{n\times n} & -k_d L_{n\times n} \end{bmatrix} \quad B = \begin{bmatrix} 0_{n\times 1} \\ I(j)_{n\times 1} \end{bmatrix}$$

and

$$L_{n\times n} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix} \tag{2.4}$$

and $I(j)_{n\times 1}$ is one at the jth entry and zero elsewhere. Thus the jth row of L has been zeroed out, which means the attacker is not following the regular control law. At this point, there is no need to separate positions and velocities, so from now on n is redefined to be the size of the state vector x.

Also one more very important consideration is that the attacker's control input is bounded in the infinity norm, which means

$$\|u\|_\infty = \max_t |u(t)| \leq c \tag{2.5}$$

Thus, the attacker cannot apply infinite acceleration. The maximum allowed value in each direction is set by choosing the value of c appropriately. This condition is extremely important, since if infinite control input is allowed, then the controllable set would be the entire state space [43]. Hence, the following sections deal with attempting to find what state configurations are reachable in finite time, with bounded control input.

The notion of a controllable set in continuous time can also be somewhat illuminated using the controllability matrix of an LTI state space model. For the given system, a test of controllability does indicate all the states are controllable, but the condition number of this matrix is not good, which means that some directions in the state space are more controllable than others.

2.2 Discrete-Time Case: For a Given Final State

Given a final time t_f and a time increment dt, conversion to a discrete-time state-space model is always possible with an LTI system,

$$x(t+1) = Ax(t) + Bu(t) \tag{2.6}$$

where $x \in \mathbb{Z}^+ \to \mathbb{R}^n,\ u \in \mathbb{Z}^+ \to \mathbb{R}\ A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times 1}$, then the solution (state) at any final time index p can be written as

$$x(p) = H_{n\times p}u_{p\times 1} = \begin{bmatrix} B & AB & A^2B & \dots & A^{p-1}B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(p) \end{bmatrix} \tag{2.7}$$

Thus, p and $x(p)$ are given as requirements, and H can be computed using the system data. So a vector u that satisfies (2.7) and the discrete-time equivalent (2.5) is required. In other words, the problem can be expressed as

$$\begin{aligned} &\text{find } u \\ &\text{s.t. } x = Hu\ ,\ \|u\|_\infty \le c \end{aligned} \tag{2.8}$$

Because H is a flat matrix and has row rank n, there are many possible u's. In fact, the set of u's form a vector space of their own, with dimension at least $p - n$. Note that if $p = n$, the solution could be a unique point, and if that u is not realizable, then nothing can be done. Higher p means more possible solutions. The important question is this: does increasing p help to find a solution that meets the constraints, or is there a brick wall somewhere?

Define $x = x(p)$. If there is a constraint such as $\|u\|_\infty \le c$ for some value of c, a feasibility problem in some solver can be solved, but here a somewhat simpler method is

proposed here. Forming the norm minimization problem as below

$$\begin{aligned} &\min_{u} \|u\|_\infty \\ &\text{s.t. } x = Hu \end{aligned} \tag{2.9}$$

This can be recast as a linear programming problem

$$\begin{aligned} &\min_{u,s} s \\ &\text{s.t. } x = Hu \\ &-s \leq u \leq s \end{aligned} \tag{2.10}$$

The variable s bounds each value of u and the optimal value of this problem is the minimum value of s. If s is greater than c, then it can be said that the pair (p, x) is not reachable. This problem is a convex optimization, which means that strong duality should hold in most cases (Slater's condition for strong duality) [44]. If the unconstrained dual problem is formed

$$\min_{\lambda_1,\lambda_2,\nu} \min_{u,s} s + \lambda_1^T(u - t) + \lambda_2^T(-u - t) + \nu^T(x - Hu) \tag{2.11}$$

an analytical solution might be possible. The superscript ^{T} denotes the vector or matrix transpose. But before evaluating the vector u, the optimal value of (or at least a lower bound on) s can be found, which if higher than c, will indicate that the given final time and state (p, x) are not achievable.

2.2.1 The Least Squares Solution is not the Infinity Norm Solution

For the system in the previous section, $x = Hu$ has many solutions and $u = (H^T H)^{-1} H^T x$ is only one of them. This is a special solution that corresponds to minimum $\|u\|_2$. While it is tempting, the least-squares solution and the infinity norm solution are not the same.

Furthermore, if it is suggested that the two are close enough to each other, the only norm identity for both is

$$\|u\|_\infty \leq \|u\|_2 \leq \sqrt{p}\|u\|_\infty \tag{2.12}$$

which essentially means that while the approximation might look good in two or three dimensions, it does get worse with increasing p, and it is desired that $p \gg n$.

2.3 Discrete-Time Case: The Range of Final States

The u without a subscript is the control input. For the expression in (2.7), a singular value decomposition can be used as

$$\begin{aligned} x &= Hu \\ &= U\Sigma V^T u \\ &= \begin{bmatrix} \vdots & \vdots & & \vdots \\ u_1 & u_2 & \dots & u_n \\ \vdots & \vdots & & \vdots \end{bmatrix}_{n\times n} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \dots & 0 \\ 0 & 0 & \dots & \sigma_n & \dots & 0 \end{bmatrix}_{n\times p} \begin{bmatrix} \dots & v_1 & \dots \\ \dots & v_2 & \dots \\ & \vdots & \\ \dots & v_p & \dots \end{bmatrix}_{p\times p} w \end{aligned} \tag{2.13}$$

By definition, U has full rank and spans the entire space of x. So x can be written as a linear combination of the u_i vectors.

$$\begin{aligned} x &= \sum \gamma_i u_i \\ &= \begin{bmatrix} \vdots & \vdots & & \vdots \\ u_1 & u_2 & \dots & u_n \\ \vdots & \vdots & & \vdots \end{bmatrix}_{n\times n} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{bmatrix}_{n\times 1} \end{aligned} \tag{2.14}$$

If (2.13) is to be written as (2.14), the γ's can be expressed as:

$$\gamma_i = \sigma_i v_i^T u \tag{2.15}$$

An orthonormal basis for U and V is always possible, which leads to $\|v_i\|_2 = 1 \ \forall i$. Under the constraint $\|u\|_\infty < c$, it is possible to write

$$\begin{aligned} \|v_i^T u\|_\infty &\leq \|v_i\|_\infty c \\ \|\gamma_i\|_\infty &\leq \sigma_i \|v_i\|_\infty c \end{aligned} \tag{2.16}$$

and plugging this back into (2.14), gives the span of x in each of the directions given by U. This would give an estimate on the set of achievable states for a given final time and an acceleration constraint. While not an exact result, it motivates the fact that the left eigenvector directions span something possibly close to the reachable set. The same question as before of adjusting the final time p so that more states are achievable, might be reduced to looking at the singular values of H.

2.4 Discrete-Time Case: Simulation

First off, the effect of increasing the final time has on the singular values of H is investigated. Then the effect that singular values have on the reachable states is discussed. For everything in this section, there is a single input from a lone attacker. The matrices for the simulation were constructed with the following values from Table. 2.1

2.4.1 Effect of Final Time on Singular Values

The singular values of the transfer matrix H are plotted against final time values in Fig. 2.1. It is observed that increasing the final time does not increase the singular values of the transfer matrix after a point. In fact, what is seen is a sharp ceiling and very quick convergence to a maximum value. For larger platoons, the singular values take longer to converge, but they still follow the same bounded pattern.

Table 2.1: Matrix Parameters

k_p	k_d	n
1	2	8

To see the effect of discretization, multiple sampling times were tried. Interestingly, the infinity norm of H is invariant to the sampling time chosen.

Because of this hard-bounding trend, it might be possible to derive an analytical expression for the bound on the largest singular value over time. This would have to take into account the structure of A and B, the final time and the discretization.

2.4.2 Reachable Sets for a Bounded Control Input

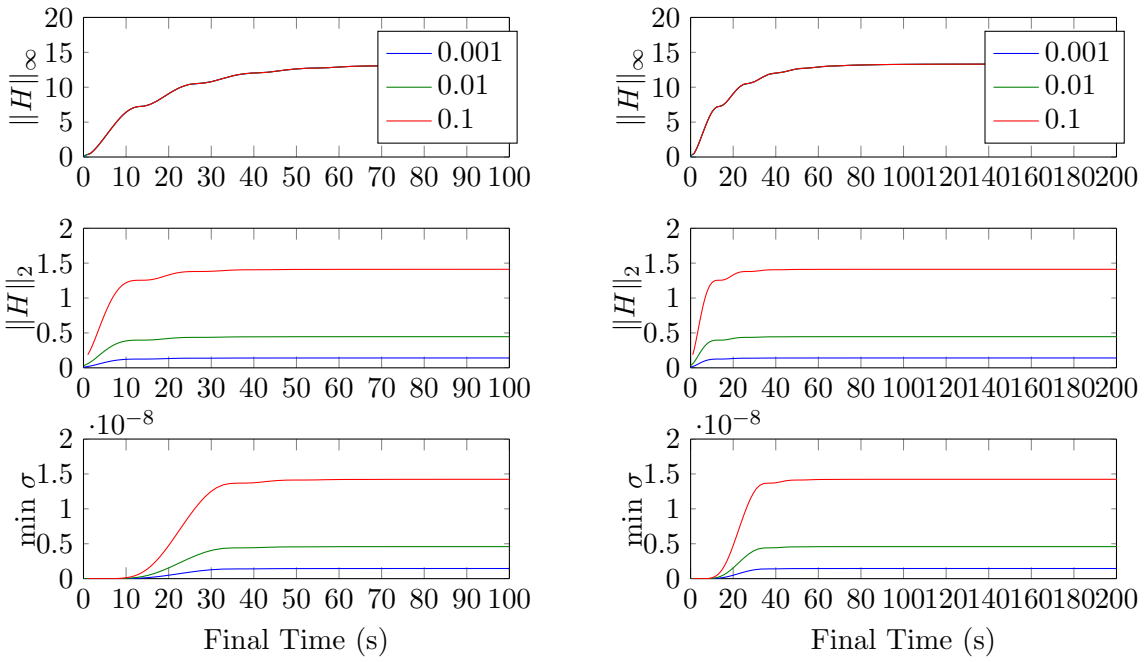
In order to find the reachable states for a given bound on u, the problem can be solved backwards.

Choosing the final state as one of the left eigenvectors $x = u_i$, there is a control input u that optimally solves the problem (2.10) and achieves the desired state, and for this control input, $\|u\|_\infty = s_i^*$.

$$\begin{aligned} u_i &= Hu \\ \frac{c}{s_i^*}u_i &= \frac{c}{s_i^*}Hu \\ \frac{c}{s_i^*}u_i &= H\frac{c}{s_i^*}u \end{aligned} \tag{2.17}$$

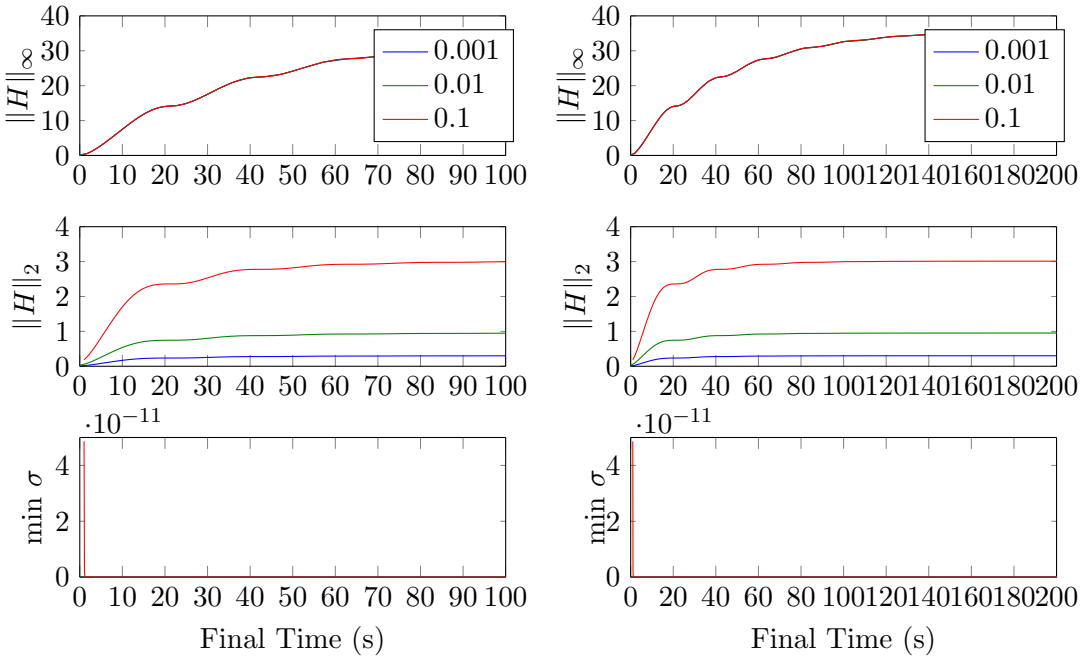
The input here is just scaled so that $\|\frac{c}{s_i^*}u\|_\infty \leq c$, with the associated achievable state $\frac{c}{s_i^*}u_i$. Again u without the subscript is a control input, while u_i is a column of U. The optimality of the solution in (2.17) is proved later in this chapter. Thus the maximum reach in the u_i direction is given by $\frac{c}{s_i^*}u_i$ with a $\|u\|_\infty \leq c$.

Solving the optimization problem given by (2.10) for the following final states $(u_1, u_2, \ldots, u_n)$, gives the minimum required power to get to each column of U. The solver CVX was used to find the optimal controls [45]. It is to be noted that the problem gets very time-consuming for larger platoons and longer final times. For a final time of 5 seconds and a platoon of



(a) Platoon size 6 ($n = 12$), $\max t_f = 100$

(b) Platoon size 6 ($n = 12$), $\max t_f = 200$



(c) Platoon size 10 ($n = 20$), $\max t_f = 100$

(d) Platoon size 10 ($n = 20$), $\max t_f = 200$

Fig. 2.1: Singular Values of H with Increasing Final Time. Multiple Traces for Different Sampling Times (dt).

size 4, Fig. 2.2 shows the control effort and errors required to get to each state.

The singular values are always ordered from largest to smallest and it can be seen that the direction with the first singular value is most easily achievable. For the last three, the solver gives up. It is a characteristic of infinity-norm-minimization problems that the solutions look like bang-bang principles [43].

Given these values, it was attempted to compare the inverse of the required powers (s_i^*'s) to the singular values of the matrix.

From Fig. 2.3, it is seen that after some scaling, the singular values follow the same trend as the inverse of required power. Thus the singular values tend to correlate with a bound on the achievable states in the directions of U. This sufficiently implies that

$$\begin{gathered} x = (1+\epsilon)C\sigma_i u_i \text{ is unachievable for} \\ \|u\|_\infty < c,\ \epsilon > 0,\ \forall\ i,\ C = C(c) \end{gathered} \tag{2.18}$$

where C should monotonically increase with c and be independent of p or the sampling time.

Putting all of this together with the proofs in the next section, the following holds.

The reachable set with $\|u\|_\infty < c$ is contained within the following set for some monotonically increasing function $C(c)$.

$$\{x : x = \sum_{i=1}^{n} C\epsilon_i\sigma_i w_i, |\epsilon_i| \leq 1\ \forall\ i\} \tag{2.19}$$

where each w_i is the normal vector to a supporting hyperplane between the convex hull $\text{conv}(C\sigma_i u_i)$ and $(1+\epsilon)C\sigma_i u_i$.

This region is defined by a rotated hypercube with the length in each direction u_i given by $C\sigma_i$. These values are easily computed using the singular value decomposition of H.

It requires a lot of imagination to visualize this in eight dimensions. However for the (unrealistic for platooning) case of three dimensions, Fig. 2.4 shows a shape of this type.

