|  |  |
| --- | --- |
| Designing of Adaptive Cruise Control (ACC) using Adaptive Model Predictive Controller Name : **Devraj Sen**Unique ID: **2005212** |  |
|  |  |

**Table of Content**

|  |  |  |
| --- | --- | --- |
| Serial number | Content | Pages |
| 1 | Introduction | 3 |
| 2  2.1 | Literature Review  Adaptive Cruise Control(ACC) Architecture | 4  5 |
| 3 | Analysis – 4W & 1H Technique | 7 |
| 4 | SWOT Analysis of Adaptive Cruise Control | 8 |
| 5  5.1  5.2  5.3  5.4  5.5  5.6  5.7  5.8 | Model Predictive Controller(MPC)and Simulink Designing  What is MPC?  Why MPC?  MATLAB Modelling  Parameters to be designed  Driving Scenario Designer  Output of the Model  Application  Comparison of Adaptive MPC and Traditional MPC | 9  9  10  12  13  15  17  18  19 |
| 6  6.1  6.2  6.3  6.4  6.5 | Skills needed to be demonstrate  Callback Function  Data Inspector  Solver Selection Strategy  MATLAB Function block  Lookup Table and Signal Builder | 21  21  22  23  29  30 |

## **Introduction**

## C:\Users\Laptop\Desktop\Adaptive_cruise sontrol.PNG

## Figure 1: Adaptive Cruise Control

Adaptive cruise control is one of the first steps in moving towards autonomous driving vehicles, as it allows the vehicle to regulate speed and braking on the highway to maintain a set following interval behind the detected car ahead.

So how does adaptive cruise control work?

The primary purpose of the system is to travel at the speed selected by the driver, just like traditional cruise control. However, if the vehicle detects another car in front of it that is traveling at a slower speed, the vehicle will reduce its speed to match that of the detected car and a then maintain a selected interval behind the car. The system works by emitting radar waves which bounce off of vehicles ahead and return to the unit. This informs the system of the distance between the two vehicles; changes in that distance inform the system of the vehicles’ relative speeds.

## **Literature Review**

## **2.1 Adaptive Cruise Control(ACC) Architecture**

## C:\Users\Laptop\Desktop\ACC\ACC arch.PNG

## Figure 2: Architecture

The main part of the ACC system is a detector whose task is to measure the distance to the vehicle in front. There have been experiments with different types of detectors. Optical systems have been tested and been abandoned due to their high demands on clear and light-reflecting surfaces. Laser systems works well in clear weather, but has showed not to be up to the mark in rainy or snowy weather. Therefore, the dominating detector system of today is the radar.

It sends an electromagnetic beam forward in the vehicle’s direction, and the distance it measured by the time it takes for the beam to be reflected. It works well in all kinds of weather; its disadvantage is that it is more expensive that the other types of detectors.

Even though it is possible to measure the time of the beam, it is not done directly in ACC systems since it demands sophisticated and expensive equipment. Instead, the time is measured indirectly. One commonly used method is the Frequency Modulated Continuous Wave (FMCW) method. A Voltage Controlled Oscillator (VCO) generates a high frequency beam. One suitable VCO is the Gunn oscillator. It is built upon gallium arsenide semiconductors that generate very high frequencies when subjected to a strong electric field. It is capable of generating frequencies in the interval 76 – 77 GHz, which is the frequency assigned for ACCs.

When sending a beam of a known frequency (in the 76 – 77 GHz interval) and detect the differential (echoed) frequency, the distance can be calculated. The differential frequency can be obtained by a low pass filter and converted to frequency spectrum by Fast Fourier transform (FFT). A peak in the spectrum corresponds to the transmission frequency. By comparing the transmitted and received frequency, the distance can be calculated. However, one problem that has to be taken into consideration is the Doppler Effect. That problem can be solved by comparing the result from several transmission frequencies.

In most cases, one beam is not enough.

In order to detect obstacles also when the vehicle if moving

through a sharp turn, it is necessary for the unit to scan an angular range. However, the angular range cannot be allowed to widen too much. If the beams are scanning a large angular range, a vehicle travelling in another lane can be considered an obstacle. Experiments have showed that an angular range of approximate eight degrees is optimal. The purpose is obtained by using three identical transmissions and receiving devices. The actual distance is simple taken to be the smallest of the three calculated distances. Naturally, it is of outmost importance that the sensor is calibrated to follow the central line of the vehicle, both in horizontal and vertical directions.

Moreover, the outmost part of the ACC sensor unit is a lens directing the beam. In order for it to work it must not be covered with snow or ice, it has to be heated in below-zero temperatures. Therefore, in vehicles aimed at the northern market, a small thermostat and a heating device is included in the ACC sensor unit. The lens must also be able to withstand splash water, pressurized steam, and stone impact.

The scanned information is then sent to the ACC control unit, which task is to transform the differential frequency into a digital value, calculate the distance to the nearest obstacle, collect information from other sensors, inform the Electronic Throttle Controller (ETC) whether to increase or decrease the throttle, and if necessary, actuate the brakes.

The ACC control unit is constituted by two units: the Regulation Processing Unit (RPU) and the Signal Processing Unit (SPU). The RPU includes an amplifier, an Analogue-to-Digital Converter (ADC),

RAM memory for temporary storing of calculations results, and an interface to the Control Area Network (CAN) bus. The ADC monitors the voltage level. If it falls below a certain level, the ACC is disabled and the driver is notified. Like all vehicle electronics, the ACC is driven by the vehicle battery. However, as a regular 24 V car battery can give voltage peaks up to 100 V, the ACC must be protected by a voltage regulator that makes sure the ACC is fed a consistent voltage, usually 8 V.

The SPU is responsible for calculating the distance to the nearest obstacle. This task demands a very fast processor. Bosch (2003) developed the CC610 circuit, which is a complex DSP developed

especially for the ACC control unit. A fast SPU is able to interpret the information from the sensor in about 80 milliseconds. Therefore, the ACC control unit often runs at a frequency of 10 Hz.

In every vehicle, the driver always has the authority to disable the ACC. The driver can also set the preferred distance to the vehicle ahead and the preferred velocity. The display is also equipped with warning lamps (and in some case a sound alert) to warn the driver of obstacles on the road if the ACC should malfunction.

In some ACCs, there is a gradual warning system: in case of an appropriate distance between the vehicles, a green lamp is shining.

When the vehicles are closing in on each other, a yellow lamp start to shine, and when the distance is dangerously small, a red lamp shines. In most ACCs, when the red lamp shines, the brakes are automatically applied.

In order to keep the set distance, the ACC needs to notify the Electronic Throttle Control (ETC) when to increase or decrease the throttle. In some cases, the velocity has to be brought to a stop so rapidly that it is not enough to just decrease the throttle. In these cases, the ACC notifies the Electronic Stability Programme (ESP) to engage the brakes.

Apart from the radar sensor, the vehicle may be equipped with other sensors that provide information about the temperature, humidity, friction against the road, the cargo weight, etc. If this is the case, the SPU does include this information in its decision making.

