AN EFFICIENCY-MOTIVATED ATTACK AGAINST VEHICLES IN A PLATOON: LOCAL VEHICLE CONTROL, PLATOON CONTROL STRATEGIES, AND DRIVE TRAIN TECHNOLOGIES CONSIDERATIONS

by

David A. Cornelio Sosa

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

of

MASTER OF SCIENCE

in

Electrical Engineering

UTAH STATE UNIVERSITY
Logan, Utah

2014

Abstract

An Efficiency-Motivated Attack Against Vehicles in a Platoon: Local Vehicle Control, Platoon Control Strategies, and Drive Train Technologies Considerations

by

David A. Cornelio Sosa, Master of Science

Utah State University, 2014

Major Professor: Dr. Ryan Gerdes
Department: Electrical and Computer Engineering

Vehicle platooning has been heavily studied the last decade. A transportation system formed by electric vehicles driven by control systems with the help of on-board sensors, wireless inter-vehicle communication, and wireless recharge capability has been shown to increase highway capacity, transportation safety, reduce travel time, save energy, and release human drivers from stress. Two layers of control are required to automate a platoon, the low-level vehicle control, and the upper-level platoon control which seeks to maintain the constant spacing of the platoon, and avoid collisions.

In order to have a robust platoon, the vehicle control system needs to be robust to gain variations. Simulations were run in Matlab's Simulink to compare how well a vehicle control system would behave in the presences of nonlinearities and disturbances. The integer order and fractional order controllers were designed with the same specifications. Fractional order controllers present better performance with no overshoot for the speed servo, and faster response for the steering system.

For platoon control, the necessity is to achieve string stability. The bi-directional and leader-follower architectures have been shown to achieve string stability. Still, what happens to all the benefits of platooning when a malicious vehicle (attacker) attempts to perturb the system? This malicious attack could be the result of a company trying to sabotage the operation of another's in order to make it spend more energy than required, and thus raise its transportation costs. By using Matlab, a simulation platform was designed. It was used to simulate the response of a robust platoon to an optimal attack profile, generated by Matlab's genetic algorithm. To calculate the energy expenditure a model for a 1995 Honda Accord LX from cappielo's analysis is used. Two scenarios are considered: 1) the attacker intends to make the whole platoon spend extra energy, and 2) the attacker focuses on affecting only one victim. The greatest amount of extra energy expenditure for the first scenario was obtained with the bi-directional architecture and a size 3 platoon (140%). The leader-follower architecture limited this peak value to 94% for a size 8 platoon. In order to really profit from the benefits of platooning, a platoon size 8 or more is recommended. In this desirable range, the bi-directional control law manages to limit the extra energy expenditure to 80% (size 8) to only 35% (size 20). For the leader-follower and a size 20 platoon, the optimal attack produced an extra 65% expenditure. For the second scenario, with the bi-directional architecture the attacker could make the victim spend up to 122% (size 10). Still, this depends on both the attacker's and the victim's position. For instance, with the attacker in position 2, only 8% extra energy was observed. The leader-follower architecture allowed between 80% to 110% in any position for the attacker while in front of the victim (the attacker cannot affect the victim from behind).

Regenerative braking in all cases saved between 35% to 50% of the energy that would be otherwise lost by the use of dissipative brakes.

In order to create an operational platoon system, that is as robust as possible to the attack, the recommended platoon size is 12 or more. The use of regenerative braking capable vehicles is a must. The control system should be the fastest possible, and make use of the bi-directional architecture to limit energy expenditure. The implementation of an attacker or defective vehicle detection system is recommend, taking the measure of making the attacker/defective vehicle reposition to the last in the platoon.

(67 pages)

Public Abstract

An Efficiency-Motivated Attack Against Vehicles in a Platoon: Local Vehicle Control, Platoon Control Strategies, and Drive Train Technologies Considerations

by

David A. Cornelio Sosa, Master of Science

Utah State University, 2014

Major Professor: Dr. Ryan Gerdes
Department: Electrical and Computer Engineering

The Automated Electric Transportation Group (AET) at Utah State University proposes to demonstrate the benefits and feasibility of vehicle platooning. A platoon is a transportation system, formed by vehicles that are controlled automatically by computers (the driver does not operate the vehicle while cruising). Platoons are interesting in view of the future because they present many advantages: increase highway capacity and safety, reduce travel time, save energy, and reduce stress for human drivers. There are many problems to solve in order to make vehicle platooning a reality. In this work, the focus is not on making it feasible, but rather the interest in placed on how, assuming platoons are a reality, a malicious party could exploit it to its favor. One example could be that of two shipping companies X and Y. Company X tries to sabotage the operation of company Y, in order to make it spend more fuel than required, and thus raise its transportation costs. By using computer software, a test platform was developed. It incorporates very well studied models of platoon control, and vehicle fuel consumption. The results from these simulations show that, in fact, such an attack could exist and make a platoon spend up to 140% extra fuel. Still, there are many recommendations that can be made to limit the effectiveness of such attack.

To my family, thanks for backing me up in the pursue of a dream, so far away from home.

Acknowledgments

I would like to thank Dr. Ryan Gerdes, Dr. YangQuan Chen, and Dr. Donald Cripps for providing motivation, guidance, and dedicating their time and support for this research along with the AET group that provided economical support through the DOE project (Grant 100859).

David A. Cornelio Sosa

Contents

List of Tables

List of Figures

Chapter 1
Introduction

Vehicle platooning has been receiving increasing attention, due to the great benefits it represents in terms of energy efficiency, reduced travel time, safety, and passenger comfort.

The investigations have focused mainly in achieving string stability, which is a requirement to guarantee that the overall system is stable, emergency braking scenario to guarantee safety of passengers, the separation distance achievable (to further reduce drag coefficient) and communications delay (to make feasible the leader-predecessor architecture). Not much has being studied about what happens to all this important benefits when a malicious vehicle is introduced in the platoon with the goal of compromising its operation. The reason for such an attacked could be one of two: 1) trying to break the string stability of the platoon to produce collisions, and 2) a corporative attack seeking to make one or more vehicles expend more energy than would be required in normal operation. Further more, since the platoon performance depends on how well the vehicles respond themselves, the possible advantages of using fractional order control design for vehicle control are analyzed.

Chapter 2
Background Information

Fractional-order control (FOC) is a field of control theory that uses the fractional-order integrator as part of the control system design toolkit. The fundamental advantage of FOC is that the fractional-order integrator weights history using a function that decays with a power-law tail. The fractional integral operator is different from any integer-order rational transfer function, in the sense that it is a non-local operator that possesses an infinite memory and takes into account the whole history of its input signal. Fractional-order control shows promise in many controlled environments that suffer from the classical problems of overshoot and resonance, as well as time diffuse applications such as thermal dissipation and chemical mixing.

It is advantageous to use fractional order control strategies when it is likely that integer order strategies will fail to satisfy the following three constraints, or design specifications, simultaneously [1].

1. Phase margin specification

$$Arg[G(jw_c)] = Arg[C(jw_c)P(jw_c)] = -\pi + \phi_m, \tag{2.1}$$

   where $G(s)$ and $C(s)$ are, respectively, the plant and controller transfer functions.

2. Robustness to variation in the gain of the plant

$$\left(\frac{d(Arg(C(jw)P(jw))}{dw}\right)\Bigg|_{w=w_c} = 0, \tag{2.2}$$

   which provides a flat frequency response around the gain crossover frequency, w_c. This ensures that over a given range, variations in the gain of the plant will not produce a change in the phase response, in addition to guaranteeing a nearly constant overshoot for the aforementioned range.

3. Gain crossover frequency specification

$$|G(jw_c)|_{dB} = |C(jw_c)P(jw_c)|_{dB} = 0. \tag{2.3}$$

This means that the gain of the system at the cross over frequency w_{gc} is 0 dB or unity.

2.1 Fractional Order Proportional Derivative (FOPD) Controller for a Second Order Plant

A vehicle with a steering system designed using the LEGO NXT kit was selected as our testing platform for demonstration porpuses.

The position servo used to control the steering system of the vehicle prototype can be modeled as a second order plant [2]. To correct for path deviations, a proportional derivate (PD) controller was chosen to control the plant so as to maximize the response of the motor [3].

The plant was identified using the transfer function

$$P_1(s) = \frac{k}{s(Ts+1)}, \tag{2.4}$$

where $k = 9.11$ degrees/%PWM and $T = 0.4$ s, while the PD controller's transfer is given by

$$C_1(s) = K_p(1 + K_d s^{\mu}), \tag{2.5}$$

where K_p and K_d are the proportional and derivative gains, respectively, used for tuning in both integer order and fractional order controllers. For integer order controllers, $\mu = 1$, while for fractional order controllers, μ is another tuning parameter with values in the interval $(0, 1)$.

The PD controller will be designed using the step response method [4]. The procedure used to find $K_d, K_p,\ and\ \mu$ that satisfy Equations (2.1)-(2.3) for the fractional order controller is the frequency response method (Bode plots) [3].

2.2 Fractional Order Proportional Integral (FOPI) Controller for a First Order Plant

The velocity servo used to control the speed of the prototype vehicle can be modeled as a first order plant [1]. A proportional integral (PI) controller was chosen to control this plant, as it ensures zero steady state error and a very fast response is not required [1]. Moreover, we will show how to compensate for the delay on the step response created by the dead-zone of the motor, with a PI controller .

The plant was identified using the transfer function,

$$P_2(s) = \frac{k}{Ts+1}, \tag{2.6}$$

where $k = 0.159$ rad/%PWM and $T = 0.125$ s, while the PI controller's transfer function is given by,

$$C_2(s) = K_p\left(1 + \frac{K_i}{s^{\mu}}\right), \tag{2.7}$$

where again, K_p and K_i are the proportional and integral gains, respectively, used for tuning in both integer order and fractional order controllers. μ is as defined for the PD controller.

The integer order PI controller was tuned using the step response method [4]. The procedure used to find K_i, K_p, and μ that satisfy Equations (2.1)-(2.3) for the fractional order controller is the frequency response method (Bode plots) [5].

Chapter 3
Fractional Order Control Design and Simulation-Based Comparison to Integer Order Performance

3.1 Simulation-Based Performance Comparison for Integer Order vs Fractional Order Controllers

The performance of integer order and fractional order controllers is compared for the steering and speed servos. Special attention is paid to the nonlinearities identified as dead zone, saturation, quantization, and delays in the closed loop.

3.1.1 Steering Servo: Integer Order PD vs Fractional Order PD

Integer Order Proportional Derivative (IOPD) Controller Design

The IOPD controller was designed using Matlab's Simulink design toolbox step response method to have the fastest possible response while keeping overshoot to under 10%. The transfer function of the controller, with filter, is given by,

$$C_1(s) = 4.74 + 1.3\frac{s}{0.019s+1}, \tag{3.1}$$

which yields a phase margin of $\phi_m = 60°$ and the cross over frequency of $w_{gc} = 29.8$ rad/sec (blue lines, Figure 3.1). By examining the Bode plot, we see that the controller fails to satisfy the robustness specification (Equation (2.2)) as the phase is not flat around w_{gc}.

Fractional Order Proportional Derivative (FOPD) Controller Design

Using the frequency response method [3], an FOPD controller was designed for the same ϕ_m and w_{gc} as the IOPD controller. The resulting FOPD controller's transfer function is of the form of Equation (2.5) where $K_p = 5.63$, $K_d = 0.6$, and $\mu = 0.7021$. As can be seen from Figure 3.1, the fractional order PD does satisfy Equation (2.2). It is important that this constraint can be met because a change in speed may produce a change in the gain of the steering system.

IOPD/FOPD Comparison

The behavior of the system with the PD controllers in terms of step response (Figure 3.2), gain variation effects for IOPD and FOPD (Figures 3.3 and 3.4), response to nonlinearities (Figure 3.5), and transport delay (Figure 3.6) is compared. From the figures, the FOPD controller exhibits a smaller overshoot with the same rise time. One might judge that IOPD performs better, but it is apparent from the response to gain variations that FOPD is a lot more robust, i.e. there is better tracking response to a 3.5 Hz sine wave in the presence of the nonlinearities identified in Section 3.1, and smaller tracking error with a transport delay of 10 ms in the feedback loop.

The FOPD outperforms the IOPD by keeping the tracking error smaller.

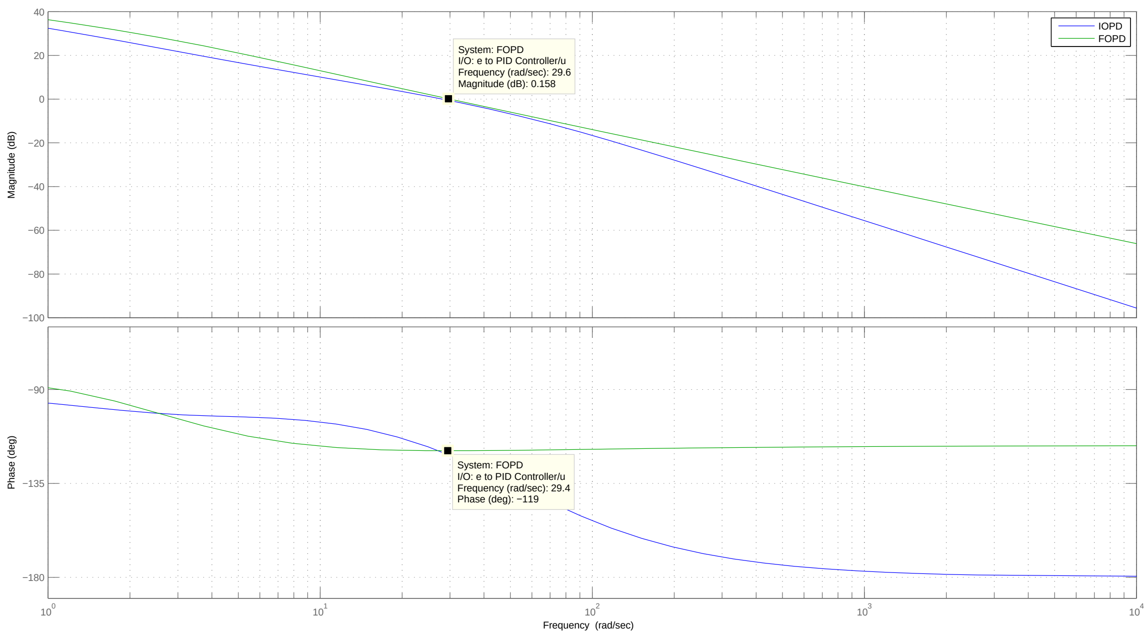


Fig. 3.1: Bode plot for integer order (blue) and fractional order (green) PDs.

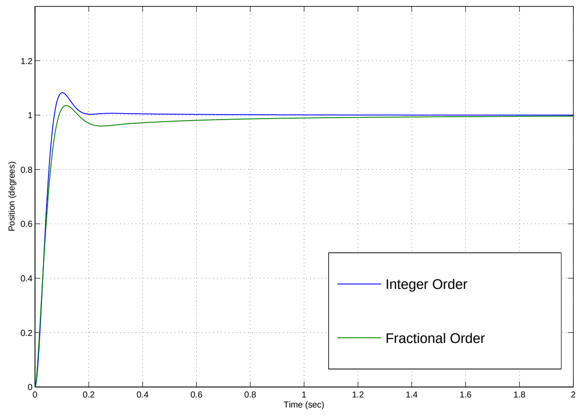


Fig. 3.2: Step response for IOPD vs FOPD.