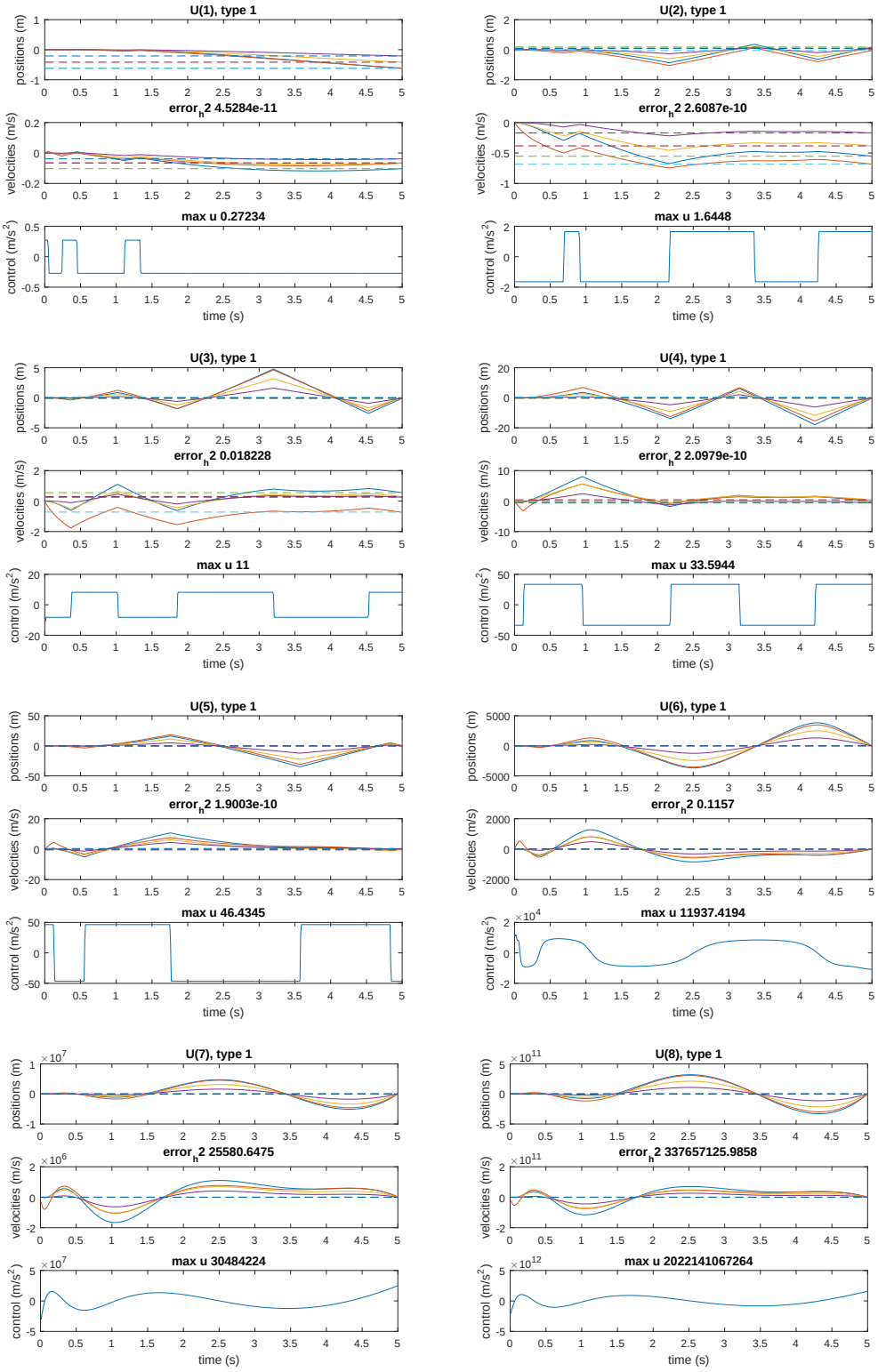


Fig. 2.2: Control Effort and States for each Column of U $(n = 8, t_f = 5, dt = 0.01)$

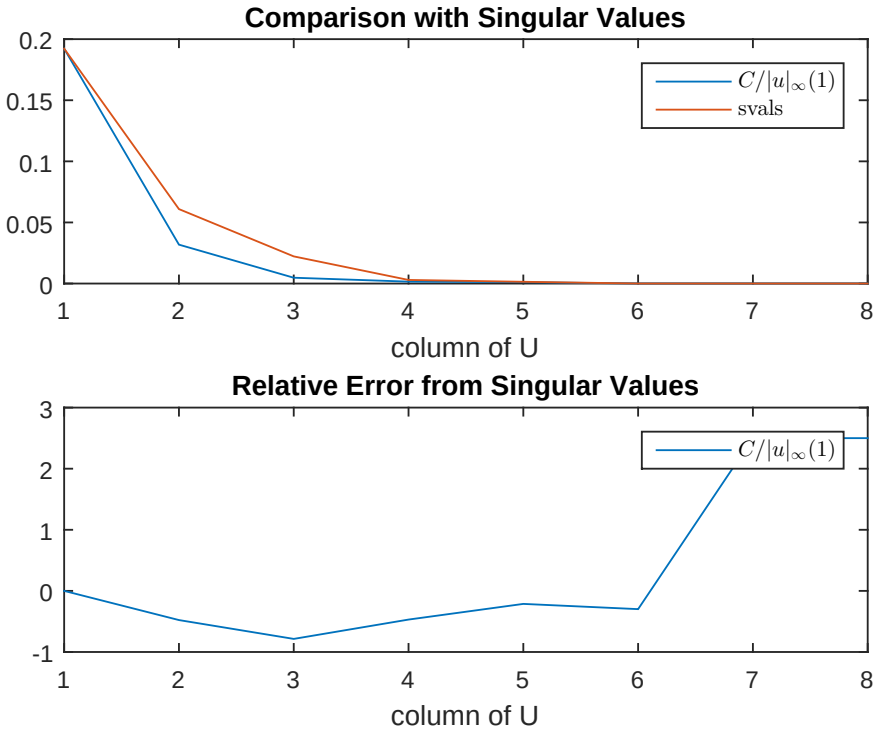


Fig. 2.3: Singular Values of H Compared to the Inverse of Required Powers for each Column u_i.

2.5 Proofs

Most of these properties also follow from the geometry of the reachable set established in [43]. The set is convex and symmetric, even for bounded control inputs.

2.5.1 Reachable Set Bounded by Hyperplanes

The reachable set is convex. It has been established that $(1+\epsilon)C\sigma_i u_i$ is an unreachable set. Hence by the supporting hyperplane theorem, there must exist a plane that bounds the reachable set that touches the point $C\sigma_i u_i$ in the state space. This is true for all i. The reachable set is symmetric as well, so there is another hyperplane on the exact opposite side (also called a slab). Hence the reachable set is an intersection of n slabs. This can be visualized with the picture in Fig. 2.5.

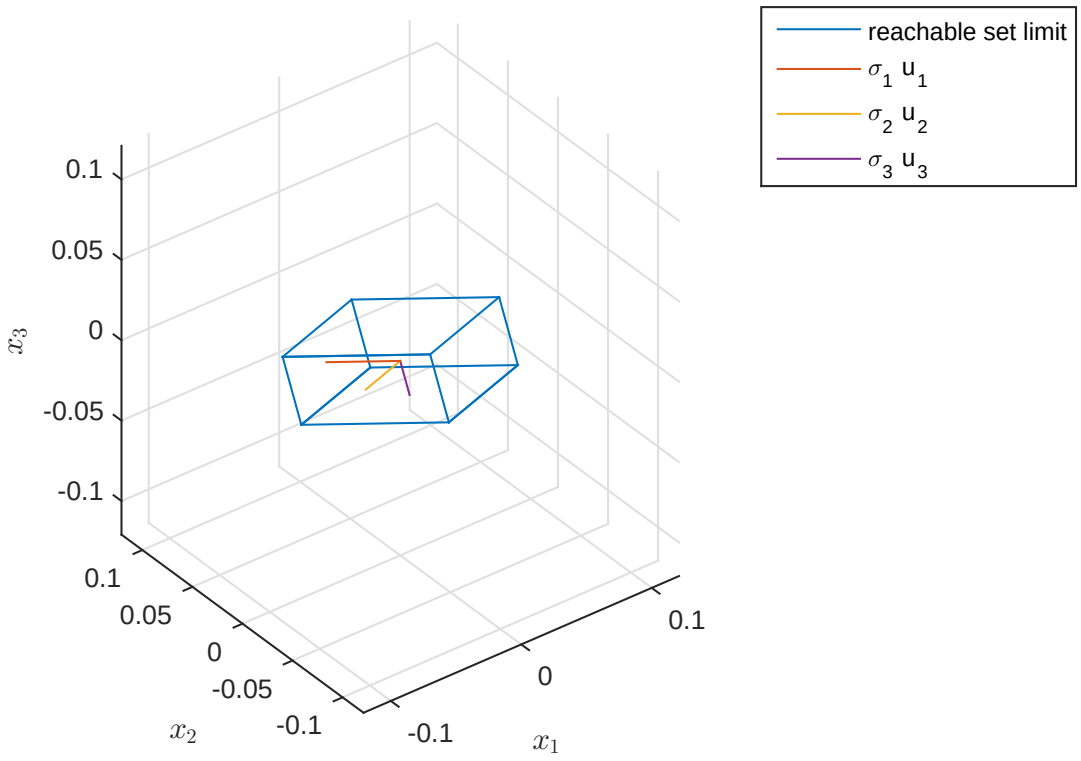


Fig. 2.4: Reachable Set Boundary in Three (Unrealistic) Dimensions.

2.5.2 Lower Bounds on Control Input

Given $u_i = Hu = U\Sigma V^T u$, and expressing the optimal u with V as a basis, $u = Va$, with $a \in \mathbb{R}^p$ a vector of optimal coefficients, with ϵ close to zero.

$$\begin{aligned}(1+\epsilon)u_i &= (1+\epsilon)U\Sigma V^T Va \\ &= (1+\epsilon)U\Sigma a \\ &= (1+\epsilon)u_i\sigma_i a_i \\ &= \sigma_i u_i(1+\epsilon)a_i\end{aligned} \tag{2.20}$$

Note that for a desired $(1+\epsilon)u_i$, only the a_i entry has to be scaled by the same amount. The other a_i's up till n have to remain the same (zeros in this case). The second last step

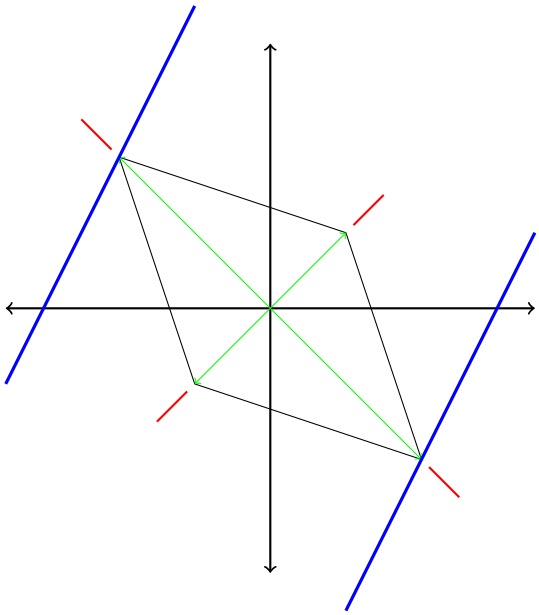


Fig. 2.5: Supporting Hyperplanes Shown in Blue. The Green Vectors are Optimally Reachable in the State Space, While the Red Areas are Not. The Reachable Set has to Exist Between Planes of This Sort.

above is necessary because U is orthonormal.

So it can be concluded that since V is orthonormal, no change in any other a can offset the change caused by a_i in the infinity norm. This indicates that at least with x in the direction of the left eigenvectors, optimal solutions of the problem (2.10) for different values of c simply scale.

2.5.3 Note on Possible Solutions

For any given final state and using V as a basis for the control input, the following relation holds

$$\begin{aligned} x = U\Sigma V^T V a = U\Sigma a = U\Sigma_{1:n} a_{1:n} \\ a_{1:n} = \Sigma_{1:n}^{-1} U^T x \end{aligned} \tag{2.21}$$

This means that the first n out of p values for a are constrained by the terminal state, while the remaining $p - n$ values have no effect on the final state. This is true no mat-

ter what sort of minimization is done, infinity-norm, least-squares etc. The remaining a's have a role in minimizing whatever objective function is chosen. Furthermore, observe that $V = [H_d \mid H_n]$, which is the concatenation of the domain and nullspace of H (easily verified). Hence $u = H_d a_{1:n} + H_n a_{n+1:p}$. Hence, by choosing the basis functions appropriately, traversing along $H_n a_{n+1:p}$ minimizes any chosen cost while keeping x fixed.

This can be further elucidated; consider minimizing over the 2-norm. Then

$$u = Va \implies \|u\|_2 = \|Va\|_2 = \|a\|_2 \tag{2.22}$$

because V is unitary and 2-norm preserving. Hence the solution to this is just to set $a_{n+1:p} = 0$. For example, in Fig. 2.6.

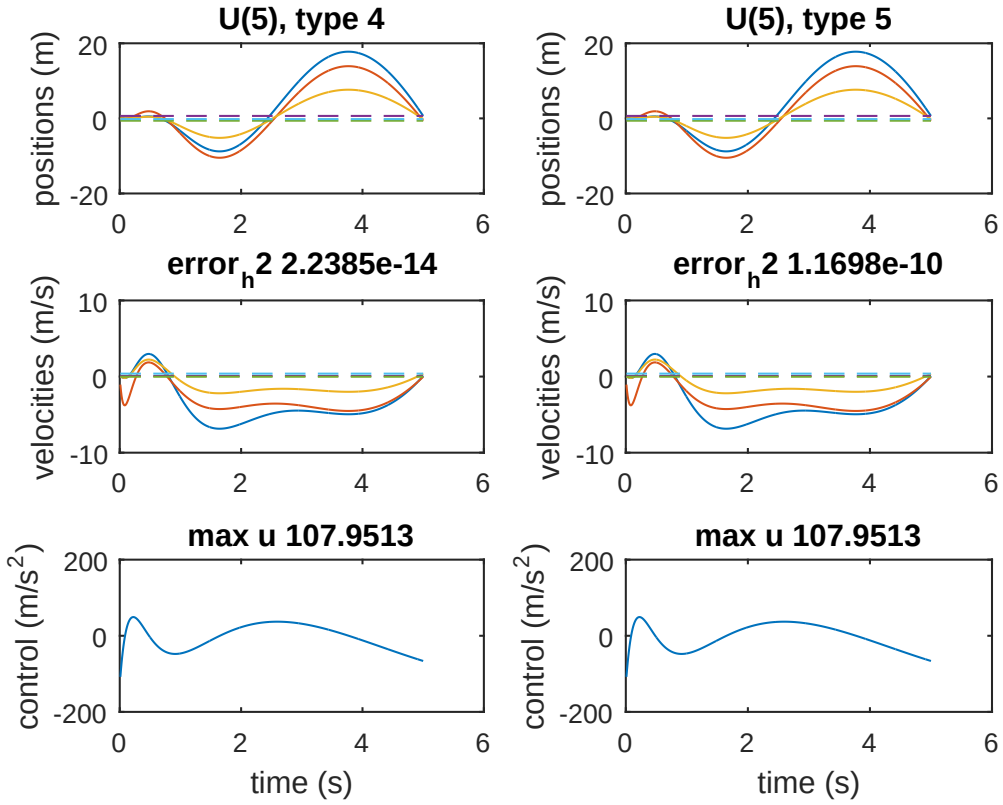


Fig. 2.6: Analytic Solution (Left), Solver Solution (Right) to 2-norm Minimizing Problem.

Perhaps this choice of basis can be used to solve the continuous time problem as well, with additional terms included for norm minimization.

Since $a_{1:n}$ is given, the optimization problem can be rewritten as follows.

$$\min_{a_{n+1:p}} \|H_d a_{1:n} + H_n a_{n+1:p}\|_\infty \tag{2.23}$$

2.5.4 Lower Bounds from Dual Problem

An alternate way to establish these properties is to form the dual optimization problem [44]. The dual of the LP given by (2.10) is

$$\begin{aligned} &\max_{\lambda_1,\lambda_2,\nu} (1+\epsilon)\nu^T x \\ \text{s.t. } &(1 - \lambda_1^T 1 - \lambda_2^T 1) = 0 \\ &(\lambda_1 - \lambda_2 - H^T\nu) = 0 \end{aligned} \tag{2.24}$$

From duality, the optimal value of this function lower bounds the optimal value of the primal problem ($\|u\|_\infty$). By setting $\epsilon = 0$, the original problem is obtained. So for any solution $(1+\epsilon)x$, the lower bound on $\|u\|_\infty$ increases with $(1+\epsilon)$. This means that optimal solutions scale up with the magnitude of the final state.

Thus if $(c/s_i^*)u_i$ is at the boundary of what is achievable, any state directly above that will not be achievable. For $x = u_i + \epsilon u_j$ with $i \neq j$ and s_i^* the optimal value for u_i,

$$\begin{aligned} &\max_{\lambda_1,\lambda_2,\nu} \nu^T u_i + \epsilon\nu^T u_j \\ \text{s.t. } &(1 - \lambda_1^T 1 - \lambda_2^T 1) = 0 \\ &(\lambda_1 - \lambda_2 - H^T\nu) = 0 \end{aligned} \tag{2.25}$$

is also maximized with a value greater than that of $\nu^T u_i$. Hence the lower bound increases and a vector with a projection on u_i greater than the limit set by $\sigma_i u_i$ is not reachable.