1. **Analysis – 4W & 1H Technique**

The analysis will be done for conventional cruise control.

|  |  |  |
| --- | --- | --- |
| **4W-1H** | **Question** | **Answers** |
| What? | **What** is the problem of conventional cruise control? | Constant Speed |
| Where? | **Where** is the problem? | Cruise control is located near the steering wheel. |
| When? | **When** is the problem? | When there is heavy traffic environment. |
| Who? | **Who** is affected? | The driver and other passengers present in the car. |
| How? | **How** the problem affects? | Conventional cruise control is when the driver sets the speed so whatever maybe the circumstances, the speed will be constant.  So when there is a heavy traffic, there is a high chance of accident occurs not realizing the speed of the car in front. |

So this is the reason why Adaptive cruise control is preferred since it automatically adjusts the speed of your car to match the speed of the car in front of you.

1. **SWOT Analysis of Adaptive Cruise Control**

|  |  |
| --- | --- |
| **Strengths** | **Weakness** |
| * Autonomy in that all necessary technology and intelligence is available on board. * Increases safety and comfort. * Smoothens traffic flow * Decreases fuel consumption * Decreases environmental pollution * Capacity increase under short gaps. * Enables forming of vehicle platoons. | * Autonomy implies that network-wide beneficial settings cannot be directly communicated and/or imposed. * Capacity decrease under conservative gaps. * On-ramp flow merging problems under short gaps and high penetration rates. * Limited speed-range operation. * Control laws that do not ensure traffic stability under all circumstances. |
| **Opportunities** | **Threats** |
| * Advice/recommendations on network-wide beneficial system settings via traditional VMS or navigation devices or build-in (autonomous) extensions. * Enabling network-wide beneficial system settings via V2I communication. * LSACC/FSRA extends speed- range operation, thus applicability to all traffic conditions. * CACC enables even shorter gaps. * V2V and/or V2I communication may assist and smooth on-ramp merging flows. * Control – theoretical research may provide more efficient control laws. * Technology maturity may reduce system cost. | * User acceptance in terms of both purchase intention and frequent activation after purchase. * Cost * MTM delayed adaptation. |

1. **Model Predictive Controller(MPC)and Simulink Designing**

**5.1 What is MPC ?**

MPC is a feedback control algorithm that uses a model to make predictions about future output of a process.

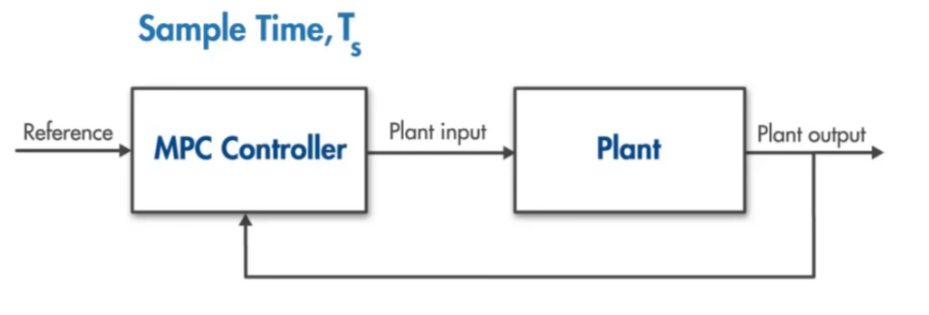


Figure 4

Normally when we have a plant and we need to use PID controller so for every reference there will be a PID controller. So for a large plant system tuning all the PIDs will be a difficulty.

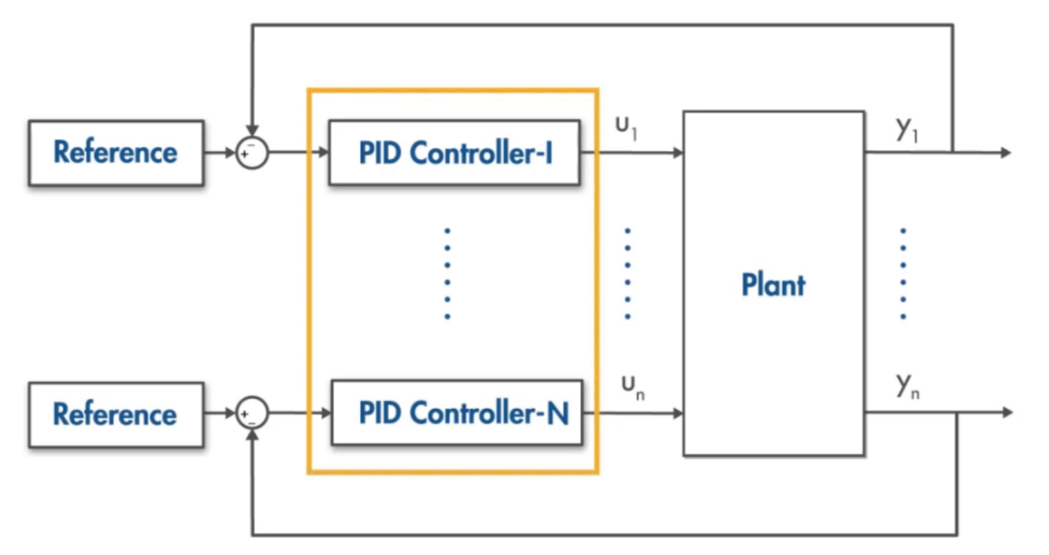


Figure 5

So to remove this difficulty, we can use Model Predictive Controller which will optimize itself according to what required.

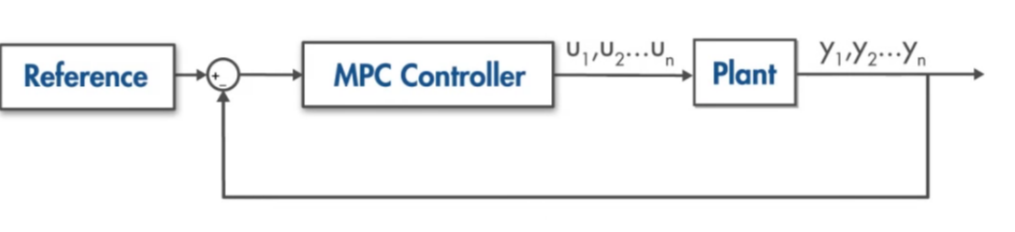


Figure 6

* 1. **Why MPC?**

1. MPC can handle Multi-input-multi-output (MIMO) system
2. MPC can handle constraints.
3. MPC has preview capability.
4. MPC can incorporate future reference information into the control problem to improve controller performance.

