Vehicle Dynamics and Control

RO47017 Homework 3

MSc Systems & Control
Delft University of Technology
Qingyi Ren 5684803
June 2, 2023

I. Introduction

In this homework, the Path-following control is taken into analysis and the default MPC Path-Follower based on kinematic model and designed MPC Path-Follower are built and the Root Mean Square Error(RMSE) is calculated based on different weighting matrices.

II. Default MPC path-follower

A. Kinematic Model

The vehicle dynamics are represented as kinematic model as:

$$\begin{aligned}
&\beta = tan^{-1}(\frac{l_r}{L}tan\delta) \\
&\dot{u} = 0 \\
&\dot{\psi} = \frac{u}{l_r}\sin\beta \\
&\dot{x} = u\cos(\psi+\beta) \\
&\dot{y} = u\sin(\psi+\beta)
\end{aligned} \tag{1}$$

where β is sideslip angle in [rad] which is defined by steering wheel angle δ, u is longitudinal velocity in [m/s], ψ is yaw angle in [rad] and x and y are x, y are the horizontal and vertical distances in [m] respectively. L represents wheelbase in [m] and l_r represents distance from rear axle to CoG in [m]. The vehicle parameters of vehicle dynamics are shown in Table IV.

The nonlinear state space is taking $[u, \psi, x, y]$ as state vector and steering wheel angle δ as input. The nonlinear state space model is written as:

$$\begin{bmatrix}\dot{u}\\ \dot{\psi}\\ \dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}0\\ \frac{u}{l_r}\sin\beta\\ u\cos(\psi+\beta)\\ u\sin(\psi+\beta)\end{bmatrix}, \quad \beta = tan^{-1}(\frac{l_r}{L}tan\delta) \tag{2}$$

In MATLAB, the ACADO setup is used to formulate kinematic model:

```
%% ACADO set up
DifferentialState vx yaw Xp Yp; % definition of controller states
Control delta; % definition of controller input
% controller model of the plant
beta = atan(par.l_r * tan (delta) / par.L);

f_ctrl = [
    dot(vx)  == 0;...
    dot(yaw) == vx / par.l_r * sin(beta);...
    dot(Xp)  == vx * cos(yaw + beta);...
    dot(Yp)  == vx * sin(yaw + beta)];
```

B. MPC controller

To achieve the goal of tracking trajectory, the linear MPC is implemented and the model of MPC is written as:

$$\begin{aligned}
&\min J = \min \int x^TQx + u^TRudt = \min_{\delta}\sum_{i=1}^{N_p}(\left\|x_{k+i}\right\|^2 Q + \left\|\delta_k\right\|^2 R) \\
&s.t. \quad x_{k+1} = f(x_k, u_k), \quad -\delta_{max} \leq \delta_k \leq \delta_{max} \quad -\beta_{max} \leq \beta \leq \beta_{max}
\end{aligned} \tag{3}$$

where $x_{k+1} = f(x_k, u_k)$ represented the vehicle kinematic model which is nonlinear state space model. In Simulink, this model is implemented in the block '**Nonlinear Vehicle Model**' which takes δ from MPC controller as input and states $[u, \psi, x, y]^T$ as output. $N_p = 40$ represents the prediction horizon. Q is a positive semi-definite matrix that defines the weight given to the different states in the control objective. R is a positive definite matrix that represents the importance given to the control effort. Also there are two constraints of δ_k and β respectively. $\delta_{max} = 23.3766$ deg and $\beta_{max} = 10$ deg.

C. *Simulation*

In this part, different weighting parameters for lateral position and control input (default $5e-3$ and $1e-1$ correspondingly) the performance of MPC controller is analysed.

By the former analysis, the weighting matrix is Q associated with the state x thus having the dimension $\Re^{4\times 4}$. R represents the importance of control input u to cost function and is a scalar. More specifically, the matrices Q and R are shown as:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{tune} \end{bmatrix} \qquad R = r_{tune}$$

where q_{tune} and r_{tune} are parameters which could be tuned to analyse the impact on MPC performance. The initial values for q_{tune} and $R = r_{tune}$ are 0.005 and 0.1 respectively. These two parameters could be tuned in stage cost matrix of '**MPC controller**' block. The initial state of x is set to be $[v_{ref}, 0, 0, 0]^T$ with the initial forward speed of 100 km/h.

1) The analysis of weighting tuning for lateral position: In this part, the value of r_{tune} is set to be fixed at 0.1, and the $Q(4,4) = q_{tune}$ is increased and decreased.

First considering gradually increasing the value of $Q(4,4)$ from 0.005 to 0.01, the Root Mean Square Error(RMSE) is decreasing from 0.1274 to 0.1023 which indicates the enhancement of tracking performance of lateral position shown in Fig (1). However when $Q(4,4)$ reached 0.02, the weighting value severely deteriorates the tracking performance leading to a significantly increased RMSE. It is obvious that the system becomes unstable with $Q(4,4) \geq 0.02$. In Fig (2), the steer wheel angle shows the same character of tracking performance, when $Q(4,4) \geq 0.02$, the steer wheel angle does not have the trend of converging to zero while when $0.005 \leq Q(4,4) < 0.02$, the steer wheel angle converges to zero which could verify the good tracking performance of lateral position.

When decreasing the value of $Q(4,4)$ from 0.005 to 0.0002, the Root Mean Square Error(RMSE) is increasing from 0.1274 to 1.1393 of lateral position to y reference. In this scenario decreasing $Q(4,4)$ leads to higher value of RMSE and worse tracking performance shown in Fig (3). Furthermore when $Q(4,4)$ reached 0.0002, the weighting value severely deteriorates the tracking performance and leads to a significantly increased RMSE. It is obvious that the system becomes unstable with $Q(4,4) = 0.0002$. In Fig (4), the steer wheel angle shows the same character of tracking performance, when $Q(4,4) = 0.0002$, the steer wheel angle does not have the decreasing amplitude of oscillation around zero point while when decreasing $Q(4,4)$ but not making it reach 0.002, the amplitude of steer wheel angle stays small which could verify performance of lateral position is weakened but not deteriorated to a unstable system (MPC fails).

In Table I, different values of weighting matrices Q with fixed R of MPC controller and corresponding Root Mean Squared Error(RMSE) are shown. It could be concluded that when increasing $Q(4,4)$ the tracking performance of lateral position is enhanced but when $Q(4,4)$ is too large or too small, the tracking performance is deteriorated which cause the failure of MPC controller.