CHAPTER 3
ATTACK MITIGATION USING DETECTION-BASED SLIDING MODE CONTROL

3.1 Introduction

The content of this chapter is a reproduction of previous work by Sajjad et al. which presents the same approach and results [37].

This chapter proposes 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, the proposed 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. This 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 approach employed here builds upon previous works and tries to solve the safety problem in an adversarial environment. While the system analyzed is linear, the choice of a sliding mode controller follows naturally when some limitations on the attacker capabilities are known. Also, maximum performance constraints are incorporated in order to measure the efficacy of this approach by measuring the severity of any collisions that take place.

Firstly, a threat model in the context of a vehicular platoon is established. Secondly, a sliding-mode controller and an attack detection scheme is shown. Lastly, simulation results which show the efficacy of this approach are presented.

3.2 Threat Model

For the purposes of this study, a platoon of n members is considered, 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

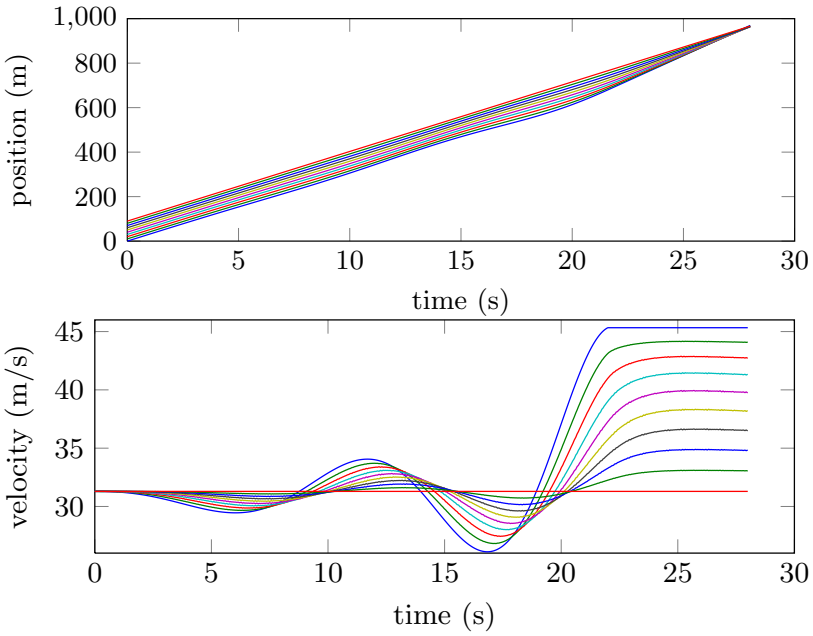


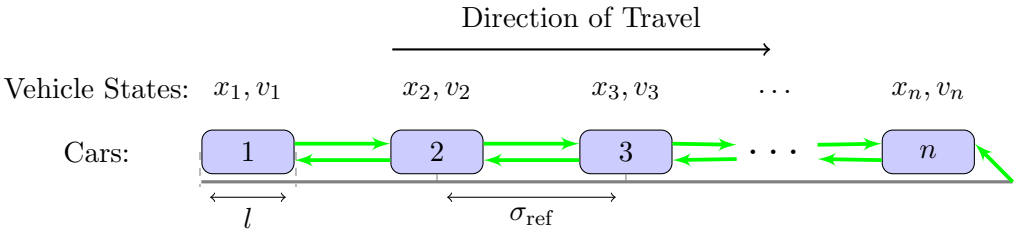
Fig. 3.1: Oscillatory Behavior Brought on by an Attacker, Resulting in a High Speed Crash [1]. Each Line Represents the Trajectory of a Vehicle in a Ten Vehicle Platoon with an Attacker at the Rear.

described in the next section. The last member is indexed as 1 and the leader is at index n. The bidirectional platoon scheme [26] is considered, 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.

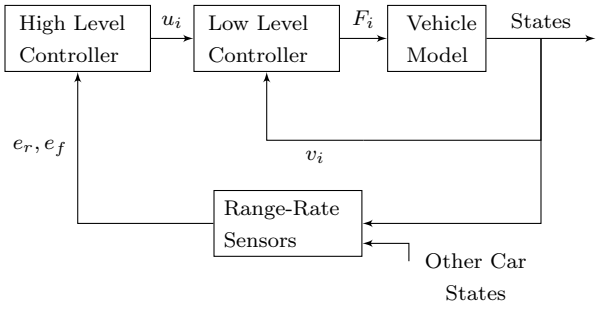
A single attacker in control of a car at an arbitrary position in the platoon is considered. The attack car is possibly more powerful than the regular cars, i.e. it may have greater acceleration capabilities. The goal of the attacker is to cause multiple collisions in time. To accomplish this the attacker follows a modified control law that induces oscillations in the platoon (Fig. 3.1). 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 [1]. 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 the tests presented.

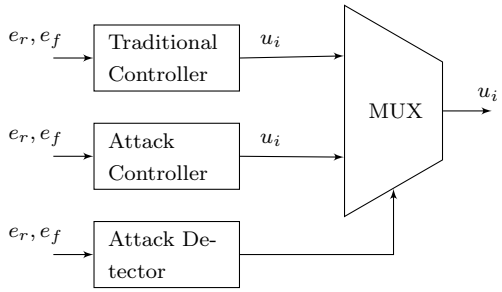
3.3 Rationale and System Overview



(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.

Fig. 3.2: 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.

The bidirectional platoon structure (Fig. 3.2a) 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 [26]. It can be shown that this structure is especially vulnerable to attack, but to retain its benefits, a scheme of altering the structure can be used 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 [1, 26].

3.3.1 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, revised platooning goals are defined 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, how errors propagate in the system in the presence of attackers needs to be investigated. 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 [22, 24, 38]. 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.

Priority is given 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. All that needs to be ensured is that each car does not collide with its neighbors. To achieve this, a decentralized controller is designed in section 3.4 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 [38]. 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, this work proposes 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 [46, 47]. To measure the accumulation of damage, the following rate of change to a state D can be used:

$$\dot{D} = c^T v_{\text{rel}} \tag{3.1}$$

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.

3.3.2 Bidirectional Platooning Control

In keeping with the current literature [27, 48], 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

$$\begin{aligned} x &= \begin{bmatrix} x_1 & x_2 & \dots & x_n & v_1 & v_2 & \dots & v_n \end{bmatrix}^T, \\ u &= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix}^T \end{aligned} \tag{3.2}$$

and the state space system becomes

$$\dot{x} = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} x + \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} u \tag{3.3}$$

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

$$u_i = f_i(x_{i-1} - x_i, v_{i-1} - v_i, x_{i+1} - x_i, v_{i+1} - v_i) \tag{3.4}$$

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. 3.2b); 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 (3.4) wherein the rearmost car uses a unidirectional law, and the leader follows a reference trajectory while maintaining a follower separation

$$u_1 = f_1(x_{i+1} - x_i, v_{i+1} - v_i) \tag{3.5}$$

$$u_n = f_n(x_{i-1} - x_i, v_{i-1} - v_i, x_{\text{ref}} - x_i, v_{\text{ref}} - v_i) \tag{3.6}$$

3.3.3 Vehicle Model

The previous section assumes that a desired acceleration can be achieved and applied directly to the 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 the possessed 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. 3.2b.

The vehicle model used 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 [22, 23, 28].

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= \alpha F_i - \beta v_i \end{aligned} \tag{3.7}$$

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 [22, 48].

For the high level controller described in (3.4) to work, the internal dynamics of the vehicle need to be compensated. Feedback linearization is to compensate for terms in the model described by (3.7) [22, 48]. This gives

$$F_i = \frac{1}{\alpha}\left(u_i + \beta v_i\right) \tag{3.8}$$

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. Simulations are done with these saturation limits in order to demonstrate the controller on a realistic system where the desired acceleration cannot always be achieved. A favorable consequence of this is that the model cannot achieve infinite acceleration.

Substituting (3.8) into (3.7) gives the required double integrator type system for each vehicle

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= u_i \end{aligned} \tag{3.9}$$

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

Additionally, another constraint on this controller which prohibits reverse motion is applied. 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.

3.3.4 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 each car respond to the movements of the vehicle in front of it only. In a three-member 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, it can be seen 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, as shown in the next subsection, the bidirectional system formed around a single car is locally uncontrollable and that at least one cooperating neighbor is required for stable operation.

If the discussion is limited to a single attacker, this work proposes to use the consensus condition that is required in both cases anyway. This thesis recommends 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. 3.2c). This approach is shown to allow a straightforward, ultimate boundedness analysis and simulation results show that it greatly reduces total damage compared to a traditional bidirectional scheme.

3.3.5 Consensus Requirement in a Bidirectional System

For the purposes of this section, the complete bidirectional system can be expressed as follows. Allowing e be the error states:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} x_{i-1} - x_i + \sigma_{\text{ref}} \\ x_{i+1} - x_i - \sigma_{\text{ref}} \\ \dot{x}_{i-1} - \dot{x}_i \\ \dot{x}_{i+1} - \dot{x}_i \end{bmatrix} \tag{3.10}$$

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

$$\begin{bmatrix} \dot{e}_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} e_3 \\ e_4 \\ \ddot{x}_{i-1} - u_i \\ \ddot{x}_{i+1} - u_i \end{bmatrix} \tag{3.11}$$

which can be written in matrix form as

$$\begin{bmatrix} \dot{e}_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1 \end{bmatrix} u_i + \begin{bmatrix} 0 \\ 0 \\ \ddot{x}_{i-1} \\ \ddot{x}_{i+1} \end{bmatrix} \tag{3.12}$$

which can then be rewritten as:

$$\dot{e} = Ae + Bu_i + [0\ 0\ \ddot{x}_{i-1}\ 0]^T + [0\ 0\ 0\ \ddot{x}_{i+1}]^T \tag{3.13}$$

The terms on the right could include inputs from the attackers, which cannot be influenced. They can be regarded as external disturbances. Only a_i is accessible.

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

$$\text{rank}([B\ AB\ A^2B\ \ldots\ A^{n-1}B]) = n \tag{3.14}$$

For the 4-dimensional system (3.13), the rank is only 2. Thus the system is not fully controllable. Only two linear combinations of the four possible states are controllable. Using the controllability staircase form, it can be shown that these controllable modes are

$$\begin{bmatrix} e_1 + e_2 \\ e_3 + e_4 \end{bmatrix} \tag{3.15}$$

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 modes follow 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:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 1 \\ -1 & 0 \end{bmatrix} \tag{3.16}$$

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 the rear controller is turned off.

3.4 Attack Controller

For the secondary controller that governs vehicle response when under attack, this section proposes 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.

The controller incorporates uncertainties and bounds on acceleration and velocity based

on the model in the previous section and derived is a suitable sliding mode controller. Then its continuous approximation is used 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.

3.4.1 Mathematical Preliminaries

To demonstrate the efficacy of the sliding mode controller, some mathematical preliminaries from control systems theory are required. 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 initially bounded. It is asymptotically stable if $\|x\| \to 0$ as $t \to \infty$ (Def. 4.1, [38]).

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, [38]). 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, u is chosen 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, [38]). Thus, a bound is possible 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 followed 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

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

front separation to mitigate attacks from either direction.

3.4.2 Single Controller Design

The design of a front error controller follows. The next car's acceleration is not assumed to be known. This car could very well be an attacker so only the bounds on this quantity are used to derive the controller. Defining error coordinates in the frame of car i with the desired separation σ_{ref} in meters

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} x_{i+1} - x_i - \sigma_{\text{ref}} \\ v_{i+1} - v_i \end{bmatrix} \tag{3.17}$$

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, the following state space model can be used.

$$\begin{bmatrix} \dot{e_1} \\ e_2 \end{bmatrix} = \begin{bmatrix} e_2 \\ \ddot{x}_{i+1} - u_i \end{bmatrix} \tag{3.18}$$

If a sliding manifold is defined as $s = k_1 e_1 + e_2$, $s = 0$ 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,

$$V_s = \frac{1}{2} e_1^2 \tag{3.19}$$

$$\dot{V}_s = e_1 \dot{e}_1 = e_1 e_2 = -k_1 e_1^2 \tag{3.20}$$

which implies asymptotic stability (Thm. 4.1 [38]). Outside of this manifold, the Lyapunov Candidate $V = \frac{1}{2} s^2$ is used to check if the system reaches the line $s = 0$, and

$$\begin{aligned} \dot{V} &= s\dot{s} \\ &= s(k_1 e_2 + \ddot{x}_{i+1} - u_i) \\ &\leq \|s\| \left(k_1 \|e_2\| + \|\ddot{x}_{i+1}\|\right) - s(u_i) \end{aligned} \tag{3.21}$$

Now the attacker's constraints can be plugged in. With $\|e_2\| \leq 2v_{\max}$ and $\ddot{x}_{i+1} \leq a_{\max}$ [2] the controller becomes

$$u_i = \text{sat}\left(\frac{s}{\epsilon}\right)[2k_1 v_{\max} + a_{\max} + \epsilon] \tag{3.22}$$

For $\|s\| > \epsilon > 0$,

$$\begin{aligned}\dot{V} &\leq \|s\|\left[k_1\|e_2\| + \|\ddot{x}_{i+1}\|\right] - s(\text{sgn}(s)\left[2k_1 v_{\max} + a_{\max} + \epsilon\right]) \\ &= \|s\|[k1(\|e_2\| - 2v_{\max}) + (\|\ddot{x}_{i+1}\| - a_{\max}) - \epsilon] \\ &\leq -\|s\|\epsilon\end{aligned} \tag{3.23}$$

Hence choosing ϵ will give an ultimate bound on the error (Thm. 14.1, [38]). 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 $e_2 = 0$, the quantity ϵ is chosen such that

$$\begin{aligned}\|s\| &> \epsilon \\ k_1\|e_1\| + \|e_2\| &> \epsilon \\ k_1(\sigma_{\text{ref}} - l) &> \epsilon\end{aligned} \tag{3.24}$$

This is a controller with two parameters (k_1, ϵ) that has 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 (3.22) and (3.8) appropriately will give the full controller. Also, n does not appear in any of these expressions. If the controller is applied as is (in only one direction), the error bounds will essentially be independent of the number of vehicles.

3.4.3 Unified Attack Controller

The front and rear controllers are combined in the graph theoretic manner presented in [32]. Let $G = (V, E)$ be the directed graph representing the interconnectivity of the

[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.

system. V is the set of nodes (same as the number cars) present in the graph and E the set of directed edges. An edge $(i, j) \in E$ (drawn from j to i in the 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:

$$A_{\text{adj}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{3.25}$$

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

$$\begin{aligned} u_i &= \sum_{j=1}^{n} a_{ij} u_{i,j} \\ &= u_{i,r} + u_{i,f} \end{aligned} \tag{3.26}$$

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. 3.3. If there is consensus in the system, all the cars

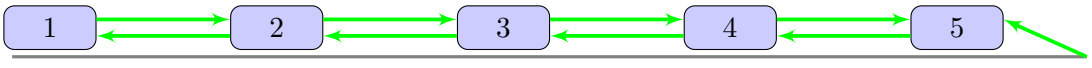


Fig. 3.3: Interaction of a 5-Member Platoon. The Leader also Follows a Reference. The Arrows Denote Information Flow; an Arrow From 3 to 4 Means 4 Senses some Information About 3, for Example Relative Distance.

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.

3.4.4 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

$$A_{\text{adj}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.27}$$

and this scheme is shown in Fig. 3.4. 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. The 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 [16]. Such filters, which

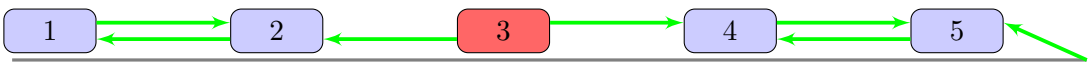


Fig. 3.4: Interaction of a Five Member Platoon with Attacker at Position 3. Note that Attacker is Assumed to be Indifferent.

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 will be demonstrated, 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. 3.5): 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

Fig. 3.5: Decision Rule for Adjusting Adjacency Matrix. Each Car Adjusts only its own Row, Based on Local Information.

If implemented correctly, the system should obtain the structure outlined in Fig. 3.6 and the following adjacency matrix

$$A_{\text{adj}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.28}$$

In this setting the ultimate boundedness analysis from the previous section 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 at best 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, this 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.

3.4.5 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 3.4.2. The error coordinates are

[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.

