POLITECNICO DI TORINO

Master Degree course in Mechatronic Engineering

Master's Degree Thesis

Path tracking control solutions via Enhanced Model Reference Adaptive Control algorithms augmented with Neural Networks and their experimental validation in scaled fully autonomous vehicles

Supervisors
Prof. Alessandro VIGLIANI
Dr. Umberto MONTANARO
Prof. Aldo SORNIOTTI

Candidate
Paolo TIMIS

ACADEMIC YEAR 2022-2023

Acknowledgements

A coloro che hanno portato serenità e motivazione in questi anni.

Abstract

Autonomous driving represents a pivotal and extensively researched technology, characterized by features that are progressively finding their way into commercial vehicles. This technology relies on a combination of sensors, actuators, and sophisticated software, seeking to improve the safety and reliability of vehicles, aiming at the replacement of human drivers.
This work targets the path tracking problem i.e. the automatic steering of the vehicle in order to follow a specific path with no human action; this task is undertaken in both simulated and real experimental environments. The employed control algorithm is the Enhanced Model Reference Adaptive Control (EMRAC), it is an adaptive control design method that allows the controlled variables of a plant to track a given reference model. In this project, two types of EMRAC control systems were developed to guide a real, scaled, and fully autonomous vehicle prototype along a predetermined path: a standard EMRAC and an EMRAC enhanced with a Neural Network.
The controllers have been designed for a wide range of velocities showing notable improvements with respect to benchmark controllers as well as the benefits of the Neural Network based augmentation. This research project contributes to reducing the gap in experimentally validated Enhanced Model Reference Adaptive Control algorithms present in the existing literature.

Contents

List of Figures

List of Tables

Chapter 1

Introduction & State of the Art

1.1 Introduction

"Likewise, cars may be made so that without a draught animal they may be moved cum impetu inaestimabili, as we deem the scythed chariots to have been from which antiquity fought. And flying machines are possible, so that a man may sit in the middle turning some device by which artificial wings may beat the air in the manner of a flying bird."
- Roger Bacon, XIII

The sentence above, attributed to the medieval English philosopher Roger Bacon [1], discusses an enduring human fascination with vehicles that stands since medieval ages. Whether vehicles are appealing solely for their practical utility or possess an inherent value for humanity in granting the ability to move, unfortunately, falls beyond the scope of this project.

Nevertheless, it is clear that the need of having sophistication in vehicles existed before the vehicle itself. This project in particular contributes to one of the main technology interesting vehicles, automation; according to experts by 2030 the share of autonomous driving in overall traffic may rise to as much as 40%, mileage-wise [2].
This work will deal with the path tracking problem i.e. the automatic steering of the vehicle in order to follow a specific path with no human action. The control algorithm that will be researched is the Enhanced Model Reference Adaptive Control that will be further augmented with Neural Networks. The control algorithm is the set of instructions/strategy that the controller employs in order to actuate the steering angle; in control engineering control algorithms are designed to operate within specified constraints and provide control control actions in response to inputs [3]

1.2 Sate of the art

In order understand the relevance and contributions of this project it is important evaluate the state of the art of both the Model Reference adaptive Control and its improvements through Artificial Intelligence Methods.

1.2.1 MRAC and EMRAC in Automotive and Robitcs Appliactions

The baseline of this work is given by recent research conducted by Dr. Umberto Montanaro in [4] and [5] in the field of Enhanced Model Reference Adaptive Control applied to automotive as well as robotics.

In [4] an EMRAC algorithm is used to design a generic lateral tracking for a vehicle, employing a σ-modification approach to bind the adaptive gain of the switching action. The controller is evaluated in a co-simulation environment based on IPG Carmaker/MATLAB, highlighting its path tracking performance in the presence of external disturbances, road surface changes, modelling errors and parameter mismatches. The simulations shown that the adaptive algorithm successfully tracked the path even in presence of large external disturbance, modelling errors, etc. In [5] this algorithm is extended to multi-input systems covering the gap with respect to previous algorithm limited to single input systems. Several EMRAC solutions are designed for the problem of trajectory tracking for space space robotic arms in the presence of unknown and noncooperative targets. The simulation analysis confirmed that the closed-loop tracking performance of the EMRAC are superior with respect to benchmark controllers.

The research mentioned above is the results of an extensive analysis developed during previous researches such as in [6], [7], and [8]. In [7] the MRAC algorithm with Minimal Control Synthesis (MCS) is considered; MRAC with MCS is an effective control algorithms that guarantees asymptotic convergence of the tracking error to zero not only for disturbance-free uncertain linear systems but also for highly nonlinear perturbed plants that show unmodeled behaviour. In this work the drift in adaptive gains is addressed trough the parameter projection algorithm. After providing the proof of stability of all the closed-loop signals the authors validate numerically the algorithm through a discrete time LTI system subjected to parameter variations and disturbances; finally, the adaptive strategy is employed on the control of a highly nonlinear electromechanical actuator is considered. Similarly, in [6] the continous time case is considered taking into account also and integral action that embeds the σ-modification; in this case the control strategy is validate on continuos systems subjected to different types of disturbances. To show the deployment to engineering problems an electrical power circuit is considered.
Finally, in [8] the development of a hyperstable, discrete-time MCS algorithm with a formal proof of asymptotic stability for generic n-dimensional plants is shown. The authors implemented and analyzed the performance on controlling an electronic throttle body in automotive engineering.

1.2.2 AI-based augmented MRAC

Since one of the objective of this work is to improve the Enhanced Model Reference Adaptive Control through the use of Artificial Intelligence, it is important to evaluate the main finding in this field.

In [9] the authors show Model Reference Controller (MRC) for robot arm trajectory

tracking using two Neural Networks (NN). The first neural network is used for the reference model, trained in such a way that it follows any desired reference trajectory; the second neural network instead is used as a controller, trained to provide the desired torque in order to minimize the error between the outputs of the actual plant and the reference model until it reaches approximately zero. Through simulations the authors validate the proposed method assuring that the NN-based MRC is capable of following the desired trajectory with approximately zero tracking error.
Another interesting research is conducted in [10] where a supervised neural dynamic programming (SNDP) approach is developed yo solve the MRAC problem for unknown nonlinear discrete-time systems. The SNDP operates with 2 modes learning and control: in the learning mode a database-based adaptive critic learning algorithm is employed to make sure that the controlled system can adaptively follow the reference model behavior; afterwards, the algorithm smoothly transitions to the control mode where the robustness of the closed-loop control systems is further enhanced. Through simulations the authors show that the developed SNDP approach ensures the adaptation to the variable reference input while guaranteeing better performance and robustness with respect to traditional MRAC methods.

In [11] a Deep Neural Network based Model Reference Adaptive Control (DMRAC) is presented. In this work the authors utilize deep neural networks (DNN) as the adaptive element and propose an algorithm for the online update of the weights of DNN utilizing a dual time adaptation scheme; demonstrating through simulations the improvements with respect to the traditional MRAC. Referring to Figure 1.1 the DNN is divided in the faster learning outer adaptive network and slower deep feature network: between successive updates of the inner layer weights, the feature provided by the inner layers of the deep network is used as the fixed feature vector for outer layer adaptive network update and evaluation. A simpler version of this architecture is proposed in [12].

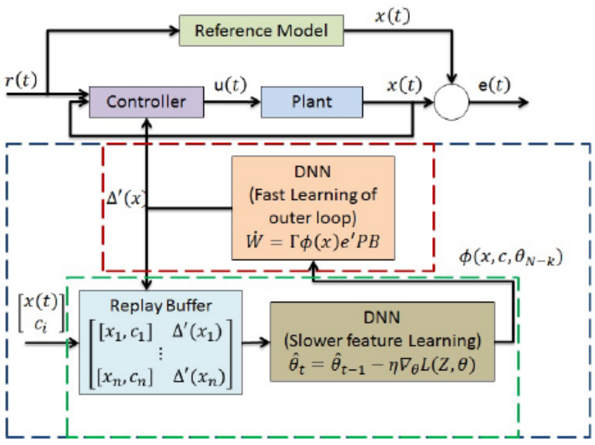

Reference Model
$x(t)$
$r(t)$
Controller
u(t)
Plant
$x(t)$
e(t)
$\Delta'(x)$
DNN
(Fast Learning of outer loop)
$\dot{W} = \Gamma\phi(x)e'PB$
$\phi(x, c, \theta_{N-k})$
$\begin{bmatrix} x(t) \\ c_i \end{bmatrix}$
Replay Buffer
$\begin{bmatrix} [x_1, c_1] & \Delta'(x_1) \\ \vdots & \\ [x_n, c_n] & \Delta'(x_n) \end{bmatrix}$
DNN
(Slower feature Learning)
$\hat{\theta}_t = \hat{\theta}_{t-1} - \eta\nabla_\theta L(Z, \theta)$

Figure 1.1: DNN architecture

1.3 Considerations

After a careful analysis of the state of the art proposed in the previous section it emerges that the Model Reference Adaptive Control is not extensively researched and has a lot of potentialities and future developments. In particular, it is clear that there is a lack related with the experimental validation of this control algorithm; with the majority of the research proving the validation only through simulations.
These premises allow to understand the novelty and relevance of this work in providing experimental validation and an AI-based augmentation to a sophisticated and innovative control algorithm.

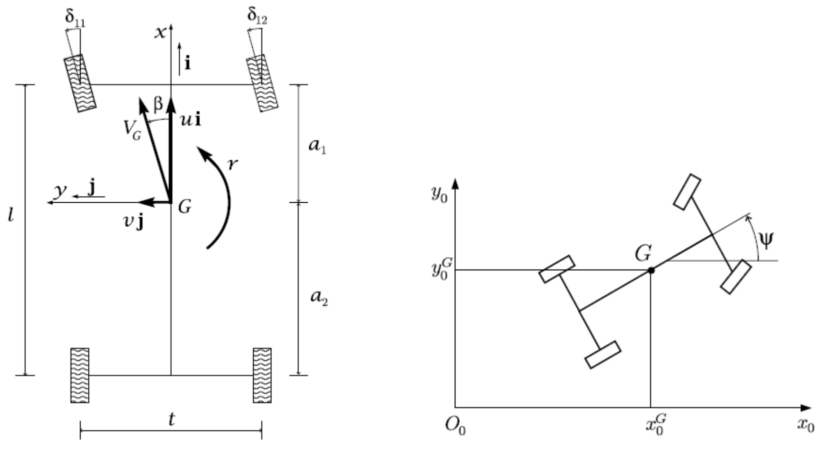
Chapter 2

Vehicle Dynamics

In order to proficiently develop path tracking solutions it is crucial to grasp key concepts related with vehicle dynamics. This chapter will discuss this topics highlighting the aspects that will be relevant for controller design.

2.1 Vehicle Kinematics

In order to analyze the kinematics of a vehicle it is particularly important to consider a body-fixed reference system as shown in Figure 2.1a; characterized as (x,y,z;G). The center of mass of the vehicle is denoted as G and it is the center of the fixed reference system; the x-axis indicates the forward direction, the y-axis the lateral direction and finally the z-axis perpendicular to the road and positive when pointing upward. The corresponding unit vectors are **(i,j,k)**



(a) Vehicle scheme (b) Yaw angle

Figure 2.1: Kinematics of a vehicle in planar motion

2.1.1 Velocities

Assuming a planar trajectory, the motion of the vehicle body can be described by its angular speed Ω and the velocity V_p of any point, such as the center of mass G. Referring to Figure 2.1a velocities V_p and Ω can be written as:

$$V_G = u\mathbf{i} + v\mathbf{j} \tag{2.1.1a}$$

$$\Omega = r\mathbf{k} = \dot{\psi}\mathbf{k} \tag{2.1.1b}$$

$$V_P = V_G + \ \Omega \times GP \tag{2.1.1c}$$

Where $r = \dot{\psi}$ is the yaw rate, the derivative of the yaw angle ψ shown in Figure 2.1b; the above equations allow to describe completely the kinematics of the vehicle body through $u(t)$, $v(t)$ and $\dot{\psi}(t)$

2.1.2 Yaw angle and trajectory

Considering the scheme shown in Figure 2.1b and defining a grownd-fixed reference system $S_0 = (x_0, y_0, z_0; O_0)$ having unit vectors (i_0, j_0, k_0), it is possible to write:

$$V_G = \dot{x}_0\mathbf{i}_0 + \dot{y}_0\mathbf{j}_0 = u\mathbf{i} + v\mathbf{j} \tag{2.1.2}$$

In particular, writing $\dot{x}_0$ and $\dot{y}_0$ in function of the forward velocity v and lateral velocity u:

$$\dot{x}_0 = u\cos(\psi) - v\sin(\psi) \tag{2.1.3a}$$

$$\dot{y}_0 = u\cos(\psi) + v\sin(\psi) \tag{2.1.3b}$$

with $\mathbf{i}_0 \cdot \ \mathbf{i} = \cos(\psi) \quad and \ \mathbf{j}_0 \cdot \ \mathbf{i} = -\sin(\psi)$.
Integrating Equation 2.1.3 it is possible to obtain the functions $x_0^G(t)$ and $y_0^G(t)$ which describe the trajectory of the center of mass G in the coordinates of the fixed reference frame S_0.

$$x_0^G\left(\hat{t}\right) = x_0^G(0) + \int_0^{\hat{t}} \dot{x}_0 dt = x_0^G(0) + \int_0^{\hat{t}} \left[u(t)\cos\psi(t) - v(t)\sin\psi(t)\right] dt \tag{2.1.4a}$$

$$y_0^G\left(\hat{t}\right) = y_0^G(0) + \int_0^{\hat{t}} \dot{y}_0 dt = y_0^G(0) + \int_0^{\hat{t}} \left[u(t)\sin\psi(t) + v(t)\cos\psi(t)\right] dt \tag{2.1.4b}$$

With the yaw angle ψ at a time instant $\hat{t}$ defined as:

$$\psi\left(\hat{t}\right) = \psi(0) + \int_0^{\hat{t}} r(t)\, dt \tag{2.1.5}$$

Planar Kinematics of a Rigid Body

(Cap. 5 Guiggiani) Rigid bodies are not subjected to deformations due to applied forces, therefore the velocity of any couple of points A and B are linked as in Equation 2.1.6; with the angular speed Ω being the angular velocity of the vehicle, the same far all of its points.

$$V_B = V_A + \Omega \times AB = V_A + V_{BA} \tag{2.1.6}$$

Considering the particular case of a planar motion, it is known that at any time instant there exists a point C of the extended rigid body named instantaneous center of velocity that has null velocity. Since the velocity field of a rigid behaves as a pure rotation around C [13], it is possible to write the velocities of any point P as in Equation 2.1.7; the scheme is shown in Figure 2.2 for two points A and B.

$$V_P = \Omega \times CP = r\mathbf{k} \times CP \tag{2.1.7}$$

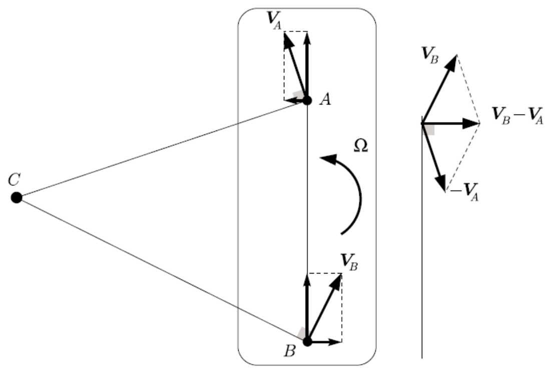


Figure 2.2: Instantaneous center of velocity and velocity of two points

Referring to the scheme in Figure 2.1a it is possible to define the ratio $\frac{v}{u}$ which is related with the vehicle slip angle β

$$\beta = \arctan\left(\frac{v}{u}\right) \tag{2.1.8}$$

Acceleration and Radius of Curvature

Also in analysing the acceleration it is important to distinguish between angular acceleration $\dot{\Omega}$ and absolute acceleration a_G of the center of mass; expressed as follows:

$$\dot{\Omega} = \dot{r}\mathbf{k} = \ddot{\psi}\mathbf{k} \tag{2.1.9a}$$

$$\mathbf{a}_G = \frac{dV_G}{dt} = \dot{u}\mathbf{i} + ur\mathbf{j} + \dot{v}\mathbf{j} - vr\mathbf{i} = a_x\mathbf{i} + a_y\mathbf{j} \tag{2.1.9b}$$

considering that, since the reference system is integral with the body, $\frac{d\mathbf{i}}{dt} = r\mathbf{j}$ and $\frac{d\mathbf{j}}{dt} = -r\mathbf{i}$. The expression of a_G shown in Equation 2.1.9b can be rewritten in terms of components tangent or normal with respect to the trajectory followed by the vehicle as shown in Equation 2.1.11. The unit vectors tangent and normal with respect to the trajectory, $\mathbf{t}$ and $\mathbf{n}$ respectively, are expressed as in Equation 2.1.10.

$$\mathbf{t} = \frac{V_G}{|V_G|} = \cos(\beta)\,\mathbf{i} + \sin(\beta)\,\mathbf{j} \tag{2.1.10a}$$

$$\mathbf{n} = \mathbf{k} \times \mathbf{t} = -\sin(\beta)\,\mathbf{i} + \cos(\beta)\,\mathbf{j} \tag{2.1.10b}$$

$$a_G = a_t\mathbf{t} + a_n\mathbf{n} \tag{2.1.11a}$$

$$a_t = a_G \cdot \mathbf{t} = a_x\cos(\beta) + a_y\sin(\beta) = \frac{\dot{u}u + \dot{v}v}{\sqrt{u^2+v^2}} \tag{2.1.11b}$$

$$a_n = a_G \cdot \mathbf{n} = -a_x\sin(\beta) + a_y\cos(\beta) = \frac{r\left(u^2+v^2\right) + \dot{v}u - \dot{u}v}{\sqrt{u^2+v^2}} \tag{2.1.11c}$$

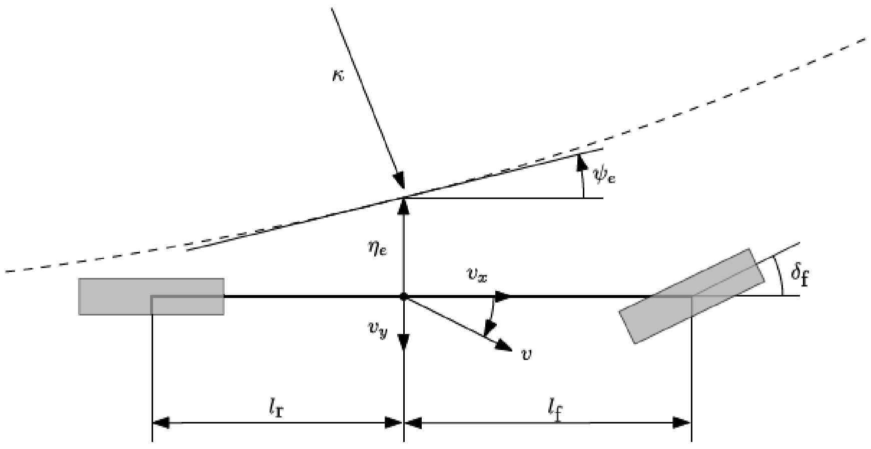
Writing the acceleration of the center of mass with respect to the trajectory is particulary important since it allows to define the radius of curvature R_G and the curvature κ, that will be particularly important in the application considered in this thesis.

$$R_G = \frac{V_G^2}{a_n} = \frac{\left(u^2+v^2\right)^{\frac{3}{2}}}{r\left(u^2+v^2\right) + \dot{v}u - \dot{u}v} = \frac{V_G}{r + \frac{\dot{v}u - \dot{u}v}{V_G^2}} \tag{2.1.12a}$$

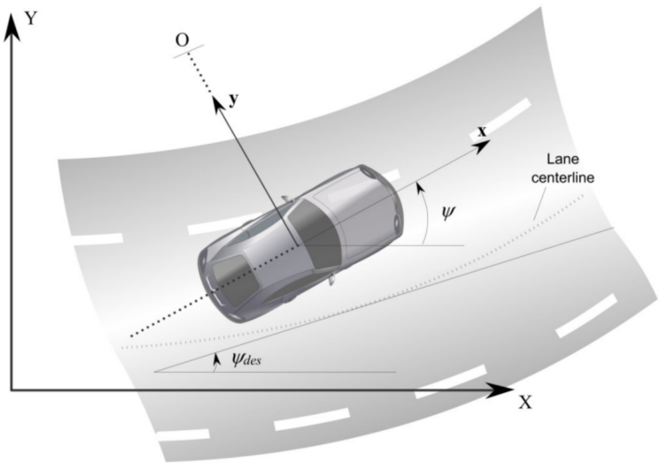
$$\kappa = \frac{1}{R_G} \tag{2.1.12b}$$

2.2 Bicycle model

The scheme shown in Figure 2.1a represents the double track model characterized by a four-wheel dynamical model of the vehicle. Although the double track model is more accurate, a simplified model named single track model (often named bicycle model) is more popular [14]. It assumes that the left and right gear ratio of the steering system are almost equal; the scheme of this model is shown in Figure 2.3a [15]



(a) Single track model



(b) Lateral vehicle dynamics

Figure 2.3: Turning Vehicle

2.2.1 Lateral dynamics

In Figure 2.3b the vehicle is considered to have two degrees of freedom:

- the lateral position y, measured from the center of rotation of the vehicle
- the yaw angle ψ measured with respect to the global axis X

applying Newton's second law along the y-axis, the moment balance about the z-axis, and considering Equation 2.2.2 it follows that:

$$ma_y = m\left(\ddot{y} + v_x\dot{\psi}\right) = F_{yf} + F_{yr} \tag{2.2.1a}$$

$$I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \tag{2.2.1b}$$

$$a_y = \left(\frac{d^2y}{dt^2}\right)_{inertial} \tag{2.2.2a}$$

$$a_y = \ddot{y} + v_x\dot{\psi} \tag{2.2.2b}$$

In particular F_{yf} and F_{yr} are the lateral tyre forces that act on the vehicle and expressed through Equation 2.2.3. The constants $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness of each front and rear tyre respectively; θ_{Vf} and θ_{Vr} are the front and rear tyre velocity angle shown in Figure 2.4 and expressed through Equation 2.2.4; if the small angle approximation i.e. $V_y = \dot{y}$ is considered, it is possible to approximate the tyre angles as in Equation 2.2.5:

$$F_{yf} = 2C_{\alpha f}\left(\delta - \theta_{vf}\right) \tag{2.2.3a}$$

$$F_{yr} = 2C_{\alpha r}\left(-\theta_{vr}\right) \tag{2.2.3b}$$

$$\tan\left(\theta_{vf}\right) = \frac{v_y + l_f\dot{\psi}}{v_x} \tag{2.2.4a}$$

$$\tan\left(\theta_{vr}\right) = \frac{v_y - l_r\dot{\psi}}{v_x} \tag{2.2.4b}$$

$$\theta_{vf} = \frac{\dot{y} + l_f\dot{\psi}}{v_x} \tag{2.2.5a}$$

$$\theta_{vr} = \frac{\dot{y} - l_r\dot{\psi}}{v_x} \tag{2.2.5b}$$

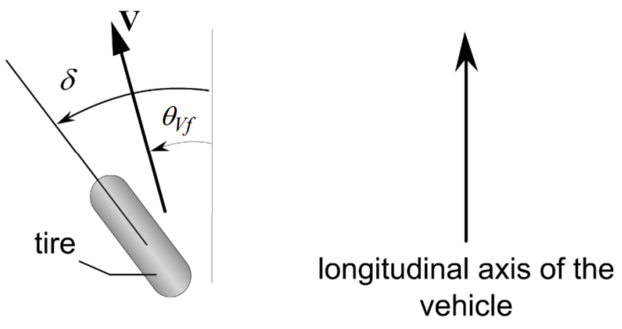


Figure 2.4: Tyre velocity angle

Substituting in Equation 2.2.1 the lateral tyre forces and the tyre velocity angles introduced in Equation 2.2.3 and Equation 2.2.4 respectively, the state space model of the lateral vehicle dynamics can be written as follows:

$$\frac{d}{dt}\begin{Bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{C_{\alpha f}+C_{\alpha r}}{mv_x} & 0 & -v_x - \frac{C_{\alpha f}l_f - C_{\alpha r}l_r}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{l_fC_{\alpha f}-l_rC_{\alpha r}}{I_zv_x} & 0 & -\frac{l_f^2C_{\alpha f}+l_r^2C_{\alpha r}}{I_zv_x} \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{Bmatrix} 0 \\ \frac{C_{\alpha f}}{m} \\ 0 \\ \frac{l_fC_{\alpha f}}{I_z} \end{Bmatrix}\delta \tag{2.2.6}$$

Recalling that the main target of the thesis is to develop a control algorithm aimed at path tracking it is essential to define the dynamic model in Equation 2.2.6 in terms of the position and orientation error with respect to the path. The key variables to take into account the errors with respect to the path are:

- e_1 the distance of the center of gravity of the vehicle from the center line of the road
- e_2 the orientation error of the vehicle with respect to the road

Recalling Figure 2.3a it is clear that if the vehicle travels with a constant longitudinal velocity v_x on a road with constant radius of curvature R, it is possible to define the rate of change of the desired orientation of the vehicle as:

$$\dot{\psi}_{des} = \frac{v_x}{R} \tag{2.2.7}$$

Clearly, Equation 3.1.6 is valid if the radius R is large enough to consider the small angle assumptions considered previously. Defining $\ddot{e}_1$ and e_2 as:

$$\ddot{e}_1 = \left(\ddot{y} + v_x\dot{\psi}\right) - \frac{v_x^2}{R^2} = \ddot{y} + v_x\left(\dot{\psi} - \dot{\psi}_{des}\right) \tag{2.2.8a}$$

$$e_2 = \psi - \psi_{des} \tag{2.2.8b}$$

Integrating Equation 2.2.8a:

$$\dot{e}_1 = \dot{y} + \int v_x e_2$$

if v_x is constant it follows that

$$\dot{e}_1 = \dot{y} + v_x\left(\psi - \psi_{des}\right) \tag{2.2.9}$$

Consequently the lateral dynamic state space model shown in Equation 2.2.6 can be rewritten in function of the errors with respect to the path, obtaining the systems shown in [4]:

$$\dot{x} = Ax + B_1\delta + B_2\kappa \tag{2.2.10}$$

$$A = \begin{bmatrix} -\frac{C_{\alpha f}+C_{\alpha r}}{mv_x} & -v_x - \frac{C_{\alpha f}l_f - C_{\alpha r}l_r}{mv_x} & 0 & 0 \\ -\frac{l_f C_{\alpha f} - l_r C_{\alpha r}}{I_z v_x} & -\frac{l_f^2 C_{\alpha f} + l_r^2 C_{\alpha r}}{I_z v_x} & 0 & 0 \\ 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.2.11a}$$

$$B_1 = \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{C_\alpha l_f}{I_z} \\ 0 \\ 0 \end{bmatrix} \tag{2.2.11b}$$

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -v_x \end{bmatrix} \tag{2.2.11c}$$

The system shown in Equation 2.2.10 describes the lateral dynamics of the bicycle model in terms of the errors with respect to the road. In particular, the state vector is $x = \begin{bmatrix} \dot{y} & \dot{\psi} & e_1 & e_2 \end{bmatrix}$; κ is the curvature and δ is the steering angle

Chapter 3

Path tracking and Experimental Setup

3.1 Path tracking

When dealing with autonomous driving vehicles, path tracking control is a common approach to realize automatic steering of the vehicle. The reference trajectory can be computed by a motion planning algorithm in order to avoid obstacles detected by environmental perception (through lidars, radars etc.) [16]. In this work several trajectories are generated and the performance of the control algorithm is evaluated taking into consideration Key Performance Indicators (KPIs).

3.1.1 KPIs definition

The definition of KPIs allow to evaluate the performance of the control algorithm as well as the improvements with respect to benchmark alternatives. For this reason it is crucial to take into consideration indicators that are related with the path tracking as well as control action.

Lateral Error

The first important aspect when considering path tracking is the deviation with respect to the path tracking is the deviation from the center-line. In particular, the maximum lateral displacement and the root mean square error is taken into account.

$$\begin{cases} e_{1_{\max}} = \max\left(e_1\right) \\ RMSE\left(e_1\right) = \dfrac{\sqrt{\sum_{i=1}^{N} e_1^2}}{N} \end{cases} \tag{3.1.1}$$

Heading angle

The same KPIs considered for the lateral error are computed also for the heading angle, constituting the other two KPIs.

$$\begin{cases} e_{2_{\max}} = \max\left(e_2\right) \\ RMSE\left(e_2\right) = \dfrac{\sqrt{\sum\limits_{i=1}^{N} e_2^2}}{N} \end{cases} \tag{3.1.2}$$

Control action

In order to evaluate the quality of the control action two KPIs were defined. The IACA shown in Equation 3.1.3 is the positive area under the control action [17], it is considered in order to account for the intensity of the control action, a good control action guarantees good tracking performance as well as a low IACA.
The Oscillation KPI expressed in Equation 3.1.4 evaluates the noise and smoothness of the control action. A high value of this KPI means that the steering angle changes with high frequency and/or with high amplitude. The formulation is the same as the one for the IACA but in this case the approximate derivatives are considered [18].

$$IACA_{\delta} = \sum_{t=t_i}^{t_f} \frac{\left|\delta_{t-1} + \delta_t\right|}{2} \Delta t \tag{3.1.3}$$

$$Oscillation_{KPI} = \sum_{t=t_i}^{t_f} \frac{\left|\dot{\delta}_{t-1} + \dot{\delta}_t\right|}{2} \Delta t \tag{3.1.4}$$

3.1.2 Trajectories Generation

As mentioned previously the trajectories can be generated in different ways: they can be computed by a motion planning algorithm or alternatively can be generated offline and than fed to the controller without taking into consideration potential obstacles. The reason why in this work only the latter alternative is considered is because the aim of the thesis is to design and validate a particular control algorithm; feeding directly the trajectory to the control action allows to focus only on the performance of the EMRAC algorithm rather than the motion planner.
Particular emphasis is put on closed trajectories, this is due to the fact that the control algorithm shown in this work is an adaptive one. Initially, the control algorithm requires some time to learn the system before steering the dynamics towards the reference one; for this reason, closed trajectories allow to repeat the tracking multiple times, showing the improvements of the EMRAC. The employed trajectories are: eight trajectory, obstacle avoidance, and circular trajectory

Methodology

The methodology employed to generate the reference trajectory to be tracked can be summarized in the following steps:

- definition of the trajectory in terms of the coordinates x and y contained in vectors
- parametrization of the vectors x and y in terms of the vehicle distance [19], [20]
- computation of the first and second derivatives [18]
- computation of the curvature taking into account concepts in the study of curves in differential geometry and calculus, as shown in Equation 3.1.5

$$\kappa = \frac{(dx/ds)\,(ddy/ds^2) - (ddx/ds^2)\,(dx/ds)}{(ddx/ds^2 + ddy/ds^2)^{3/2}} \tag{3.1.5}$$

Trajectories

As previously mention the trajectories considered in this work are the eight trajectory, circular trajectory and obstacle avoidance.
The eight trajectory shown in Figure 3.1 is defined considering an initial straight section and then two circular adjacent trajectories repeated three times. This trajectory will be widely employed in the experimental validation of the EMRAC algorithm for two reasons: at each lap it is possible to evaluate the change in the KPIs, highlighting the capabilities of the control algorithm; with respect to the circular trajectory, the reference curvature changes continuously and therefore is more demanding.

