Nat. Volatiles & Essent. Oils, 2021; 8(5): 4325 - 4339

Model Predictive controller in an Autonomous Vehicle

[1]Abhishek, [2]Dr.M.P.R.Prasad, [3]Dr.C.R.Mariappan

[1] (M.TECH in INSTRUMENTATION, Dept.of Physics NIT Kurukshetra), [2](Assistant Professor,Dept. of Electrical Engineering,NIT Kurukshetra), [3](Assistant Professor,Dept.of Physics,NIT Kurukshetra)

Abstract-

Model predictive control is a part of advanced process control that uses a collection of constraints to control a process.Model Predictive control is a multivariate technique used to solve optimization problem and predicts the future behaviour or optimum control by selecting best control action.With increase in computation and advanced microelectronics it's uses had spread over a large area of applications including automation and space technology.It had gained so much importance over PID controllers because it can candle MIMO systems in an effective way,it can handle both hard and soft constraints depending upon the uses. It also has feed forward (receding action) control or preview capability and many more advantages.

In this work along with study of MPC, lane keeping assist (LKA) feature of MATLAB is used for designing MPC controller model in an automated system scenario and then simulation is done with the help of MATLAB/SIMULINK for analysing the tracking done by the MPC controllers and then results are verified.Further scope of MPC and it's future works had also been described at last.

Index terms- Control horizon, constraints, Model Predictive control, PID controllers,prediction horizon,Quadratic cost function,

INTRODUCTION

MPC is a type of feed-back control algorithm that uses a model to make future predictions by deciding specific control actions based on optimization problem and an optimizer that ensures that the output tracks the desired reference.At each time steps optimization problem is solved to find desired control action.It finds best predictive path that is closest to the reference, so it first simulates multiple future scenario in a systematic way using model and optimizer by minimizing error between the reference and predicted path thus selecting the best control action to make specific future action.

Apart from internal dynamic model and optimizer,MPC also uses a cost function J over a receding horizon.

Quadratic cost function for optimization is generally given by:

$$J = \sum_{i=1}^{N} w_{x_i} (r_i - x_i)^2 + \sum_{i=1}^{N} w_{u_i} \Delta {u_i}^2 \qquad \text{eq.(1)}$$

where,

x_i is the ith controlled variable

r_i is the ith reference variable

$\omega_{x \cdot i}$ is the weighting coefficient with respect to x_i

$\omega_{u \cdot i}$ weighting coefficient with respect to u_i

u_i is the i^{th} manipulated variable

The MPC controller uses an objective function to minimize the deviation of the controlled variable from the set-point or reference and accordingly it plans control moves for the purpose of minimizing that objective function with the help of decision variables.

P-I-D Controller- It is a combination of the $proportional, integral\ and\ derivative$ controller.It is similar to lead-lag compensator and band reject filter.It improves both steady state as well as transient response.It reduces the rise time (t_r) and helps in increasing bandwidth and stability.It also eliminates steady state error between input and output.It increases type and order of the system by one.
The mathematical equation of PID controller is given by
actuating signal

$$e_A = Kp * e(t) + K_I * \int e(t).dt + K_D * \frac{d}{dt}e(t) \qquad \text{eq.(2)}$$

Where $_I$ = Integral gain and K_P = Proportional gain of the controller K_D= Derivative gain and $e(t)$ is the error signal

After taking Laplace transform,transfer function of PID controller can be represented as

$$T(s) = K_P + K_I * \frac{1}{s} + K_D * s \qquad \text{eq.(3)}$$

To overcome the demerits of PID controllers,MPC is used.

Main disadvantages of PID controllers is that-

1.Constraints cannot be included

2.PID controllers for multivariate systems is not straight forward.Controlling MIMO systems that have interactions between inputs and outputs requires a lot of complexities and challenges for PID controllers especially for larger systems.

So to address the limitations and disadvantages of various controllers especially PID had provided the motivation of Model Predictive Controllers.

The main motivation of MPC controllers are-

1.To address the limitations of PID

2.To make use of explicit model of the process in finding the control law

3.To include the effect of current action on future outputs,defining control horizon (m) and prediction horizon (p)

So unlike in PID controllers where control law is generally an analytical solution,here in MPC our main focus is in finding solution to an explicit optimization problem at every time step.