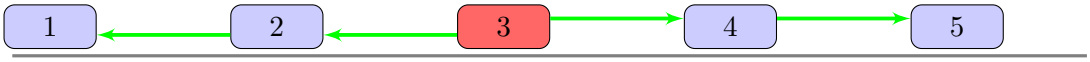


Fig. 3.6: Interaction of a 5-Member Platoon with Attacker at Position 3 with Adjacency Adjusted.

then passed through a squaring function with gains l_1, l_2 and then low pass filtered. The time and frequency domain representations are given as

$$r(t) = h_{\text{lp}}(t) * e(t)^T \begin{bmatrix} l_1 & 0 \\ 0 & l_2 \end{bmatrix} e(t) \tag{3.29}$$

where $h_{\text{lp}}(t)$ is the impulse response of the low pass filter and for the simulation presented, 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 loses detection of 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 it is known that errors propagate in interconnected platoons [22, 23, 27, 28], 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. It is emphasized again that this constitutes an attack detection scheme, not one for identification.

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. The system is still completely decentralized.

3.5 Simulation and Results

To demonstrate the effectiveness of the proposed approach, a five vehicle platoon is considered 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}. The chosen 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 3.1. 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.

Table 3.1: 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

Simulation results using an attack of a single frequency and attacker power equal to that of normal vehicles (α_{att}) are shown in Fig. 3.7 with a sliding mode controller without attack detection. For the same parameters, a simulation was performed with attack detection and its results are shown in Fig. 3.8.

3.5.1 Evaluating Attack Efficacy

To calculate the effect of an attack, a damage state to the platoon is assigned along the lines of (3.1). This damage state starts with a value of zero and all the collisions' relative velocities are accumulated as the simulation progresses and cars collide.

A collision line is defined as follows: 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 $f_{\text{col}} = \frac{1}{2t_{\text{col}}}$

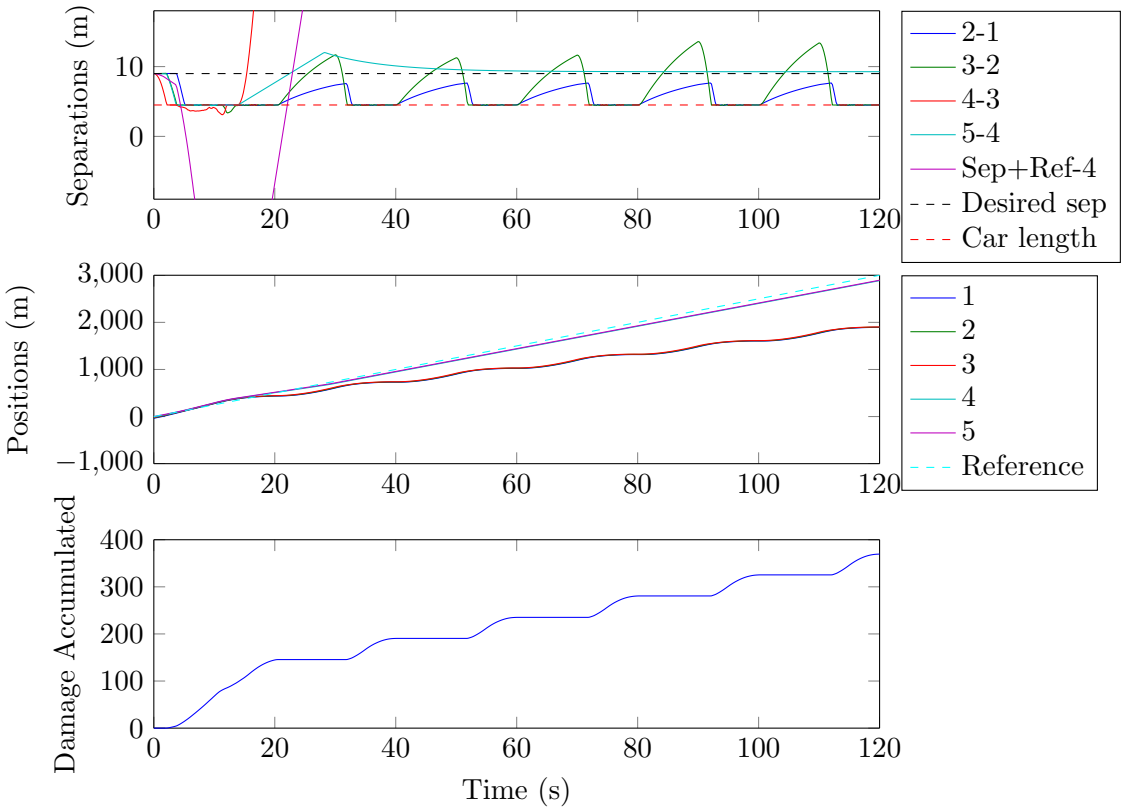


Fig. 3.7: Sliding Mode Controller Without Detection. Separations, Positions and Damage Data, Single Attacker at 3.

is a function of relative attacker power and initial conditions.

The value of f_{col} gives 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, collisions can be avoided if a suitable control scheme is adopted.

In other words, for Fig. 3.9, Fig. 3.10 and Fig. 3.11, 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.

3.5.2 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.

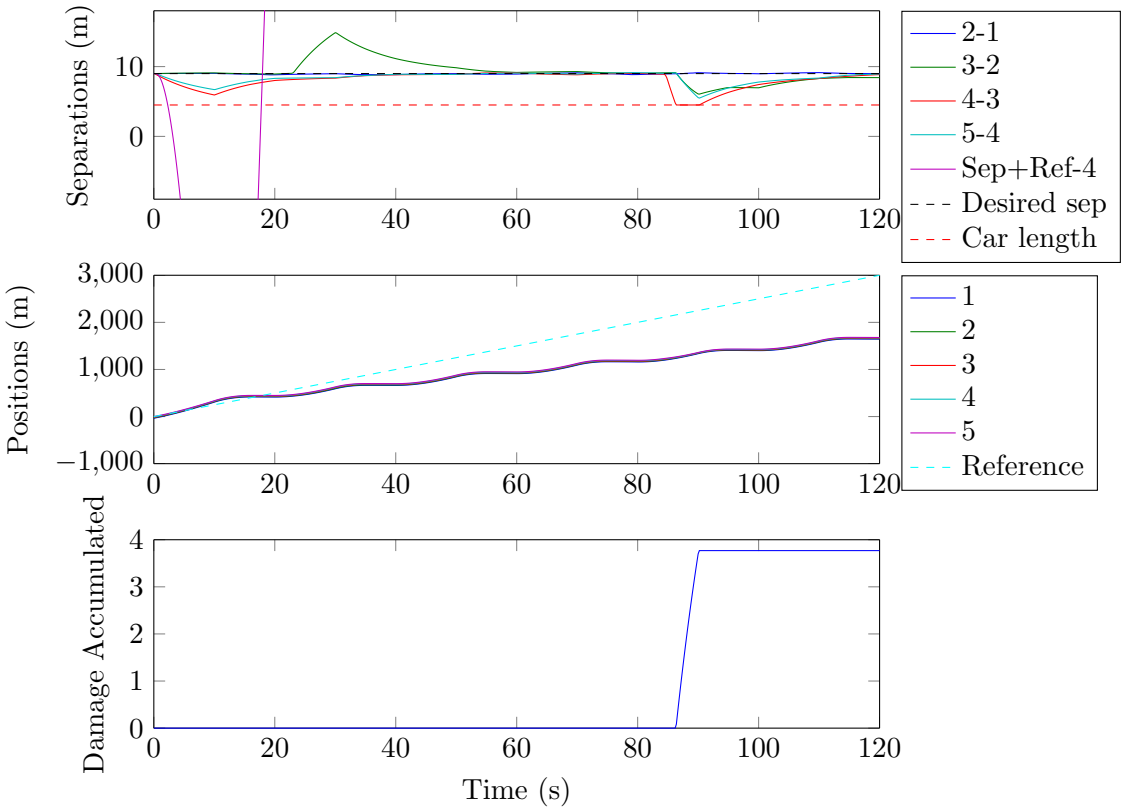


Fig. 3.8: Sliding Mode Controller with Detection. With Detection, There are a few Collisions Where the Filters Detect a False Negative and are not Quick Enough to Register the Attack Again.

From a comparison at a single frequency of attack (Fig. 3.7, Fig. 3.8), it is shown 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. 3.10, Fig. 3.11). The numbers on the y-axis correspond to the ratio of attacker power over normal vehicle power. For a reference, the total damage measurement using a linear bidirectional control law is also presented in Fig. 3.9. The high level controller for this was

$$u_i = k_p(x_{i+1} - x_i - \sigma_{\text{ref}}) + k_p(x_{i-1} - x_i + \sigma_{\text{ref}}) + \\ k_d(v_{i+1} - v_i) + k_d(v_{i-1} - v_i) \tag{3.30}$$

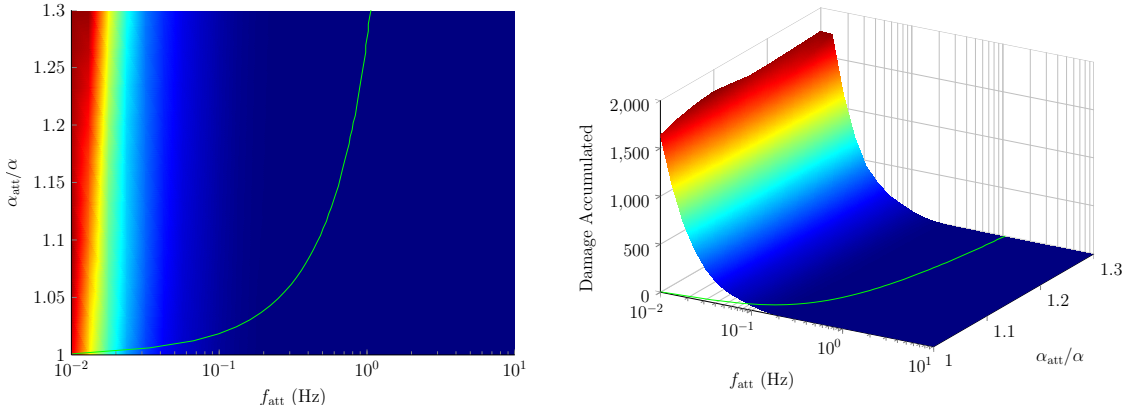


Fig. 3.9: Linear Controller Without Attack Detection. Total Damage Across Relative Attacker Power and Frequencies. Collision Line in Green.

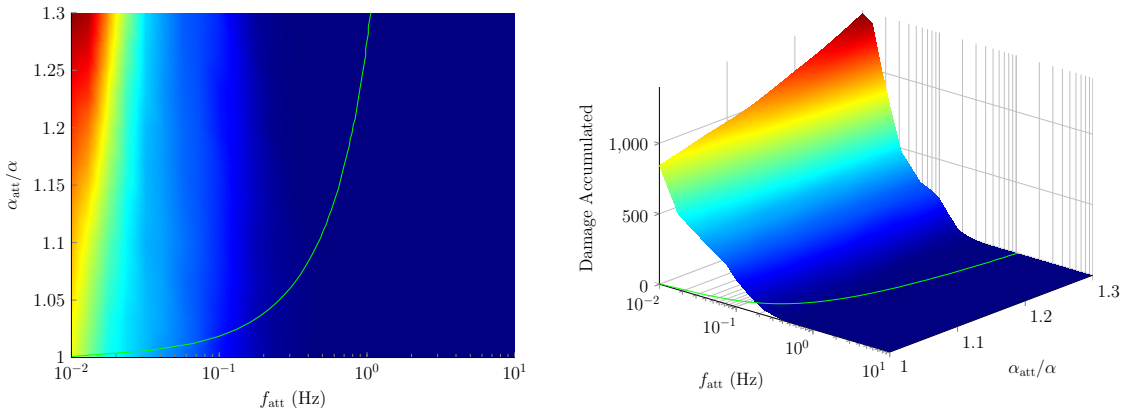


Fig. 3.10: Sliding Mode Controller Without Attack Detection. Outside the Collision line, There are Still Collisions, Especially when Attacker is as Strong as the Normal Vehicles.

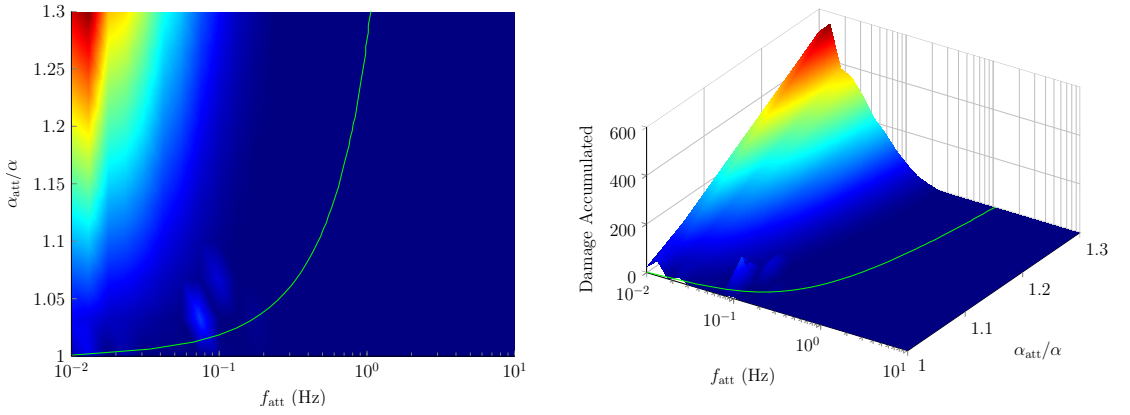


Fig. 3.11: Sliding Mode Controller with Attack Detection. Outside the Collision line, There is very Little Damage with Detection.

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 [1].

The collision line bounds a region in which whatever controller is designed, 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. It is observed that with detection, the number of collisions outside this region can be effectively eliminated, whereas damage is still seen with normal bidirectional control.

The little damage that occurs in the low frequency region with relative attacker power of 1 is small. It is surmised that the 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 the proposed controller.

Across this landscape, it is observed that damage is greatly reduced using attack detection in many cases, most notably 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.

3.6 Conclusion and Future Work

In this chapter, a sliding mode control scheme's effectiveness at stopping collisions in adversarial platooning environment was examined. 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. The approach was tested on a realistic model of a vehicle and presented the methodology for developing a controller based on this model. Through simulation results, it was observed that damage is greatly reduced when the proposed 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, a drastic change was not observed 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 sliding mode control 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.

CHAPTER 4
GAME THEORETIC ATTACK

4.1 Introduction

In this chapter, an attack on a string of automated vehicles, or platoons, is considered from a game-theoretic standpoint. Game theory enables asking the question of optimality in an adversarial environment; what is the optimal strategy that an attacker can use to disrupt the operation of automated vehicles, considering that the defenders are also optimally trying to maintain normal operation. A zero-sum game is formulated and optimal controllers for different game parameters are found. A platoon is then simulated and its closed loop stability is then evaluated in the presence of an optimal attack. It is shown that with the constraint of optimality, the attacker cannot significantly degrade the stability of a vehicle platoon in nominal cases.

It is a common theme in CPS work that a threat model is proposed and any reactionary measures such as detection or mitigation are then formulated around the characteristics of that attack. The game-theoretic approach provides an alternate setting more related to optimal control that can be used to find an attack that optimizes certain criteria in an adversarial setting. While optimal control tries to find a minimizing solution to a problem, game theory similarly involves finding a saddle-point or Nash Solution to a problem, which is, in a sense, optimal for both the players [49].

Game theory has been used in the field of control systems for some time now. There are classic problems in game theory such as pursuit-evasion that have been adapted to modern control problems [33]. The area has also been used in the area of system design and disturbance rejection [34, 35]. Game theory is also used in communications and designing data networks [36].

The main contribution of this chapter is that it applies methods from game theory to

the problem of vehicular platooning and illustrates the behavior of an attacker employing an optimal strategy (in a game-theoretic sense). The proposition is motivated that in a realistic setting where the attacker is also a real world actor, it is very hard to achieve an optimal solution in which the attacker is actually causing catastrophic collisions and/or destabilization.

This chapter is organized as follows. A brief introduction to certain concepts from game theory and a quick formulation of the zero-sum case is given in section 4.2. In section 4.3, a problem formulation specific to the platooning system is presented, with the necessary assumptions and simplifications elucidated. Section 4.4 and 4.5 provide some numerical and simulation results. Section 4.6 discusses these results and section 4.7 concludes this chapter.

4.1.1 Assumptions and Attacker Capabilities

In order to permit a real-world solution, all assumptions are stated prior to problem formulation. In section 4.3, an infinite-horizon zero-sum differential game is formulated , which naturally leads to static gains and requires full-state feedback. This can be accomplished through inter-vehicle communication. Each vehicle can apply an acceleration input.

The discussion is limited to one attacker. One of the imposed conditions on the method used to solve Riccati equations is that the resulting system be stable. This then leads to the restriction that the attacker can only degrade stability and not cause instability.