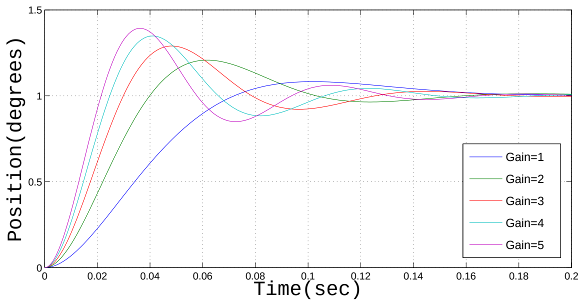


Fig. 3.3: Response to gain variations IOPD.

3.1.2 Speed Servo: Integer Order PI vs Fractional Order PI

Integer Order Proportional Integral (IOPI) Controller Design

An integer order PI controller was designed using Matlab's Simulink and the step response method for the velocity servo. The resulting controller transfer function is,

$$C_3(s) = 0.067 + \frac{34.447}{s}, \tag{3.2}$$

The controller is designed to keep a small overshoot, less than 10% and a reasonable rise time of about 350 milliseconds. Figure 3.7 shows the Bode plot of the system with the designed controller (blue line).

The designed control system has the specifications $\phi_m = 60^o$ and $w_{gc} = 4.72\ rad/sec$. The flat phase around w_{gc} is obviously not achieved by this design procedure.

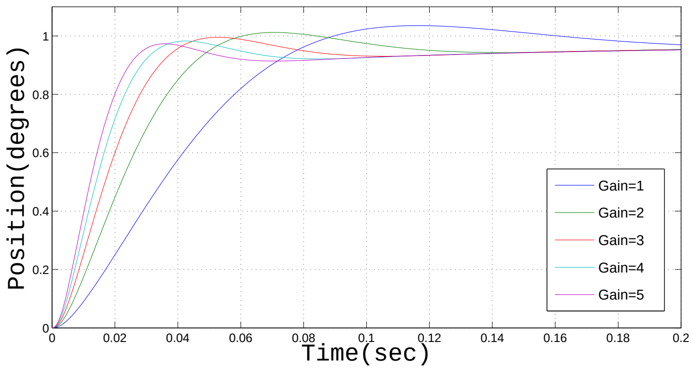


Fig. 3.4: Response to gain variations FOPD.

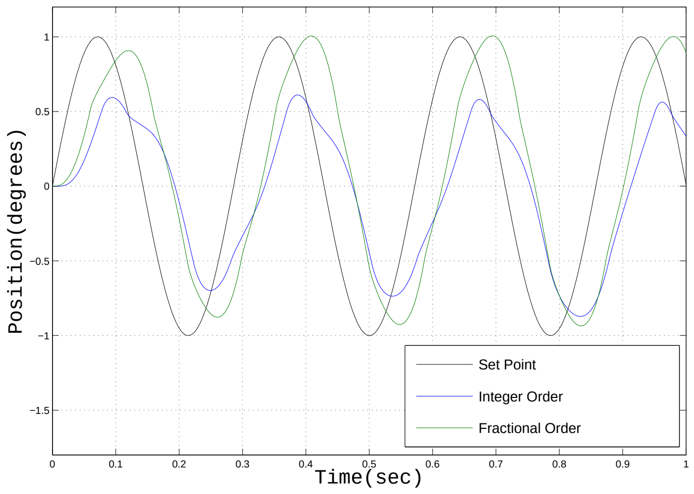


Fig. 3.5: Tracking response for IOPD vs FOPD.

Fractional Order Proportional Integer (FOPI) Controller Design

Again using the same ϕ_m and w_{gc} obtained for the IOPI, a FOPI is designed using the frequency response method [5]. The resulting controller's transfer function is of the form of Equation (2.7): with K_p=3.8, K_i=7.74 and $\mu = 0.995$.

Figure 3.7 shows the Bode plot for the system with the designed fractional order PI (green line). The phase response is better than expected. The plot is almost totally flat, the change in the phase response due to the pole is less than 0.5 degrees, so the overshoot is constant for a wide range of changes in the gain of the plant.

The obtained $\phi_m = 90^o$ in this case, which guarantees no overshoot, and the frequency is the same $w_{gc} = 4.72\ rad/sec$.

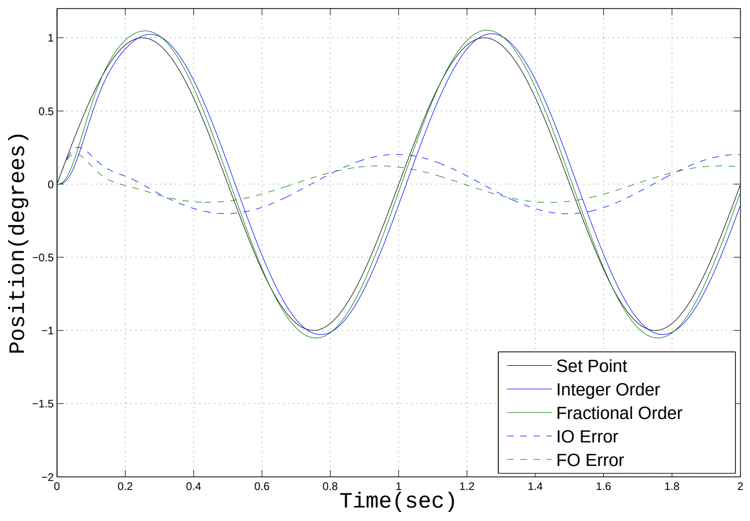


Fig. 3.6: Tracking a 1 Hz sine wave input with a transport delay in the feedback loop.

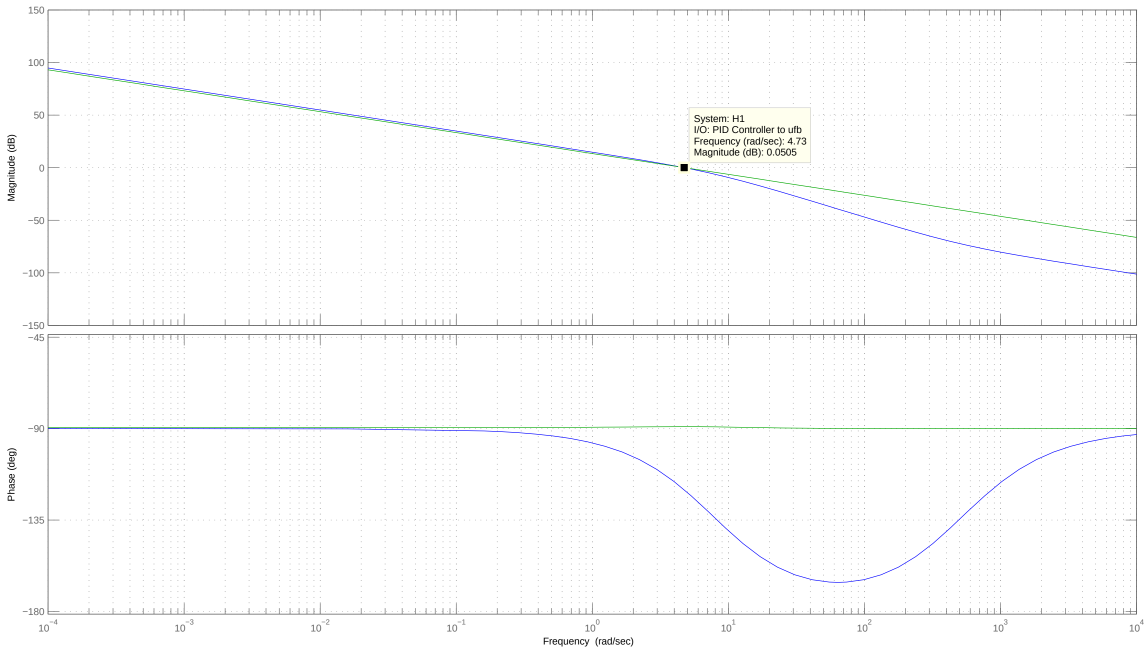


Fig. 3.7: Bode plot for integer order (blue) and fractional order (green) PIs.

Step Response, Gain Variations, Disturbance Rejection, and Nonlinearities Effect Comparison

The PI controllers are compared in terms of step response (Figure 3.8), response to gain variations for IOPI and FOPI (Figures 3.9 and 3.10), disturbance rejection (Figure 3.11), and the effect of the identified dead-zone (Figure 3.12). From the figures, the FOPI controller exhibits a response with no overshoot with about the same rise and settling times as the IOPI. The FOPI controller makes the system more robust to gain variations (keeps overshoot at zero as predicted from the Bode plot, while overshoot increases with gain for the IOPI), and behaves better in response to disturbances, i.e. keeps the magnitude of the effect smaller, and recovers without oscillation. The delay effect produced by the dead-zone is smaller on the FOPI (about half of what is for the IOPI). This delay effect could be reduced by increasing the controllers gains, still this changes the w_{gc}, and thus the frequency response for both controllers. A correction method for the dead-zone is introduced in the next section.

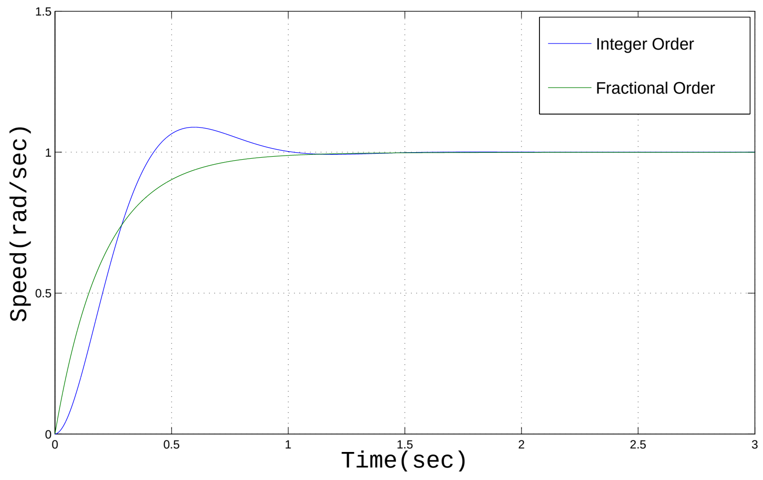


Fig. 3.8: Step response comparison IOPI vs FOPI.

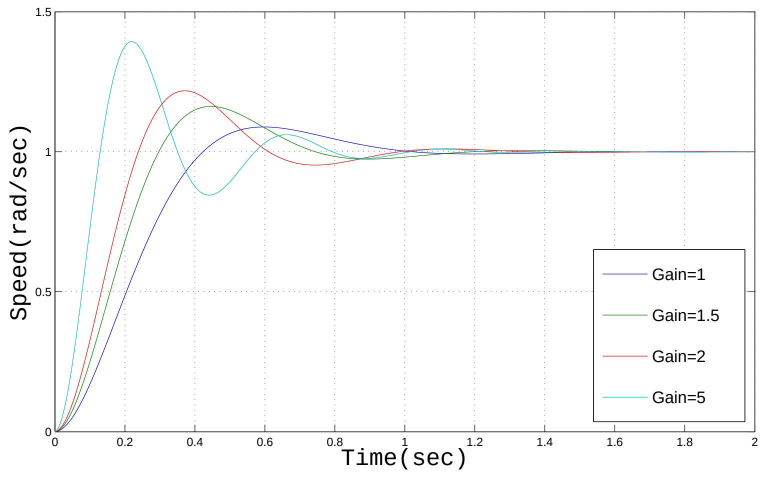


Fig. 3.9: Step response comparison with gain variation IOPI.

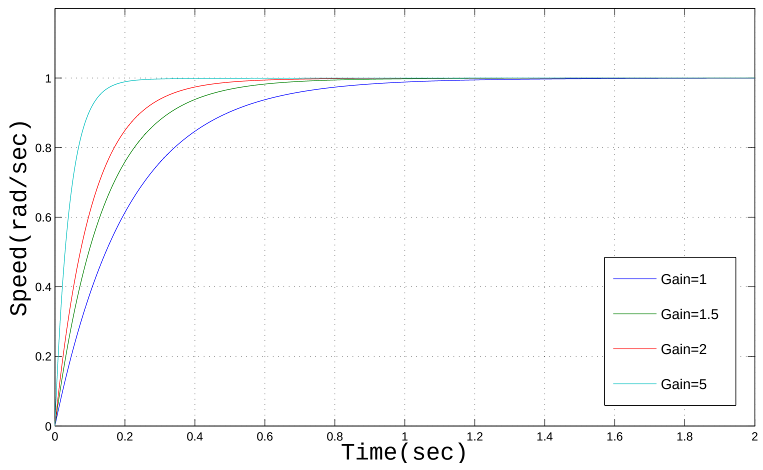


Fig. 3.10: Step response comparison with gain variations FOPI.

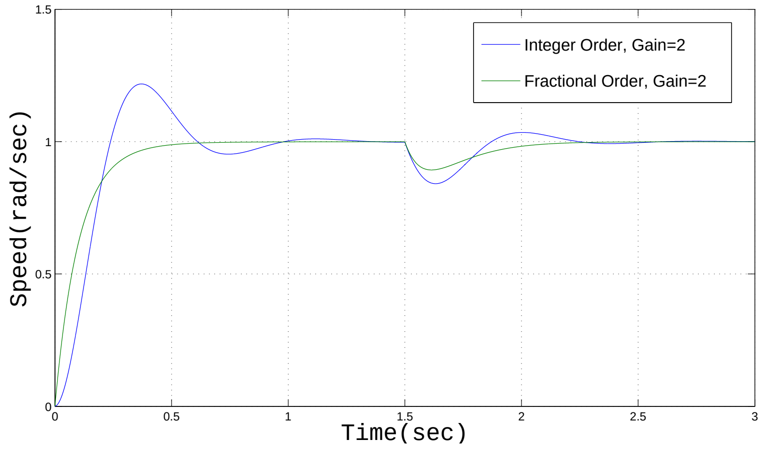


Fig. 3.11: Disturbance rejection comparison.

3.2 A Simple Dead-Zone Compensation Method for IOPI and FOPI Controllers

3.2.1 Modified Control Law

In order to compensate for the dead-zone, the control law is modified to take the form of Equation (3.3). Here $r(t)$ is the reference speed, and the sign function will give as the desired direction of movement (+ forward, - backwards). A is the total width of the dead-zone divided by two, in this case A=12, and K is a tuning constant to obtain the optimal

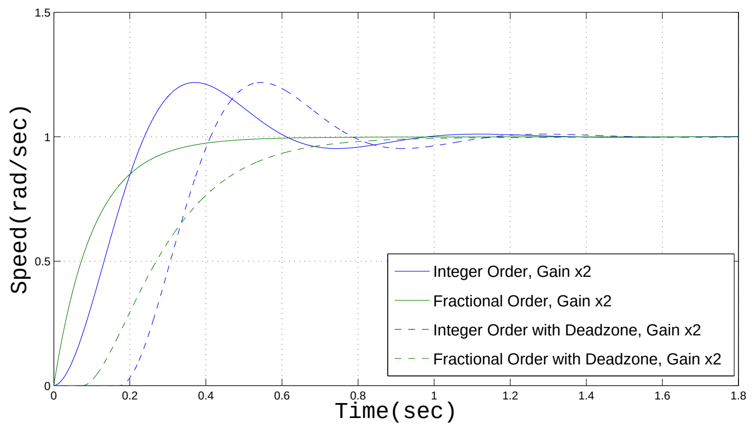


Fig. 3.12: Effects of dead-zone (-12, 12) on IOPI and FOPI.

effect for the compensation. R is a constraint, meaning that we want the compensation law to work for a desired range of speeds, in our case for the range $-0.5 \leq r(t) \geq 0.5$ the compensation is not used.

$$u(t) = (r(t) - y(t))K_p + K_i \int (r(t) - y(t))dt$$

$$+[(sign(r(t)))(A)(K)], \qquad (3.3)$$

for $r(t) > R$ and $r(t) < -R$, otherwise the term $(sign(r(t)))(A)(K) = 0$.