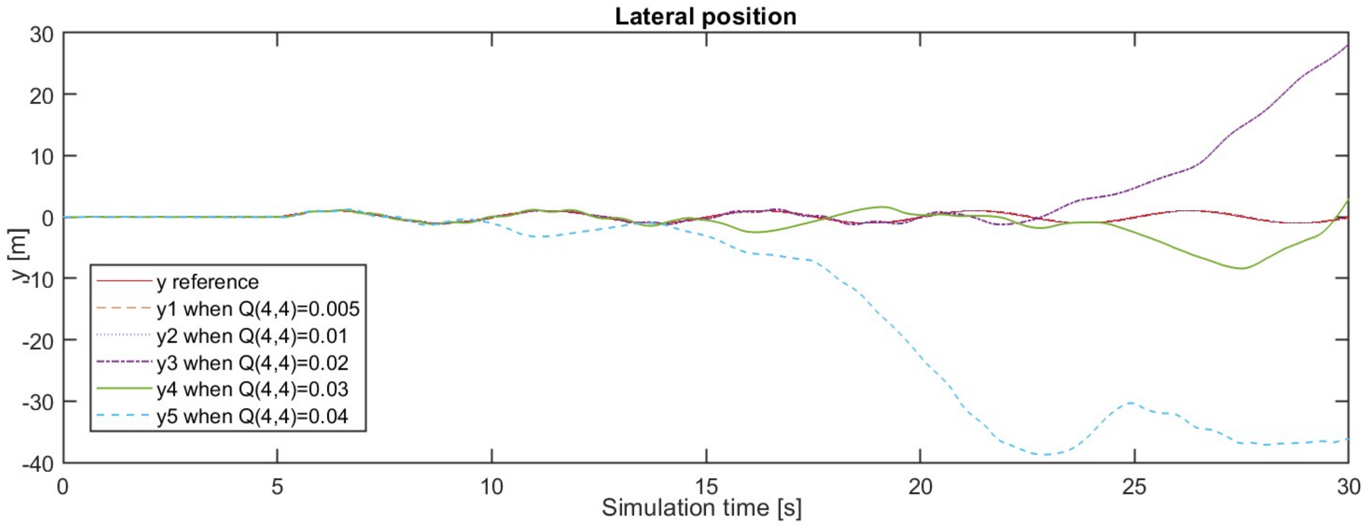

Fig. 1. The lateral position when increasing $Q(4,4)$.

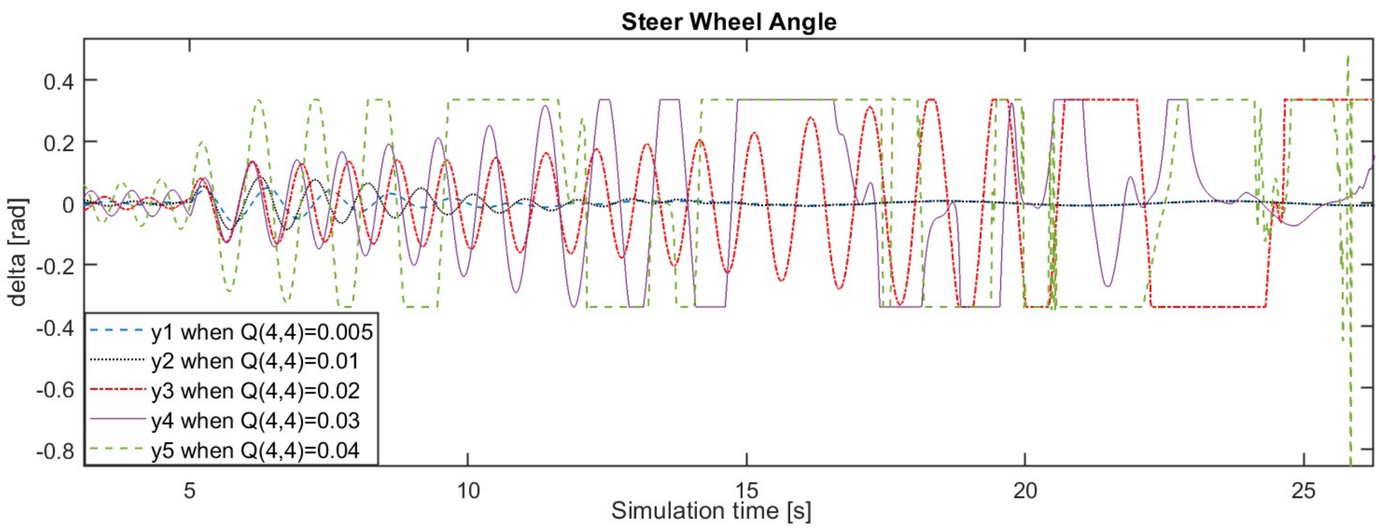

Fig. 2. The steer wheel angle when increasing $Q(4,4)$.

$Steps$	$Q(4,4)$	R	Root Mean Squared Error (RMSE)
1	0.0002	0.1	1.1393
2	0.0005	0.1	0.2325
3	0.001	0.1	0.1935
4	0.002	0.1	0.1655
5	0.005	0.1	0.1274
6	0.01	0.1	0.1023
7	0.02	0.1	7.048
8	0.03	0.1	2.5235
9	0.04	0.1	20.4849

TABLE I
DIFFERENT VALUES OF WEIGHTING MATRICES Q WITH FIXED R OF MPC CONTROLLER AND CORRESPONDING ROOT MEAN SQUARED ERROR(RMSE).

