Safe Driving Using Model Predictive Control For Autonomous Vehicles

Adhavan Jayabalan, Janani Mohan, Kritika Iyer, Sathya Narayanan Kasturi Rangan

***Abstract*—Situations that require deviation from the desired path present a challenge for traditional path tracking controllers because they are not given sufficient information about where it is safe to deviate from the path. The controller presented in this paper uses environmental information and a vehicle model to ensure that, should deviations from the path be necessary to avoid obstacles, the vehicle will deviate in a path that is safe. In this paper, we propose a strategy for path tracking and obstacle avoidance using Model Predictive Control.**

***Keywords*—Autonomous vehicles, Model Predictive Control, Obstacle Avoidance**

I. INTRODUCTION AND BACKGROUND

A number of new technologies are empowering modern vehicles like never before. Sensing technologies like radar,cameras, and laser systems provide rich information about the environment in real-time. Road surface friction estimates and vehicle states can be made available in real-time in vehicles equipped with steer-by-wire or power-steering as demonstrated by Hsu et al. (2010). In addition, steer-by-wire provides increased actuation capabilities. These new technologies have the potential to make cars much safer.Vehicles traversing dynamic and static environments can be subjected to many different types of obstacles. To avoid these obstacles, the vehicle will need to swerve. The obstacle avoidance of these vehicles, however, must also be safe. For this purpose, some control algorithm must be applied to ensure that the vehicles move safely around the obstacle and maneuver in safe regions. The predictive nature and constraint handling capabilities of Model Predictive Control (MPC) make it an attractive framework for leveraging these new technologies.MPC is highly popular as it yields higher performance control without expert intervention for long periods of time [1]. Both Linear and Non-Linear variants of MPC can be applied to any problem in the autonomous driving scenario Kawabe et al. (2004) present a receding horizon control framework that uses information about the surrounding environment to generate optimal paths to help guide a human driver.Falcone et al. (2007) present a reduced computation MPC scheme for trajectory tracking using active steering actuation.

In MPC [2] at each sampling time step, starting at the current state, an open loop optimal control problem is solved over a finite horizon.The optimal command signal is applied to the process only during the following sampling interval. At the next time step, a new optimal control problem based on new measurements is solved over a shifted horizon. A dynamic model of the process respects input and output constraints, and minimizes a performance index.MPC is formulated in the state space form and the system is a linear discrete time model [3].Because of its capability of systematically handling nonlinear time-varying models and constraints, and operating close to the limits of admissible states and inputs, Model Predictive Control (MPC) has been widely used to address the autonomous vehicle guidance problem [12][16]. In [13],the MPC problem has been formulated as a quadratic program (QP) by limiting the intervention to the steering, and linearizing the vehicle dynamics around a constant vehicle speed and small slip angles. In [14][16], the authors address the problem of integrated braking and steering control by using a hierarchical control architecture. A high-level controller generates an obstacle-free trajectory, while a low level controller tracks this planned trajectory. In order to combine braking and steering, both levels implement a nonlinear MPC formulation which requires the on-line solution of a non-convex optimization problem. In order to reduce the real-time computational complexity, in [12], the authors have proposed the use of a spatial vehicle model which simplifies the problem. However, the nonlinear nature of the model used in the MPC problem significantly limits the maximum prediction horizon implementable.

In this paper, we propose a linear MPC-based control architecture suitable for vehicles driving in low curvature roads, such as highways. It addresses the lane keeping and obstacle avoidance problems by combining steering and braking actions [4]. Model Predictive Control have been used for speed control and torsional vibration suspension [5], variable environment for ecological driving [6], adaptive cruise control [7] and vehicle stabilization [8].The MPC that is proposed here gives a trajectory using position and velocity discrete model while taking into consideration the stability and safety of the vehicle. It is similar to [9] but the obstacles will be modeled as potential field functions [10].

In Section II, we explain preliminaries design and assumptions undertaken for this project.In Section III, we define the problem statement that is solved through our proposed solution. Section IV gives an overview of the variables used in this paper. Section V explains the methodology and our approach used in this project. Section VI discusses the experiments that we conducted both in simulation. It also shows the results that was obtained and its significance to the work. Finally in Section VII, we conclude by summarizing the project accomplishments, identifying potential problem areas and future prospectives for the project.In Section VIII, the reachable goals and desired goals are formulated and the achieved goals are highlighted.

II. PRELIMINARIES AND ASSUMPTIONS

A. Vehicle Model

In this project, we assume the car is modeled as a 3DOF bicycle. Due to this assumption for the remainder of this project we will be using a bicycle model as the vehicle model. Since we are testing this for roads with very little curvature,

the banking angle is assumed to be 0. We always keep the rear steering angle to zero and take inputs as front steering angle. There exists two approaches to model the vehicle as a bicycle namely Kinematic Bicycle Model and Dynamics Vehicle Model.
A detailed study of the various vehicle models and the controllers used for autonomous vehicles has been conducted.Most of published MPC schemes use dynamic vehicle model combined with linear tire model [3]. This approach has two main disadvantages: it is computationally expensive and any tire model becomes singular at low vehicle speed. Tire models use a tire slip angle estimation term which has the vehicle velocity in the denominator. This prohibits the use of the same control design for stop-and-go scenarios, common in urban driving [3].The kinematic bicycle model, on the other hand, is relatively similar to design and implement as it considers only the front steering angle and the acceleration. In the case of dynamic bicycle model, the tire slip angle and tire cornering stiffness are also considered. Dynamic bicycle model is primarily used in drastic road change conditions.Since most of the real road conditions are based on nonlinear bicycle model, we have done a comparison of the Kinematic Bicycle Model and Dynamic Bicycle Model in this project.Moreover, Model Predictive Control requires an analytical and accurate dynamic mathematical model of the system to be controlled. Our current implementation uses a nonlinear model of vehicle dynamics to predict the future trajectory of system states.The project further aims to develop the controller for obstacle avoidance as well.

B. Controller Design

A simple trajectory tracking problem usually is well handled with the use of LQR control. In this project , since we plan to handle lane keeping as well as the obstacle avoidance problem of a vehicle, we propose to use a Model Predictive Controller. A very general architecture of a Model Predictive Controller is given in Fig. 1. MPC controller contains three basic functional blocks. The Optimizer finds the optimal control input u(t) which when applied to the plant, gives the minimum value of the cost J. Of course, this optimization must be done in the presence of the constraints and the cost function.The State estimator is used to predict unmeasured states $\hat{x}(t)$ form the plant.Model Predictive Control optimizes the output of a plant over a finite horizon in an iterative manner as in Fig. 2.

Suppose the step size of the controller is T. At time step k the current plant state is sampled and the optimizer computes a cost minimizing control strategy u(t) for finite time steps in future k = t + 0T, t + 1T,.....,t + pT where p is the number of look ahead prediction horizon steps. In practical situations, the whole optimal control sequence cannot be applied to the process. This is due to the inaccurate process model and added disturbances in the process which can cause error between the predicted output and the actual process output. If only a mathematical model was used for prediction and state calculation, prediction errors could accumulate. Thus only the first step of the control strategy is applied to the plant and the plant state is measured again to be used as the initial state for the next time step. This feedback of the measurement information to the optimizer adds robustness to the control.The plant state is sampled again and the whole process is repeated again with the newly acquired states. The prediction time window[t + 0T,t + 1T,.....,t + pT] shifts forward at every time step (reason why MPC is also known as Receding Horizon Control).

C. Assumptions

- The environment around the vehicle is completely observable.
- The environment currently has only static obstacles.
- Currently,the control input is steering angle.
- The lane banking is assumed as zero.
- The location of the obstacles in the environment is known in advance.

III. FORMAL PROBLEM STATEMENT

In an autonomous vehicle, it can be useful to divide motion control into path planning and path tracking. In this structure, the path planner uses data from perception systems to generate the desired path (desired positions and orientations) for the vehicle to follow. The path tracker then calculates and applies steering action to guide the vehicle along the path. This path tracker is not effective in a real-world scenario, as it may not have the sufficient perception information to handle the changing road conditions. Even if a path planner has access to

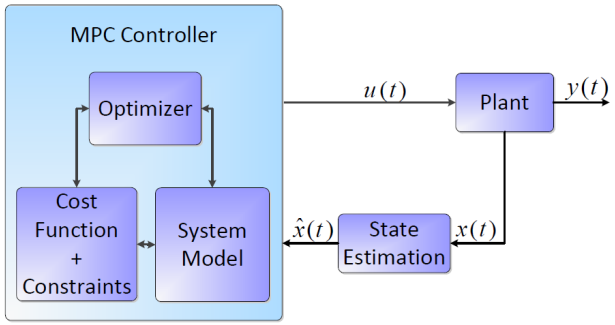

Fig. 1: Model Predictive Control Architecture

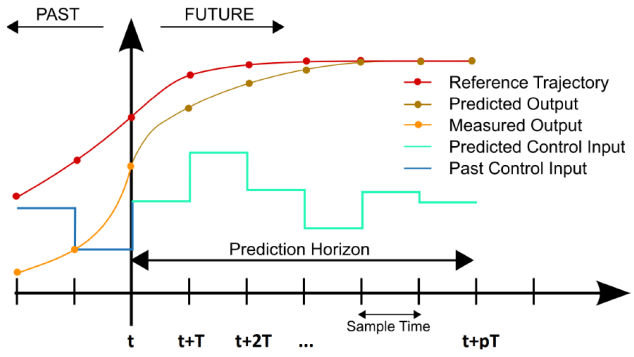

Fig. 2: Model Predictive Control Scheme

this data, it is challenging to encode all information needed to safely navigate the environment as a single path. If the path cannot be perfectly tracked, a path-tracking controller should possess information about where it is safe to deviate from the path, such as within a lane, and where it is not next to another vehicle. Tackling obstacle avoidance and lane keeping is the problem addressed in this project.