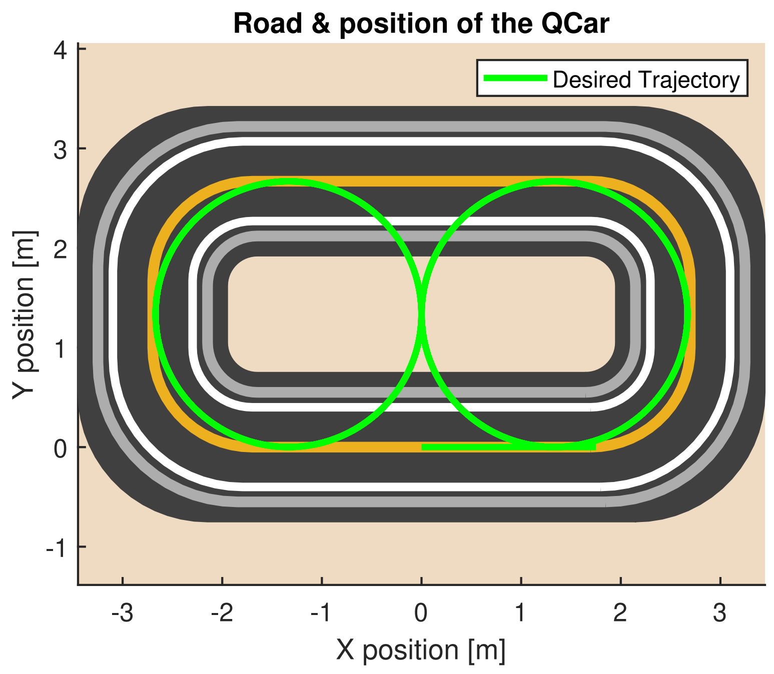


Figure 3.1: Eight trajectory

The obstacle avoidance trajectory, shown in Figure 3.2, is realized through the first two laps of the eight trajectory while the last lap is substituted with a semicircular trajectory and an obstacle avoidance; this is done in order to properly evaluate the performance of the adaptive algorithm, allowing the adaption in the first two laps.

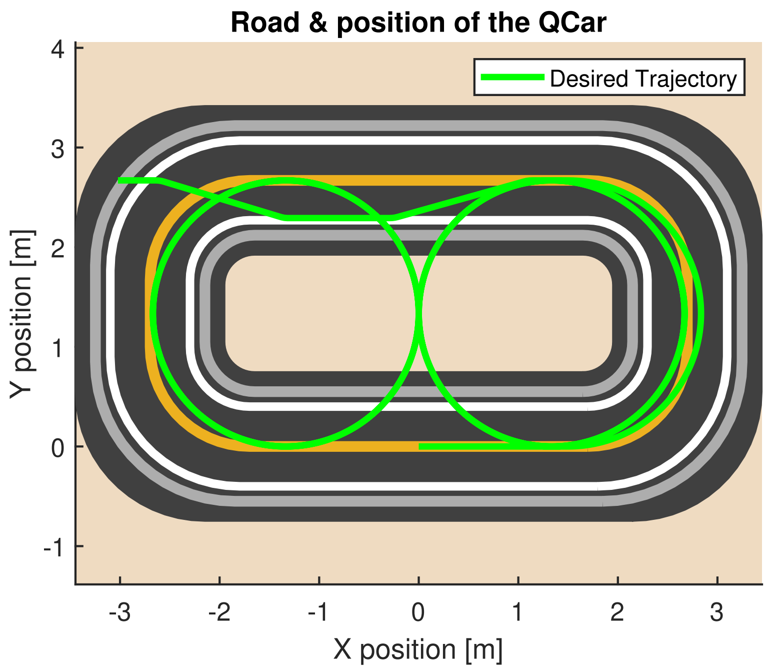


Figure 3.2: Abstacle avoidance trajectory

Finally, the circular trajectory in Figure 3.3 is composed of an initial straight segment and a circular trajectory repeated three times.

3.1.3 Errors and travelled distance computation

The trajectory generation considered in the previous section provides the reference path that the vehicle is supposed to follow; as said the reference path is provided in terms of the distance travelled by the vehicle as shown in Figure 3.4 where look-up tables are used to provide x_{ref}, y_{ref}, and κ. Regarding the reference yaw angle ψ_{ref} it can be either stored in a look-up table or computed during the simulations integrating Equation 3.1.6 recalled below.

$$\dot{\psi}_{des} = \frac{v_x}{R}$$

The mentioned look-up tables for different trajectories are shown in Figure 3.5, Figure 3.6, and Figure 3.7.

With those considerations it becomes important to define a method to compute the distance travelled by the vehicle. The most straight-forward strategy is to use to compute

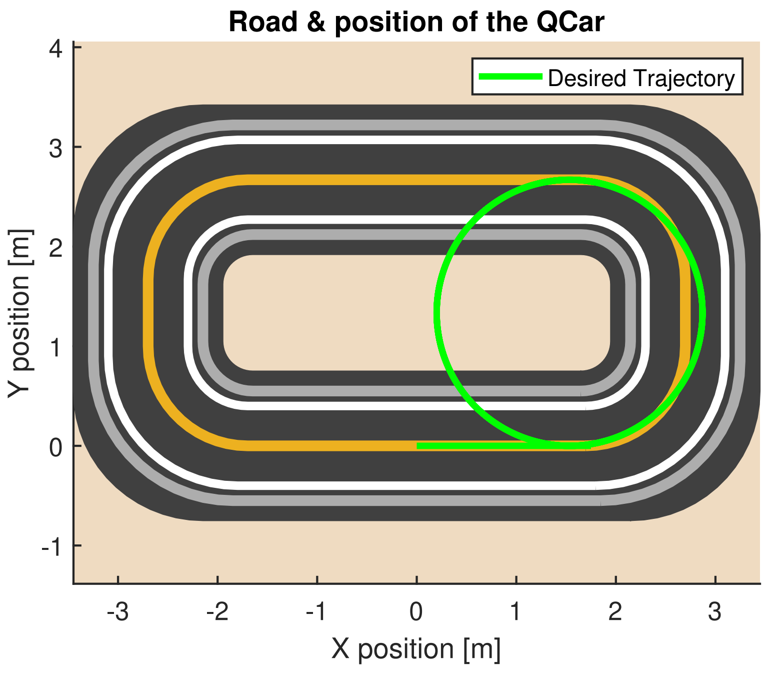


Figure 3.3: Circular Trajectory

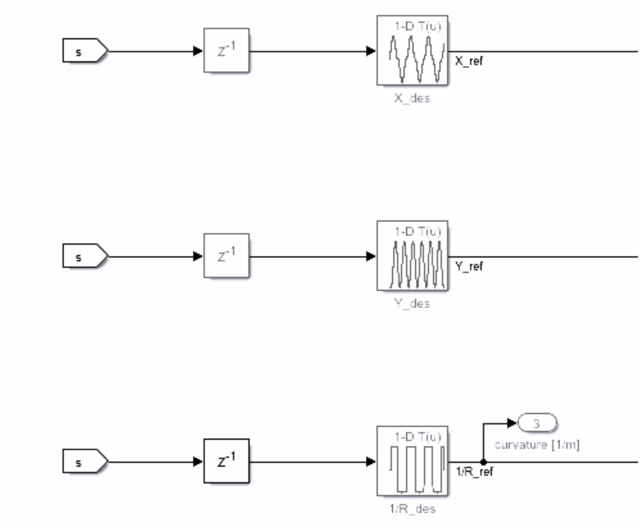


Figure 3.4: Look-up tables

the velocity of the vehicle as in Equation 3.1.6 and than integrate. This procedure, although being correct can lead to complications during the experimental phase of the project; this is due to the fact that this method relies on the vehicle lateral velocity which is not directly measurable and needs to be estimated.

$$\dot{s} = \sqrt{v_x^2 + v_y^2} \tag{3.1.6}$$

Another strategy for computing the travelled distance is to consider the sum of distances travelled between two consequent time instants. Lets consider two time instants t and $t-1$, the distance ds_t travelled by the vehicle in this period is shown in Equation 3.1.7b

$$ds_t^* = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \tag{3.1.7a}$$

$$ds_t = \frac{ds_t^*}{1 - \frac{e_1}{\kappa}} \tag{3.1.7b}$$

Consequently the distance travelled by the vehicle at a specific time instant t_f is

$$s_{t_f} = \sum_{t=t_0}^{t_f} ds_t \tag{3.1.8}$$

The considerations related with the vehicle lateral velocity also influence the computation of the lateral displacement error. In the previous section this error, named e_1, was subjected to the following law:

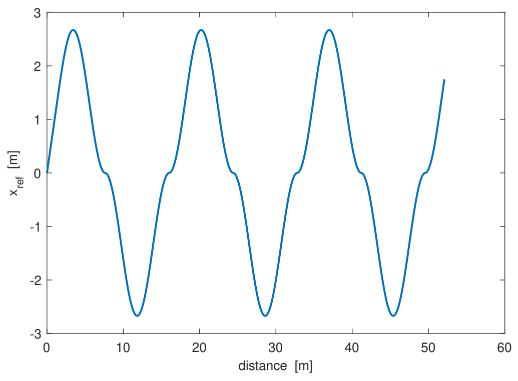
$$\dot{e}_1 = v_y + v_x\left(\psi - \psi_{des}\right)$$

Since experimentally v_y is not known the lateral displacement error will be computed as in [21]. This formula presented in Equation 3.1.9 can be applied in scenarios where the vehicle exhibits either solely a lateral mismatch or both a lateral and longitudinal mismatch.

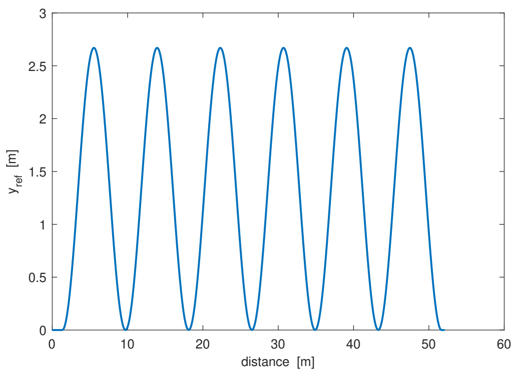
$$e_y = (y - y_{ref})\cos\left(\psi_{ref}\right) - (x - x_{ref})\sin\left(\psi_{ref}\right) \tag{3.1.9}$$

The considerations related with the heading angle error instead are still valid, and it can be simply computed as in Equation 2.2.8b, reported below.

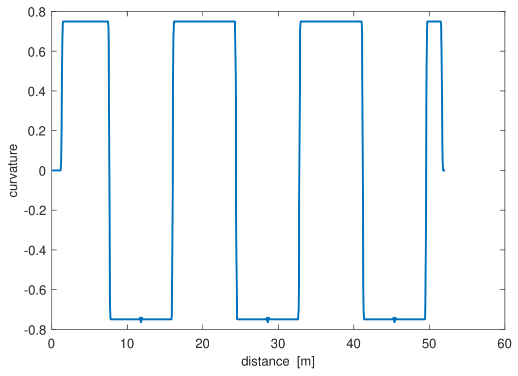
$$e_2 = \psi - \psi_{des}$$



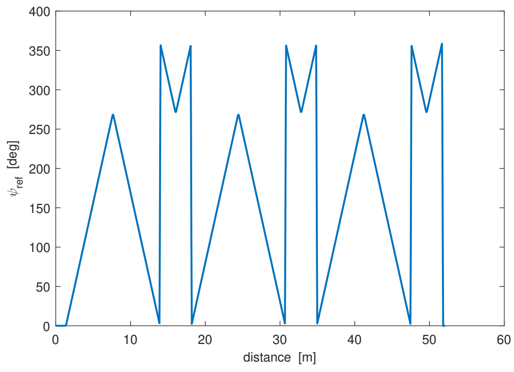
(a) eight trajectory: x_{ref}



(b) eight trajectory: y_{ref}

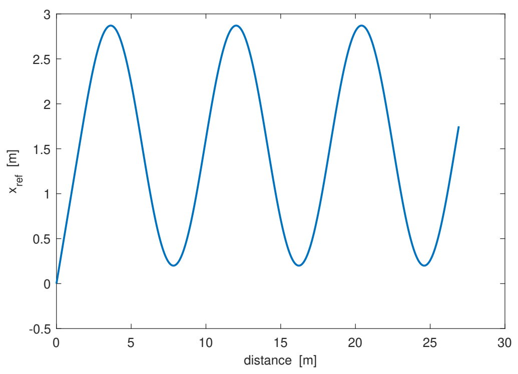


(c) eight trajectory: curvature

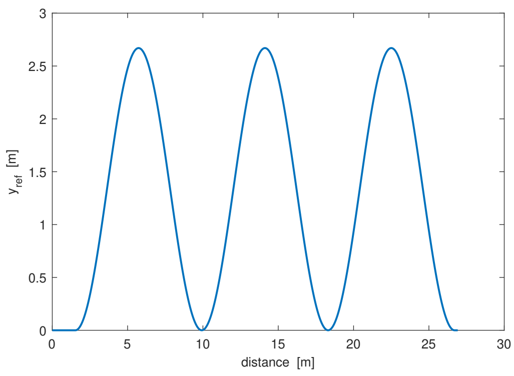


(d) eight trajectory: ψ_{ref}

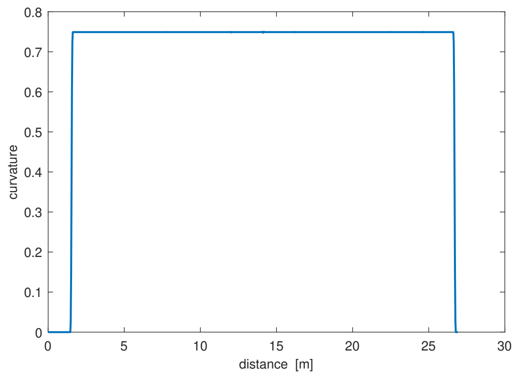
Figure 3.5: Eight trajectory: look-up tables



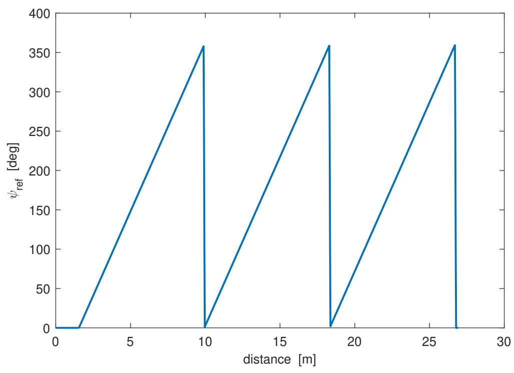
(a) Circular trajectory: x_{ref}



(b) Circular trajectory: y_{ref}

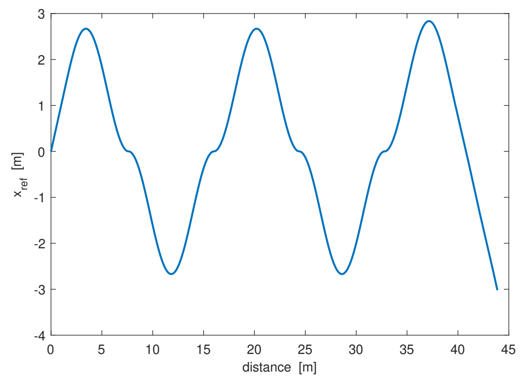


(c) Circular trajectory: curvature

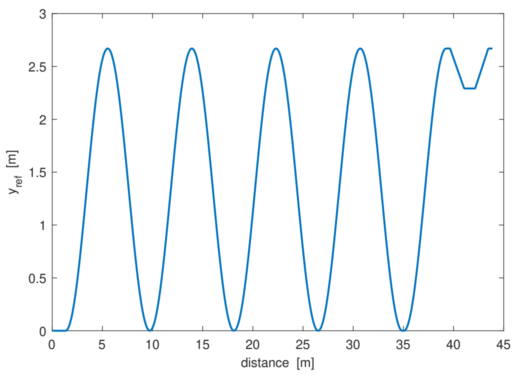


(d) Circular trajectory: ψ_{ref}

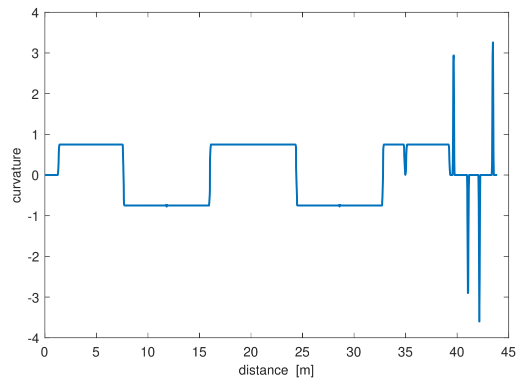
Figure 3.6: Circular trajectory: look-up tables



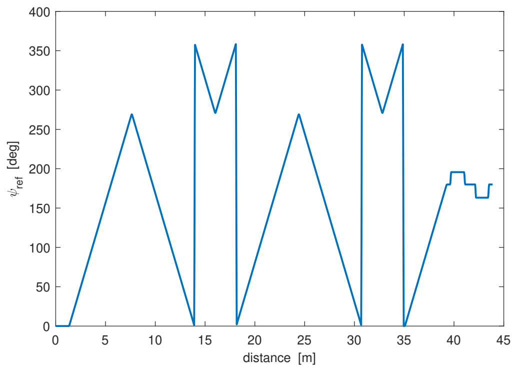
(a) Obstacle avoidance: x_{ref}



(b) Obstacle avoidance: y_{ref}



(c) Obstacle avoidance: curvature



(d) Obstacle avoidance: ψ_{ref}

Figure 3.7: Obstacle avoidance trajectory: look-up tables

3.2 QCar and Lab equipement

The vehicle used in this work, shown in Figure 6.1, is a scaled autonomous vehicle designed by the Canadian company Quanser [22]. It is equipped with a wide range of sensors such as lidar, camera for 360° vision, proximity sensor, IMU, encoder, and I/O port. The QCar, while showing great manufacturing quality, has a wide range of software features like: Simulink and Python Libraries with the possibility of creating new ROS nodes as well as a complete virtual environment. This technology allows a direct interface between the usher and the scaled vehicle via the setup shown in Figure 3.10.

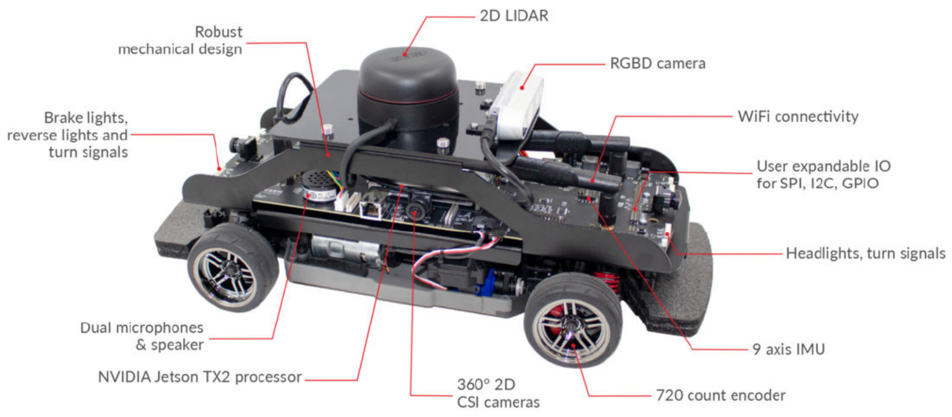

2D LIDAR
Robust mechanical design
RGBD camera
WiFi connectivity
Brake lights, reverse lights and turn signals
User expandable IO for SPI, I2C, GPIO
Headlights, turn signals
Dual microphones & speaker
9 axis IMU
NVIDIA Jetson TX2 processor
360° 2D CSI cameras
720 count encoder

Figure 3.8: QCar vehicle

The vehicle moves within the area considered in Figure 3.9. As previously said, the QCar follows the imposed trajectory given through a look-up table in terms of the distance; therefore, even if the yellow line is not followed by the vehicle the rubber carpet shown in this figure allows a good grip with the QCar's tyre.

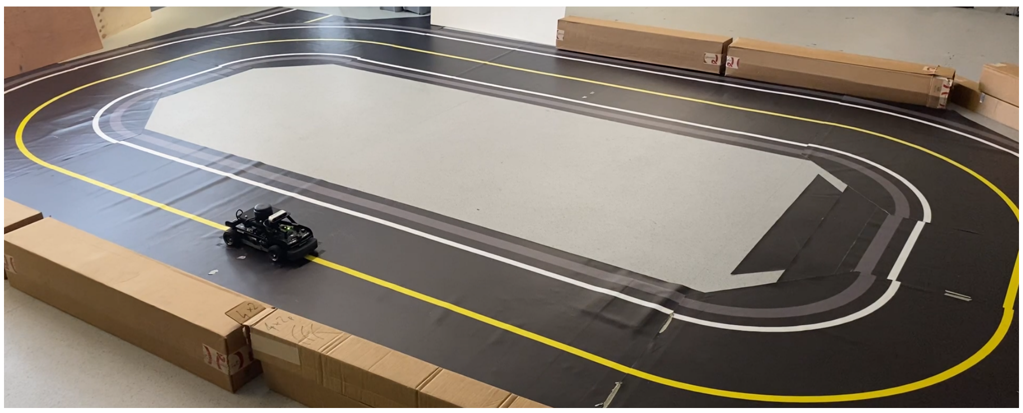


Figure 3.9: Experiments area

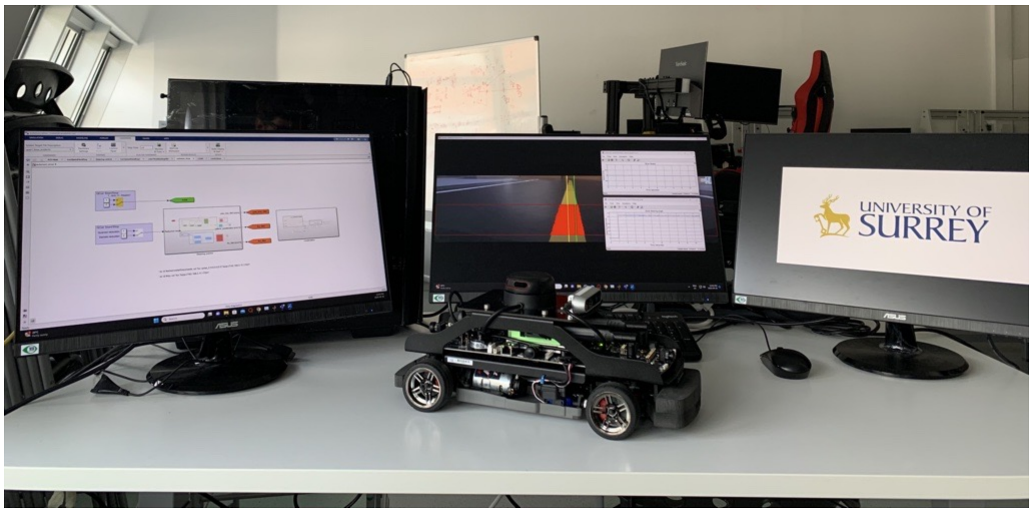
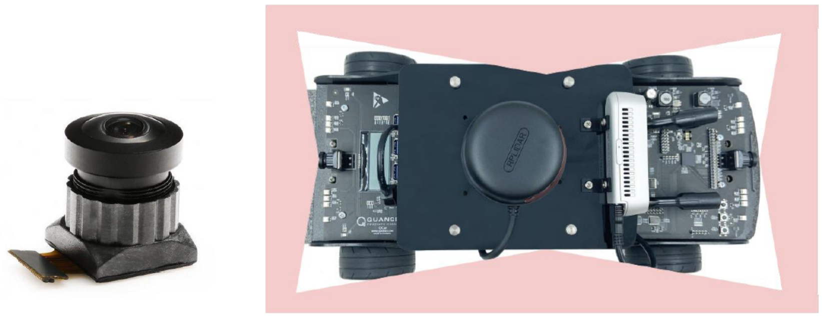


Figure 3.10: Laboratory setup

3.2.1 QCar Hardware

The complex hardware shown in Figure 6.1 is managed by the onboard computer NVIDIA Jetson TX2 with CPU: 2GHz quad-core ARM Cortex-A57 64-bit and GPU:256 CUDA core NVIDIA Pascal; as well as the following sensors and actuators [23].

To guarantee a 360° vision the QCar has 4 Camera Serial Interface (CSI); those are 2D cameras mounted on each side of the vehicle, having quadrangular lens with 160° vertical Field of View (FOV) and 120° horizontal Field of View. In Figure 3.11a one of this cameras is shown, it is important to acknowledge that those cameras are 8MP and have dimensions comparable with the ones present on a modern smartphone; in Figure 3.11b the coverage (in pink) and the blind-spots (in white) are shown.

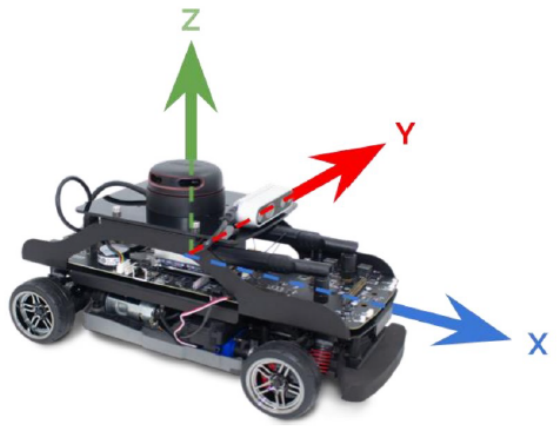


(a) Camera (b) Coverage and blind-spots

Figure 3.11: Camera Serial Interface (CSI)

The QCar is equipped with an Intel RealSense D435 RGBD shown in Figure 3.12a. RGBD stands for Red, Green, Blue, and Depth; in particular the depth image is grayscale and can detect the distance with respect to objects framed by the camera allowing the avoidance of obstacles and/or providing information related with the position.

(a) Camera RGBD



(b) Axis

Figure 3.12: QCar axis and camer

Another important hardware component is the Intertial Measurement Unit (IMU) with 9 axis. Three of them are used to measure the acceleration with respect to the axis x,y,z of the vehicle as shown in Figure 3.12b; other three are relevant for the gyroscope that measures the angular velocity around the three axis of the reference system solidal with the vehicle; finally, the last three axis are used by the magnetometer in order to measure the magnetic field along the three axis. The magnetometer is used to get an absolute angular direction. [24]

Furthermore, the QCar has a CC motor for the actuation of the four wheels and a servomotor for the steering. The CC motor chosen by Quanser is the Titan 12T 550 [24] while the servomotor has a rotor that is physically constrained to steer between 0.5rad and -0.5rad.

Figure 3.13: RPLidar A2M8

Finally, a crucial component is the Light Detection and Ranging sensor, the Lidar shown in Figure 3.13; this device emits laser, receives signals and elaborates data. The

laser emitter sends impulses towards a specific direction; this signal is reflected by the object and received by the device; elaborating the time needed by the signal to come back to the receiver this sensor is able to find the distance of the object.
In this work the main use of thee Lidar is for the Simultaneous Localization and Mapping (SLAM), i.e. to get information related to the environment and the position of the Vehicle. Quanser developed an algorithm aimed at the localization of the QCar as follows:

- initially a scan of the environment is performed while the vehicle is stopped, saving this data on the local memory. This initial position constitutes the origin of the reference system.
- once the vehicle starts moving the Lidar will capture information that are confronted with the data obtained during the initial scan, this allows the computation of the position and the orientation.

This algorithm shows two main vulnerabilities that can affect the precision of the position and orientation: first of all, the initial scan saves information related with one specific point, requiring the need of reference objects withing the environment; secondly, since the maximum frequency is 15 Hz the control algorithm will be constrained to operate at this frequency that may be too low. In order to improve the estimation of the position a Kalman filter will be employed.