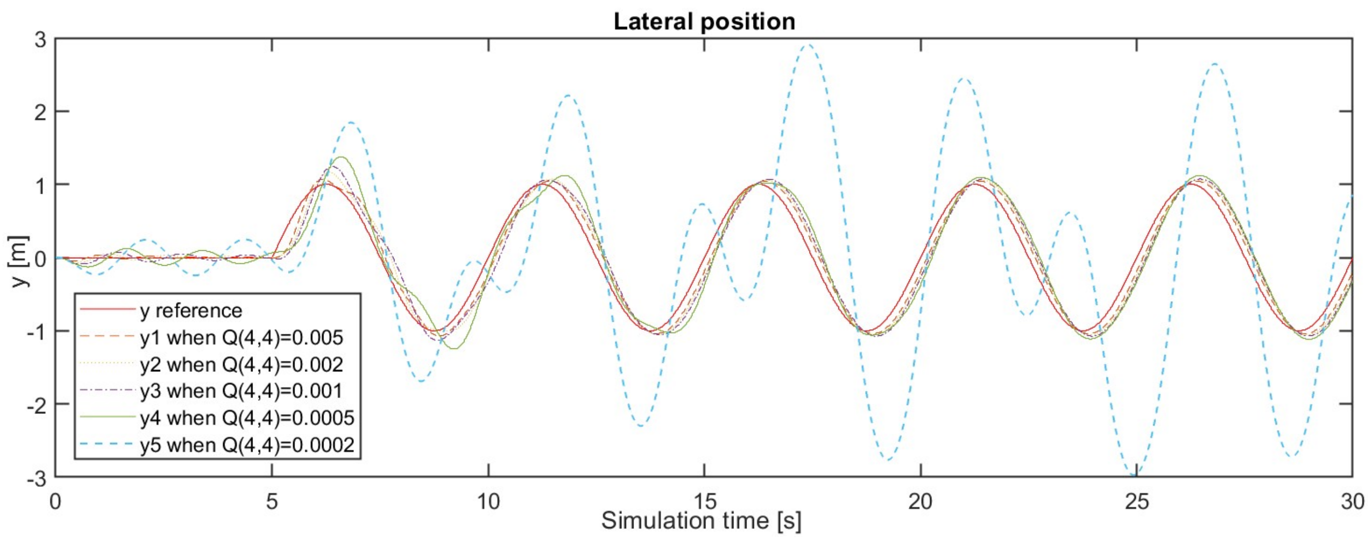

Fig. 3. The lateral position when decreasing $Q(4, 4)$.

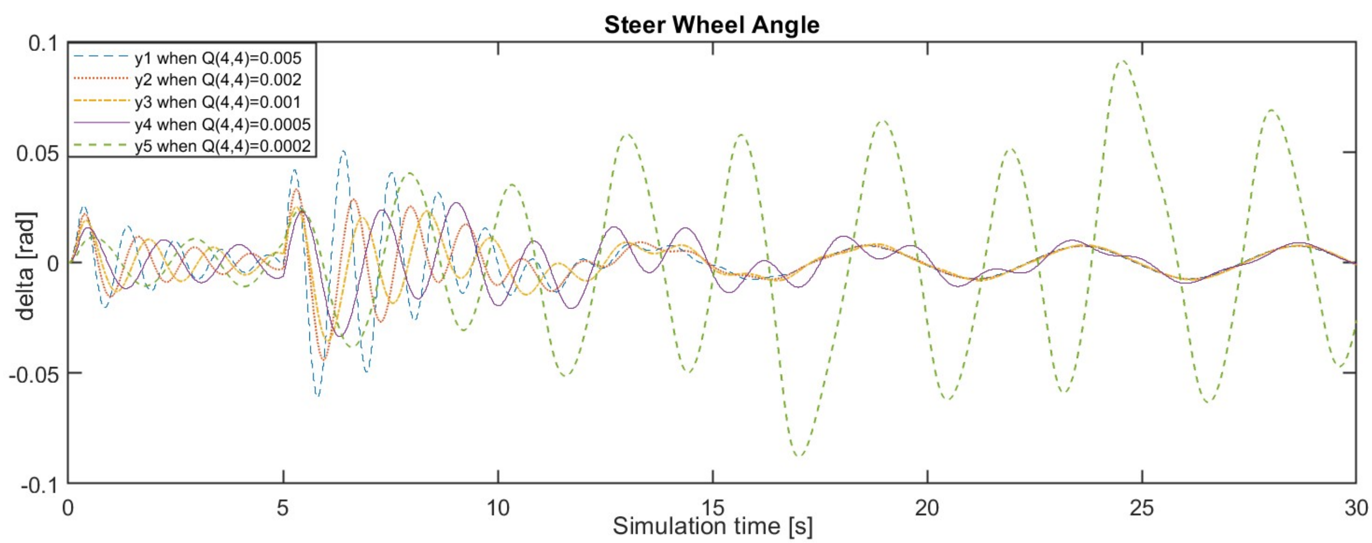

Fig. 4. The steer wheel angle when decreasing $Q(4, 4)$.

2) The analysis of weighting tuning for control input: In this part, the value of $Q(4, 4) = q_{tune}$ is set to be fixed at 0.005, and the $R = r_{tune}$ is increased and decreased.

First considering gradually increasing the value of R from 0.1 to 10, the Root Mean Square Error(RMSE) is decreasing from 0.1274 to 5.4919 and there is a tremendously increased RMSE when R reaches 10. In Fig (5), the plots of lateral position are indicating that when increasing R the tracking performance of MPC is weakened and when R reaches 10, the tracking performance is deteriorated which leads to the failure of MPC. In Fig (6), the steer wheel angle shows the same character of tracking performance. When $R = 10$, the steer wheel angle oscillates with large amplitude and does not have the trend to reduce.

When decreasing the value of R from 0.1 to 0.05, the Root Mean Square Error(RMSE) is decreasing from 0.1274 to 0.1023 of lateral position to y reference. When decreasing to the value of 0.02, the plots of lateral position are not tracking the y reference anymore. In this scenario decreasing R leads to higher value of RMSE and worse tracking performance shown in Fig (7). Furthermore when R reached 0.02, the weighting value severely deteriorates the tracking performance and leads to a significantly increased RMSE. It is obvious that the system becomes unstable with $R = 0.02$. In Fig (8), the steer wheel angle shows the same character of tracking performance. When $R = 0.02$, the steer wheel angle does not have the decreasing amplitude of oscillation around zero point when decreasing R from 0.1 but not making it reach 0.02, the amplitude of steer wheel angle stays small which could indicate that the performance of lateral position is weakened but not deteriorated to a unstable system (MPC fails).

In Table II, different values of weighting matrices R with fixed Q of MPC controller and corresponding Root Mean Squared Error(RMSE) are shown. It could be concluded that when decreasing R the tracking performance of lateral position is enhanced but when R is too large or too small, the tracking performance is deteriorated which cause the failure of MPC controller.