IV. LIST OF VARIABLES

F_{yr}, F_{yf} = Force in the y direction on the rear,front wheel
N = Number of Prediction Horizons
m = Mass of the vehicle
a_y = acceleration in y direction
F_{bank} = Banking force
y = y coordinates of center of mass
x = x coordinates of center of mass
ϕ = Yaw angle
V_x = Longitudinal Velocity
V = Velocity
$V_x\dot{\varphi}$ = Centripetal Acceleration
β = Slip angle
T_s = Time step
δ_f, δ_r = steering angle of front, rear wheel
η =reference trajectory
u = input
C_f, C_r = stiffness co-efficient front, rear
J = cost function
γ = Set of variables that J is dependent on

V. METHODOLOGY

In this project, a comparison between the kinematic and dynamic bicycle model is done. The following sections elaborates the vehicle model and the controller implementation in each case.

A. Kinematic Model

In this project we implement a car model as a Kinematic bicycle model as shown in Fig. 3 and hence the equation for the model can be written as,

$$\dot{x} = vcos(\psi + \beta) \tag{1}$$
$$\dot{y} = vsin(\psi + \beta) \tag{2}$$
$$\dot{v} = a \tag{3}$$
$$\dot{\psi} = \frac{v}{l_r} sin\beta \tag{4}$$

Fig. 4 shows the control architecture that will be implemented in this project.

1) Discrete Time Linearized kinematic model: The model constitutes the equations of motion of the vehicle, and has three states (x, y, and ψ) and one input, the steering angle (δ), since we assume a constant velocity. Discretized representation of the kinematic bicycle model sampled at sampling time of T_s gives

$$x_{k+1} = x_k + T_s v_k cos(\psi_k + \beta_k) \tag{5}$$
$$y_{k+1} = y_k + T_s v_k sin(\psi_k + \beta_k) \tag{6}$$
$$\psi_{k+1} = \psi_k + T_s \frac{v}{l_r} sin\beta_k \tag{7}$$

where

$$\beta_k = tan^{-1}\Big(\frac{l_r}{l_r + l_f} tan\delta_{k-1}\Big) \tag{8}$$

Linearizing the kinematic bicycle model, we form the Jacobian for matrices A, B and evaluate them at time $t = k$ around the current state $\psi = \psi_k$, and current input $\delta = \delta_{k-1}$ (δ_k is to be determined at time k):

$$A = \begin{bmatrix} 1 & 0 & -T_s v_k sin(\psi_k + \beta_k) \\ 0 & 1 & T_s v_k cos(\psi_k + \beta_k) \\ 0 & 0 & 1 \end{bmatrix}, \text{or} \tag{9}$$

$$A = \begin{bmatrix} 1 & 0 & -T_s v_k sin\Big(\psi_k + tan^{-1}(l_q tan\delta_{k-1})\Big) \\ 0 & 1 & T_s v_k cos\Big(\psi_k + tan^{-1}(l_q tan\delta_{k-1})\Big) \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$B = \begin{bmatrix} -T_s v_k sin(\psi_k + \beta_k) \dfrac{l_q}{l_q^2 sin^2\delta_{k-1} + cos^2\delta_{k-1}} \\ T_s v_k cos(\psi_k + \beta_k) \dfrac{l_q}{l_q^2 sin^2\delta_{k-1} + cos^2\delta_{k-1}} \\ \dfrac{T_s v_k}{l_r} cos(\beta_k) \dfrac{l_q}{l_q^2 sin^2\delta_{k-1} + cos^2\delta_{k-1}} \end{bmatrix} \tag{11}$$

where $l_q = \dfrac{l_r}{l_r + l_f}$

Now we can express the linear model as

$$z_{k+1} = Az_k + Bu_k \tag{12}$$

where

$$z_k = \begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} \tag{13}$$

and

$$u_k = \delta_k \tag{14}$$

2) Optimization problem: The optimization problem is formulated at time t as follows:

$$\text{minimize} \sum_{k=0}^{N-1} (z_k - z_k^{ref})^T Q (z_k - z_k^{ref}) + u_k^T R u_k \tag{15}$$
$$\text{subject to } z_{k+1} = A_k z_k + B_k u_k \tag{16}$$
$$\delta^{min} \leq \delta_k \leq \delta^{max} \tag{17}$$
$$z_k^{ref} = (x_k^{ref}, y_k^{ref}, \psi_k^{ref}) \tag{18}$$
$$z_0 = (x_t, y_t, \psi_t) \tag{19}$$

where A_k or B_k are the Jacobian obtained via linearization around the state of the vehicle at time t, in which case they are constant across all k's. ($A_k = A$, $B_k = B$, $\forall k$) and the problem is solved for every time step [k + 0T,k + 1T,.....,k + NT].

3) Trajectory Generation: For testing the Model Predictive Controller, we are implementing two trajectories: a) Straight Line Trajectory and b) Circular Trajectory. In the Straight line trajectory, the vehicle from any initial state has to track and converge to the reference trajectory. The state of the vehicle at each time step is calculated using the system dynamics and the optimal control input obtained from the optimization problem is applied at each prediction horizon to drive the system to the final state. In the circular trajectory generation, a circle of constant radius is the reference trajectory for the controller to track. The steering angle to the vehicle is varied at each time step ensuring the vehicle is along the track of the circle.

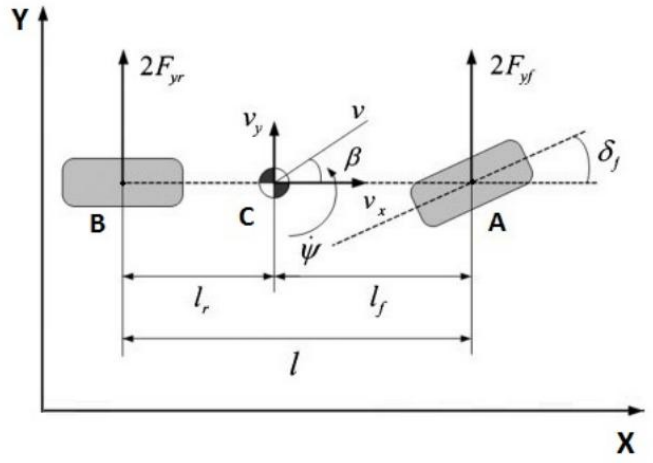

Fig. 3: Bicycle Model

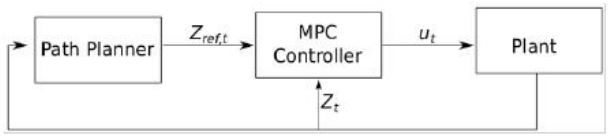

Fig. 4: Control Architecture

B. Dynamic Model

The most important requirement for Model Predictive Controller is the plant model. The optimizer uses this model to predict the future behaviors of the plant and plans a safe and optimal trajectory for the plant to follow. Linearized plant models, which are an approximation of the non-linear models can be easily implemented. In certain cases like non-holonomic vehicles, these linearized models cannot accurately model the highly dynamic nature of the plant. Hence, in order to capture the dynamics of the plant accurately, we are implementing a non-linear bicycle model to predict the future trajectory of the system states.

In a bicycle model, front left and front right wheels are joined together and represented by point A as shown in Fig. 3. The rear left and rear right wheels are joined together and represented as B. The Center of Gravity is represented by point C. The distance CA is represented by l_f, distance BC is represented by l_r and entire length of the vehicle is given by $l = l_f + l_r$. The vehicle's location is given in a $X - Y$ co-ordinate system. The steering angle is given by δ_f. It is also assumed that co-efficient of friction between the tire and road is small and the weight transfer function between the front and the rear wheel is negligible which means that the body roll and the pitch behavior of the vehicle can be neglected. The orientation of the vehicle is given by φ in the $X - Y$ co-ordinate frame and it is measured with respect to the global X co-ordinate frame. The velocity v of the vehicle makes angle β with the longitudinal axis of the vehicle. β is the vehicle slip angle. v_x is the velocity of the vehicle along the longitudinal axis and $\dot{\phi}$. F_y and F_r are the lateral forces along the Y direction. Now, the lateral dynamics of the vehicle along the Y direction can be derived by applying Newton's Second law of motion as follows.

$$ma_y = \Sigma(\text{forces acting along Y axis}) \tag{20}$$
$$ma_y = 2F_{yf} + 2F_{yr} + F_{bank} \tag{21}$$

We assume that the for due to the banking angle of the road as zero.

$$F_{bank} = 0 \tag{22}$$
$$ma_y = 2F_{yf} + 2F_{yr} \tag{23}$$

Now, lateral acceleration is given by:

$$a_y = \ddot{y} + v_x\dot{\psi} \tag{24}$$
$$\text{now,} \quad 2F_{yf} + 2F_{yr} = m(\ddot{y} + v_x\dot{\psi}) \tag{25}$$
$$\text{also,} \quad \beta = \frac{\dot{y}}{v_x} \tag{26}$$
$$\text{hence,} \quad 2F_{yf} + 2F_{yr} = mV_x(\dot{\beta} + \dot{\psi}) \tag{27}$$
$$\text{so,} \quad \dot{\beta} = \frac{2F_{yf} + 2F_{yr}}{mV_x} - \dot{\psi} \tag{28}$$

The lateral dynamics of the vehicle is heavily dependent on the tire properties. The driver's input is directly proportional to the primary forces acting on the vehicle due to braking, accelerating and lateral maneuvers. We are incorporating a widely used Pacejka tire model.

$$F_c = f_c(\alpha, s, \mu, F_z) \tag{29}$$
$$F_l = f_l(\alpha, s, \mu, F_z) \tag{30}$$
$$\tag{31}$$