This range of operation for $r(t)$ is set to avoid oscillations with set point changes (similar to chattering). The tuning results are shown in Table 3.1.

3.2.2 PI Controller vs Dead-Zone Compensated PI

In Figure 3.13, the response of the compensated and not compensated IOPI and FOPI controllers is shown. The compensated FOPI totally eliminates the effect of the dead-zone.

Table 3.1: Tuning parameters for IOPI and FOPI.

IOPI		FOPI	
A	12	A	12
K	0.7	K	0.6
R	0.5	R	0.5

The compensated IOPI on the other hand would need bigger gains to eliminate the delay in the response, but this would also produce a greater overshoot. Figure 3.14 shows the response of the compensated versions of PI controllers, being the dead-zone compensated FOPI the one with the best overall performance, i.e. it manages to track the changes in setpoint with no overshoot, and faster than the others.

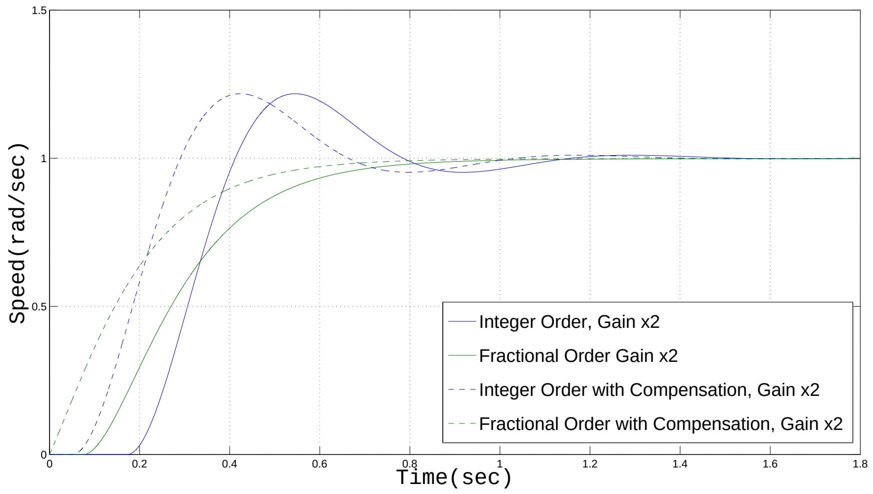


Fig. 3.13: PI vs dead-zone compensated PI.

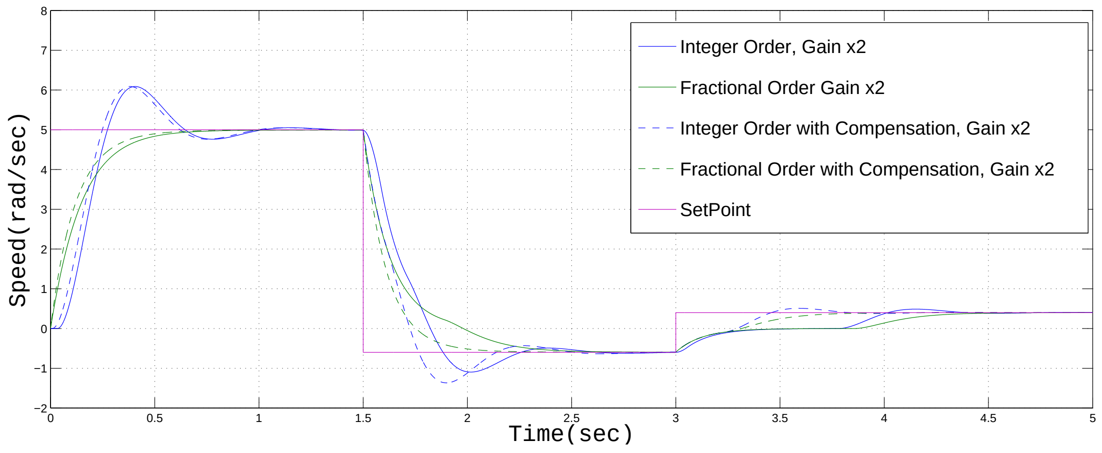


Fig. 3.14: Response to setpoint changes.

Chapter 4

A Malicious Vehicle Attack to a String Stable Platoon, Control Architectures, and Robustness Analysis

Vehicle platooning is capable of providing great benefits in terms of energy efficiency, reduced travel time, safety, and passenger comfort [6–8]. The investigations have focused mainly in achieving string stability [9–12] (to make sure that the overall systems is stable), emergency braking scenario (safety in cases of eminent collision), the separation distance achievable [13] (to further reduce drag coefficient) and communications delay (to make feasible the leader-predecessor architecture) [10].

This chapter shows why and how these benefits could be jeopardized.

4.1 Motivating Example

Consider two rival companies, X and Y, whose business consist in shipping products nationwide. Let us further assume that company X is a national leader in the business and company Y is a growing rival who is gaining a lot of ground by offering an efficient and less expensive service. Furthermore, there exist automated highway systems which are the mean both use to provide their service. Company X has two options in order not to keep losing ground: reduce their service price (which for the amount of service they provide could mean less millions in revenue a year), or to make company Y less competitive by engaging in a corporate attack (which results in higher operation costs and pushes company Y to increase their service rates). The second results a lot cheaper for company X so they decide to use a malicious vehicle in the platoons formed with company Y's vehicles.

This scenario has been studied in order to identify the most effective behavior for the attacker. In this paper we examine the effects under different circumstances, taking into account the drive train technology of the vehicles (i.e. internal combustion engine, hybrid, and electric vehicles), the platoon size, and the tuning of the controllers for the bi-directional and leader-predecessor architecture.

4.2 Threat Model

Following the motivating example, the case of a single actor in control of a vehicle within a platoon traveling at a constant speed targeting two different scenarios is examined. In the first one, the objective is to attack another vehicle in the platoon with the intent of causing the victim's vehicle to expend more energy than would otherwise be necessary. In the second, the intent is to affect the whole platoon. To determine the most effective way of behavior for the attacker to meet its goal (by accelerating and braking), we will use Matlab's GA (genetic algorithm) [14,15]. To analyze the impact of such a maliciously-minded actor, we considered a straight, dedicated automated highway lane employing leader-follower dynamics under a constant-spacing policy [13]. This scenario is analyzed with the following constraints: 1) The attacker cannot act as the leader vehicle; all vehicles in the platoon, including the attacker, are the same in terms of performance, dimensions, and mass. 2) The attacker obeys the control law. 3) The acceleration, and speed of the leader, are not affected by the attack, i.e. the attacker cannot set the platoon headway.

4.3 Bi-Directional Architecture

The bi-directional scenario can be represented by the dynamics of a mass-spring-damper system [9]. The state space equations for the system are given as

$$\begin{aligned}
\dot{x}_1 &= v_1, \\
\dot{v}_1 &= -\frac{k}{m}x_1 - \frac{c}{m}v_1 + \frac{k}{m}x_2 + \frac{c}{m}v_2 + \frac{u}{m}, \\
\dot{x}_2 &= v_2, \\
\dot{v}_2 &= \frac{k}{m}x_1 + \frac{c}{m}v_1 - \frac{2k}{m}x_2 - \frac{2c}{m}v_2 + \frac{k}{m}x_3 + \frac{c}{m}v_3, \\
&\cdot \\
&\cdot
\end{aligned} \tag{4.1}$$

.

$$\begin{aligned}
\dot{x}_{n-1} &= v_{n-1}, \\
\dot{v}_{n-1} &= \frac{k}{m}x_{n-2} + \frac{c}{m}v_{n-2} - \frac{2k}{m}x_{n-1} - \frac{2c}{m}v_{n-2} + \frac{k}{m}x_n + \frac{c}{m}v_n, \\
\dot{x}_n &= v_n, \\
\dot{v}_n &= \frac{k}{m}x_{n-1} + \frac{c}{m}v_{n-1} - \frac{k}{m}x_n - \frac{c}{m}v_n.
\end{aligned}$$

In this the matrix representation of the system is used,

$$\begin{aligned}
\dot{x} &= Ax + Bu + Ew, \\
y &= Cx,
\end{aligned} \tag{4.2}$$

where $x = [x_i \ v_i....x_n \ v_n]'$ and x_i is the position of the i-th vehicle in the platoon, v_i is the velocity of the i-th vehicle in the platoon, and n is the total number of vehicles in the platoon for $i = 1, 2, ..., n$ and matrices A, B , C, and E have the following form:

$$\mathbf{A} = \left[\begin{array}{cccccccccc}
 & 1 & & & & & & & & \\
-k/m & -c/m & k/m & c/m & & & & & & \\
 & & & 1 & & & & & & \\
k/m & c/m & -2k/m & -2c/m & k/m & c/m & & & & \\
 & & & & & 1 & & & & \\
 & & k/m & c/m & -2k/m & -2c/m & k/m & c/m & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & & & & & & & & \\
 & & & & k/m & c/m & -2k/m & -2c/m & k/m & c/m \\
 & & & & & & & & & \\
 & & & & & & & & & 1 \\
 & & & & & & k/m & c/m & -k/m & -c/m
\end{array}\right]_{2nx2n},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 1/m & 0 & .... & 0 \end{bmatrix}^T_{1x2n}, \qquad \mathbf{C} = \begin{bmatrix} I \end{bmatrix}_{2nx2n},$$

and $\mathbf{E}$ is a $1x2n$ column matrix with zero elements except for the row coinciding with the acceleration of the attacker which is $1/m$, i.e. this matrix specifies the position of the attacker and w the disturbance force the attacker inputs to the system. In this model, the only concern is the longitudinal control of the platoon; the lateral control of the vehicles is not considered. All the controllers are tuned with the same parameters k and c (spring and damper constant represent the position and speed gains together with the response of the vehicles). The mass of the vehicles is assumed to be the same, and in order to make the system response analysis simpler we set $m = 1$. Furthermore, we do not allow the attacker to disturb the leader, i.e. set the headway, and thus the second row of matrix A is set to zero. We change the coordinates of the system to be $x_i = 0$ nominal position of the i^{th} vehicle and $v_i = 0$ nominal velocity of the i^{th} vehicle and thus setting $B = 0$ we have the platoon traveling at a constant velocity and maintaining nominal separation (we are using the constant space policy because if provides greater energy benefits) [6].

String stability requires that the error magnitude attenuates along the platoon, i.e. $|z_i| > |z_{i+1}|$. By defining new variables $z_i = x_i - x_{i+1}$ and $h_i = v_i - v_{i+1}$, we get the system in error coordinates which is better to analyze string stability as follows [9]:

$$
\begin{aligned}
\dot{z}_1 &= h_1, \\
\dot{h}_1 &= -\frac{2k}{m} z_1 - \frac{2c}{m} h_1 + \frac{k}{m} z_2 + \frac{c}{m} h_2 + \frac{u}{m}, \\
\dot{z}_2 &= h_2, \\
\dot{h}_2 &= \frac{k}{m} z_1 + \frac{c}{m} h_1 - \frac{2k}{m} z_2 - \frac{2c}{m} h_2 + \frac{k}{m} z_3 + \frac{c}{m} h_3, \\
&. \\
&. \\
&. \\
\dot{z}_{n-2} &= h_{n-2}, \\
\dot{h}_{n-2} &= \frac{k}{m} z_{n-3} + \frac{c}{m} h_{n-3} - \frac{2k}{m} z_{n-2} - \frac{2c}{m} h_{n-2} + \frac{k}{m} z_{n-1} + \frac{c}{m} h_{n-1}, \\
\dot{z}_{n-1} &= h_{n-1}, \\
\dot{h}_{n-1} &= \frac{k}{m} z_{n-2} + \frac{c}{m} h_{n-2} - \frac{2k}{m} z_{h-1} - \frac{2c}{m} h_{n-1}.
\end{aligned}
\tag{4.3}
$$

In matrix form,

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u, \\ \bar{y} &= \bar{C}\bar{x}, \end{aligned} \tag{4.4}$$

where $\bar{x} = [z_i \ h_i .... z_{n-1} \ h_{n-1}]'$ and n is the total number of vehicles in the platoon for $i = 1, 2, ..., n-1$ and matrices $\bar{A}$, $\bar{B}$, and $\bar{C}$ have the following form:

$$\mathbf{\bar{A}} = \left[\begin{array}{cccccccccc} & 1 & & & & & & & & \\ -2k/m & -2c/m & k/m & c/m & & & & & & \\ & & & 1 & & & & & & \\ k/m & c/m & -2k/m & -2c/m & k/m & c/m & & & & \\ & & & & & 1 & & & & \\ & & k/m & c/m & -2k/m & -2c/m & k/m & c/m & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & k/m & c/m & -2k/m & -2c/m & k/m & c/m \\ & & & & & & & & & 1 \\ & & & & & & k/m & c/m & -2k/m & -2c/m \end{array} \right]_{2(n-1)x(n-1)},$$

$$\mathbf{\bar{B}} = \left[\begin{array}{ccccc} 0 & 1/m & 0 & .... & 0 \end{array} \right]^T_{1x2(n-1)}, \quad \text{and} \quad \mathbf{\bar{C}} = \left[\begin{array}{c} I \end{array} \right]_{2(n-1)x2(n-1)}.$$

String stability is achieved (gain of the system < 1 for all frequencies) depending on the number of vehicles in the platoon by a ratio $c^2 > pkm$ [9], where p increases with the number of vehicles in the platoon. Computing this constant p is hard, it is different for all platoon length and thus the knowledge of p (increases with number of vehicles in the platoon)is used to graphically tune the gains by plotting the system response using Matlab's Linear Simulation Tool "lsim," and readjusting the gains to obtain string stability. The result implies having an asymptotically stable system matrix $\bar{A}$, i.e. all eigenvalues are real negative. Table 4.1 shows the obtained gains to achieve string stability for the different platoon lengths.

Then calculating $p < c^2/km$, we have can obtain the minimum ratios that guarantee string stability as shown in Table 4.2.

Table 4.1: Gains for a string stable platoon.

Platoon Size	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Spring Constant K	1																	
Damper Constant C	2.1	2.7	3.3	4.2	5.1	6	7	7.7	8.6	9.2	9.8	10.4	11	11.5	12	12.5	13.1	13.9

Table 4.2: p ratio for a string stable platoon.

Platoon Size	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ratio p	4.41	7.29	10.89	17.64	26.01	36	49	59.26	73.96	84.64	96.04	108.16	121	132.25	144	156.25	171.61	193.21

A question emerged after tunning the system to make the platoon string stable; "Is the platoon robust to disturbances?" If not, "How can it be made robust?" Two important parameters of a disturbance have to be considered, which are magnitude and duration. Typical vehicle acceleration reaches $5m/s^2$, and thus we the platoon must be robust to such a magnitude. The duration of an oscillatory disturbance, i.e. breaking and accelerating of a vehicle, will produce resonance and increase the possibility of collisions. An arbitrary duration of 80 *sec* is selected in order to analyze how the tuning can help the system to be more robust. Table 4.3 shows the results for a 5-vehicle platoon. Increasing the ratio p and the gains have different effects. In general, increasing the gains results in more robustness gained for the system.