3.2.2 Vehicle State Variables Estimation

To estimate the vehicle's state variables a Kalman Filter is used, whose algorithm is composed of two steps: prediction and update. Through the plant model and prior estimate the filter predicts the state variables as well as a covariance matrix (Q) that reflects the prediction's level of confidence; during the update phase the filter uses the measurements provided by the sensor in order to correct the prediction of the previous stage. Through a weighted average the Kalman filter is able to provide a better estimation with respect tot the one provided by the model only; the mentioned weights are based on the uncertainties present in the measurements (covariance matrix R) and in the predicted state variables.
The Kalman filter that will be employed is the one developed in [25]; it is a non-linear Extended Kalman Filter that employs the single track model described in subsection 2.2.1, as well as a longitudinal model defined in terms of the dynamics of the electrical motor. The Kalman filter scheme is shown in Figure 3.14 and it estimates the variables vector $\hat{x}_a$ relying on the measurements: vehicle pose X, Y, the heading angle ψ, the velocity v_x; and the input to the system: the front steering angle δ_f and the armature voltageV_a.
The complete model's equations are shown in Equation 3.2.4, this model is derived by combining:

- equations related with the lateral dynamics model in the case of small front steering angle, shown in Equation 3.2.1

$$\dot{\beta} = -\dot{\psi} + \frac{C_{\alpha f}}{mv}\left(\delta - \beta - \frac{l_f\dot{\psi}}{v}\right) + \frac{C_{\alpha r}}{mv}\left(-\beta + \frac{l_r\dot{\psi}}{v}\right) \tag{3.2.1a}$$

$$\ddot{\psi} = \frac{l_f C_{\alpha f}}{I_z}\left(\delta - \beta - \frac{l_f \dot{\psi}}{v}\right) - \frac{l_r C_{\alpha r}}{I_z}\left(-\beta + \frac{l_r \dot{\psi}}{v}\right) \tag{3.2.1b}$$

- velocities expressed in the global reference frame, Equation 3.2.2

$$\dot{X} = v\cos\left(\beta + \psi\right) \tag{3.2.2a}$$

$$\dot{Y} = v\sin\left(\beta + \psi\right) \tag{3.2.2b}$$

$$\dot{\psi} = r = \frac{v\cos\left(\beta\right)}{l_f + l_r}\left(\tan\left(\delta_f\right)\right) \tag{3.2.2c}$$

- the differential equation that describes the electric motor dynamic, and consequently, considering the transmission ratio τ, the longitudinal velocity [?]

$$\dot{\omega} = P_1 V_a - P_2\omega - P \tag{3.2.3a}$$

$$v_x = \omega\tau r_\omega \tag{3.2.3b}$$

$$\dot{v}_x = \tau r_\omega\left(P_1 V_a - \frac{P_2}{\pi r_\omega} v_x - P_3\right) \tag{3.2.3c}$$

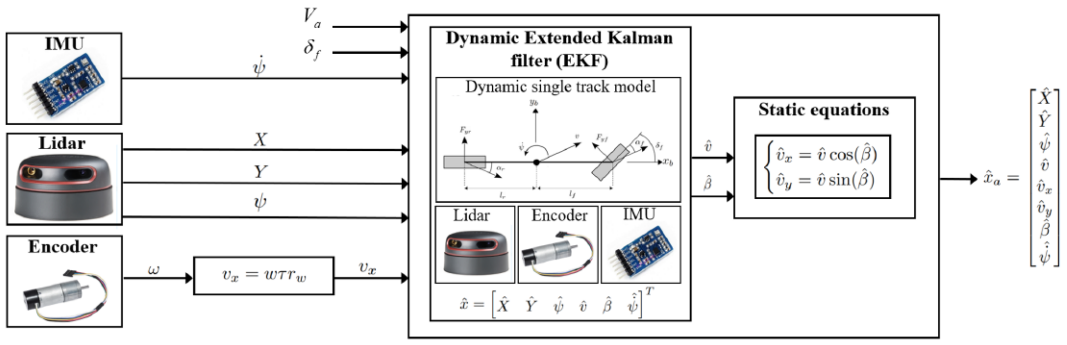

IMU
Lidar
Encoder
V_a
δ_f
$\dot{\psi}$
X
Y
ψ
ω
$v_x = w\tau r_w$
v_x
Dynamic Extended Kalman filter (EKF)
Dynamic single track model
Lidar
Encoder
IMU
$\hat{x} = \left[\hat{X} \ \hat{Y} \ \hat{\psi} \ \hat{v} \ \hat{\beta} \ \hat{\dot{\psi}}\right]^T$
$\hat{v}$
$\hat{\beta}$
Static equations
$\hat{v}_x = \hat{v}\cos(\hat{\beta})$
$\hat{v}_y = \hat{v}\sin(\hat{\beta})$
$\hat{x}_a = \left[\hat{X} \ \hat{Y} \ \hat{\psi} \ \hat{v} \ \hat{v}_x \ \hat{v}_y \ \hat{\beta} \ \hat{\dot{\psi}}\right]^T$

Figure 3.14: Dynamic Extended Kalman Filter

$$\begin{cases} X\left(k+1\right) = X\left(k\right) + \left(v\left(k\right)\cos\left(\beta\left(k\right) + \psi\left(k\right)\right)\right) T_s \\ Y\left(k+1\right) = Y\left(k\right) + \left(v\left(k\right)\sin\left(\beta\left(k\right) + \psi\left(k\right)\right)\right) T \\ \psi\left(k+1\right) = \psi\left(k\right) + \dot{\psi}\left(k\right) T \\ v\left(k+1\right) = v\left(k\right) + \left(\tau r_\omega\left(P_1 V_a\left(k\right) - \frac{P_2}{\tau r_\omega} v\left(k\right) - P_3\right)\right) T_s \\ \beta\left(k+1\right) = \beta\left(k\right) + \left(-\dot{\psi}\left(k\right) + \frac{C_{\alpha f}}{m v(k)}\left(\delta\left(k\right) - \beta\left(k\right) - \frac{l_f \dot{\psi}(k)}{v(k)}\right) + \frac{C_{\alpha r}}{m v(k)}\left(-\beta\left(k\right) + \frac{l_r \dot{\psi}(k)}{v(k)}\right)\right) T_s \\ \dot{\psi}\left(k+1\right) = \dot{\psi}\left(k\right) + \left(\frac{l_f C_{\alpha f}}{I_z}\left(\delta\left(k\right) - \beta\left(k\right) - \frac{l_f \dot{\psi}(k)}{v(k)}\right) - \frac{l_r C_{\alpha r}}{I_z}\left(-\beta\left(k\right) + \frac{l_r \dot{\psi}(k)}{v(k)}\right)\right) T_s \end{cases} \tag{3.2.4}$$

Chapter 4

Enhanced Model Reference Adaptive Control (EMRAC)

In this chapter the Enhanced Model Reference Adaptive control will be discussed starting from the peculiarity of the Model Reference Adaptive Control (MRAC) and deepening the characteristics of its enhanced version considered in [5].

4.1 Model Reference Adaptive Control (MRAC)

The Model Reference Adaptive Control is an adaptive control design method that allows the controlled variable of a plant to track a given reference model. Even though this method imposes a required reference dynamics with a limited knowledge of the plant parameters, it is sensitive to external disturbances that may lead to a drift of the adaptive gains and consequently a reduced tracking performance [4].

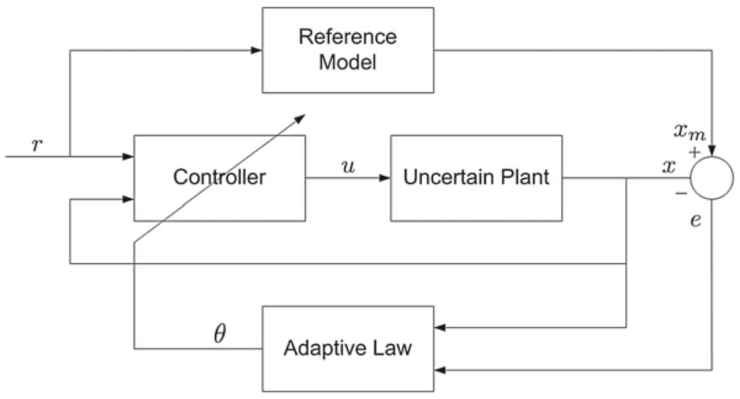


Figure 4.1: Model Reference Adaptive Control Scheme

In Figure 4.1 the composition of the MRAC is shown, in particular [12]:

- The plant can contain structured uncertainty (i.e. parametric variation in the plant dynamics) or unstructured uncertainty (i.e. frequency dependant uncertainty)

- The reference model is used in order to specify a desired behaviour and is typically formulated as an LTI system

- The controller is designed to provide overall system performance and stability for a nominal plant

- The adaptive law specifies the mathematical relationship that relates the adaptive parameters with the tracking error

4.2 EMRAC Algorithm

The implemented EMRAC augments the MRAC algorithm through the presence of an adaptive integral and adaptive switching control actions; in particular, in this case a single input version of the one shown in [5] is considered; the scheme of this control system is shown in Figure 4.2

$$u(t) = u_{\mathrm{MRAC}}(t) + u_D(t) + u_I(t) + u_N(t) \tag{4.2.1}$$

$$u_{\mathrm{MRAC}}(t) = K_X(t)x(t) + K_R(t)r(t) \tag{4.2.1a}$$

$$u_D(t) = K_D(t)d(t) \tag{4.2.1b}$$

$$u_I(t) = K_I(t)x_I(t) \tag{4.2.1c}$$

$$\dot{x}_I = x_e - \sigma_I\left(\|x_I\|\right)\rho_e x_I \quad \textit{and} \quad x_e = x_m - x \tag{4.2.1d}$$

The adaptive integral control action $u_I(t)$ aims at improving the tracking of the reference model with respect to unmodoeled biases in the plant, the σ-modification strategy used is shown in 4.2.3; the adaptive switching control action $u_N(t)$ increases the robustness of the closed-loop tracking performance with respect to varying bounded disturbances.

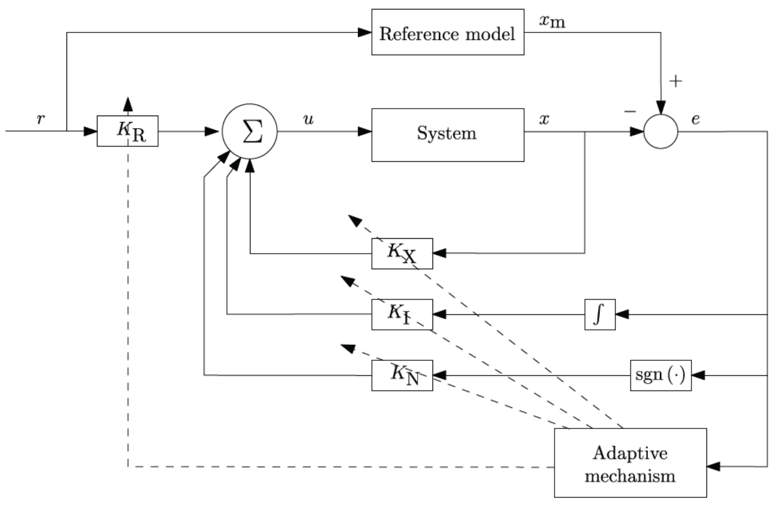


Figure 4.2: EMRAC control scheme

4.2.1 Reference System & Plant

Taking into consideration the control objective of the EMRAC algorithm, it is essential to define a reference system that describes the behaviour expected from the original system. In this section it is shown the original formulation present in [5] that considers the reference system as linear and time invariant (LTI), Equation 4.2.2; in further sections the reference system will be function of a dynamic parameter of the plant.

$$\dot{x}_m = A_m x_m + B_m r + E_m d \tag{4.2.2}$$

being:

- $x_m \in \mathbb{R}^{n_x}$ the reference model state, with n_x the dimensions of the state space
- $r \in \mathbb{R}^{n_u}$ the reference input assumed bounded, with n_u the dimension of the control input
- $d \in \mathbb{R}^{n_d}$ the measurable disturbance and n_d the dimension of its space
- $A_m \in \mathbb{R}^{n_x \times n_x}$, $B_m \in \mathbb{R}^{n_x \times n_u}$, $E_m \in \mathbb{R}^{n_x \times n_d}$ are the dynamics matrix, the input matrix and the disturbance matrix of the reference model, with A_m a Hurwitz matrix.

The plant instead has the form shown in Equation 4.2.3

$$\dot{x} = Ax + Bu + Ed + G, \quad x(t_0) \in \mathbb{R}^{n_x} \tag{4.2.3}$$

with:

- $x \in \mathbb{R}^{n_x}$ the state vector of the plant and $t_0 \in \mathbb{R}$ the initial time istant
- $u \in \mathbb{R}^{n_u}$ is the plant input vector as defined in Equation 5.2.1

- $G \in \mathbb{R}^{n_x}$ is the nonmeasurable disturbance while $E \in \mathbb{R}^{n_x \times n_d}$ is the disturbance matrix of the plant

Both the the measurable and the non measurable disturbance are considered to be bounded, meaning that there exists $G_\infty > 0$ and $d_\infty > 0$ such that

$$\|G\| \leq G_\infty, \;\; \|d\| \leq d_\infty \;\; \forall t \geq t_0 \tag{4.2.4}$$

The reference system is derived from the plant assuming that there exist ideal gains $\widehat{\Phi}_R \in \mathbb{R}^{n_u \times n_u}$, $\widehat{\Phi}_X \in \mathbb{R}^{n_u \times n_x}$, $\widehat{\Phi}_D \in \mathbb{R}^{n_u \times n_d}$ and $S \in \mathbb{R}^{n_u \times n_u}$ that satisfy:

$$B_m = B\widehat{\Phi}_R \tag{4.2.5a}$$

$$A_m = A + B\widehat{\Phi}_X = A + B_m\widehat{\Phi}_R^{-1}\widehat{\Phi}_X \tag{4.2.5b}$$

$$E_m = E + B\widehat{\Phi}_D = A + B_m\widehat{\Phi}_R^{-1}\widehat{\Phi}_D \tag{4.2.5c}$$

$$P_\phi = \widehat{\Phi}_R S = S^T\widehat{\Phi}_R^T > 0 \tag{4.2.5d}$$

furthermore, the ideal gains $\widehat{\Phi}_X$, $\widehat{\Phi}_R$, $\widehat{\Phi}_D$ can be collected in the matrix $\widehat{\Phi} \in \mathbb{R}^{n_u \times n_w}$ with $n_w = 2n_x + n_u + n_d$:

$$\widehat{\Phi} = \begin{bmatrix}\widehat{\Phi}_X & \widehat{\Phi}_R & \widehat{\Phi}_D & \widehat{\Phi}_I\end{bmatrix} = \begin{bmatrix}\widehat{\phi}_1 & \widehat{\phi}_2 & \cdots & \widehat{\phi}_{n_w-1} & \widehat{\phi}_{n_w}\end{bmatrix} \tag{4.2.6a}$$

$$\widehat{\Phi}_I = O_{n_u,n_x} \tag{4.2.6b}$$

$$\widehat{\phi} = \begin{bmatrix}\phi_1^T & \phi_2^T & \cdots & \phi_{n_w-1}^T & \phi_{n_w}^T\end{bmatrix}^T, \quad \textit{and} \quad \left\|\widehat{\phi}\right\| \leq M_\phi \tag{4.2.6c}$$

4.2.2 Adaptive gains computation

The adaptive gains present in Equation 5.2.1 are computed as follows:

$$K_X = \Phi_X + S^T y_e x^T \beta_X \;\; \textit{and} \;\; \dot{\Phi}_X = S^T y_e x^T \alpha_X + F_X \tag{4.2.7a}$$

$$K_R = \Phi_R + S^T y_e r^T \beta_R \;\; \textit{and} \;\; \dot{\Phi}_R = S^T y_e r^T \alpha_R + F_R \tag{4.2.7b}$$

$$K_D = \Phi_D + S^T y_e d^T \beta_D \;\; \textit{and} \;\; \dot{\Phi}_D = S^T y_e d^T \alpha_D + F_D \tag{4.2.7c}$$

$$K_I = \Phi_I + S^T y_e x_I^T \beta_I \;\; \textit{and} \;\; \dot{\Phi}_I = S^T y_e x_I^T \alpha_I + F_I \tag{4.2.7d}$$

$F_X, F_I \in \mathbb{R}^{n_u \times n_x}, F_R \in \mathbb{R}^{n_u \times n_u}, F_D \mathbb{R}^{n_u \times n_d}$ are the locking strategies that prevent an unbounded evolution of the gains due to disturbances or unmodeled behaviour :

$$F_X = -\sigma_\phi(\|\phi\|)\Phi_X \rho_X \tag{4.2.8a}$$

$$F_I = -\sigma_\phi(\|\phi\|)\Phi_I \rho_I \tag{4.2.8b}$$

$$F_R = -\sigma_\phi(\|\phi\|)\Phi_R \rho_R \tag{4.2.8c}$$

$$F_D = -\sigma_\phi(\|\phi\|)\Phi_D \rho_D \tag{4.2.8d}$$

Collecting the integral parts of the adaptive gains present in Equation 4.2.7 into the matrix $\Phi \in \mathbb{R}^{n_u \times n_w}$ it is possible to define the vector ϕ employed in Equation 4.2.8

$$\Phi = [\Phi_X \ \Phi_R \ \Phi_D \ \Phi_I] = [\phi_1 \ \phi_2 \ \cdots \ \phi_{n_w - 1} \ \phi_{n_w}] \tag{4.2.9a}$$

$$\phi = \left[\phi_1^T \ \phi_2^T \ \cdots \ \phi_{n_w-1}^T \ \phi_{n_w}^T\right]^T \tag{4.2.9b}$$

Additionally the following strictly positive diagonal matrices will have to be tuned:

$$\alpha_X, \beta_X, \beta_I \in \mathbb{R}^{n_x \times n_x}, \alpha_R, \beta_R \in \mathbb{R}^{n_u \times n_u}, \alpha_D, \beta_D \in \mathbb{R}^{n_d \times n_d}$$

Finally $y_e \in \mathbb{R}^{n_u}$ is computed as:

$$y_e = B^T P_e x_e \quad with \quad P_e A_m + A_m^T = -Q \tag{4.2.10}$$

4.2.3 σ-modification

The sigma modification strategy is employed to guarantee the ultimate boundess of the closed-loop tracking error dynamics also in the presence of unmatched disturbances and unmodeled dynamics, in this case in particular the σ-modification is used.
It is used to prevent the drift of the integral tracking error in Equation 4.2.1d, in fact:

$$\sigma_I(\|x_I\|) = \begin{cases} 0 & \text{if } \|x_I\| \le \widehat{M}_I \\ \eta_I \left(\frac{\|x_I\|}{\widehat{M}_I} - 1\right) & \text{if } \widehat{M}_I \le \|x_I\| \le 2\widehat{M}_I \\ \eta_I & \text{if } \|x_I\| \ge 2\widehat{M}_I \end{cases} \tag{4.2.11}$$

where η_I and $\widehat{M}_I$ are strictly positive constants.
In Equation 4.2.8 the σ-mnodification strategy $\sigma_\phi(\|\phi\|)$ for the adaptive gains of the smooth control action is computed as:

$$\sigma_\phi(\|\phi\|) = \begin{cases} 0 & \text{if } \|\phi\| \leq \widehat{M}_\phi \\ \eta_\phi \left(\frac{\|\phi\|}{\widehat{M}_\phi} - 1 \right) & \text{if } \widehat{M}_\phi \leq \|\phi\| \leq 2\widehat{M}_\phi \\ \eta_\phi & \text{if } \|\phi\| \geq 2\widehat{M}_\phi \end{cases} \tag{4.2.12}$$

in particular, the constants $\widehat{M}_\phi$ and η_ϕ must satisfy the following conditions

$$\widehat{M}_\phi \geq \sqrt{\frac{\lambda_{\max}(\Gamma_\rho \Gamma_a^{-1} \otimes P_\phi^{-1})}{\lambda_{\min}(\Gamma_\rho \Gamma_a^{-1} \otimes P_\phi^{-1})}} M_\phi, \quad \textit{and} \quad \eta_\phi \lambda_{\min} \left(\Gamma_\rho \Gamma_a^{-1} \otimes P_\phi^{-1} \right) > \frac{3}{4} \lambda_{\min}(Q) \tag{4.2.13}$$

with $\otimes$ being the Kronecker product and the strictly positive matrices Γ_ρ, $\Gamma_a \in \mathbb{R}^{n_w \times n_w}$ defined considering the constant parameters defined in Equation 4.2.7 and Equation 4.2.8:

$$\Gamma_a = \Delta(\alpha_X, \alpha_R, \alpha_D, \alpha_I) = diag(\alpha_1, \alpha_2, ..., \alpha_{n_w}) \tag{4.2.14a}$$

$$\Gamma_\rho = \Delta(\rho_X, \rho_R, \rho_D, \rho_I) = diag(\rho_1, \rho_2, ..., \rho_{n_w}) \tag{4.2.14b}$$

Adaptive switching control action

The adaptive switching control action $u_N(t)$ can be set using two different formulations $u_N^{(uv)}(t)$ and $u_N^{(ew)}(t)$:

$$u_N^{(uv)}(t) = K_N^{(uv)}(t) \frac{y_e}{\|y_e\|}, \quad K_N^{(uv)} = S^T \Phi_{N0} \tag{4.2.15a}$$

$$\dot{\Phi}_{N0} = \alpha_{N0} h_0 \left(\|y_e\|_\Omega \right) - \sigma_{N0} \left(\|\Phi_{N0}\| \right) \rho_{N0} \Phi_{N0} \tag{4.2.15b}$$

$$u_N^{(ew)}(t) = K_N^{(ew)}(t) \psi(y_e), \qquad K_N^{(ew)} = S^T \Phi_N \tag{4.2.16a}$$

$$\psi(y_e) = [\text{sgn}(y_{e_1}) \ \text{sgn}(y_{e_2}) \ \cdots \text{sgn}(y_{e_{nu}})]^T \tag{4.2.16b}$$

$$\dot{\Phi}_{Nj} = \alpha_{Nj} h_j \left(|y_e| \right) - \sigma_{Nj} \left(\|\Phi_{Nj}\| \right) \rho_{Nj} \Phi_{Nj}, \ \ j = 1, \ldots, n_u \tag{4.2.16c}$$

being $\Phi_{N0} \in \mathbb{R}$, $\Phi_N = diag \left(\Phi_{N1}, \Phi_{N2}, \ldots, \Phi_{N_{nu}} \right) \in \mathbb{R}^{n_u}$ and the σ-modification defined as following:

$$\sigma_{Nj}\left(\|\Phi_{Nj}\|\right) = \begin{cases} 0 & \text{if } \|\Phi_{Nj}\| \leq \widehat{M}_{Nj} \\ \eta_{Nj}\left(\frac{\|\Phi_{Nj}\|}{\widehat{M}_{Nj}} - 1\right) & \text{if } \widehat{M}_{Nj} \leq \|\Phi_{Nj}\| \leq 2\widehat{M}_{Nj} \\ \eta_{Nj} & \text{if } \|\Phi_{Nj}\| \geq 2\widehat{M}_{Nj} \end{cases} \tag{4.2.17}$$

where $\|y_e\|_\Omega$ with $\Omega \in \mathbb{R}^{n_u \times n_u}$ is a strictly positive matrix and similarly $\alpha_{Nj},\ \rho_{Nj},\ \eta_{Nj}, \overset{\frown}{M}_{Nj}, j = 1,\ \ldots,\ n_u$ strictly positive constants; in particular $\overset{\frown}{M}_{Nj}$ and η_{Nj} must satisfy:

$$\overset{\frown}{M}_{N0} > \frac{\delta_\infty}{\lambda_{\min}\left(SP^{-1}S^T\right)},\ \ \textit{and}\ \ \overset{\frown}{M}_{Nj} > \frac{\delta_{j\infty}}{\tilde{c}_j},\ \ j = 1,\ \ldots,\ n_u \tag{4.2.18}$$

Finally, h-functions are defined as following

$$h_0 = \left(\|y_e\|_\Omega\right) = \frac{\|y_e\|_\Omega^{\varsigma_0}}{\xi_0 + \gamma_0\,\|y_e\|_\Omega^{\varsigma_0}} \tag{4.2.19a}$$

$$h_j = \left(\|y_{ej}\|_\Omega\right) = \frac{|y_e|^{\varsigma_j}}{\xi_j + \gamma_j|y_e|^{\varsigma_j}},\ \ j = 1,\ \ldots, n_u \tag{4.2.19b}$$

considering $\xi_j, \varsigma_j, \gamma_j$ with $j = 0,\ \ldots, n_u$ strictly positive constants.

4.3 EMRAC with parameter projection

Recalling subsection 4.2.2,

$$K_X = \Phi_X + S^T y_e x^T \beta_X \ \ \textit{and}\ \ \dot{\Phi}_X = S^T y_e x^T \alpha_X + F_X$$

$$K_R = \Phi_R + S^T y_e r^T \beta_R \ \ \textit{and}\ \ \dot{\Phi}_R = S^T y_e r^T \alpha_R + F_R$$

$$K_D = \Phi_D + S^T y_e d^T \beta_D \ \ \textit{and}\ \ \dot{\Phi}_D = S^T y_e d^T \alpha_D + F_D$$

$$K_I = \Phi_I + S^T y_e x_I^T \beta_I \ \ \textit{and}\ \ \dot{\Phi}_I = S^T y_e x_I^T \alpha_I + F_I$$

the locking strategies $F_X, F_I \in \mathbb{R}^{n_u \times n_x}, F_R \in \mathbb{R}^{n_u \times n_u}, F_D \mathbb{R}^{n_u \times n_d}$ used to prevent an unbounded evolution of the gains due to disturbances or unmodeled behaviour, were designed through the σ-modification. An alternative locking strategy is the parameter projection, as shown in [6] and [7].

In [6] the parameter projection is used to preserve the convergence to zero of the tracking error when the disturbance is bounded and L_2; the σ-modification instead guarantees global uniform ultimate boundedness under continuous L_∞ disturbances.

$$F_{X_j} = \begin{cases} 0 \;\; if \;\; \phi_{X_j} \in \left(\phi^l_{X_j}, \phi^u_{X_j}\right), \;\; or \;\; \phi_{X_j} = \phi^l_{X_j} \;\; and \;\; h_{X_j} \geq 0 \\ or \;\; \phi_{X_j} = \phi^u_{X_j} \;\; and \;\; h_{X_j} \leq 0 \\ -h_{X_j}(t) \;\; otherwise \end{cases} \tag{4.3.2a}$$

$$F_R = \begin{cases} 0 \;\; if \;\; \phi_R \in \left(\phi^l_R, \phi^u_R\right), \;\; or \;\; \phi_R = \phi^l_R \;\; and \;\; h_R \geq 0 \\ or \;\; \phi_R = \phi^u_R \;\; and \;\; h_R \leq 0 \\ -h_R(t) \;\; otherwise \end{cases} \tag{4.3.2b}$$

$$F_{I_j} = \begin{cases} 0 \;\; if \;\; \phi_{I_j} \in \left(\phi^l_{I_j}, \phi^u_{I_j}\right), \;\; or \;\; \phi_{I_j} = \phi^l_{I_j} \;\; and \;\; h_{I_j} \geq 0 \\ or \;\; \phi_{I_j} = \phi^u_{I_j} \;\; and \;\; h_{I_j} \leq 0 \\ -h_{I_j}(t) \;\; otherwise \end{cases} \tag{4.3.2c}$$

As shown in Equation 4.3.2 the parameter projection proposes a design for the locking strategy that keeps the evolution of ϕ within Λ_ϕ in a componentwise manner. In fact:

- when a component of ϕ exits the corresponding boundary the gain adaptation is stopped and its value remains constant
- when a component of ϕ is contained within the boundary, the locking strategy is not active and the adaptation proceeds
- finally, if the component reaches the upper bound and has the tendency to decrease its value (derivative $h_j \leq 0$) the adaptation proceeds; vice-versa if the the component reaches the lower bound

$$h_X = \left[h_{X_1} \; \dots \; h_{X_{nx}}\right]^T = S^T y_e x^T \alpha_X \tag{4.3.3a}$$

$$h_R = \left[h_{R_1} \; \dots \; h_{R_{nu}}\right]^T = S^T y_e r^T \alpha_R \tag{4.3.3b}$$

$$h_I = \left[h_{I_1} \; \dots \; h_{X_{nx}}\right]^T = S^T y_e x_I^T \alpha_I \tag{4.3.3c}$$

$$\Phi_X = \left[\phi_{X_1} \;\dots\; \phi_{X_{n_x}}\right]^T \qquad (4.3.4a) \qquad\qquad F_X = \left[F_{X_1} \;\dots\; F_{X_{n_x}}\right]^T \qquad (4.3.5a)$$

$$\Phi_R = \left[\phi_{R_1} \;\dots\; \phi_{R_{n_u}}\right]^T \qquad (4.3.4b) \qquad\qquad F_R = \left[F_{R_1} \;\dots\; F_{R_{n_u}}\right]^T \qquad (4.3.5b)$$

$$\Phi_I = \left[\phi_{I_1} \;\dots\; \phi_{I_{n_x}}\right]^T \qquad (4.3.4c) \qquad\qquad F_I = \left[F_{I_1} \;\dots\; F_{I_{n_x}}\right]^T \qquad (4.3.5c)$$

In this work the Enhanced Model Reference Adaptive Control will be analyzed both with the σ-modification and the parameter projection and the best otpion will be choosen for further analysis.