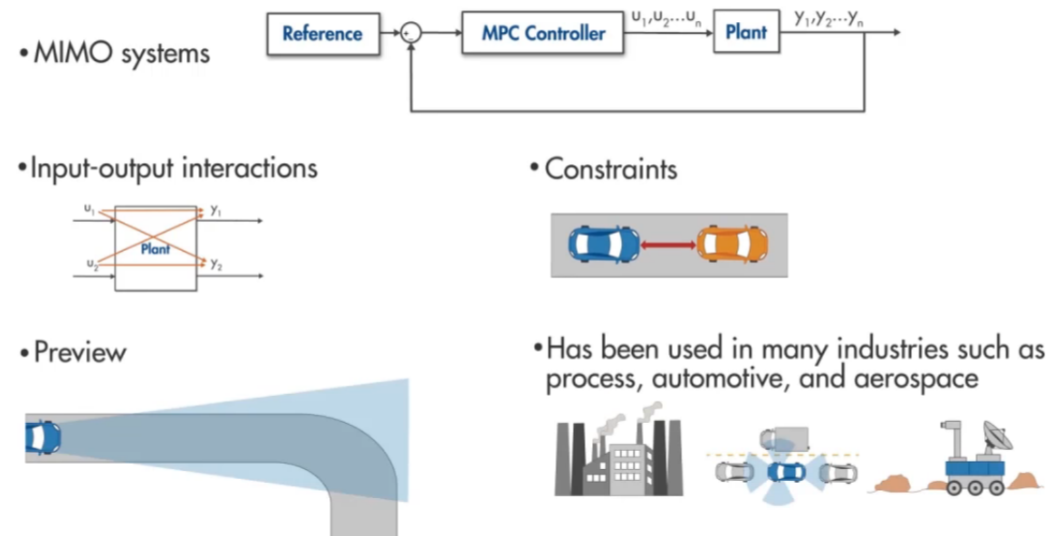


Figure 7

MPC requires a powerful, fast processor with a large memory because MPC solves an online optimization problem at each time step.

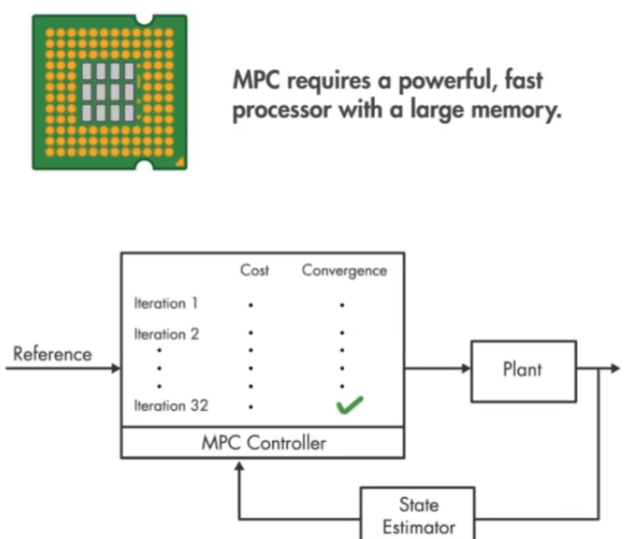


Figure 8

* 1. **MATLAB MODELLING**

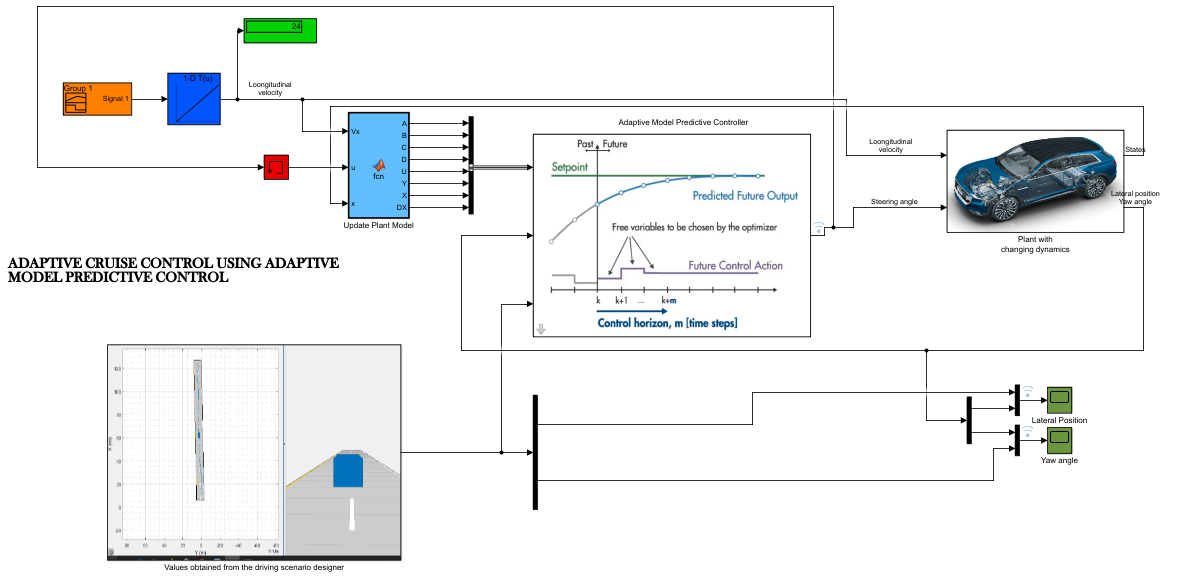


Figure 9

The plant is developed as a state space system with an input of steering angle and longitudinal velocity and output position of lateral position and yaw angle.

In Adaptive MPC, No Measured Disturbance is assumed.

The Adaptive MPC block has same input and output as the regular MPC block except that it takes the plant model that is updated at each time step for current operating condition.

We connect plant output to the measured output port of Adaptive MPC and steering angle to the controller output port and reference (designed by Driving Scenario Designer) to the controller port of Adaptive MPC.

We design custom reference for lateral position and yaw angle.

* 1. **Parameters to be designed**

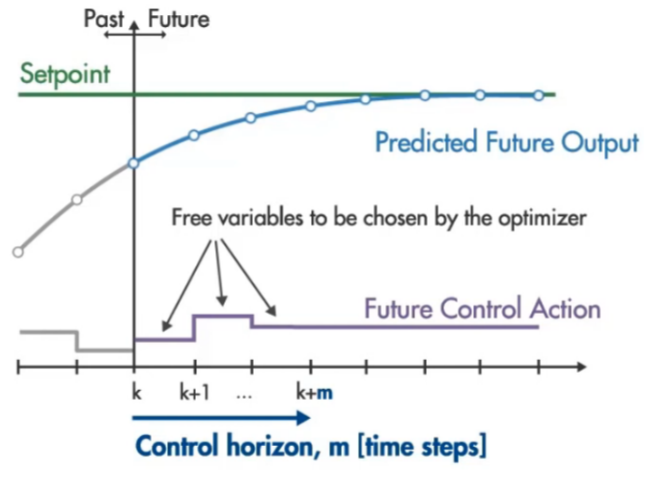


Figure 10

The parameters to be designed are :

1. Set point
2. Predicted Future Output
3. Control horizon
4. Future Control Action

Choosing proper values for these parameters is important as they affect not only the controller performance but also the complexity of the MPC algorithm that solves an online optimization problem at each time step.

If these parameters are not properly designed then we face Quadratic Programming Problem in MPC and as, the number of states and constraints and length of control horizon and prediction horizon, increases – the complexity of MPC increases.