The trade-offs of the parameter tuning can be better observed in Figure. 4.1. Increasing the ratio p 5 times reduces the maximum error produced but in trade-off of an increased recovery time (time for the system to go back to equilibrium). Increasing only the gains has good effect in reducing the error and recovery time as well. In the last configuration, to avoid augmenting the gains (may not be possible due to saturation in plants), by also increasing the ratio p, further error reduction is gained, again at expense of increasing recovery time.

Table 4.3: Robustness of a string stable platoon.

5 Vehicle Platoon				
Ratio p	10.21	51.1	10.22	51.1
Damper constant C	3.3	7.155	7.155	15.99
Spring constant K	1	1	5	5
Disturbance magnitude it can handle without ending in coallision	1.2 m/s^2	1.7 m/s^2	4.3 m/s^2	5 m/s^2

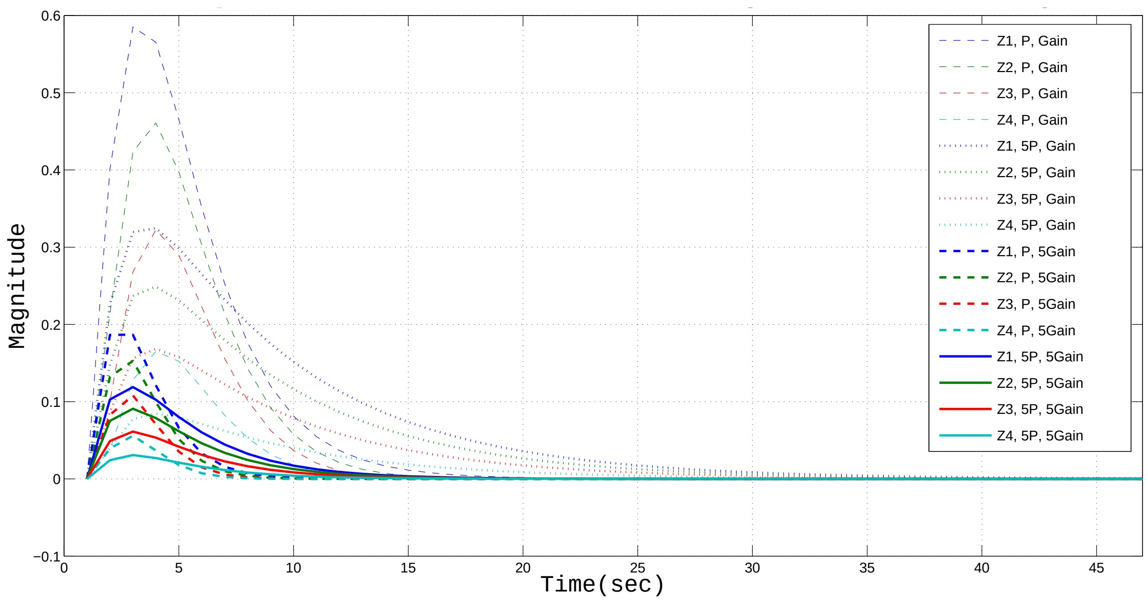


Fig. 4.1: System response to different tuning, input=10/Exp(T), where T=1:01:100, bi-directional architecture.

4.4 Leader-Follower Configuration

With the constant spacing policy, string stability can be achieved when the leader transmits its desired velocity only [9]. The state-space representation of such configuration represented by a mass-spring-damper system is

$$
\begin{aligned}
\dot{v}_d &= u/m,\\
\dot{x}_1 &= v_1,\\
\dot{v}_1 &= \frac{Cd}{m}(v_d - v_1),\\
\dot{x}_2 &= v_2,\\
\dot{v}_2 &= \frac{k}{m}x_1 + \frac{c}{m}v_1 - \frac{k}{m}x_2 - \frac{c}{m}v_2 + \frac{Cd}{m}(v_d - v_2),\\
&.\\
&.\\
&.\\
\dot{x}_{n-1} &= v_{n-1},\\
\dot{v}_{n-1} &= \frac{k}{m}x_{n-2} + \frac{c}{m}v_{n-2} - \frac{k}{m}x_{n-1} - \frac{c}{m}v_{n-1} + \frac{Cd}{m}(v_d - v_{n-1}),
\end{aligned}
\tag{4.5}
$$

$$\dot{x}_n = v_n,$$
$$\dot{v}_n = \frac{k}{m}x_{n-1} + \frac{c}{m}v_{n-1} - \frac{k}{m}x_n - \frac{c}{m}v_n + \frac{Cd}{m}(v_d - v_{n-1}).$$

In this work we are going to use the matrix representation of the system

$$\begin{aligned} \dot{x} &= Ax + Bu + Ew, \\ y &= Cx, \end{aligned} \tag{4.6}$$

where $x = [v_d\ x_i\ v_i....x_n\ v_n]'$ and x_i is the position of the i-th vehicle in the platoon, v_i is the velocity of the i-th vehicle in the platoon, v_d is the desired platoon speed communicated by the leader, and n is the total number of vehicles in the platoon for $i = 1, 2, ..., n$ and matrices A, B , C, and E have the following form:

$$\mathbf{A} = \begin{bmatrix} \cdots & 0 & \cdots & & & & & & & \\ \cdots & 0 & 1 & 0 & \cdots & & & & & \\ c_d/m & 0 & -c_d/m & 0 & \cdots & & & & & \\ & & \cdots & 0 & 1 & 0 & \cdots & & & \\ c_d & k/m & c/m & -k/m & -(c+c_d)/m & 0 & \cdots & & & \\ & & & & \cdots & 0 & 1 & 0 & \cdots & \\ c_d & 0 & 0 & k/m & c/m & -k/m & -(c+c_d)/m & 0 & \cdots & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & & & & \cdots & 0 & 1 \\ \cdots & & & & & & & & & \\ c_d & 0 & 0 & \cdots & 0 & 0 & k/m & c/m & -k/m & -(c+c_d)/m \end{bmatrix}_{(2n+1)x(2n+1)},$$

$$\mathbf{B} = \begin{bmatrix} 1/m & 0 & .... & 0 \end{bmatrix}^T_{(2n+1)x1}, \qquad \mathbf{C} = \begin{bmatrix} I \end{bmatrix}_{(2n+1)x(2n+1)},$$

and $\mathbf{E}$ is a $(2n+1)x1$ column matrix with zero elements except for the row coinciding with the acceleration of the attacker which is $1/m$, i.e. this matrix specifies the position of the attacker and w the disturbance force in time, the attacker inputs to the system. The first column of the matrix A is set to zero in order to define v_d as zero. As for the bi-directional case, the system can be represented in the error coordinate form as

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u, \\ \bar{y} &= \bar{C}\bar{x}, \end{aligned} \tag{4.7}$$

where $\bar{x} = [h_d \; z_i \; h_i....z_{n-1} \; h_{n-1}]'$, $h_d = x_1 - v_d$ and n is the total number of vehicles in the platoon for $i = 1, 2, ..., n-1$ and matrices $\bar{A}$, $\bar{B}$, and $\bar{C}$ have the following form:

$$\bar{\mathbf{A}} = \begin{bmatrix} -c_d/m & 0 & \dots & & & & & & \\ 0 & 0 & 1 & 0 & \dots & & & & \\ 0 & -k/m & -(c+c_d)/m & 0 & \dots & & & & \\ & & \dots & 0 & 1 & 0 & \dots & & \\ 0 & k/m & c/m & -k/m & -(c+c_d)/m & 0 & \dots & & \\ & & & & \dots & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & k/m & c/m & -k/m & -(c+c_d)/m & 0 & \dots \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & & & & \dots & 0 & 1 \\ & & & \dots & 0 & k/m & c/m & -k/m & -(c+c_d)/m \end{bmatrix}_{[2n-1]x[2n-1]} ,$$

$$\mathbf{B} = \begin{bmatrix} 1/m & 0 & .... & 0 \end{bmatrix}^T_{[(2n-1)]x1} , \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} I \end{bmatrix}_{[2n-1]x[2n-1]} .$$

The string stability of such a system is guaranteed if the following statement holds [9]:

$$c > (2km - c_d^2)/2c_d. \tag{4.8}$$

Still string stability is a necessary but not sufficient condition for robustness. Analyzing this system, one can see that the robustness of the platoon again depends on the weighting of the speed, and position gains. Table 4.4 shows two different tuning for a 5-vehicle platoon. The first is for a string stable platoon (see Figure 4.2), while the second is for a string stable robust platoon (see Figure 4.3). Figure 4.2(b) shows the behavior of a string stable platoon in terms of position affected by an oscillatory disturbance produced by the genetic algorithm GA (optimizing the total energy spent by the platoon). The position of the attacker and vehicles behind it (in this configuration the attacker cannot affect vehicles in front of him) is drifting from the nominal values, meaning that if the attack prolongs further, it will keep increasing until they end in a collision. In Figure 4.3(b), with the second tuning, by keeping a closer ratio between c and k, the position response is going to tend to the nominal values. It could tolerate the disturbance without resulting in a collision for a longer time.

For both architectures the gains were chosen to make the system robust, in the sense that it would be capable of recovering to a normal state after being perturbed (i.e. without ending in a collision). This is because a feasible platoon should be able to guarantee safety, and comfort to passengers.

Table 4.4: String stable vs robust platoon.

	String stable platoon	Robust platoon
K	2	10
C	129.4	10.45
C_d	0.017	0.1

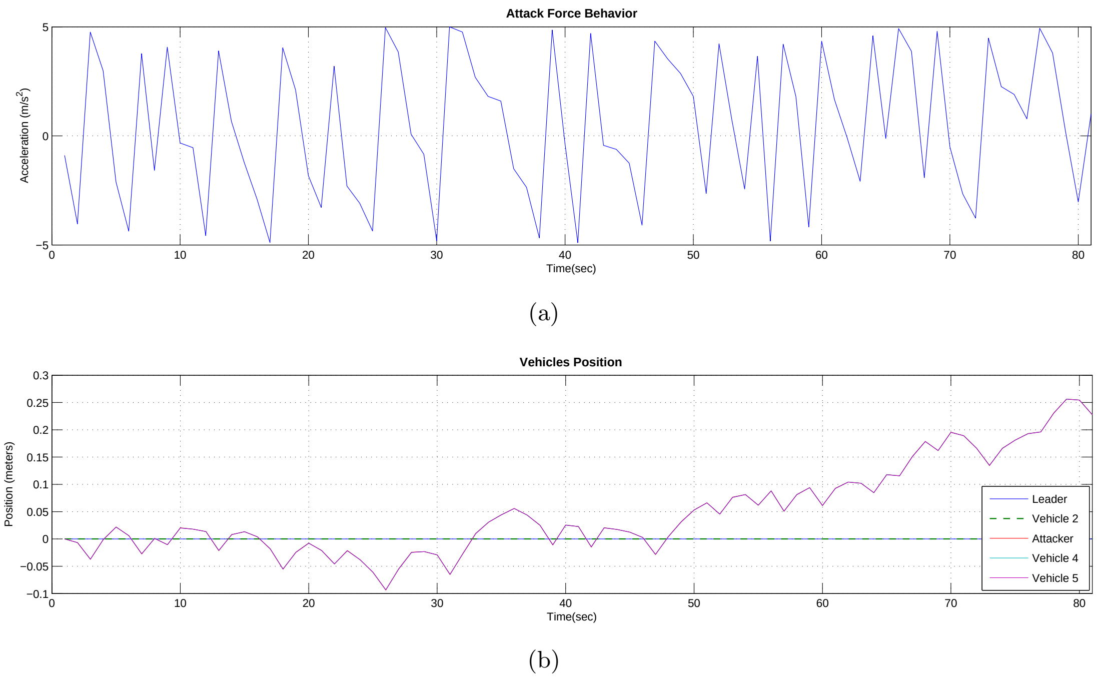
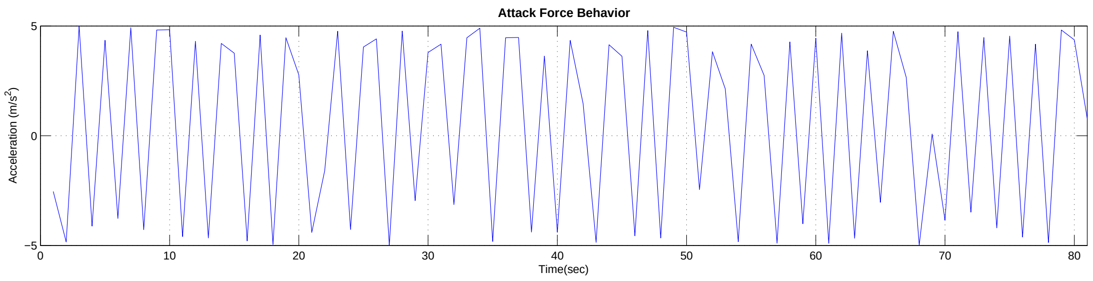
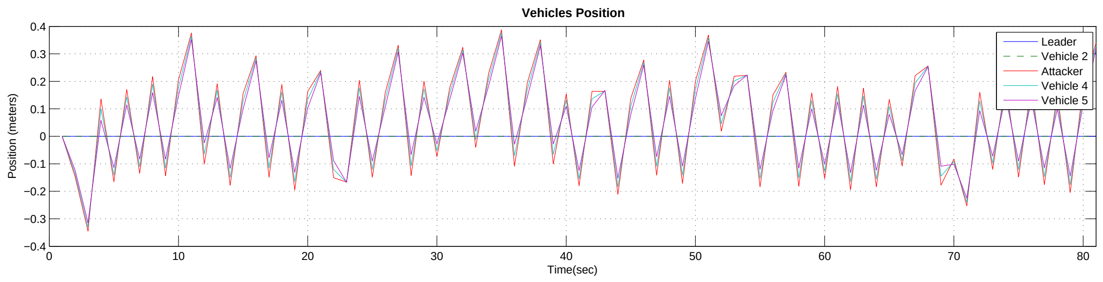


Fig. 4.2: String stable platoon response, leader-follower architecture a) attacker input force, b) platoon position response.



(a)



(b)

Fig. 4.3: Robust platoon response, leader-follower architecture a) attacker input force, b) platoon position response.

Chapter 5
Energy Calculation Model and Simulation

The goal of an attacker is formulated as an optimization problem. The simulation framework created to determine the platoon response to, and calculate energy expenditure as a result of, a malicious vehicle movement is described.

5.1 Energy Calculations

Using the system models for the bi-directional and leader-predecessor architectures in matrix form and Matlab's linear simulation toolbox lsym, the response of the system (position, velocity, and acceleration of each vehicle) to an input u for a time T can be determined. This information can be used to calculate the fuel consumption of a vehicle. According to cappiello's model [16], the fuel consumption rate (g/s) for a vehicle with velocity v and acceleration a is stated as

$$FR = \begin{cases} \alpha_{FR} + \beta_{FR} v + \gamma_{FR} v^2 + \delta_{FR} v^3 + \zeta_{FR} a v & \text{for } P > 0 \\ \alpha'_{FR} & \text{for } P \leq 0, \end{cases} \tag{5.1}$$

where P is

$$P = Av + Bv^2 + Cv^3 + Mav + Mg \sin \vartheta v, \tag{5.2}$$

and ϑ is the inclination angle of the road. A flat road was assumed in the simulations $\vartheta = 0$.

