MPC Tracking Controller Parameters Impacts in Roundabouts

Hang Cao * and Mate Zoldy *

Department of Automotive Technologies, Budapest University of Technology and Economics, 1111 Budapest, Stoczek J. u. 6., Hungary
* Correspondence: hang.cao@edu.bme.hu (H.C.); zoldy.mate@kkj.bme.hu (M.Z.)

Abstract: The work is proposed to design a controller of the form known as the roundabout scenario trajectory tracking problem. The road condition is a four-leg, single-lane roundabout; the reference path is given. Due to the decision choice of exits, the MPC tracking controller is used to test the effect of weight parameter and target speed on the performance of the tracking controller. Two sets of test cases are proposed to make the experimental comparison, see the relationship between the control parameters and road conditions (different curvature path), and also see how the weight parameters Q and R and sample time affect the tracking performance. Our work, MPC controller utilization in a roundabout, plays an essential role with the increasing autonomy of vehicles.

Keywords: MPC; roundabout; weight parameter; path tracking

1. Introduction and Motivation

Self-driving vehicles are now receiving more and more attention and research from academic and industrial communities and are maturing in various research directions. Many test conditions are most intensively studied in the highway scenario, while the work on roundabouts needs more time and effort to be further studied.

Linear model predictive control (LMPC) has been widely used in the industry and developed well. Due to tight performance specification demands in the academic and industry researching process, nonlinear model predictive control (NMPC) [1–3] plays a significant role. For example, NMPC is the main model that used in tracking ability testing [4]. Proposed dealing with fuzzy adaptive weight variables to fit the autonomous path tracking cost function with a refined model predictive control (MPC) controller. Considering the vehicle dynamic problem, the authors mainly focused on the tracking accuracy and driving comfort. The result was verified as a comparison study with classical MPC controller and pure-pursuit controller and showed a better tracking performance. Research work in [5] built an MPC simultaneous controller to reach the trajectory planning and tracking purpose. The novel idea of the optimization problem is to use an initial and intermediate two sets of variables to define the MPC control sequence and basic information of lane-change trajectories, including velocity. Unconstrained and constrained conditions with lane change to the left and right directions are all discussed intensely in the paper. An MPC controller was used to track piecewise clothoid trajectory in work [6]. In this method, the authors considered the ego car constraints, the yaw rate conditions, and the predefined boundary constraints. The proposed idea is based on the robust control invariant (RCI) system to ensure the boundaries and constraints are fully satisfied. Demonstrating the good tracking ability of NMPC, the authors presented their work as [1], which utilized lateral driving behavior with an NMPC controller. This method was transformed through a linear time-varying (LTV) MPC to control different constraints and vehicle conditions. Two different study cases verified the validity of the NMPC controller’s tracking ability and accuracy. The work in [7] calculated the ego car velocity and steering demand with a built nonlinear MPC controller framework. Using the genetic algorithms (GA) as an optimization solver, it met the flexible MPC structure requirement, and the proposed method provided the tracking accuracy based on different road conditions.
Different researchers have also studied MPC with different methods and purposes. A studied MPC controller dealt with changing the horizon parameter and illustrated the key point of selecting vehicle model and optimization solver [8]. Brake and steering behavior was tested with a designed MPC controller, changing over time as the ego car front-steering angle controller followed the predefined path [9]. The paper [10] studied steering behavior with the road surface conditions, used tire parameters as variables, and estimated the initial tire-force curve slope by the NMPC control method. In this case, the nonlinear tire curve was fully adapted by the proposed NMPC controller, leading the vehicle into a stable and well-adapted driving level. To avoid the computational difficulties, a single dimension artificial potential fields approach (SDAPF) was used to obtain the environments and build a convex quadratic programming (CQP)-based MPC controller [11]. Highway lane-merge action is also an important module in autonomous vehicle simulation and testing. Researchers developed a cooperative nonlinear MPC method implementing the lane merge scenario via ACADO code. The proposal of performance-cost function and the receding horizon was also validated in the numerical results of the paper [12].

There are few research-built MPC scheme applied to the roundabout road scenario. Our goal is building a MPC system both related to the global planner and local planner, applied to the whole roundabout scenario. In this case, the controller parameter tuning is an important issue.

**Hypothesis 1.** A steady state of trajectory controller can be built and run out, fitting in the curvature and changing roundabout scenario cases.

Investigation novelty:
- The ability of the MPC-optimization method for path tracking in the given roundabout environment is proven.
- The relationship between MPC controller parameters and tracking performance based on roundabout road features is analyzed.

