ENPM – 667

CONTROL OF ROBOTIC SYSTEMS

PROJECT – 1 TECHNICAL REPORT

ON

LONGITUDINAL CONTROL OF AUTOMATED CHVs WITH SIGNIFICANT ACTUATOR DELAYS

Kumara Ritvik Oruganti (117368963)

Venkata Sairam Polina (118436579)

Date: 11/25/2021

Contents:

List of Figures:

ABSTRACT

This technical paper report on "Longitudinal Control of Automated CHVs With Significant Actuator Delays" discusses about the problem of longitudinal control in Commercial heavy vehicles (CHVs) where the presence of significant delays in the fuel and brake actuators leads to degraded performance. The main objective of this report is to discuss about three main autonomous controller schemes namely adaptive PIQ scheme with variable time headway and variable gain, backstepping-based nonlinear scheme with elaboration on Lyapunov function and prediction using MPC control block and PID-based nonlinear scheme with variable time headway and variable gain with and without actuator delays.

CHAPTER - 1

INTRODUCTION

Advanced Vehicle Control and Safety Systems (AVCSS) are an integral part of the rapidly growing world-wide initiatives on intelligent transportation systems (ITS) and automated highway systems (AHS). One of such automation scenarios is platooning, [2], [3], [4], [6], in which vehicles travel at highway speeds in fully automated and tightly spaced groups.

As described in the paper in the first stage of AHS deployment vehicles would have only longitudinal control capabilities for vehicle following without intervehicle communication, with the driver assuming responsibility for steering and emergency situations. In that respect, systems currently in various stages of research and development are classified into three categories such as Autonomous systems, Cooperative systems, Automated highway systems.

The problem of slow brake response is prevalent in the commercial trucking industry, which lead to CHV manufacturers beginning to equip their vehicles with brake-by-wire systems, commonly referred to as electronic braking systems (EBS), which significantly reduce brake actuator delays. The authors also mentioned that simulations where delays were assumed to be small, our adaptive nonlinear controllers and nonlinear spacing policies demonstrated robust behavior in demanding merge-and-brake interplatoon maneuvers. They also highlighted those nonlinear spacing policies which proved so beneficial in vehicles with negligible actuator delays are not able to cope with the effects of large delays.

The problem of dealing with actuator delays has already been addressed in the literature. Huang and Ren [4] provided an upper bound for the time delay and derived conditions under which the errors attenuate as they propagate upstream through the platoon.

It is stressed that the design of back stepping controller that it is not based on a model which explicitly contains such delays; instead, it is based on a delay-free model that is more accurate in its representation of the vehicle dynamics. The resulting control law, while not explicitly accounting for the delays, is robust enough to mitigate their effect. The original paper talked about the traditional approach which is used for improving performance of systems with known delays and available plant models is by using prediction. While beneficial to the performance of a single vehicle, a predictive approach was not expected to be able to compensate for the cumulative effect of the delay in a platoon under autonomous operation. Nevertheless, the inclusion of an aggressive predictor in the control loop improves both the platoon performance and the control smoothness.

This report starts with a PIQ controller which is modified to adaptive PIQ, which is further enhanced by modelling the headway as a function of relative velocity and variable gain. Next controller is designed using backstepping procedure, where a new control law is proposed which demonstrates significantly improved performance in the presence of large actuator delays. In this report, we have discussed about the backstepping procedure and choice of the Lyapunov function for making the derivative negative semi definite. In this report, we are not simulating the backstepping controller, but a deep discussion was provided. The performance of the backstepping-based nonlinear controller with the predictor is good according to the researchers but comes at the cost of increase in controller complexity. This scenario led to development of a significantly simpler PID-like controller whose performance is better than the backstepping controller without the prediction according to the simulated results in the paper. This controller is not an actual PID controller, but uses the concept of PID control, because it uses the nonlinear spacing terms of initially designed PIQ controller and has thus the same nonlinear complexity. Note that we are using a single vehicle as a follower for all the simulations and results. The parameters were not mentioned in the research paper and hence, we have assumed the parameters for simulations and simulated the same using matlab.

CHAPTER - 2
PIQ CONTROLLER

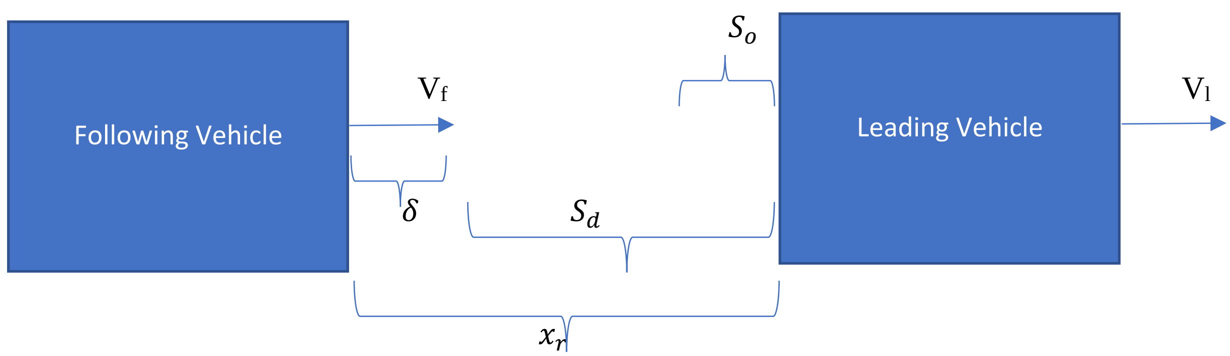

Figure 1: Description of a two-vehicle platoon with one leading vehicle and one following vehicle

2.1 Description:

PIQ scheme was introduced to overcome the cons of PID controller. Initially, a PID controller with an implementable approximate derivative term and an anti-windup scheme was used to reduce speed overshoot in the paper. In speed tracking, overshoot is much more undesirable than undershoot because of passenger comfort considerations. It is even more undesirable in a vehicle-following scenario, since it may lead to collisions. Therefore, a signed quadratic (Q) term is added to the PI controller. This nonlinear term allows fast compensation of large tracking errors without the need for high gains. As a result, PIQ controller significantly reduces the overshoot associated with high-gain control.

The control objective is to minimize the spacing error and make the relative velocity to zero [7]. In the platoon scenario, the controller at the following vehicle has the information of the leading vehicle velocity and the separation between them. Hence the relative velocity is given from Figure 1:

$$V_r = V_l - V_f \tag{1}$$

If the desired spacing between the two vehicles is constant, it is called as the fixed spacing policy. The relation between the desired spacing and the vehicle velocity is given by:

$$S_d = S_o + hV_f \tag{2}$$

If we take only spacing as the control objective, which is a function of the leading vehicle velocity, there is a possibility for collision as it depends on the initial velocity as we don't have the intervehicle communication to update the vehicle velocity on the go. Hence the desired spacing should be involved in the control object to minimize the possibility of collisions.

The separation error is given by:

$$\delta = x_r - S_d = x_r - S_o - hV_f \tag{3}$$

To minimize the relative velocity and the separation error, the control objective can be modelled as

$$\mathrm{V_r} + \mathrm{k}\delta = 0 \ (\mathrm{k} > 0) \tag{4}$$

Case-1:

When two of the vehicles are closer than desired, separation error is negative, which means the relative velocity is positive, $V_l > V_f$ and controller need not take drastic decision.

Case-2:

When two of the vehicles are apart than desired, separation error is positive, which means the relative velocity is negative, $x_r > S_d$ and $V_r < 0$. Follower should increase speed. The selection of k influences the response of the controller and can be changed depending on requirements. In [7], if k is nonlinear function of δ, performance increases significantly.

We now, derive and show that when relative velocity is zero, the separation error also becomes zero analytically.

2.2 Derivation:

Taking lead vehicle velocity V_l = const $\Longrightarrow \dot{V_l} = 0$

$$\dot{\delta} = \frac{d}{dt}\left(x_r - S_o - hV_f\right) = \mathrm{V_r} - h\dot{V_f} \quad \left[\because \frac{d}{dt}(x_r) = V_r\right] \tag{5}$$

We know that

$$\mathrm{V_r} = \mathrm{V_l} - \mathrm{V_f} \Longrightarrow \dot{V_r} = -\dot{V_f} \tag{6}$$

If $\mathrm{V_r} + \mathrm{k}\delta = 0 \Longrightarrow \dot{V_r} + \mathrm{k}\dot{\delta} = 0$

$$\dot{V_r} + \mathrm{k}\,(\mathrm{V_r} - h\dot{V_f}) \Longrightarrow \dot{V_r} + \mathrm{k}\,(\mathrm{V_r} + h\dot{V_r}) \Longrightarrow (1 + \mathrm{kh})\,\dot{V_r} + \mathrm{k}\,\mathrm{V_r} = 0 \tag{7}$$

Which shows that $\mathrm{V_r} \rightarrow 0$ *and* $\delta \rightarrow 0$

2.3 Linearizing the control objective:

As the system is nonlinear, we will now linearize the model by using the transfer function as:

$$\frac{\delta(\mathrm{V_r} + \mathrm{k}\delta)}{\delta\,\mathrm{u}} = \frac{-\mathrm{b}}{\mathrm{s+a}}$$

$$S\delta(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{a}\delta(\mathrm{V_r} + \mathrm{k}\delta) = -\mathrm{b}\delta\mathrm{u} \tag{8}$$

$$S\delta(\mathrm{V_r} + \mathrm{k}\delta) = -\mathrm{a}\delta(\mathrm{V_r} + \mathrm{k}\delta) - \mathrm{b}\delta\mathrm{u} \tag{9}$$

When steady state is achieved $\delta = const$

$$(\mathrm{V_r} + \mathrm{k}\delta) = -\mathrm{a}\delta(\mathrm{V_r} + \mathrm{k}\delta) - \mathrm{b}\delta\mathrm{u} \tag{10}$$

$$\left(\dot{\mathrm{V}}_\mathrm{r} + \mathrm{k}\dot{\delta}\right) = -\mathrm{a}(\mathrm{V_r} + \mathrm{k}\delta) - \mathrm{bu} \tag{11}$$

$$-\dot{V_f} \;\; = -\mathrm{a}\;(\mathrm{V_r} + \mathrm{k}\delta) - \mathrm{bu}\;\left(\because\;\dot{\delta} = 0\right) \tag{12}$$

$$\dot{V_f} \;\; = \; \mathrm{a}(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{bu} + \bar{\mathrm{d}} \tag{13}$$

Where $\bar{\mathrm{d}}$ is the disturbance.