Here, F_c is the lateral tire force or cornering force and F_l is the longitudinal tire force. These forces are complex functions (f_c, f_l) of tire slip angle α, longitudinal slip ratio s, friction μ and normal force F_z. On simplifying these equations, we get

$$F_{yf} = C_f(\delta_f - (\beta + \frac{l_f\dot{\psi}}{v_x})) \tag{32}$$
$$F_{yr} = C_r(-(\beta + \frac{l_f\dot{\psi}}{v_x})) \tag{33}$$
$$\tag{34}$$

The position and orientation of the vehicle can be determined in $X-Y$ co-ordinates using the following set of equations

$$\dot{x} = v\cos(\psi+\beta) \tag{35}$$
$$\dot{y} = v\sin(\psi+\beta) \tag{36}$$
$$\dot{x} = v\cos(\psi)\cos(\beta) - \sin(\beta)\sin(\psi) \tag{37}$$
$$\dot{y} = v\cos(\psi)\sin(\beta) + \cos(\beta)\sin(\psi) \tag{38}$$
$$v_x = v\cos(\beta) \tag{39}$$
$$\dot{x} = v_x\cos(\psi) - v_x\tan(\beta)\sin(\psi) \tag{40}$$
$$\dot{y} = v_x\sin(\psi) - v_x\tan(\beta)\cos(\psi) \tag{41}$$

1) Discretization of the System: On-line implementation of Model Predictive Control requires the equations to be in a discretized manner since the controller has to evaluate the model hundreds of times while tracking the trajectory. We are using Euler's forward method for discretization our model. Here, ζ represents the system in the continuous time.

$$\dot{\zeta}(t) = f_{cont}(\zeta(t), u(t)) \tag{42}$$

$$\Rightarrow \frac{\zeta(t_{k+1}) - \zeta(t_k)}{T_s} \approx f_{cont}(\zeta(t), u(t)) \tag{43}$$

$$\Rightarrow \zeta(t_{k+1}) \approx \zeta(t_k) + T_s f_{cont}(\zeta(t), u(t)) \tag{44}$$

Using the above method shown in Eqs. (42) to (44), we can write the discretization of each states as,

$$\beta(k+1) = \beta(k) + T_s\Big[\frac{2C_f}{mv_x}\big[\delta_f(k) - \beta(k) - \frac{l_f\dot{\psi}(k)}{v_x}\big] + \frac{2C_r}{mv_x}\big[-\beta(k) + \frac{l_r\dot{\psi}(k)}{v_x}\big] - \dot{\psi}(k)\Big] \tag{45}$$

$$\psi(k+1) = \psi(k) + T_s\psi(k) \tag{46}$$

$$\dot{\psi}(k+1) = \psi(k) + T_s\Big[\frac{2l_fC_f}{I_z}\big[\delta_f(k) - \beta(k) - \frac{l_f\dot{\psi}(k)}{v_x}\big] - \frac{2l_rC_r}{I_z}\big[-\beta(k) + \frac{l_r\dot{\psi}(k)}{v_x}\big]\Big] \tag{47}$$

$$X(k+1) = X(k) + T_s\big[v_x\cos(\psi(k)) - v_x\tan(\beta(k))\sin(\psi(k))\big] \tag{48}$$

$$Y(k+1) = Y(k) + T_s\big[v_x\sin(\psi(k)) + v_x\tan(\beta(k))\cos(\psi(k))\big] \tag{49}$$

Eqs. (45) to (49) represents the discretized model of the system.

2) MPC Formulation :

Objective Function: We define a generic cost function for generating trajectories which are inside the lane and also helps the vehicle to avoid obstacles as follows:

$$J = f(\gamma) \tag{50}$$
$$J = \frac{1}{2}\eta_N^T Q_0\eta_N + \sum_{K=0}^{N-1}\Big(\frac{1}{2}\bar{\eta}_N^T Q\bar{\eta}_N + \zeta_k^T S\zeta_k + u_k^T Ru_k\Big) \tag{51}$$

$\frac{1}{2}\eta_N^T Q_0\eta_N$ is the terminal cost function and penalizes the trajectory deviation at the last time step $(k=N)$ of look-ahead horizon. The running cost penalizes the trajectory deviation at each time step $(k=(1,2,....N-1))$ during the entire horizon.The matrices P, Q, R and S are the penalty weighing matrices. The matrices $Q_0(2x2)$ and $Q(2x2)$ penalize the terminal trajectory deviation and the running trajectory deviation respectively[10]. The matrices $S(5x5)$ and $R(1x1)$ penalize the large state values and the large input values. All the penalty weighing matrices are diagonal matrices[10].

Constraints: A simple method of implementing constraints is by projecting the candidate input u_k^* on the constraint set as per Eq. (52).

$$u_{k,min} \le u_k^* \le u_{k,max} \quad u = \delta_f(\text{steering angle}) \tag{52}$$

Also in order to ensure that the car deviates from the obstacle but not too far away is by imposing the constraint on the lateral position error as per Eq. (53).

$$\text{lateral displacement error} \le \frac{1}{2}\text{lane width} \tag{53}$$

MPC Optimization: Minimization of the cost function as defined by Eq. (51) is done by one of the traditional method called Gradient descent algorithm.

- Start with the initial guess of u for the initial state ζ_0.
- Find slope $\frac{\mathrm{d}J}{\mathrm{d}\gamma}$.
- Take step Δ along negative slope.
- Repeat until $\frac{\mathrm{d}J}{\mathrm{d}\gamma} = 0$.

By the end of the Gradient Descent algorithm, we are expecting to obtain the optimal control input which drives the vehicle along the lane avoiding the obstacle.

C. Obstacle Avoidance Problem

1) Obstacle Modeling: The obstacles are considered as static point obstacles or geometric obstacles. Different methodologies were tried out to incorporate the obstacle in the environment, into the cost function using YALMIP library and fmincon function.

- Brute Force
- Potential Field Function
- Gaussian Function

In paper [10], the obstacle avoidance strategy is handled as per Fig. 5. At each time step d_k is calculated as a point, which lies on straight lines from vehicle CG to the reference line and selecting the one which is closer to the goal point. When the vehicle is at point $\eta_{k,1}$, two points are found on the reference

line corresponding to this point $\eta_{k,1,X}$ and $\eta_{d,k,1,Y}$, which are calculated by a horizontal and a vertical line from the vehicle CG to the reference line. But only point $\eta_{k,1,X}$ is considered to be the desired point $\eta_{d,k,1}$ because it is closer to the goal point. Thus the reference trajectory criterion has considered the approach to the goal point rather than just following the trajectory in any direction. Similarly when the vehicle is at point $\eta_{k,2}$, only $\eta_{k,2,Y}$ is considered the desired point $\eta_{k,2}$ due to its proximity to the goal point [10].

In this project, we have utilized a similar approach but instead moving about the obstacle in a triangular path, we have implemented the same using trapezoidal path. This reduces the sharp changes in the steering angle when compared to the triangular path approach.

2) Cost Functions for Obstacle Representation:

Brute Force: In this naive method, the states are constrained (bounded) within the obstacle-free space. Whenever the vehicle moves closer to obstacle in the x-axis, the constraint in the y-axis is given less weightage enabling the vehicle to deviate from the obstacle as well as from the reference trajectory. The constraints are defined as per the Eq. (54).

$$x \leq (x_{obs} - \epsilon) \quad and \quad x \geq (x_{obs} + \epsilon) \tag{54}$$

where (x, y) is the current location of the vehicle and (x_{obs}, y_{obs}) is the location of the obstacle to be avoided. ϵ is a small positive number for defining boundary around the obstacle. The limitation involved in this method is the drastic change in the steering angle when the system approaches the obstacle.This limitation can be observed in Fig. 21.

Potential Field Function: In this method, a repulsive potential field around the obstacle is created which is computed based on the current location of the vehicle. The function P_k in Eq. (55) is used to compute the potential field for the vehicle's current location.

$$P_k = \frac{1}{(x - x_{obs})^2 + (y - y_{obs})^2 + \epsilon} \tag{55}$$

where (x, y) is the current location of the vehicle and (x_{obs}, y_{obs}) is the location of the obstacle to be avoided. ϵ is a small positive number for non singularity. The advantage with such simple function is that it is easily differentiable and does not lead to complex differential terms in the optimization part [10].

Gaussian Function: This method is similar to potential field function approach. Here, a Gaussian distribution N_0 is created with the peak value at the obstacle's location.

$$N_0 = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(Z - Z_{obs})^2}{2\sigma^2}\right\} \tag{56}$$

where, $Z = [x, y]$, $Z_{obs} = [x_{obs}, y_{obs}]$, $\sigma = 4X4$ matrix.
The advantage of using this method is that we can manipulate how close the car can move towards the obstacle by changing the covariance matrix. Fig. 14 in appendix A shows the cost map generated by Eq. (56).

The above two methodologies couldn't be implemented effectively because of the limitations in the YALMIP library and optimization solver used.

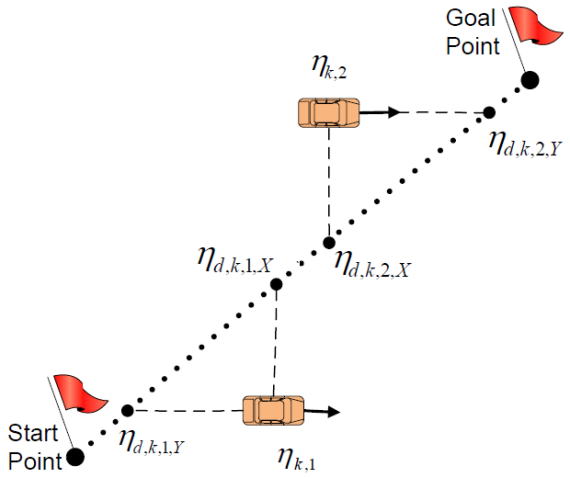

Fig. 5: Obstacle Modeling

VI. RESULTS AND DISCUSSIONS

A. Kinematic Bicycle Model

In this section, the trajectory tracking behavior of the MPC for straight line and circular trajectory is explained. The control algorithms were initially implemented using fmincon function of Matlab.The results were not satisfactory and were varying with different initial guesses for state. Hence we have chosen YALMIP optimization library implemented in Matlab for our experiments [11].Appendix A contains the figures and plots relevant to this section.