In this paper, our research is based on the roundabout global planner, which has been built to test the vehicle trajectory-tracking performance. We built an MPC tracking controller that can fit the roundabout scenario and track the continuous large curvature path smoothly. Roundabout road condition is a special case in urban scenarios. The key point of this paper is building the MPC model, which can connect to the roundabout global planner, and discussing the parameters of controller impacts on the tracking performance in a roundabout path. The optimization problem is set as a convex problem and is to be solved at each sample by an efficient quadratic programming (QP) method, the slack variable designed as a solver to the inequality constraints. Our work provides further analysis and development of applied mathematics to autonomous driving.

The remaining sections of this article are organized as follows: Section 2 describes the related work of the MPC design methodology and fundamental theory. This section gives a comprehensive and straightforward overview of MPC system from [13–21]. The MPC system-cost function and scenario constraints are described in Section 3. Section 4 presents the performance analysis of the proposed MPC parameters changing under various scenarios and discussion on the choice of MPC parameters for each roundabout scenario. Section 5 concludes the work with future direction.

2. Related Works

2.1. Vehicle Kinematics Model

Since the vehicle is not considered to be driven at high speed, the state, control, and reference vectors, respectively, are denoted as

\[
\begin{bmatrix}
\dot{X} \\
\dot{Y} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
\cos \phi \\
\sin \phi \\
\tan \delta_f / l
\end{bmatrix} v
\]  

(1)
The state of the model is \( \xi = [X, Y, \varphi]^{T} \), and the control quantity of the model is the steering angle of the front wheel, \( u = [v, \delta_{f}]^{T} \).

### 2.2. Model Predictive Control Fundamental Theory

Control system is the fundamental work for MPC. Special attention is given to the observer design in work [22]. The system model \( (1) \) obtained by kinematic modeling is a nonlinear system that cannot be directly used for linear time-varying model predictive control. The linearization of the nonlinear system needs to be performed first. As a widely used linearization method, Taylor series expansion can approximate the value at the operating point of the nonlinear system very well using polynomials.

Figure 1 illustrates the reference trajectory system and MPC trajectory tracking system relationship. The state \( \xi_{\text{ref}} \) is obtained from roundabout reference trajectory module as the waypoints \( X_{\text{ref}}, Y_{\text{ref}} \), and the MPC system control inputs are the reference velocity \( V_{\text{ref}} \). Assume that the state \( \xi_{\text{ref}} \) and control quantities \( u_{\text{ref}} \) at any given moment satisfy the following equations; \( k \) is the sampling instant,

\[
\dot{\xi}_{\text{ref}}(k+1) = f(\xi_{\text{ref}}(k), u_{\text{ref}}) \tag{2}
\]

The above reference equation performs a Taylor expansion at the point \( (\xi_{\text{ref}}, u_{\text{ref}}) \), ignoring the higher-order terms and retaining only the first-order terms

\[
\dot{\xi} = f(\xi_{\text{ref}}, u_{\text{ref}}) + J_f(\xi) \left( \xi - \xi_{\text{ref}} \right) + J_f(u) \left( u - u_{\text{ref}} \right) \tag{3}
\]

\( J_f(\xi) \) and \( J_f(u) \) are the Jacobi matrices of \( f \) concerning \( \xi \) and \( u \), respectively. The equation after linearization is obtained as

\[
\dot{\xi} = A(t)\hat{\xi}(t) + B(t)\bar{u}(t) \tag{4}
\]

where \( \hat{\xi}(t) = \xi - \xi_{\text{ref}}, \bar{u}(t) = u - u_{\text{ref}}, A(t) = J_f(\xi), B(t) = J_f(u) \)

\[
\begin{bmatrix}
    \dot{x} - \dot{x}_{\text{ref}} \\
    \dot{y} - \dot{y}_{\text{ref}} \\
    \dot{\varphi} - \dot{\varphi}_{\text{ref}}
\end{bmatrix} =
\begin{bmatrix}
    0 & 0 & -vsin\varphi \\
    0 & 0 & vcos\varphi \\
    0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    x - x_{\text{ref}} \\
    y - y_{\text{ref}} \\
    \varphi - \varphi_{\text{ref}}
\end{bmatrix} +
\begin{bmatrix}
    cos\varphi & 0 & 0 \\
    sin\varphi & 0 & v \\
    tan\delta & 0 & \frac{v}{lcos\delta}
\end{bmatrix}
\begin{bmatrix}
    \delta_f - \delta_{\text{ref}}
\end{bmatrix} \tag{5}
\]

After discretization, the system state can be represented as

\[ A_{k,t} = I + TA(t)B_{k,t} = TB(t) \tag{6} \]
I is the unit matrix and \( T \) is the discrete step length; \( k \) is the sampling instant. The results of the discretization using Euler’s method are

\[
\tilde{\xi}(k+1) = A_{k,t} \tilde{\xi}(k) + B_{k,t} \Delta u(k)
\]  

(7)

\[
A_{k,t} = I + TA(t) = \begin{bmatrix}
1 & 0 & -vsin\phi T \\
0 & 1 & vcos\phi T \\
0 & 0 & 1 \\
\end{bmatrix}, \\
B_{k,t} = TB(t) = \begin{bmatrix}
cos\phi T & 0 & \tan\delta f T \\
cos\phi T & 0 & \frac{vT}{\cos^2\delta f} \\
\end{bmatrix}
\]  

(8)

(9)

And the observation equation represents as

\[
\eta(k) = C_{k,t} \tilde{\xi}(k)
\]  

(10)

\[
C_{k,t} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]  

(11)

In the next process of designing the objective function, replacing the control with the control increment and adding the relaxation factor can directly restrict the control increment and prevent the situation that there is no feasible solution during the execution.