Therefore in MPC solution of optimization problem becomes computation of control moves (u)

MPC design parameters-

Sample Time,Ts-It is the time taken by the controller to determine the rate for the execution of controller algorithm.

If Ts is too big then the controller needs much time to react, in other words controller becomes slow to react to the disturbances fast enough and on the contrary if the sampling time, Ts is very small then reaction time of the controller becomes fast w.r.t the disturbances and reference (set point) also changes but this causes a large amount of computational efforts.Therefore it is necessary to find the right amount of sampling time,so it is generally recommended to choose Ts as

$Tr/20 \leq Ts \leq Tr/10$,

Where Tr is the rise time of open loop system response

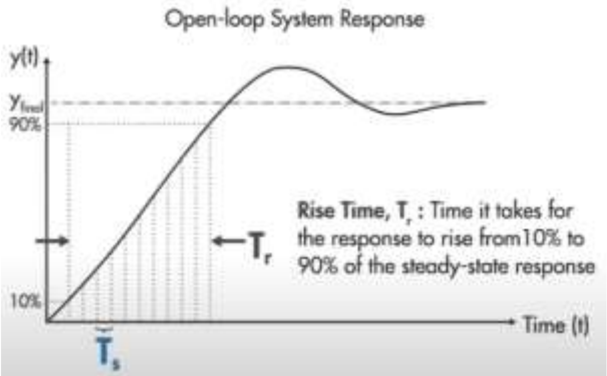

Figure 1 : Rise Time T_r

Prediction horizon-It is a measure of how far the controller can predicts into the future therefore the number of time steps to be predicted by the MPC controller is called prediction horizon

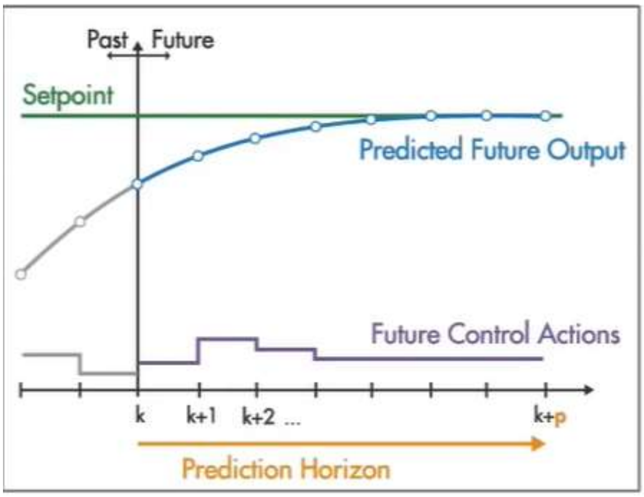

Figure 2 : Prediction Horizon range

It should be optimum i.e neither too big nor too small, so the recommened prediction horizon is approximately 20 to 30 samples covering the transient response of the open loop system as shown in the figure.

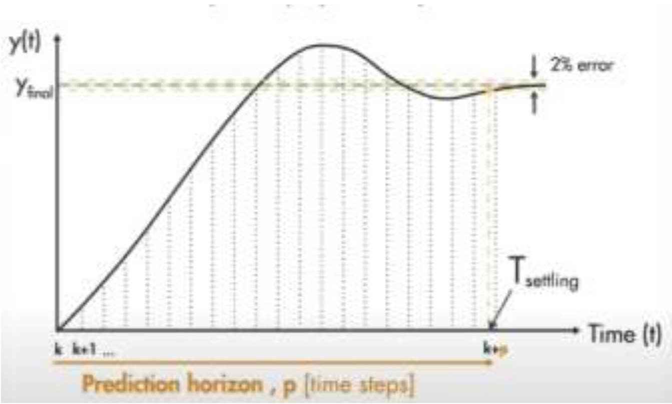

Figure 3 : Recommended prediction horizon range

Control horizon-The number of control moves required by the controller till time steps 'm' is called control horizon

The input is held constant after the m control moves. The control horizon till m is calculated so that a set of p (prediction horizon) having the predicted outputs reaches the set point in an optimal manner having minimum error between reference and predicted path thus selecting the best control action to make specific future action.