4.4 EMRAC-NN

The Neural Network based augmentation considered in this work is inspired from the architecture considered in [12] and shown in Figure 4.3. The control action is therefore augmented with a contribution named u_{NN} as in Equation 4.4.1.

$$u_{EMRAC-NN} = u_{EMRAC} + u_{NN} = u_{EMRAC} + g\left(\Theta^T \Phi\left(W^T \bar{x}\right)\right) \qquad (4.4.1)$$

The network weights are updated considering the law shown in Equation 4.4.2 considering that: $V \in \mathbb{R}^m \times \mathbb{R}^n$, $W_x \in \mathbb{R}^n \times \mathbb{R}^m$, $W_0 \in \mathbb{R}^m$, $V_0 \in \mathbb{R}^n$, and $W_j \in \mathbb{R}^{n+1}$ $j = 1, \dots, m$ column vectors of W; in particular m is the number of neurons, while n is the number of states.

$$\dot{\Theta} = \Gamma_\Theta \Phi\left(W^T \bar{x}\right) e^T P B = \Gamma_\Theta \Phi\left(W^T \bar{x}\right) y_e \qquad (4.4.2a)$$

$$\dot{W} = \Gamma_W \bar{x} e^T P B V^T = \Gamma_W \bar{x} y_e V^T f'\left(W^T \bar{x}\right) \qquad (4.4.2b)$$

The matrices are build as follows in Equation 4.4.3, with f being the activation function of the neurons and f' its derivative.

$$\Theta^T = \begin{bmatrix} V_0 & V^T \end{bmatrix} \in \mathbb{R}^n \times \mathbb{R}^{m+1} \qquad (4.4.3a)$$

$$W^T = \begin{bmatrix} W_0 & W_x^T \end{bmatrix} \in \mathbb{R}^m \times \mathbb{R}^{n+1} \qquad (4.4.3b)$$

$$\Phi\left(W^T \bar{x}\right) = \begin{bmatrix} 1 & f^T\left(W^T \bar{x}\right) \end{bmatrix}^T \in \mathbb{R}^{m+1} \qquad (4.4.3c)$$

$$f\left(W^T \bar{x}\right) = \begin{bmatrix} f\left(W_1^T \bar{x}\right) & f\left(W_2^T \bar{x}\right) & \dots & f\left(W_m^T \bar{x}\right) \end{bmatrix}^T \in \mathbb{R}^m \qquad (4.4.3d)$$

The adaptation laws can be corrected in order to include a locking strategy as well as an integral contribution as shown in Equation 4.4.4.

$$\Theta^* = \Theta + \beta_\Theta \Phi\, y_e \qquad \textit{and} \qquad \dot{\Theta} = \Gamma_\Theta \Phi\left(W^T \bar{x}\right) y_e - \rho_\Theta \Theta \qquad (4.4.4a)$$

$$W^* = W + \beta_\Theta f'\left(W^T\bar{x}\right) y_e \qquad and \qquad \dot{W} = \Gamma_W \bar{x} y_e V^T f'\left(W^T\bar{x}\right) - \rho_W W \qquad (4.4.4b)$$

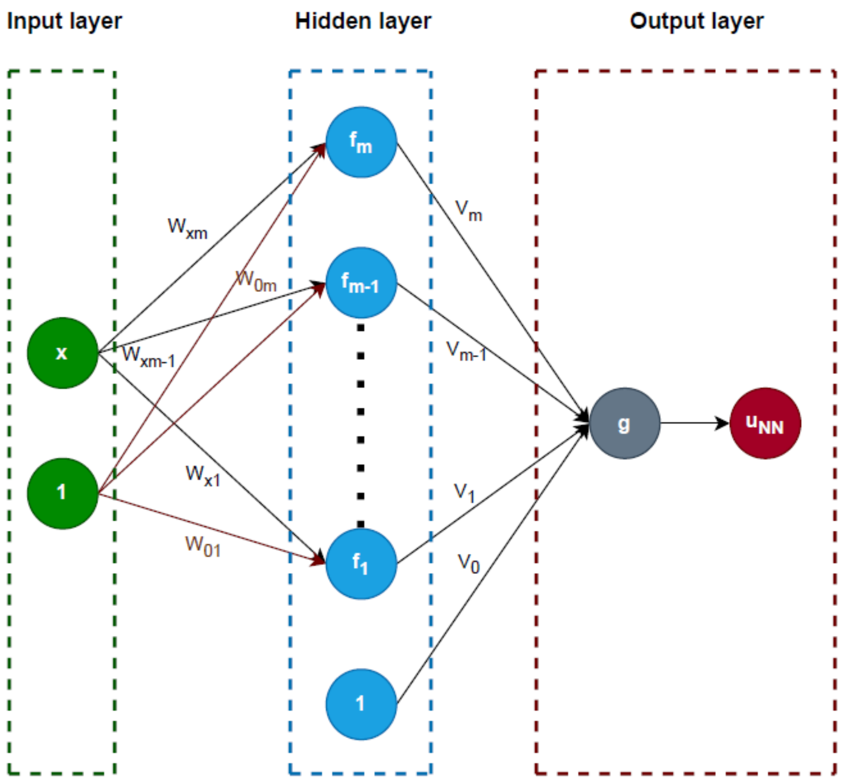


Figure 4.3: Neural Network scheme

Chapter 5

Controller Design & Simulations Results

5.1 Reference Model

The first step of the design of MRAC controller is the choice of the reference model that can be represented as in Equation 5.1.1; in order to do so, following the reasoning present in [4], a closed loop version of the system proposed in subsection 2.2.1 and recalled in Equation 5.1.2 will be considered.

$$\dot{x}_m = A_m x_m + B_m \kappa \tag{5.1.1}$$

$$\dot{x} = A x_1 + B_1 u + B_2 \kappa \tag{5.1.2}$$

With A, B_1, and B_2 being:

$$A = \begin{bmatrix} -\frac{C_{\alpha f}+C_{\alpha r}}{m v_x} & -v_x - \frac{C_{\alpha f} l_f - C_{\alpha r} l_r}{m v_x} & 0 & 0 \\ -\frac{l_f C_{\alpha f} - l_r C_{\alpha r}}{I_z v_x} & -\frac{l_f^2 C_{\alpha f} + l_r^2 C_{\alpha r}}{I_z v_x} & 0 & 0 \\ 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{C_{\alpha} l_f}{I_z} \\ 0 \\ 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -v_x \end{bmatrix}$$

Meaning	Symbol	Value
front cornering stiffness [N/deg]	$C_{\alpha f}$	11.798
rear cornering stiffness [N/deg]	$C_{\alpha r}$	8.680
mass	m	2.720
front distance [m]	l_f	0.107
rear distance [m]	l_r	0.149
inertia moment $[kg\,m^2]$	I_z	0.042

Table 5.1: Vehicle Parameters

The reference system in Equation 5.1.1 is linear time invariant (LTI) and asymptotically stable, designed assuming that there exist two constant matrices K_X^* and K_R^* such that the following equations are verified.

$$A_m = A + B_1 K_X^* \tag{5.1.3a}$$

$$B_m = B_1 K_R^* + B_2 \tag{5.1.3b}$$

It is clear that the reference system is designed through the use of nominal controllers, in particular the control law is the one shown in Equation 5.1.4; i.e. a feedback controller K_X^* and a feed forward controller K_R^* designed following the results present in [26].

$$u^*(t) = K_X^* x(t) + K_R^* \kappa(t) \tag{5.1.4}$$

The feedback controller is designed through a discrete infinite-horizon Linear Quadratic Regulator (LQR) from optimal control theory. Considering A_d and B_d the discrete-time version of A and B_1 respectively; an optimization algorithm places the eigenvalues of the closed-loop matrix $A_m = A + B_1 K_X^*$.
In particular, K_X^* is computed as in Equation 5.1.5 with the objective function to be minimized shown in Equation 5.1.6; additionally, the matrix P satisfies the matrix difference

Riccati equation in Equation 5.1.7.

$$K_X^* = \left(R^* + B_d^T P^* B_d\right)^{-1} B_d^T P^* A_d \tag{5.1.5}$$

$$J = \sum_{k=0}^{\infty} x\left(k\right) Q^* x\left(k\right) + u\left(k\right) R^* u\left(k\right) \tag{5.1.6}$$

$$P^* = A_d^T P^* A_d - A_d^T P^* B_d \left(R^* + B_b^T P^* B_d\right)^{-1} B_d^T P^* A_d + Q^* \tag{5.1.7}$$

The matrix Q^* is a diagonal weighting matrix with an entry for each state, while R^* is a weighting factor corresponding to the control effort.

$$Q^* = \begin{bmatrix} q_1^* & 0 & 0 & 0 \\ 0 & q_2^* & 0 & 0 \\ 0 & 0 & q_3^* & 0 \\ 0 & 0 & 0 & q_4^* \end{bmatrix}$$

$$R^* = 1$$

$$q_2^* = q_3^* = q_4^* = 0$$

The feed forward term is present to ensure that the steady state lateral position error will be zero. This contribution is necessary since, due to the presence of the term $B_2\kappa$, while travelling on a curve the error states will not converge to zero otherwise. The gains that will be considered are scheduled in function of the velocity and obtained in [25]; Q^* was selected through a trial and error procedure, choosing the values that guaranteed the best experimental results.

Velocity [m/s]	K_X^*	Q^*
0.5	[-0.845 -0.071 -14.142 -1.8741]	200
0.6	[-0.921 -0.081 -13.229 -2.029]	175
0.7	[-0.968 -0.089 -12.247 -2.176]	150
0.8	[-0.897 -0.097 -10.000 -2.190]	100
0.9	[-0.864 -0.103 -8.660 -2.239]	75
1.0	[-0.783 -0.110 -7.071 -2.204]	50
1.2	[-0.531 -0.116 -3.873 -1.864]	15

Table 5.2: K_X^* gain scheduling

5.2 Design

Following the theory of the Enhanced Model Reference Adaptive Control defined in the chapter 4 in this section the design of the controller is shown, starting from the standard MRAC untill the NN-augmented EMRAC.

5.2.1 MRAC

If the standard MRAC is considered, as previously said, the steering command is only influenced by the feedback and the feed forward contribution. The adaptation laws presented in Equation 4.2.7 and reported below show that the gains that need to be tuned are α_X, β_X , α_R, and β_R. The constants α_X and α_R represents the pace of the gain adaptation; the constants β_R and β_X relate with the intensity of the integral contribution.

$$K_X = \Phi_X + S^T y_e x^T \beta_X \quad \textit{and} \quad \dot{\Phi}_X = S^T y_e x^T \alpha_X + F_X$$

$$K_R = \Phi_R + S^T y_e r^T \beta_R \quad \textit{and} \quad \dot{\Phi}_R = S^T y_e r^T \alpha_R + F_R$$

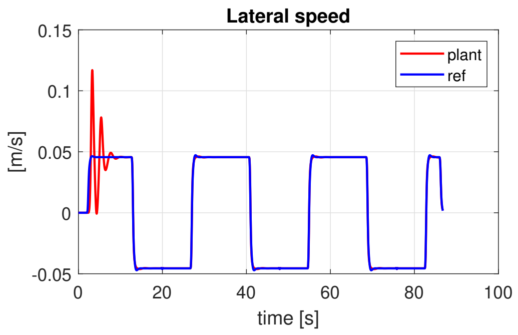
In Figure 5.1 it is shown the tracking of the errors at velocity 0.6 m/s, using the following constants:

$$\alpha_X = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad \beta_X = \begin{bmatrix} 1e-05 & 0 & 0 & 0 \\ 0 & 1e-06 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

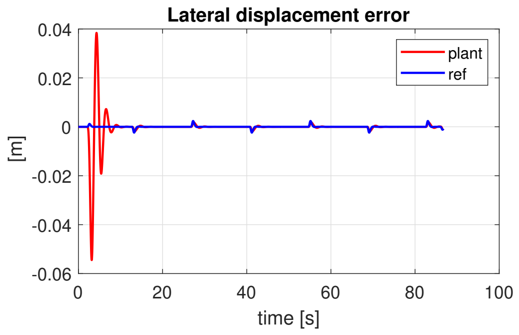
$$\alpha_X = \begin{bmatrix} 0.0001 \end{bmatrix} \quad \beta_R = \begin{bmatrix} 0.0001 \end{bmatrix}$$

It appears clear that the states that requires priority with respect to the other is the third state i.e. the lateral displacement error. Intuitively, it is reasonable to have this type of dynamic since in path tracking the lateral displacement is crucial. Figure 5.1 also highlights the adaptive nature of this type of control algorithm; initially the gains are adapted starting from zero, leading to drifts with respect to the reference system.

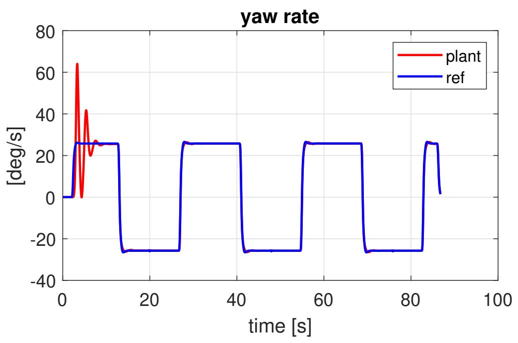
In Figure 5.2 more detailed are shown regarding the lateral displacement error adaptation, during most of the first lap the controller is not able to steer the dynamics properly Figure 5.2a; while starting from half of the first lap the tracking is optimal as shown in Figure 5.2b.



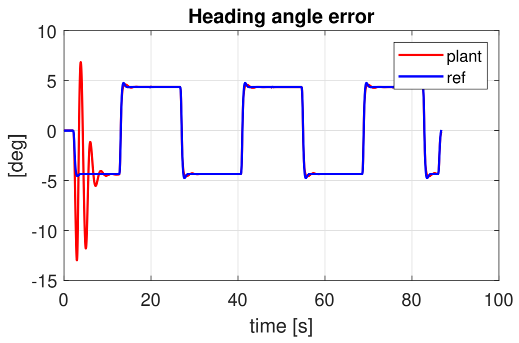
(a) Lateral speed x_1



(b) Lateral displacement error x_3

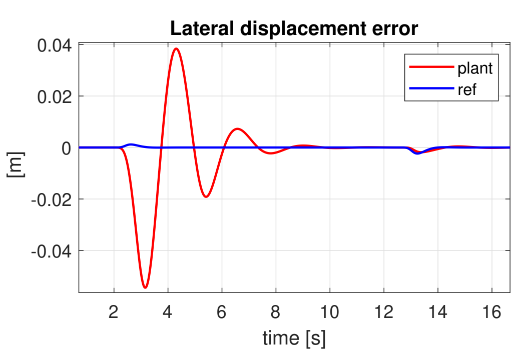


(c) Yaw rate x_2

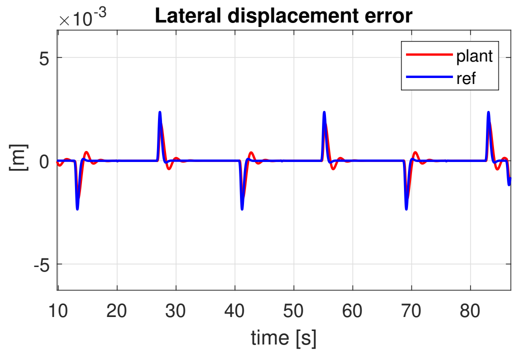


(d) Heading angle error x_4

Figure 5.1: MRAC: states tracking



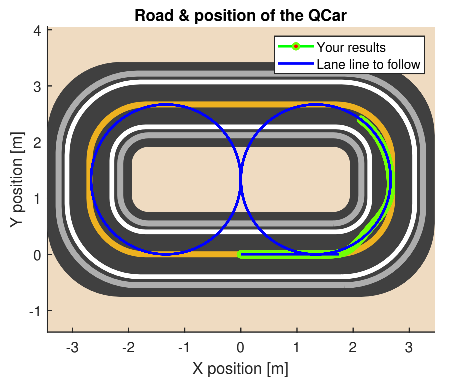
(a) x_3 first lap



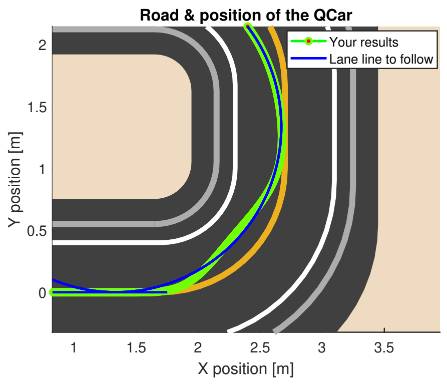
(b) x_3 second and third lap

Figure 5.2: Lateral displacement error adaptation

The high error present at the beginning of the path is better highlighted in Figure 5.3, showing the bigger error at the beginning of the curve. This initial behaviour of the MRAC algorithm is inherently related with its adaptive nature.



(a) Path tracking during the first lap



(b) Detail about the first curve

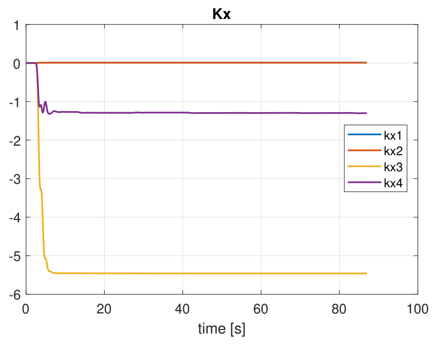
Figure 5.3: Deviation with respect to the reference trajectory during the first lap

In Figure 5.4 and Figure 5.5 is shown that at the beginning of the simulation the gain are rapidly adapted in order to reach a configuration in which the vehicle is controlled by the algorithm. It can be seen that the components of α_X and α_R reach stability within 20 seconds. The constants α and β are not the only one that require tuning, in fact recalling the Equation 4.2.10 reported below it becomes clear that properly tuning the matrix Q is crucial since it impacts on y_e.

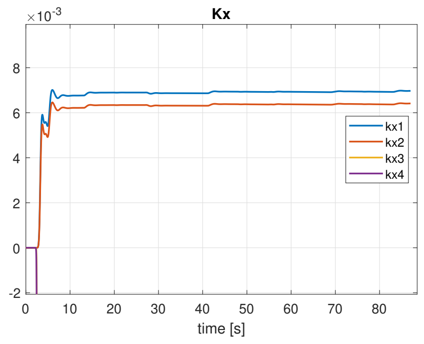
$$y_e = B^T P_e x_e \quad with \quad P_e A_m + A_m^T = -Q$$

In section B.1 the sensitivity towards the tuning of α, and Q is reported; the results reported in this section rely on the following tuning Q matrix:

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 1e4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(a) K_X adaptation



(b) Detail regarding K_{X1} and K_{X2}

Figure 5.4: MRAC K_X adaptation

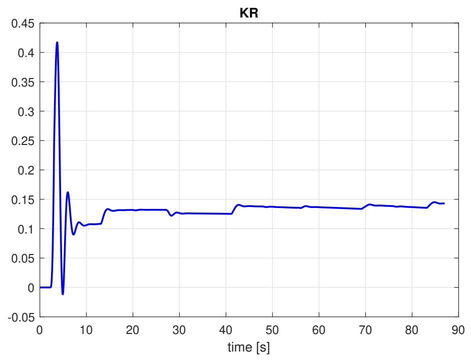
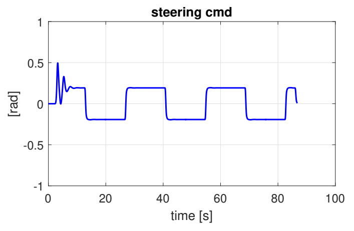


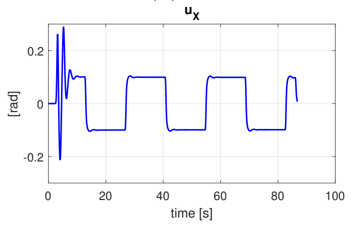
Figure 5.5: MRAC K_R adaptation

Finally, the control input for the MRAC algorithm presented in Equation 5.2.1 (recalled below), is shwon in Figure 5.6

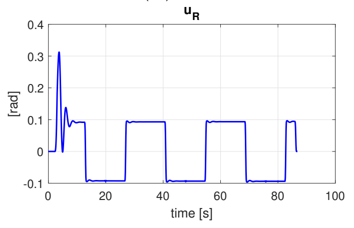
$$u_{\text{MRAC}}(t) = K_X(t)x(t) + K_R(t)\kappa(t)$$



(a) u



(b) u_x



(c) u_r

Figure 5.6: MRAC control input

5.2.2 EMRAC

The enhanced version of the MRAC considered previously in section 4.2 relies on the presence of the integral action u_I as well as an adaptive switching control action u_N; while the first one aims at improving the tracking of the reference model in the case of unmodeled biasis in the plant, the second one increases the robustness of the closed-loop tracking performance in case of varying bounded disturbances. The control action is therefore the one presented in Equation 5.2.1 recalled below.

$$u(t) = u_{\text{MRAC}}(t) + u_D(t) + u_I(t) + u_N(t)$$

With the gain adaptation laws recalled from Equation 4.2.7 and Equation 4.2.15 reported below:

$$K_I = \Phi_I + S^T y_e x_I^T \beta_I \quad and \quad \dot{\Phi}_I = S^T y_e x_I^T \alpha_I + F_I$$

$$K_N^{(uv)} = S^T \Phi_{N0}$$

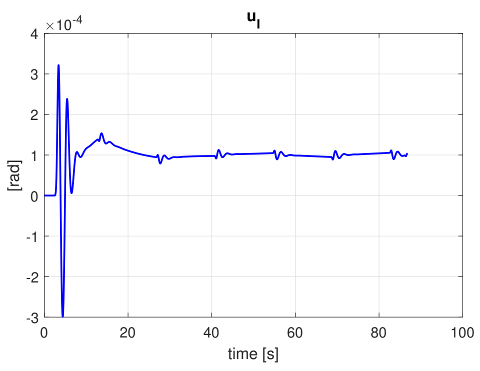
$$\dot{\Phi}_{N0} = \alpha_{N0} h_0 \left(\|y_e\|_\Omega\right) - \sigma_{N0}\left(\|\Phi_{N0}\|\right) \rho_{N0} \Phi_{N0}$$

In the simulation environment, the plant is the single track model of the vehicle developed in [24], this implies that the reference model accurately represents the actual plant. For this reason, as seen in previous section, the standard MRAC is already capable to steer the dynamics of the plant towards the reference model, making contributes like u_I and u_N unnecessary; however, being able to test the EMRAC in a simulation environment allows to understand wether the control architecture is correct or not.
For this reason, considering the same configuration as in the EMRAC, the following tuning parameters are considered for the integral action and the adaptive switching control action:

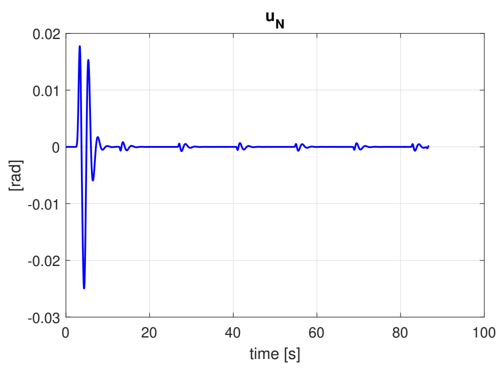
$$\alpha_I = \begin{bmatrix} 1e-4 & 0 & 0 & 0 \\ 0 & 1e-4 & 0 & 0 \\ 0 & 0 & 1e-3 & 0 \\ 0 & 0 & 0 & 1e-3 \end{bmatrix} \quad \beta_I = \begin{bmatrix} \text{1e-06} & 0 & 0 & 0 \\ 0 & \text{1e-6} & 0 & 0 \\ 0 & 0 & \text{1e-5} & 0 \\ 0 & 0 & 0 & \text{1e-5} \end{bmatrix}$$

$$\alpha_N = [0.005]$$

The contribution of the enhancing terms are shown in Figure 5.7a and Figure 5.7b, clearly they have negligible values with respect to u_X and u_R. The adaptive gains converge pretty rapidly as shown in Figure 5.8 where steady state is reached within 10 seconds. The treshold selected for $\widehat{M}_{N0}$ is coherent with the dynamics as shown in Figure 5.9; therefore, the σ-modification is able to properly control the norm of the switching control action.
In section B.2 more details are shown regarding the key aspects of the EMRAC tuning: showing the impact of different α_N and α_I as well as the discharge factor ρ_{N0}.



(a) Integral action u_I



(b) Adaptive switching control action u_N

Figure 5.7: u_I and u_N control input

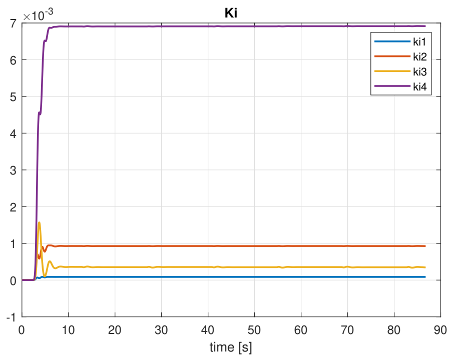
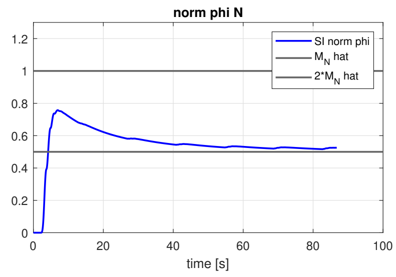
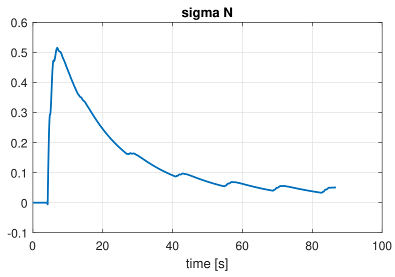


Figure 5.8: K_I adaptation



(a) Φ_N evolution



(b) σ_{N0} evolution

Figure 5.9: Φ_N norm control

5.2.3 EMRAC-NN

The neural network based augmentation has been introduced previously in section 4.4, where a two-layer NN is employed in order to deal with systems characterized by unstructured uncertainty; the structure is recalled in Figure 5.10, with the corresponding simulink block shown in Figure 5.11. The control action of this EMRAC-NN version was presented in Equation 4.4.1 and recalled below.

$$u_{EMRAC-NN} = u_{EMRAC} + u_{NN} = u_{EMRAC} + g\left(\Theta^T \Phi\left(W^T \bar{x}\right)\right)$$

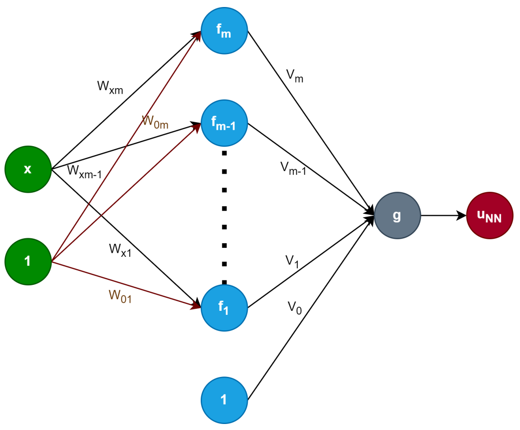


Figure 5.10: NN structure

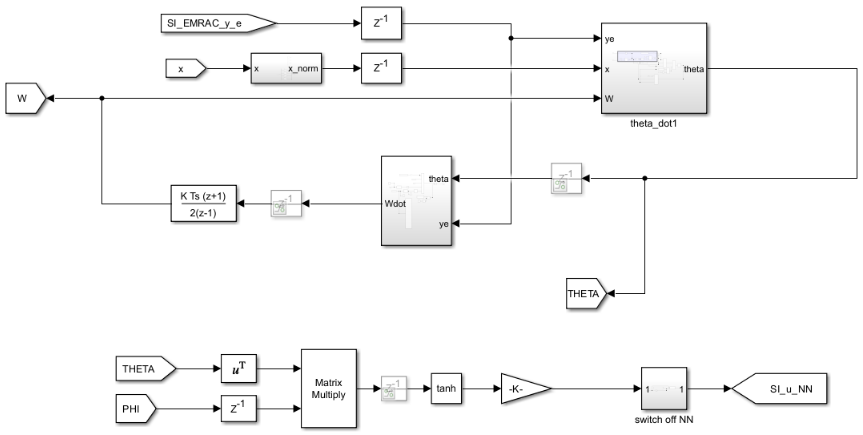


Figure 5.11: NN-augmentation, simulink block

In this work the definitive version of the Neural Network utilizes the RELU as activation function f, shown in Equation 5.2.2a; therefore f', being the derivative, is shown in Equation 5.2.2b; finally, the output activation function g is shown in Equation 5.2.2c

$$f(c) = \max(0,\ c) \tag{5.2.2a}$$

$$f'(c) = \begin{cases} 0 & if\, c < 0 \\ 1 & if\, c \geq 0 \end{cases} \tag{5.2.2b}$$

$$g(c) = 0.3 \tanh(c) \tag{5.2.2c}$$

Recalling Equation 4.4.2 reported below, it becomes clear that the initialization of the neurons weights and the learning rates Γ_Θ and Γ_W are parameters to be tuned.

$$\dot{\Theta} = \Gamma_\Theta \Phi \left(W^T \bar{x} \right) y_e$$

$$\dot{W} = \Gamma_W \bar{x} y_e V^T f' \left(W^T \bar{x} \right)$$

Considering the number of Neurons $m = 4$ and setting Γ_Θ and Γ_W as:

$$\Gamma_\Theta = \Gamma_W = \begin{bmatrix} 0.07 & 0 & 0 & 0 & 0 \\ 0 & 0.07 & 0 & 0 & 0 \\ 0 & 0 & 0.07 & 0 & 0 \\ 0 & 0 & 0 & 0.07 & 0 \\ 0 & 0 & 0 & 0 & 0.07 \end{bmatrix}$$

the following results are obtained. In Figure 5.13 the tracking of the reference states is shown, together with a detail regarding the lateral error in the third lap shown in Figure 5.13. It is important to recall that in the simulation environment the plant is well represented by the reference system, and therefore the errors are still very small and there is no actual need for for this NN contribution. However, it is crucial to assess the the control architecture works well before proceeding with the deployment in an experimental setting.
Nevertheless it appears that the lateral displacement error has improved with respect to previous EMRAC and MRAC controllers.