Constant Velocity:

Following sections discusses the results obtained for the constant velocity and the steering angle as the only the control input.

1) Straight Line Trajectory Tracking: The initial conditions for the experiments were [x,y,ψ]=[2 m, -2 m, pi/2 rad]. The constant velocity is fixed to be 1 m/s. The parameters in Tables I and II were fixed for the respective experiments.

Two experiments were conducted for tracking the straight line with two different horizons first with 20 time steps and second with 30 time steps. The red line in the trajectory represents the path of the vehicle and the blue line represents the desired trajectory. On observing the Figs. 6 and 7 we can say that the trajectory tracking performance is improved for horizon with 30 time steps when compared to horizon with 20

TABLE I: Parameter Table for Experiment 1

Parameters	Values
T_s (sampling time)	0.01
Q	[1 0 0; 0 1 0; 0 0 1]
R	[1]
N	20
Number of states	3
Number of inputs	1
Constraint on steering angle	$\delta \in [-60, 60]$

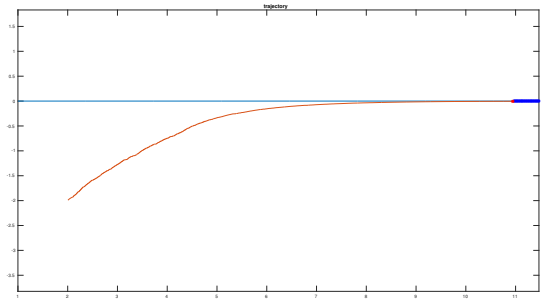

Fig. 6: Experiment 1-Tracking Straight Line for N = 20

TABLE II: Parameter Table for Experiment 2

Parameters	Values
T_s (sampling time)	0.01
Q	[1 0 0; 0 1 0; 0 0 1]
R	[1]
N	30
Number of states	3
Number of inputs	1
Constraint on steering angle	$\delta \in [-60, 60]$

time steps. On observing the deviations in the Figs. 15 and 17 it is evident that designed controller is giving the optimal control action to track the trajectory. The Figs. 16 and 18 represents the optimal steering angle given to propagate the system. Since the initial orientation of the vehicle is set as $pi/2$, we observe a steady variation in optimal steering angle as there is continuous optimization for the x , y and ψ of the vehicle at each time step.

2) Circular Trajectory Tracking: The initial conditions for the experiments were [x,y,ψ]=[1.5 m,0 m,pi/2 rad]. The constant velocity is fixed to $1m/s$. The parameters in Table III were fixed for this experiment.

In order to track the circular trajectory, the point on the desired trajectory nearest to the start point is chosen as the reference point for the first horizon. Thereafter, the state updation and control input optimization for each prediction horizon is updated at each time step based on the previous control input.The circular trajectory tracking is performed by

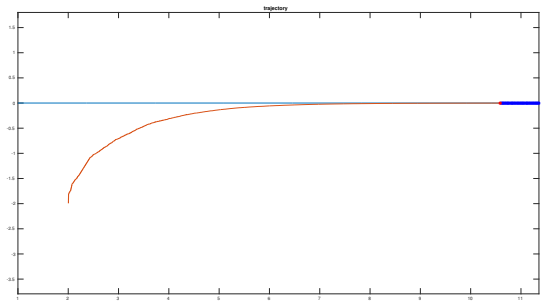

Fig. 7: Experiment 2-Tracking Straight Line for N = 30

TABLE III: Parameter Table for Experiment 3

Parameters	Values
Radius of circle	1.5
T_s (sampling time)	0.01
Q	[1 0 0; 0 1 0; 0 0 1]
R	[1]
N	10
Number of states	3
Number of inputs	1
Constraint on steering angle	$\delta \in [-60, 60]$

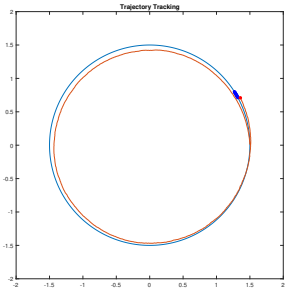

Fig. 8: Experiment 3-Tracking circular trajectory for N = 10

constraining the control input within the limits as shown in Table III. On observing Fig. 19 it can be seen that there is a slight deviation from the desired trajectory along the curvature of the circle. This is due to the fact that the angular velocity of the vehicle is limited to ensure the stability of the vehicle and prevent toppling. This prohibits vehicle trajectory to take steep turns along the curvature while tracking the desired trajectory.

Variable Velocity:

Following sections discusses the results obtained when both velocity and steering angle are the control inputs.

3) Circular Trajectory Tracking: The initial conditions of the experiments were [x,y,ψ,v]=[0 m,1.6 m,pi rad, 1 m/s].The parameters in Table IV were fixed for this experiment.

Disturbance Rejection Nature of MPC:

A random noise is introduced as a disturbance to the vehicle model. The MPC is modeled to reject this disturbance introduced in the vehicle states. The MPC controller rejects these disturbance rather than propagating them at each horizon.

TABLE IV: Parameter Table for Experiment 4

Parameters	Values
T_s (sampling time)	0.01
Q	[20 0 0 0; 0 20 0 0; 0 0 20 0; 0 0 0 30]
R	[0.01 0;0 0.01]
N	30
Number of states	4
Number of inputs	2
Constraint on steering angle	$\delta \in [-60, 60]$
Constraint on velocity	$v \in [0, 2\ reference\ velocity]$

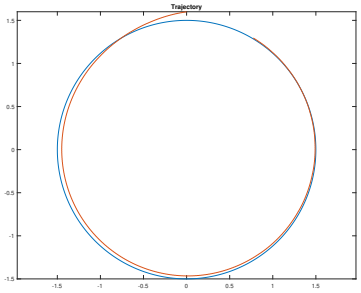

Fig. 9: Experiment 4-Tracking Circular trajectory with velocity as an control input

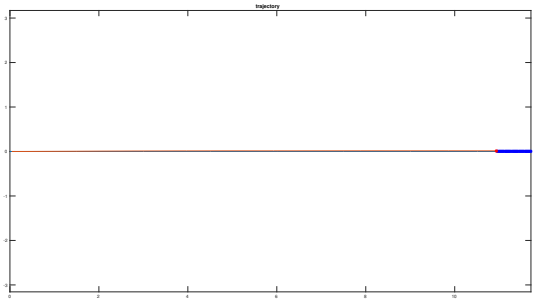

Fig. 10: Overall disturbance rejection performance V=4m/s

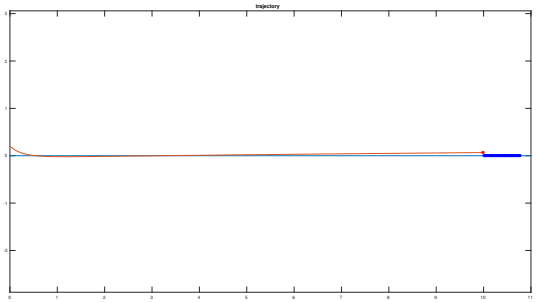

Fig. 11: Overall disturbance rejection performance V=1m/s

Comparing Fig. 10 and Fig. 11 we can see that there is a deviation from the path when the velocity is 1 m/s whereas, the trajectory is followed accurately for velocity = 4 m/s. This is because the velocity of 1 m/s is not enough to converge in one time-step and the noise makes a difference in the amount of distance travelled in a time step. For velocity=4 m/s the distance to converge to trajectory is easily covered in one time-step and hence there is no deviation from the desired trajectory.

Smoothing of control input:

Until now we have seen much deviation in steering angle inputs as in Figs. 16, 18 and 20. This is undesirable. We have introduced an extra term in the cost function to smooth this input. The term added is $(u_k - u_{k-1})R_c(u_k - u_{k-1})^T$.

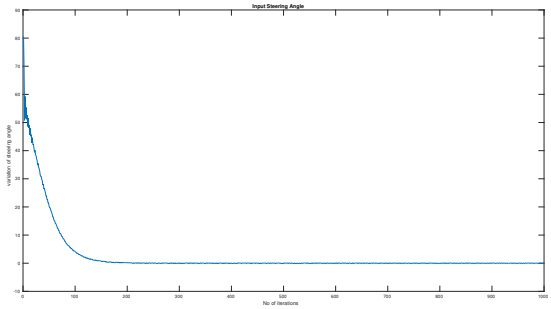

Fig. 12: Smoothing of input

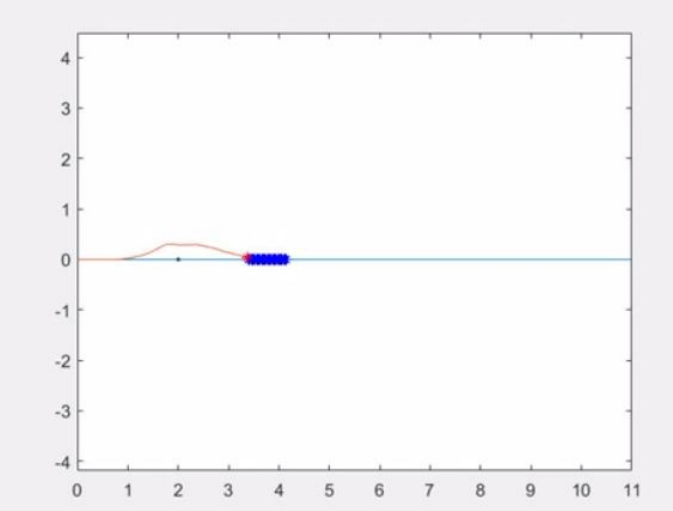

Fig. 13: Experiment 5-Avoiding Obstacle using MPC

Fig. 12 shows the effects of smoothing.This makes the MPC implementation possible in real-time vehicle.