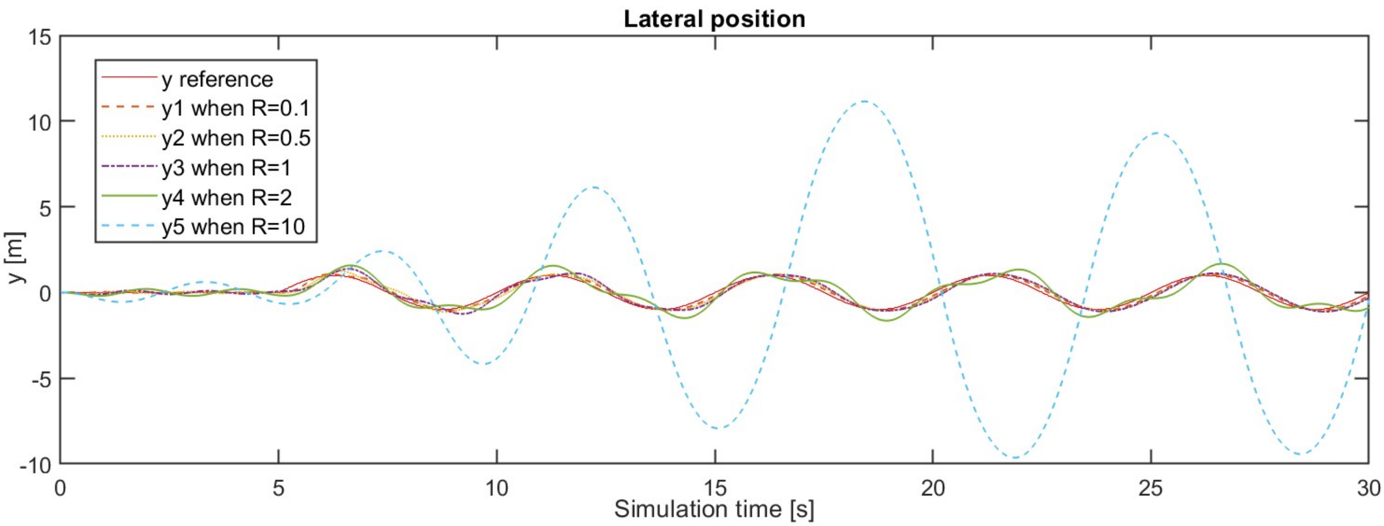

Fig. 5. The lateral position when increasing the R.

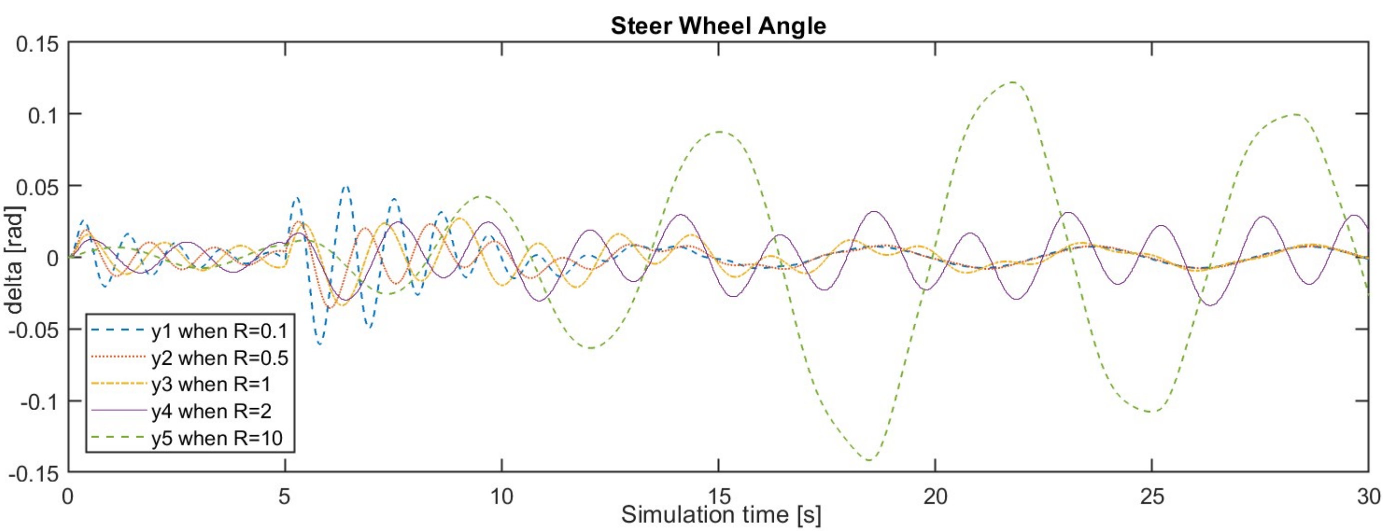

Fig. 6. The steer wheel angle when increasing the R.

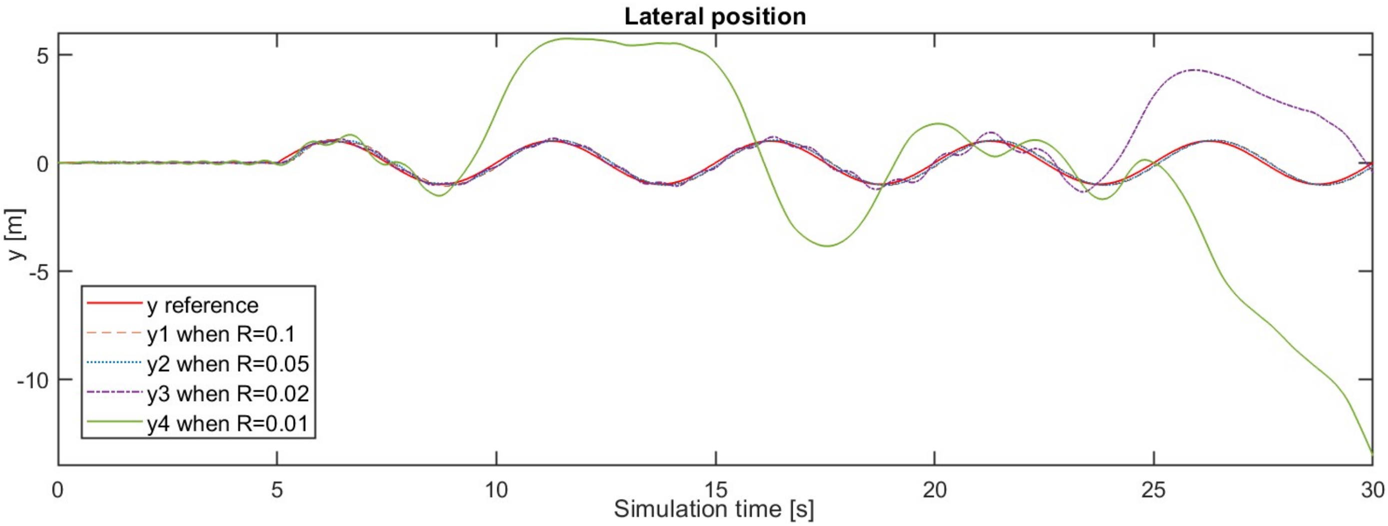

Fig. 7. The lateral position when decreasing the R.