Setting new state variables as \( \tilde{\xi}(k|t) = \begin{bmatrix} \tilde{\xi}(k|t) \\
u(k-1|t) \end{bmatrix} \), the control variable from the previous moment is added to the state variable, and the control increment is used as the new control variable. According to the relationship between the control volume and the control increment, it is obtained that

\[
\hat{\xi}(k+1|t) = \tilde{A}_{k,t} \tilde{\xi}(k|t) + \tilde{B}_{k,t} \Delta u(k|t) \\
\eta(k|t) = \tilde{C}_{k,t} \tilde{\xi}(k|t)
\]  

(12)

As given the system state quantity \( \hat{\xi}(k|t) \) and the control increment \( \Delta u(k|t) \) at the moment \( t \), the output quantity \( \eta(k+1|t) \) of the system (12) at the next moment can be calculated by equation, and the system output quantity in the prediction time domain can be obtained by continuously iterating the calculation, as

\[
A_{k,t} = A_t, k = t, \ldots t + P \\
B_{k,t} = B_t, k = t, \ldots t + P \\
C_{k,t} = C_t, k = t, \ldots t + P
\]  

(13)

The output in \( P \) step represents in the following way:

\[
\eta(t+P|t) = \tilde{C}_t \hat{A}^P \hat{\xi}(t|t) + \tilde{C}_t \hat{A}_t \hat{B}_t \Delta u(t|t) + \cdots + \tilde{C}_t \hat{A}_t \hat{B}_t \Delta u(t+ N|t)
\]  

(14)

The above calculation process between state, control, and output quantities in the predicted time domain can be organized as follows:

\[
Y(t) = \Psi_t \hat{\xi}(t|t) + \Theta_t \Delta U(t)
\]  

(15)
and the system predicts the output and input sequences as

$$Y(t) = \begin{bmatrix}
\eta(t+1|t) \\
\eta(t+2|t) \\
\vdots \\
\eta(t+N|t) \\
\vdots \\
\eta(t+P|t)
\end{bmatrix}, \quad (16)$$

$$\Delta U(t) = \begin{bmatrix}
\Delta u(t|t) \\
\Delta u(t+1|t) \\
\vdots \\
\Delta u(t+N|t)
\end{bmatrix}, \quad (17)$$

in this case, the prediction matrices can be expressed, respectively, as

$$\Psi_t = \begin{bmatrix}
\tilde{c}_f \tilde{A}_t \\
\tilde{c}_r \tilde{A}_t^2 \\
\vdots \\
\tilde{c}_r \tilde{A}_t^P
\end{bmatrix}, \quad (18)$$

$$\Theta_t = \begin{pmatrix}
\tilde{c}_r \tilde{A}^{-1} \tilde{B}_t & 0 & 0 & \cdots & 0 \\
\tilde{c}_r \tilde{A}^{-1} \tilde{B}_t & \tilde{c}_r \tilde{A}_t \tilde{B}_t & \cdots & \tilde{c}_r \tilde{B}_t \\
\tilde{c}_r \tilde{A}^{-1} \tilde{B}_t & \tilde{c}_r \tilde{A}_t \tilde{B}_t & \cdots & \tilde{c}_r \tilde{A}^{R-N-1} \tilde{B}_t
\end{pmatrix}, \quad (19)$$

3. Model Predictive Control System

The essential point of MPC controller is to control the input by designing and solving the cost function. Minimizing the error between the future system output and the reference output obtains a stable vehicle state tracking system. As shown in Figure 2, the main steps of the model predictive control system are built. In this section, model calculation and optimization are the key content.