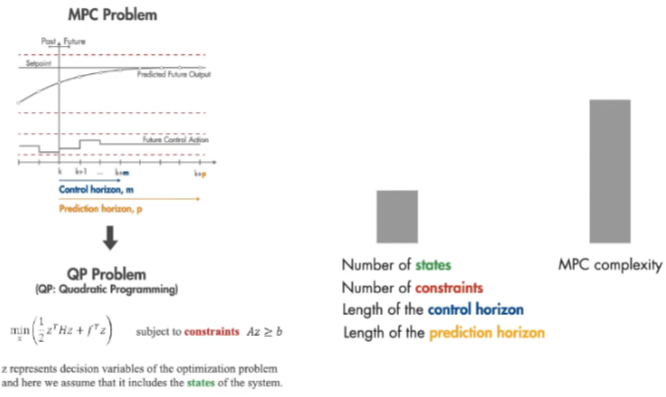


Figure 11

Methods to run MPC faster –

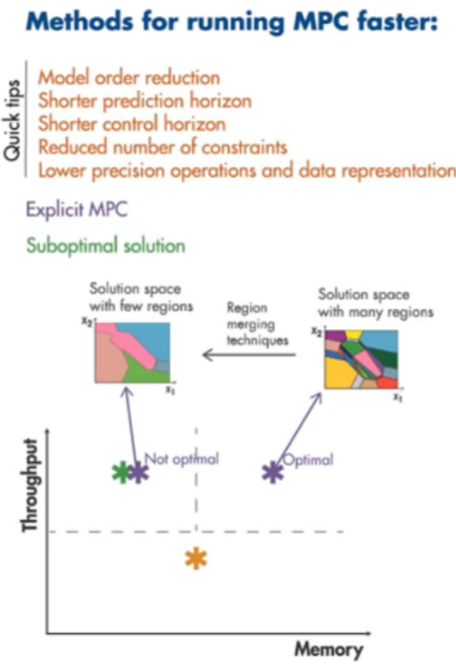


Figure 12

* 1. **Driving Scenario Designer**

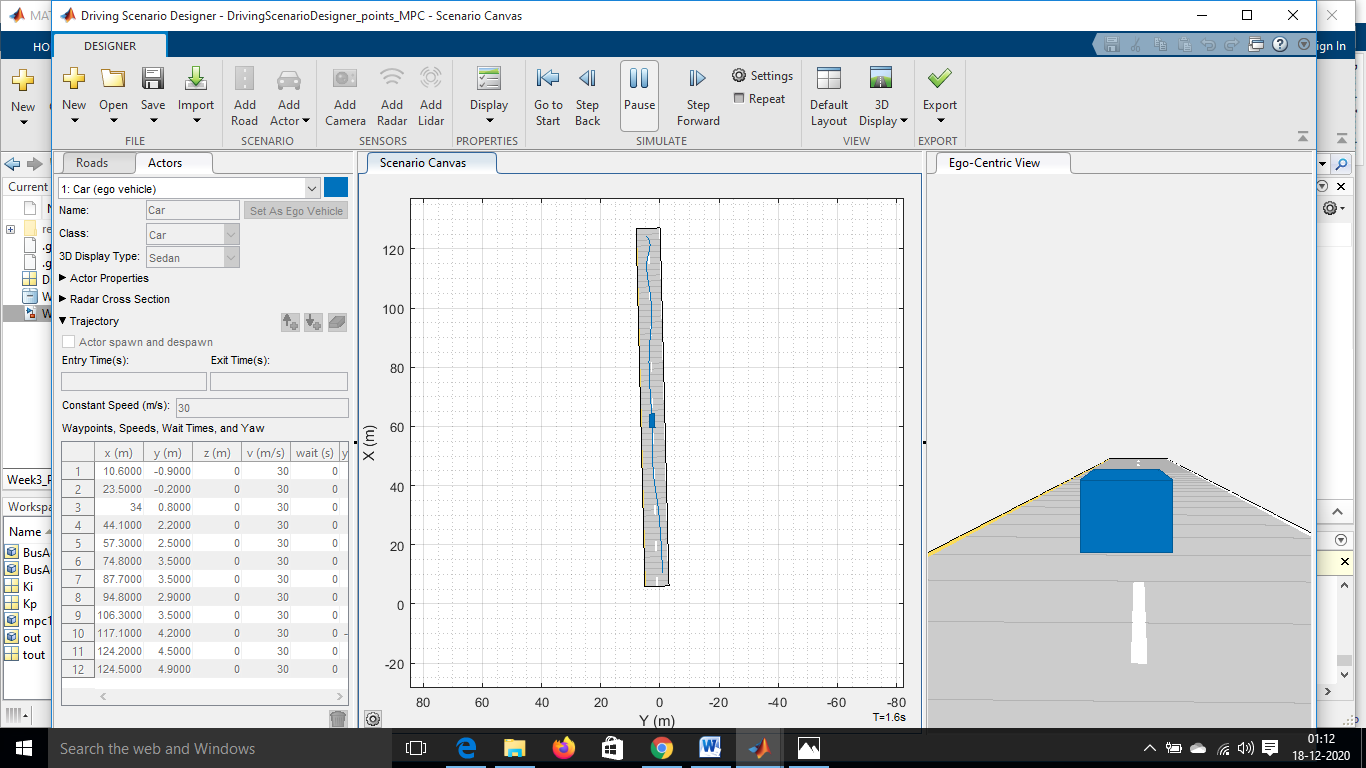


Figure 13

Steps followed to design the points where the car travels and adding those points as reference (position and yaw rate) in the MATLAB model:

1. In the Automotive Toolbox, you can open this app “Driving Scenario Designer”.
2. Click on “Add Road” and then design the road and then specify the width and lane.
3. Specify the longitudinal velocity.
4. Click on “Add car” and then click on “Add Forward Waypoints” so that the car moves forward.
5. Click on Run or Step Forward to analyze the movement of car after every time step.
6. Then import the data as MATLAB function to the respective Simulink model.

You can also run the model which you designed in 3D :



Figure 14

If you type “mpc1” in the command window – All the settings done in Adaptive Model Controller will be shown.

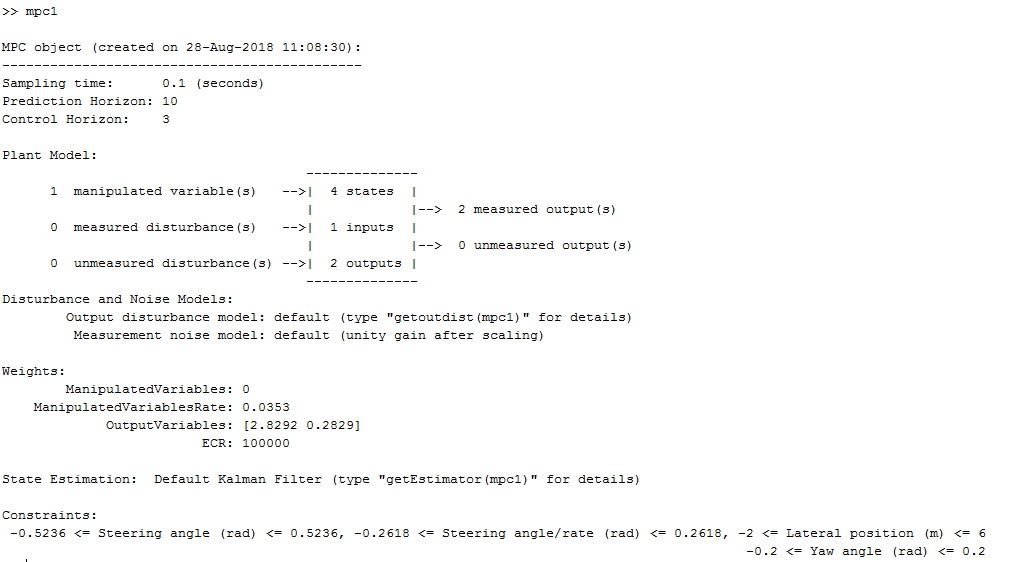


Figure 15

* 1. **Output of the MATLAB Model**

My objective is to get the output similar to desired reference irrespective of whatever longitudinal velocity input is given. In other words to check that is the Adaptive MPC controller optimizes the system during dynamic condition.