2.4 Adaptive PIQ Controller design:

Most of the results available on the longitudinal (vehicle velocity) control of the automatic vehicles have separate controllers from the fuel and the brake. However, our adaptive control takes care of both the fuel and brake commands, in turn it makes the controller design less complex. The magnitude of the output of designed controller is negative, it means that we need to apply brakes. If the output is positive, we will issue the fuel command [7].

General PIQ control input is given by:

$$u = \hat{k}_p \mathrm{e_r} + \hat{\mathrm{k}}_\mathrm{i} + \hat{k}_q e_r |\mathrm{e_r}| \tag{14}$$

where,

e_r is the error.

$\hat{k}_p$ is the proportional gain

$\hat{k}_i$ is the integral gain

$\hat{k}_q$ is the quadratic gain

Taking the above linearized model of the control objective, the PIQ control law is proposed.

$$u = \hat{k}_p(\mathrm{V_r} + \mathrm{k}\delta) + \hat{\mathrm{k}}_\mathrm{i} + \hat{k}_q(\mathrm{V_r} + \mathrm{k}\delta)|\mathrm{V_r} + \mathrm{k}\delta| \tag{15}$$

$$\dot{V_f} \;\; = \; \mathrm{a}(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{bu} + \bar{\mathrm{d}} \tag{16}$$

Where,

$\bar{\mathrm{d}}$ is the external disturbance.

Substituting 15 in 16 we get

$$\dot{V_f} \;\; = \; \left(\mathrm{a} + \mathrm{bk_p}\right)(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{bk_i} + \mathrm{bk_q}(\mathrm{V_r} + \mathrm{k}\delta)|\,\mathrm{V_r} + \mathrm{k}\delta| + \bar{\mathrm{d}} \tag{17}$$

Reference equations:

$$\dot{V} = -(a + bk_1)e_v - bk_2 - bk_3 e_v|e_v| + d \tag{18}$$

$$\dot{V}_m = -a_m(V_m - V_d) - q_m(V_m - V_d)|e_v| \quad (19)$$

Where,

V_m is the desired velocity

Comparing the equations,

$$\begin{aligned} a + bk_1 &= a_m \\ bk_2 &= d \\ bk_3 &= q_m \end{aligned} \quad (20)$$

Where k_1, k_2, k_3 *are* time varying parameters which are adjusted by control law.

$$\dot{e}_r = \dot{V} - \dot{V}_m \quad (21)$$

$$V_m = a_m(V_l - V_m + k\delta) + q_m(V_l - V_m + k\delta)|\,\mathrm{V_r} + \mathrm{k}\delta| \quad (22)$$

$$\begin{aligned} -bk_i &= d \\ bk_q &= q_m \\ a + bk_p &= a_m \end{aligned} \quad (23)$$

Tracking error

$$e_r = V_f - V_m \quad (24)$$

Taking derivative of the tracking error,

$$\dot{e}_r = \dot{V}_f - \dot{V}_m = -a_m e_r - q_m e_r|\,\mathrm{V_r} + \mathrm{k}\delta| + b\left[\,k_p(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{k_i} + \mathrm{k_q}(\mathrm{V_r} + \mathrm{k}\delta)\left|\mathrm{V_r} + \mathrm{k}\delta\right|\,\right] \quad (25)$$

Instead of using the absolute tracking error, we can use the estimate of the error to improve the robustness of the adaptive controller.

Estimate of error

$$\epsilon = e_r - Z \quad (26)$$

where Z is determined from

$$\dot{Z} = -a_m Z - q_m\,|\,\mathrm{V_r} + \mathrm{k}\delta|\,Z + \epsilon\lambda\,(V_r^2 + \delta^2) \quad (27)$$

as proposed by Xu and Ioannou in the Adaptive throttle control for speed tracking research paper.

Where,

λ is a very small number.

Taking derivative of the error estimate,

$$\dot{\epsilon} = \dot{e_r} - \dot{z} = -a_m\,(e_r - Z) - q_m(e_r - Z)|\,\mathrm{V_r} + \mathrm{k}\delta| + b[\,k_p(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{k_i} + \mathrm{k_q}(\mathrm{V_r} + \mathrm{k}\delta)|\,\mathrm{V_r} + \mathrm{k}\delta|] - \epsilon\lambda\,(V_r^2 + \delta^2) \quad (28)$$

$$K_p = -\gamma_1 \epsilon (V_r + k\delta)$$
$$K_i = -\gamma_2 \ \epsilon \tag{29}$$
$$K_q = -\gamma_3 \ \epsilon \ (\mathrm{V_r} + \mathrm{k}\delta)|\ \mathrm{V_r} + \mathrm{k}\delta|$$

The choice of K_p, K_i, K_q are made from the Lyapunov function given by:

$$V \ = \ Va + \frac{z^2}{2} \tag{30}$$

Where,

z is the error.

Let us choose

$$V_a = \ \frac{bk_p^2}{2\gamma_1} + \frac{bk_i^2}{\gamma_2} + \frac{bk_q^2}{2\gamma_3} \tag{31}$$

Then,

$$V = \frac{\epsilon^2}{2} + \frac{bk_p^2}{2\gamma_1} + \frac{bk_i^2}{\gamma_2} + \frac{bk_q^2}{2\gamma_3} \tag{32}$$

2.5 Stability check for the Lyapunov function:

Let us take the derivative of the function and substituting the equations of ϵ and $\dot{\epsilon}$, we get,

$$\dot{V} = -a_m \epsilon^2 - q_m \epsilon^2 |e_V| - \epsilon^2 e_r^2 \tag{33}$$

$$\dot{V} = -a_m \epsilon^2 - q_m |V_r + k\delta| \epsilon^2 - \lambda (V_r^2 + \delta^2) \epsilon^2 < \ 0 \tag{34}$$

As $\mathrm{V_r}$, $\delta \rightarrow 0$ K_1 , K_2, K_3 are bounded and hence the gains chosen are valid and the system is stable.

Rewriting the above equations,

$$\dot{V_f} = a(v_r + k\delta) + bu + \overline{d} \tag{35}$$

$$u(t) = k_p (V_r + k_\delta) + k_i + K_q (V_r + k\delta)|V_r + k\delta| \tag{36}$$

$$\dot{\epsilon} = -a_m \in -q_m \in |v_r + k\delta| + bu - \in \lambda (v_r^2 + \delta^2) \tag{37}$$

Substituting (36) in (35) we get,

$$\dot{V_f} \ \ = \left(\mathrm{a} + \mathrm{bk_p}\right)(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{bk_i} + \mathrm{bk_q}(\mathrm{V_r} + \mathrm{k}\delta)|\ \mathrm{V_r} + \mathrm{k}\delta| + \overline{\mathrm{d}} \tag{38}$$

$$\dot{e}_r = \ -a_m e_r - q_m e_r |\ \mathrm{V_r} + \mathrm{k}\delta| + b\left[\ k_p(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{k_i} + \mathrm{k_q}(\mathrm{V_r} + \mathrm{k}\delta)\left|\mathrm{V_r} + \mathrm{k}\delta\right|\ \right] \tag{39}$$

$$\dot{Z} = -a_m Z - q_m \,|\ \mathrm{V_r} + \mathrm{k}\delta|\ Z + \epsilon\lambda\ (V_r^2 + \delta^2) \tag{40}$$

The state-space form of the above equations with the chosen states $\mathrm{V_r}$ and $\in$ is given as

$$\begin{bmatrix}\dot{V_f}\\ \dot{\epsilon}\end{bmatrix} = \begin{bmatrix}a + bk_p|V_r + k\delta| & 0\\ bk_p|V_r + k\delta| & -a_m - q_m|V_r + k\delta| + \lambda\,(V_r^2 + \delta^2)\end{bmatrix}\begin{bmatrix}V_r + k\delta\\ \epsilon\end{bmatrix} + \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}b\,k_i\\ d\end{bmatrix}$$

Where

$$A = \begin{bmatrix}a + bk_p|V_r + k\delta| & 0\\ bk_p|V_r + k\delta| & -a_m - q_m|V_r + k\delta| + \lambda\,(V_r^2 + \delta^2)\end{bmatrix} \qquad X = \begin{bmatrix}V_r + k\delta\\ \epsilon\end{bmatrix}$$

$$\dot{X} = \begin{bmatrix}\dot{V_r} + k\dot{\delta}\\ \dot{\epsilon}\end{bmatrix} \qquad BU = \begin{bmatrix}bk_i & d\\ bk_i & o\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix}$$

The generalized state space form with the estimate and the acceleration of the following vehicle can be obtained by taking the general forms of $\dot{V_f}$ and $\dot{\epsilon}$ (without substituting the equation of u).

$$\dot{V_f} \;=\; \mathrm{a(V_r + k\delta) + bu + \bar{d}} \tag{41}$$

$$\dot{\epsilon} = -a_m \in -q_m \in |v_r + k\delta| + bu - \in \lambda(v_r^2 + \delta^2) \tag{42}$$

The state-space model becomes,

$$\begin{bmatrix}\dot{V_f}\\ \dot{\in}\end{bmatrix} = \begin{bmatrix}-q & 0\\ 0 & -a_m - q_m|V_r + k\delta| - \lambda(V_r^2 + \delta^2)\end{bmatrix}\begin{bmatrix}V_r + k\delta\\ \epsilon\end{bmatrix} + b\,u\,\begin{bmatrix}1\\ 1\end{bmatrix}$$

The adaptive PIQ controller can operate autonomously using this speed dependent spacing policy. However, the fixed time headway *h* should be large for passenger cars to improve the performance [7]. Hence, a variable headway yields small separation errors without increased intervehicle spacing. This variable headway can also work even in the scenario of intervehicle communication. This gives non-linear spacing policy. The variable time headway is a function of the relative velocity. This intuition is presented by D. Yanakiev and I. Kanellakopoulos in their "variable time headway for string stability of automated heavy-duty vehicles"[8] paper[7]. It is given by

$$h \;=\; h_0 - \; c_h v_r \tag{43}$$

where,

h_0 and c_h are positive and constants.