3.1. Objective Function

For the path tracking controller, the goal in this section is to follow the reference path quickly and stably, so the objective function is designed as follows:

$$J(\tilde{z}(t), u(t-1), \Delta U(t)) = \sum_{i=1}^{P} \| \eta(t+i|t) - \eta_{ref}(t+i|t) \|^2 + \sum_{i=0}^{N-1} \| \Delta u(t+i|t) \|^2 + \rho \varepsilon^2 \quad (20)$$

The role of $\sum_{i=1}^{P} \| \eta(t+i|t) - \eta_{ref}(t+i|t) \|^2$ is used to be the deviation between the state output of the predicted system and the reference system reference in the prediction horizon, reflecting the system’s tracking ability to the reference trajectory. $\sum_{i=0}^{N-1} \| \Delta u(t+i|t) \|^2$ reflects the constraints on the control quantity, i.e., the smoothness of the control. By adjusting the weight matrices $Q$ and $R$, the system can quickly and stably follow the reference trajectory. The slack variable is mainly to avoid the situation that there is no feasible set.
3.1. Objective Function

For the path tracking controller, the goal in this section is to follow the reference path quickly and stably, so the objective function is designed as follows:

$$J(\xi(t), u(t-1), \Delta U(t)) = \sum_{i=1}^{5} \|\eta(t+i|t) - \eta_{ref}(t+i|t)\|^2 + \sum_{i=1}^{5} \|\Delta u(t+i|t)\| + \rho \varepsilon^T \varepsilon$$  \hspace{1cm} (20)

The role of $$\sum_{i=1}^{5} \|\eta(t+i|t) - \eta_{ref}(t+i|t)\|^2$$ is used to be the deviation between the state output of the predicted system and the reference system reference in the prediction horizon, reflecting the system's tracking ability to the reference trajectory. $$\sum_{i=1}^{5} \|\Delta u(t+i|t)\|$$ reflects the constraints on the control quantity, i.e., the smoothness of the control. By adjusting the weight matrices $Q$ and $R$, the system can quickly and stably follow the reference trajectory. The slack variable is mainly to avoid the situation that there is no feasible set.

3.2. QP (Quadratic Programming) Standard Form Transformation

To solve the objective function and constraints upward, some corresponding matrices transform are needed, and let those functions turn to quadratic programming problem, which is easier to calculate.

$$\min_x \frac{1}{2} x^T H x + f^T x$$  \hspace{1cm} (21)

subject to

$$\begin{align*}
A \cdot x & \leq b \\
A_{eq} \cdot x &= b_{eq} \\
lb \leq x \leq ub
\end{align*}$$  \hspace{1cm} (22)

$H$ is the positive definite Hessian matrix; $A, b$ are the inequality constraint matrices of the control variables; $A_{eq}$ and $b_{eq}$ are the equation constraint matrices of the control variables; $lb, ub$ are the upper and lower constraints of the control variables, respectively.

Define the deviation of the output quantity in the predicted horizon with

$$E(t) = Y_{ref}(t|t) - Y_r(t)$$  \hspace{1cm} (23)

where the reference value $Y_{ref}(t) = [\eta_{ref}(t+1|t), \ldots, \eta_{ref}(t+P|t)]$. Substituting (23) into (20), the function is transformed as

$$J(\xi(t)), u(t-1), \Delta U(t)) = [\Delta U(t)^T, \varepsilon]^T H [\Delta U(t)^T, \varepsilon]^T + G [\Delta U(t)^T, \varepsilon]^T + P_t$$  \hspace{1cm} (24)
where
\[
H_t = \begin{bmatrix} \Theta_t^T Q_e \Theta_t + R_e & 0 \\ 0 & \rho \end{bmatrix}
\]
(25)
\[
G_t = \begin{bmatrix} 2E(t)^T Q_e \Theta_t \\ 0 \end{bmatrix}
\]
(26)
\[
P_t = E(t)^T Q_e E(t)
\]
(27)

\(Q_e\) and \(R_e\) are expanded forms of the weight matrices \(Q\) and \(R\), respectively.

\[
Q_e = \begin{bmatrix} Q_{p \times p} & 0_{p \times p} & \cdots & 0_{p \times p} \\ 0_{p \times p} & Q & \cdots & 0_{p \times p} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{p \times p} & 0_{p \times p} & \cdots & Q \end{bmatrix}_{p \times p}
\]
(28)
\[
R_e = \begin{bmatrix} R_{u \times u} & 0_{u \times u} & \cdots & 0_{u \times u} \\ 0_{u \times u} & R & \cdots & 0_{u \times u} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{u \times u} & 0_{u \times u} & \cdots & R \end{bmatrix}_{u \times u}
\]
(29)

The final form of the objective function is described in the following way:

\[
J(\hat{x}(t), u(t-1), \Delta U(t)) = \Delta \hat{U}(t) H_t \Delta \hat{U}(t)^T + G_t \Delta \hat{U}(t)^T
\]
(30)

where
\[
\Delta \hat{U}(t) = [\Delta U(t)^T, e]^T
\]
(31)

3.3. Constraints Setting

The constraints adopted MPC controller are mainly set in three aspects, respectively:

- Control constraints

\[
u_{\min}(t+k) \leq u(t+k) \leq u_{\max}(t+k), k = 0, 1, \ldots, N
\]
(32)

- Control incremental constraints

\[
\Delta u_{\min}(t+k) \leq \Delta u(t+k) \leq \Delta u_{\max}(t+k), k = 0, 1, \ldots, N
\]
(33)