Output measured :

1. Steering angle
2. Lateral position
3. Yaw angle

The input of the system is longitudinal velocity which is given a dynamic response and the following graphs are recorded.

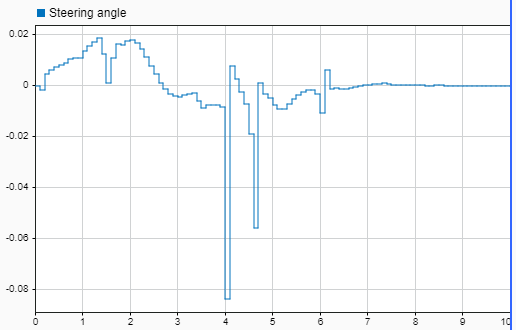


Figure 16

Lateral positon :

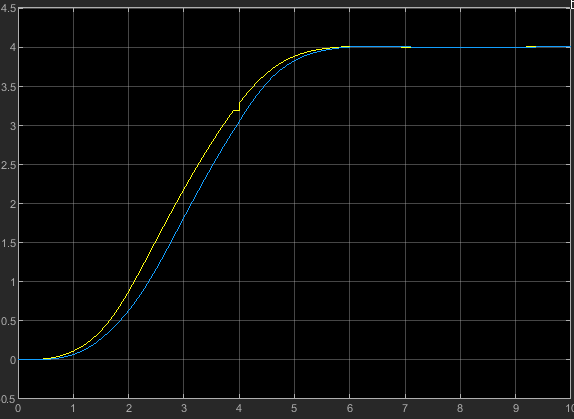


Figure 17

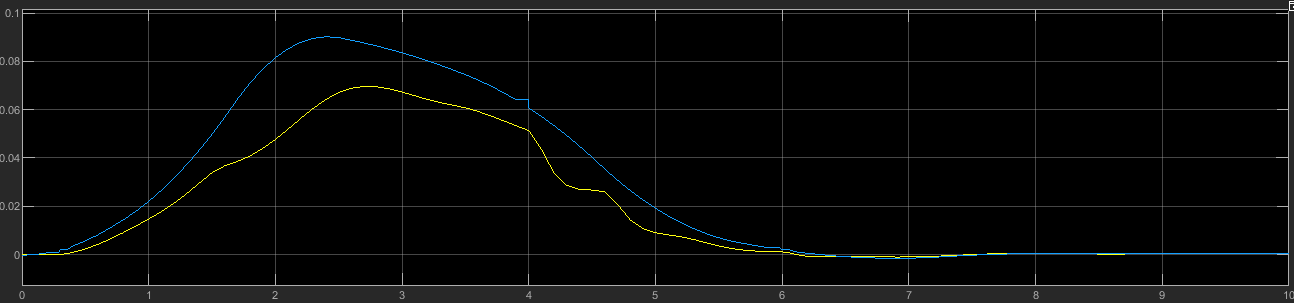
 Yaw rate :

Figure 18

So though the longitudinal velocity changes but the Adaptive Model Predictive Controller optimizes the system in such a way that it always keep the required output similar to the desired output.

* 1. **Application**

If you want to run your controller on your autonomous car , you can simply generate code using Embedded coder and display it in your car through image processing.

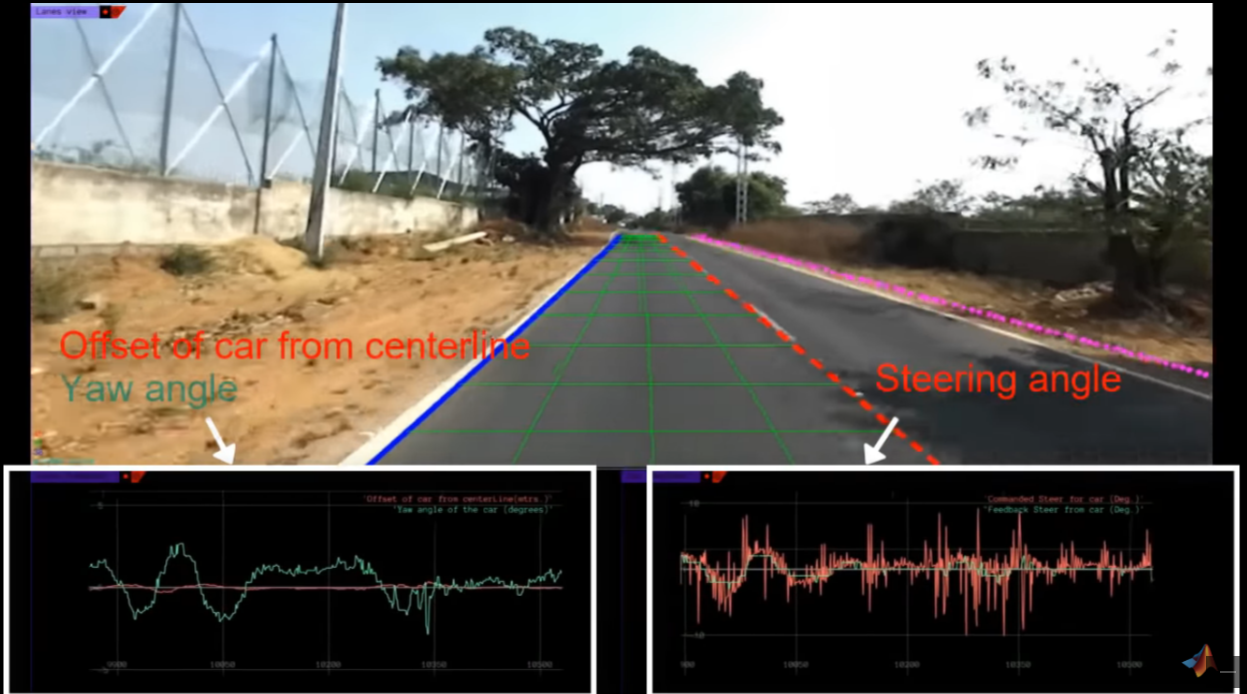


Figure 19

The image processing and lane detection algorithm developed outside of Simulink provide these inputs to the MPC controller.