Obstacle Avoidance:

The initial conditions of the experiments were [x,y,ψ]=[0 m,0 m,0 rad].The parameters in Table V were fixed for this experiment. Figs. 13 and 21 shows the obstacle avoided by the controller and the control input (steering angle) generated by the controller.

TABLE V: Parameter Table for Experiment 5

Parameters	Values
T_s (sampling time)	0.01
Q	[1 0 0 ; 0 1 0 ; 0 0 1]
R	[1]
N	30
Number of states	3
Number of inputs	1
Constraint on steering angle	$\delta \in [-60, 60]$

VII. CONCLUSION AND FUTURE WORK

In this project,we have presented a kinematic bicycle model of the vehicle dynamics,and used it to formulate an MPC problem for obstacle avoidance and trajectory tracking. The non-linear model is linearized using Jacobian linearization and a linearized system model is used as the plant. The linearity of the model and convexity of the constraints is used to cast the MPC problem as Quadratic Programming problem.The YALMIP library integrated with MATLAB is used as the tool to implement this project. The optimization solver used in Quadratic Programming solver which is in-built in YALMIP. Preliminary results demonstrate smooth integration of the controller showing promise of this control scheme on real-time implementations. The main advantage of the Model Predictive Control(MPC) being able to incorporate safety constraints during real time and its adaptive disturbance rejection capability is verified through this project results. The additional constraint on the steering rate smoothen the variation in steering angle which qualifies the controller to be tested real-time.The modeling of the obstacle as Gaussian functions and defining a safety envelope around the obstacle works well in simulation. The real time implementation of this obstacle avoidance algorithm is still a question as the obstacles considered in this project as static and the location of the obstacles and known in prior.In this project, we also made an attempt to implement the dynamic bicycle model. The main issue with the dynamic bicycle model is the unknown inertia parameter which has to be estimated in real time. The parameter estimation can be performed by using a more robust control approach, like Robust MPC or Adaptive MPC. The dynamic bicycle model is more alike the real-time car model hence MPC implemented in dynamic bicycle model would definitely yield a better performance close to real world.
Future work focuses on incorporation of moving obstacles and methodologies to leverage brake actuation and variable speed. A more practical approach to on-line obstacle avoidance need to be developed in the future.

VIII. GOALS

By the end of this project,we were able to accomplish the Reachable Goals listed below. The desired goals can be carried out as a future work.

A. Reachable Goals

- To implement a trajectory tracking controller that can follow any given set of trajectory points.
- Use the environmental information and vehicle model to implement a Model Predictive Control that can deviate from the desired path along with maintaining the stability of the vehicle and also avoid obstacles such that vehicle is within the lane.

B. Desired Goals

- Implementing a Model Predictive Control for path tracking in roads having high curvatures.
- Design a safe environment envelopes which are a convex set of points namely tubes, that are free of obstacles and are within road boundaries.
- Implementation of developed MPC on the F1 tenth race car.

REFERENCES

[1] Garcia, Carlos E., David M. Prett, and Manfred Morari. "Model predictive control: theory and practicea survey." Automatica 25.3 (1989): 335-348.

[2] Mayne, David Q., et al. "Constrained model predictive control: Stability and optimality." Automatica 36.6 (2000): 789-814.

[3] Kong, Jason, et al. "Kinematic and dynamic vehicle models for autonomous driving control design." Intelligent Vehicles Symposium (IV), 2015 IEEE. IEEE,2015

[4] Morari, Manfred, and Jay H. Lee. "Model predictive control: past, present and future." Computers and Chemical Engineering 23.4 (1999): 667-682.

[5] Cychowski, Marcin, Krzysztof Szabat, and Teresa Orlowski-Kowalska. "Constrained model predictive control of the drive system with mechanical elasticity." IEEE Transactions on Industrial Electronics 56.6 (2009):1963-1973.

[6] Kamal, Md Abdus Samad, et al. "On board ecodriving system for varying road-traffic environments using model predictive control." Control Applications (CCA), 2010 IEEE International Conference on. IEEE, 2010.

[7] Li, Shengbo, et al. "Model predictive multi-objective vehicular adaptive cruise control." IEEE Transactions on Control Systems Technology 19.3 (2011): 556-566.

[8] Beal, Craig Earl, and J. Christian Gerdes. "Model predictive control for vehicle stabilization at the limits of handling." IEEE Transactions on Control Systems Technology 21.4 (2013): 1258-1269.

[9] Brown, Matthew, et al. "Safe driving envelopes for path tracking in autonomous vehicles." Control Engineering Practice (2016).

[10] NON-LINEAR MODEL PREDICTIVE CONTROL FOR AUTONOMOUS VEHICLES by MUHAMMAD AWAIS ABBAS.

[11] https://yalmip.github.io/

[12] Y. Gao, A. Gray, J. Frasch, T. Lin, E. Tseng, J. Hedrick, and F. Borrelli, Spatial predictive control for agile semi-autonomous ground vehicles, 13th International Symposium on Advanced Vehicle Control, 2012.

[13] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, An optimal-control-based framework for trajectory planning, threat assessment,and semi-autonomous control of passenger vehicles in hazard avoidance scenarios, International Journal of Vehicle Autonomous Systems, vol. 8, no. 2, pp. 190216, 2010.

[14] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat,Predictive active steering control for autonomous vehicle systems, IEEE Trans. on Control System Technology, vol. 15, no. 3, 2007.

[15] P. Falcone, B. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, Low complexity MPC schemes for integrated vehicle dynamics control problems, 9th International Symposium on Advanced Vehicle Control, 2008.

[16] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads, Dynamic Systems and Control Conference, 2010, 2010.

[17] A. Gray, M. Ali, Y. Gao, J. Hedrick, and F. Borrelli, Integrated threat assessment and control design for roadway departure avoidance, in Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, Sept., pp. 17141719.

IX. Appendix A

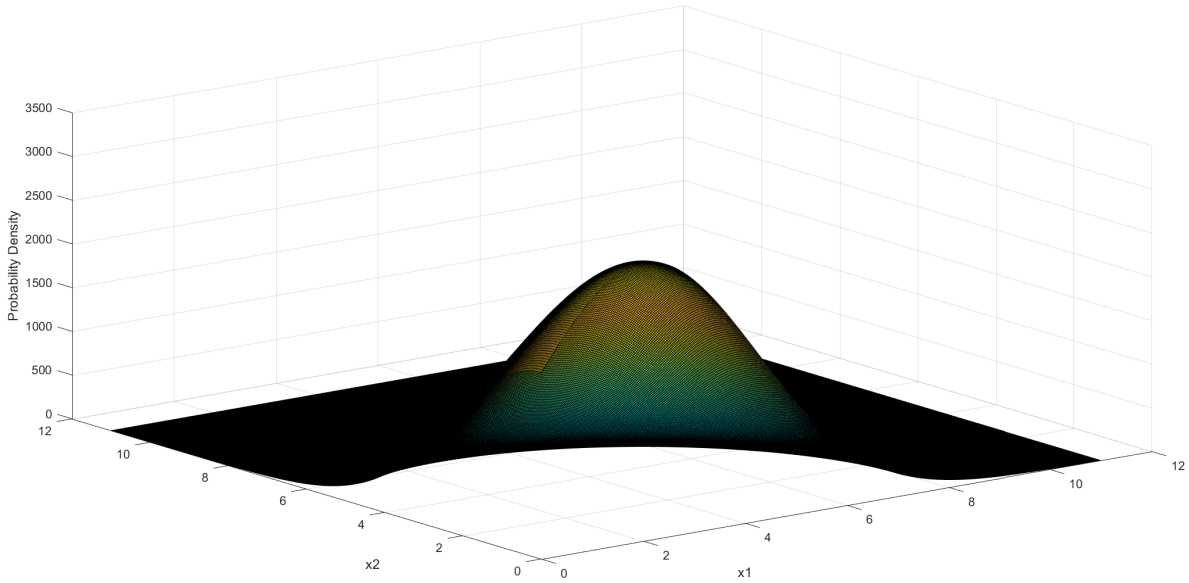

Fig. 14: Cost map generated using Gaussian Function

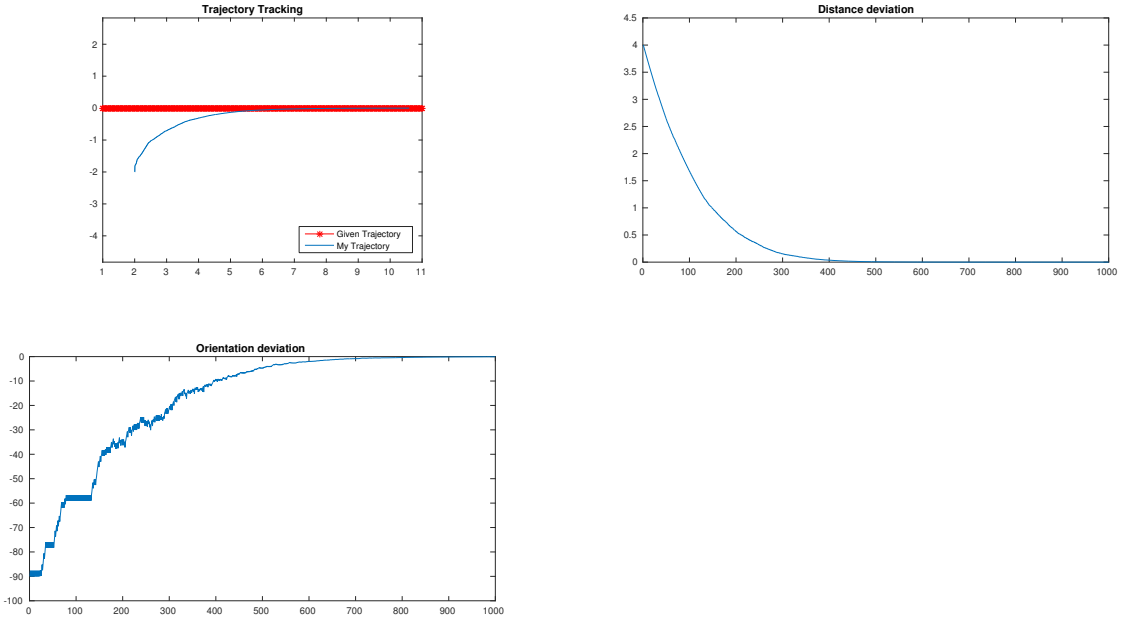

Fig. 15: Tracking Straight Line for N = 30 along with orientation and distance deviation