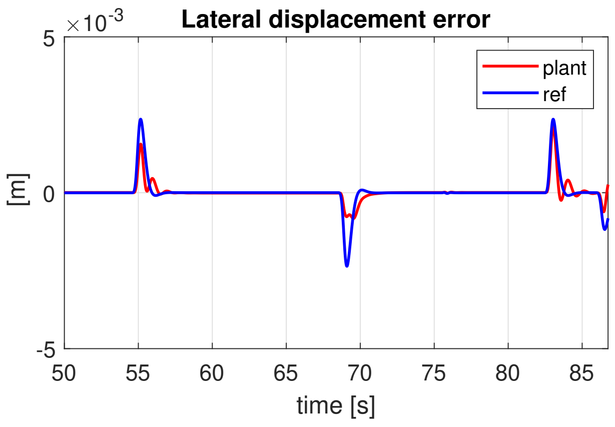
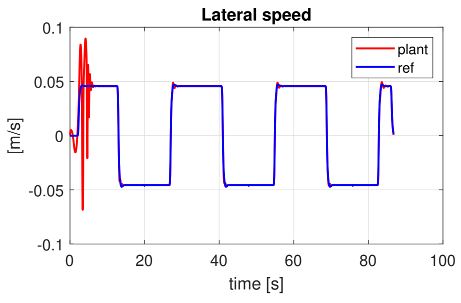
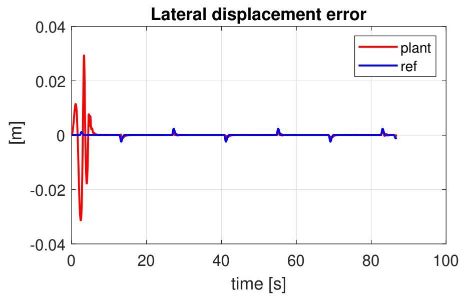


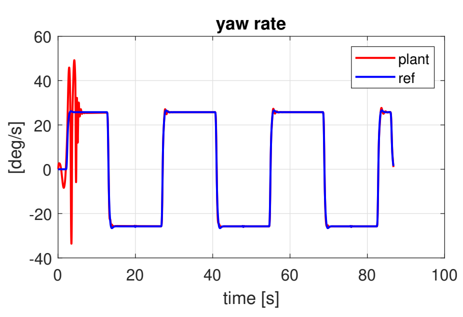
Figure 5.12: x_3 third lap



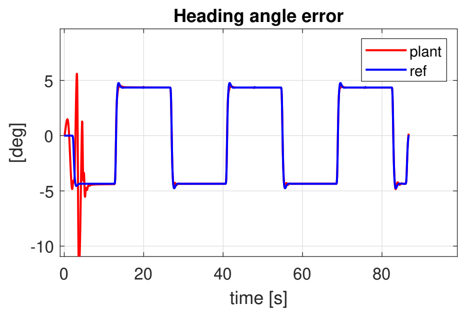
(a) Lateral speed x_1



(b) Lateral displacement error x_3



(c) Yaw rate x_2



(d) Heading angle error x_4

Figure 5.13: EMRAC-NN: states tracking

Furthermore, in Figure 5.14 the contribution u_{NN} of the Neural Network is shown; clearly, since the output function is $g(c) = 0.3 \tanh(c)$, the output is limited between -0.3 rad an +0.3 rad. This saturation avoids a potential divergence due to the initial adaptation of the NN; this adaptation is shown in Figure 5.15.

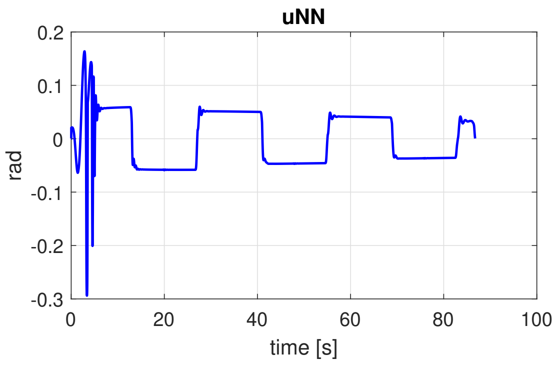


Figure 5.14: Neural Network contributionu_{NN}

In section C.2 the impact of the learning rate is presented in Figure B.10; it is evident that a higher learning rate allows to capture better the dynamic of the plant. Nevertheless, if the learning rate is too high it leads to divergence as shown in Figure B.10f, where a learning rate equal to 1 stops the simulation earlier due to divergence. This issue will be particularly important in the experimental setup; were it will be important to have a fast enough adaptation while guaranteeing stability.

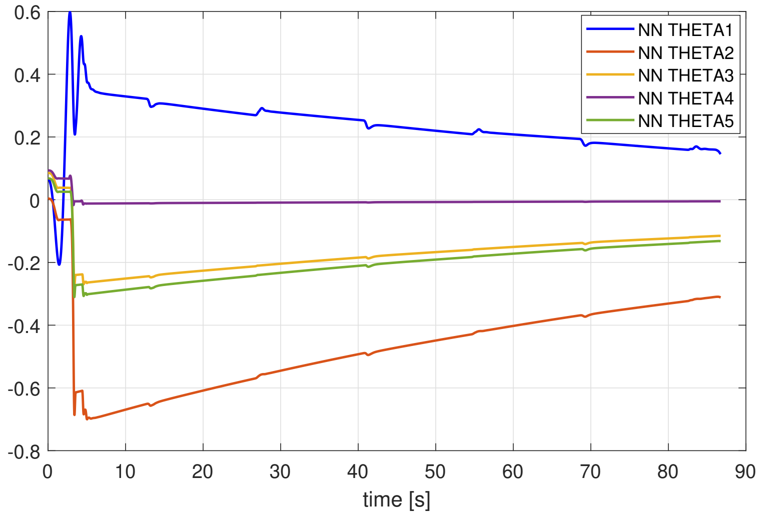


Figure 5.15: Neural Network contribution u_{NN}

5.3 Results

The results reported so far regard the path tracking on the eight trajectory at a velocity of 0.6 m/s. In section B.4 the results for other trajectories at 0.6 m/s are shown, however when other velocities are employed the tuning of the Enhanced Model Reference Adaptive Control needs to be updated. The first strategy in order to tune this controller was the scaling of the constants α and β; while this is a valid choice, it is definitely a tedious procedure since it requires a trial and error procedure that, even if automated, is computationally expensive (due to the high number of constants).
Recalling the adaptive gains computation in subsection 4.2.2, it is possible to indirectly tune y_e by tuning the the matrix Q, i.e. the solution of Equation 4.2.10. To better highlight this issue consider section B.6,it is reported the impact of the Q matrix on KPIs related with the lateral displacement error as well as the heading angle error at a velocity equal to 0.5 m/s; with respect to the Q matrix at 0.6 m/s a scaled version in the form of $Q_{0.5} = 1.5Q_{0.6}$ guarantees lower errors and better tracking of the reference system.
Furthermore, in section B.5 the simulations are performed at different velocities.

Chapter 6

Experimental Design

Through the results obtained during the design and validation of the controller in a simulation environment, the next stage of this thesis is the validation of the controller in an experimental environment. As stated in previous sections the vehicle employed is the QCar, a scaled fully automated vehicle recalled in Figure 6.1. As for the simulations environment, the analysis is conducted at a velocity equal to 0.6 m/s and subsequently the to other velocities, comparing the performance of the EMRAC and EMRAC-NN with respect to the benchmark controller.

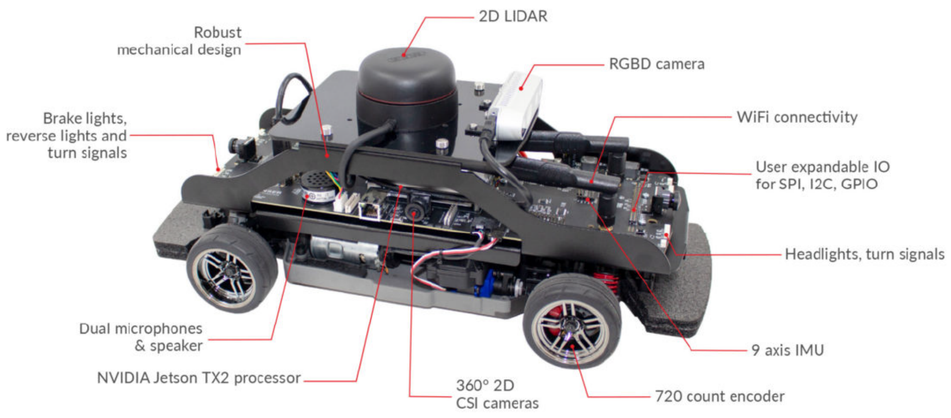
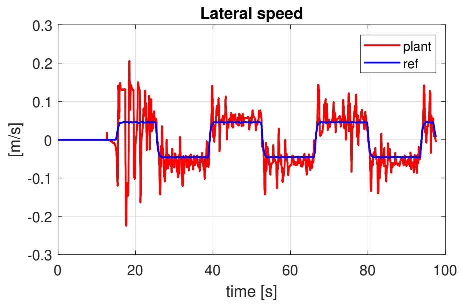

2D LIDAR
Robust mechanical design
RGBD camera
WiFi connectivity
Brake lights, reverse lights and turn signals
User expandable IO for SPI, I2C, GPIO
Headlights, turn signals
Dual microphones & speaker
9 axis IMU
NVIDIA Jetson TX2 processor
360° 2D CSI cameras
720 count encoder

Figure 6.1: QCar vehicle

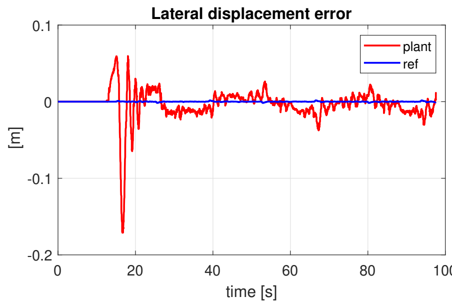
6.1 EMRAC

Starting from the tuning done in the simulation environment, a scaling and some adjustments were required in order to guarantee stability of the control algorithm; coherently with the sampling time of 10 ms of the lidar employed for the localisation. With the tuning reported in subsection C.1.1 the experimental simulation lead to the tracking of the reference system shown in Figure 6.2 with the corresponding KPIs presented in Table D.4; finally, the vehicle localisation is shown in subsection C.1.2.

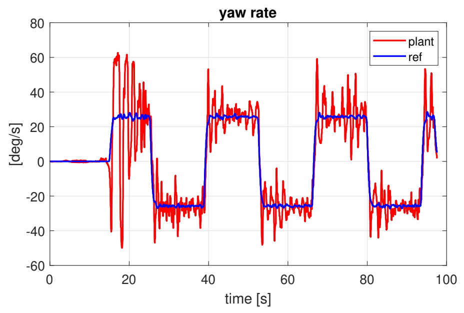
It is clear that reference system is extremely simplified with respect to the actual plant. In fact, the lateral displacement error of the reference system in the third lap, consistently with what seen in the previous section, has values up to 2.5mm much different with respect to the 2.5cm recorded during this experimental simulation.



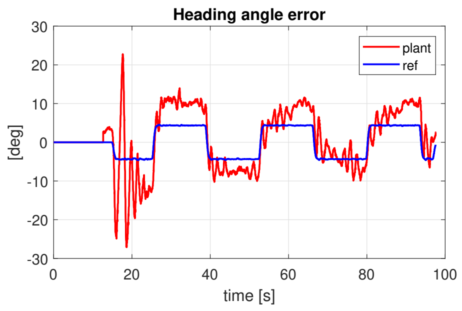
(a) Lateral speed x_1



(b) Lateral displacement error x_3



(c) Yaw rate x_2



(d) Heading angle error x_4

Figure 6.2: EMRAC: states tracking, experimental results

Likewise the simulation environment, in the experimental environment the adaptive behaviour of this algorithm is even more evident. In Figure 6.3, Figure 6.4, and Figure 6.5 the path followed during the different laps is shown. It is clear that the EMRAC algorithm improves continuously its tracking performance as it proceeds during the path tracking.
Furthermore, the control action is shown in Figure 6.6 and the different contributions are shown in Figure 6.7. After careful experimental analysis results suggested to notably decrease the contribution of the integral action u_I in order to avoid instability; as shown in Figure 6.13d. The contribution of the feed-forward action u_R seems to get more and more significant during the experiment suggesting that it may lead to instability; this consideration would be wrong since the evolution of the feed-forward gain is limited by the locking strategy. In fact, as shown in Figure C.2e ϕ_R has not reached the upper bound, therefore its evlution is under control, guaranteeing stability. Finally the gains evolution is reported in Figure 6.8.

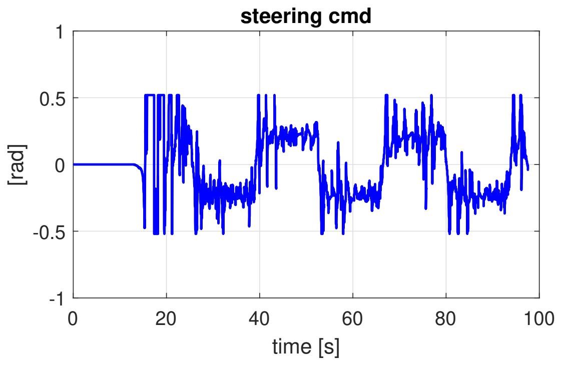
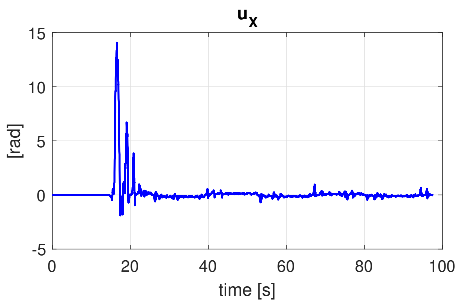
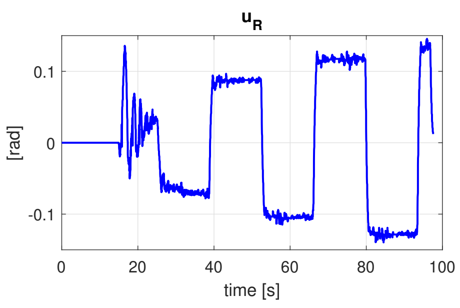


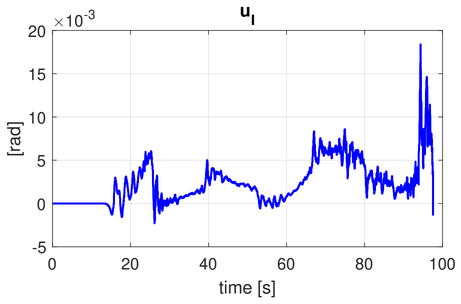
Figure 6.6: Control action u, experimental



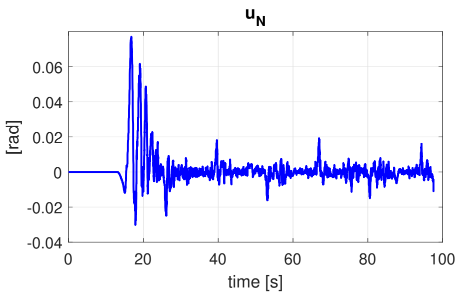
(a) u_X action



(b) u_R action

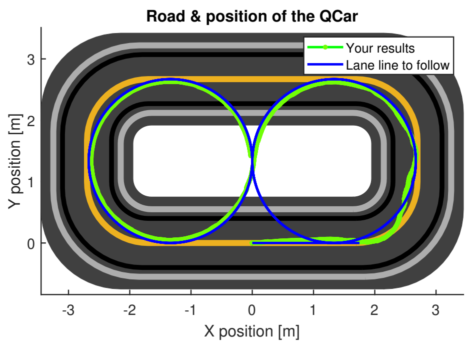


(c) u_I action

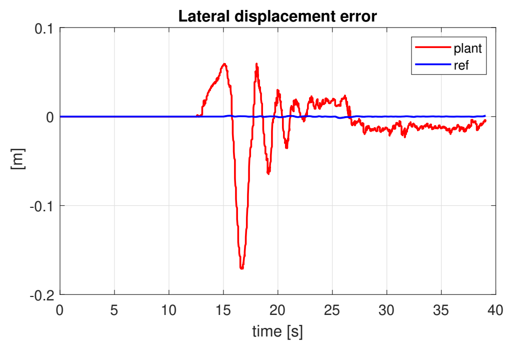


(d) u_N action

Figure 6.7: EMRAC control contributions

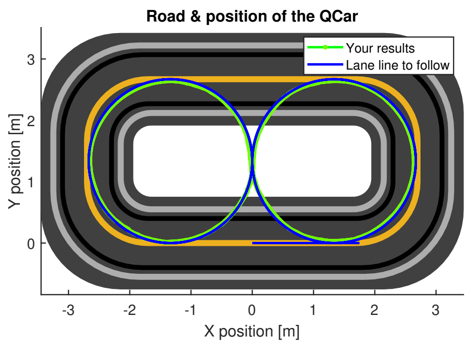


(a) Trajectory during the first lap

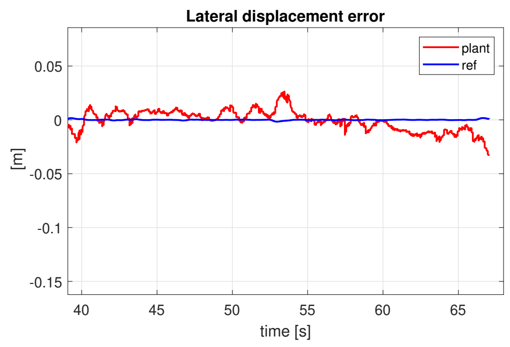


(b) Lateral displacement error x_3, 1st lap

Figure 6.3: Path tracking during the first lap

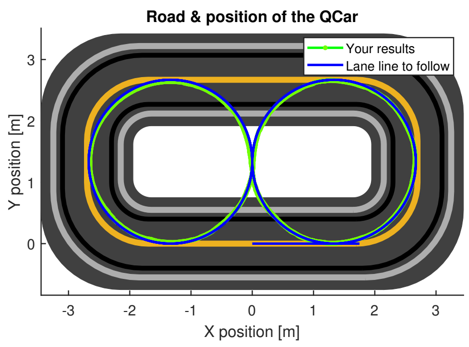


(a) Trajectory during the 2nd lap

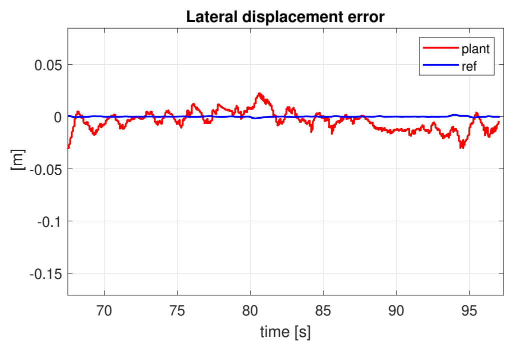


(b) Lateral displacement error x_3, 2nd lap

Figure 6.4: Path tracking during the second lap

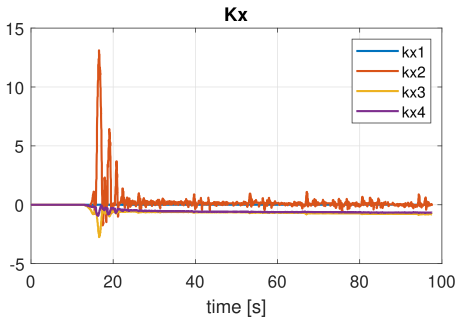


(a) Trajectory during the 3rd lap

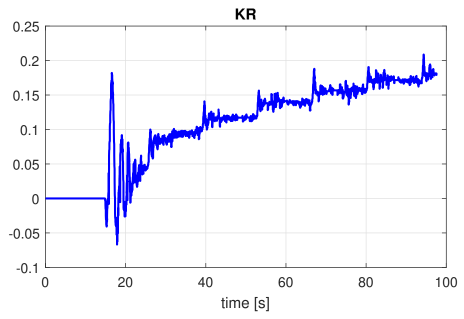


(b) Lateral displacement error x_3, 3rd lap

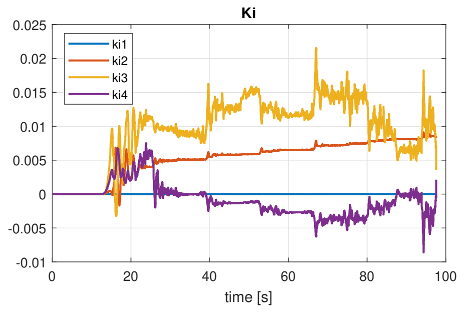
Figure 6.5: Path tracking during the third lap



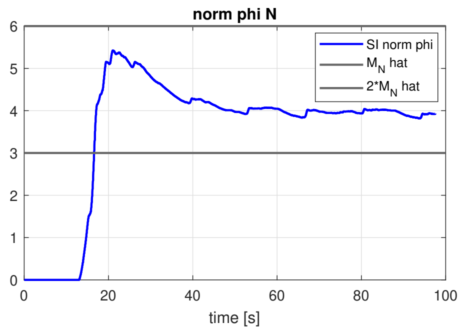
(a) K_X



(b) K_R



(c) K_I

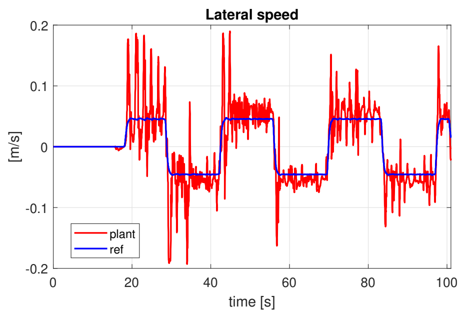


(d) ϕ_N

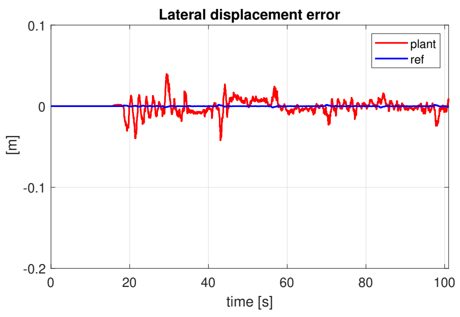
Figure 6.8: EMRAC: gains evolution, experimental results

6.2 EMRAC-NN

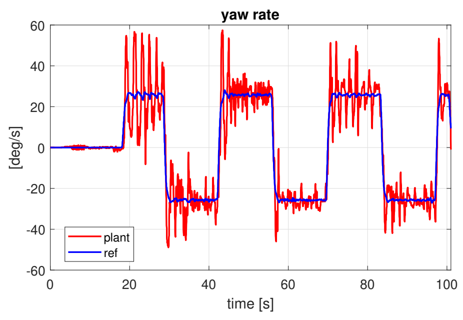
The experiments carried for the EMRAC-NN follow the same procedure as for the standard EMRAC, with the additional contribution due to the neural network. The experimental tuning is carried at a velocity equal to 0.6 m/s and consequently the design constants are scaled for different velocities.
In Figure 6.9 the tracking of the reference states is shown; it is already possible to see a notable improvement with respect to the EMRAC shown in previous section, however this will be discussed in the following. As per the EMRAC, in Figure 6.10, Figure 6.11, and Figure 6.12 the lateral displacement error and the tracking for different laps is reported; furthermore, in Figure 6.13a the control input is shown with its contributions in Figure 6.13. Regarding the Neural Network, the adaptation interests the neuron's weights, reported in Figure 6.14; with the corresponding control action in Figure 6.13f. Finally, the experimental figures are reported in section C.2.



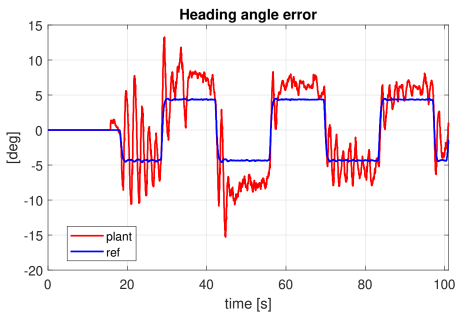
(a) Lateral speed x_1



(b) Lateral displacement error x_3

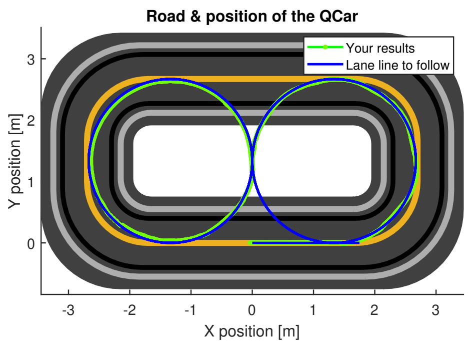


(c) Yaw rate x_2

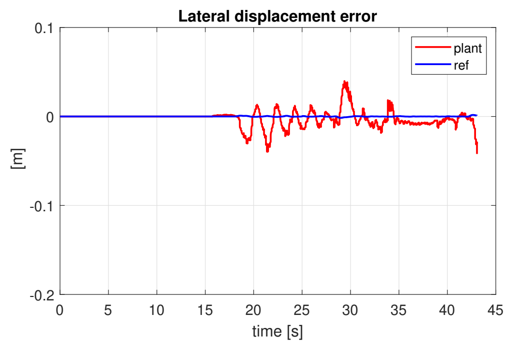


(d) Heading angle error x_4

Figure 6.9: EMRA-NN: states tracking, experimental results

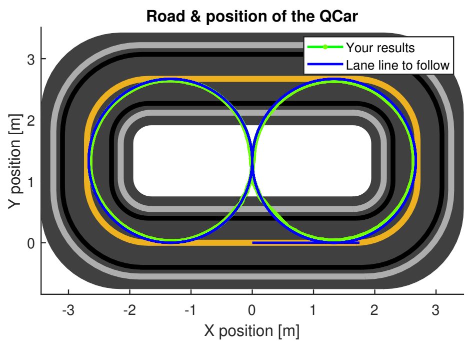


(a) Trajectory during the first lap

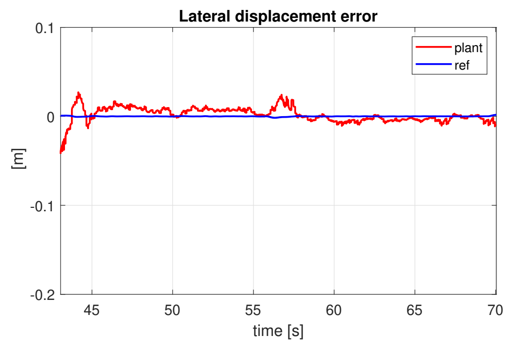


(b) Lateral displacement error x_3, 1st lap

Figure 6.10: Path tracking during the first lap

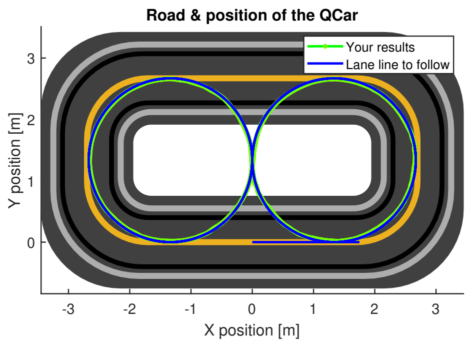


(a) Trajectory during the 2nd lap

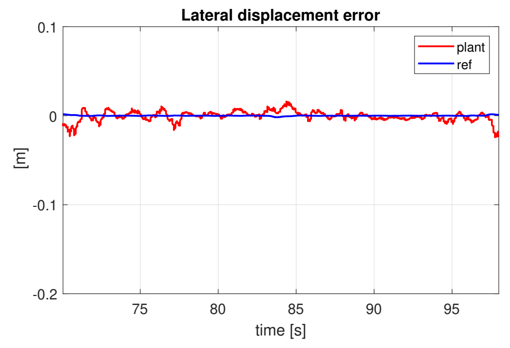


(b) Lateral displacement error x_3, 2nd lap

Figure 6.11: Path tracking during the second lap

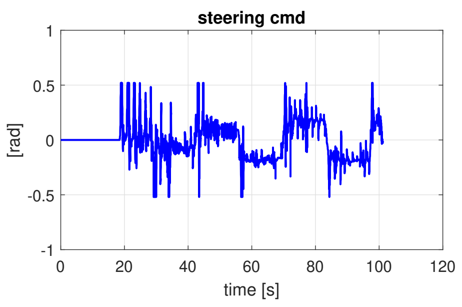


(a) Trajectory during the 3rd lap

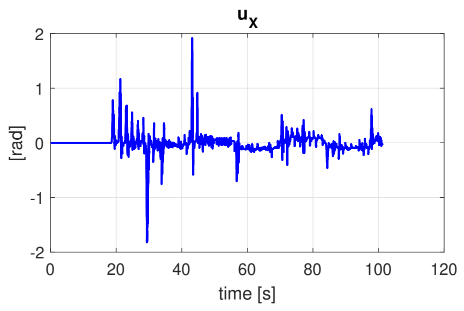


(b) Lateral displacement error x_3, 3rd lap

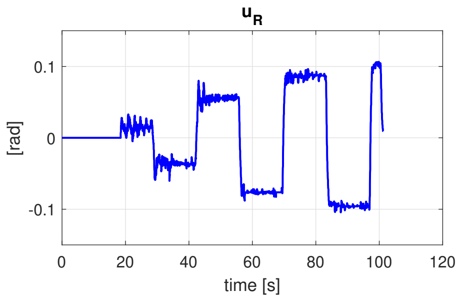
Figure 6.12: Path tracking during the third lap