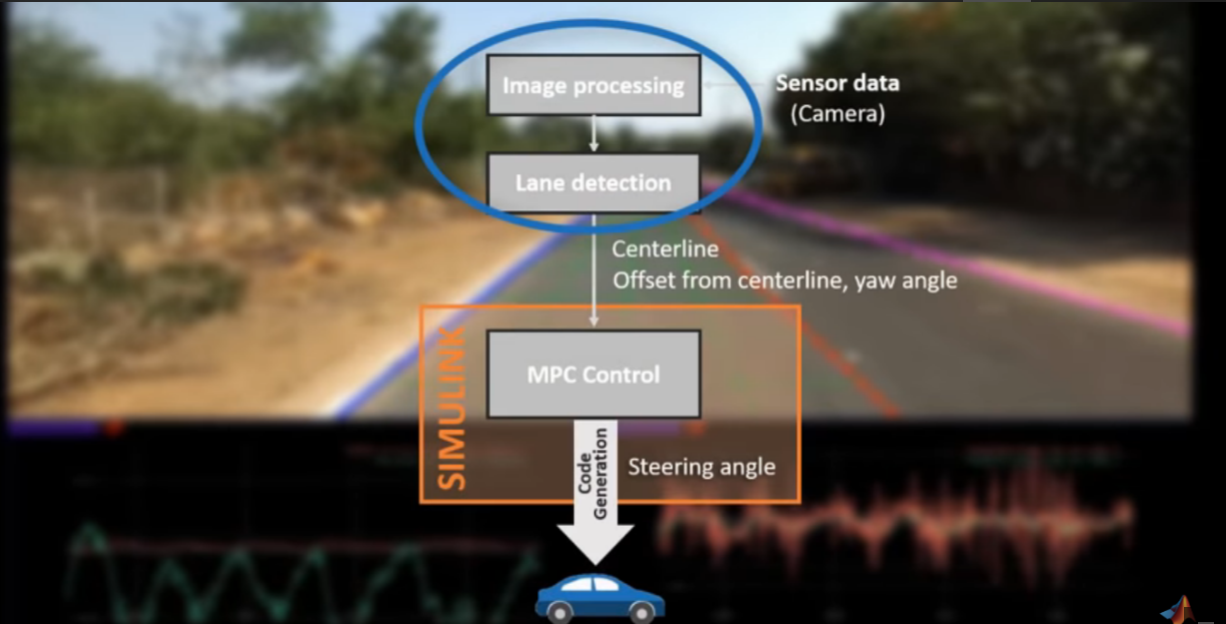


Figure 20

* 1. **Comparison of Adaptive MPC and Traditional MPC**

In Traditional MPC, the system does not work for dynamic longitudinal input.

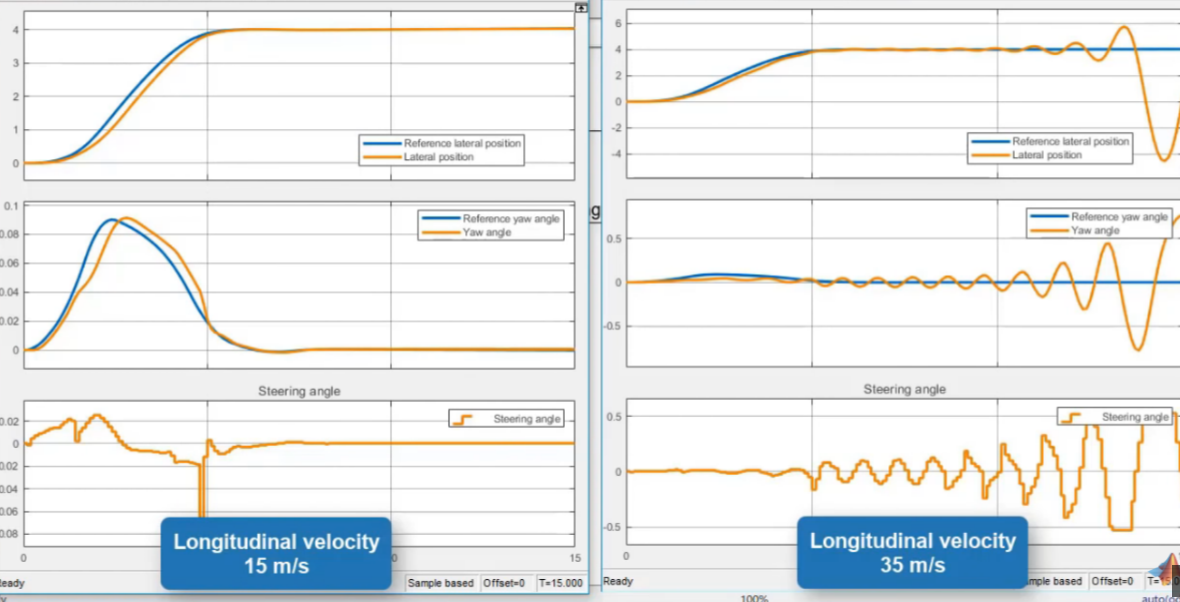


Figure 21

So as you can see as the velocity changes , the response changes and becomes poor system.

But if you use Adaptive MPC, the system works efficiently for dynamic longitudinal input.

So whatever longitudinal input we give , the response is similar to the desired reference.

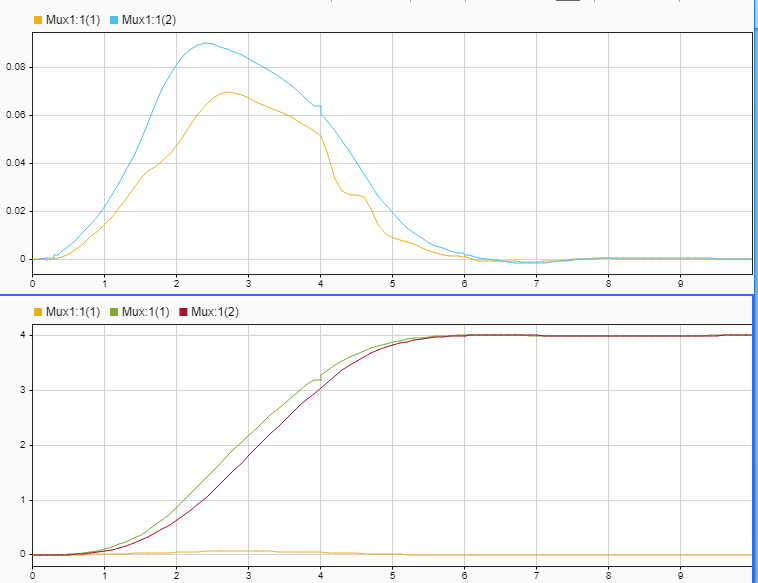


Figure 22

1. **Skill need to be demonstrate :** 
   1. **Callback Function :**

In Preload function – Added the script of Lookup Table.

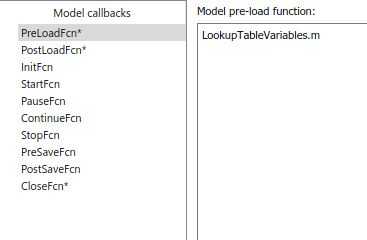


Figure 23

In Postload function – Added all the variables of the model workspace.