Logically, if the relative speed between the two vehicles is positive, i.e, if the leading vehicle is moving faster, then, the headway can be reduced. If it is moving slower, the headway can be reduced and it results in reduction of errors and smoother control. Another modification that can be done is to introduce variable separation gain *k* given by

$$k = c_k + (k_0 - c_k)\, e^{(-\delta\hat{}2)\,*\sigma} \tag{44}$$

where

$0 < c_k < k_0$ *and* $\sigma \geq 0$.

This design modification is to reduce the undesirable aggressive reaction of the controller for large separation with high acceleration. The gain also gets reduced even when the separation error is negative. This is due to the lower actuation to weight ratio of the commercial heavy vehicles which limits their accelerations and decelerations. This has an advantage for the following vehicles in the platoon as they need not move with higher accelerations/decelerations as the first few will take care of them and this headway for the other vehicles smoothens the action. Remember that this is obtained from the decentralization of the controllers for the vehicles [7].

In the simulations, we have assumed that there are no disturbances.

2.6 Codes for Adaptive-PIQ Controller:

1. **Constant time headway (fixed spacing policy) with actuator delay**

```
%Adaptive PIQ controller code for 0.2s actuator delay

clc;

clear all;

close all;

sl = 30; %initial Position of lead car

sf = 0; %initial position of following car

vf = 60; %initial velocity of following vehicle

vl = 40; %leading vehicle velocity

s0 = 5; %minimum distance required

h = 0.2; %fixed time headway

sd = s0+h*vf; %desired distance sd = so+hvf

xr = sl - sf; %initial relative distance

delta = xr-sd; %spacing error

vr = vl-vf; %following - leading velocities

vf_acc_list=[];

kp = 3;

ki = 0.5;

kq = 0.01;

a = 0.1;

b = 0.2;
```

```
figure
i=1;
for t = 0:0.1:50
    u = kp*(vr+0.3*delta) + ki + kq*(vr+0.3*delta)*abs(vr+0.3*delta); %u=kp(vr+k*delta) + ki + kq(vr+k*delta)|vr+k*delta|
    vf_acc = a*(vr+0.3*delta)+b*u;
    vf_acc_list(end+1) = vf_acc;
    %comment the if condition (29-32 lines) to get simulation with no
    %actuator delay and uncomment line 33
    if(t>0.1)
        vf = vf+(vf_acc_list(i)*0.1);
        i=i+1;
    end
    %vf = vf+vf_acc*0.1;
    subplot(2,2,1)
    plot(t,vf,'.','Color','red');
    xlabel('time')
    ylabel('m/s')
    title('Following Vehicle Velocity')
    xlim([0 50])
    ylim([0 60])
    hold on
    drawnow
    subplot(2,2,2)
    plot(t,delta,'.','Color','blue');
    xlabel('time')
    ylabel('m')
    title('Separation Error')
    xlim([0 50])
```

```
ylim([-30 30])
hold on
drawnow
subplot(2,2,3)
plot(t,vf_acc,'.','Color','black');
xlabel('time')
ylabel('m/s2')
title('Following Vehicle Acceleration')
xlim([0 50])
ylim([-30 30])
hold on
drawnow
subplot(2,2,4)
plot(t,xr,'.','Color','m');
xlabel('time')
ylabel('m')
title('Vehicle Separation')
xlim([0 50])
ylim([-30 60])
hold on
drawnow
sd = s0+h*vf;
if(t>2 && t<4.1)%accelerating lead vehicle with 2m/s^2 for 2 seconds
   vl = vl+2*0.1;
end
vr = vl-vf;
sl = sl+vl*0.1;
sf = sf + vf*0.1;
```

```
    xr = sl-sf;
    delta = xr-sd;
    xlim([0 50])
    ylim([0 60])
end
```

2. Variable time headway (nonlinear spacing policy) with variable separation gain, variable time headway, and actuator delay

```
%Adaptive PIQ controller code with variable separation error gain(k) and
%0.2s actuator delay
clc;
clear all;
close all;
sl = 30; %initial Position of lead car
sf = 0; %initial position of following car
vf = 60; %initial velocity of following vehicle
vl = 40; %leading vehicle velocity
s0 = 5; %minimum distance required
h = 0.2; %fixed time headway
sd = s0+h*vf; %desired distance sd = so+hvf
xr = sl - sf; %initial relative distance
delta = xr-sd; %spacing error
vr = vl-vf; %following - leading velocities
vf_acc_list=[];
kp = 1;
ki = 0.3;
kq = 0.01;
a = 0.1;
b = 0.2;
figure
```

```
ax = axes();
i=1;
for t = 0:0.1:50
    k=0.1+(1-0.1)*exp(-0.1*delta*delta);
    h = 0.1-0.2*vr; %for variable time headway - comment this line for
    u = kp*(vr+k*delta) + ki + kq*(vr+k*delta)*abs(vr+k*delta); %u=kp(vr+k*delta) + ki +
kq(vr+k*delta)|vr+k*delta|
    vf_acc = a*(vr+k*delta)+b*u;
    vf_acc_list(end+1) = vf_acc;
    %comment the if condition (32-35 lines) to get simulation with no
    %actuator delay and uncomment line 36
    if(t>0.1)
        vf = vf+(vf_acc_list(i)*0.1);
        i=i+1;
    end
    %vf = vf+vf_acc*0.1;
    subplot(2,2,1)
    plot(t,vf,'.','Color','red');
    xlabel('time')
    ylabel('m/s')
    title('Following Vehicle Velocity')
    xlim([0 50])
    ylim([0 60])
    hold on
    drawnow
    subplot(2,2,2)
    plot(t,delta,'.','Color','blue');
    xlabel('time')
    ylabel('m')
```

```
title('Separation Error')
xlim([0 50])
ylim([-30 30])
hold on
drawnow
subplot(2,2,3)
plot(t,vf_acc,'.','Color','black');
xlabel('time')
ylabel('m/s2')
title('Following Vehicle Acceleration')
xlim([0 50])
ylim([-30 30])
hold on
drawnow
subplot(2,2,4)
plot(t,xr,'.','Color','m');
xlabel('time')
ylabel('m')
title('Vehicle Separation')
xlim([0 50])
ylim([-30 60])
hold on
drawnow
sd = s0+h*vf;
if(t>2 && t<4.1)%accelerating lead vehicle with 2m/s^2 for 2 seconds
   vl = vl+2*0.1;
end
vr = vl-vf;
```

```
    sl = sl+vl*0.1;
    sf = sf + vf*0.1;
    xr = sl-sf;
    delta = xr-sd;
    xlim([0 50])
    ylim([0 80])
end
```

3. Constant time headway with fixed separation gain and actuator delay

```
%Adaptive PIQ controller code for 0.2s actuator delay
clc;
clear all;
close all;
sl = 30; %initial Position of lead car
sf = 0; %initial position of following car
vf = 60; %initial velocity of following vehicle
vl = 40; %leading vehicle velocity
s0 = 5; %minimum distance required
h = 0.2; %fixed time headway
sd = s0+h*vf; %desired distance sd = so+hvf
xr = sl - sf; %initial relative distance
delta = xr-sd; %spacing error
vr = vl-vf; %following - leading velocities
vf_acc_list=[];
kp = 3;
ki = 0.5;
kq = 0.01;
a = 0.1;
b = 0.2;
```

```
figure
i=1;
for t = 0:0.1:50
    u = kp*(vr+0.3*delta) + ki + kq*(vr+0.3*delta)*abs(vr+0.3*delta); %u=kp(vr+k*delta) + ki + kq(vr+k*delta)|vr+k*delta|
    vf_acc = a*(vr+0.3*delta)+b*u;
    vf_acc_list(end+1) = vf_acc;
    %comment the if condition (29-32 lines) to get simulation with no
    %actuator delay and uncomment line 33
    if(t>0.1)
        vf = vf+(vf_acc_list(i)*0.1);
        i=i+1;
    end
%     vf = vf+vf_acc*0.1;
    subplot(2,2,1)
    plot(t,vf,'.','Color','red');
    xlabel('time')
    ylabel('meters/s')
    title('Following Vehicle Velocity')
    xlim([0 50])
    ylim([0 60])
    hold on
    drawnow
    subplot(2,2,2)
    plot(t,delta,'.','Color','blue');
    xlabel('time')
    ylabel('meters')
    title('Separation Error')
    xlim([0 50])
```

```
ylim([-30 30])
hold on
drawnow
subplot(2,2,3)
plot(t,vf_acc,'.','Color','black');
xlabel('time')
ylabel('meters/s2')
title('Following Vehicle Acceleration')
xlim([0 50])
ylim([-30 30])
hold on
drawnow
subplot(2,2,4)
plot(t,xr,'.','Color','m');
xlabel('time')
ylabel('meters')
title('Vehicle Separation')
xlim([0 50])
ylim([-30 60])
hold on
drawnow
sd = s0+h*vf;
if(t>2 && t<4.1)%accelerating lead vehicle with 2m/s^2 for 2 seconds
   vl = vl+2*0.1;
end
vr = vl-vf;
sl = sl+vl*0.1;
sf = sf + vf*0.1;
```

```
    xr = sl-sf;
    delta = xr-sd;
end
```

CHAPTER - 3

BACKSTEPPING CONTROLLER

3.1 Description:

The Adaptive PIQ controller gives very good performance when we assume that the control input is immediately present, and the vehicles are equipped with electronic braking systems (EBS). Hence the possibility is to introduce the actuator dynamics [7].