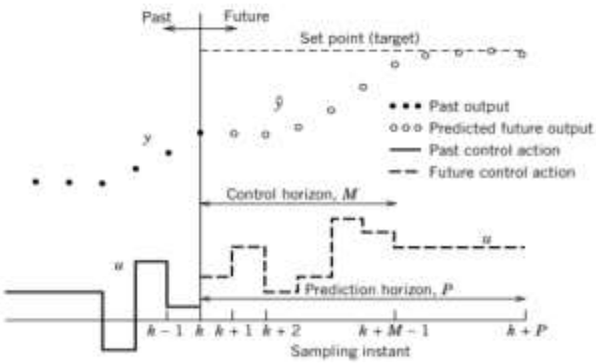

Figure 4 : Control horizon

Generally control horizon 'm' is 10% to 20% of the prediction horizon.

$0.1p \leq m \leq 0.2p$ where p : prediction horizon

Referring to the figure the first half of it represents the plot of control output (Y) and the lower half of the figure refers to the manipulated input (u) and in the x-axis we have time steps at regular intervals.

Control moves which is u(t) of MPC are predicted at pre-decided time steps which is generally of discrete in nature.So control moves are decided at each sampling time (Ts) i.e at time t = k * Ts (where k can be 1,2,3,4…. and accordingly time steps will be like Ts,2Ts,3Ts,4Ts and so on at discrete intervals).On the basis of control moves, output Y will change accordingly at various time steps.

Control moves are planned till specific time steps starting from t = k,k+1,k+2……till t = k+M-1 at each time instants where M is known as control horizon starting from t=k.Control moves are planned in such a way that the errors between each point of predicted future outputs (Y(k+t)) till Y (k+P) and the set-point (reference) at various time points as minimum or as small as possible like set-point tracking.

Where p=prediction horizon of MPC.

So control moves u(t) at t=k, k+1, k+2,k+3…..and so on till t = k+M-1 will predict outputs, Y at time starting from t= k+1 till the prediction horizon (P) i.e t = k+P of MPC at each instants where predicted output must be as close to set point which implies that error should be minimum.Therefore the present change in control move (say at u(t=k)) will effect the output Y having some delay i.e at t = k+1 which is a feature of dynamic system and also instantaneous effects are not possible.

Constraints

MPC can incorporate constraints on the input and as well on the output and that too in a systematic way,that is it can ensure optimum input and can anticipate proper output using constraints.

Based on this there are broadly 2 types of constraints-

Soft constraints can be violated and there is a relaxation in this type of constraints depending upon the uses and need whenever required whereas,

Hard constraints must be ensured that it cannot be violated and it is used when there is need of it.

Generally output constraints are kept soft and hard constraints are avoided on the input side and rate of change of inputs.

Weights

Depending upon the priorities and goals MPC can assign weights. Since MPC has multiple goals, so it sets weights to each goals depending upon the priorities.Largest weight is assigned to the goal having highest priority, which is needed to be accomplished first.

Making MPC faster

Since due to increase in number of states, number of constraints, length of control and prediction horizon complexities of MPC increases which tends to make MPC slower.Since MPC problem is generally a quadratic programming (QP) problem that solves optimization problem at each time step which makes MPC more complex.Such complexities and slow speed of MPC is a big problem especially for industries where high computational efforts are required and on the same type fast dynamics are also required where sampling time Ts is generally in milliseconds.

Another challenge in MPC is that with increase in number of optimization variables dimensions of matrices used in MPC also increases which makes it a storage concern.So it needs to be rectified.Optimally we need high throughput of MPC with reduced memory space.

We can run controller of MPC faster by-

Model order reduction technique where the states that don’t contribute to the dynamics of the system are removed.

Shorter prediction and control horizon

We can also reduce the number of unnecessary constraints along with minimizing precision and repeated date usage.

For more lower sample time Ts we can use EXPLICIT MPC to increase the effectiveness of Model Predictive Controller.