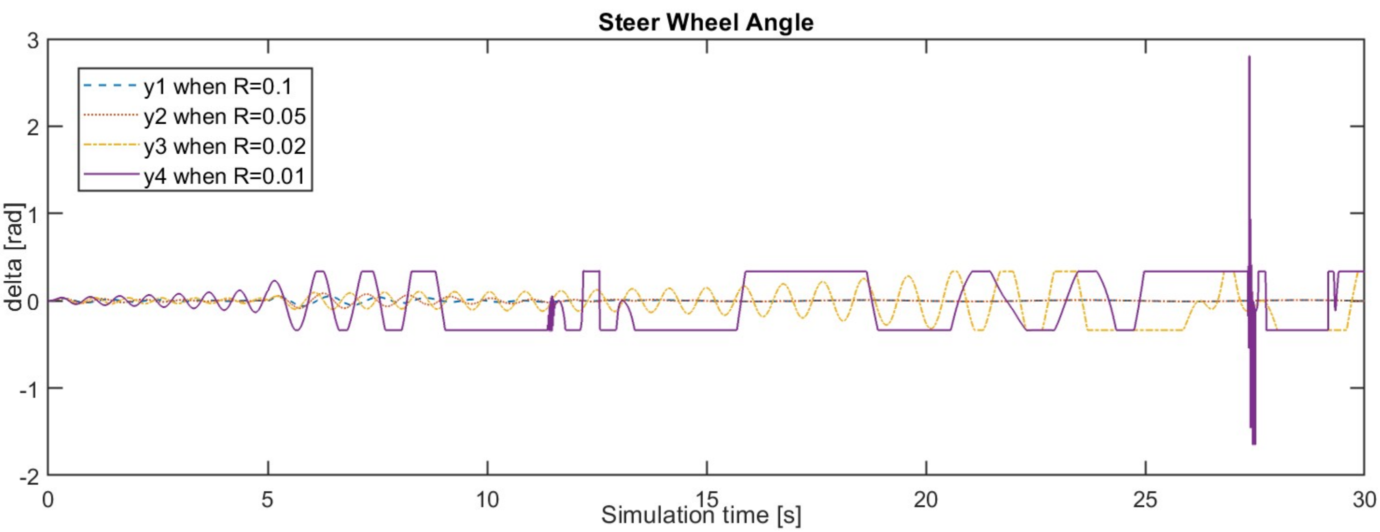

Fig. 8. The steer wheel angle when decreasing the R.

Steps	$Q(4,4)$	R	Root Mean Squared Error (RMSE)
1	0.005	0.01	3.9218
2	0.005	0.02	1.2811
3	0.005	0.05	0.1023
4	0.005	0.1	0.1274
5	0.005	0.5	0.1935
6	0.005	1	0.2325
7	0.005	2	0.3953
8	0.005	10	5.4919

TABLE II
DIFFERENT VALUES OF WEIGHTING MATRICES R WITH FIXED Q OF MPC CONTROLLER AND CORRESPONDING ROOT MEAN SQUARED ERROR(RMSE).

III. DESIGNED MPC PATH-FOLLOWER

A. *Linear bicycle model*

In this part, the linear bicycle model is constructed as extended controller plant instead of a kinematic model. It is worth mentioning that the steering rate is taken as control input of linear bicycle model and the dynamics of $\ddot{y}$ and $\ddot{\psi}$ are both related to δ. Thus the augmented state vector is $\left[y, \dot{y}, \psi, \dot{\psi}, \delta\right]^T$. The extra state equation is shown as:

$$\frac{\mathrm{d}\delta}{\mathrm{d}t} = \dot{\delta}$$

The linear bicycle model is represented as:

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{\delta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{C_{af}+C_{ar}}{mu} & \frac{C_{af}+C_{ar}}{m} & \frac{l_r C_{ar}-l_f C_{af}}{mu} & \frac{C_{af}}{m} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{l_r C_{ar}-l_f C_{af}}{I_{zz}u} & \frac{l_f C_{af}-l_r C_{ar}}{I_{zz}} & -\frac{l_r^2 C_{ar}+l_f^2 C_{af}}{I_{zz}u} & \frac{l_f C_{af}}{I_{zz}} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \\ \delta \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B} \dot{\delta} \tag{4}$$

where C_{af}, C_{ar}, m, l_f, l_r and I_{zz} represent front axle cornering stiffness, rear axle cornering stiffness, vehicle mass in [kg], distance from front axle to CoG in [m], distance from rear axle to CoG in [m] and body inertia around z-axis in [kgm^2] respectively.

B. *MPC controller design*

Based on linear bicycle model, the MPC controller is designed as:

$$\begin{aligned} &\min J = \min \int x^T Q x + u^T R u dt = \min_{\delta} \sum_{i=1}^{N_p} (\|x_{k+i}\|^2 Q + \left\|\dot{\delta}_k\right\|^2 R) \\ &s.t. \quad x_{k+1} = A_d x_k + B_d u_k, \quad -23deg \leq \delta_k \leq 23deg \quad -51.95deg/s \leq \Delta\delta_k \leq 51.95deg/s \end{aligned} \tag{5}$$

where the predict horizon $N_p = 40$ and $x_{k+1} = A_d x_k + B_d u_k$ represents the discrete linear bicycle model with sampling time 0.01s at forward speed u. In this assignment, the forward speed will have the constant reference $V_{ref} = 100$ km/h and A_d and B_d are discretized matrices of A and B in equation (4). Apart from the constraint of δ_k, there is also the constraint for control input of linear bicycle model $\dot{\delta}$. In MATLAB, the ACADO setup is used to formulate linear bicycle model as:

```
% DifferentialState y y_dot psi psi_dot delta; % definition of controller states
DifferentialState y y_dot psi psi_dot delta; % definition of controller states

Control delta_dot; % definition of controller input

% State Differentials
Ct = par.Caf + par.Car;
Cs = par.l_f*par.Caf - par.l_r*par.Car;
Vx = V_ref;
Cq2 = par.l_f^2*par.Caf + par.l_r^2*par.Car;
f_ctrl = [
    dot(y)   == y_dot;...
    dot(y_dot) == -Ct/(par.mass*Vx)*y_dot + Ct/par.mass*psi - Cs/(par.mass*Vx)*
   psi_dot + par.Caf/par.mass*delta;...
    dot(psi)  == psi_dot;...
```

```
 dot(psi_dot) == - Cs/(par.Izz*Vx)*y_dot + Cs/par.Izz*psi - Cq2/(par.Izz*Vx)*
psi_dot + par.l_f*par.Caf/par.Izz*delta;...
 dot(delta) == delta_dot];
```

Correspondingly the weighting matrix is Q associated with the state x thus having the dimension $\Re^{5\times5}$ and R represents the importance of control input $\dot{\delta}$ to cost function and is a scalar.