The backstepping controller takes care of the presence of actuator dynamics, driving/braking torque as the input and we remodel the first-order equation which introduces new variable T where, T is the braking/driving torque. It also includes the aerodynamic drag and $\bar{d}$ takes care of the other disturbances.

As mentioned in the introduction, in this report, we have discussed about the design of backstepping controller, choice Lyapunov function for the boundedness, derivation of generalized equations for integrator backstepping controller and thereby designing the new control law for our system model. Note that, the parameters for backstepping were not given in the research paper and the parameters for predictive design were not specified. Hence, we are using the matlab MPC block for adaptive cruise control using model predictive control to check the simulations using the prediction.

3.2 System Model:

The proposed new model is:

$$\dot{V}_f = a(V_r + k\delta) + bT - \frac{c_a}{m}V_f^2 + d \tag{45}$$

$$\dot{T} = -a_1 T + a_1 u \tag{46}$$

where,

c_a is the aerodynamic drag coefficient,

m is the mass of the vehicle

There are two steps to control the $V_r + k\delta$ with T as an input.

1. Let us consider

$$\dot{V}_f = a(V_r + k\delta) + bT - \frac{c_a}{m}V_f^2 + d \tag{47}$$

and determine the desired torque T_d such that we regulate the velocity assuming that T as the actual input. Using this equation, we model our original u so that the T converges to the desired torque T_d i.e $T - T_d \rightarrow 0$.

3.3 Derivation of generalized form for backstepping controller:

The general procedure for backstepping is explained below [10][11].

$$\dot{x} = f(x) + g(x)\xi \tag{48}$$

$$\dot{\xi} = u = input \tag{49}$$

Adding and subtracting $g(x)\phi(x)$ to $\dot{x}$, where $\phi(x)$ is the desired input,

$$\dot{x} = f(x) + g(x)\xi + g(x)\phi(x) - g(x)\phi(x) \tag{50}$$

$$\dot{x} = f(x) + g(x)\phi(x) + g(x)[\xi - \phi(x)] \tag{51}$$

Let $z = \xi - \phi(x)$.

Then,

$$\dot{z} = \dot{\xi} - \dot{\emptyset}(x)\left[\text{ because } \dot{\xi} = u = input\right] \tag{52}$$

$$\dot{z} = u - \dot{\emptyset}(x) \tag{53}$$

This is backstepping because $\phi(x)$ is stepped back by differentiation

$$\phi = \frac{\partial\phi}{\partial x}\dot{x} = \frac{\partial\phi}{\partial x}[f(x) + g(x)\xi] \tag{54}$$

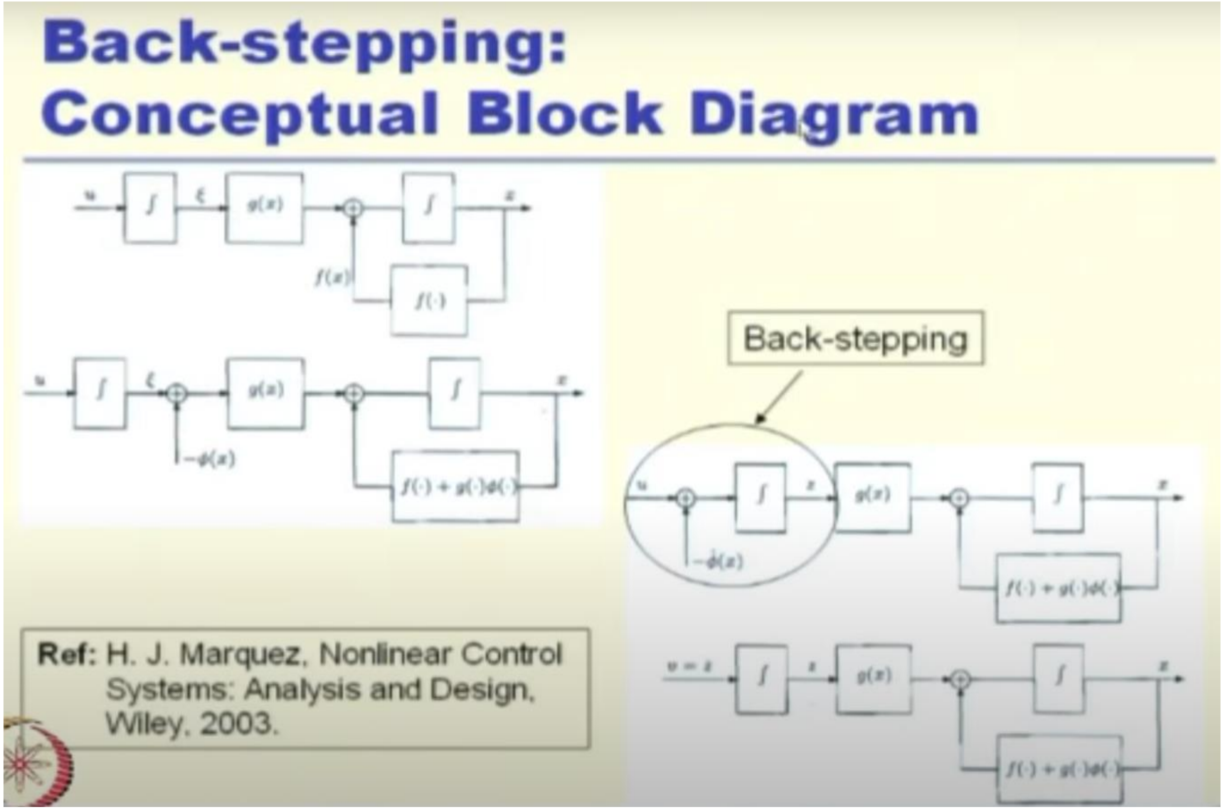

Figure 2: NPTEL Lecture on Advanced Control System Design by Prof Radhakant Padhi, IISc Banglore

Let Lyapunov function

$$v(x,z) = v_1(x) + \frac{1}{2}z^2 \tag{55}$$

where,

$v_1(x)$ is a non-negative polynomial function of x.

The general equation of $v_1(x)$ can be taken as $\frac{1}{2}x^2$ and z is the error [1]

$$\dot{v} = \frac{\partial v_1}{\partial x}\dot{x} + z\dot{z} \tag{56}$$

$$\dot{v} = \left(\frac{\partial v_1}{\partial x}\right)[f(x) + g(x)\phi(x) + g(x)z] + z\dot{z} \le -\mathrm{Va}(\mathrm{X}) = -\mathrm{Q} \tag{57}$$

which is negative definite.

Where,

$$\mathrm{V_a}(\mathrm{X}) = \left(\frac{\partial v_1}{\partial x}\right)[f(x) + g(x)\phi(x)] \tag{58}$$

Now, writing the equation of $\dot{v}$,

$$\dot{v} = -v_a(x) + \left[\frac{\partial v_1}{\partial x}g(x) + \dot{z}\right]z \tag{59}$$

We know that,

$$\dot{z} = \frac{-\partial v_1}{\partial x}g(x) - kz \tag{60}$$

Hence,

$$\dot{v} \le -v_a(x) - kz^2 < 0 \tag{61}$$

The above derivative of v is negative definite and hence the chosen Lyapunov function is valid.

Designing the control input:

Let us go back and take the equation of $\dot{z}$

$$\dot{z} = u - \dot{\phi} = -\frac{\partial v_1}{\partial x}g(x) - kz \tag{62}$$

Taking the $\dot{\phi}$ to the right side of the equation,

$$u = \dot{\phi} - \frac{\partial v_1}{\partial x}g(x) - kz \tag{63}$$

But we know that

$$\dot{\phi} = \frac{\partial \phi}{\partial x}[\dot{x}] \text{ and } \dot{x} = f(x) + g(x)\xi \tag{64}$$

Substituting $\dot{x}$ in $\dot{\phi}$ we get,

$$\dot{\phi} = \frac{\partial \phi}{\partial x}[f(x) + g(x)\xi] \tag{65}$$

Now, substituting the $\dot{\phi}$ in right hand side of u

$$u = \frac{\partial \phi}{\partial x}[f(x) + g(x)\xi] - \frac{\partial v_1}{\partial x} g(x) - kz \tag{66}$$

$$u = \frac{\partial \phi}{\partial x}[f(x) + g(x)\xi] - \frac{\partial v_1}{\partial x} g(x) - k[\xi - \emptyset(x)] \tag{67}$$

Since we are unsure of the boundedness of the above equation, we must design the $\emptyset(x)$ such that the control input is negative definite. This can be designed based on the actual control input of the system instead of generalizing the function.