The constants of Equations (5.1) and (5.2) depend on the vehicles (hardware capability and state). Cappiello relied upon data from the National Cooperative Highway Research Program (NCHRP) vehicle emissions database [17] in order to calibrate the models. The newest vehicle with the fewest number of miles was selected to act as our representative platooning vehicle (i.e. each of the vehicles in the platoon is identical and represented by a specific vehicle for which we have access to fuel consumption data). This corresponded to a 1995 Honda Accord LX (vehicle 259 in the NCHRP list [17]) with $49,764$ miles, a 3-way catalytic converter and fuel injection system, and a high power-to-weight ratio. In the nomenclature used for the NCHRP database, it is a category 7 vehicle with $A = 0.0286\,\mathrm{kW/(m/s)}, B = 4.556 \times 10^{-8}\,\mathrm{kW/(m/s^2)}, C = 1.392 \times 10^{-12}\,\mathrm{kW/(m/s^3)},$ and $M = 1361\,\mathrm{kg}$. The constants used for Equation (5.1) are $\alpha_{FR} = 0.326, \beta_{FR} = 0.002\,28, \gamma_{FR} = 0, \delta_{FR} = 9.42 \times 10^{-7}, \zeta_{FR} = 0.0957,$ and $\alpha'_{FR} = 0.300$.

From lsim, the response the system contained in a matrix is obtained. After some reorganization, two matrices, one containing speed information, and one containing the acceleration are generated. First, the base fuel consumption (unperturbed system $a = 0$) is calculated. This value is subtracted from the total energy consumed (where a is due to the input u) in order to find the extra energy consumed at each sampling time. Matlab's numerical integrator "trapz" was used to find the total energy consumed by each vehicle during the time T. These values add up to sum the total energy expended by the platoon.

5.2 Optimization Problem

In the first attack (victim-based) using the model defined in Section 5.1, the genetic algorithm "GA" determines the optimal attacker input in order to minimize the ratio $1/Ev$, where Ev is the energy expended by the victim and thus by minimizing its inverse, we are actually maximizing Ev. The second attack seeks to maximize the overall energy expenditure of the whole platoon, the ratio to minimize is $1/Ep$ where Ep is the sum of the energy expended by all vehicles in the platoon.

To calculate the energy, the state space system is analyzed using Matlab's lsim (Linear Simulation Tool) to readily obtain acceleration, position, and velocity from all vehicles. The input is provided by the genetic algorithm with the constraint $F \pm 5\ kg.m/s^2$. The genetic algorithm passes a vector u specifying the best level of force at each second. The genetic algorithm runs a hundred times and the result is the input u that maximizes the energy expended Ep or Ev. An additional constraint is introduced to make sure there are no collisions in the platoon. If a collision is detected, the system returns Ep=0 or Ev=0.

The problem can be stated as

$$\begin{aligned}
&\underset{F_{atk}(t)}{\text{maximise}}\ E(v_{vct}, F_{atk}(t)) \\
&\qquad \textit{subject to} \\
&-\beta \leq a_i(t) \leq \alpha, i = 1, \cdots, n, \\
&x_i(t) - x_{i-1}(t) < \zeta, i = 2, \cdots, n, \\
&|x_i(T_a) - \tilde{x}_i(T_a)| < \eta, i = 1, \cdots, n,
\end{aligned} \tag{5.3}$$

over the interval $t = [0, T_a]$, where α and β are the maximum amount of positive and negative acceleration the vehicles are capable of, in this case $\beta = -$, ζ is the minimum inter-vehicle separation allowable before a collision is declared, $\tilde{x}_i(T_a)$ the final position of the i^{th} vehicle in the absence of malicious movement on the part of the attacker.

In plain words we have three constraints:

- Performance capabilities of the vehicles in the platoon cannot be exceeded;
- Collisions cannot occur;
- The platoon must be capable to recover to initial state, as to guarantee that at end of the attack, the positions would be the same as if it had never happened.

Chapter 6

Simulation Results from Malicious Attack

The results of applying the simulation framework to platoons of varying size and composition are presented and discussed.

6.1 Simulation Parameters

In order to determine the effects of the attacker's behavior and figure out which measures can be taken in order to reduce its effectives, we examined energy expenditure of the platoon for 1) different attacker positions, 2) control architectures, 3) tuning of the system, 4) drive train technology (dissipative and regenerative braking). The results were taken for the two different attacks, platoon focused and an individual victim. First, the focus will be placed on the bi-directional control architecture and then in the leader-predecessor configuration. The fixed simulations parameters are the vehicle specifications as in Section 5.1, the inter-vehicle separation which is 1 m, the maximum allowable acceleration $-5m/s^2 < a < 5m/s^2$, the nominal platoon velocity which is 11 m/s and the attack duration $T = 80\,\sec$.

6.2 Bi-Directional Architecture Results

In the platoon-based attack for the bi-directional case, we can see from Figure 6.1 that increasing the gains has the effect of reducing the attack effectiveness. Also, the optimal position for the attacker is $Pa = N$, i.e. the last vehicle in the platoon. For the robust platoon the extra energy expended increases as the attacker's position is closest to the last. Another fact that could be determined is that the percentage of extra fuel consumed by the platoon is linear and inversely proportional to the platoon size. Figure 6.2 shows how the extra fuel consumption varies for different platoon sizes, using the robust control tuning.

For each case, the attacker is in last position.

In the case of an attack focused on a victim, we could determine that it is better for the attacker to be closer to the victim. Still, the optimal position for this attacker is behind the victim. Even being two vehicles behind the victim is better for the attacker than being in front. But for the two cases, in front or behind, the closer the better, as depicted in Figure 6.3. Also, the extra energy expended by the victim depends on its position in the platoon as well. In fact, it follows the same pattern as in the platoon based attack. It is better for the attacker as the victim moves farther way from the leader to the last position in the platoon.

The results presented so far describe the behavior of the conventional vehicles with dissipative braking (all the energy when braking is dissipated as heat). It is important to analyze what happens when we a modern drive train technology which are the trend, such as hybrid, and electric vehicles are used. Both of these technologies use an electrical motor to drive the vehicle which allows for regenerative braking capability (an amount of the energy expended while accelerating, returns to the battery through the electric motor which acts as a generator).

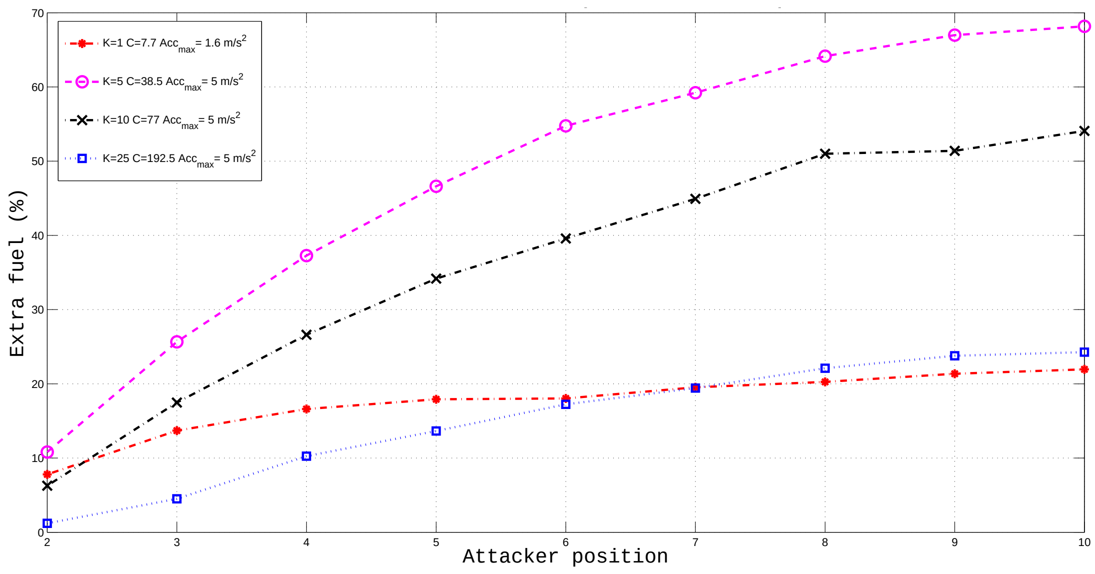


Fig. 6.1: Extra energy expended by a platoon, bi-directional scenario, all possible attacker positions, and different tuning: a 10 size platoon.

Some companies, such as Ford, have claimed regenerative breaking efficiency up to 90%. This depends on the way of braking, and thus having a dynamical model braking system, the attacker could try to reduce its efficiency. A detailed model of the regenerative braking system is not available for this study and thus, a 50% constant regenerative breaking capability is assumed. Figure 6.4 shows the extra energy expenditure profiles for different platoon sizes. The red line represents dissipative brakes while the blue line represents the regenerative capability brakes. Clearly, the attack is less effective when modern drive train technologies are used. The green line represents the percentage of energy saved by the regenerative braking. From 35% to 50% less energy is spent when using a regenerative breaking system when compared to a dissipative one.

Also, the strategy of the attacker changes when regenerative braking is applied. Figure 6.5 shows the solution for the optimization problem found by the genetic algorithm. It is a different solution than the dissipative braking case where the attacker accelerates and brakes with about the same period (see Figure 4.2). In this case, because of the regenerative braking, the optimal solution implies keeping the braking times as short as possible.

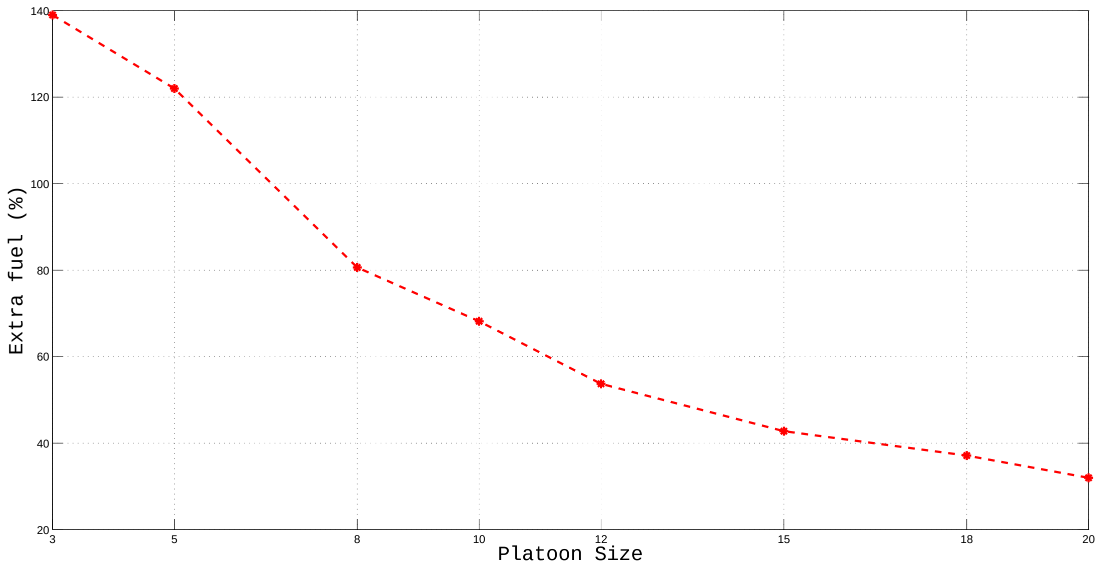


Fig. 6.2: Extra energy expended by a robust platoon, bi-directional scenario, attacker in last position.

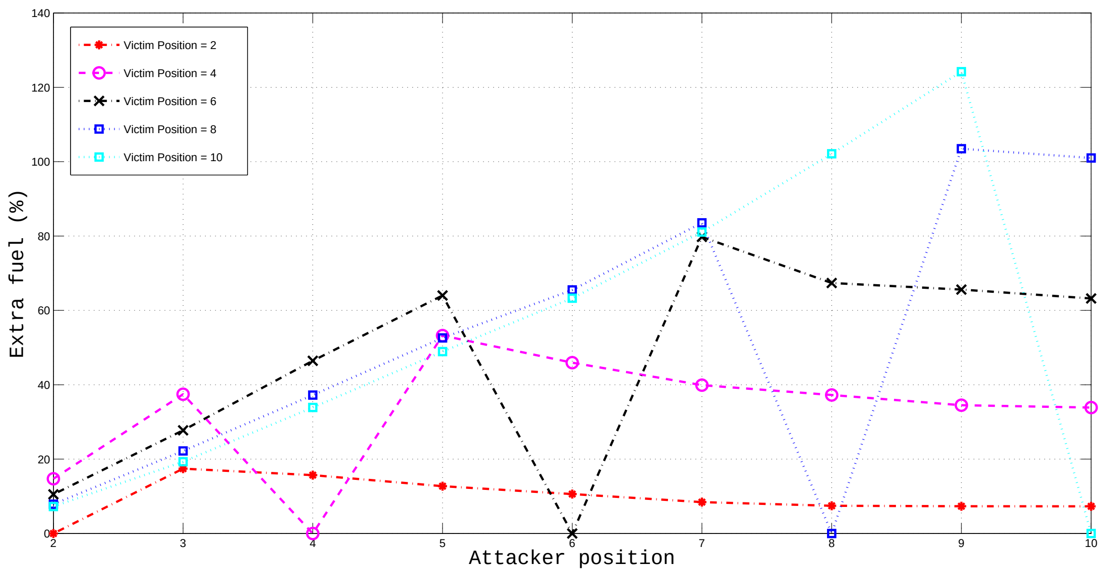


Fig. 6.3: Extra energy expended by a victim, bi-directional scenario, different attacker positions, different victim positions.

6.3 Leader-Predecessor Architecture Results

The platoon-based attack in this scenario shows a different result than for the bi-directional case, where the optimal attacker position was the last vehicle in the platoon. Figure 6.6 shows that in this leader-predecessor configuration, the closest the attacker is to the leader the more effective its maneuvers are. This result is not surprising, from the dynamics of the system one can see that the vehicles do not take into account the response of the followers, and thus the attacker can only affect vehicles behind him. The optimal attacker position is $Pa = 2$. Figure 6.7 shows how the effects of the attack vary with platoon size. The biggest amount of extra energy expenditure happens for platoons sizes 3 to 11, and in a nonlinear manner. For platoons size 12 to 17, the bigger the platoon, the less extra energy spent. In platoons size 17 or more, the variation in energy spent seems minimal.

Figure 6.8 shows a victim-focused attack. It verifies our early statement that the attacker can only affect vehicles behind it. A surprising result arises. The optimal position for this attack scenario is not defined. It depends on the position of the victim and the attacker as well, and it is not what we may expect. The optimal position for the attacker is 3-4 vehicles in front of the victim. It was expected to be right before the victim.

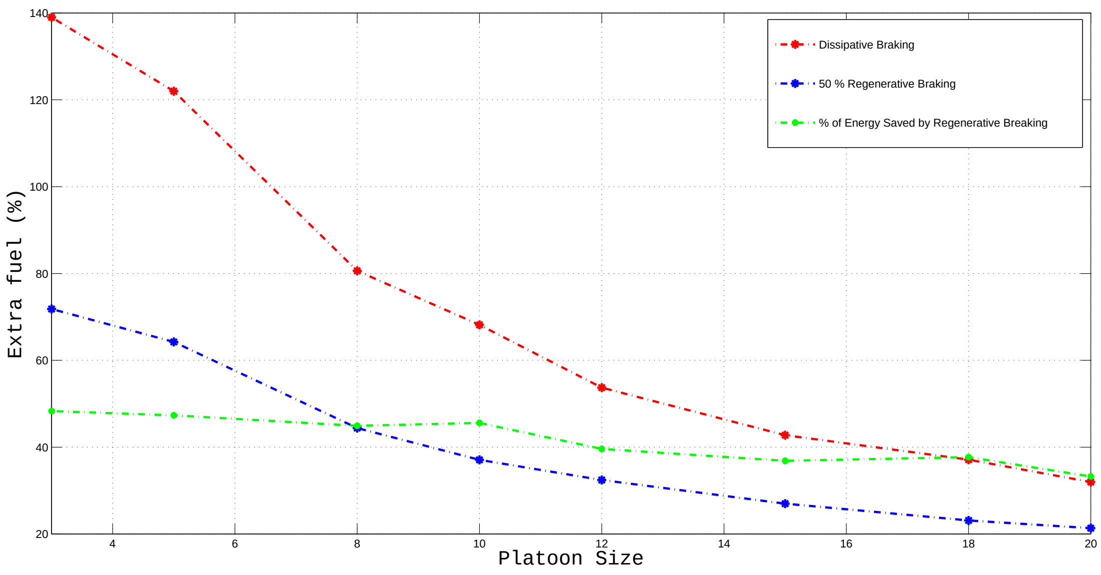


Fig. 6.4: Extra energy expended by a platoon, bi-directional scenario, attacker in last position, dissipative braking, regenerative braking, robust platoon.