4.2 Game Theory Preliminaries

Game theory provides a setting in which two or more players influence their respective scores by choosing different strategies [50]. Naturally, there are constraints on how different strategies affect the scores of each player, which are referred to as rules of the game. These three items–score, strategies and rules–are found in every well-formulated game. There are other extensions of this idea such as mixed-strategy and stochastic games but for the purposes of this discussion, the above three ideas are sufficient.

There is also a distinction between static and dynamic games, which differ in the following sense: if a player is able to react after another player's decision as the game proceeds, the game is dynamic, otherwise it is static. Dynamic games have a sense of successive moves and thus a sense of progression. When extended to continuous-time systems, this naturally extends to the idea of time.

In most cases of game theory, the approach is centered on the Nash and saddle-point solutions [50] and the strategies which produce them. These solutions have the property where one player cannot surely gain anything by deviating from this strategy. Suppose two players are at a Nash equilibrium and have their respective scores. If one player decides to chance a move away from this point in order to gain some score, there exists a move (strategy) for the other player to make the first one lose more than his initial score. Hence it is in the best interests of the first player to stay at the Nash equilibrium.

Game theory applied to continuous-time control systems results in what are known as differential games [50]. In this case, the players have access to their specific control inputs (strategies) and the system dynamics form the constraints to the system (rules). When the question of control or optimality arises, naturally there is some notion of cost (scores).

Consider the system defined by the following state space description.

$$\dot{x} = f(t, x, u) \tag{4.1}$$

where $x \in \mathbb{R}^{rn}$ is the state vector, $u \in \mathbb{R}^{p}$ is the input vector. In the state vector, n is the number of agents, while r is the system order for each agent[1]. Usually the inputs are partitioned along respective players. Each player then chooses his or her control input from a space of allowable control inputs. Limiting the discussion to two players and considering

[1]This choice of state vector size is one which applies to the real system dealt with in the next section. It is presented now to avoid providing two conflicting formulations and avoiding confusion in vector sizes.

the following formulation,

$$\begin{aligned} J_1 &= \int_0^T L_1(x,u,t)dt + \Phi_1(x(T)) \\ J_2 &= \int_0^T L_2(x,u,t)dt + \Phi_2(x(T)) \\ \dot{x} &= f(t,x,u) \end{aligned} \tag{4.2}$$

where J_i is the total cost of player i, L_i is an instantaneous cost and Φ_i is a terminal cost at time T. If the vector u is partitioned into two vectors u_1 and u_2 (not the individual elements of u) each of which contains the control inputs for each player, the effects of these two inputs can be dealt with separately, that is they can be weighted in the cost functions separately. Both players want to minimize their respective costs J_1 and J_2. In the platooning case, all the defending cars can be thought of as one player with multiple different control inputs, and all the attackers can be thought of as the other player[2].

One very useful simplification made here is that the game is restricted to be zero sum. This means that the equality $J_1 + J_2 = 0$ always holds. Zero-sum games are a good starting point in many game-theoretic approaches since they imply that the players are completely non cooperative. This imposes some important restrictions on the formulation and implies certain properties of the solution itself. For example, all solutions to non-cooperative zero-sum games will be saddle points [50].

Another very useful class of games are linear quadratic games. This sort of a formulation is similar to the linear quadratic regulator (LQR) and the construction of its solution also follows a similar process [51]. The functions L_i are replaced with the quadratic forms

[2] "Teams" might be a better word here, but "player" is used throughout the literature.

and the system is restricted to be LTI as shown in (4.3).

$$\begin{aligned} J_1 &= \frac{1}{2}\int_0^T x^T Q_1 x + u_1^T R_{11} u_1 + u_2^T R_{12} u_2 dt + \frac{1}{2} x^T(T) S_T x(T) \\ J_2 &= \frac{1}{2}\int_0^T x^T Q_2 x + u_1^T R_{21} u_1 + u_2^T R_{22} u_2 dt - \frac{1}{2} x^T(T) S_T x(T) \\ Q_1 + Q_2 &= 0,\ R_{11} + R_{21} = 0,\ R_{12} + R_{22} = 0 \\ \dot{x} &= Ax + B_1 u_1 + B_2 u_2 \end{aligned} \tag{4.3}$$

It is interesting to note at this point that the two cost functions are opposing each other. This captures the noncooperation and is absolutely necessary, since if both players have their goals aligned, a saddle point solution will not exist [50]. Also note that the values of R_{12} and R_{21} can be chosen to weight the other player's control input.

Without going into the details of constructing the solution, which can be found in [50, 51], the optimal control inputs u_1^* and u_2^* are given by

$$\begin{aligned} u_1^* &= -R_{11}^{-1} B_1^T P x \\ u_2^* &= R_{22}^{-1} B_2^T P x \end{aligned} \tag{4.4}$$

where P is the solution to the following Ricatti Equation

$$\dot{P} = -PA - A^T P - Q_1 + P(B_1 R_{11}^{-1} B_1^T - B_2 R_{22}^{-1} B_2^T) P^T \tag{4.5}$$

One further simplification which applies in this case of platooning is that a terminal time of infinity (infinite horizon) is used and, consequently, the solution to (4.5) is evaluated by setting the left hand side to zero.

4.3 Problem Formulation

A platoon of n vehicles is shown in Fig. 4.1. The state vector is

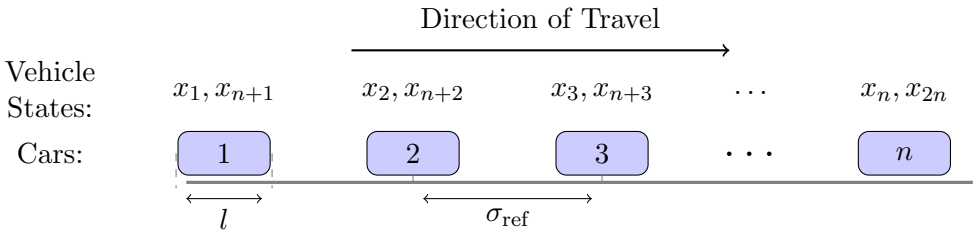


Fig. 4.1: A Platoon of n Vehicles. Each Car is l Meters Long and the Desired Separation from Center of one Car to that of the Other is σ_{ref}. Car n is the Leader.

$$\begin{aligned} x &= \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_{n+1} & x_{n+2} & \dots & x_{2n} \end{bmatrix}^T, \\ u &= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix}^T \end{aligned} \tag{4.6}$$

and the complete LTI system is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ A &= \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} \quad B = \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} \end{aligned} \tag{4.7}$$

where car i has position and velocity x_i, x_{n+i} respectively and control input u_i. These positions are measured from the center of mass of all the cars. Each car is essentially a double integrator in this setting. This choice is motivated by [48] where a split level control architecture is presented so that a higher level controller commands an acceleration and a lower level controller realizes it. A good portion of literature also assumes acceleration is commanded directly [26, 27].

In order to have the system track a reference and have the cars maintain a separation, a state vector z of error coordinates can be used as follows

$$\begin{aligned} z_1 &= x_2 - x_1 - \sigma_{\text{ref}} & z_{n+1} &= x_{n+2} - x_{n+1} \\ z_2 &= x_3 - x_2 - \sigma_{\text{ref}} & z_{n+2} &= x_{n+3} - x_{n+2} \\ &\vdots & &\vdots \\ z_n &= x_{\text{ref}} - x_n & z_{2n} &= \dot{x}_{\text{ref}} - x_{2n} \end{aligned} \tag{4.8}$$

where x_{ref}, $\dot{x}_{\text{ref}}$ are the position and velocity of a reference trajectory for the leader. Under this transformation, the A matrix remains the same, but the B matrix becomes

$$B' = \begin{bmatrix} & & 0_{n\times n} & & \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -1 \end{bmatrix} \tag{4.9}$$

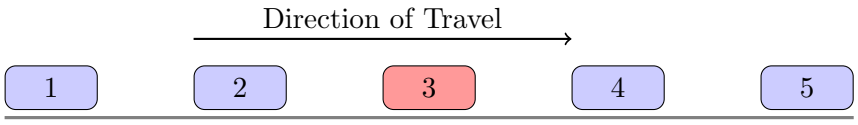


Fig. 4.2: A Platoon of 5 Vehicles with the Attacker at 3.

Considering the case where there is one attacker at position 3 in a 5 vehicle platoon (Fig. 4.2), the following parameters for the cost matrices are used for the game formulation.

$$\begin{aligned} Q_1 &= \text{diag}[1\ 1\ q_3\ 1\ 1\ 2\ 2\ q_8\ 2\ 2] \\ R_{11} &= \text{diag}[1\ 1\ 1\ 1] \\ R_{22} &= r_2 \end{aligned} \tag{4.10}$$

where q_3, q_8 and r_2 are parameters which will be varied in the next section. In Q_1 the

first half of the parameters weight position error and the second half velocity error. As mentioned before, the four cars except the attacker will constitute one player (the defender) and the single attacking car will be the other player.

Since the game is zero sum, it is not needed to define the other three cost matrices explicitly. Also, since the problem is formulated in error coordinates which the platoon directly has to minimize, diagonal cost matrices are a good starting point. The defenders want to minimize their own error states so their entries in the cost matrix are positive. In turn, the attacker would then want to maximize these error states. The variables q_3 and q_4 signify how much the attacker is willing to move in order to cause the other cars to move.

When solved with these values set, the result is two gain matrices (for the defenders and attackers) that use full state feedback for each control input. This was expected because of the choice of infinite time horizon. But even with these gains, the closed loop stability of the system can be evaluated by looking at the pole locations and how they change with varying parameters.

As mentioned before, some form of inter-vehicle communication will probably be required to realize full state feedback. Local sensing will probably prove to be insufficient. But readers might be interested in decentralized cooperative localization where individual agents estimate a full system state based on locally observed data and possibly intermittent global data. [52, 53] provide some solutions on whether a full state estimator is possible or not only using local sensing.

4.4 Game Parameters and Stability Margins

In Fig. 4.3, the attacker with half the control cost of the defenders ($r_{22} = 0.5$) i.e. the attacker's input is half as expensive as that of the other cars. This is equivalent to saying the attacker can apply double the control input of the regular cars. It is observed that by choosing just the negative of standard parameters ($q_3 = -1,\ q_8 = -2$), the first pole pair in the first plot is achieved, which is very well damped. In attempting to move more towards instability, q_8 changes sign before it gets close to the imaginary axis.

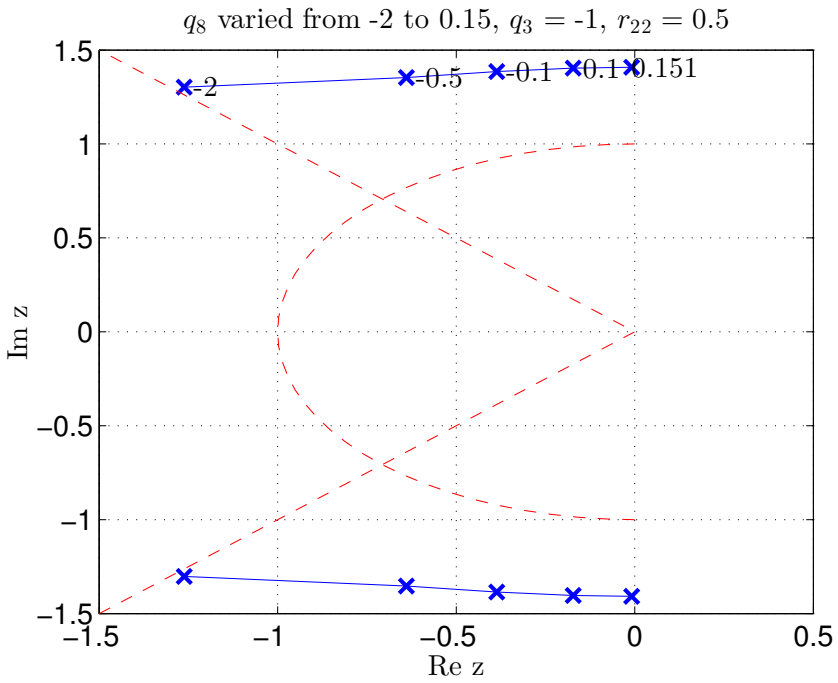


Fig. 4.3: Least-Damped Pole Locations Change Based on Parameter Varied. Unit Circle and 45° Lines Shown in Dashed Red. Parameters are Changed till the Least Stable Poles are Close to Being Unstable.

In Fig. 4.4, where q_3 is varied, q_8 needs to be zero in order to produce any instability at all. However, when it is set to zero, instability is quickly brought on.

Finally, Fig. 4.5 r_{22} is varied to make the attacker's control almost twenty times as cheap as the defenders'. Only then does instability ensue. Again this is an unrealistic scenario and it is noted that just buy applying more control input, instability is not easily achieved.

It is worth mentioning at this point that these parameters are not varied by the attacker. They are in fact means to set up a situation in which to find the optimal control law. Another way of saying this is that if these parameters are set and then the attacker tries to cause oscillations or collisions in the platoon by choosing some unstable gains, it will not be an optimal saddle-point solution. Hence the attacker will end up losing out in terms of the game itself.

The purpose of this section was to illustrate that a game would have to be set up

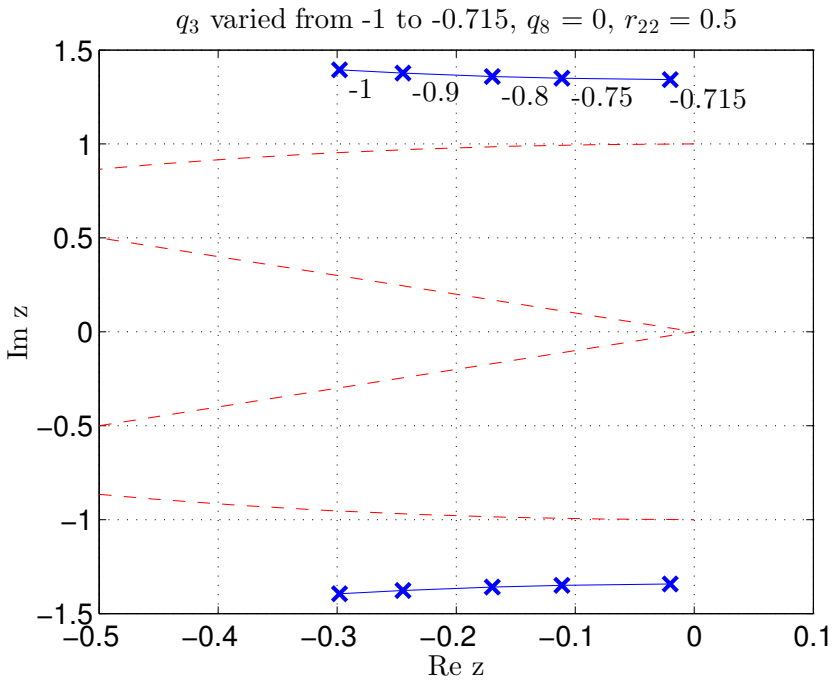


Fig. 4.4: Least-Damped Pole Locations Change Based on Parameter Varied. Unit Circle and 45° Lines Shown in Dashed Red. Parameters are Changed till the Least Stable Poles are Close to Being Unstable.

dramatically different from any usual setup to produce a situation where the attacker would benefit by causing instability. The following section shows two sets of simulation data which also highlight one more aspect of the pole locations–their frequency.

4.5 Simulation Results

Though it is assumed that acceleration can be commanded directly, a realistic model of the cars is still used in this section and the split level architecture proposed by [48] is employed. The low level controller realizes the acceleration command by compensating for the effects of drag and air resistance, but saturates whenever the car reaches its maximum power. This model lends itself naturally to a top speed $v_{\max}$ and a maximum standstill acceleration $a_{\max}$, which can be adjusted to match those of a real automobile.

The data in table 4.1 is used for the subsequent simulations. There are 5 cars in total and the attacker is at position 3.