C. Simulation

In this part, the matrices of Q and R are tuned as in Table III. It is worth mentioning that the first, third and fifth entry on the diagonal of Q matrix, which corresponding the importance of states y, $/psi$ and δ, are set to be nonzero because they are the most influential of all states in Path-Follower.

$Steps$	Q	R	Root Mean Squared Error (RMSE)
1	$diag(1, 0, 1, 0, 1)$	0.1	5.4247
2	$diag(1, 0, 10, 0, 1)$	0.1	2.0266
3	$diag(1, 0, 10, 0, 0.01)$	0.01	$3, 335$
4	$diag(1, 0, 10, 0, 0.01)$	0.001	0.0919
5	$diag(1, 0, 10, 0, 0.01)$	0.0001	0.0846
6	$diag(1, 0, 10, 0, 0.01)$	0.0001	0.2325
7	$diag(1, 0, 9, 0, 0.01)$	0.0001	0.083
8	$diag(1.5, 0, 9, 0, 0.01)$	0.0001	0.0772
9	$diag(1.5, 0, 9, 0, 0.02)$	0.0001	0.0755

TABLE III
DIFFERENT VALUES OF WEIGHTING MATRICES R AND Q OF MPC CONTROLLER AND CORRESPONDING ROOT MEAN SQUARED ERROR(RMSE).

The optimal matrices of Q and R are thus chosen as:

$$Q = \begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.02 \end{bmatrix} \qquad R = 0.0001$$

The plots of lateral position, steer wheel angle and steer wheel rate under the weighing matrices Q and R are shown in Figs (9), (10) and (11) respectively. The RMSE of lateral position is 0.0755 and the lateral position could track the reference lateral position well.

Although the vehicle's tracking performance for the lateral position is commendable, characterized by a low Root Mean Square Error (RMSE), it is important to note that the plots of the steer wheel angle and steer wheel angle rate exhibit high-frequency variations and larger fluctuations. These irregularities may pose challenges in effectively controlling the steering system.

Furthermore, it is worth highlighting that the performance of the Model Predictive Control (MPC) based on a linear bicycle model demonstrates less robustness compared to the nonlinear kinematic model. This suggests that the linear model may struggle to adapt to varying control parameters caused by real plant which is actually nonlinear vehicle model but is linearized as a linear bicycle model and taken as system dynamic constraint in MPC.

IV. COMPARISON

The results from this implementation regarding the efficiency of reference tracking are illustrated in Figure 3.1. These results also correspond to an RMSE error of 0.0755, which is lower than the one found in the default MPC path-follower (the minmal RSME value is 0.1023 with $Q(4, 4) = 0.01$ and $R = 0.1$).

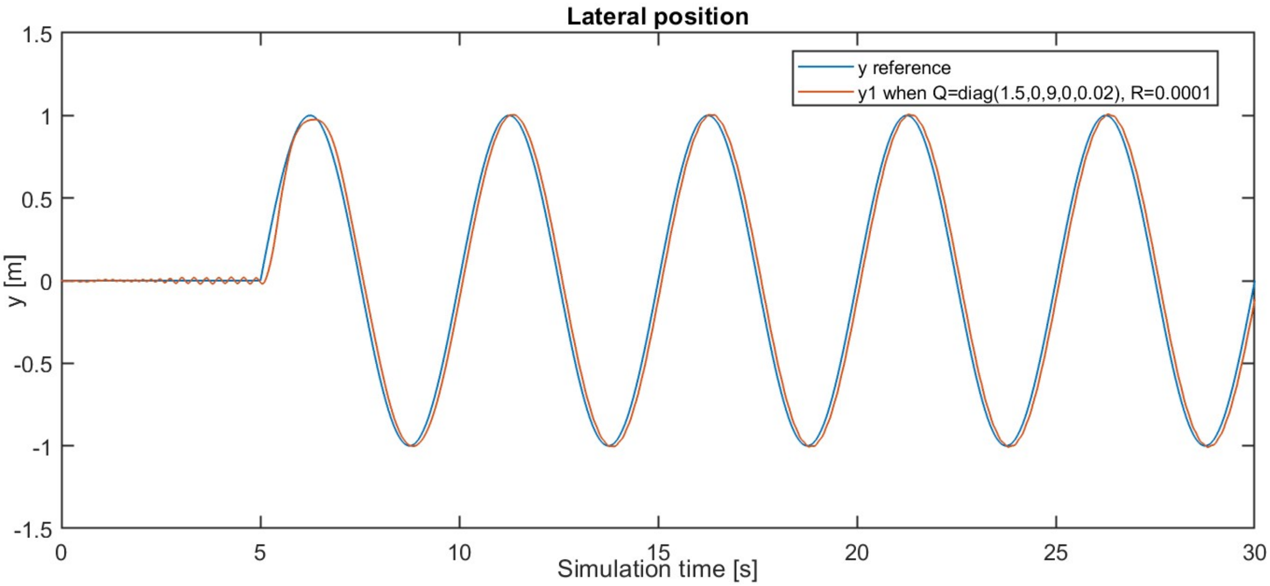

Fig. 9. The lateral position when $Q = diag(1.5, 0, 9, 0, 0.02)$ and $R = 0.0001$.