Regenerative capability braking shows the same advantages for the leader-predecessor configuration as for the bi-directional architecture.

6.4 Vehicle Performance Considerations

In an attempt to better understand how performance would affect the attack, vehicles were grouped in three categories:

- **Low performance vehicle**, those with acceleration $\leq 2.0m/s^2$;
- **Medium performance vehicle**, those with acceleration $> 2.0m/s^2$ but $\leq 4.0m/s^2$;
- **High performance vehicle**, those with acceleration $> 4.0m/s^2$ but $\leq 6.0m/s^2$.

Simulations were run with the attacker in the last position and with different controller gains to better understand the relationship between the increased gains, the position error and the acceleration required for the followers in a platoon formed by five automobiles. Figures 6.9-6.12 show that the higher the gains in the control system, the smaller the error produced and thus the acceleration required overcoming it. In the case of low gains as can be seen in Figure 6.9, the maximum acceleration required for the attacker is $2.8m/s^2$,

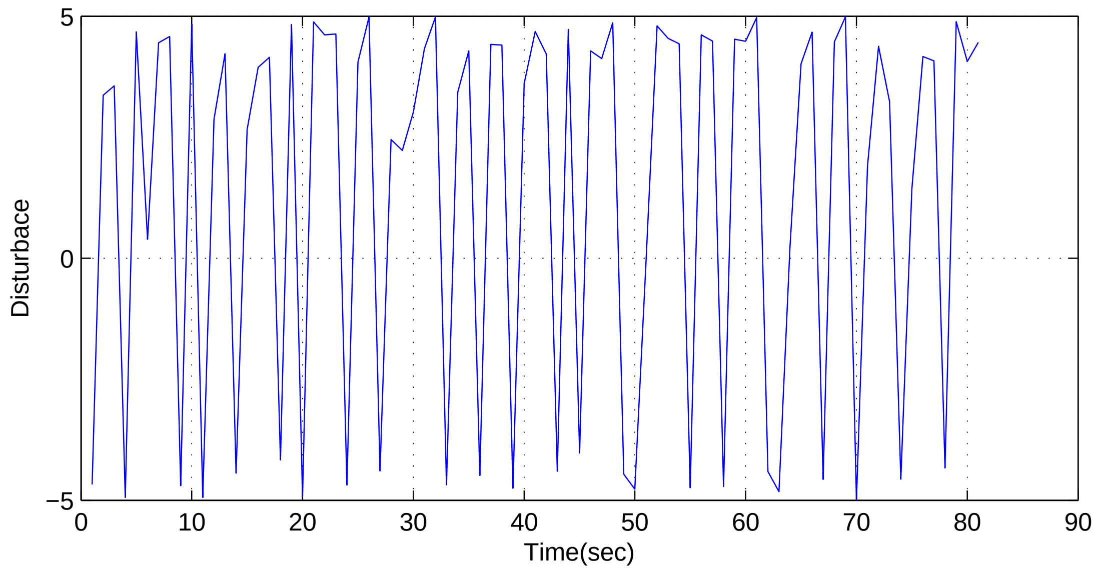


Fig. 6.5: Optimal attack force found by the GA, force level in newton, since m=1 the Y axis represents force, and acceleration in m/s^2 as well.

which means that the attacker's vehicle would need to be in the medium range, and thus it is beneficial for the attacker to have a higher performance vehicle. We can also add that the higher the gains in the control system, the better a platoon formed by low performance vehicles (i.e. trucks and trailers) will react in the presence of an optimal attack.

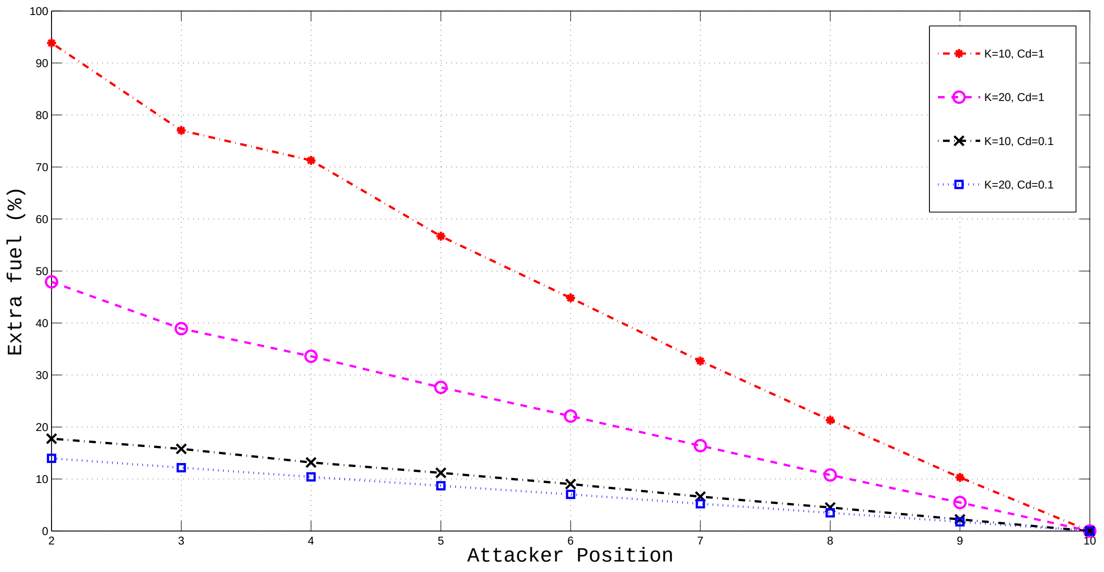


Fig. 6.6: Extra fuel consumed by a 10 size platoon, leader-predecessor architecture, all different attacker positions, and tuning effects.

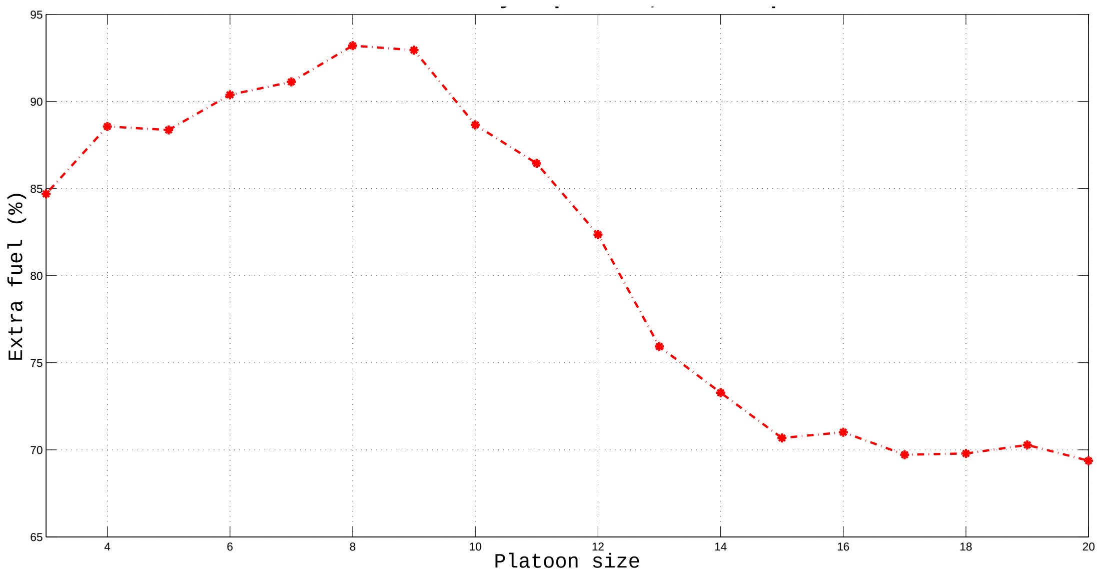


Fig. 6.7: Extra fuel consumed by a platoon, leader-predecessor architecture, attacker position=2.

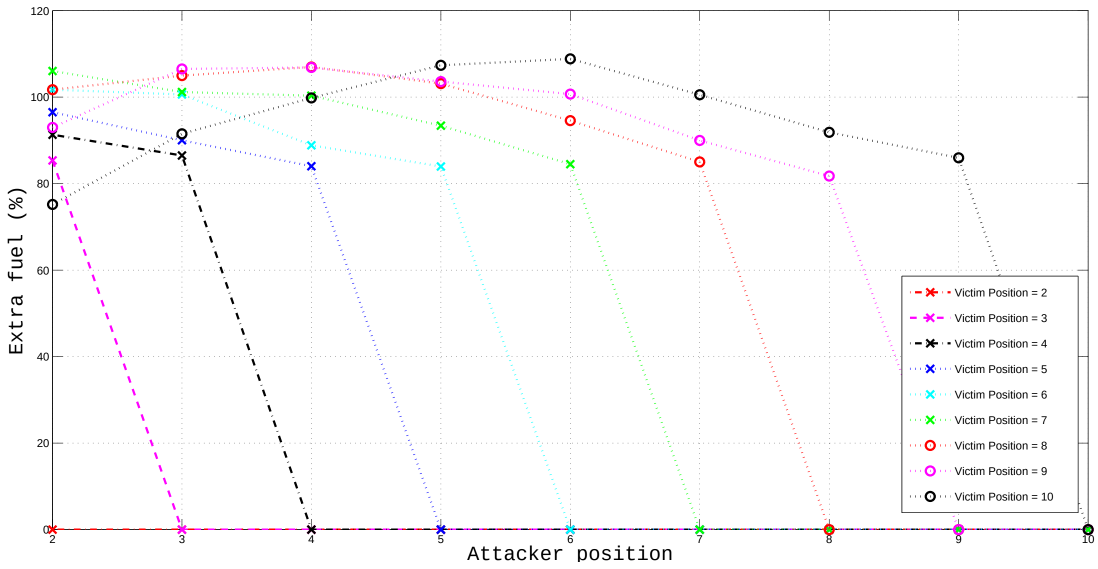


Fig. 6.8: Extra fuel consumed by a victim, 10 size platoon, leader-predecessor architecture, all different attacker positions, all victim positions.

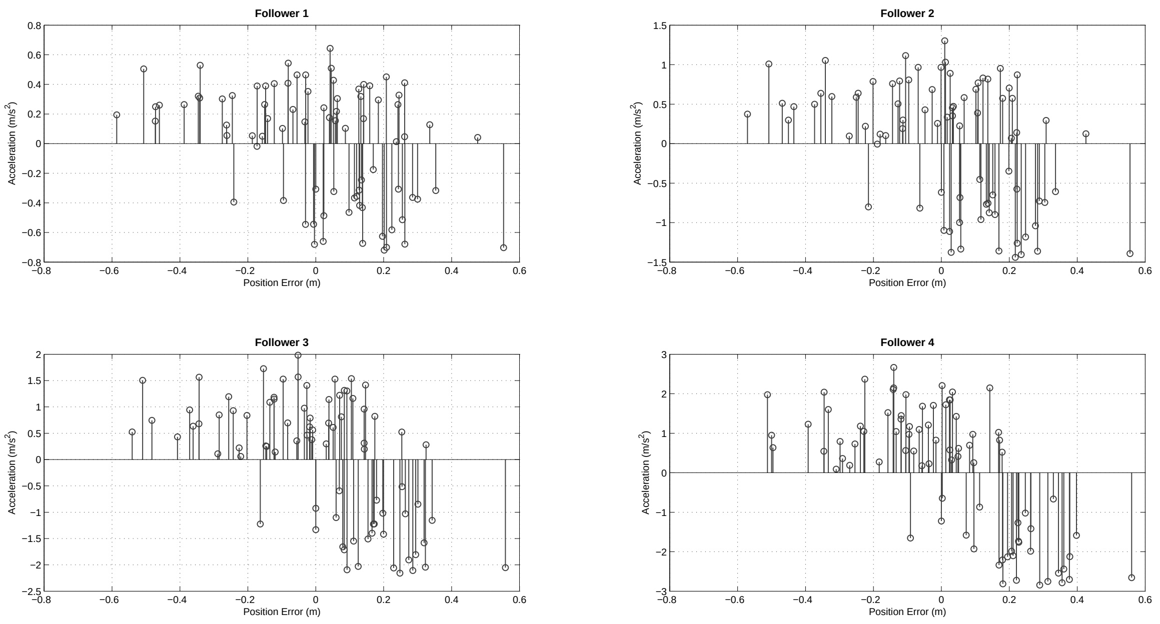


Fig. 6.9: Acceleration of followers in a 5-vehicle platoon during an optimal attack, C=9.9 and K=6.

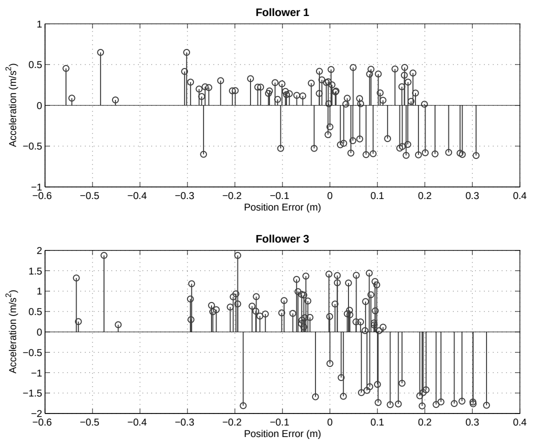


Fig. 6.10: Acceleration of followers in a 5-vehicle platoon during an optimal attack, C=13.2 and K=8.