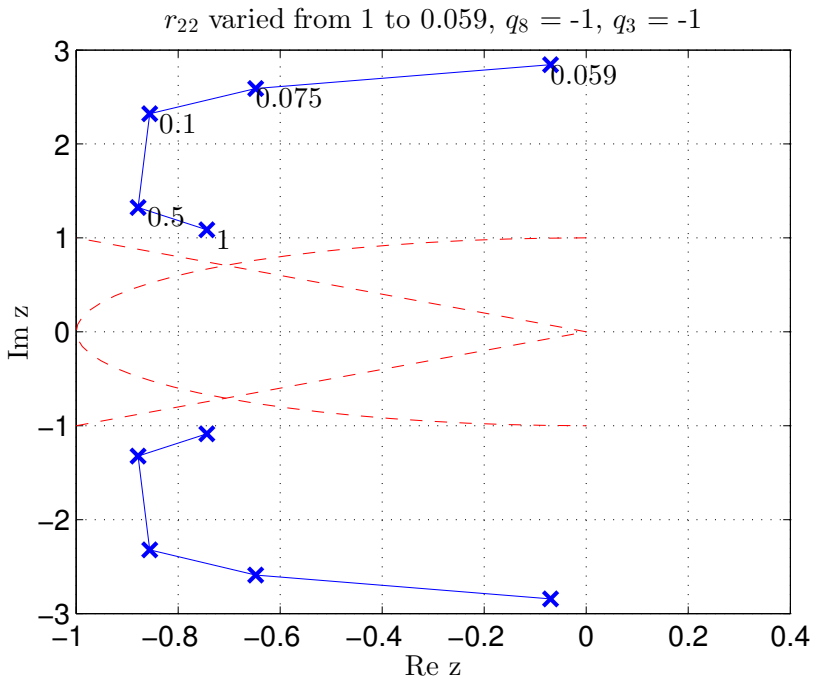


Fig. 4.5: Least-Damped Pole Locations Change Based on Parameter Varied. Unit Circle and 45° Lines Shown in Dashed Red. Parameters are Changed till the Least Stable Poles are Close to Being Unstable.

Table 4.1: Platooning Data used in Simulation.

Parameter	Value
σ_{ref}	9 m
l	4.5 m
$\dot{x}_{\text{ref}}$	25 $\text{m}\,\text{s}^{-1}$
$v_{\max}$	50 $\text{m}\,\text{s}^{-1}$
$a_{\max}$	5 $\text{m}\,\text{s}^{-2}$

Since the overshoot and damping properties needed to be visualized, the cars are started at their desired separations but with an initial velocity of zero.

In Fig. 4.6, it is observed that the nominal parameters used in the start produce a very stable solution as expected. The cars speed up from zero and catch up to the reference trajectory. At the start (since they get an instantaneous command to speed up to 25 $\text{m}\,\text{s}^{-1}$) the accelerations saturate and reduce as the speed increases. This is an effect of using a realistic car model.

In Fig. 4.7, it is seen that choosing the parameters as such does produce some oscillatory

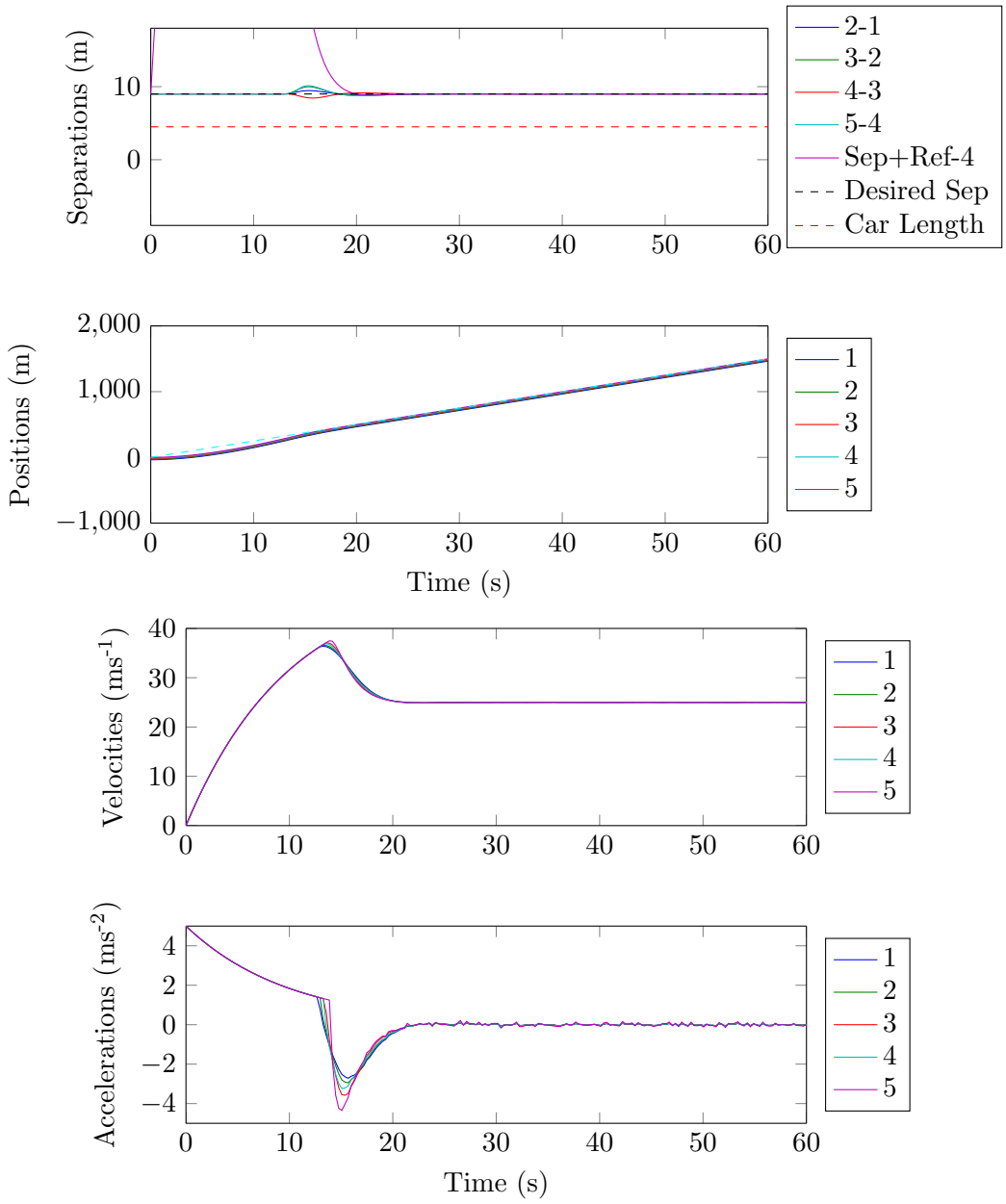


Fig. 4.6: Simulation Results of a Game Theoretic Solution ($q_3 = -1,\ q_8 = -2,\ r_{22} = 0.5$). These Parameters are Ones that Would be Realistically Set.

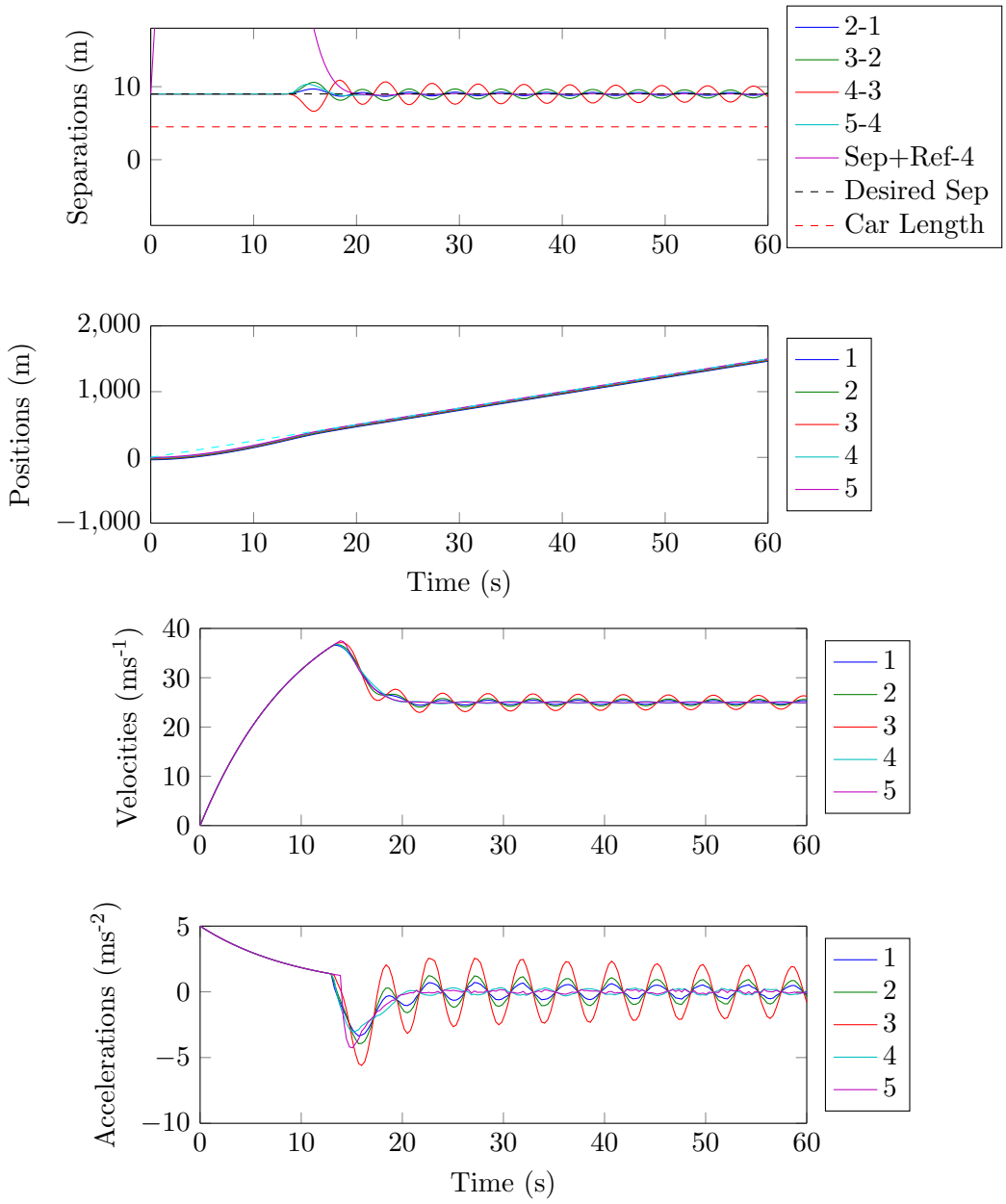


Fig. 4.7: Simulation Results of a Game Theoretic Solution ($q_3 = -1,\ q_8 = 0.151, r_{22} = 0.5$). These Parameters are Ones that Would Yield a Near Oscillatory Solution.

behavior. However, the amplitude is not nearly enough to cause any car to collide into another. The reason is simply that most physical systems (like automobiles) act as low pass filters and in order to produce a higher amplitude, a lower frequency is needed.

At this point, the reader is referred back to Figs. 4.3, 4.4 and 4.5 and it is pointed out that these poles are not at low frequencies. At least in all the cases shown, the poles do not move towards the area around the origin when varying parameters. This is another observed property that low-frequency, high-swinging solutions are not optimal in the game theoretic setups analyzed here.

4.6 Discussion

In this section, some of the key aspects of this game-theoretic framework are highlighted and also some limitations to the current approach of solving a Riccati equation are discussed.

It should be noted that forming a proper game-theoretic problem requires one or two conditions to be met. Just as in optimal control, the R_{ii} matrices that weight the player's own control inputs have to be positive definite. This means that it can never benefit a player to apply ever increasing control input. However the Q matrices as they have appeared in this chapter, can have a mix of nonnegative and negative eigenvalues. This aspect of the solution makes it interesting to ask the question: when do saddle-point solutions exist. [50] treats this question and derives some necessary and sufficient conditions, albeit with some limitations, for having a valid solution.

For the presented system, a single Riccati equation was solved which emerges in the zero-sum case. If nonzero-sum formulations were allowed, a system of Riccati equations would need to be solved simultaneously [51]. The computational tools used here were built primarily for the purposes of standard optimal control and thus only allowed unique stabilizing solutions to a single Riccati equation. As this equation can normally have more than one solution, but at maximum only one stabilizing one, it might be worthwhile, in order to entertain the unstable case, to ask which unstable solution actually corresponds to the saddle-point or Nash equilibrium point in a game-theoretic formulation [54].

Furthermore, this setup can be generally extended to a higher number of cars and

multiple attackers (who would still form the same single player). Those results might enable the sort of attacks that might have been more devastating in the traditional damage sense, but varying those parameters would add another dimension of complexity to an already complex system. It is humbly submitted that the results presented here are not entirely exhaustive and that the conclusions one might like to draw from these results might be overturned by a case where the attack does cause the desired behavior. However, these results are similar to those provided by optimal control theory and perhaps some general statements can be made on the nature of solutions provided by game theory in terms of system performance such as a guaranteed phase margin.

4.7 Conclusion

In this chapter, a game-theoretic framework was applied to the problem of vehicular platooning. An infinite time horizon, linear quadratic game was formulated whose saddle-point solution was found using methods from optimal control. The various solutions to this problem demonstrated that it is hard to achieve a setting (with the above qualities) that sees the attacker colliding with other cars or causing large oscillations in traffic flow while still fairing well in the game itself; it is usually not in the best interest of the attacker to disrupt regular operation. Further extensions of this work can involve solving the nonzero sum case and possibly entertaining unstable solutions. The effect of collusion by multiple attackers can also be considered in future work.

CHAPTER 5
EXPERIMENTAL VALIDATION

5.1 Introduction

This chapter presents the methodology and results from an experimental validation of the controllers presented in chapter 3. A brief discussion of the results is also presented.

5.2 Testbed Setup

The entirety of the following experiment was performed at the Robust Intelligent Sensing and Control (RISC) lab at Utah State University using the RISC Multi-Agent Analysis Platform (MAAP). This testbed uses the Cortex motion capture system integrated with the Robotics Operating System (ROS) [55].

This effort was greatly expedited by a handful of students who had previously performed their own experiments here. The integration of the Cortex visual capture system and ROS was accomplished mainly by Maughan [56]. Further integration and optimization of the testbed was performed by Mehrok and Manjunath [57, 58]. Erekson assembled the ground vehicles used in this experiment and used custom reflection marker templates to enable them to be uniquely identified by Cortex motion capture system [2].

The ground robots role was fulfilled by the Polulu m3pi robot shown in Fig. 5.1. These cars require a voltage level to apply to the left and right motors which form a differential drive system. Communications between the robots and the ground station computer were handled by Xbee Series 1 RF Modules, which was only used to transmit control commands to the robots. No on-board sensing was needed as the Cortex system supplied all position and velocity data required. The control algorithm was implemented in ROS using C++ on a single linux machine that served as a ground station. The system level architecture is shown in Fig. 5.2.

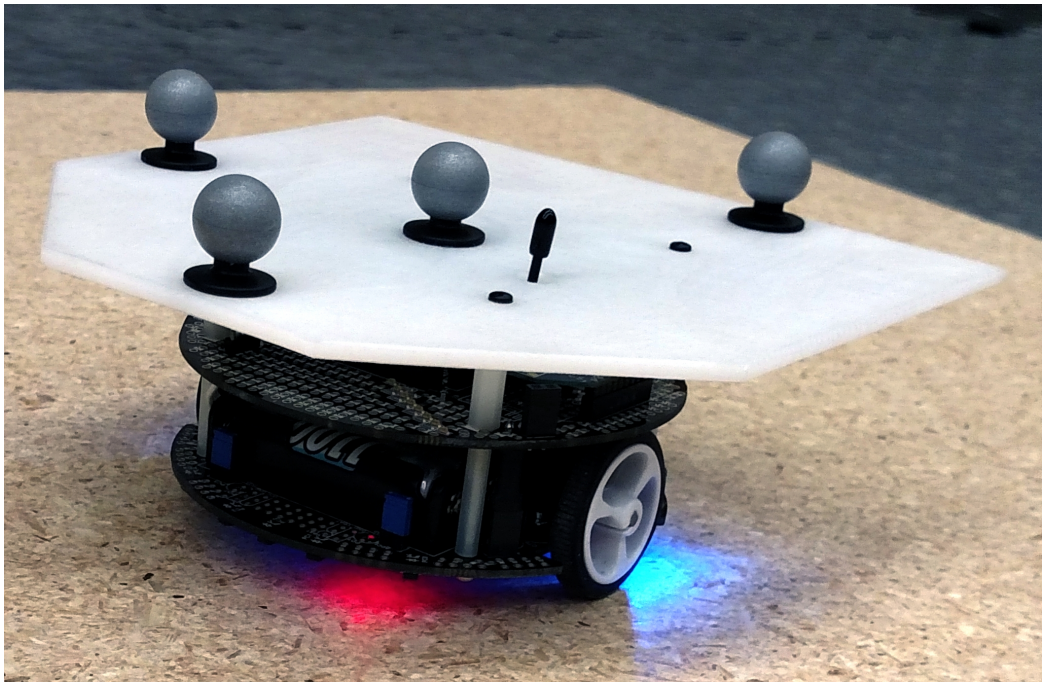


Fig. 5.1: The Polulu m3pi Robot with Custom Reflector Template [2].