Now, let us take our system model equations,

$$\dot{V}_f = a(V_r + k\delta) + bT - \frac{c_a}{m} V_f^2 + d \tag{68}$$

$$\dot{T} = -a_1 T + a_1 u \tag{69}$$

Let us take the desired torque as,

$$T_d = u_{orig} + \frac{C_a}{bm}\, v_f^2 \tag{70}$$

So that, if we substitute in the new equation of $\dot{V}_f$, we get our original equation of $\dot{V}_f$ (the equation from Adaptive PIQ control)

$$u_{orig} = k_p(\mathrm{V_r} + \mathrm{k\delta}) + \mathrm{k_i} + \mathrm{k_q}(\mathrm{V_r} + \mathrm{k\delta})|\, \mathrm{V_r} + \mathrm{k\delta}| \tag{71}$$

Derivating the T_d,

$$\dot{T}_d = \dot{u}_{orig} + 2\frac{C_a}{bm} v_f \dot{v}_f \tag{72}$$

Now subtracting the above equation from $\dot{T}$,

$$\dot{T} - \dot{T}_d = -a_1 T + a_1 u - \dot{u}_{orig} - 2\frac{C_a}{bm} v_f \dot{v}_f \tag{73}$$

If we compare the equations from the generalized backstepping control, the design of $\emptyset(x)$ is such that the control input is negative definite. Hence to ensure that $T - \dot{T}_d \rightarrow 0$ the updated law Lyapunov function is given by,

$$v_a = \frac{1}{2}x_1^2 + \frac{1}{2\gamma}\,(T - T_d)^2 = V + \frac{1}{2\gamma}(T - T_d)^2 \tag{74}$$

Where,

γ is a design constant.

Let us now take the derivative of the above Lyapunov function. We get,

$$v_a = \dot{V} + \frac{1}{\gamma}(T - T_d)(\dot{T} - \dot{T_d}) \tag{75}$$

Let us now complete the squares with $C_1(T_d - T)$ and $C_1 = a_1\beta$ and $\beta = design\ constant$ with the RHS of $(\dot{T} - \dot{T_d})$.

The equation now becomes,

$$\dot{T} - \dot{T}_d = -a_1 T + a_1 u - \dot{u}_{orig} - 2\frac{C_a}{bm} v_f \dot{v_f} - C_1(T_d - T) + C_1(T_d - T) \tag{76}$$

$$\Rightarrow \dot{T} - \dot{T}_d = -C_1(T - T_d) - \mathrm{a}_1(1 - \beta)T + a_1 u - \dot{u}_{orig} - 2\frac{C_a}{bm} v_f \dot{v_f} \tag{77}$$

Now,

$$v_a = \dot{V} + \frac{1}{\gamma}(T - T_d)\left(-C_1(T - T_d) - \mathrm{a}_1(1 - \beta)T + a_1 u - \dot{T}_d - C_1 T_d\right) \tag{78}$$

$$= \dot{V} - \frac{C_1}{\gamma}(T - T_d)^2 + \frac{1}{\gamma}(T - T_d)\left[-\mathrm{a}_1(1 - \beta)T + a_1 u - \dot{T}_d - C_1 T_d\right] \tag{79}$$

If we see the above equation and compare with the generalized form that we have got in the above backstepping control design, we had to choose the $\emptyset(x)$ such that the derivative is negative definite. Here, we have u and $\dot{V}$ that doesn't guarantee the negative definiteness. Hence, we choose u such that the term $\frac{1}{\gamma}(T - T_d)[-\mathrm{a}_1(1 - \beta)T + a_1 u - \dot{T}_d - C_1 T_d]$ vanishes from the equation. Now, we are left with the $\dot{V}$. Using the mathematical method of asymptotic convergence, we take the design constant γ very small such that the term $\frac{C_1}{\gamma}(T - T_d)^2$ is greater than $\dot{V}$.

Let us now model u such that

$$u = (1 - \beta)T + \frac{1}{a_1}\dot{T}_d - \beta T_d \tag{80}$$

Substituting the value of $\dot{T}_d$,

$$u = (1 - \beta)T + \frac{1}{a_1}(\dot{u}_{orig} + 2\frac{C_a}{bm} v_f \dot{v_f}) + \beta\left(u_{orig} + \frac{C_a}{bm}\ v_f^2\right) \tag{81}$$

Now let us see the equation of $\dot{u}_{orig}$

$$\dot{u}_{orig} = k_p \frac{d}{dt}(V_r + k\delta) + k_i + k_p \frac{d}{dt}(V_r + k\delta)\ |V_r + k\delta| \tag{82}$$

As we don't know about the acceleration of the vehicle ahead, we use the concept of dirty derivative given by

$$\frac{S}{St_d + 1} \tag{83}$$

Where,

t_d is infinitesimally small.

If it is small, the dirty derivative can be approximated to the normal derivative.

The β value is also taken as very small; we can approximate the new input u as

$$u_s = T + \frac{1}{a_1}\left(\dot{u}_{orig} + \frac{2\,C_a}{m} v_f \dot{v}_f\right) \tag{84}$$

instead of

$$u = (1-\beta)T + \frac{1}{a_1}(\dot{u}_{orig} + 2\frac{C_a}{bm} v_f \dot{v_f}) + \beta\left(u_{orig} + \frac{C_a}{bm}\ v_f^2\right) \tag{85}$$

CHAPTER- 4

PREDICTIVE DESIGN

4.1 Description:

A widely used approach for systems with known delays is to use a predictor in the control loop. One of the predictor is smith's predictor. It assumes that a compensator is already provided for plant to give a desired command response in delay free case.

$$k_s = \frac{k_0}{1 + (1 - e^{-s\tau}) p_0 k_0} \tag{86}$$

Where k_0 *is compensator* and p_0 is the plant, τ is the delay of the predictor in seconds.

For a platoon, we need to have a decentralized predictive controller with each vehicle. Hence it requires more robust predictor with error estimation which is not discussed in this report. As the model parameters are not given, we are using a Model Predictive Control in Simulink provided by MATLAB [9] for the prediction of the model for cruise control.

4.2 Simulink Model:

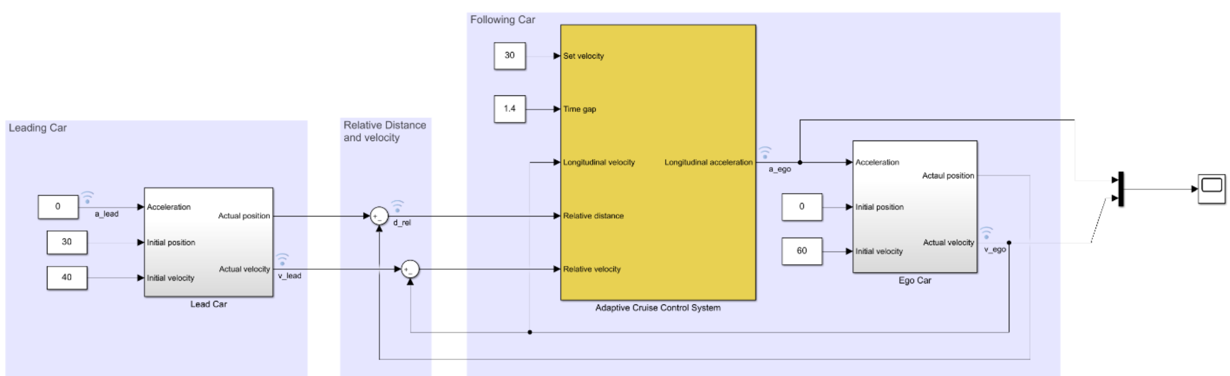

Figure 3: Model Predictive Control

CHAPTER - 5

PID CONTROLLER

5.1 Description:

PID controller algorithm consists of three basic coefficients; proportional, integral and derivative which are varied to get optimal response. Above mentioned controllers are all complex and involve a lot of estimations. Hence, for hardware easiness and cost reduction, there is a need to simplify the controllers.

Originally, we avoided PID control because we have no information about the derivative of the error term. Now that we have proposed the dirty derivative in the backstepping controller, we will use that for control design.

5.2 Input Model:

The input

$$u = K_p \frac{d}{dt}(V_r + k\delta) + k_i \frac{1}{S}(V_r + k\delta) + k_d \frac{S}{St_d + 1}(V_r + k\delta) \qquad (87)$$

Which can be used in the equation of

$$\dot{V_f} = \mathrm{a}(\mathrm{V_r} + \mathrm{k}\delta) + \mathrm{bu} + \bar{\mathrm{d}} \qquad (88)$$

In the simulations, we have assumed that there are no disturbances and differentiation calculated is discrete i.e., differentiation is the ratio of change in the error and time difference.

5.3 Code:

```
%PID controller without actuator delay of 0.2ms

clc;

clear all;

close all;

sl = 30; %initial Position of lead car

sf = 0; %initial position of following car

vf = 60; %initial velocity of following vehicle

vl = 40; %leading vehicle velocity

s0 = 5; %minimum distance required

h = 0.2; %fixed time headway

sd = s0+h*vf; %desired distance sd = so+hvf
```

```
xr = sl - sf; %initial relative distance
delta = xr-sd; %spacing error
vr = vl-vf; %following - leading velocities
past_vf =  vf;
kp = 1.5;
ki = 0.3;
kd = 0.01;
vf_acc_list=[];
i=1;
figure
for t=0:0.1:50
   k=0.1+(1-0.1)*exp(-0.1*delta*delta);
   h = 0.1-0.2*vr; %for variable time headway - comment this line for
   u = kp*(vr+k*delta) + ki*(vr+k*delta)*0.1 + kd*(vr+k*delta)*((vf-past_vf)/0.1); %u = kp*(vr+k*delta)+ki*(1/s)(vr+k*delta)+kd*(s/sTd+1)*(vr+k*delta): Td is small and discrete differentiation (dv/dt)
   vf_acc = 0.1*(vr+k*delta)+0.2*u; %vfdot = a*(vr+k*delta)+b*u
   subplot(2,2,1)
   plot(t,vf,'.','Color','red');
   xlabel('time')
   ylabel('m/s')
   title('Following Vehicle Velocity')
   xlim([0 50])
   ylim([0 60])
   hold on
   drawnow
   subplot(2,2,2)
   plot(t,delta,'.','Color','blue');
   xlabel('time')
```

```
ylabel('m')
title('Separation Error')
xlim([0 50])
ylim([-250 30])
hold on
drawnow
subplot(2,2,3)
plot(t,vf_acc,'.','Color','black');
xlabel('time')
ylabel('m/s2')
title('Following Vehicle Acceleration')
xlim([0 50])
ylim([-30 30])
hold on
drawnow
subplot(2,2,4)
plot(t,xr,'.','Color','m');
xlabel('time')
ylabel('m')
title('Vehicle Separation')
xlim([0 50])
ylim([-30 60])
hold on
drawnow
past_vf = vf;
vf_acc_list(end+1) = vf_acc;
%comment the if condition (36-39 lines) to get simulation with no
%actuator delay and uncomment line 40
```

```
    if(t>0.1)
        vf = vf+(vf_acc_list(i)*0.1);
        i=i+1;
    end
%     vf = vf+vf_acc*0.1;
    sd = s0+h*vf;
    if(t>2 && t<4.1) %accelerating lead vehicle with 2m/s^2 for 2 seconds
        vl = vl+2*0.1;
    end
    vr = vl-vf;
    sl = sl+vl*0.1;
    sf = sf + vf*0.1;
    xr = sl-sf;
    delta = xr-sd;
end
```