Explicit MPC solves optimization problem within a given range offline to reduce the complexity of operation.It pre-computes the optimum solution for each value of independent variable for a mentioned range of operation in an offline way.Generally the solution consists of linear piecewise functions that are continuos in nature which in turns are divided into regions to get mapped and unique optimal solution for each region.For one state as shown in figure

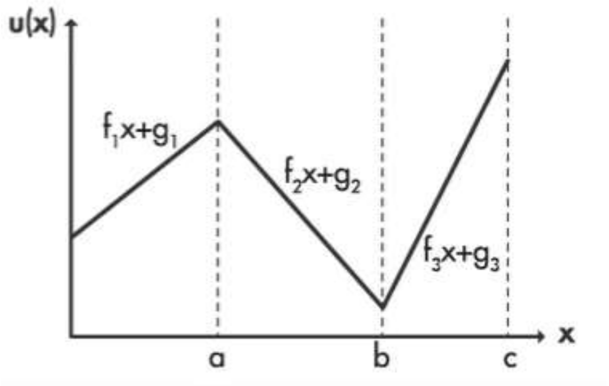

Figure 5 : mapping of linear piecewise functions to get unique optimal solution in Explicit MPC

This idea can also be expanded to a two state problem.

After finding offline solution Explicit MPC becomes online to find current state region and then evaluate previous linear functions to predict the control action for future.This also reduce unnecessary iterations which had already been calculated offline to reduce complexities.

If there are many regions and states then it is very complex for MPC to solve for optimum solution and also it is very time taking to find present states out of many states.Also large memory is required for this.

So we can use SUB-OPTIMAL solution.In this we can assign maximum number of counts of iteration and when the controller is reached to that count then the controller stops and finds sub-optimal solution upto that iteration.Despite being a sub-optimal solution it can still satisfy all constraints.It also helps in reducing unnecessary iterations.Maximum number of iteration is chosen in such a way that it satisfies-

Execution time per iteration × Maximum number of iterations = Controller sample time

Solution execution time is always less than the sample time Ts of MPC and apart from this some extra time must also be left within Ts to perform other tasks.

Generally in MPC input and output variables are referred as manipulated variables (MV's) and controlled variables (CV's) respectively and disturbances are denoted by disturbance variables (DV's)

Controllers

Controller is a type of scheme which monitors and then transforms the system parameters to a desired value of output. The controller can be used to check the accuracy, reliability, and stability of the system. According to the requirements and performance specified controllers can be connected in parallel or series to the plant. A controller is used to reduce the error which is equal to the difference between the actual value and the desired value (set point) of the system that is to minimize the error to zero.

Similarly MPC is a type of Advanced process control which first simulates multiple future scenario in a systematic way using model and optimizer by minimizing error between the reference and predicted path thus selecting the best control action to make specific future action.

MPC formulation-

The optimization problem of MPC has 3 components generally-

Objective function

Constraints

Decision variables

Objective function which can be scalor or vector (for multi-objective optimization problem) function.It is basically of the form of scalor function like f (x1,x2,x3,x4……xn) where x1,x2,x3,x4…..xn are the n decision variables which is to be optimized.Decisions should be made about these decision variables such that the function f (x1,x2…xn) is either maximized or minimized i.e function should be optimized by finding optimum solution.So for minimization of objective problem we have to minimize the error for output y(k+i) at each time steps and with respect to the set-point y^{sp} like in the equation given below

$$0 = \sum_{i=1}^{P}[y(K+i) - ysp]^2 \qquad \text{eq.(4)}$$

Where y(k+i) – y^{sp} is the i^{th} error (e_i).

The error is squared to cancel the effects of positive and negative errors and to compute that all errors are actually going down to zero (0), instead of cancellation by negative and positive errors.

Constraints can be applied on both inputs (u) and outputs (y) but generally they are applied on the input side.

$$U^L \leq U \leq U^U \qquad \text{eq.(5)}$$

$$\Delta U^L \leq U \leq \Delta U^U \qquad \text{eq.(6)}$$

Where 'u' and 'Δu= u (k) – u (k-1)' and so on are the control inputs and change in control inputs respectively at every instants of time and the superscripts L and U means lower limit and upper limit of control inputs.

Soft constraints can be violated whereas hard contraints cannot be violated so these contraints are used accordingly whenever it is required.

Decision variables (like x1,x2,x3,x4……xn) are the variables for which we have to make the control move plan like u(k), u(k+1), u(k+2), ….u(k+M-1) such that objective function is optimized and error between outputs (y) at each instants and y^{SP} (set-point) is minimum i.e minimization of error.