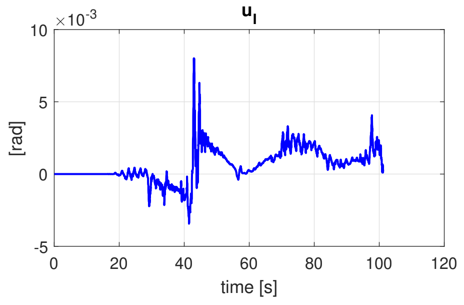
(a) u



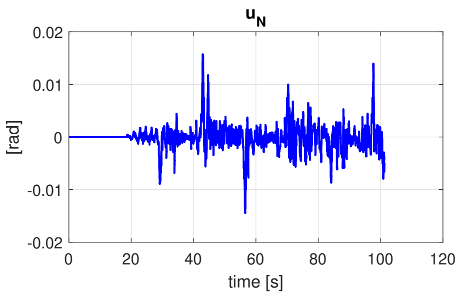
(b) u_X action



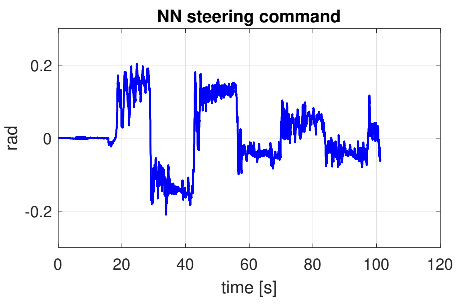
(c) u_R action



(d) u_I action

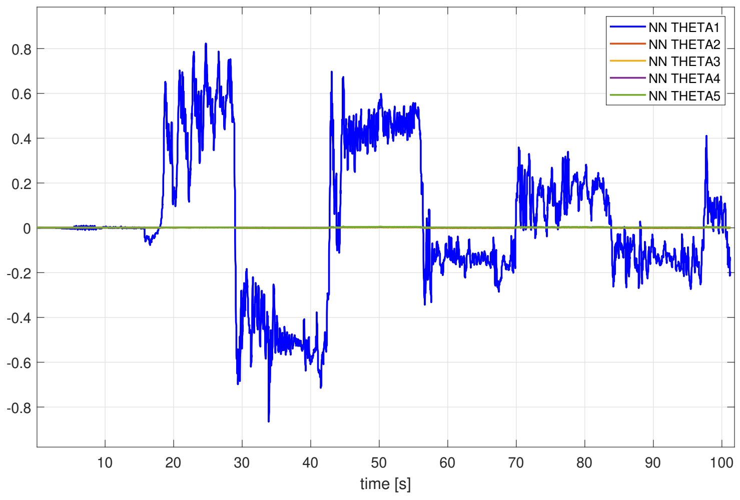


(e) u_N action

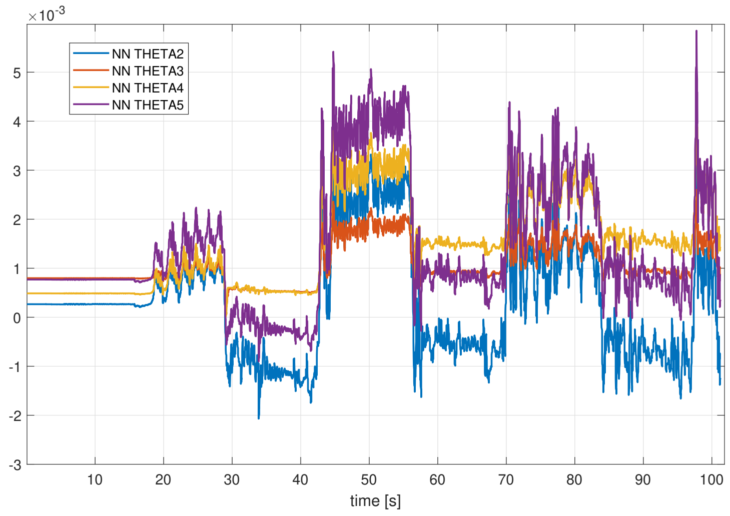


(f) u_{NN} action

Figure 6.13: EMRAC control contributions



(a)



(b)

Figure 6.14: Neural Network weights

6.3 Results Analysis

After a collecting the experimental results in the form of KPIs reported in Appendix D it is now possible to evaluate objectively the performance of the designed EMRAC and EMRAC-NN with respect to the benchmark controller. By considering the following figures, regarding the third lap, it emerges that the EMRAC augmented with the Neural Network is capable of beating the Pole Placement at every velocity, improving also the performance with respect to the standard EMRAC, both in terms of path tracking and control.

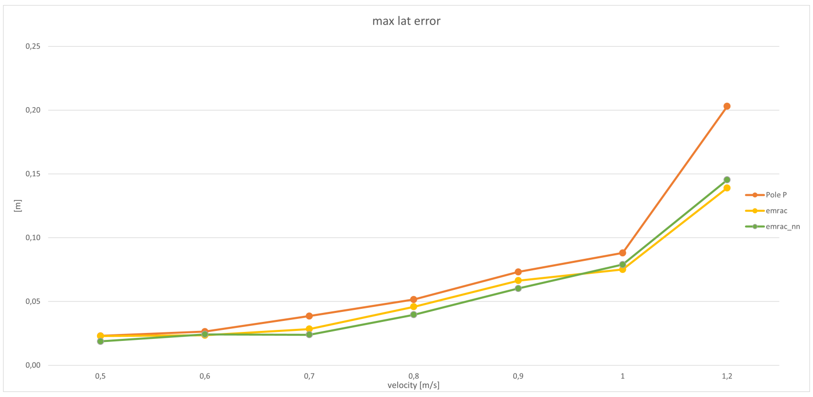


Figure 6.15: Maximum lateral displacement error, 3rd lap

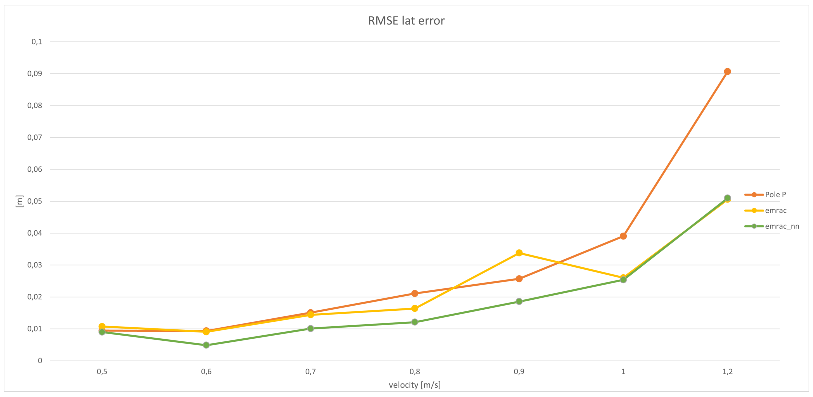


Figure 6.16: RMSE lateral displacement error, 3rd lap

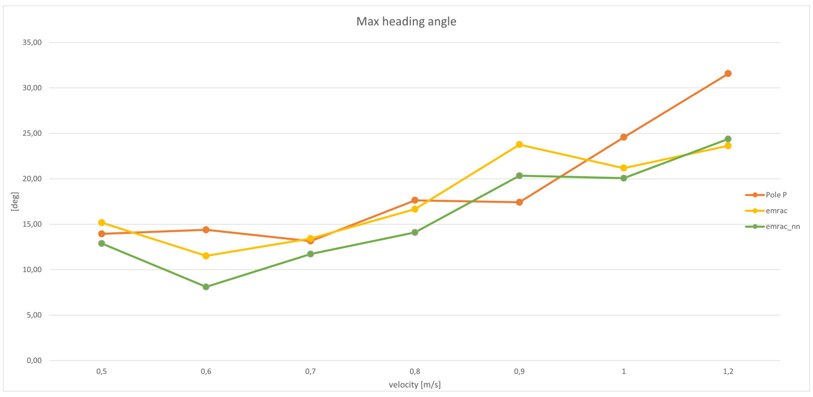


Figure 6.17: Maximum heading angle error, 3rd lap

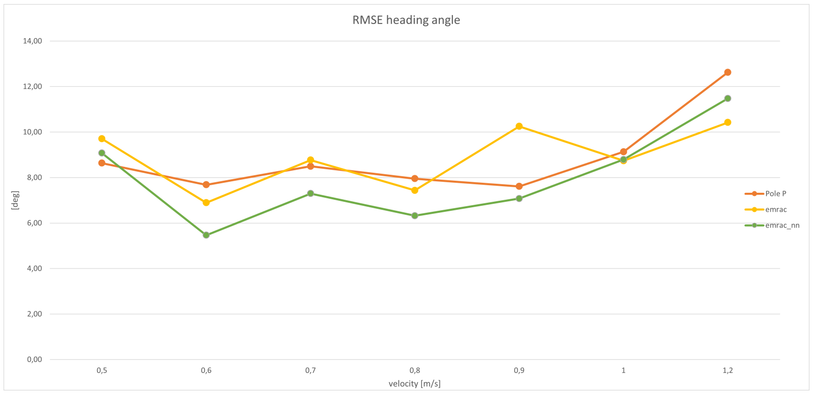


Figure 6.18: RMSE heading angle error, 3rd lap

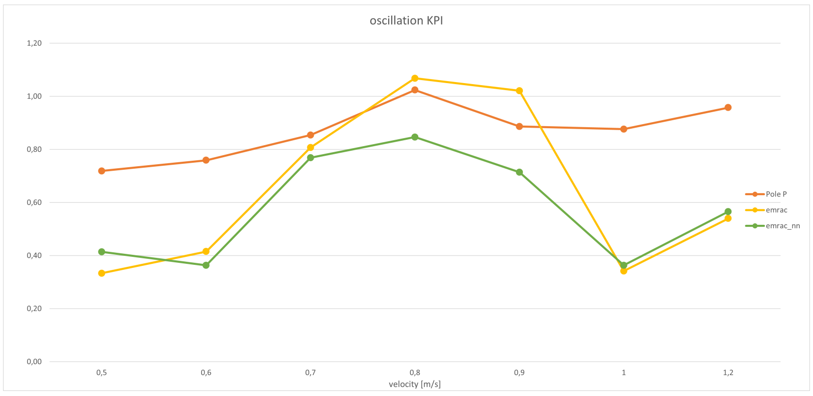


Figure 6.19: Control action oscillation KPI, 3rd lap

Chapter 7

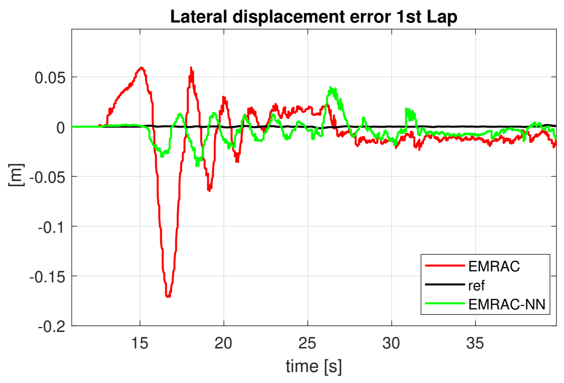
Conclusion & future work

7.1 Conclusion

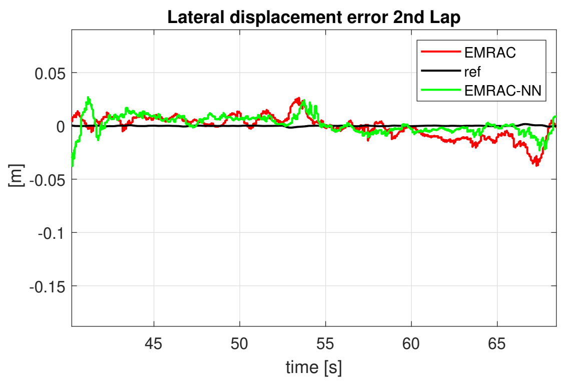
The research activity successfully achieved the predetermined objectives, culminating in a proficient experimental validation of the Enhanced Model Reference Adaptive Control algorithm developed in [5]. This project also outlined an additional augmentation through the use of a Neural Network.
The EMRAC demonstrated superior outcomes in lateral displacement error and heading angle error compared to the benchmark controller across nearly all velocities. As depicted in Figure C.19 the upgraded path tracking performance of the EMRAC was reached at the expense of a more aggressive control action; the Neural Network augmentation solved this problem while concurrently achieving superior path-tracking.
To better visualize the improvements to the EMRAC provided by the neural network, Figure 7.1 and Figure 7.2 are particularly useful. Those figures show how the NN-based augmentation is able to improve the tracking of the reference system. Furthermore, the Neural Network is capable of of mitigating the problems related with the adaptation, in fact in the first lap the controller is capable of guaranteeing better tracking capabilities with respect to the standard EMRAC.

7.2 Future work

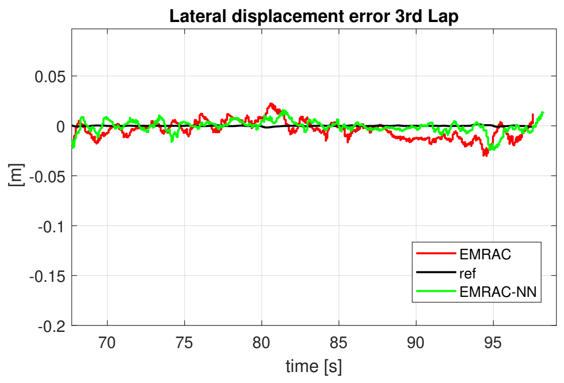
Starting from the results of this research project it is possible to continue in several ways. First of all, the development of a more accurate plant model will lead to a faster design of the controller; in fact, in simulation environment the plant was the bicycle model that clearly has some limitations that required to re-do the tuning in the experimental setting.
Secondly, the Neural Network augmentation can be substituted with more sophisticated architectures that surely can lead to improvements. In this case the attention should shift on the communication between the QCar and laboratory computer in order to assure that more complex architectures are possible and how to make them less demanding from the computation point of view.
Finally, vehicle platooning technologies may be developed in order to test the EMRAC on scenarios that resemble real traffic conditions.



(a) x_3 1st lap

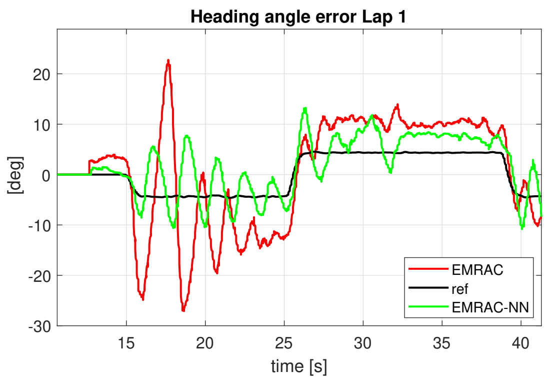


(b) x_3 2nd lap

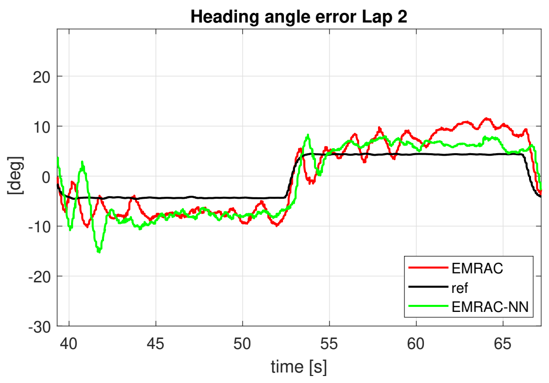


(c) x_3 3rd lap

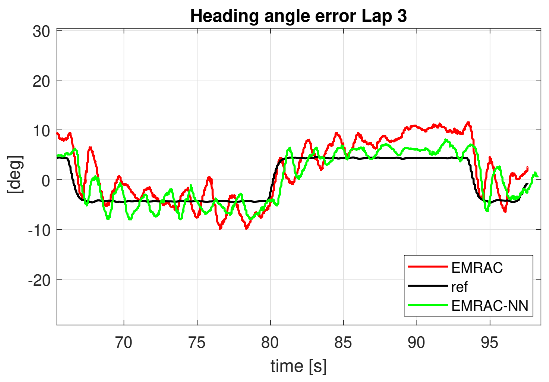
Figure 7.1: Lateral displacement error at 0.6 m/s, comparison between EMRAC and EMRAC-NN



(a) x_4 1st lap



(b) x_4 2nd lap



(c) x_4 3rd lap

Figure 7.2: Heading angle error at 0.6 m/s, comparison between EMRAC and EMRAC-NN

Appendix A

Code

A.1 Initialization script

```
%% LAUNCH the version without optimization
%load ("QCar_param.mat")
cd('C:\Users\user\Documents\Qcar\PAOLO\emrac0409\scripts and simulink')
load('gains') % gains of the reference system for different velocities
run('Init_QCar.m'); % initialization of the QCar parameters (velocity also)
run('vehicle_parameters.m') % Used for the reference system for the EMRAC
run("system_parameters_PAPER.m"); % gain computation and reference system

load('MAF.mat');
run('init_EKF_D_vy_lab.m');
% initialization of the EMRAC parameters
run("SI_EMRAC2_bicycle22v2Discrete_newRef_withPP.m")
run("initNeuralNetwork.m") % neural networks
% discrete time integral sampling time, I suggest to not change this value
TsIntegral = 10/1000;

%%
run('Init_circular_trajectory.m');
traj_name = 'C';
%%
run('Init_eight_trajectory.m');
traj_name = 'E';

%%
run('Init_eight_trajectory.m');

traj_name = 'E';

%%
run('Init_obstEMRAC.m');
traj_name = 'E';
```

QCar Initialization

```
load('QCar_param.mat');

step_time_voltage = 5;

QCar.long_speed = 0.6;
QCar.Lidar_Ts = 0.1;
QCar.Ts = 0.01;
QCar.MaxSteerAngle = 0.5;
%QCar.Gr*QCar.r_w; % v = w*tau*R (This value represents tau*R)
QCar.tau_R = 0.0031;
QCar.P1 = QCar.K_t/(QCar.J*QCar.R);
QCar.P2 = QCar.K_t*QCar.K_v/(QCar.J*QCar.R)+QCar.B/QCar.J;
QCar.P3 = QCar.C/QCar.J;
QCar.v_threshold = 0.1;
```

A.2 System parameters

```
%% Define QCAR system parameters
%load ("QCar_param.mat")
% [N/rad] Front tire cornering stiffness
Ca_F        = SI_EMRAC.Vehicle.Ca_lin;
% [N/rad] Rear tire cornering stiffness
Ca_R        = SI_EMRAC.Vehicle.Cp_lin;
% [kg] Vehicle mass
m           = SI_EMRAC.Vehicle.m;
% [m/s] Longitudinal speed
Vx          = QCar.long_speed;
% [kg*m^2] Vehicle inertia
I_z         = SI_EMRAC.Vehicle.Iz;
% [m] CoG - front tire distance
l_f         = SI_EMRAC.Vehicle.Wb_half_a;
% [m] CoG - rear tire distance
l_r         = SI_EMRAC.Vehicle.Wb_half_b;
% [m] Wheelbase
L           = SI_EMRAC.Vehicle.Wb_half_a + SI_EMRAC.Vehicle.Wb_half_b;
la_dist     = 0.5;


%% PAPER MODEL MODIFIED WITH RIGHT CONVENTION

A = [        -(Ca_F+Ca_R)/(m*Vx)                  ...
    -((l_f*Ca_F - l_r*Ca_R)/(m*Vx))-Vx         0                 0;
     -((l_f*Ca_F - l_r*Ca_R)/(Vx*I_z))        ...
     -((l_f^2*Ca_F + l_r^2*Ca_R)/(Vx*I_z))      0                 0;
                   1      0         0                 Vx;
                   0      1         0                 0];

B1 = [Ca_F/m   (l_f*Ca_F)/(I_z)      0         0]';

B2 = [0    0    0    -Vx]';


%% Feedback term calculation (PAPER MODEL MODIFIED)


% reference system: gains computed through gain scheduling
index = find(gains(:,1)==QCar.long_speed)
K = gains(index,2:end-1)

Acl = A+B1*K;
Am = Acl;
sysclosed = ss(Acl, B1, eye(4), []);
Pcl = pole(sysclosed);

k1 = K(1);
k2 = K(2);
k3 = K(3);
k4 = K(4);

%% Feedforward term calculation (PAPER MODEL MODIFIED)
Kr_star = (Ca_F*Ca_R*l_f^2 + Ca_F*Ca_R*l_r^2 + Ca_F*Ca_R*k4*l_r^2 - ...
    Ca_F*Vx^2*l_f*m + Ca_R*Vx^2*l_r*m + ...
             2*Ca_F*Ca_R*l_f*l_r - Ca_F*Ca_R*Vx*k2*l_f - Ca_F*Ca_R*...
             Vx*k2*l_r + Ca_F*Ca_R*k4*l_f*l_r - ...
             Ca_F*Ca_R*Vx*k1*l_r^2 + Ca_F*Vx^3*k1*l_f*m - ...
             Ca_F*Vx^2*k4*l_f*m - Ca_F*Ca_R*Vx*k1*l_f*l_r)/...
             (Ca_F*Ca_R*(l_f + l_r))

Bm = B1*Kr_star + B2;
%% Q must be diagonal and positive (must be tuned)
Q = 2*diag([1 1 500 1])
```

A.3 Controller Design Parameters

```
% the gains saved in "gains.mat" are stored in SI_EMRAC.gainsKx and SI_EMRAC.gainsKr
% because needed in a look-up table
SI_EMRAC.gainsKx = gains(1:end,2:end-1);
SI_EMRAC.gainsKr = [0.1242 0.1557 0.1933 0.2398...
    0.2895 0.3436 0.4485 0.4531]';

% if needed it is possible to initialize the gains of the emrac so the
% integral wont start from zero. Those gains should be stored in the
% "gainInitialization" folder as "gainInit_velocity.mat" as row vectors
initializeGainsToZero = 1;
if  initializeGainsToZero==0
    gainsToInitialize = ['gainInit_',num2str(QCar.long_speed),'.mat'];
    load(strcat("gainInitialization\",gainsToInitialize))
    SI_EMRAC.initKx = gainInit.Kx;
    SI_EMRAC.initKr = gainInit.Kr;
    SI_EMRAC.initKi = gainInit.Ki;
    SI_EMRAC.initPhiN = gainInit.PhiN;
    clear("gainInit","gainsToInitialize")
else
    SI_EMRAC.initKx = zeros(1,4);
    SI_EMRAC.initKr = 0;
    SI_EMRAC.initKi = zeros(1,4);
    SI_EMRAC.initPhiN = 0;
end

%%
%
% how much in advance the curvature is passed at the controller
SI_EMRAC.preview = 0.05;
%low pass filter for smooth reading from the look up table
SI_EMRAC.filter_den = [0.3 1];
SI_EMRAC.filter_num = [1];
SI_EMRAC.gain_yeN = 1; % correction on the third component of ye
%%

SI_EMRAC.n_x       = 4;                    % state dimension
SI_EMRAC.n_u       = 1;                    % input dimention

SI_EMRAC.S         = 1;
SI_EMRAC.P_phi_vet = Phi_R_hat*SI_EMRAC.S; % vector set of P_phi



%% gains
uu = 10/100; % coefficient used to scale everything on alpha
SI_EMRAC.alpha_X    = uu*diag([1e-2 1e1 5e1 5e1]);%100*diag([ones(MI_EMRAC.n_x,1)]); %nx*nx dimension positive
      diagonal
SI_EMRAC.alpha_R    = uu*1e-0; %nu*nu dimension positive diagonal
SI_EMRAC.alpha_N   = 5e-1; %positive constant
SI_EMRAC.alpha_I    = 1e-4*SI_EMRAC.alpha_X;%1e-7*uu*diag([1e-2 1e-2 1e-1 1e-2]); % 100*diag([ones(MI_EMRAC.
     n_x,1)]); %nx*nx dimension positive diagonal
%SI_EMRAC.alpha_I(3,3)    =  10*SI_EMRAC.alpha_I(3,3);

%%
tt=1/2; % coefficient to scale beta wrt to alfa
SI_EMRAC.beta_X     = tt*1*SI_EMRAC.alpha_X; %positive
SI_EMRAC.beta_R     = tt*SI_EMRAC.alpha_R; %positive
SI_EMRAC.beta_I     = tt*SI_EMRAC.alpha_I; %positive


%% Parameter Projection upper and lower bounds
SI_EMRAC.PP_active = 1;
SI_EMRAC.Kx_Lb = 1*K;
SI_EMRAC.Kx_Ub = -1*K;

SI_EMRAC.Kr_Lb = -1.5*Kr_star;
SI_EMRAC.Kr_Ub = 1.5*Kr_star;

SI_EMRAC.Ki_Lb = 10*K;
SI_EMRAC.Ki_Ub = -10*K;

%% Normalization
SI_EMRAC.Rx = 10*diag([1 1 1 1]);
SI_EMRAC.Rr = 10;
SI_EMRAC.normalization = 1;

%% e1 treshold
% when the lateral error is smaller than this treshold the adaptation is
% blocked
SI_EMRAC.e1_treshold = 5e-2;
SI_EMRAC.e1_treshold = 9e-2;


%% SI-EMRAC sigma modification parameters
SI_EMRAC.sigma_active = 0; % not used (it was a previous version were all
% sigma-modification were turned on and off all at the same time). Before
% activating this MAKE SURE to turn OFF the Parameter projection

SI_EMRAC.sigmaN_active = 1;
SI_EMRAC.sigmaI_active = 0;
if SI_EMRAC.sigma_active == 0
    disp('Sigma Modification ON')
else
    disp('Sigma mModification OFF')
end
%discharge factors
c = 1e-3;
%nx*nx dimension positive diagonal
SI_EMRAC.rho_e      = diag([ones(SI_EMRAC.n_x,1)]);

%nx*nx dimension positive diagonal
```

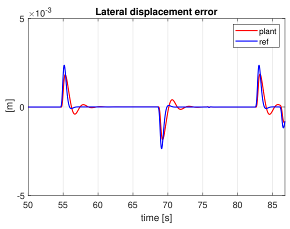
```
SI_EMRAC.rho_X      = c*diag([ones(SI_EMRAC.n_x,1)]);

%nu*nu dimension positive diagonal
SI_EMRAC.rho_R      = c*diag([ones(SI_EMRAC.n_u,1)]);

%nx*nx dimension positive diagonal
SI_EMRAC.rho_I      = 50*c*diag([ones(SI_EMRAC.n_x,1)]);
SI_EMRAC.rho_N     = 1e-1; %positive constant
SI_EMRAC.rho_N1 = 5e-2;
SI_EMRAC.rho_N2 = 5;

SI_EMRAC.GAMMA_alpha = diag([diag(SI_EMRAC.alpha_X);...
    diag(SI_EMRAC.alpha_R); ...
    diag(SI_EMRAC.alpha_I)])
SI_EMRAC.GAMMA_rho   = diag([diag(SI_EMRAC.rho_X);...
    diag(SI_EMRAC.rho_R); ...
    diag(SI_EMRAC.rho_I)])

SI_EMRAC.M_sigma          = 2*1e-3;


%%
%memory for M_phi_hat vector
SI_EMRAC.M_phi_hat_vet   = [];

for index = 1:length(SI_EMRAC.P_phi_vet)

    SI_EMRAC.P_phi = SI_EMRAC.P_phi_vet(index);
    % save each SI_EMRAC.M_phi_hat_temp into SI_EMRAC.M_phi_hat_vet
    SI_EMRAC.M_phi_hat_temp  = sqrt(max(eig(kron(SI_EMRAC.GAMMA_rho...
        *inv(SI_EMRAC.GAMMA_alpha),inv(SI_EMRAC.P_phi)))))...
                              /min(eig(kron(SI_EMRAC.GAMMA_rho...
                              *inv(SI_EMRAC.GAMMA_alpha),...
                              inv(SI_EMRAC.P_phi)))))*SI_EMRAC.M_sigma;
    SI_EMRAC.M_phi_hat_vet   = [SI_EMRAC.M_phi_hat_vet ...
        SI_EMRAC.M_phi_hat_temp];
end

% take max value of SI_EMRAC.M_phi_hat_vet as SI_EMRAC.M_phi_hat
SI_EMRAC.M_phi_hat = 1.1*max(SI_EMRAC.M_phi_hat_vet);
SI_EMRAC.M_phi_hat = 2;
clear index

%SI_EMRAC.M_I_hat    = 10000;                     %positive constant
%SI_EMRAC.M_I_hat    = 1e-1
SI_EMRAC.M_I_hat    = 0.5; % MODIFICA PAOLO
%SI_EMRAC.M_N_hat    = 10000;                     %positive constant
SI_EMRAC.M_N_hat = 8;
SI_EMRAC.eta_I      = 2;                         %positive constant
SI_EMRAC.eta_phi    = 12;
% SI_EMRAC.eta_phi    = 2*1e-4;

disp('condition to be verified')
[SI_EMRAC.eta_phi*min(eig(kron(SI_EMRAC.GAMMA_rho*...
    inv(SI_EMRAC.GAMMA_alpha),inv(SI_EMRAC.P_phi))))...
    3/4*min(eig(Q))]
[SI_EMRAC.M_phi_hat  sqrt(max(eig(kron(SI_EMRAC.GAMMA_rho...
    *inv(SI_EMRAC.GAMMA_alpha),inv(SI_EMRAC.P_phi))))/...
    min(eig(kron(SI_EMRAC.GAMMA_rho*inv(SI_EMRAC.GAMMA_alpha),...
    inv(SI_EMRAC.P_phi)))))*SI_EMRAC.M_sigma]
min(eig(kron(SI_EMRAC.GAMMA_rho*inv(SI_EMRAC.GAMMA_alpha),...
    inv(SI_EMRAC.P_phi))))
min(eig(inv(SI_EMRAC.GAMMA_alpha)*SI_EMRAC.GAMMA_rho))
SI_EMRAC.eta_N     = 1;                          %positive constant

% h(y_e) calculation
SI_EMRAC.sigma_0    =  1;
SI_EMRAC.gamma_0      = 2;
SI_EMRAC.xi_0        = 1;

%sign y_e calculation
SI_EMRAC.epsilon    = 70;
SI_EMRAC.epsilon    = 200;

return
```