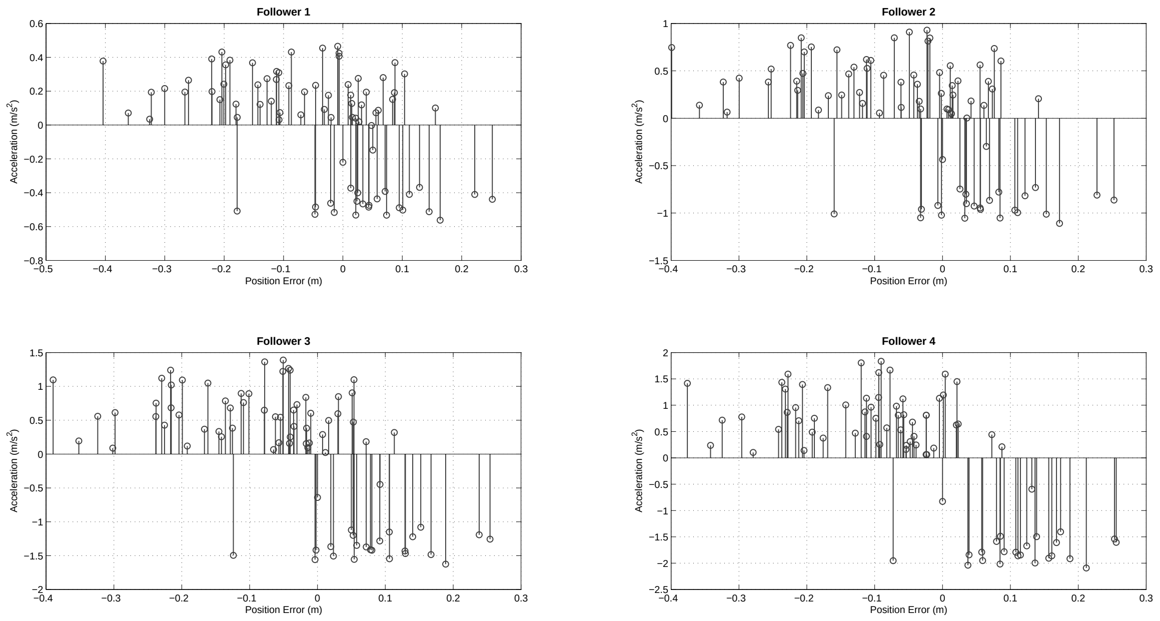


Fig. 6.11: Acceleration of followers in a 5-vehicle platoon during an optimal attack, C=16.5 and K=10.

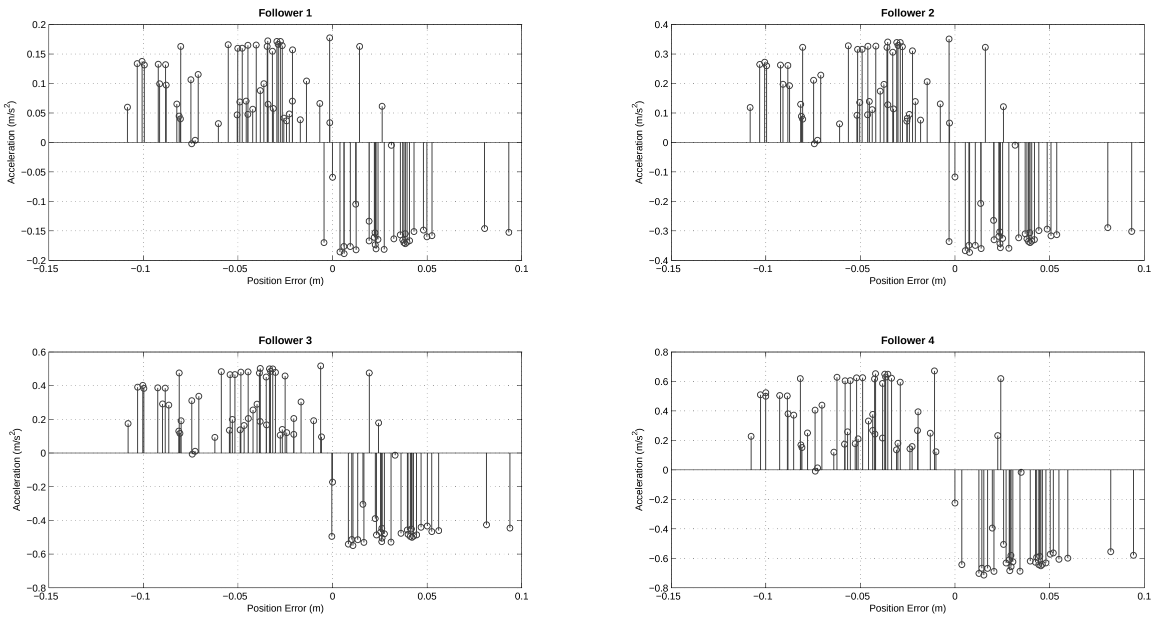


Fig. 6.12: Acceleration of followers in a 5-vehicle platoon during an optimal attack, C=49.5 and K=30.

Chapter 7
Conclusions and Future Work

7.1 Conclusions

With the help of the existing platoon dynamic models and taking advantage of the available computing power of actual hardware and software, a platform to simulate the behavior of a vehicle platoon of different size was developed. The bi-directional and leader-predecessor control strategies which have been proven to achieve string stability were taking into consideration in order to try to verify the hypothesis that one vehicle, behaving in a particular manner, could jeopardize the valuable benefits of having an automated vehicle platoon system.

Two different approaches were analyzed: the impact of the attack when trying to affect the entire platoon, and the impact of the attack while focusing only on one victim in particular. It was found that depending on the control strategy used, the attack will in fact have a very different effect. Still in all cases, a considerably big amount of extra energy would be expended.

For both cases we consider a robust control system. Recall from Section 6.2 that increasing the controller gains, which also increases the overall response of the system, decreases the maximum extra energy the attacker could make the platoon spend. In Section 6.4 it was shown that these gains are directly related to the magnitude of error that could be caused by a perturbation (attack) on the system, and thus the magnitude of acceleration required to correct it. This showed us that in fact the attacker would profit from having a superior performance vehicle than the average in the platoon, and also lead us to determine that a platoon formed by low performance vehicles will need a very responsive control system (high gains) in order to reduce the effectiveness of the attack on such a platoon. In general, for the platoon to be more robust when facing an attack, we need to increase the gains, since the gains are the inverse of the sum of the reaction and response time of the system (control system and the vehicles together). When designing a platoon, it is better to choose a high bandwidth control system and vehicles with the higher possible performance. That is one of the reasons why we propose to study the implementation of fractional order control on this distributed control system since we have shown that it can enhance in many cases the response and stability when compared to the integer order control design.

7.1.1 Platoon-Based Attack

The bi-directional scenario, as can be derived from the dynamic equations, has a disadvantage over the leader-predecessor. In the bi-directional architecture, vehicles obtain information equally from other vehicles that are right in front and right behind them; while the lead-predecessor architecture takes information only from the leader and vehicle right in front (predecessor).

As a result, in the bi-directional architecture, the attacker can operate from any position. And from the simulations, it was observed that the further from the leader the better, being its optimal position for the attack, the last in the platoon. On the other hand, in the leader-predecessor configuration, vehicles do not take into consideration actions of vehicles behind them, and thus the attacker can only affect vehicles placed behind it on the string. The attacker optimal position is right after the leader. This single fact would make it seem that the leader-predecessor is a more feasible solution. Still we need to recall that to achieve string stability a reliable wireless communication system is required, which is not available for now.

Another fact to consider is that the extra energy spent, as a percentage, depends on the attacker position for both architectures but also on the platoon size. The highest amount of energy expenditure obtained for a robust platoon was in the bi-directional architecture with an overspent of 140%. Still this peak value was obtained for a platoon of three vehicles while the peak value of 94% for the leader-predecessor was obtained for a platoon of size 8. In order to maximize the benefits of platooning, one would like to form platoons with size of 8 or more. In fact, on this desirable range, the bi-directional architecture does a lot better job by limiting the extra energy expended to a maximum of 80% for a size 8 and to only 35% for a size 20 platoon, while the leader-predecessor would allow an extra expenditure of 94% for a size 8 platoon to 65% for a size 20.

One positive factor to consider in the leader-predecessor architecture is that there is a null spot for the attacker which is the last position. This fact could be used as a corrective measure to avoid identified suspect of malfunctioning vehicles to affect the platoon. Regenerative braking capability proved to be able to save 35% to 50% of the energy that otherwise (dissipative braking) would be lost.

In order to create an operational platoon system that is as robust as possible to the attack we would recommend to make the size of the platoon as big as possible, use regenerative breaking capable vehicles, design the fastest possible control system and choose the bi-directional architecture to limit energy expenditure. One could also opt for the leader-predecessor configuration implementing an attacker or defective vehicle detection system, and in case of an attack, take the measure of making the attacker vehicle reposition to the last in the platoon.

7.1.2 Victim-Based Attack

When targeting just one victim inside the platoon, the leader-predecessor architecture has a great advantage. Same as before, the attacker can only attack a vehicle that is placed behind it. In the other hand, the bi-directional architecture allows the attacker to effectively affect a vehicle in any position within the platoon.

On the bi-directional case, the further the victim is from the leader, the more energy the attacker can make it spend. The optimal position for the attacker is to be right after the victim. This is, the more favorable situation for the attacker is when the victim is in position n-1, and the attacker is in position n (where n is the last vehicle in the platoon). For a 10 size platoon with this configuration, the attacker could make the victim spend up to 122% extra energy. This percentage is drastically reduced as the attacker gets close to the leader (still being the victim in position N). For instance, with the attacker in position 2, only an 8% extra energy was observed.

On the leader-predecessor architecture, the victim has to be behind the attacker in order to be affected by its maneuvers. There is a safe spot for the victim, which is position 2. Any other spot would allow the attacker to produce an extra energy expended within 80% and 110% (as long as the attacker is in front). The optimal position for the attacker could not be determined, it was depending on the victim's position, platoon size, and ranging within 3-4 vehicles in front of the victim.

Again regenerative braking capability proved to be able to save 35% to 50% percent of the energy that otherwise (dissipative braking) would be lost.

Which of the two architectures makes the attack less feasible? If the attacker can position itself before its victim in the leader-predecessor, it will make it spend 80%+ extra energy, which is a lot compared to a non-optimal position in the bi-directional architecture. Again the advantage of a safe spot for the victim, combine with a null spot for the attacker (position N), renders great flexibility in terms of taking corrective actions against an attacker or malfunctioning vehicle. Still, this could be also considered true for the bi-directional architecture with attacker in position 2 (only 8% extra energy). The recommendations for the platoon-based attack also apply to the victim-based attack. The bi-directional is a better choice even more when we consider that the leader-predecessor requires a very fast and reliable wireless communication system not yet available.

7.2 Future Work

In order to further refine this work, there is a need to develop a more realistic model of the platoons. This can be done by incorporating the dynamics of the vehicles, the effects of the alternating gaps on the drag coefficient, and adding the effects of inclined and declined roadways to the model. The capabilities of the existing and near-term platooning-like control strategies such as adaptive cruise control, cooperative adaptive cruise control, and other control architectures should be also analyzed.

In this paper, platoons formed with identical vehicles were analyzed. It is important to extend the case study to platoons formed by different types of vehicles (i.e. trucks and personal cars in the same platoon). Also, the energy expenditure was calculated using the model of a 1995 Honda Accord LX. Drive train technology has advanced a lot in therms of fuel efficiency in the last decade. With the model of an actual (2010+) vehicle, and an approximated model of a future vehicle (high performance), a better estimate of the energy expenditure in a near-term platoon could be obtained, i.e. considering that platoons will be formed by vehicles with the best technology available to begin with. Another important topic would be the analysis of the effects that the attack can produced to the wireless energy transfer technology efficiency proposed for the highways.

A great contribution to this work would be to develop a physical experimentation platform, to demonstrate the impact of the attack to the efficiency of the vehicles, and applicable control systems, especially while using and nontraditional control strategy such as fractional order control.

References

[1] Y. Luo and Y. Chen, "Stabilizing and robust fractional order pi controller synthesis for first order plus time delay systems," *Automatica*, vol. 48, pp. 2159–2167, 2012.

[2] M. Martinec and H. Zdenek, "Vehicular platooning experiments with lego mindstorms nxt," in *Proceedings of the 2011 IEEE International Conference on Control Applications (CAA)*, pp. 927–932, 2011.

[3] H. Li, Y. Luo, and Y. Chen, "A fractional order proportional and derivative (fopd) motion controller: Tuning rule and experiments," *IEEE Transactions On Control Systems Technology*, vol. 18, no. 2, pp. 516–520, Mar. 2010.

[4] K. J. Åström and R. M. Murray, "Feedback systems," 2008. [Online]. Available: http://www.cds.caltech.edu/~murray/amwiki/index.php/Main_Page

[5] C. Wang, Y. Luo, and Y. Chen, "An analytical design of fractional order proportional integral and [proportional integral] controllers for robust velocity servo," in *Proceedings of the 2009 American Control Conference (ACC09)*, pp. 1412–1417, 2009.

[6] S. Jansuwan, S. Ryu, D. Freckleton, A. Chen, and K. Heaslip, "An evaluation framework of an automated electric transportation system," in *Proceeding of the 92th Annual Meeting of the Transportation Research Board 40*, Washington, DC, 2013.

[7] K. Heaslip, "Automated electric transportation," technical report, 2012. [Online]. Available: http://www.timelab.usu.edu/htm/automated-electric-transportation

[8] K. Heaslip, K. Womack, and J. Muhs, "Automated electric transportation: A way to meet America's critical issues," *American Society of Civil Engineers: Leadership and Management in Engineering*, vol. 11, no. 1, pp. 23–28, 2011.

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

[10] X. Liu, A. Goldsmith, S. Mahal, and J. Hedrick, "Effects of communication delay on string stabiliy in vehicle platoons," in *IEEE Intelligent Transportation Systems Conference Proceedings*, pp. 625–630, Aug. 2001.

[11] P. Kavathekar and Y. Chen, "Vehicle platooning: A brief survey and categorization," in *Proceedings of The 7th ASME/IEEE International Conference on Mechatronics and Embedded Systems and Applications (MESA11)*, Aug. 2011.

[12] C. Liang and H. Peng, "String stability analysis of adaptive cruise controlled vehicles," 2000. [Online]. Available: http://www-personal.umich.edu/~hpeng/JSME2000.pdf

[13] D. Swaroop, J. K. Hedrick, C. C. Chien, and P. Iannou, "A comparison of spacing and headway control laws for automatically controlled vehicles," *Vehicle System Dynamics*, vol. 23, no. 8, pp. 597–625, Nov. 1994.

[14] P. Godefroid and S. Khurshid, “Exploring very large state spaces using genetic algorithms,” in *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 266–280, London, UK, 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=646486.694632

[15] C. R. Houck, J. Joines, and M. G. Kay, “A genetic algorithm for function optimization a matlab implementation,” 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22.4413&rep=rep1&type=pdf

[16] A. Cappiello, I. Chabini, E. Nam, A. Lue, and M. A. Zeid, “A statistical model of vehicle emissions and fuel consumption,” in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 801–809, 2002.