So overall objective function should be like

$$\min_{U} \sum_{i=1}^{P} (y^{sp}[k+i] - \hat{y}[k+i])^2 + \sum_{i=1}^{M} u[k+i-1]^2$$

eq.(7)

MPC algorithm-

Step 1: Choose a control horizon M,such that M<P (since control moves affect only in future),where P is called as prediction horizon

Step 2: Lets assume that we are at time t = K

Step 3: Since we need variables which are to be optimized therefore seek outputs at each time step like y(K+1),y(K+2)…………y(K+P) as close to set-point y^{sp}.

Step 4: Write an objective function

Step 5: Minimize objective function "O" by manipulating u(K),u(K+1),u(K+2)……..u(K+M-1) with u(K+m) = u(K+M-1), where m is M,M+1…..P where P is prediction horizon of the controller.

Step 6: Get optimum control moves with the help of optimizer and controller as u*(K),u*(K+1),u*(K+2)……u*(K+M-1) at time t = "K" which is the present time for now and this information is further used to do analysis,evaluation and the to process move plans at next time steps also i.e at t=K+1,K+2……till K+M-1 which also makes controller less aggressive.

Step 7: Implement control move u*(K) and execute it.

Step 8: Next move to t = K+1 and similarly repeat the process

SYSTEM DETAILS AND SIMULATION

This work uses MATLAB/SIMULINK along with MPC designer app to design and present Model Predictive Controller that steers a car in an autonomous way in a lane change maneuvering situation or scenario.

Overall model and SYSTEM design can be represented as :

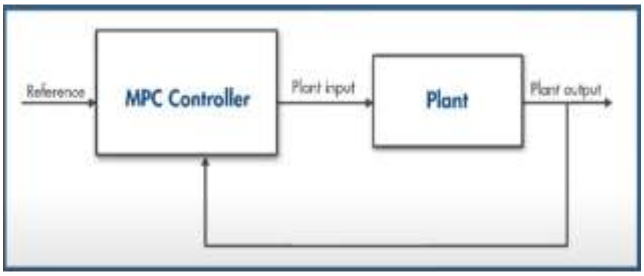

Figure 6 : Closed loop model of MPC controller

Components of the MODEL :

Plant MODEL : Since we are using autonomous vehicle for MPC controller design so it is therefore our plant model.In this work plant model are defined using vehicle dynamics with the help of state space equation where input is steering angle and output are lateral position and Yaw angle.

Design parameters of plant model are as follows

V_y : lateral velocity

V_x : longitudinal velocity

(X,Y) : Vehicle's global location or position

ψ : Yaw angle

δ : Front steering angle

Y_{ref} : lateral position with respect to horizontal axis

ψ_{ref}: Yaw angle with respect to horizontal axis

Let us first assume constant V_x : longitudinal velocity as 15 m/sec and then we can change according to our requirements

The vehicle model can be represented as:

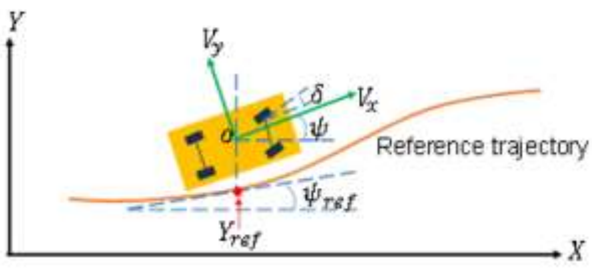

Figure 7 : The lateral dynamics of motion

The lateral equations of motion can be represented by state space matrix equation as :

PARAMETERS	VALUES
C_{af}	19000 lbs/rad
C_{ar}	33000 lbs/rad
I_Z	2875 kg m^2
I_F	1.2000 kg m^2
I_R	1.6000 kg m^2
m	1575 kg
V_X	15 m/sec

$$\frac{d}{dt}\begin{Bmatrix} y \\ \dot{y} \\ \Psi \\ \dot{\Psi} \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2(C_{af}+C_{ar})}{mV_x} & 0 & -V_x - \frac{2(C_{af}l_f+C_{ar}l_r)}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2(C_{af}l_f+C_{ar}l_r)}{l_zV_x} & 0 & -\frac{2(C_{af}l_f^2+C_{ar}l_r^2)}{l_zV_x} \end{bmatrix} \begin{Bmatrix} y \\ \dot{y} \\ \Psi \\ \dot{\Psi} \end{Bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2C_{af}l_f}{I_z} \end{bmatrix}\delta$$

eq.(8)