Appendix B

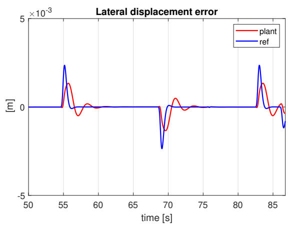
Tuning in simulation environment

B.1 MRAC

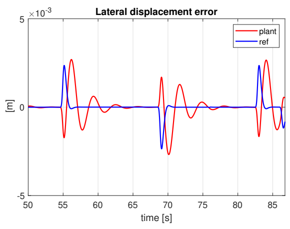
Tuning α_X and α_R



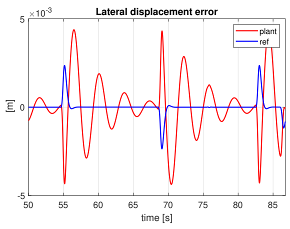
(a) $\alpha_{X1} = 1$



(b) $\alpha_{X1} = 0.7$

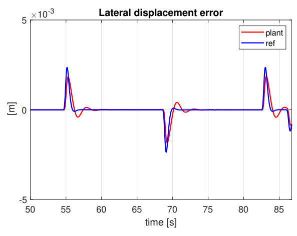


(c) $\alpha_{X1} = 0.3$

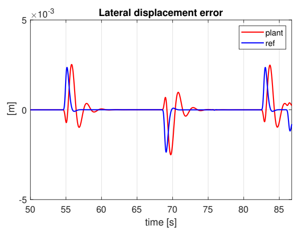


(d) $\alpha_{X1} = 0.1$

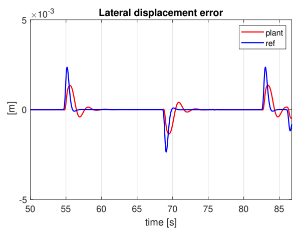
Figure B.1: Lateral displacement error x_3 in the 3^{rd} lap, tuning α_{X1}



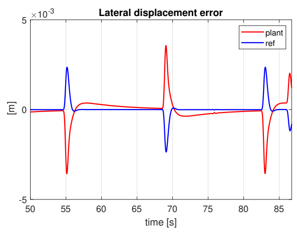
(a) $\alpha_R = 1e-2$



(b) $\alpha_R = 2e-2$



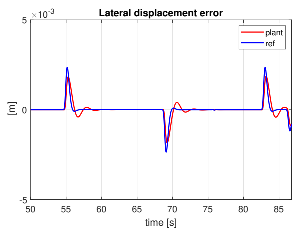
(c) $\alpha_R = 1.2e-2$



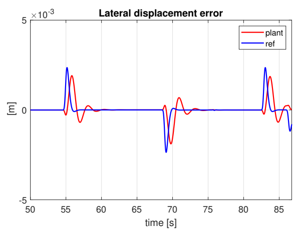
(d) $\alpha_R = 5e-5$

Figure B.2: Lateral displacement error x_3 in the 3^{rd} lap, tuning α_R

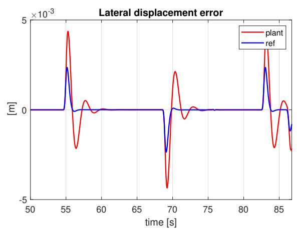
Tuning of Q



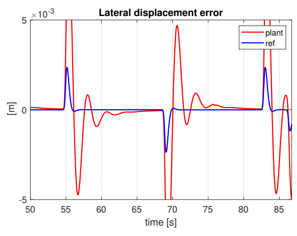
(a) $q_3 = 1e4$



(b) $q_3 = 1.5e4$



(c) $q_3 = 5e3$

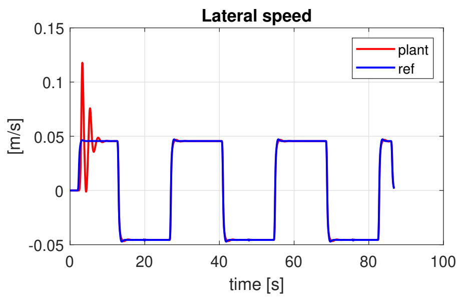


(d) $q_3 = 0.8e3$

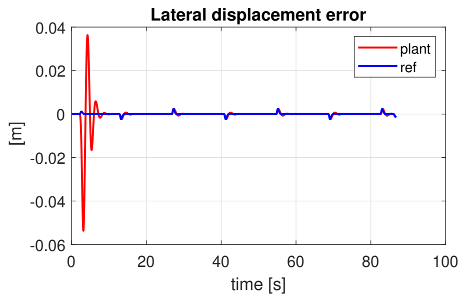
Figure B.3: Lateral displacement error x_3 in the 3^{rd} lap, tuning q_3

B.2 EMRAC

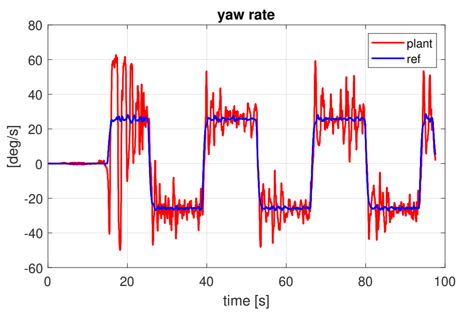
States tracking



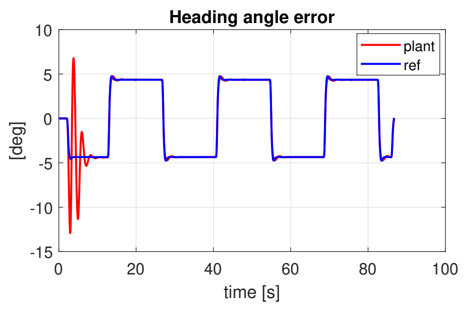
(a) Lateral speed x_1



(b) Lateral displacement error x_3



(c) Yaw rate x_2



(d) Heading angle error x_4

Figure B.4: EMRAC: states tracking

Parameter projection

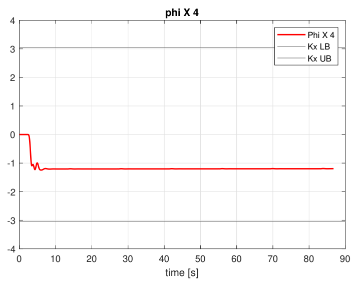
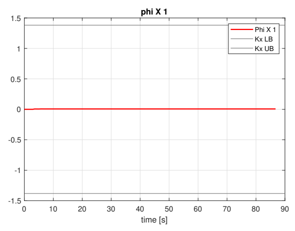
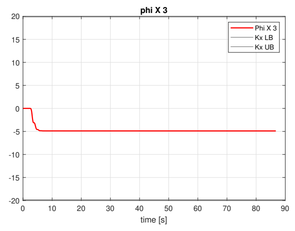


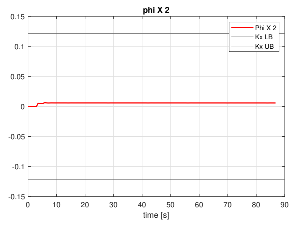
Figure B.5: EMRAC: parameter projection on Φ_R



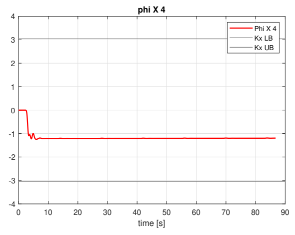
(a) ϕ_{X_1}



(b) ϕ_{X_3}



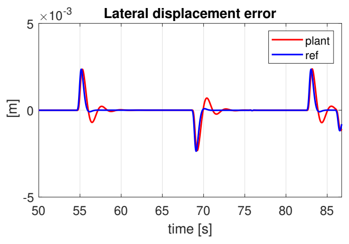
(c) ϕ_{X_2}



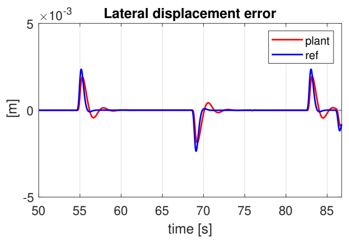
(d) ϕ_{X_4}

Figure B.6: EMRAC: parameter projection on Φ_X

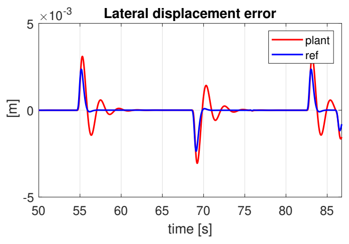
α_N and α_I tuning



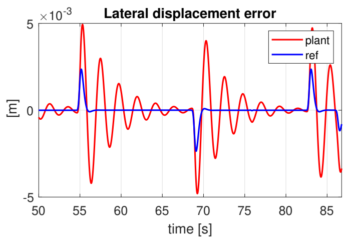
(a) $\alpha_N = 5e-3$



(b) $\alpha_N = 5e-4$

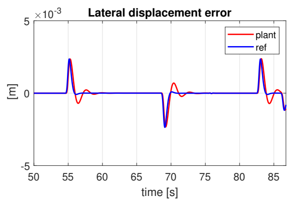


(c) $\alpha_N = 5e-2$

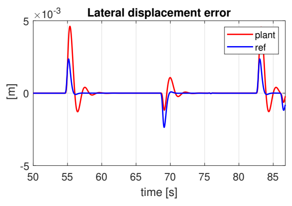


(d) $\alpha_N = 5e-1$

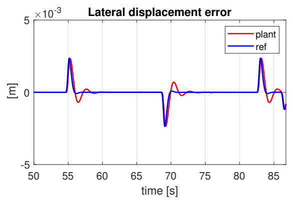
Figure B.7: Lateral displacement error x_3 in the 3^{rd} lap, tuning α_N



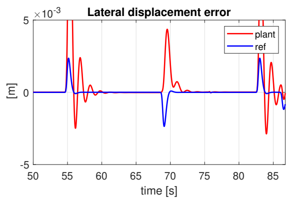
(a) α_I scaled by 1e0



(b) α_I scaled by 1e2

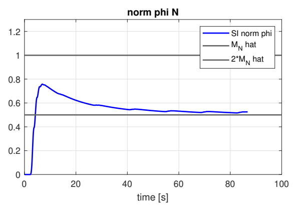


(c) α_I scaled by 1e-1

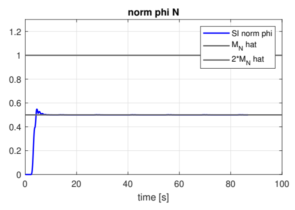


(d) α_I scaled by 1e+3

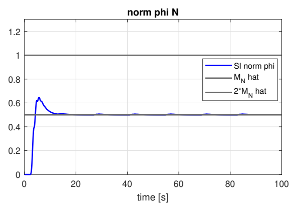
Figure B.8: Lateral displacement error x_3 in the 3^{rd} lap, scaling α_I



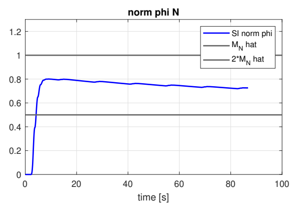
(a) $\rho_{N0} = 5e - 2$



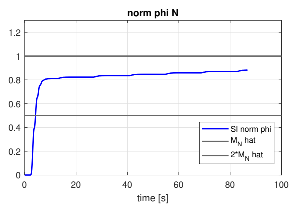
(b) $\rho_{N0} = 5e0$



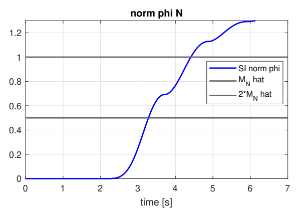
(c) $\rho_{N0} = 5e - 1$



(d) $\rho_{N0} = 5e - 3$



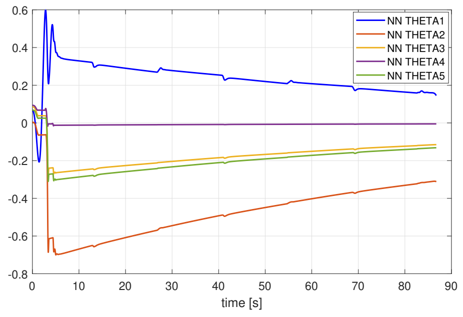
(e) $\rho_{N0} = 5e - 5$



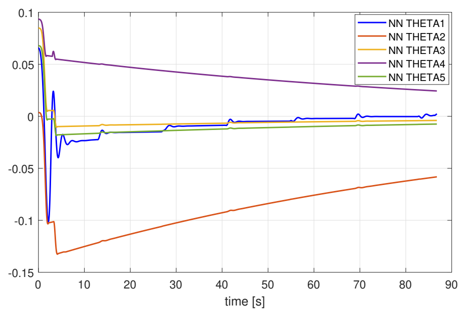
(f) $\rho_{N0} = 5e - 7$

Figure B.9: Effect of the discharge factor ρ_{N0} on Φ_N

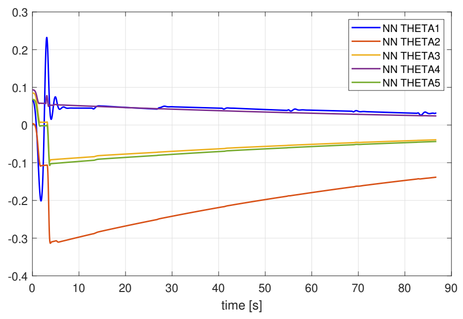
B.3 EMRAC-NN



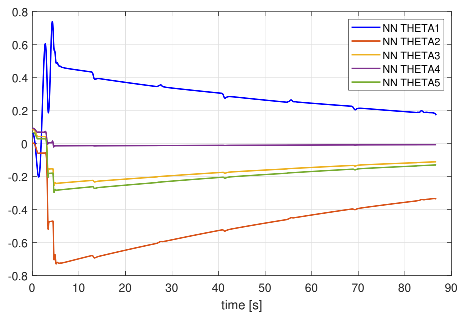
(a) Learning rate = 0.7



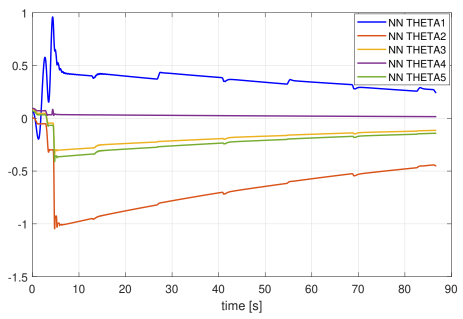
(b) Learning rate = 0.1



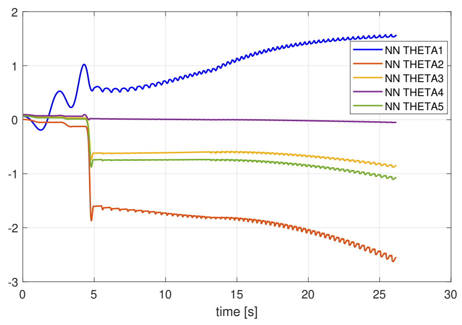
(c) Learning rate = 0.3



(d) Learning rate = 0.8



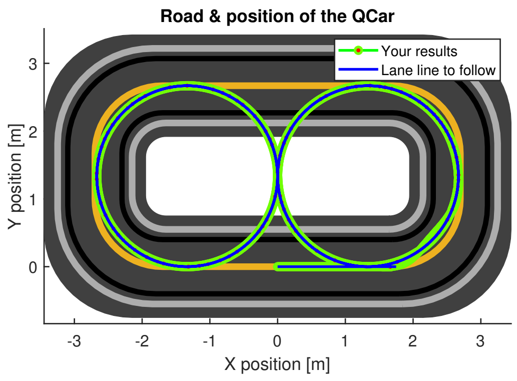
(e) Learning rate = 0.9



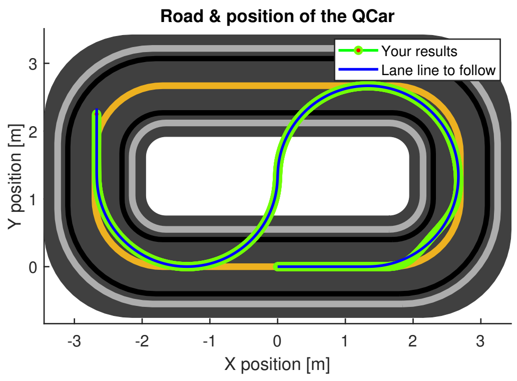
(f) Learning rate = 1

Figure B.10: Effect of the learning rate on the NN weights adaptation

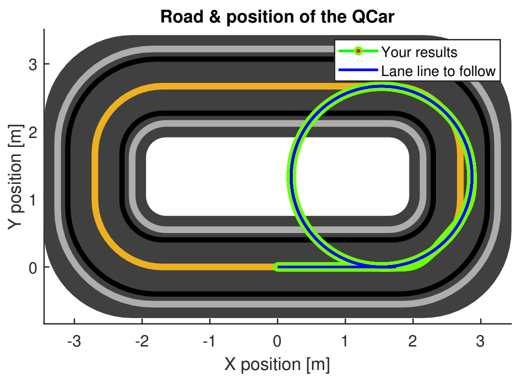
B.4 EMRAC, trajectories



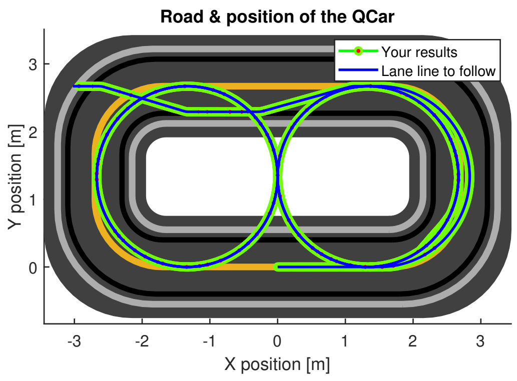
(a) EMRAC, eight trajectory



(b) EMRAC, S trajectory



(c) EMRAC, O trajectory



(d) EMRAC, Obstacle Avoidance trajectory

Figure B.11: Path tracking for differnt trajectories, EMRAC algorithm

B.5 KPIs at different velocities

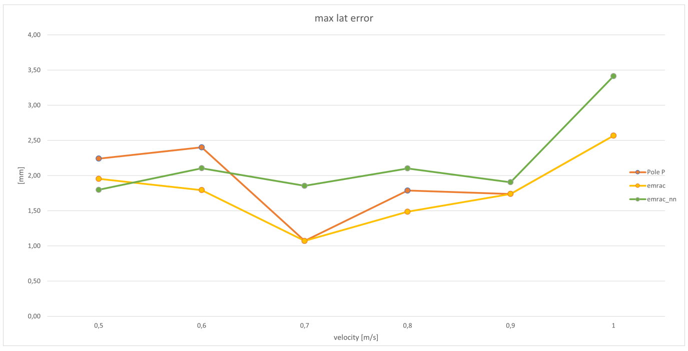


Figure B.12: Maximum lateral displacement error, 3rd lap

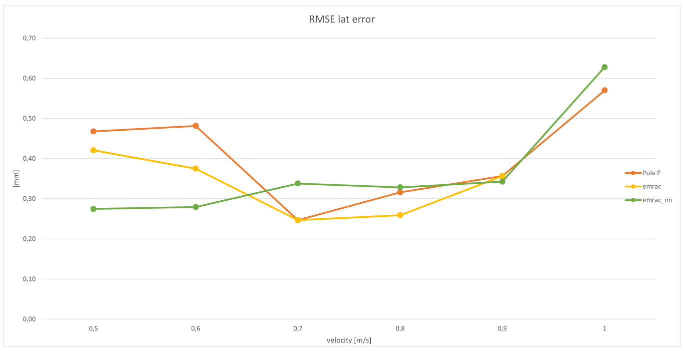


Figure B.13: RMSE lateral displacement error, 3rd lap

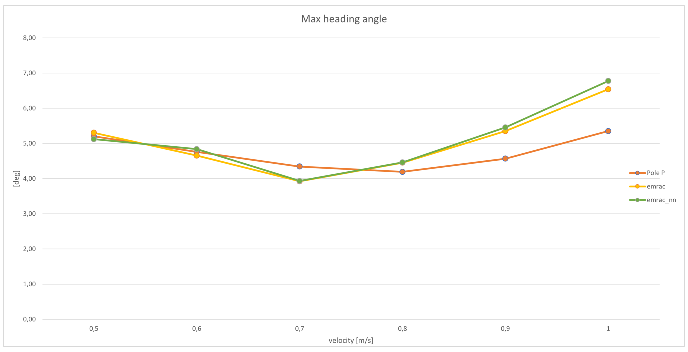


Figure B.14: Maximum heading angle error, 3rd lap

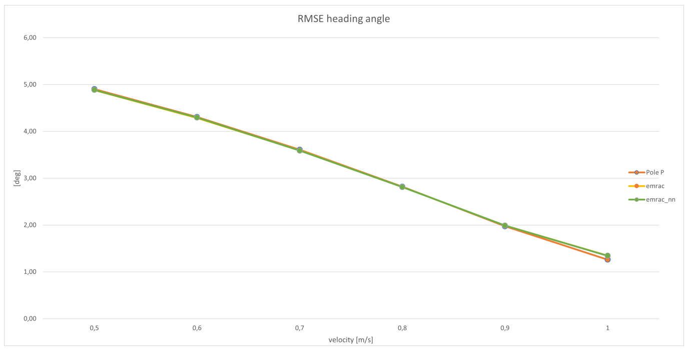


Figure B.15: RMSE heading angle error, 3rd lap

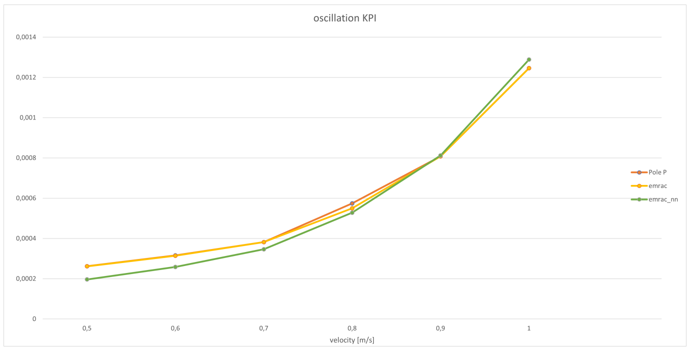
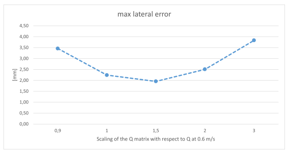
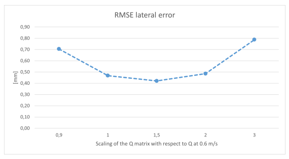


Figure B.16: Control action oscillation KPI, 3rd lap

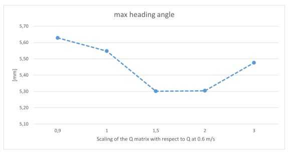
B.6 Q matrix and performance



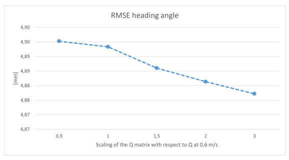
(a)



(b)



(c)



(d)

Figure B.17: Tuning based on the Q matrix

Appendix C

EMRAC Experimental figures

C.1 EMRAC

C.1.1 Tuning Parameters

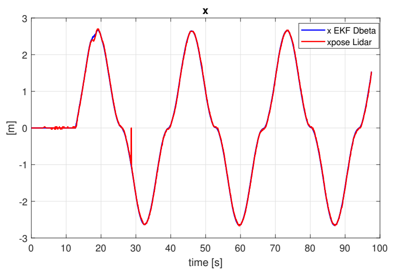
$$\alpha_X = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \beta_X = \begin{bmatrix} 0.005 & 0 & 0 & 0 \\ 0 & 2.50 & 0 & 0 \\ 0 & 0 & 2.50 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

$$\alpha_R = \begin{bmatrix} 0.1 \end{bmatrix}, \quad \beta_R = \begin{bmatrix} 0.05 \end{bmatrix}$$

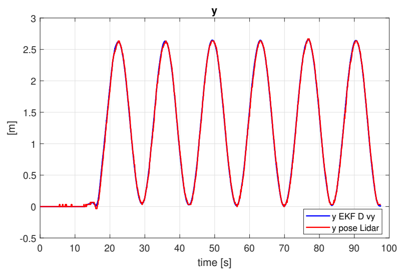
$$\alpha_I = \begin{bmatrix} \text{1e-05} & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.06 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}, \quad \beta_I = \begin{bmatrix} \text{5e-06} & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.03 & 0 \\ 0 & 0 & 0 & 0.005 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1.1 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 \\ 0 & 0 & 825 & 0 \\ 0 & 0 & 0 & 1.1 \end{bmatrix}$$

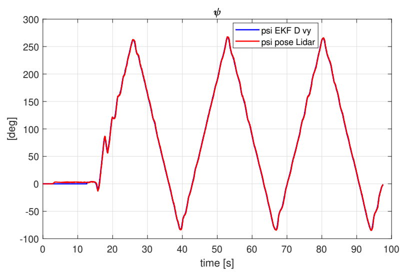
C.1.2 Vehicle Localization



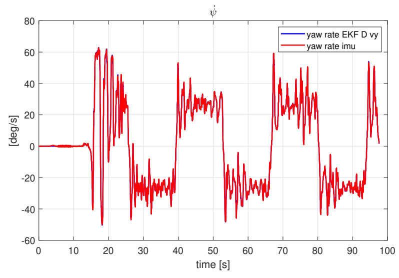
(a) x coordinate



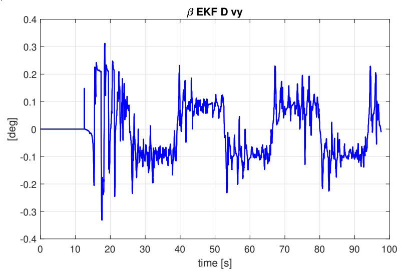
(b) y coordinate



(c) yaw angle ψ



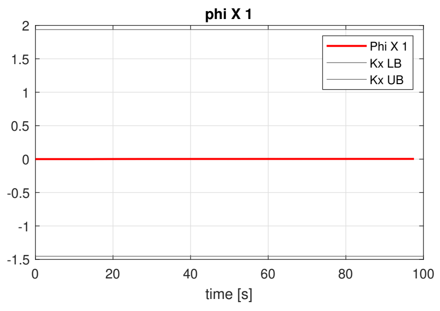
(d) yaw rate $\dot{\psi}$



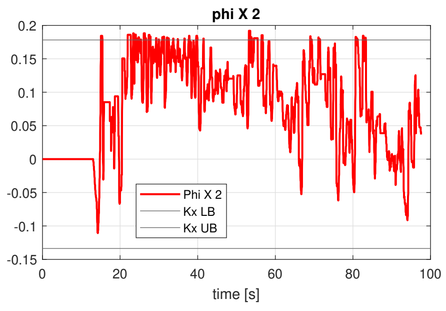
(e) side slip angle β

Figure C.1: Vehicle localization, EMRAC

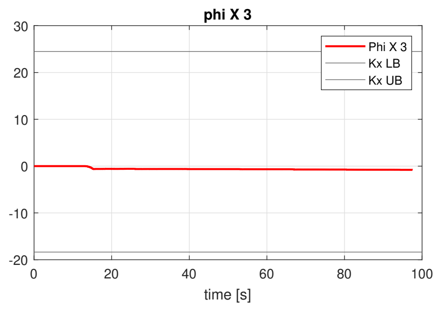
C.1.3 Parameter Projection



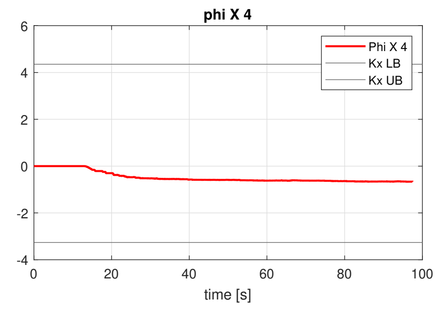
(a) ϕ_{X_1}



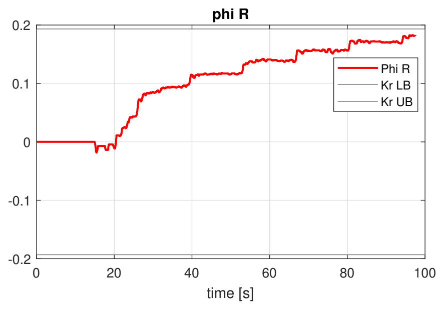
(b) ϕ_{X_2}



(c) ϕ_{X_3}



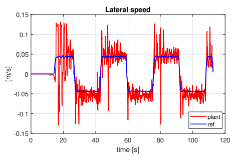
(d) ϕ_{X_4}



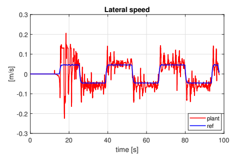
(e) ϕ_R

Figure C.2: Parameter Projection, EMRAC

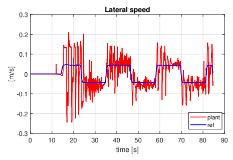
C.1.4 States tracking for all velocities