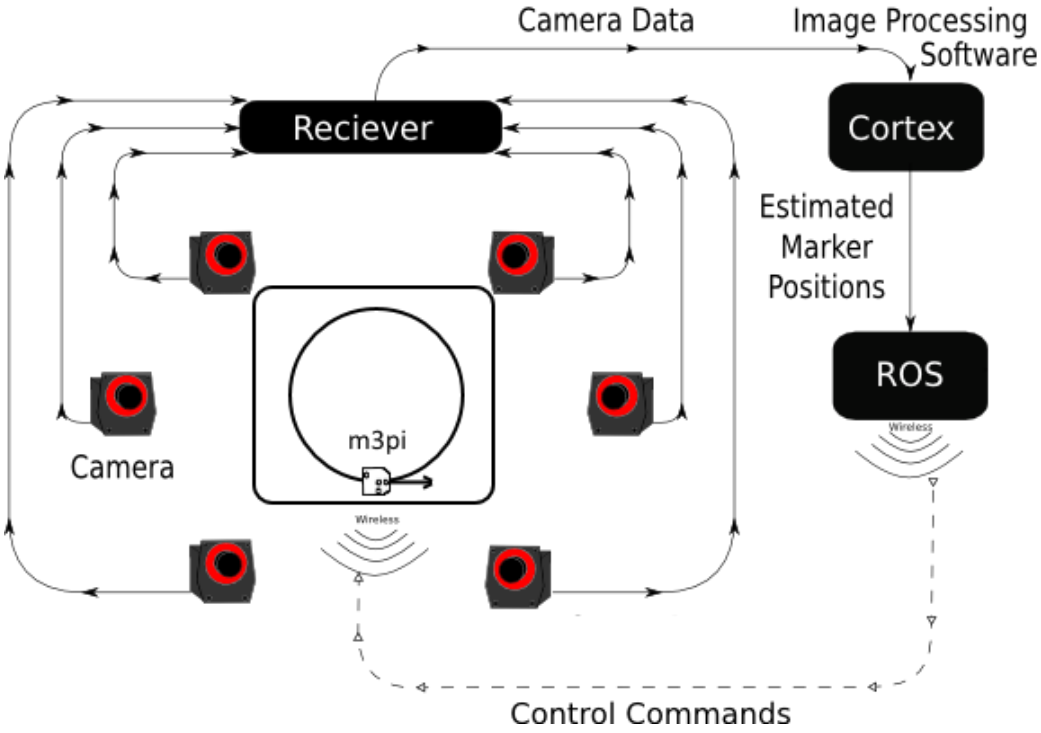


Fig. 5.2: USU's RISC MAAP System [2].

5.3 Experiment Parameters

In Table 5.1 are some parameters some of which had to be altered for this test setup. Two key changes had to be made in order to carry out this experiment in this testbed, both of which exhibit at most minor effect on the results. Firstly, the platoon had to follow a circular path as shown in Fig. 5.3 for which a lateral controller was required. This was required to be independent of the longitudinal controller and, in the end, was designed using a method similar to the differential flatness controller by Ferrin et al. [59]. Lastly, the weights of the bidirectional controller had to be modified to slightly favor front control under normal operation in a ratio of 70/30. This was done to decrease the initial overshoot of the cars, which would sometimes cause saturation.

Table 5.1: Platooning Data used in Experiment.

Type	Parameter	Value
Common	σ_{ref}	0.5 m
	l	0.15 m
	$\dot{x}_{\text{ref}}$	0.25 $\text{m}\,\text{s}^{-1}$
	r_{rad}	1.0 m
Lateral	k_r	0.22
	k_ψ	0.35
Attack	Car	2
	f_{att}	0.2 Hz
	amplitude	0.67 V
Linear Bidirectional	k_p	3
	k_d	7
Sliding Mode	k	0.5
	ϵ	0.75
	$v_{\max}$	1.5 $\text{m}\,\text{s}^{-1}$
	$a_{\max}$	2.0 $\text{m}\,\text{s}^{-2}$

Naturally, the distance and velocity calculations were made along the circular path, and thus the were calculated from raw motion capture data. The state-space model of one

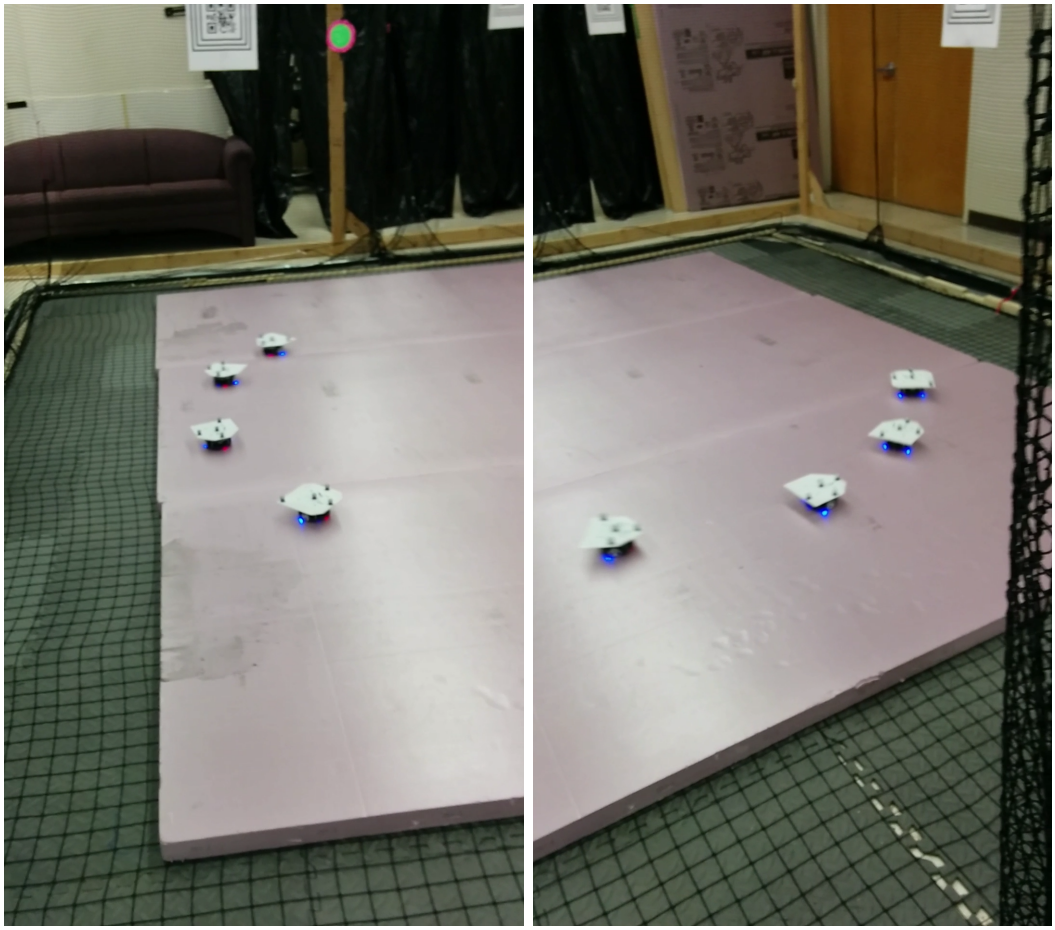


Fig. 5.3: Platoon Positions Along a Circular Path.

differential drive robot is as follows

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\psi} \\ \dot{v}_+ \\ \dot{v}_- \end{bmatrix} = \begin{bmatrix} v_+ \cos\psi \\ v_+ \sin\psi \\ \frac{v_-}{w} \\ a_+ v_+ \\ a_- v_- \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ b_+ & 0 \\ 0 & b_- \end{bmatrix} \begin{bmatrix} V_+ \\ V_- \end{bmatrix} \tag{5.1}$$

where

$p_x,\ p_y$ are the Cartesian position coordinates,

ψ is the heading measured from the x-axis counter clockwise,

v_+ is forward velocity,

v_- is the rotational velocity at the wheels.

V_+, V_- are the common-mode and differential voltages applied to the motors.

w is the distance between the wheels and

a's and b's are some motor constants.

The a_+ and b_+ variables were characterized for each robot, which turned out to be roughly -5 and 5 respectively. The differential motor constants were not found and the lateral controller was tuned so that this part of the system would converge to the path quickly. The lateral controller was

$$V_- = k_r(r_{\text{rad}} - r) + k_\psi(\psi_{\text{des}} - \psi) \tag{5.2}$$

where r are the distance of the car from the origin calculated from p_x, p_y and ψ_{des} is a desired heading based on the local direction of the path.

This allowed the use of the same inversion procedure as used in chapter 3

$$V_+ = \frac{1}{\hat{b}_+}(u - \hat{a}_+ v_+) \tag{5.3}$$

where the "$\hat{}$" denotes an estimate and u is the input to the second order system (acceleration).

This enabled the decoupling of the lateral and longitudinal parts of the control and allowed using the second-order acceleration based models used in the literature.

5.4 Results

The plots shown here are made to resemble the ones in preceding chapters, in order to facilitate comparisons. There are a total of four cars and car 4 is the leader, which follows a virtual target.

5.4.1 Linear Bidirectional Base Case

The linear bidirectional case is shown in Fig. 5.4, which does achieve all the platooning objectives.

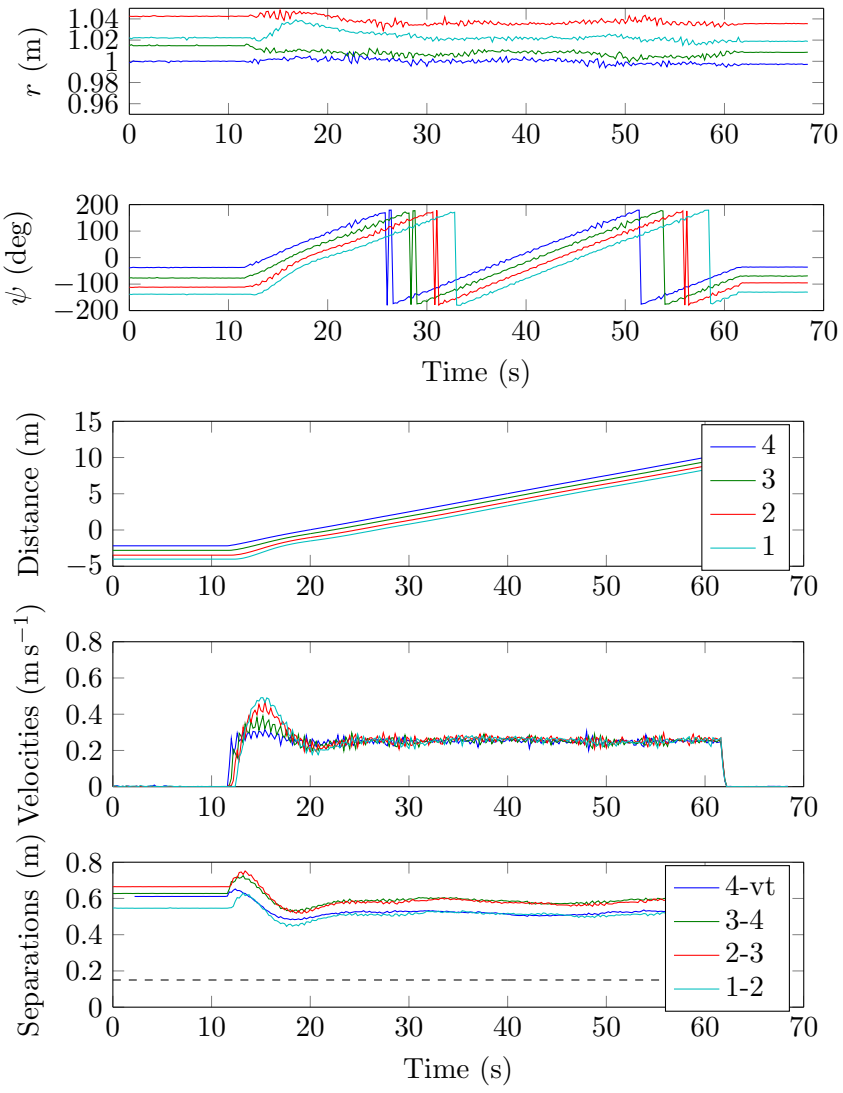


Fig. 5.4: Linear Bidirectional Base Case. Last Three Axes Represent Platooning Data. First Two are Raw State Variables.

5.4.2 Linear Bidirectional Under Attack

Here, the linear bidirectional case is shown under attack in Fig. 5.5. The attacker is able to split the platoon. The Lateral controller here keeps the error in r under $\pm$10cm.

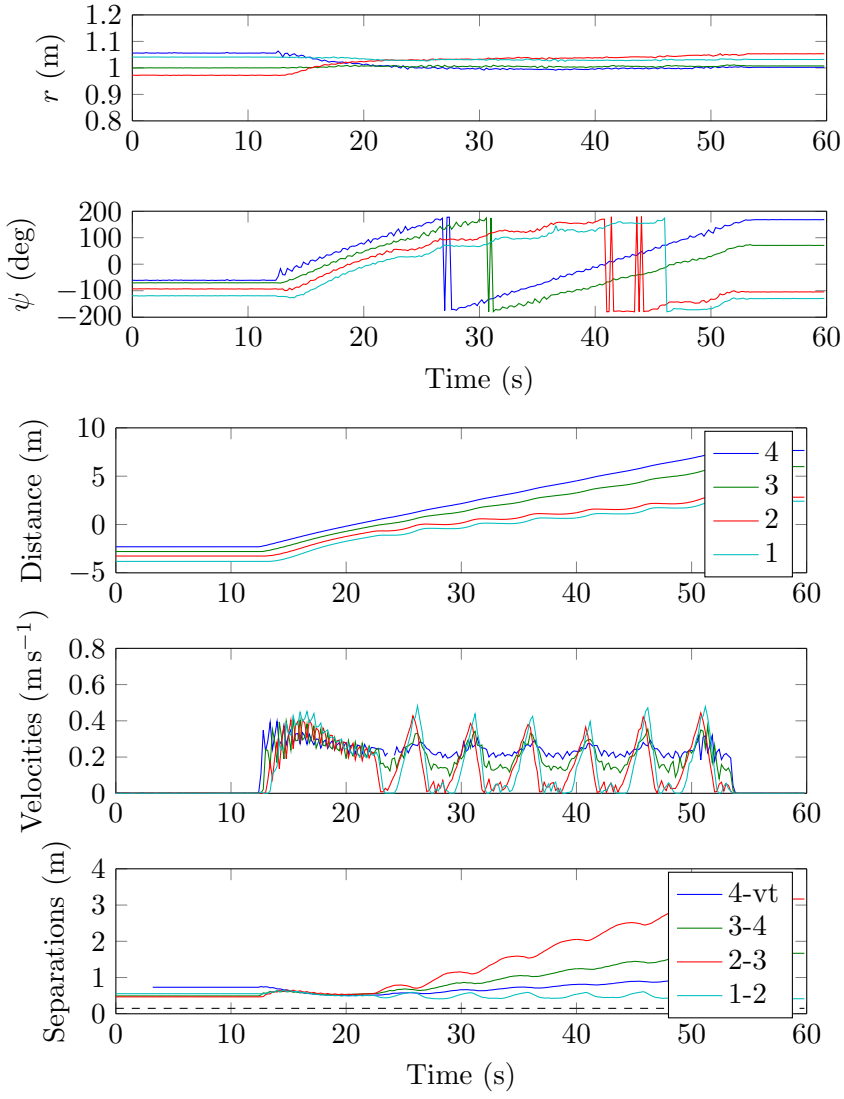


Fig. 5.5: Linear Bidirectional Under Attack. Last Three Axes Represent Platooning Data. First Two are Raw State Variables.

5.4.3 Sliding Mode Control Under Attack

The sliding mode controller in Fig. 5.6 does not incorporate any attack detection. There are even a few collisions in this case. The lateral controller still works well.