Global Position Y can be written as

$$\dot{Y} = V_x\Psi + V_y$$ eq.(9)

Table 1: Defining values of parameters used in defining matrix

Where, state matrix A =

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2(C_{af}+C_{ar})}{mV_x} & 0 & -V_x - \frac{2(C_{af}l_f+C_{ar}l_r)}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2(C_{af}l_f+C_{ar}l_r)}{l_zV_x} & 0 & -\frac{2(C_{af}l_f^2+C_{ar}l_r^2)}{l_zV_x} \end{bmatrix}$$ eq.(10)

C_{af}, C_{ar} are cornering stiffness

l_f longitudinal distance from center of gravity to front tires

l_r longitudinal distance from center of gravity to front tires

$V_x$vehicle longitudinal speed or velocity

Ψ yaw angle of the vehicle

δ front steering angle

m total net mass of vehicle

I_z Yaw moment of inertia of vehicle

For the given values of parameters as mentioned below in the table we will calculate system matrix [A] and other matrix [B],[C] and [D],

State matrix [A] on rearranging as shown on MATLAB=

$$\begin{bmatrix} -4.4021 & 0 & -12.4603 & 0 \\ 0 & 0 & 1 & 0 \\ 1.3913 & 0 & -5.1868 & 0 \\ 1 & 15 & 0 & 0 \end{bmatrix} \quad \text{eq.(11)}$$

Matrix [B]= $$\begin{bmatrix} 24.1270 \\ 0 \\ 15.8609 \\ 0 \end{bmatrix} \quad \text{eq.(12)}$$

Matrix [C] = $$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{eq.(13)}$$

Matrix [D] = $$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{eq.(14)}$$

Therefore Plant Model is a state-space block defined by above equations having the mentioned values of matrix [A],[B],[C] and [D] as given

Reference block : For reference with the help of custom reference trajectory we will use Driving scenario Designer application of MATLAB automated driving system toolbox.We must also design waypoints for the car trajectory to generate the lane change manoeuvre.

MPC Controller

We will define and linearize MPC structure using MATLAB feature.We will set various parameters like Prediction horizon p,control horizon m,Sample time T_S, measured output,manipulated variables,weights,constraints etc according to our requirements.

After define and linearize the app imports and linearizes the plant which is autonomous car, from the Simulink model and uses it as a internal plant model.We can see the input and output responses of the system.Here the output of the MPC controller is steering angle (δ) which is used to control lateral position (Y) and Yaw angle (ψ) of the plant output. Here the plant model is autonomous car or vehicle.

Arrange all blocks to get final model

Final simulation model will look like

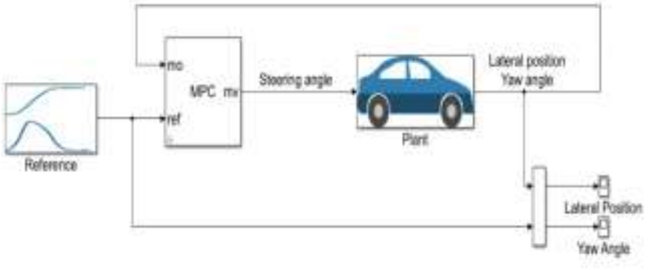

Figure 8 : Final Simulation Model

OUTPUTS

As we can see from the outputs that we get satisfactory tracking of both the outputs which are lateral position and Yaw angle.In each plot one graph represents reference and other shows tracking done by MPC.Both are nearly same, it means tracking done by MPC is good.

Also the steering angle plot is within the limits of constraints.

We can Adaptive model predictive controller that will automatically update the internal plant model for different operating condition and velocities.