CHAPTER - 6

RESULT

6.1 Simulations:

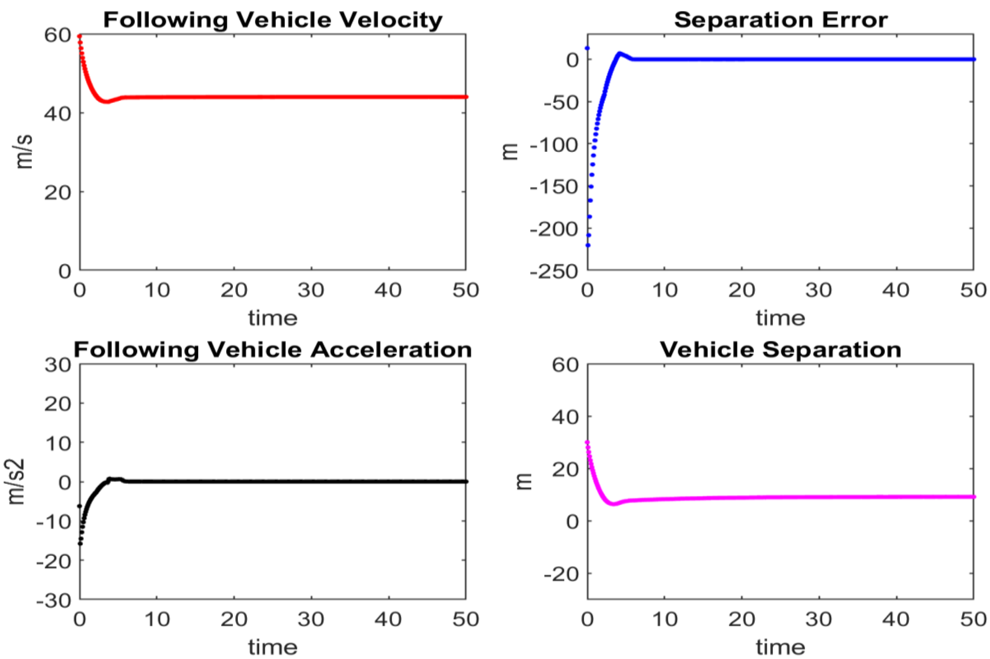

Figure 4 No Delay: Adaptive PIQ Controller with Variable Time Headway and Variable Separation Gain