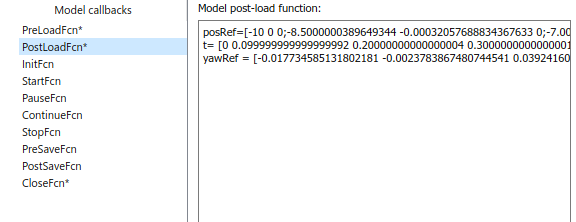


Figure 24

In Close function – Give the command “clearvars” so that when the system closes the base workspace gets cleared.

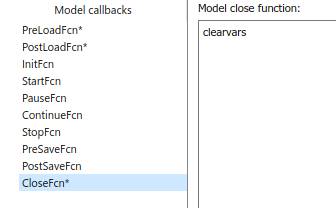


Figure 25

* 1. **Data Inspector**

The output response signal, which is Lateral position and yaw rate has been logged.

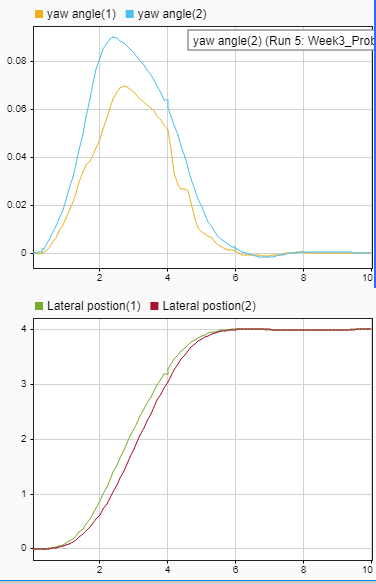


Figure 26

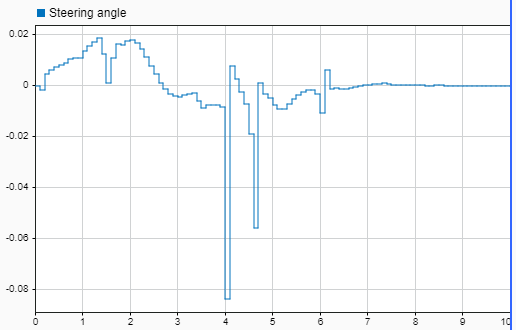
The steering angle signal which comes out of Adaptive Model Predictive controller has been logged.

Figure 27

* 1. **Solver Selection Strategy**

**Names and their Meaning**

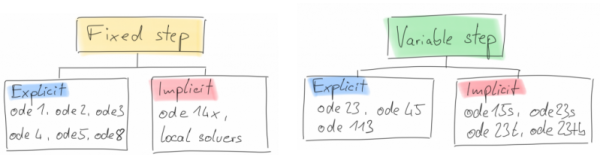


The most important info is carried by the solver names themselves. A single digit number represents a fixed-step solver, the number itself gives info about the order. Information on the integration scheme can also be obtained from the name. Solvers using an implicit scheme always come with an amended letter, explicit solvers only have the number. The amendment letters are abbreviations: “t” trapezoidal, “tb” trapezoidal-backward, “s” stiff and “x” extrapolation. If there is no amendment, that’s an explicit solver.

**Classification**

MathWorks solvers can be classified into 4 categories:  
Step Size – Order – State Updates – Integration Scheme.

**Step Size**



Fixed-step solvers are typically used for to meet accuracy requirements and are used when models will be deployed onto hardware such as microcontrollers. Variable step solvers automatically change the time step to meet requirements. They are typically used for plant model development. A variable-step solver might shorten the simulation time of your model significantly.

**Order**

As mentioned before, the numbers used in the solver name specify the solver order. Choice of order has an impact on accuracy. Most often, a high-order solver is more efficient than a low-order solver. Variable-order solvers use multiple orders to solve the system of equations.

For example, the implicit, variable-step ode15s solver uses first-order through fifth-order equations while the explicit, variable-step ode113 solver uses first-order through thirteenth-order. Solver’s **step size** and **integration order**have a huge impact on accuracy. In my mind, the simplest way of explaining is the following:

1. Smaller step size and higher solver order both increase accuracy
2. More accuracy is slower (at least for a fixed-step solver)

**State Updates: Discrete vs. Continuous**

**i**f your model has no continuous states, then Simulink switches to either the fixed-step discrete solver or the variable-step discrete solver. If your model has only continuous states or a mix of continuous and discrete states, choose a continuous solver from the remaining solver choices based on the dynamics of your model. Otherwise, an error occurs.

**Integration Scheme: Explicit vs. Implicit**

While you can apply an implicit or explicit continuous solver to solve all these systems, implicit solvers are designed specifically for solving stiff problems. Explicit solvers solve non-stiff problems. An ordinary differential equation problem is said to be stiff if the desired solution varies slowly, but there are closer solutions that vary rapidly.

In short, one could say: a stiff system has both slowly and quickly varying continuous dynamics. The numerical method must then take small time steps to solve the system. Stiffness is an efficiency issue. The more stiff a system, the longer it takes to for the explicit solver to perform a computation.

**Practical Aspects :**

**Solver Choice**

When you build and simulate a model, you can choose either type of solver based on the dynamics of the model. A model that contains several switches, like an inverter power system, needs a fixed-step solver. A variable-step solver is better suited for purely continuous models, like the dynamics of a mass spring damper system.

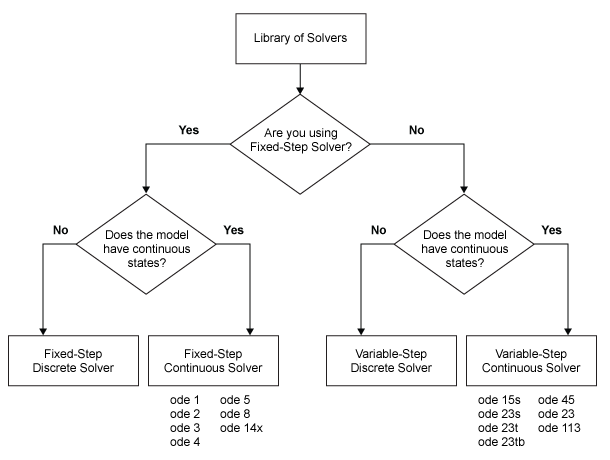


Figure 28

A variable-step solver dynamically adjusts the time step size, causing it to increase when a variable is changing slowly and to decrease when the variable changes rapidly. Thus, the solver takes many small steps near a discontinuity, e.g. a zero-crossing. Zero crossing events may be sign changes or hard stops. Overall, this behavior improves accuracy but can lead to excessive simulation times.

In my model , I have used 3 solvers and compared the result :

1. ode15s(stiff/NDS)

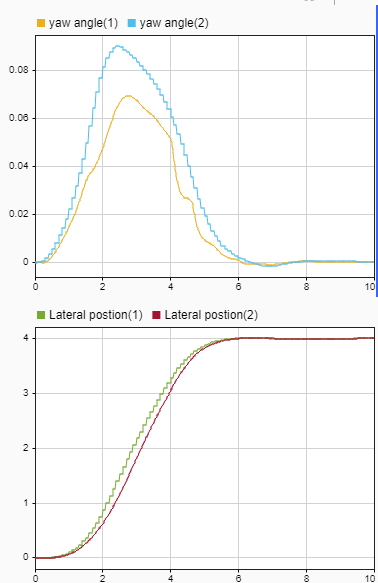


Figure 29

As we can see there are lot of disturbance in the desired response.