OUTPUT for lateral position (Y)-

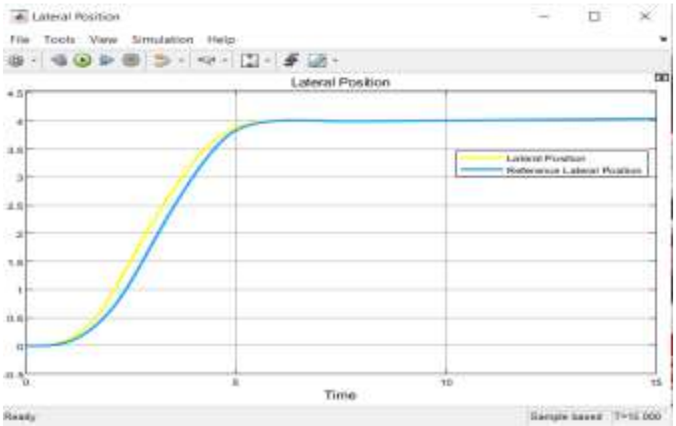

Figure 9 : Plot for lateral position (Y)

OUTPUT for yaw angle (Ψ)

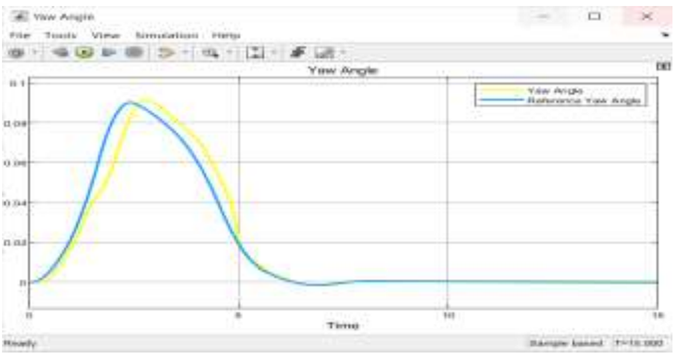

Figure 10 : Plot for Yaw Angle (Ψ)

For both lateral position and Yaw angle plot of reference (blue clour) and plot by MPC tracking (Yellow colour) are nearly same, it means tracking done by MPC is good

Plot of steering angle-Here we can see that plot of steering angle is within the specifies constraints and contraints on steering angle had been applied here which is equal to $\pm 30^0 = \pi / 6$ radian = 0.52 and also after some initial disturbance steering angle is finally set to steady state value equal to 0.

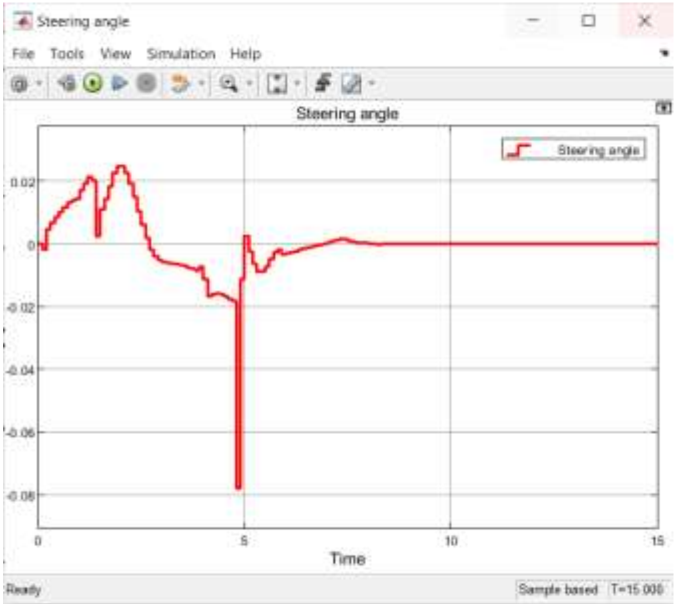

Figure 11 : Plot of steering angle

CONCLUSION

This work shows analysis and study of MODEL PREDICTIVE CONTROLLERS supported by MATLAB/SIMULATION to implement MPC using an example of autonomous driving system.Various parameters such as sampling time,control horizon,prediction horizon,sampling time etc. had been studied in this work.

This work shows how at each time steps optimization problem is solved to find desired control action and how MPC finds best predictive path that is closest to the reference by simulating multiple future scenario in a systematic way using model and optimizer, by minimizing error between the reference and predicted path thus selecting the best control action to make specific future action.