- Output constraints

\[
y_{\min}(t+k) \leq y(t+k) \leq y_{\max}(t+k), k = 0, 1, \ldots, N
\]
(34)

The above three constraints need to be transformed according to the standard form. The relationship between the control quantity and the control increment can be described as (35)

\[
u(t+k) = u(t+k-1) + \Delta u(t+k)
\]
(35)

In the prediction horizon, we can obtain the Equation (36), and \(1_N\) is the N-dimensional column vectors with elements of 1, \(L_N\) is the N-dimensional all lower triangular matrix, \(I_n\) is the unit matrix with the same dimension as the control variable dimension, and \(\otimes\) is the Kronecker product.

\[
\begin{bmatrix} u_{t|t} \\ u_{t+1|t} \\ \vdots \\ u_{t+N|t} \end{bmatrix} = \begin{bmatrix} u_{t-1} \\ u_{t-1} \\ \vdots \\ u_{t-1} \end{bmatrix} + \begin{bmatrix} \Delta u_{t|t} \\ \Delta u_{t+1|t} \\ \cdots \\ \Delta u_{t+N|t} \end{bmatrix}
\]
(36)

\[
= 1_N \otimes u_{t-1} + L_N \otimes I_n \Delta \hat{U}(t)
\]
Finally, the constraint (32) can be transformed as:

\[ U_{\text{min}} A \Delta U_t + U_{t-1} U_{\text{max}} \]

\[ A = L_N \otimes I_u \]

\[ U_{t-1} = 1_N \otimes u_{t-1} \]

constraint (34) can be illustrated as:

\[ Y_{\text{min}} \leq \Psi_t \xi(t|t) + \Theta_t \Delta U(t) \leq Y_{\text{max}} \] (38)

Calculate the Kronecker product of matrices \( A \) and \( B \), and combine with the optimization formulation we mentioned at (30),

\[
A_{\text{cons}} \Delta \tilde{U}_t \leq b_{\text{cons}}
\]

\[
A_{\text{cons}} =
\begin{bmatrix}
A & 0_{u \times N, u \times N} \\
-A & 0_{u \times N, u \times N} \\
\Theta_t & \\
-\Theta_t & 0_{u \times N, u \times N}
\end{bmatrix}
\]

\[
B_{\text{cons}} =
\begin{bmatrix}
U_{\text{max}} - U_t \\
-U_{\text{min}} + U_{t-1} \\
Y_{\text{max}} - \Psi_t \xi(t|t) \\
-Y_{\text{min}} + \Psi_t \xi(t|t)
\end{bmatrix}
\]

Equation (39) can be the constraints (33), and \( M \) is the upper bound of the slack variable \( \varepsilon \).

\[
lb \leq \Delta \tilde{U}_t \leq ub \\
lb = [\Delta U_{\text{min}}; 0] \\
ub = [\Delta U_{\text{max}}; M]
\] (40)

3.4. Geometric Constraints for Vehicles and Structured Roads

Based on the real vehicle steering demand and the roundabout road constraints, the system needs to follow three constraints:

To restrict the lateral stability of the vehicle, predefine the turning angle as ±0.25 rad, the front-wheel steering angle constraint is

\[-0.25 \leq \delta_f \leq 0.25 \] (41)

Considering the ego car safety and comfort, the lateral acceleration set in the range of \([-1, 1]\),

\[-1 \text{ m/s}^2 < \dot{v}_{la} < 1 \text{ m/s}^2 \] (42)

To assure vehicles are driving inside the road, the feasible domain boundary in the road coordinate system is predefined in constraints setting at a half-car width away from the edge of the road.

\[
\left( S_{L_{\text{min}}} - \frac{W_c}{2} \right) \leq S_L \leq \left( S_{L_{\text{min}}} + \frac{W_c}{2} \right) \]

\( S_{L_{\text{max}}} / S_{L_{\text{min}}} \) are the maximum and minimum offsets of the road edges, and \( W_c \) set as the vehicle width.

4. Experimental Results

In this section, the model predictive controller performance is validated through coding in MATLAB 2017b, and the quadratic programming problem is run by MATLAB quadprog solver. Two road conditions are carried out and used to determine \( N_p \) and \( N_c \) parameter values after many tests. Taking the initial control parameters \( N_p = 60, N_c = 30, \)
and \( T = 0.027 \) s, the target speed takes 50 km/h. This work sets a simple PI controller to drive the longitudinal speed.

Case 1 tests the MPC performance while changing the weight parameters \( Q \) and \( R \); Case 2 makes the MPC performance comparison of only changing the target speed and tuning the sample time corresponding to the different speed.