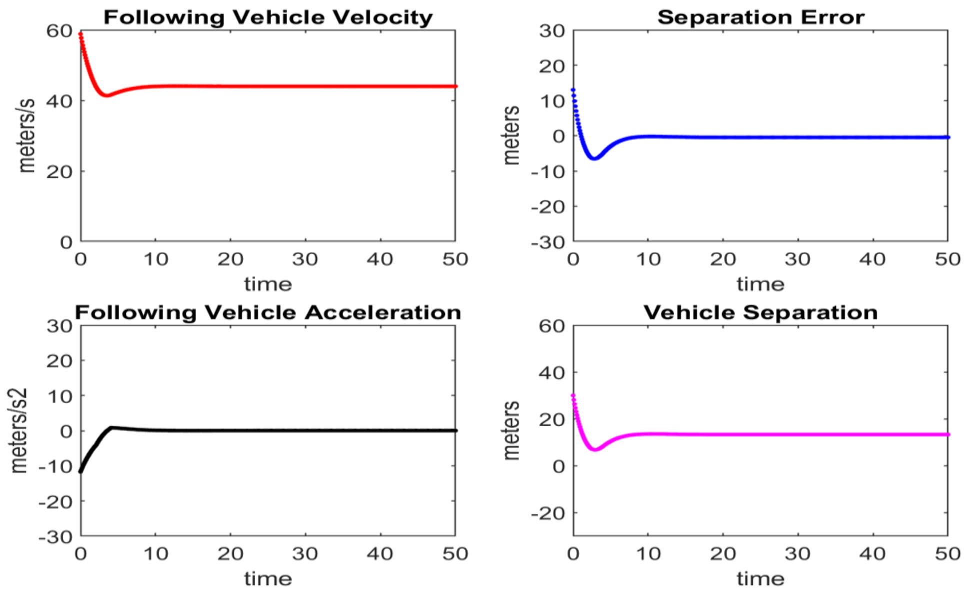

Figure 5 No Delay: Adaptive PIQ with Fixed Spacing

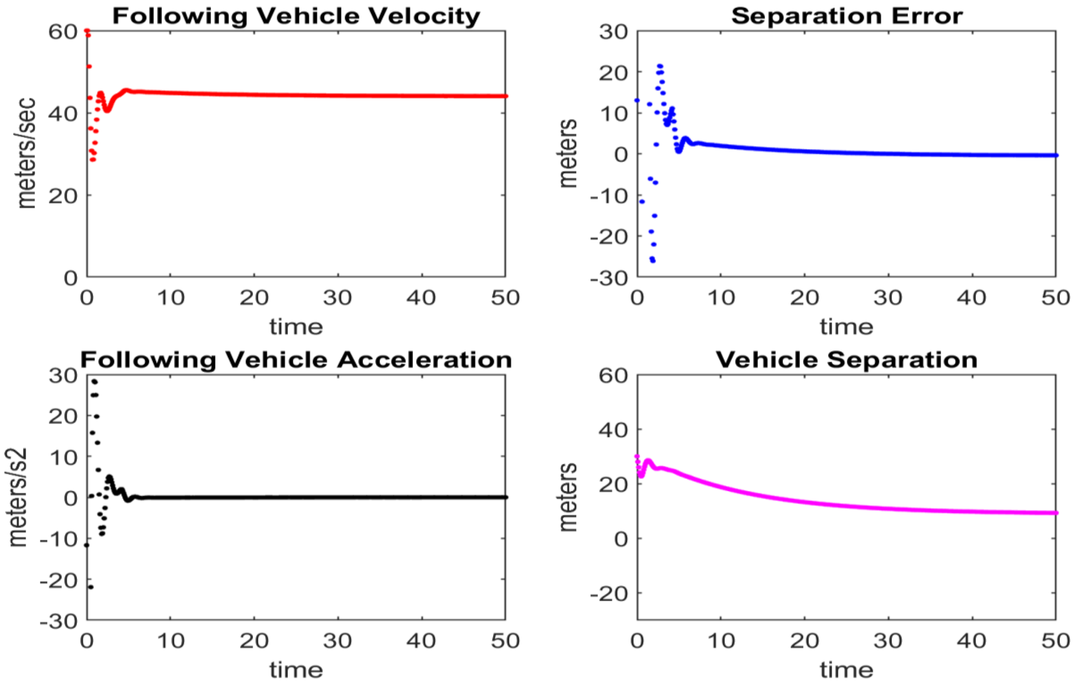

Figure 6 Actuator Delay: Adaptive PIQ with Variable Time Headway

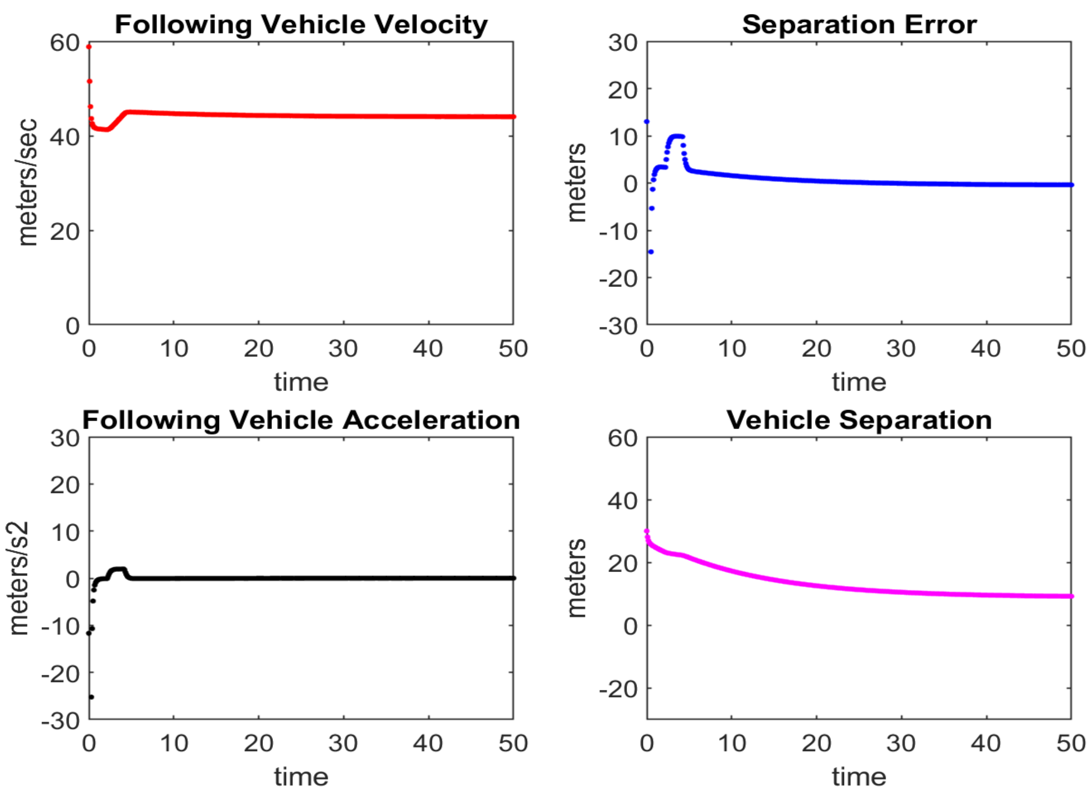

Figure 7 No Actuator Delay: Adaptive PIQ with Variable Time Headway

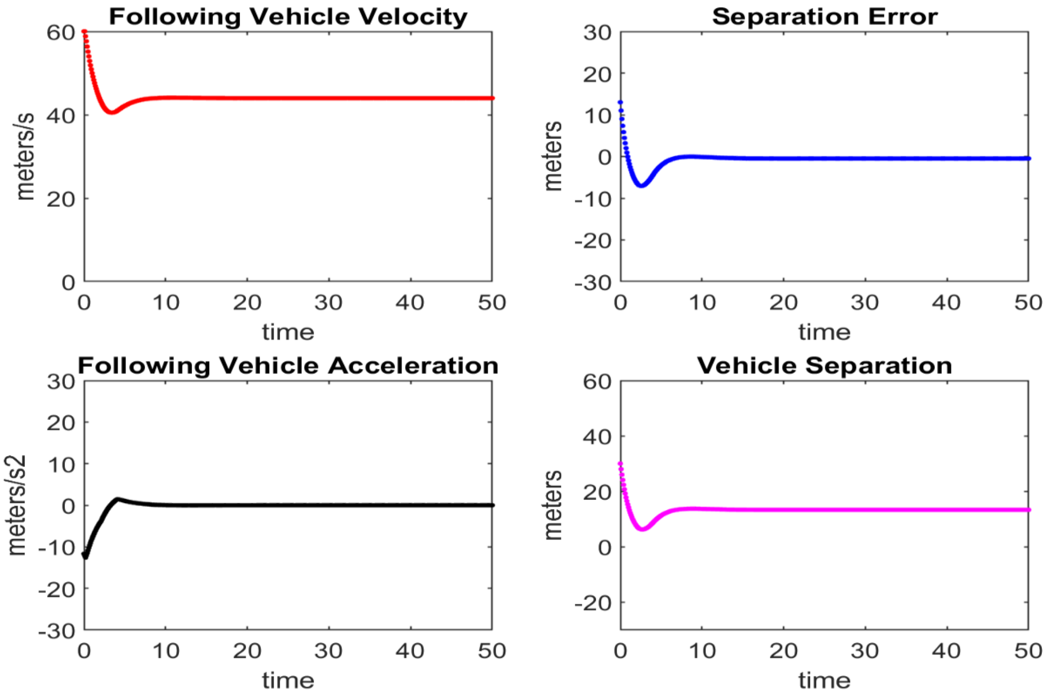

Figure 8 Actuator Delay: Adaptive PIQ with Fixed Spacing

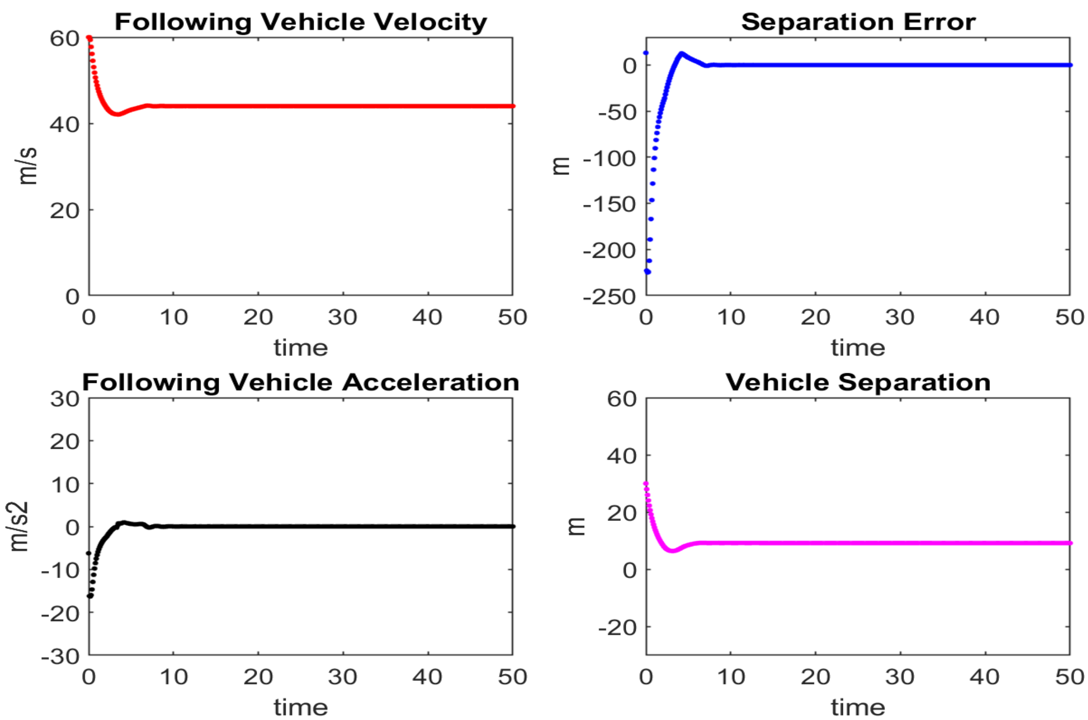

Figure 9 Actuator Delay: Adaptive PIQ Controller with Variable Time Headway and Variable Separation Gain

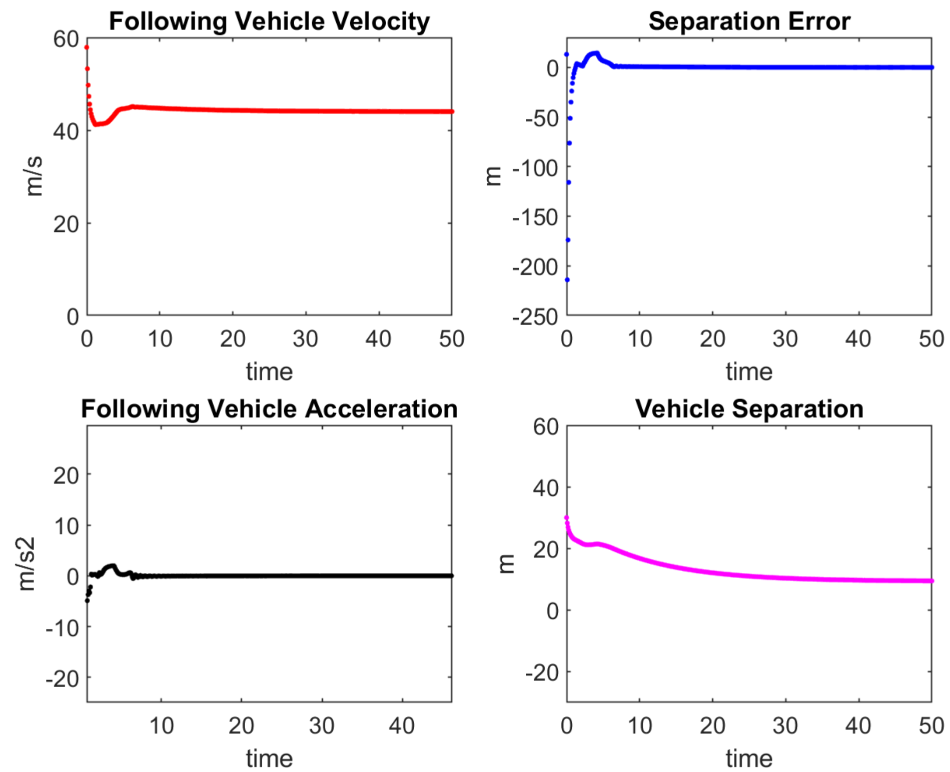

Figure 10 Actuator Delay: Adaptive PIQ Controller with Variable Time Headway and Variable Separation Gain with similar PIQ gains of Adaptive PIQ Controller without variable time Headway