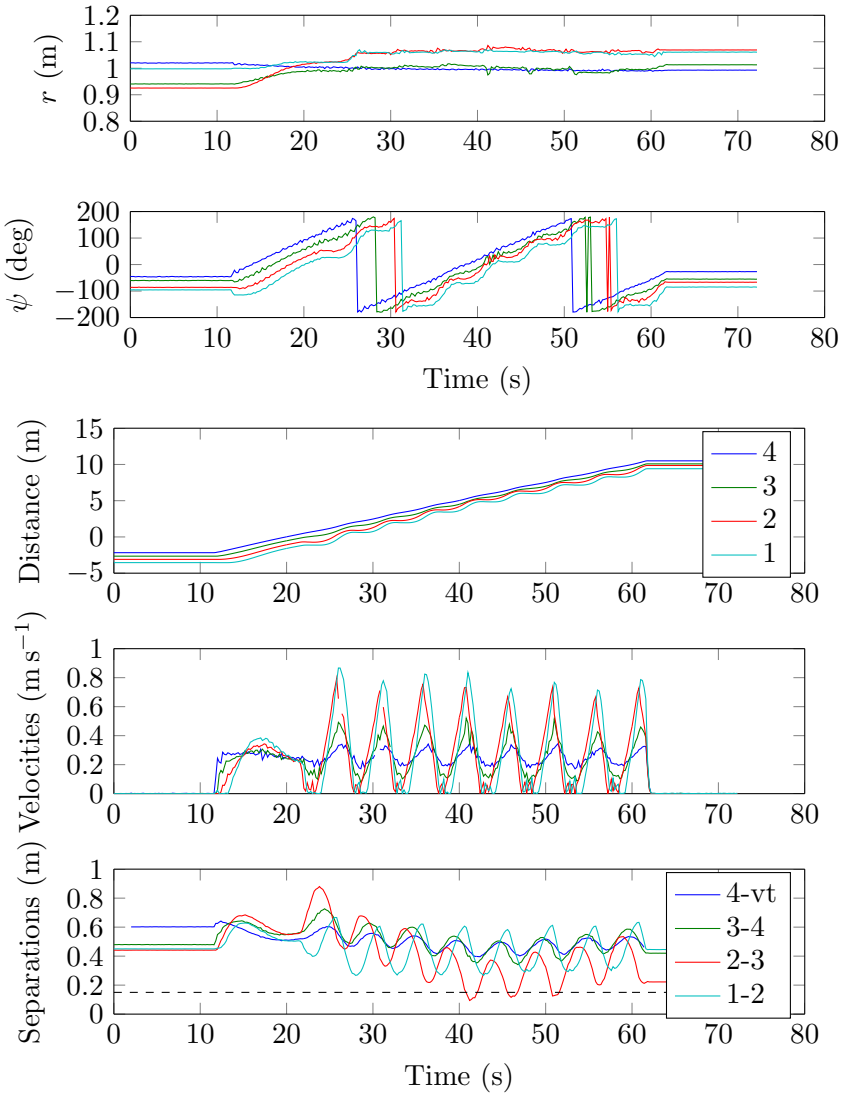


Fig. 5.6: Sliding Mode Under Attack, with no Detection. Last Three Axes Represent Platooning Data. First Two are Raw State Variables.

5.4.4 Sliding Mode Control with Attack Detection

The attack is properly detected and the control is adjusted accordingly for controller

in Fig. 5.7. As a result, and as expected, there are no collisions and separations are rightly maintained, even under attack. The blue separation trace is just the distance between the leader and the virtual target, which means the first car had to leave the virtual target behind to save itself from a collision.

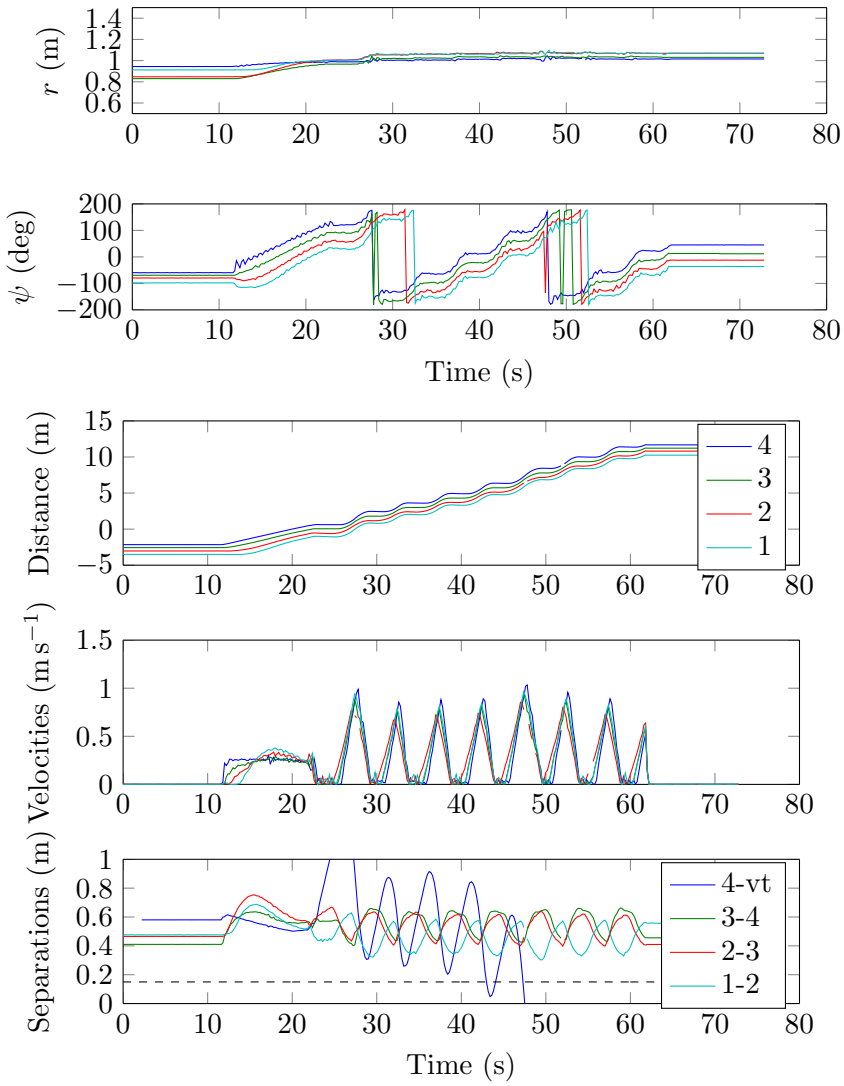


Fig. 5.7: Sliding Mode Under Attack but with Perfect Detection. Last Three Axes Represent Platooning Data. First Two are Raw State Variables.

CHAPTER 6
CONCLUSION

6.1 Summary of Results

The results of attacker capability from chapter 2 reinforce the assertion that, even in a fully controllable system, there can be some linear directions in the state space which are more reachable than others. This is especially true for a system like a platoon of cars where the usual controllability matrix is full rank but is ill conditioned.

It was proposed that the singular values and the corresponding left eigenvectors would also exhibit this trend. With the discrete-time equivalent problem, it was possible to express a necessary and sufficient control input as the minimizer of a convex optimization problem. Numerically solving this problem using the CVX solver, it was shown that the singular values do in fact indicate the different directions in which the reachable set extends.

Then using the properties of singular values and the fact that the control input derived was a minimizing function, the effect of final time on the reachability could be visualized. It was observed that increasing the final value beyond a point corresponds to a an ever smaller increase in the reachable set.

Thus it was established that there are some configurations that would be too hard for an attacker to achieve. As such, if an attack is to be mounted, it might be more devastating in terms of increasing a functional over time rather than driving the platoon to a final state.

Based on this alternative goal, the new damage metric described by (3.1) was established. In fact, it has been shown that the severity of a single collision can be greatly increased if an attacker injects an oscillatory disturbance [1].

The bidirectional platoon was chosen to serve as the basis for a mitigation strategy [26]. One of the beneficial and salient features of this architecture is that it does not require V2V or V2I communication; only local sensing is needed. It would be extremely desirable to

avoid adding a layer of complexity (another attackable system) such as data exchange or global localization and still be able to provide a robust countermeasure to this sort of an attack.

The sliding mode controller was chosen for its guarantee of stability and error boundedness in the presence of an uncertain but bounded disturbance input. When coupled with an attack detection scheme, this controller has been shown to achieve these characteristics. It was shown that in any platooning scenario, a single car does not have full control over all the states it can observe. Based on this, it can be decided what direction in the state space is to be stabilized by a vehicle, and what directions can be left for other members of the platoon. It was decided to prioritize those directions which an attacker would affect first, that is, if a malicious car is somewhere downstream, emphasize that direction and vice versa.

The specific detection scheme employed for this work is not the only one that can be used; as long as it adjusts the adjacency matrix to the one in (3.28), the error boundedness properties for the sliding mode controller can be actualized completely for half of the state directions around a car. The other states are left for the other cars to handle. Simulation Results show that this approach successfully mitigates most avoidable collisions and ameliorates the unavoidable ones.

This controller was tested experimentally using USU's RISC MAAP system. The results seem to follow what was expected from simulation. While there were collisions under normal operation during an attack, the proper adjacency matrix adjustment ensures that the cars are doing whatever they can to avoid coming together.

Apart from these attacks and controllers, the question of optimality was proposed in a game-theoretic sense. The problem statement asks if there exists an equilibrium between the severity of an attack and the defensive control that normal cars apply. A linear-quadratic game with different parameters was used to formulate a solution which was optimal for both the attackers and defenders.

Such a game would yield an optimal (saddle-point or Nash) solution which, if any

player deviated from, would result in that particular member losing out. It was found that under most game formulations, the attacker does not cause collisions or instability.

6.2 Conclusions

This thesis extends the domain of knowledge regarding autonomous cars before their imminent arrival on the streets. With an objective and mathematical idea of what a malicious actor can and cannot do, controllers can be designed with these capabilities in mind that ensure a list of priorities in which passenger safety is at the very top.

Based on the analysis provided, a possible countermeasure is presented which lends priority to passenger life and avoiding damage rather than the usual platooning goals. It is hoped that this countermeasure and the game-theoretic approach will serve to make platoons safer and more secure.

6.3 Future Work

In all of the cases above, there was no mention of string stability which is essential to understanding platooning. Further work in this area may incorporate how these attacks are mitigated and how their effect is attenuated up and downstream.

In most of the cases examined in this thesis, only a single attacker is assumed. It would be interesting to know the effect that multiple attackers can have on platoons of these sorts– both on the consensus and controllability of the entire system.

The game-theoretic controller could be extended beyond linear quadratic games. Game theory provides an alternate view on the attack and defend scenario. It would most likely be beneficial to see exactly what effect increasing the range of possible attacks can have on platooning. Also, the question whether or not a game exists in which the attacker always causes damage and wins is worth investigating based on alternate game formulations.

An independent lateral controller was developed which required the heading angle and minimum distance to the desired path. It might be possible to develop such a controller for real-world platooning that emulates a path by extrapolating data from the measurements of cars around itself and still retain the decentralized architecture.

REFERENCES

[1] 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*. ACM, 2015, pp. 167–178.

[2] I. T. Erekson, "Modified trajectory shaping guidance for autonomous path following control of platooning ground vehicles," 2016, all Graduate Theses and Dissertations. Paper 4919. [Online]. Available: http://digitalcommons.usu.edu/etd/4919

[3] W. F. Powers and P. R. Nicastri, "Automotive vehicle control challenges in the 21st century," *Control engineering practice*, vol. 8, no. 6, pp. 605–618, 2000.

[4] J. Sousanis, "World vehicte popuiation tops 1 billion units," 2011.

[5] D. L. Schrank and T. J. Lomax, *The 2007 urban mobility report*. Texas Transportation Institute, Texas A & M University, 2007.

[6] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "2015 urban mobility scorecard," 2015.

[7] "2015 motor vehicle crashes: Overview," 2015, [Online; accessed 17-November-2016]. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812318

[8] S. E. Shladover, "The california path program of ivhs research and its approach to vehicle-highway automation," in *Intelligent Vehicles '92 Symposium., Proceedings of the*, Jun 1992, pp. 347–352.

[9] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 68–77, March 2016.

[10] L. Xiao and F. Gao, "Practical string stability of platoon of adaptive cruise control vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1184–1194, Dec 2011.

[11] "The sartre project," 2002, [Online; accessed 15-June-2015]. [Online]. Available: www.sartre-project.net

[12] 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*, vol. 1, 2010, p. 12.

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

[14] 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*, June 2014, pp. 1061–1068.

[15] 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*. ACM, 2013, pp. 99–108.

[16] 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*, Dec 2010, pp. 5961–5966.

[17] 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*. IEEE, 2010, pp. 5967–5972.

[18] Y. Chen, S. Kar, and J. M. Moura, "Cyber physical attacks constrained by control objectives," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 1185–1190.

[19] D. Grimsman, V. Chetty, N. Woodbury, E. Vaziripour, S. Roy, D. Zappala, and S. Warnick, "A case study of a systematic attack design method for critical infrastructure cyber-physical systems," in *2016 American Control Conference (ACC)*, July 2016, pp. 296–301.

[20] B. Ramasubramanian, M. Rajan, and M. G. Chandra, "Structural resilience of cyber-physical systems under attack," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 283–289.

[21] D. D. Dunn, "Attacker-induced traffic flow instability in a stream of automated vehicles," 2015, all Graduate Theses and Dissertations. Paper 4455. [Online]. Available: http://digitalcommons.usu.edu/etd/4455

[22] D. Swaroop and J. Hedrick, "String stability of interconnected systems," *Automatic Control, IEEE Transactions on*, vol. 41, no. 3, pp. 349–357, Mar 1996.

[23] L. Peppard, "String stability of relative-motion pid vehicle control systems," *Automatic Control, IEEE Transactions on*, vol. 19, no. 5, pp. 579–581, Oct 1974.

[24] R. Caudill and W. Garrard, "Vehicle-follower longitudinal control for automated transit vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 4, pp. 241–248, 1977.

[25] 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*, vol. 42, no. 4, pp. 546–554, Nov 1993.

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

[27] 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*, Dec 2005, pp. 4964–4969.

[28] 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*, vol. 4, Dec 2004, pp. 3780–3785 Vol.4.

[29] 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*, June 2006, pp. 262–267.

[30] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1465–1476, Sept 2004.

[31] H. Tanner, "On the controllability of nearest neighbor interconnections," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3, Dec 2004, pp. 2467–2472 Vol.3.

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

[33] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization.* Courier Corporation, 1999.

[34] T. Basar, "A dynamic games approach to controller design: disturbance rejection in discrete-time," *IEEE Transactions on Automatic Control*, vol. 36, no. 8, pp. 936–952, Aug 1991.

[35] C. J. Tomlin, J. Lygeros, and S. S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, July 2000.

[36] Z. Han, *Game theory in wireless and communication networks: theory, models, and applications.* Cambridge University Press, 2012.

[37] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes, "Attack mitigation in adversarial platooning using detection-based sliding mode control," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, ser. CPS-SPC '15. New York, NY, USA: ACM, 2015, pp. 43–53. [Online]. Available: http://doi.acm.org/10.1145/2808705.2808713

[38] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002. [Online]. Available: https://books.google.com/books?id=t_d1QgAACAAJ

[39] L. S. Pontryagin, *Mathematical theory of optimal processes.* CRC Press, 1987.

[40] R. Bellman, "Dynamic programming and lagrange multipliers," *Proceedings of the National Academy of Sciences*, vol. 42, no. 10, pp. 767–769, 1956.

[41] B. Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods*, ser. Dover Books on Engineering. Dover Publications, 2007. [Online]. Available: https://books.google.com/books?id=fW6TAwAAQBAJ

[42] E. Lavretsky and K. Wise, *Robust and Adaptive Control: With Aerospace Applications*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2012. [Online]. Available: https://books.google.com/books?id=a2128lhlWfQC

[43] L. C. Evans, "An introduction to mathematical optimal control theory," *Lecture Notes, University of California, Department of Mathematics, Berkeley*, 2005.

[44] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[45] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

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

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

[48] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer, 2011. [Online]. Available: https://books.google.com/books?id=eoy19aWAjBgC

[49] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951. [Online]. Available: http://www.jstor.org/stable/1969529

[50] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*. London, San Diego: Academic Press, 1995. [Online]. Available: http://opac.inria.fr/record=b1090171

[51] A. W. Starr and Y. C. Ho, "Nonzero-sum differential games," *Journal of Optimization Theory and Applications*, vol. 3, no. 3, pp. 184–206, 1969. [Online]. Available: http://dx.doi.org/10.1007/BF00929443

[52] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 1402–1409.

[53] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, 2012.

[54] H. K. Wimmer, "The algebraic riccati equation: Conditions for the existence and uniqueness of solutions," *Linear algebra and its applications*, vol. 58, pp. 441–452, 1984.

[55] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[56] D. S. Maughan, "Robust intelligent sensing and control multi agent analysis platform for research and education," 2016, all Graduate Theses and Dissertations. Paper 4965. [Online]. Available: http://digitalcommons.usu.edu/etd/4965

[57] P. S. Mehrok, "Quadrotor uav path following using trajectory shaping," 2016, all Graduate Theses and Dissertations. Paper 4997. [Online]. Available: http://digitalcommons.usu.edu/etd/499

[58] A. Manjunath, "Path following by a quadrotor using virtual target pursuit guidance," 2016, all Graduate Theses and Dissertations. Paper 4990. [Online]. Available: http://digitalcommons.usu.edu/etd/4990

[59] J. Ferrin, R. Leishman, R. Beard, and T. McLain, "Differential flatness based control of a rotorcraft for aggressive maneuvers," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 2688–2693.