### 4.1. Case 1. Fourth-Leg Enter, Second-Leg Exit

Case 1 tracks the path through the fourth-leg enter area and second-leg exit area and tests the MPC performance while changing the weight parameters \( Q \) and \( R \), as shown in Table 1. Changing three sets of weight parameters alters the tracking performance. Figure 3 shows the tracking route used to change the weight parameter \( Q \) and the weight parameter \( R \).

#### Table 1. Fourth-leg enter, second-leg exit MPC tracking changing parameters.

| No. of Test Cases | Test Description       | \( N_p \) | \( N_c \) | \( T \) (s) | \( Q \) | \( R \) | Target Speed (km/h) |
|------------------|------------------------|---------|---------|---------|------|------|---------------------|
| 1                | Scenario 1 change R    | 60      | 30      | 0.027   | 40   | 0.8  | 50                  |
| 2                | Scenario 1 change Q    | 60      | 30      | 0.027   | 80   | 0.8  | 50                  |
| 3                | Scenario 1 change Q    | 60      | 30      | 0.027   | 8000 | 0.8  | 50                  |
| 4                | Scenario 1 change Q    | 60      | 30      | 0.027   | 80   | 0.8  | 50                  |
| 5                | Scenario 1 change Q    | 60      | 30      | 0.027   | 500  | 0.8  | 50                  |
| 6                | Scenario 1 change Q    | 60      | 30      | 0.027   | 0.5  | 0.8  | 50                  |

The tracking controller performs well when the ego vehicle is going along the straight road and stable curvature curve area. The tracking performance is altered mainly in the two merging zones of the roundabout. Both Figure 4a,b emphasize their tracking differences on the lateral axis when the ego car is approaching the roundabout merging zones. When changing the parameter \( R \), the blue line \((Q = 40, R = 0.8)\) in Figure 4a is the most stable line and is also closer to the reference trajectory. Figure 4b demonstrates that the pink line, \( Q = 80, R = 0.8 \), is the most stable tracking route even in the merging zones. The specific tracking error on longitudinal and later directions are shown in Figure 5a,b.
Table 1. Fourth-leg enter, second-leg exit MPC tracking changing parameters.

| Case | Parameters | Np | Nc | T (s) | Q  | R   | Target Speed (km/h) |
|------|------------|----|----|-------|----|-----|---------------------|
| 1    | Scenario 1 change R weighting parameter | 60 | 30 | 0.027 | 40 | 0.8 | 50                  |
| 2    | Scenario 1 change Q weighting parameter | 60 | 30 | 0.027 | 40 | 80  | 50                  |
| 3    | Scenario 1 change Q weighting parameter | 60 | 30 | 0.027 | 40 | 8000 | 50                  |

The tracking controller performs well when the ego vehicle is going along the straight road and stable curvature curve area. The tracking performance is altered mainly in the two merging zones of the roundabout. Both Figure 4a,b emphasize their tracking differences on the lateral axis when the ego car is approaching the roundabout merging zones.

When changing the parameter R, the blue line (Q = 40, R = 0.8) in Figure 4a is the most stable line and is also closer to the reference trajectory. Figure 4b demonstrates that the pink line, Q = 80, R = 0.8, is the most stable tracking route even in the merging zones. The specific tracking error on longitudinal and lateral directions are shown in Figure 5a,b.

Figure 4. (a) Shows the tracking difference with reference path after changing the weighting parameter R; Q = 40, R = 0.8 achieves the best performance. (b) Compares yaw angle after changing the weighting parameter Q; those performances look similar, but the pink line (Q = 80, R = 0.8) is still closer to the red reference line.

Figure 5a,b are the steady-state errors in longitudinal direction, lateral position, and yaw angle. The idea of the steady-state errors is consistent with the literature [21]. Since the value of errors were positive, the ego car started measurement of yaw angle at the beginning of the simulation, while the longitudinal axis of the actual vehicle was pointing to the right.

At the end of the following path, the ego car was actually on a straight path, as demonstrated clearly by the shape of the lateral position plotting and zero steering angle. Furthermore, the measured yaw angle can be seen as the orientation error, both illustrated in Figure 5a,b.
Figure 5. (a) Shows the comparison of trajectory errors after changing the weighting parameter $R$: $Q = 40$, $R = 0.8$ still achieves the best performance. (b) Compares trajectory errors after changing the weighting parameter $Q$: $Q = 500$, $R = 0.8$ still achieves the best performance.

Figure 6a,b show the vehicle tracking heading performance and the longitudinal velocity. The yaw rate changes rapidly when passing through the curve area.
Figure 6. (a) Shows the comparison of yaw rate after changing the weighting parameter $R$; $Q = 40$, $R = 0.8$ still achieves the best performance. (b) Shows the comparison of yaw rate after changing the weighting parameter $Q$; $Q = 50$, $R = 0.8$ still achieves the best performance.