1. ode23(Bogacki-Shamphine)

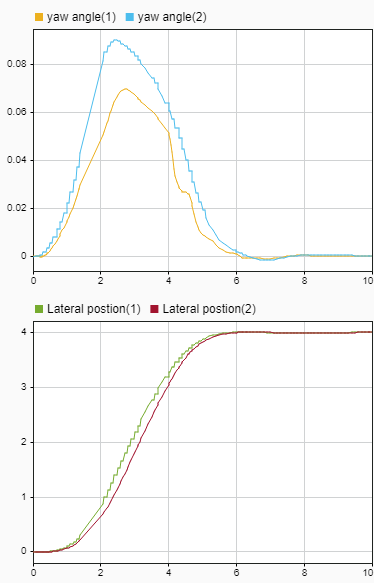


Figure 30

Here the response is better than the ode15s but still there are disturbances.

1. ode45(Dormand-Prince)

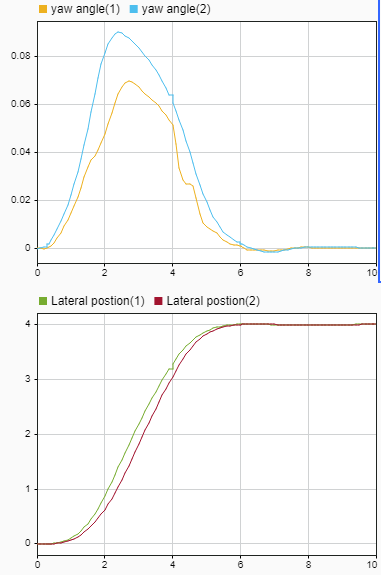


Figure 31

As you notice here that there is minimal disturbance in the desired output.

**So out of all the 3, ode45 has been proved effective for my system.**

* 1. **MATLAB function block**

In the model designed , there has been 2 MATLAB functions that has been used.

1. To Update plant model(Before Adaptive MPC) – The code inside the function is :

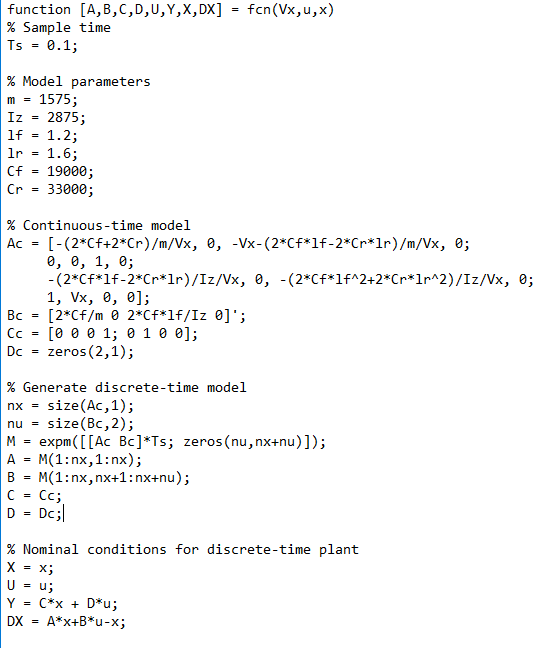


Figure 32

1. Inside the plant model(After Adaptive MPC) – The code inside the function is :

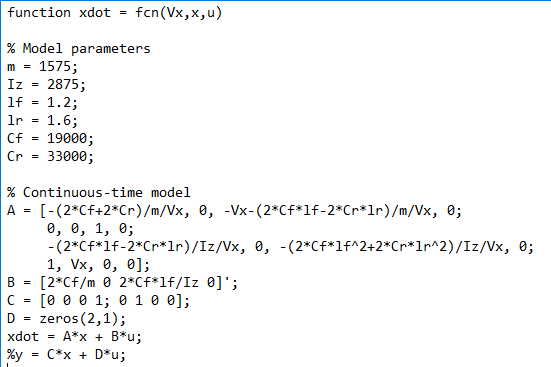


Figure 33

* 1. **Lookup Table and Signal Builder**

The lookup table and signal builder both are joined together and given as input(longitudinal velocity) to the system.

Signal builder :

The signal in signal builder is designed as shown –

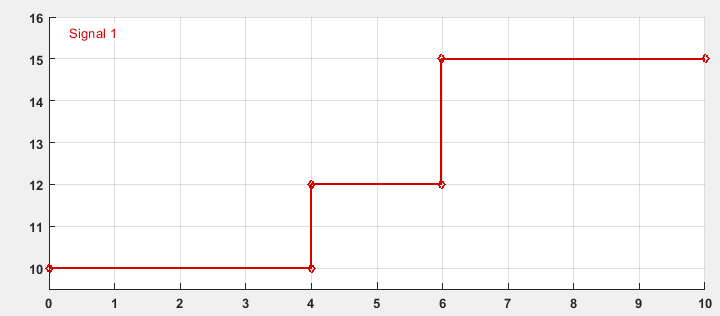


Figure 34

Now the output of the signal builder is given to a lookup table in such a way that the output of the lookup table is double the input.

The script written for lookup table is shown :



Figure 35

And the plot of the lookup table is :

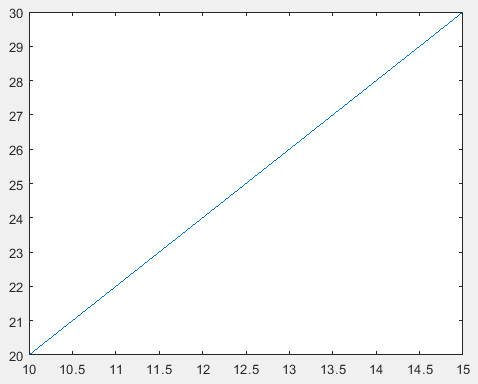


Figure 36

## References:

## Genesis – MBD Training course

## Genesis – AAEL Training Course

## Adaptive Cruise Control – Wikipedia

## “Overview and analysis of Vehicle Automation and Communication Systems from a motorway traffic management perspective” – Christiana Diakaki

1. Straits Research –

https://straitsresearch.com/report/adaptive-cruise-control-market

1. “A Framework for Modeling and Analyzing Fault-Tolerance” - Betty H.C. Cheng
2. MATLAB Documentation – Driver Design Scenario
3. “Model Predictive Control Approach to Design Practical Adaptive Cruise Control for Traffic Jam” – Taku TAKAHAMA
4. MATLAB Documentation of Adaptive Cruise Control
5. Solver choice for Simulink - <https://blogs.mathworks.com/racing-lounge/2017/12/08/solver-choice/>
6. Adaptive MPC videos by MATLAB - <https://www.youtube.com/watch?v=aQhpvrQPxD4>
7. “Development and Validation of Functional Model of a Cruise Control System” - Marcel Romijn
8. Development of Cruise Control using Rhapsody – G.R de Boer