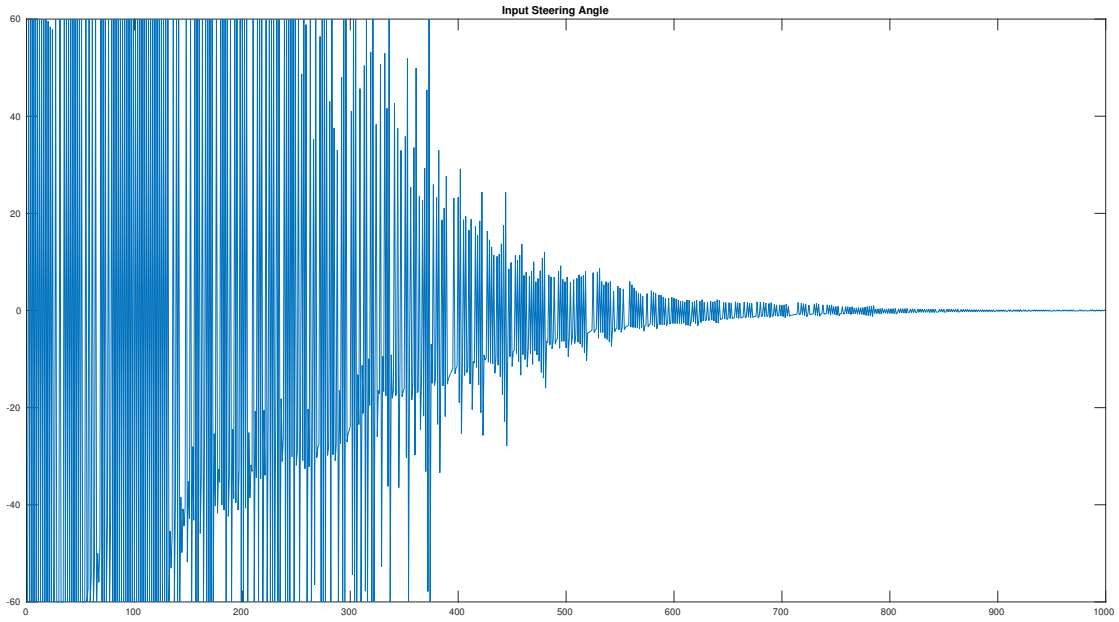

Fig. 16: Steering Angle (optimized control action) for N=30

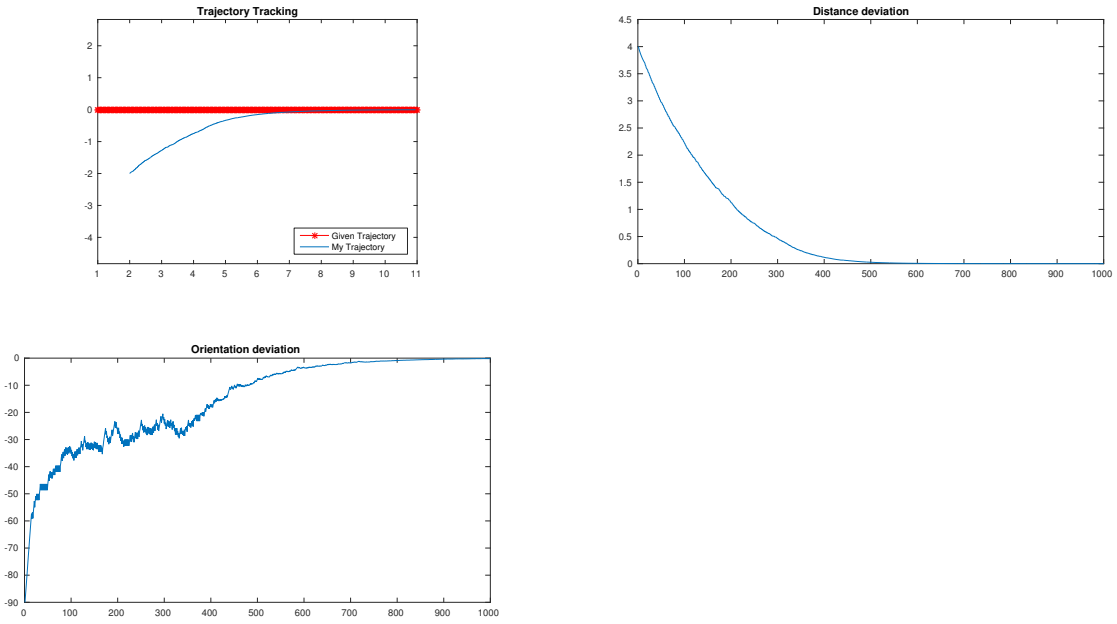

Fig. 17: Tracking Straight Line for N = 20 along with orientation and distance deviation

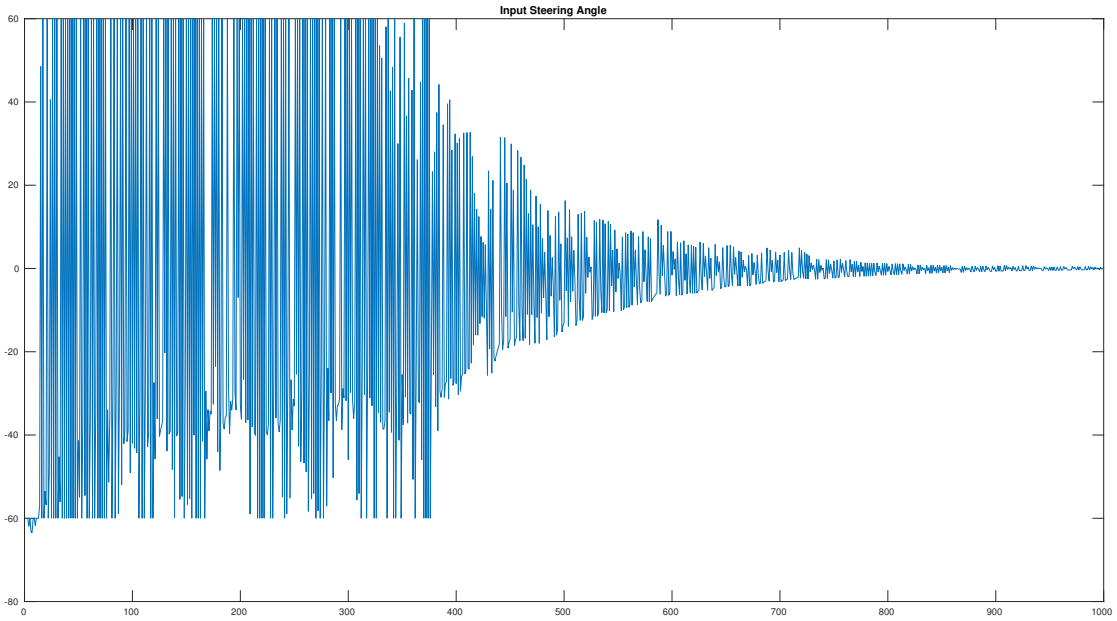

Fig. 18: Steering Angle (optimized control action) for N=20

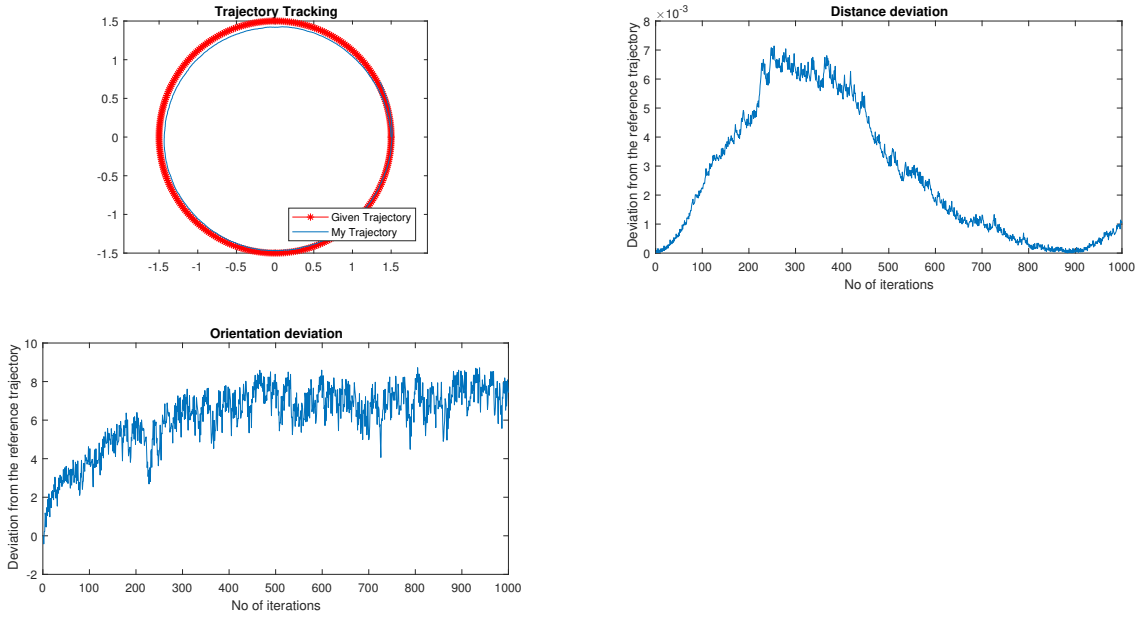

Fig. 19: Tracking circular trajectory for N = 10 along with orientation and distance deviation