4.2. Case 2. Fourth-Leg Enter, Third-Leg Exit

Case 2 tracks the path through the fourth-leg enter area and third-leg exit area and tests the MPC performance while changing the target speed and sample time $T$. Figure 7 shows the Case 2 tracking route used to change the target speed and sample time. Two scenario sets are tested in Case 2. As Table 2 shows, tests 1–4 change the vehicle target speed without tuning the sample time, and tests 5–8 adjust the vehicle target speed and the sample time to reach the trajectory tracking purpose.
Table 2. Fourth-leg enter, third-leg exit MPC tracking changing parameters.

| No. of Test Cases | Test Description                        | Np | Nc | T (s) | Q   | R | Target Speed (km/h) |
|-------------------|-----------------------------------------|----|----|-------|-----|---|---------------------|
| 1                 | Scenario 2 change target speed, no change on sample time T | 60 | 30 | 0.027 | 100 | 1 | 50                  |
| 2                 | Scenario 2 change target speed, no change on sample time T | 60 | 30 | 0.027 | 100 | 1 | 45                  |
| 3                 | Scenario 2 change target speed, no change on sample time T | 60 | 30 | 0.027 | 100 | 1 | 40                  |
| 4                 | Scenario 2 change target speed, no change on sample time T | 60 | 30 | 0.027 | 100 | 1 | 35                  |

Figure 8a,b show the tracking route after changing the weight parameter Q and the weight parameter R.

The tracking controller did not perform well in the case of only changing the speed-related parameters. The tracking performance was altered from the first merging zone of the roundabout. The yellow line (target speed 50 km/h) reached the destination first, but the tracking error was significant. Even after changing the target speed, the controller did not reach the destination as planned. When the target speed decreased, the pink line test set (target speed 35 km/h) did not reach the final destination. However, after tuning the sample time T, the test sets all cleared the final goal successfully, as shown in Figure 8b. The specific tracking error on longitudinal and later directions are shown in Figure 9a,b.
Figure 8. (a) Shows the tracking difference with the reference path after changing the target speed. (b) Shows the tracking difference with reference path after changing the target speed and the sample time T tuning condition.

The tracking controller did not perform well in the case of only changing the speed-related parameters. The tracking performance was altered from the first merging zone of the roundabout. The yellow line (target speed 50 km/h) reached the destination first, but the tracking error was significant. Even after changing the target speed, the controller did not reach the destination as planned. When the target speed decreased, the pink line test set (target speed 35 km/h) did not reach the final destination. However, after tuning the sample time T, the test sets all cleared the final goal successfully, as shown in Figure 8b. The specific tracking error on longitudinal and lateral directions are shown in Figure 9a,b.
Figure 9. (a) Shows the comparison of trajectory errors after changing the target speed. (b) Compares trajectory errors after changing the target speed and the sample time $T$ tuning condition.

Figure 10a,b are the vehicle tracking heading performance and the longitudinal velocity. The yaw rate changes rapidly when passing through the curve area.
5. Conclusions and Future Works

In this work, an MPC controller is designed for path tracking purposes, and it is validated that the proposed MPC controller can fully track the given path in roundabouts. Case 1 and Case 2 used the same set of control horizons parameters; however, the case 2 tracking performance was not as advantageous as the Case 1 road condition. Prediction horizon and control horizon parameters need to be tuned according to the reference path curvature. Weight parameters Q, R did affect the tracking performance significantly when changing the target speed. Sample time T did not greatly affect the tracking performance.

In this work, we present a novel combination of MPC scheme and roundabout condition considering the effects of road constraints. Accounting for the effect of path curvature, a single-track vehicle model is derived. In the future, our research would like to enable collision avoidance and improve the prediction accuracy in the short-term.

Managing a MPC system from Matlab coding for real practice is also a future challenge for us. Simulations carried out in the MATLAB/CarSim environment validated the effectiveness and real-time ability of the proposed scheme. In the case of our study, it seems programming the algorithm in Matlab and then simulating the MPC system for verification is a method for practice. Setting up the compilation environment for C/C++ is
achieved by creating an S-function file in Matlab. Once the requirements are met, it can be combined with software such as Carsim and connected to the automation hardware controller module for real vehicle tests. If the simulation results verify the positive real-time performance of the designed control system in engineering applications, the robustness in trajectory tracking performance and the effectiveness in lateral stability can be verified as well; then, a real-world test with a real vehicle should be carried out in the near future. In this case, the calculated output is sent to the simulator, which, in practice, would be sending signals to the steering wheel and pedals. There are also predicted track points as output to the simulator. In the near future, the work should continue to contribute to applied mathematics and autonomous vehicle aspects.

**Author Contributions:** Conceptualization, H.C.; software, H.C.; formal analysis, H.C.; investigation, H.C.; writing—original draft preparation, H.C.; writing—review and editing, H.C. and M.Z.; supervision, M.Z.; project administration, M.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Ming, T.; Deng, W.; Zhang, S.; Zhu, B. MPC-Based Trajectory Tracking Control for Intelligent Vehicles. In Proceedings of the SAE 2016 World Congress & Exhibition—Society of Automotive Engineers, Detroit, MI, USA, 12–14 April 2016; SAE Technical Paper Series; SAE International: Miami, FL, USA, 2016.