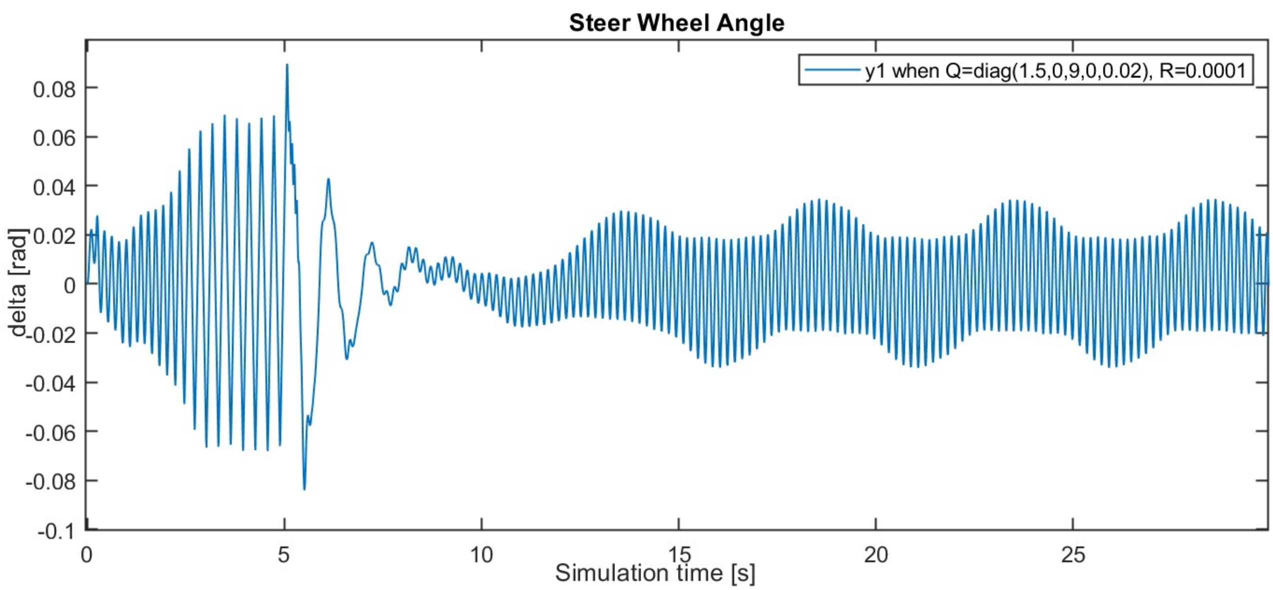

Fig. 10. The steer wheel angle when $Q = diag(1.5, 0, 9, 0, 0.02)$ and $R = 0.0001$.

The reasons of improvement in the tracking performance when using the linear bicycle model might be as followed:

1) One reason is that when using a nonlinear kinematic model in linear MPC controller, the model needs to be first linearized and then discretized. While the linear bicycle model is only discretized. To some extent, the linearization perhaps could have small loss of model's accuracy to depict the system model and capture system dynamics accurately.
2) The linear bicycle model is augmented system with extra dynamics of steer wheel angle and taking steer wheel angle rate as system control input. This may provide more control information about how the steering angle changes, resulting into a smoother tracking of the reference.

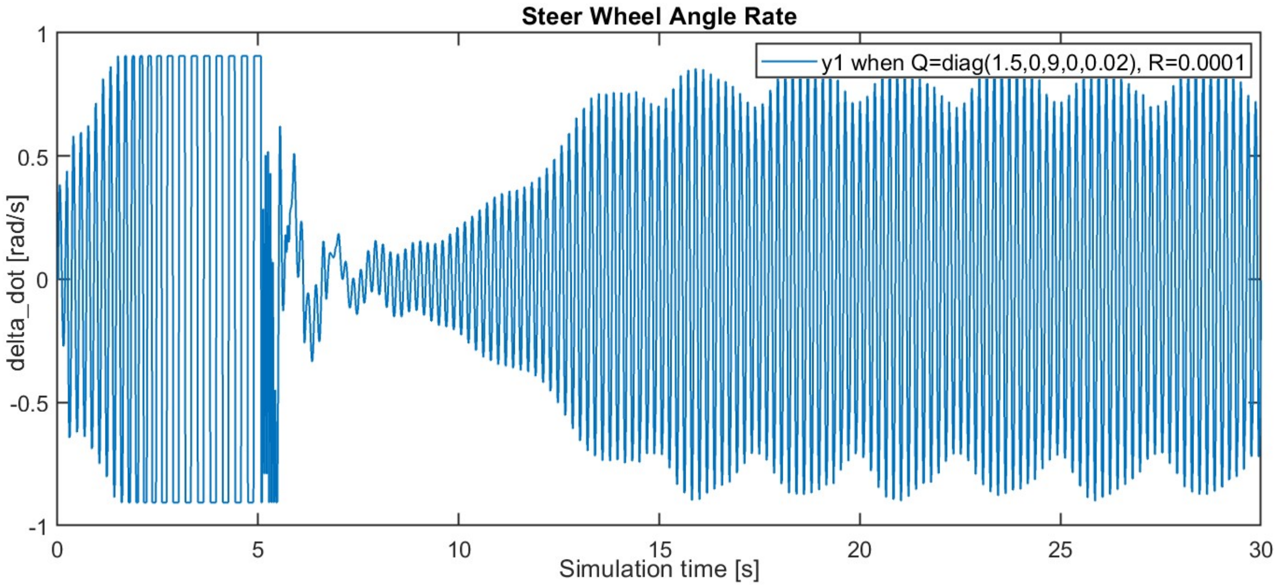

Fig. 11. The steer wheel angle rate when $Q = diag(1.5, 0, 9, 0, 0.02)$ and $R = 0.0001$.

However, using the linear bicycle model has disadvantage that it is overall less robustness are more sensitive of tuning the weighting matrices. This suggests that the linear model may struggle to adapt to varying control parameters caused by real plant which is actually nonlinear vehicle model but is linearized as a linear bicycle model and taken as system dynamic constraint in MPC.

V. Self reflection

Firstly, the most difficult part is about how to interpret kinematic model, linear bicycle model and relation between vehicle parameters and how the vehicle model functions together. This takes me long time to have the basic idea for it. After doing it, the difficulty is to understand the kinematic model is the real plant of system, the linear bicycle model could only be the system dynamics of MPC controller. The states coming out from nonlinear bicycle model. Also it takes me so much time the analyse different values of matrices Q and R. Then learning the structure of MPC control is also a big challenge. Even this task is not easy for me, still do I understood the path-follower control. I am glad that effort leads to great harvest and I will keep on exploring the course of vehicle dynamics and control.

VI. Conclusions

In this assignment the two different MPC path-follower control models are designed and implemented to vehicle model with the goal of performing a manoeuvre through tracking a reference.

The MPC controller of these two models are both linear. For default MPC path-follower, the nonlinear kinematic model is used and the tracking performance under different weighting matrices is analysed. It could be inferred that when values of parameters in weighting matrices are too small or too large, the tracking performance will be deteriorated and make MPC fail. When using the linear bicycle model, the designed MPC path-follower shows better tracking performance with less RMSE but is overall less robustness are more sensitive of tuning the weighting matrices. This suggests that the linear model may struggle to adapt to varying control parameters caused by real plant which is actually nonlinear vehicle model but is linearized as a linear bicycle model and taken as system dynamic constraint in MPC.