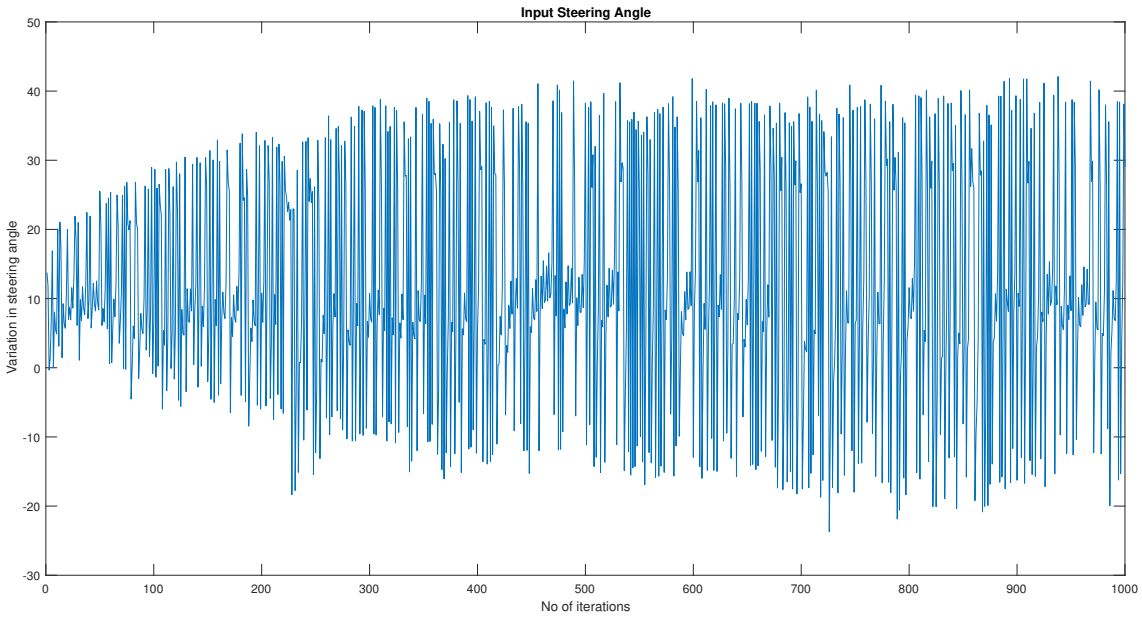

Fig. 20: Steering Angle (optimized control action) for N=10

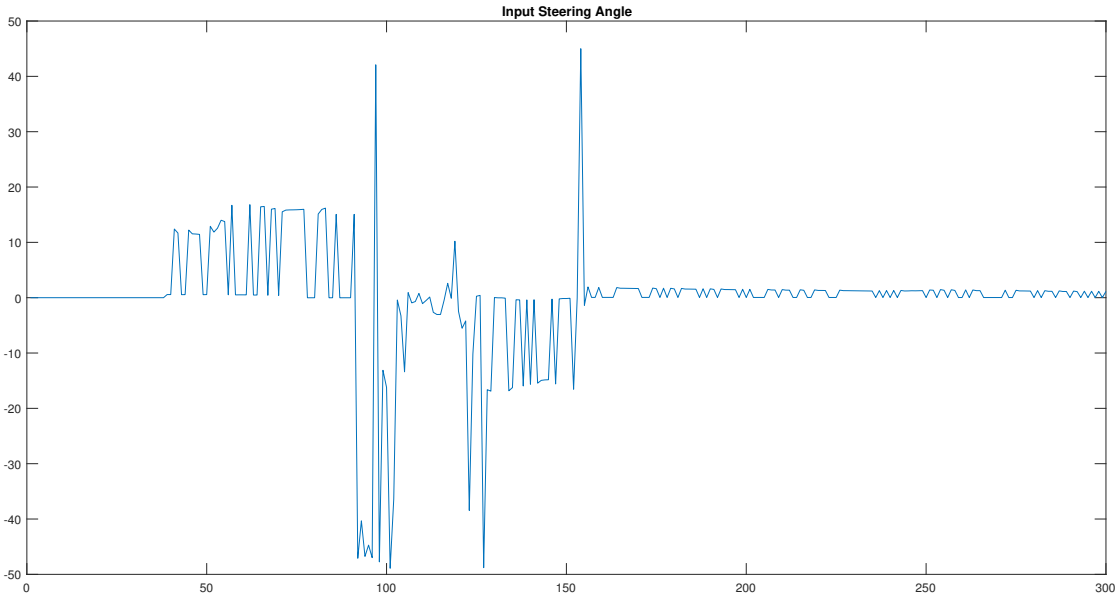

Fig. 21: Steering angle: Avoiding obstacle using Brute Force method

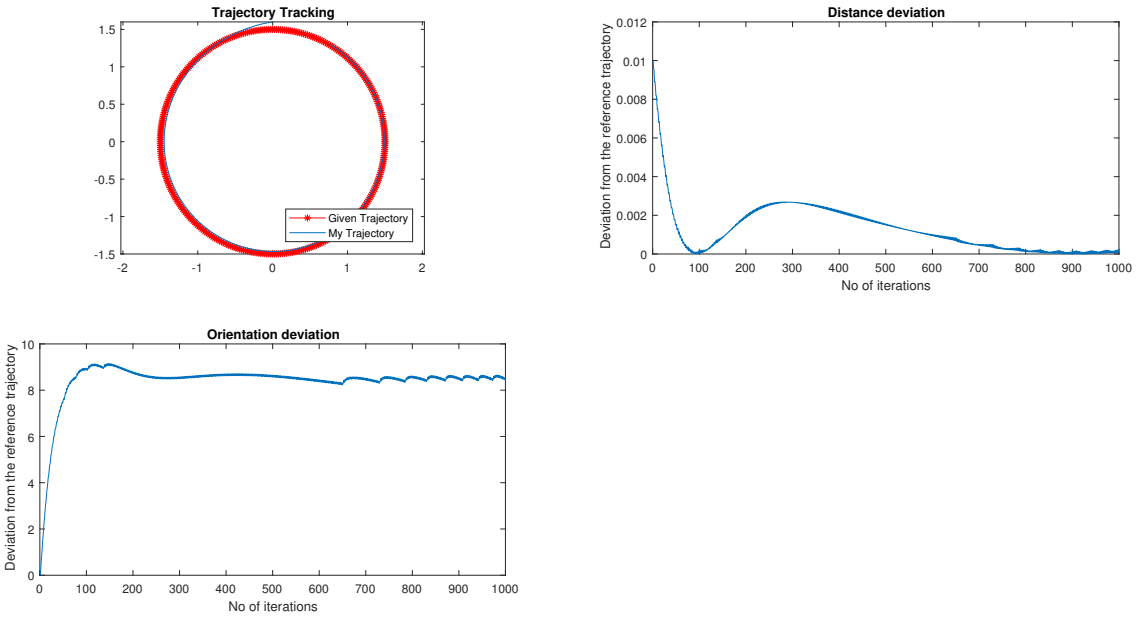

Fig. 22: Tracking circular trajectory with velocity as control input

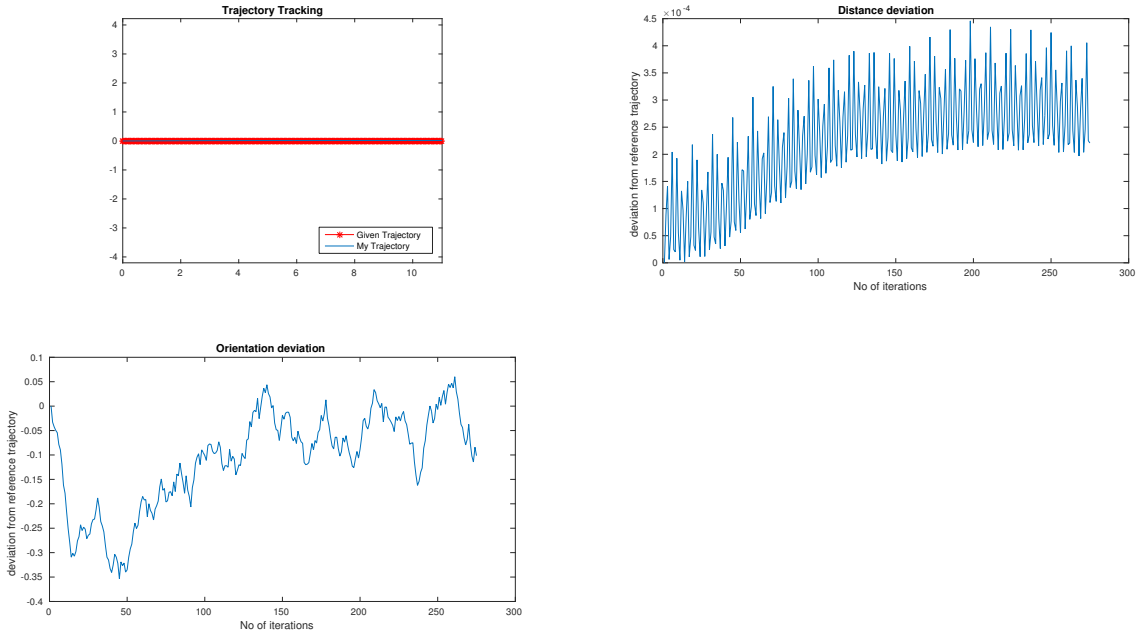

Fig. 23: Details for Disturbance Rejection V=4m/s