2. Henson, M.A. Nonlinear model predictive control: Current status and future directions. *Comput. Chem. Eng.* **1998**, *23*, 187–202. [CrossRef]

3. Allgöwer, F.; Findeisen, R.; Nagy, Z. Nonlinear Model Predictive Control: From Theory to Application. *J. Chin. Inst. Chem. Eng.* **2004**, *35*, 299–315.

4. Wang, H.; Liu, B.; Ping, X.; An, Q. Path Tracking Control for Autonomous Vehicles Based on an Improved MPC. *IEEE Access* **2019**, *7*, 161064–161073. [CrossRef]

5. Guo, H.; Shen, C.; Zhang, H.; Chen, H.; Jia, R. Simultaneous Trajectory Planning and Tracking Using an MPC Method for Cyber-Physical Systems: A Case Study of Obstacle Avoidance for an Intelligent Vehicle. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4273–4283. [CrossRef]

6. Di Cairano, S.; Kalabic, U.; Berntorp, K. Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 709–714. [CrossRef]

7. Du, X.; Htet, K.K.K.; Tan, K.K. Development of a Genetic-Algorithm-Based Nonlinear Model Predictive Control Scheme on Velocity and Steering of Autonomous Vehicles. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6970–6977. [CrossRef]

8. Quirynen, R.; Berntorp, K.; Di Cairano, S. Embedded Optimization Algorithms for Steering in Autonomous Vehicles based on Nonlinear Model Predictive Control. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 3251–3256. [CrossRef]

9. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. A model predictive control approach for combined braking and steering in autonomous vehicles. In Proceedings of the 2007 Mediterranean Conference on Control & Automation, Athens, Greece, 27–29 June 2007; pp. 1–6. [CrossRef]

10. Berntorp, K.; Quirynen, R.; Di Cairano, S. Steering of Autonomous Vehicles Based on Friction-Adaptive Nonlinear Model-Predictive Control. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 965–970. [CrossRef]

11. Jiang, H.; Wang, Z.; Chen, Q.; Zhu, J. Obstacle avoidance of autonomous vehicles with CQP-based model predictive control. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2017; pp. 001668–001673. [CrossRef]

12. Hussain, S.A.; Jahromi, B.S.; Cetin, S. Cooperative Highway Lane Merge of Connected Vehicles Using Nonlinear Model Predictive Optimal Controller. *Vehicles* **2020**, *2*, 249–266. [CrossRef]

13. Mathworks. Model Predictive Control Toolbox. Available online: https://www.mathworks.com/help/mpc/ug/choosing-sample-time-and-horizons.html (accessed on 15 June 2021).
14. Zak, S.H. An Introduction to Model-Based Predictive Control (MPC). Available online: https://engineering.purdue.edu/~zak/ECE680/MPC_handout.pdf (accessed on 15 June 2021).
15. Wikipedia. Model Predictive Control. Available online: https://en.wikipedia.org/wiki/Model_predictive_control (accessed on 15 June 2021).
16. Németh, B.; Gáspár, P.; Hegedűs, T. Optimal Control of Overtaking Maneuver for Intelligent Vehicles. *J. Adv. Transp.* **2018**, *2018*, 1–11. [CrossRef]
17. Lin, F.; Chen, Y.; Zhao, Y.; Wang, S. Path tracking of autonomous vehicle based on adaptive model predictive control. *Int. J. Adv. Robot. Syst.* **2019**, *16*. [CrossRef]
18. Kalman, R.E. When is a Linear Control System Optimal? *J. Basic Eng.* **1964**, 86, 51–60. [CrossRef]
19. Mizushima, Y.; Okawa, I.; Nonaka, K. Model Predictive Control for Autonomous Vehicles with Speed Profile Shaping. *IFAC-PapersOnLine* **2019**, *52*, 31–36. [CrossRef]
20. Babu, M.; Oza, Y.; Singh, A.K.; Krishna, K.M.; Medasani, S. Model Predictive Control for Autonomous Driving Based on Time Scaled Collision Cone. In Proceedings of the 2018 European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018; pp. 641–648. [CrossRef]
21. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive Active Steering Control for Autonomous Vehicle Systems. *IEEE Trans. Control. Syst. Technol.* **2007**, *15*, 566–580. [CrossRef]
22. Mercorelli, P. Trajectory tracking using MPC and a velocity observer for flat actuator systems in automotive applications. In Proceedings of the 2008 IEEE International Symposium on Industrial Electronics, Cambridge, UK, 30 June–2 June 2008; pp. 1138–1143.