References

[1] MF-Tyre/MF-Swift 6.2.0.4 Installation Guide. (2017, March 1). http://www.delft-tyre.nl.

[2] Vehicle Dynamics and Control (RO47017). (2023, May). Retrieved from https://brightspace.tudelft.nl/d2l/le/content/500965/viewContent/2936161/View

[3] PID controller - MATLAB Simulink. (2023, May 15). Retrieved from https://nl.mathworks.com/help/simulink/slref/pidcontroller.html

Appendix

A. Simulink block diagrams.

The Simulink block diagram of default MPC path-follower based on kinematic model and designed MPC path-follower based on linear bicycle model are shown in Figs 12 and 13 respectively.

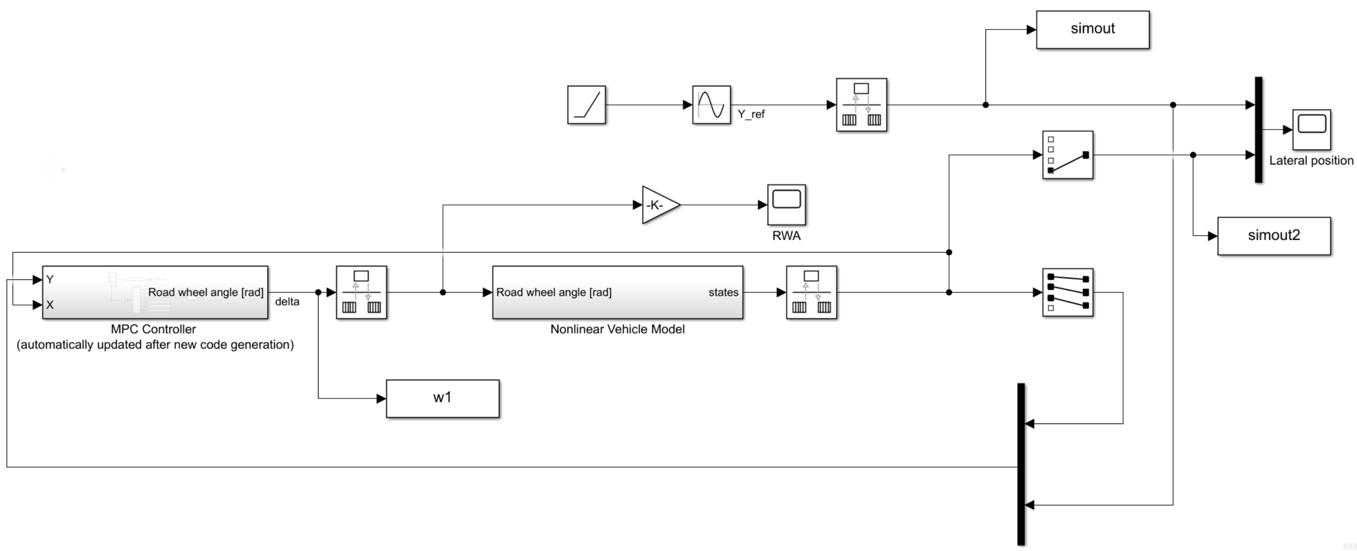

Fig. 12. The block diagram of default MPC path-follower based on kinematic model.

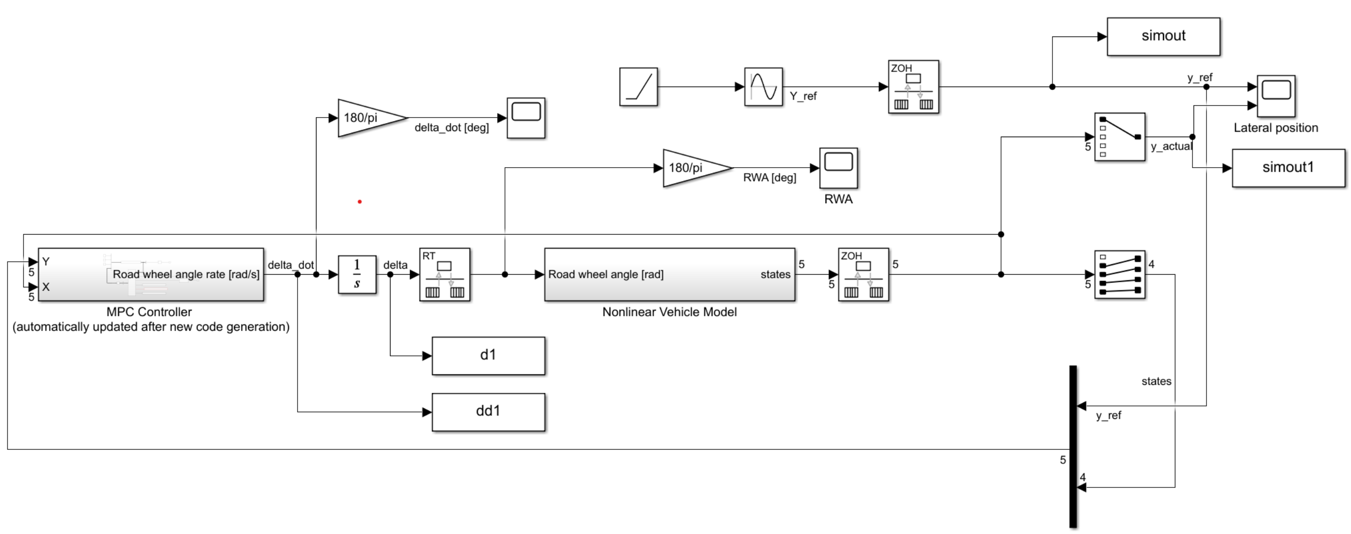

Fig. 13. The block diagram of designed MPC path-follower based on linear bicycle model.

B. Vehicle parameters.

Vehicle parameters	Values
Initialization velocity	50 km/h
Vehicle mass	1380 kg
Body inertia around z-axis	2634.5 kgm^2
Wheelbase	2.79 m
Distance from front axle to CoG	1.384 m
Steering ratio	15.4
Front axle cornering stiffness	120000
Rear axle cornering stiffness	190000
Friction coefficient	1

TABLE IV
VEHICLE PARAMETERS.