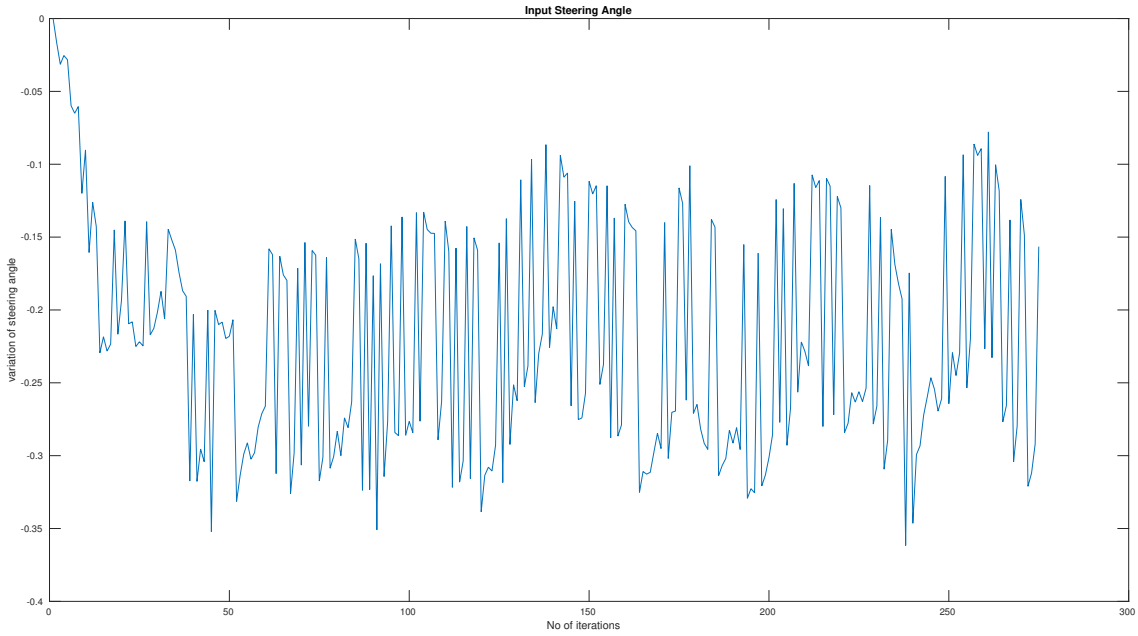

Fig. 24: steering parameters and disturbance rejection V=4m/s

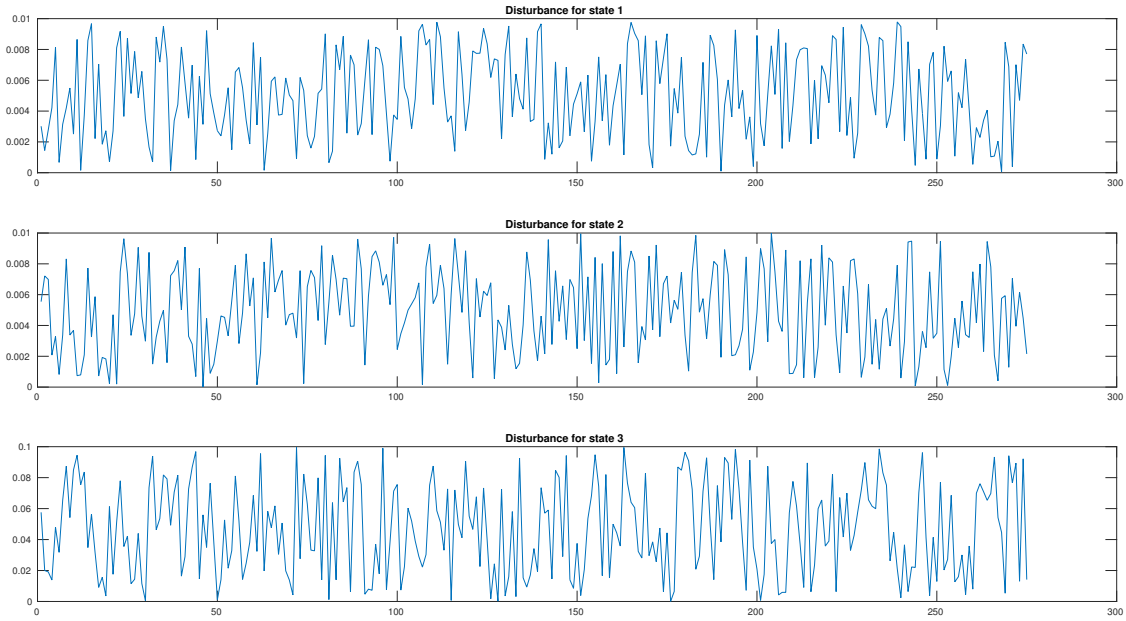

Fig. 25: Disturbance for Disturbance Rejection V=4m/s

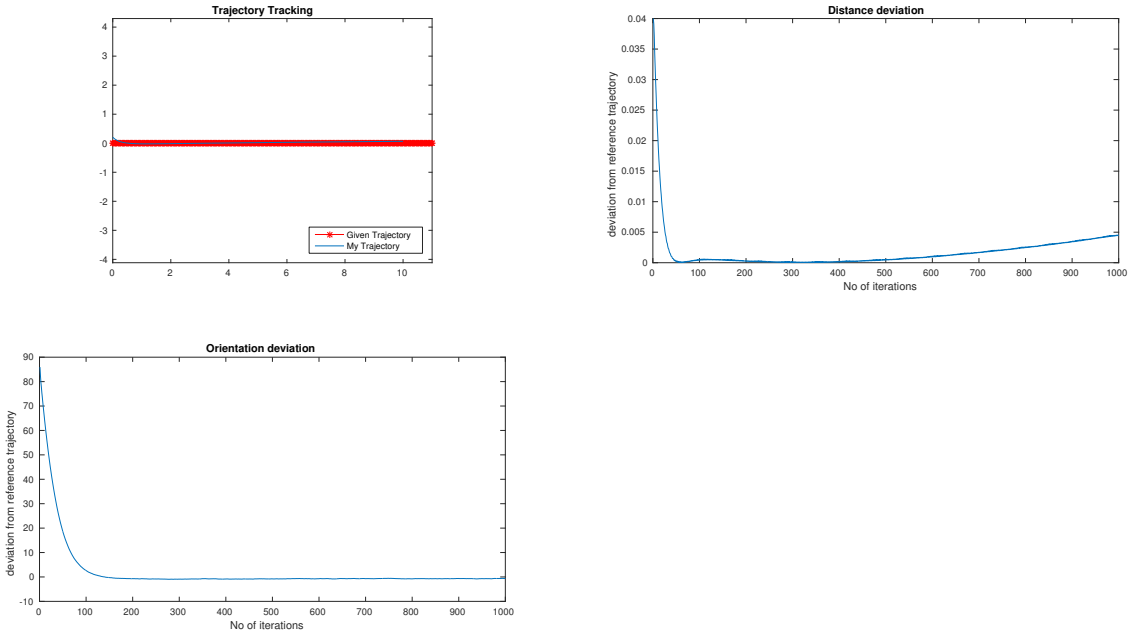

Fig. 26: Details for Disturbance Rejection V=1m/s

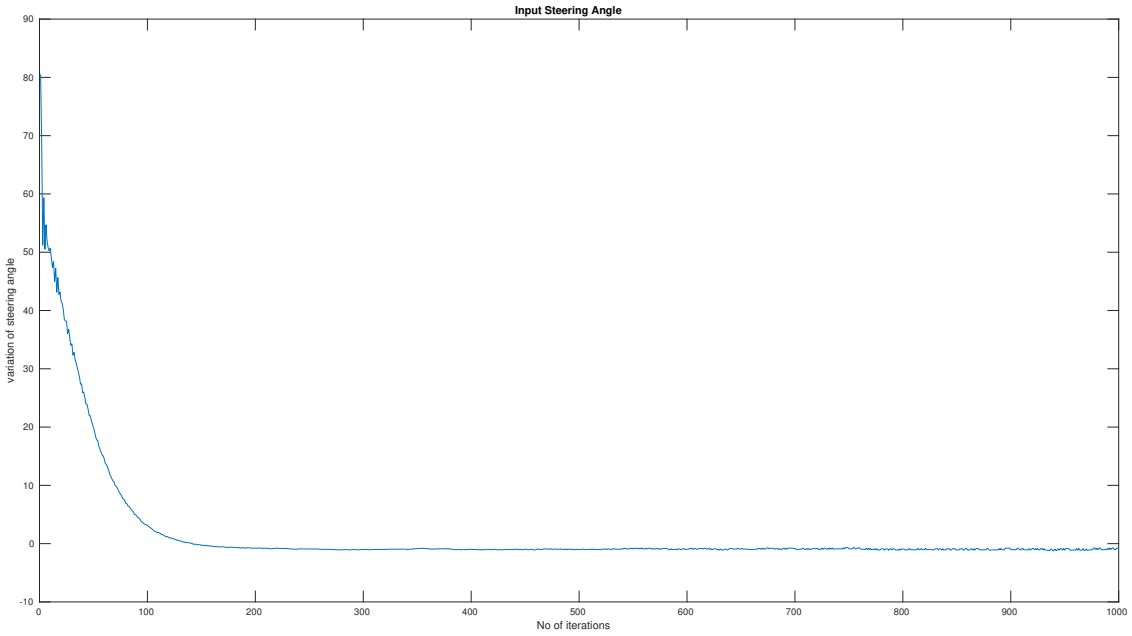

Fig. 27: steering parameters and disturbance rejection V=1m/s