SIMULATION RESULTS-As we can see from the outputs that we get satisfactory tracking of both the outputs which are lateral position (Y) and Yaw angle (Ψ).In each plot one graph represents reference and other shows tracking done by MPC.Both are nearly same, it means tracking done by MPC is satisfactory.Also for steering angle (δ) also after some initial disturbance steering angle is finally set to steady state value equal to 0 and plot of steering angle is within the specified constraints.So results have been verified

This also present how MPC is even better than PID controllers, so to address the limitations and disadvantages of various controllers especially PID had provided the motivation of Model Predictive Controllers.Also constraints cannot be included in MPC and it is very effective in MIMO systems also.

MPC achieves its objective by focusing on finding solution to an explicit optimization problem at every time step.

FUTURE WORKS

With the advancement in explicit MPC's we can even make MPC more faster and advanced by reducing its sampling time and by modifying its working algorithm.

A formal stability analysis to study on even better, performance guarantee would be a welcome advance

More advancements in processors and memory using Embedded MPC systems are possible. Economic MPC's, distributed MPC's can be used in in different areas of applications in an efficient and an optimum way.

This work can be extended for Linear and Nonlinear MIMO systems with different controllers such as PID,other advanced process control,sliding mode controller,Linear Quadratic regulators (LQR), Fuzzy logic controller, Artificial Neural Networks (ANN). etc.

References

[1] Practical Design and Application of Model Predictive Control MPC for MATLAB® and Simulink® Users by Nassim Khaled (Controls and Engineering Systems Manager, Dover, Atlanta, Georgia) and Bibin Pattel (Solution Architect, KPIT Infosystems Inc, Columbus, Indiana, USA) ,Butterworth-Heinemann (imprint of Elsevier) publication.

[2] The MPC Simulink Library . User's Guide Version 1,COPYRIGHT 2000 by A. Bemporad, M. Morari, and N. L. Ricker

[3] The MathWorks, Inc.,1994-2021

[4] G.E. Stewart, F. Borrelli, J. Pekar, D. Germann, D. Pachner, D. Kihas, Toward a systematic design for turbocharged engine control, in: L. del Re, F. Allgower, L. Glielmo, C. Guardiola (Eds.), "Automotive Model Predictive Control", Lecture Notes in Control and Information Science, Springer-Verlag, Berlin-Heidelberg, 2010.

[5] N. Khaled, J. Pekar, A. Fuxman, M. Cunningham, O. Santin. Multivariable control of dual loop EGR diesel engine with a variable geometry turbo. 2014-01-1357, SAE World Congress Conference, Detroit, 2014.

[6] Suresh, D. Langenderfer, C. Arnett, M. Ruth, Thermodynamic systems for Tier 2 Bin 2 diesel engines, SAE Int. J. Eng. 6 (1) (2013) 167-183. Available from: https://doi.org/ 10.4271/2013-01-0282.

[7] J. Shutty, R. Czarnowski. Control strategy for a dual loop EGR system to meet Euro 6 and Beyond. Deer Conference, August 3-6, Dearborn, Michigan, 2009.

[8] F. Millo, P.F. Giacominetto, M.G. Bernardi, Analysis of different exhaust gas recirculation architectures for passenger car Diesel engines, Appl. Energy 98 (1) (2012) 7991.

[9] M. van Aken, F. Willems, D. de Jong. Appliance of high EGR rates with a short and long route EGR system on a heavy duty diesel engine. SAE Technical Paper 2007-01-0906, 2007, https://doi.org/10.4271/2007-01-0906.

[10] M. Zheng, G.T. Reader, J.G. Hawley, Diesel engine exhaust gas recirculation-a review on advanced and novel concepts, Energy Conv. Manage. 45 (1) (2004) 883-900.

[11]J. Shutty, H. Benali, L. Daeubler, M. Traver. Air system control for advanced diesel engines. SAE Technical Paper 2007-01-0970, 2007, https://doi.org/10.4271/2007-01-0970.

[12]O. Grondin, P. Moulin, J. Chauvin. Control of a turbocharged Diesel engine fitted with high pressure and low pressure exhaust gas recirculation systems. Proceedings of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference, 6582-6589, 2009, https://doi.org/10.1109/CDC.2009.5400922.

[13]F. Yan, J. Wang. Control of dual loop EGR air-path systems for advanced combustion diesel engines by a singular perturbation methodology. Proceedings of the American Control Conference, 1561-1566, 2011.