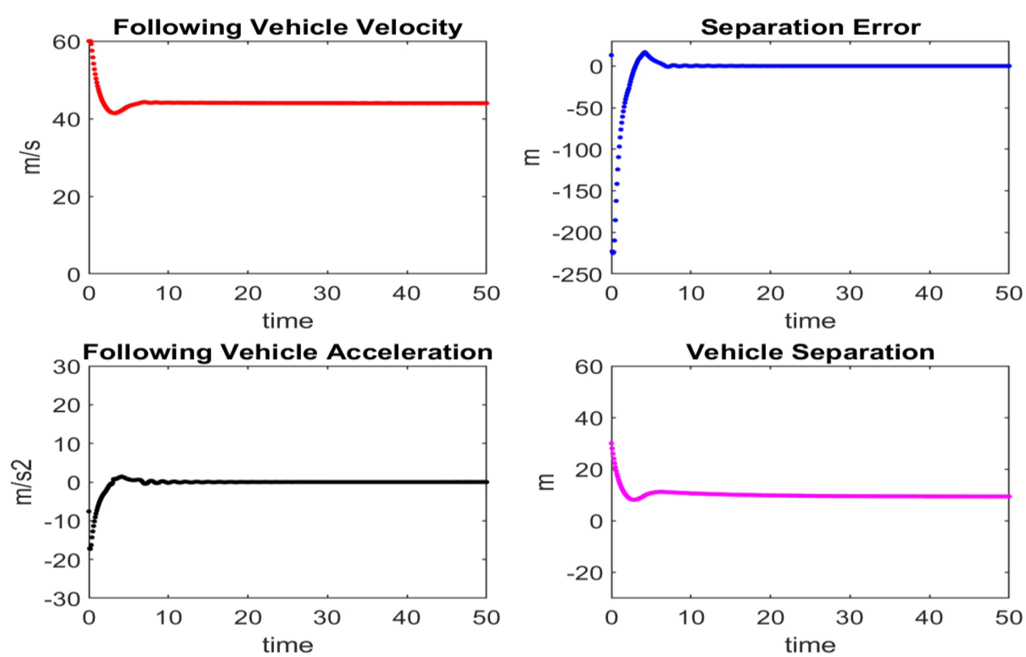

Figure 11 No Delay: PID Controller with Variable Time Headway and Variable Separation Gain

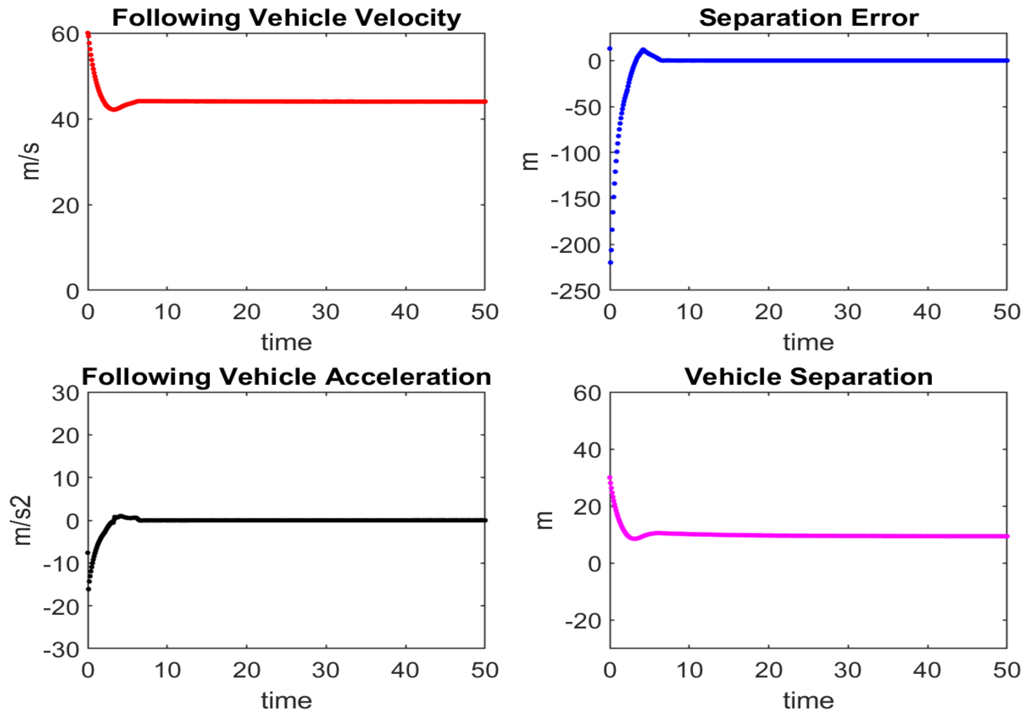

Figure 12 Actuator Delay: PID Controller with Variable Time Headway and Variable Separation Gain

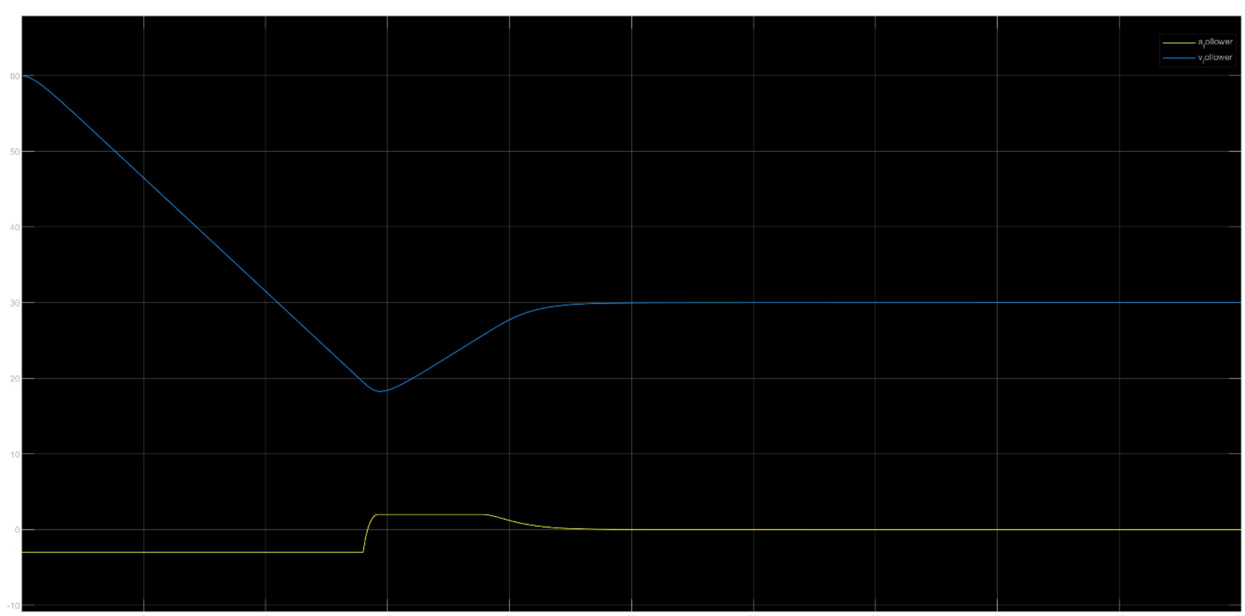

Figure 13 Actuator Delay: Model Predictive Control

6.2 GitHub Link for Codes and MPC model:

https://github.com/okritvik/ENPM-667_Project-1

Note that the simulations will take time due to multiple subplots and iterative live plotting of the parameters. Normal code run time is 4 minutes with Nvidia GeForce GTX 1060 6gb graphics card, i7 hexacore processor.

6.3 Conclusions:

We have taken the follower vehicle velocity higher than the leading velocity to prove that the vehicle doesn't collide with the vehicle in front.

In the Adaptive PIQ controller with variable time headway and variable separation gain, we can see that we had to reduce the PIQ gains so that the error is reduced, and the cruise is smooth. This is because of the separation gain and the headway are function of the relative velocity.

If we compare the nonlinear PID controller with the Adaptive PIQ controller, the simulations are nearly close and hence we can assume that the complexity can be reduced in the form of converting the PIQ to PID, thus reducing the manufacturing cost and hardware complexity.

The variable time headway and variable separation gain has made the relative distance between the follower and leader smoother for changes in the acceleration of leader. The changes were encountered in the separation error.

The fixed time headway (fixed spacing policy) might be looking good in the simulations but the disadvantages of the above were said in explanation of opting for the variable time headway.

The controller with Model Predictive Control did not give better performance than the ones without Model Predictive Control. For some velocities, the vehicles collided with each other.

6.4 Future Scope:

In this technical paper report, we have assumed our own design parameters (gains) and also the velocities and initial spacing were taken different to make sure that the linearized model works well as per the controller design. The authors in the research paper have mentioned that the simulations were taken with the original nonlinear model which involves practical regression values and parameters that were not mentioned in their previous work.

This work can be extended for simulating the model with multiple followers and intervehicle communication. The backstepping controller, backstepping with prediction were not simulated in this report due to insufficient parameters but can be designed with proper information.

BIBLIOGRAPHY

[1] M. Krstic´, I. Kanellakopoulos, and P.V.Kokotovic´, Nonlinear and Adaptive Control Design. New York: Wiley, 1995.

[2] J. K. Hedrick, D. H. McMahon, V. K. Narendran, and D. Swaroop, "Longitudinal vehicle controller design for IVHS systems," in Proc. Amer. Contr. Conf., Boston, MA, 1991, pp. 3107–3112.

[3] P. Ioannou and Z. Xu, "Throttle and brake control systems for automatic vehicle following," IVHS J, vol. 1, pp. 345–377, 1994.

[4] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in Proc. Amer. Contr. Conf., San Diego, CA, 1990, pp. 291–297.

[5] S. E. Shladover, "Longitudinal control of automated guideway transit vehicles within platoons," ASME J. Dynamics Syst., Meas., Contr., vol. 100, pp. 302–310, 1978.

[6] P. Varaiya, "Smart cars on smart roads: Problems of control," IEEE Trans. Automat. Contr., vol. 38, pp. 195–207, 1993.

[7] D. Yanakiev and I. Kanellakopoulos, "Nonlinear spacing policies for automated heavy-duty vehicles," IEEE Trans. Veh. Technol., vol. 47, pp. 1365–1377, 1998.

[8] D. Yanakiev and I. Kanellakopoulos, "Variable time headway for string stability of automated heavy-duty vehicles," in Proc. 34th IEEE Conf. Decision and Control, New Orleans, LA, 1995, pp. 4077–4081.

[9]https://www.mathworks.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html

[10] https://www.youtube.com/watch?v=gCqQUbMuQ2w&ab_channel=nptelhrd

[11] Horacio Márquez "Nonlinear Control Systems: Analysis and Design" Wiley, 2003