[17] M. Barth, F. An, T. Younglove, C. Levine, G. Scora, M. Ross, and T. Wenzel, “Development of a comprehensive modal emissions model,” Final report submitted to the National Cooperative Highway Research Program, 2000. [Online]. Available: http://www-personal.umich.edu/~hpeng/JSME2000.pdf

Appendices

Appendix A

Simulation Codes

A.1 Bi-Directional Architecture Code

Absolute Coordinates

```
%% Bidirectional Configuration
clear
N=5;        %Number of Vehicles in the platoon MAX=20
m = 1;       %Mass of the Vehicles
c=[1 1 2.1 2.7 3.3 4.2 5.1 6 7 7.7 8.6 9.2 9.8 10.4 11 11.5 12 12.5 13.1 13.9]
*15;
k=[1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3]*15;
c = c(N);    %Speed Gain
k = k(N);    %Position Gain
Pa=N;  %Attacker Position
Pv=2;  %Victime Position
Attack=Pa*2;
F=5;  %Force Levels
%% Compute matrix A
n=N*2;
A=zeros(n,n);
for u=0:2:n-1
A(1+u,2+u)=1;
end
A(2,1:4)=[-k/m -c/m k/m c/m];
A(2,1:4)=0;
for u=0:2:n-4
A(4+u,1+u)=k/m;
A(4+u,2+u)=c/m;
A(4+u,3+u)=-2*k/m;
A(4+u,4+u)=-2*c/m;
end
for u=0:2:n-5
A(4+u,5+u)=k/m;
A(4+u,6+u)=c/m;
end
A(n,n-1)= -k/m;
```

```
A(n,n)= -c/m;
clear B
%Compute matrix B
B=zeros(n,1);
B(Attack,1)=1;  %input
clear C
%Compute matrix C
C=eye(n);
%Compute matrix D
D=0;
%% poles to verify Stability
poles = eig(A); %must be negative and real to guarantee string stability
sys=ss(A,B,C,D);
%% Data analysis
T=0:1:80;
%t=6.1;
x0=zeros(1,length(B));         %Set Initial Conditions to zero
%u = 0.1./(2.*T);              %Define attacker input
%u = 54*sin(T);
%u(u>500) = 500;
%u(u<0)= 0;
%% optimisation parameters
% x(1) = T (duration of attack)
% x(2:nl+1) = l (levels of attack)
nl = 10; %number of levels to attacker behaviour
nl_r = 1; %num of levels to recover from attack
% upper and lower bounds for optimisation parameters
lb = -F*ones(1,length(T));
% upper
ub = F*ones(1,length(T));
% lower bounds for recovery
lb_r = [1 0];
% upper
ub_r = [100 5];
%% simulation parameters
n_ga = 5; %num of times to run ga optimiser
n_pos = n-1; %positions for attacker and victim
% variables we may wish to sweep (defaults)
vsep = 1.0; %vehicle separatio
%Victim = 3;
%% single attack (NOTE: for function-based version, use atk())
% expenditure phase
disp('Expenditure phase...');
options = gaoptimset('PopulationSize',10*(1+nl+2),'PopInitRange',[lb;ub],
'EliteCount',5,'SelectionFcn',@selectiontournament,'MutationFcn',
@mutationadaptfeasible,'UseParallel','always'); %one attacker
```

```
dl=10;
%[EnerG]=Ener(u,T,sys,x0,N,c,k,vsep,Pa,Pv)
EnerG=@(u)Ener(u,T,sys,x0,N,c,k,vsep,Pa,Pv);
Atk = ga(EnerG,length(T),[],[],[],[],lb,ub,[],options)
[EnerG,EnerGv,u,Speed,FR,FR1,FR2,Position,Vb,Vi,J] = EnerG(Atk);
TEner=sum(EnerGv);
VEner=EnerGv(Pv)
%% Results Plot
figure('name','Attack Results Inertal Combustion Engine','numbertitle','off')
subplot(2,2,1), plot(Atk)
title('Attack Force Behavior','FontSize',11,'Fontweight','b');
xlabel('Time(sec)');
ylabel('Disturbace');
grid;
subplot(2,2,2), plot(Position)
title('Vehicles Position Change','FontSize',11,'Fontweight','b')
xlabel('Time(sec)');
ylabel('Position (meters)');
legend('Vh1','Vh2','Vh3','Vh4','Vh5','Vh6','Vh7','Vh8','Vh9','Vh10',...
    'Vh11','Vh12','Vh13','Vh14','Vh15','Vh16','Vh17','Vh18','Vh19','Vh20')
grid;
subplot(2,2,3), plot(Speed)
title('Vehicle Extra Speed','FontSize',11,'Fontweight','b')
xlabel('Time(sec)');
ylabel('Speed (m/s)');
legend('Vh1','Vh2','Vh3','Vh4','Vh5','Vh6','Vh7','Vh8','Vh9','Vh10',...
    'Vh11','Vh12','Vh13','Vh14','Vh15','Vh16','Vh17','Vh18','Vh19','Vh20')
grid;
subplot(2,2,4), plot(FR)
title('Extra Fuel Consumed','FontSize',11,'Fontweight','b')
xlabel('Time(sec)');
ylabel('Gasoline (grams/sec)');
legend('Vh1','Vh2','Vh3','Vh4','Vh5','Vh6','Vh7','Vh8','Vh9','Vh10',...
    'Vh11','Vh12','Vh13','Vh14','Vh15','Vh16','Vh17','Vh18','Vh19','Vh20')
grid;
Ebase=(FR2*80)*(N-1)
Ratio=TEner/Ebase
% Ebase=FR2*80
% Ratio=VEner/Ebase
```

Error Coordinates

```
%Bidirectional Configuration
tic
clear
N=4;         %Number of Vehicles in the platoon MAX=20
```

```
m = 1;       %Mass of the Vehicles
c=[1 1 2.1 2.7 3.3 4.2 5.1 6 7 7.7 8.6 9.2 9.8 10.4 11 11.5 12 12.5 13.1 13.9];
k=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
c = c(N);    %Speed Gain
k = k(N);    %Position Gain
%Compute matrix A
n=N*2 - 2;
A=zeros(n,n);
for u=0:2:n-1
A(1+u,2+u)=1;
end
if n<3
   A(2,1:2)=[-2*k/m -2*c/m];
else
   A(2,1:4)=[-2*k/m -2*c/m k/m c/m];
end
for u=0:2:n-4
A(4+u,1+u)=k/m;
A(4+u,2+u)=c/m;
A(4+u,3+u)=-2*k/m;
A(4+u,4+u)=-2*c/m;
end
for u=0:2:n-5
A(4+u,5+u)=k/m;
A(4+u,6+u)=c/m;
end
clear B
%Compute matrix B
B=zeros(n,1);
B(2,1)=1;  %input
clear C
%Compute matrix C
C=eye(n);
%Compute matrix D
D=0;
%poles to verify Stability
poles = eig(A); %must be negative and real to guarantee string stability
%Observe States
[y,x,t]=step(A,B,C,D);
%plot(t,x);
xlabel('Time(sec)','fontsize',20, 'fontname','IrisUPC');
ylabel('Magnitude','fontsize',20, 'fontname','IrisUPC');
title('Error States','fontsize',26, 'fontweight','b','fontname','IrisUPC')
%legend('Z_{1,1}','Z_{1,2}','Z_{2,1}','Z_{2,2}','Z_{3,1}','Z_{3,2}',
'Z_{4,1}','Z_{4,2}','Z_{5,1}','Z_{5,2}','Z_{6,1}','Z_{6,2}','Z_{7,1}',
'Z_{7,2}','Z_{8,1}','Z_{8,2}');
```

```
grid;
%Bode Plot
sys=ss(A,B,C,D);
Lsys=tf(sys);
%figure;
%bode(sys)
%grid;
toc;
%%Linear Simulation
T=1:1:100
u=10./exp(T)
[Y,Tsim,X] = lsim(sys,u,T);
i=1:2:(N-1)*(2);
Position=X(:,i);
plot(Tsim,Position);
```

A.2 Leader-Predecessor Architecture Code

Absolute Coordinates

```
%% Unidirectional Configuration + Leader info
clear
N = 20;         %Number of Vehicles in the platoon MAX=20
m = 1;       %Mass of the Vehicles
k=[5 5 5 5 10 7 10 12 14 10 15 15 25 30 32 35 35 37 40 10];
k = 10;    %Position Gain
cd= 1;        %Comunication Additional Damping
c = 1.1*abs(2*k*m - cd^2)/(2*cd);    %Speed Gain
Pa=2;  %Attacker Position
Pv=2;  %Victime Position
Attack=Pa*2 + 1;
%% Compute Matrices
%Compute matrix A
n=N*2;
A=zeros(n,n);
%A(1,1)=-cd/m;
for u=[0:2:n-2]
A(2+u,3+u)=1;
end
A(3,1)=0;     %cd/m;
A(3,3)=-cd/m;
for u=[0:2:n-4]
A(5+u,1)=0;               %[cd/m];
A(5+u,2+u)=[k/m];
A(5+u,3+u)=[c/m];
```

```
A(5+u,4+u)=[-k/m];
A(5+u,5+u)=[-(c+cd)/m];
end
clear B
%Compute matrix B
B=zeros(n+1,1);
B(Attack,1)=1;
clear C
%Compute matrix C
C=eye(n+1);
%Compute matrix D
D=0;
%% poles to verify Stability
poles = eig(A); %must be negative and real to guarantee string stability
sys=ss(A,B,C,D);
%% Data analysis
T=0:1:80;
%t=6.1;
x0=zeros(1,length(B));        %Set Initial Conditions to zero
%u = 0.1./(2.*T);         %Define attacker input
%u = 54*sin(T);
%u(u>500) = 500;
%u(u<0)= 0;
%% optimisation parameters
% x(1) = T (duration of attack)
% x(2:nl+1) = l (levels of attack)
nl = 10; %number of levels to attacker behaviour
nl_r = 1; %num of levels to recover from attack
% upper and lower bounds for optimisation parameters
lb = [-5*ones(1,length(T))];
% upper
ub = [5*ones(1,length(T))];
% lower bounds for recovery
lb_r = [1 0];
% upper
ub_r = [100 5];
%% simulation parameters
n_ga = 5; %num of times to run ga optimiser
n_pos = n-1; %positions for attacker and victim
% variables we may wish to sweep (defaults)
vsep = 1.0; %vehicle separatio
%Victim = 3;
%% single attack (NOTE: for function-based version, use atk())
% expenditure phase
disp('Expenditure phase...');
options = gaoptimset('PopulationSize',10*(1+nl+2),'PopInitRange',[lb;ub],
'EliteCount',5,'SelectionFcn',@selectiontournament,'MutationFcn',
@mutationadaptfeasible,'UseParallel','always'); %one attacker
dl=10;
%[EnerG]=Ener(u,T,sys,x0,N,c,k,vsep,Pa,Pv)
EnerG=@(u)Ener12(u,T,sys,x0,N,c,k,vsep,Pa,Pv);
Atk = ga(EnerG,length(T),[],[],[],[],lb,ub,[],options);
[EnerG,EnerGv,u,Speed,FR,FR1,FR2,Position,Vb,Vi,J] = EnerG(Atk);
TEner=sum(EnerGv)-EnerGv(Pa);
% VEner=EnerGv(Pv)
%% Results Plot
% figure('name','Attack Results Leader Predeccesor','numbertitle','off')
% subplot(2,1,1), plot(Atk)
% title('Attack Force Behavior','FontSize',11,'Fontweight','b');
% xlabel('Time(sec)');
% ylabel('Disturbace Force');
% grid;
% subplot(2,1,2), plot(Position)
% title('Vehicles Position','FontSize',11,'Fontweight','b')
% xlabel('Time(sec)');
% ylabel('Position (meters)');
% grid;
% subplot(2,2,3), plot(Speed)
% title('Vehicle Extra Speed','FontSize',11,'Fontweight','b')
% xlabel('Time(sec)');
% ylabel('Speed (m/s)');
% grid;
% subplot(2,2,4), plot(FR)
% %title('Extra Fuel Consumed','FontSize',11,'Fontweight','b')
% xlabel('Time(sec)');
% ylabel('grams/sec');
% grid;
Ebase=(FR2*80)*(N-2)
Ratio=(TEner/Ebase)*100
% Ebase=FR2*80
% Ratio=(VEner/Ebase)*100
```

Error Coordinates

```
%Unidirectional Configuration + Leader info
clear
N=5;          %Number of Vehicles in the platoon MAX=20
m = 1;        %Mass of the Vehicles
k=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
k = k(N);       %Position Gain
cd= 0.1          %Comunication Additional Damping
c = 1.1*abs(2*k*m - cd^2)/(2*cd);    %Speed Gain
```

```
%Compute matrix A
n=N*2 - 1
A=zeros(n,n);
A(1,1)=-cd/m;
A(3,2)=-k/m;
A(3,3)=-(c+cd)/m;
for u=[0:2:n-2]
A(2+u,3+u)=1;
end
for u=[0:2:n-4]
A(5+u,2+u)=[k/m];
A(5+u,3+u)=[c/m];
A(5+u,4+u)=[-k/m];
A(5+u,5+u)=[-(c+cd)/m];
end
clear B
%Compute matrix B
B=zeros(n,1);
B(1,1)=1/m;
clear C
%Compute matrix C
C=eye(n);
%Compute matrix D
D=0;
%poles to verify Stability
poles = eig(A) %must be negative and real to guarantee string stability
%Observe States
%[y,x,t]=step(A,B,C,D);
%plot(t,x);
%xlabel('Time(sec)','fontsize',20, 'fontname','IrisUPC');
%ylabel('Magnitude','fontsize',20, 'fontname','IrisUPC');
%title('Error States','fontsize',26, 'fontweight','b','fontname','IrisUPC')
%legend('Z_{d,2}','Z_{1,1}','Z_{1,2}','Z_{2,1}','Z_{2,2}','Z_{3,1}','Z_{3,2}',
'Z_{4,1}','Z_{4,2}','Z_{5,1}','Z_{5,2}','Z_{6,1}','Z_{6,2}','Z_{7,1}','Z_{7,2}',
'Z_{8,1}','Z_{8,2}');
%grid;
%Bode Plot
sys=ss(A,B,C,D);
Lsys=tf(sys);
%figure;
%bode(sys)
%grid;
T=1:1:100
u=10./exp(T)
[Y,Tsim,X] = lsim(sys,u,T);
i=1:2:(N-1)*(2);
```

```
Position=X(:,i+1);
plot(Tsim,X);
```

A.3 Energy Function

```
function
[EnerG,EnerGv,u,Speed,FR,FR1,FR2,Position,Vb,Vi,J]
=Ener(u,T,sys,x0,N,c,k,vsep,Pa,Pv)
[Y,Tsim,X] = lsim(sys,u,T,x0);
i=1:2:N*(2);
Position=X(:,i);               %odd columns only == delete even
Speed=X(:,1+i);                %even columns only == delete odd
%% collision detection
th_c = 0.05; %collision threshold: masses any closer indicates collision
%th_b = 6;
if any(any(Position(:,2:N) + vsep - Position(:,1:N-1) < th_c))
    EnerGv = 0;
      disp('Collision detected!');
    return;
end
%% Define Coeficients
Aa=2.86e-2;
Bb=4.556e-8;
Cc=1.392e-12;
M=1361;
Vb=11;
%% Calculate Acc
Vi=Speed+Vb;
C=ones(size(Speed));
Kk=ones(size(Speed));
C=c*C;
Kk=k*Kk;
J=diff(Vi);
%J=abs(J);
J(length(Vi),:)=J(length(J),:);
%J1=diff(Vb);
%J1(length(Speed),:)=J1(length(J1),:);
J1=0;
%% Total power
Pi=Aa.*Vi + Bb.*Vi.^2 + Cc.*Vi.^3 + (M.*J.*Vi)./1000;
Pb=Aa*Vb + Bb*Vb^2 + Cc*Vb^3 ;
%% Fuel Compsumtion
afr=0.326;
bfr=2.28e-3;
yfr=0;
dfr=9.47e-7;
```

```
lfr=9.57e-2;
Afr=.3;
%Base Fuel Consumption
FR2 = afr + bfr*Vb + dfr*Vb^3 + lfr.*J1*Vb;
FR2(Pb==0)= Afr ;
FR2(Pb<0)= Afr ;
%Total Fuel Consumption
FR1= afr + bfr.*Vi + dfr.*Vi.^3 + lfr.*abs(J).*Vi;
FR1(Pi==0)= Afr;
FR1(Pi<0)= Afr;
FR=(FR1-FR2);
%Regenerative Braking
FR(FR<0)=.000001;
FR3=mean(FR);
FR3=mean(FR3(1,2:N));
FR(FR==.000001)=-(2.4*FR3)*0.5;
EnerGv=trapz(FR);
EnerG=1/(sum(EnerGv));
EnerG1=1/(EnerGv(Pv+1));
end
```