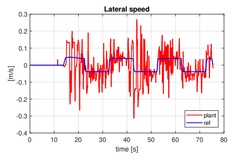
(a) 0.5 m/s



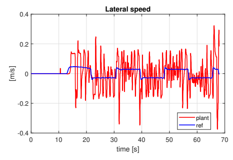
(b) 0.6 m/s



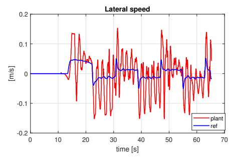
(c) 0.7 m/s



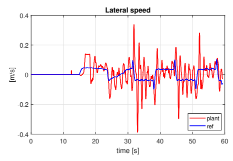
(d) 0.8 m/s



(e) 0.9 m/s

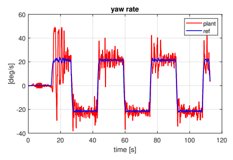


(f) 1.0 m/s

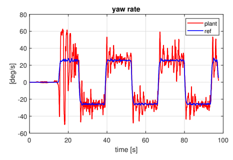


(g) 1.2 m/s

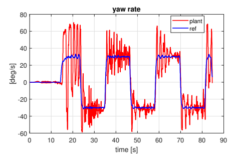
Figure C.3: Lateral speed tracking



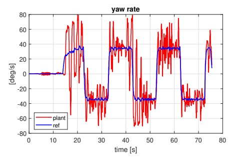
(a) 0.5 m/s



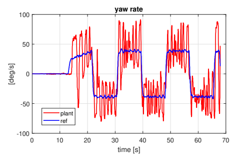
(b) 0.6 m/s



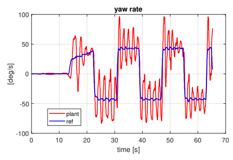
(c) 0.7 m/s



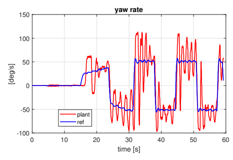
(d) 0.8 m/s



(e) 0.9 m/s

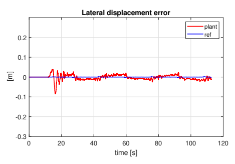


(f) 1.0 m/s

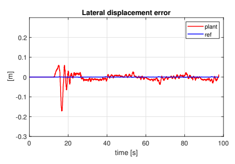


(g) 1.2 m/s

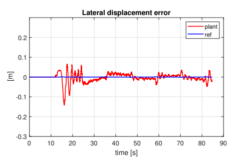
Figure C.4: Yaw rate tracking



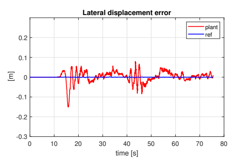
(a) 0.5 m/s



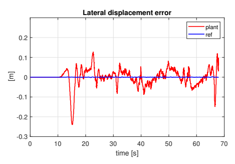
(b) 0.6 m/s



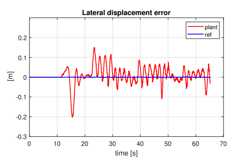
(c) 0.7 m/s



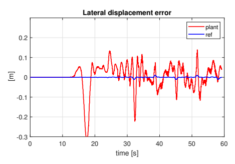
(d) 0.8 m/s



(e) 0.9 m/s

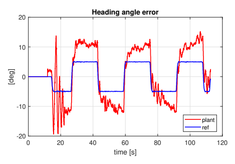


(f) 1.0 m/s

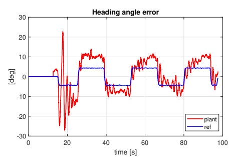


(g) 1.2 m/s

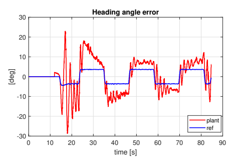
Figure C.5: Lateral displacement error evolution at different velocities



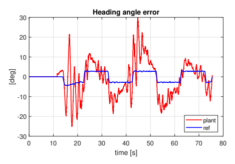
(a) 0.5 m/s



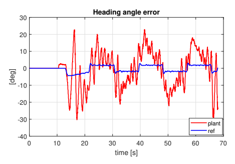
(b) 0.6 m/s



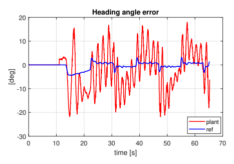
(c) 0.7 m/s



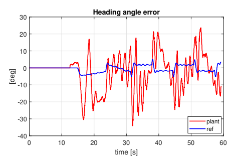
(d) 0.8 m/s



(e) 0.9 m/s



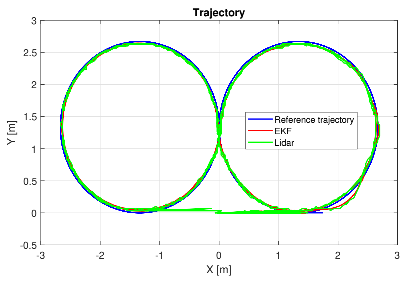
(f) 1.0 m/s



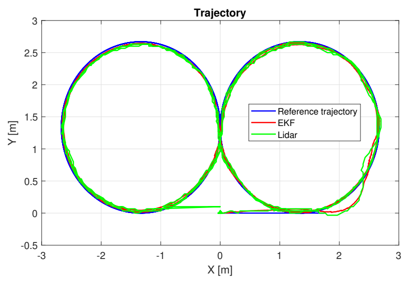
(g) 1.2 m/s

Figure C.6: Heading angle error

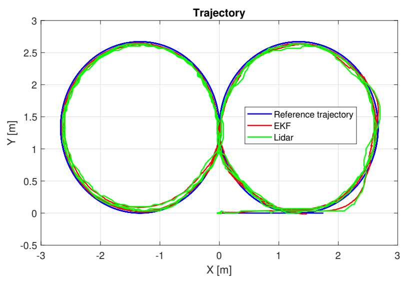
C.1.5 Trajectory at different velocities



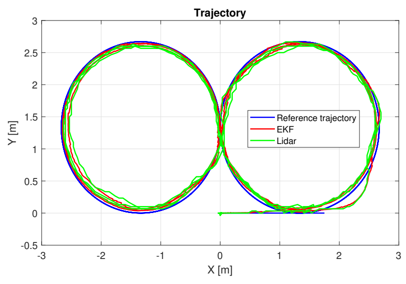
(a) 0.5 m/s



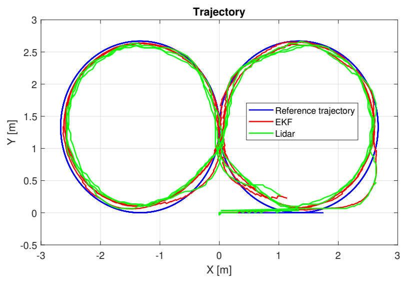
(b) 0.6 m/s



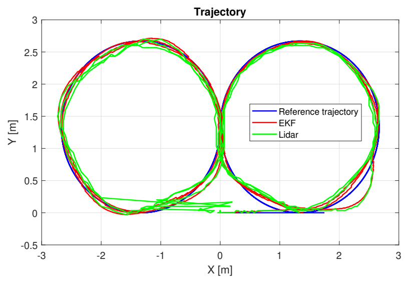
(c) 0.7 m/s



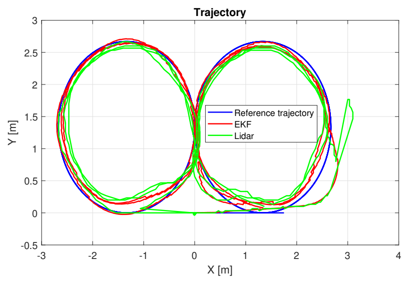
(d) 0.8 m/s



(e) 0.9 m/s



(f) 1 m/s

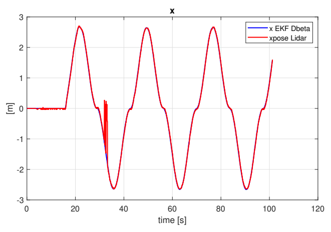


(g) 1.2 m/s

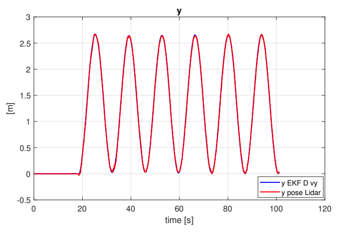
Figure C.7: Trajectory at different velocities

C.2 EMRAC-NN

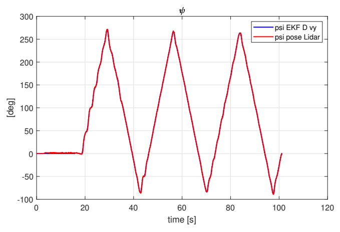
C.2.1 Vehicle Localization



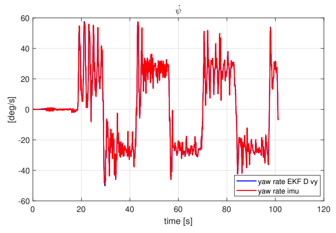
(a) x coordinate



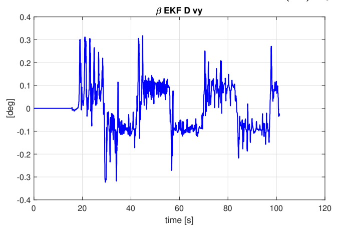
(b) y coordinate



(c) yaw angle ψ



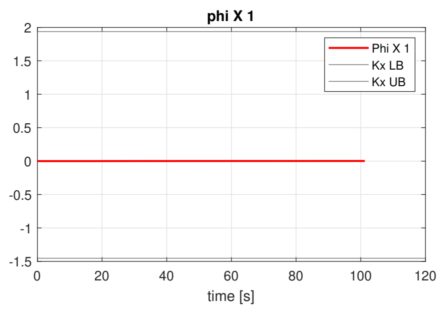
(d) yaw rate $\dot{\psi}$



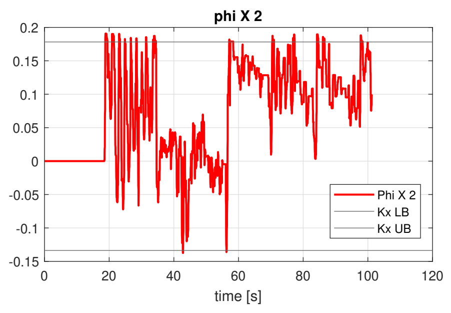
(e) side slip angle β

Figure C.8: Vechicle Localization, EMRAC-NN

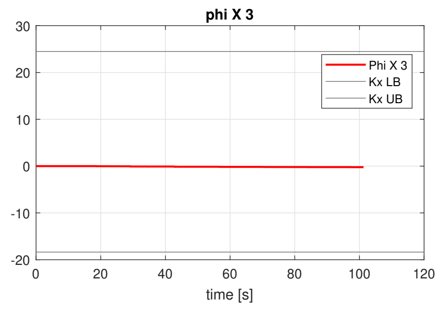
C.2.2 Parameter Projection



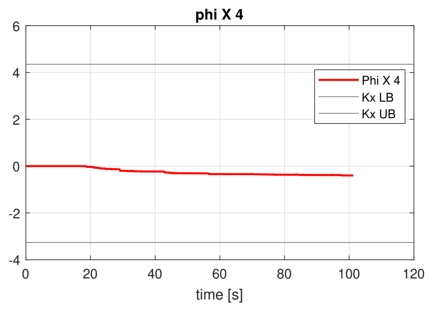
(a) ϕ_{X_1}



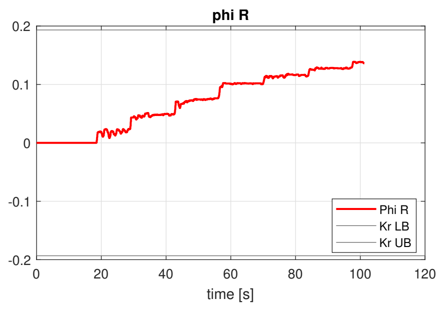
(b) ϕ_{X_2}



(c) ϕ_{X_3}



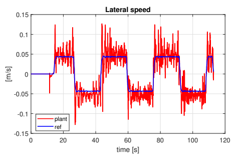
(d) ϕ_{X_4}



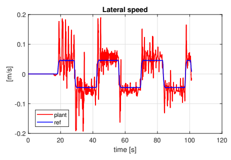
(e) ϕ_R

Figure C.9: Parameter Porjection, EMRAC-NN

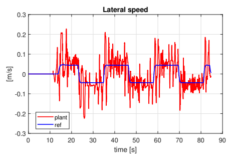
C.2.3 States tracking for all velocities



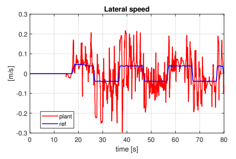
(a) 0.5 m/s



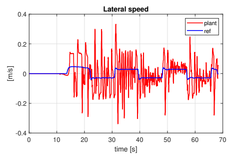
(b) 0.6 m/s



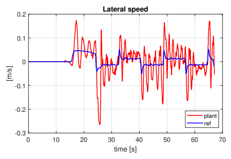
(c) 0.7 m/s



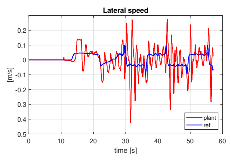
(d) 0.8 m/s



(e) 0.9 m/s

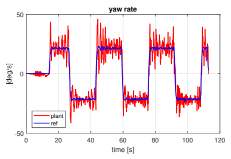


(f) 1.0 m/s

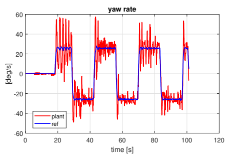


(g) 1.2 m/s

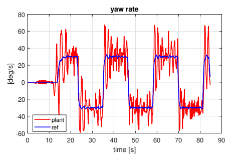
Figure C.10: Lateral speed tracking



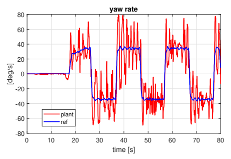
(a) 0.5 m/s



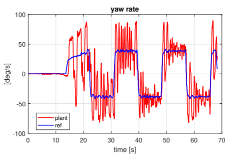
(b) 0.6 m/s



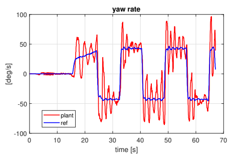
(c) 0.7 m/s



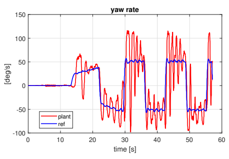
(d) 0.8 m/s



(e) 0.9 m/s

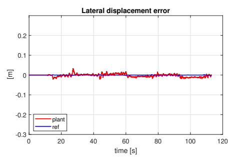


(f) 1.0 m/s

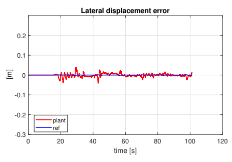


(g) 1.2 m/s

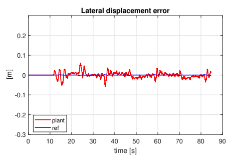
Figure C.11: Yaw rate tracking



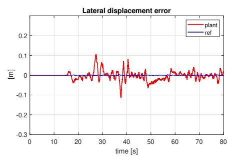
(a) 0.5 m/s



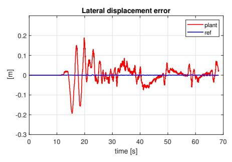
(b) 0.6 m/s



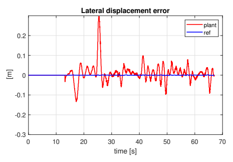
(c) 0.7 m/s



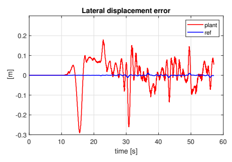
(d) 0.8 m/s



(e) 0.9 m/s

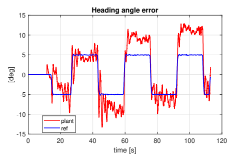


(f) 1.0 m/s

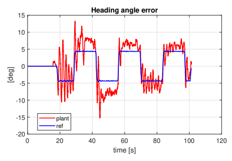


(g) 1.2 m/s

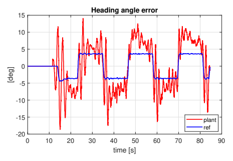
Figure C.12: Lateral displacement error evolution at different velocities



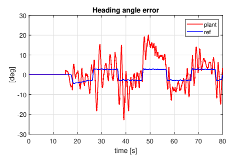
(a) 0.5 m/s



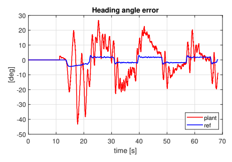
(b) 0.6 m/s



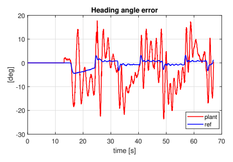
(c) 0.7 m/s



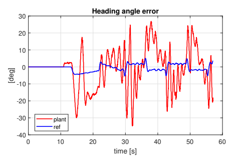
(d) 0.8 m/s



(e) 0.9 m/s



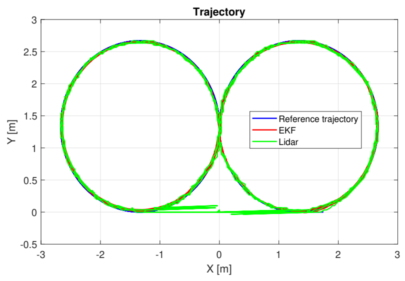
(f) 1.0 m/s



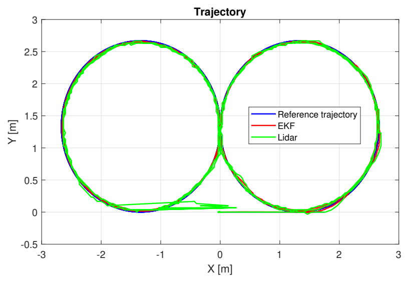
(g) 1.2 m/s

Figure C.13: Heading angle error

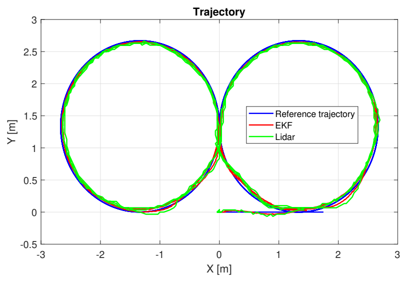
C.2.4 Trajectory at different velocities



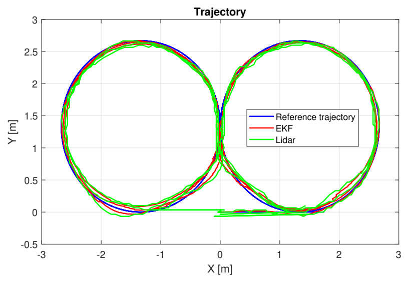
(a) 0.5 m/s



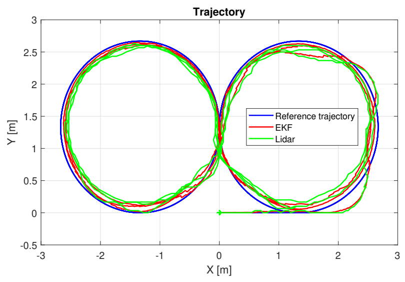
(b) 0.6 m/s



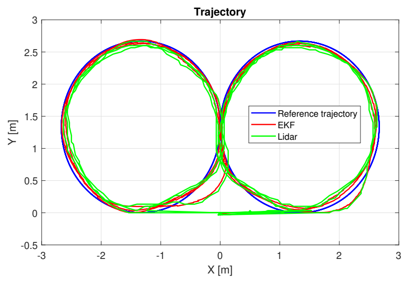
(c) 0.7 m/s



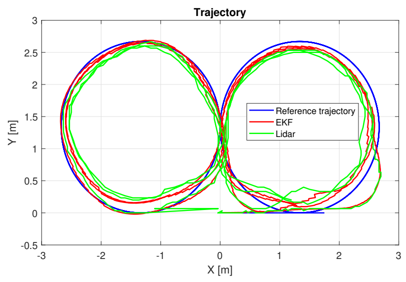
(d) 0.8 m/s



(e) 0.9 m/s



(f) 1 m/s



(g) 1.2 m/s

Figure C.14: Trajectory at different velocities

C.3 KPIs graphs, 2nd lap

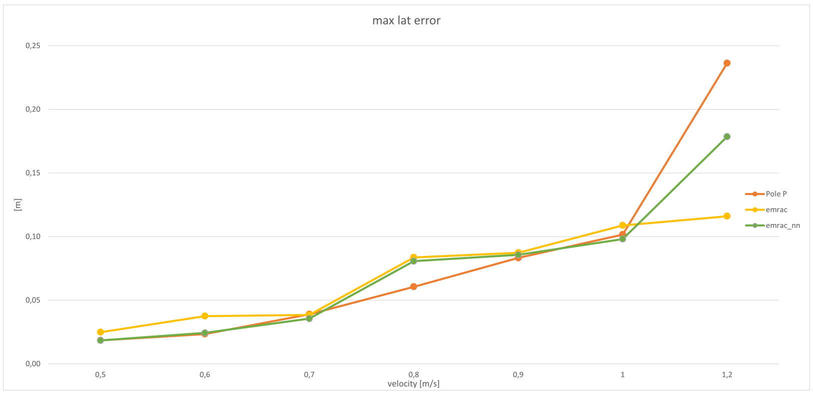


Figure C.15: Maximum lateral displacement error, 2nd lap

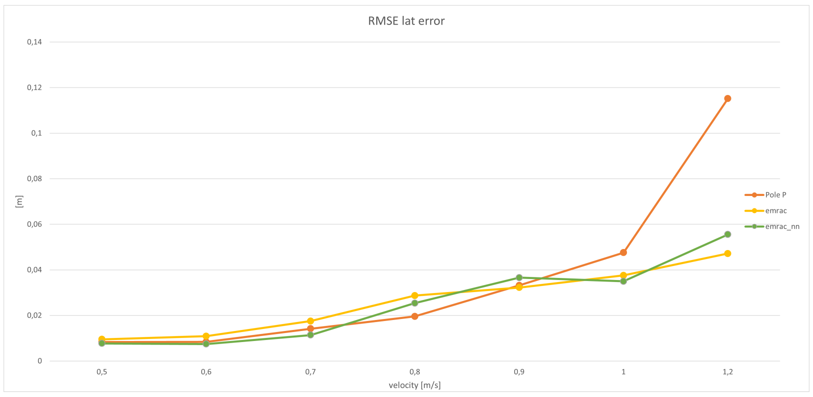


Figure C.16: RMSE lateral displacement error, 2nd lap

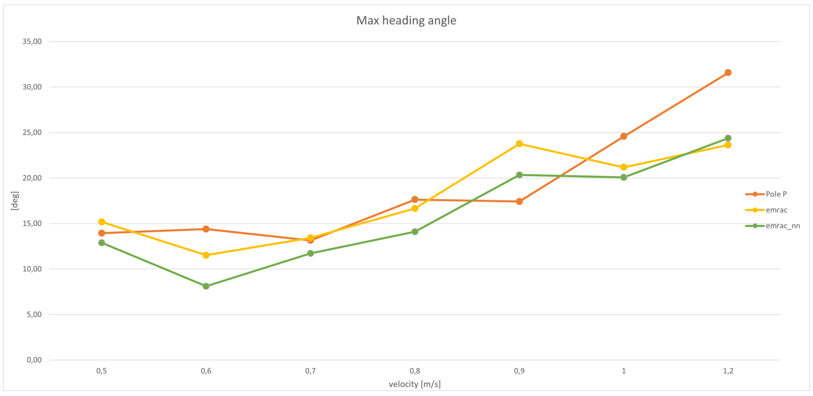


Figure C.17: Maximum heading angle error, 2nd lap

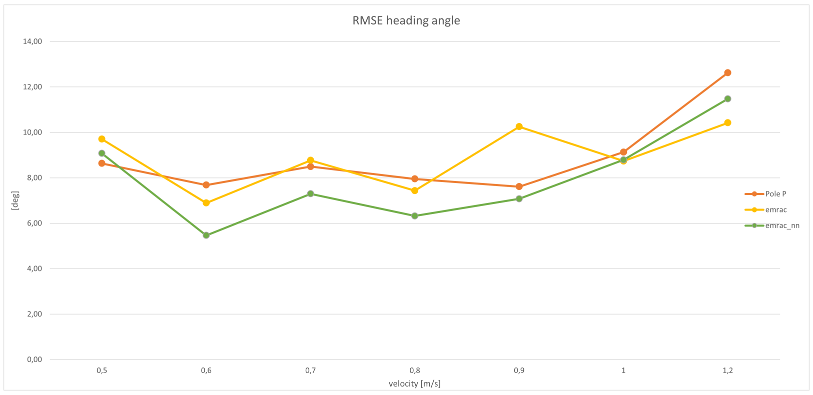


Figure C.18: RMSE heading angle error, 2nd lap

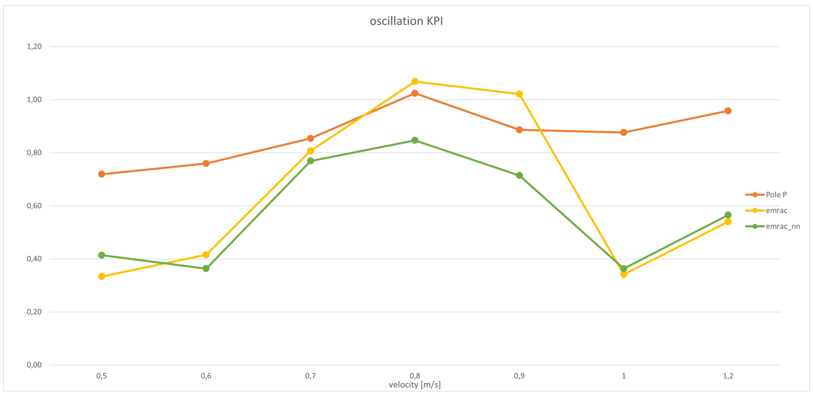


Figure C.19: Control action oscillation KPI, 2nd lap

Appendix D

Experimental KPIs

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.021	0.085	0.033
RMSE lat.err [m]	0.006	0.020	0.007
Max head.angle err [deg]	12.442	19.287	13.281
RMSE head. angle err [deg]	7.000	10.203	4.091
IACA control input [deg]	11.104	12.259	9.668
Oscillation KPI	0.638	0.468	0.294

Table D.1: KPIs for Lap 1 at velocity 0.5 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.019	0.025	0.018
RMSE lat.err [m]	0.008	0.009	0.008
Max head.angle err [deg]	12.806	12.637	11.897
RMSE head. angle err [deg]	8.792	9.142	8.384
IACA control input [deg]	11.940	12.194	12.200
Oscillation KPI	0.671	0.291	0.445

Table D.2: KPIs for Lap 2 at velocity 0.5 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.023	0.023	0.019
RMSE lat.err [m]	0.009	0.011	0.009
Max head.angle err [deg]	13.948	15.176	12.886
RMSE head. angle err [deg]	8.636	9.703	9.076
IACA control input [deg]	11.850	12.283	12.131
Oscillation KPI	0.719	0.334	0.414

Table D.3: KPIs for Lap 3 at velocity 0.5 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.027	0.171	0.042
RMSE lat.err [m]	0.008	0.035	0.010
Max head.angle err [deg]	13.569	27.103	15.283
RMSE head. angle err [deg]	6.594	11.050	5.321
IACA control input [deg]	11.339	14.092	9.540
Oscillation KPI	0.723	0.653	0.561

Table D.4: KPIs for Lap 1 at velocity 0.6 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.023	0.038	0.024
RMSE lat.err [m]	0.008	0.011	0.007
Max head.angle err [deg]	12.387	11.575	10.656
RMSE head. angle err [deg]	7.376	7.279	6.711
IACA control input [deg]	12.156	12.632	12.561
Oscillation KPI	0.669	0.416	0.397

Table D.5: KPIs for Lap 2 at velocity 0.6 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.026	0.024	0.024
RMSE lat.err [m]	0.009	0.009	0.005
Max head.angle err [deg]	14.385	11.520	8.106
RMSE head. angle err [deg]	7.685	6.894	5.466
IACA control input [deg]	12.060	12.541	12.634
Oscillation KPI	0.759	0.415	0.364

Table D.6: KPIs for Lap 3 at velocity 0.6 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.030	0.142	0.061
RMSE lat.err [m]	0.011	0.047	0.017
Max head.angle err [deg]	13.036	28.735	18.654
RMSE head. angle err [deg]	7.316	12.860	6.158
IACA control input [deg]	11.479	16.941	11.234
Oscillation KPI	0.881	1.372	0.767

Table D.7: KPIs for Lap 1 at velocity 0.7 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.039	0.038	0.035
RMSE lat.err [m]	0.014	0.018	0.011
Max head.angle err [deg]	16.109	16.731	16.362
RMSE head. angle err [deg]	8.209	10.407	7.463
IACA control input [deg]	12.265	13.467	13.506
Oscillation KPI	0.843	0.945	0.970

Table D.8: KPIs for Lap 2 at velocity 0.7 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.039	0.028	0.024
RMSE lat.err [m]	0.015	0.014	0.010
Max head.angle err [deg]	13.142	13.425	11.726
RMSE head. angle err [deg]	8.498	8.771	7.297
IACA control input [deg]	12.117	13.133	13.101
Oscillation KPI	0.854	0.807	0.769

Table D.9: KPIs for Lap 3 at velocity 0.7 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.056	0.151	0.113
RMSE lat.err [m]	0.019	0.038	0.032
Max head.angle err [deg]	17.703	25.096	18.306
RMSE head. angle err [deg]	7.083	9.684	6.122
IACA control input [deg]	11.696	13.823	15.398
Oscillation KPI	1.060	1.173	0.898

Table D.10: KPIs for Lap 1 at velocity 0.8 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.061	0.084	0.081
RMSE lat.err [m]	0.020	0.029	0.025
Max head.angle err [deg]	16.464	29.828	22.789
RMSE head. angle err [deg]	7.713	11.279	10.338
IACA control input [deg]	12.368	16.636	14.712
Oscillation KPI	0.874	1.351	1.277

Table D.11: KPIs for Lap 2 at velocity 0.8 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.052	0.046	0.040
RMSE lat.err [m]	0.021	0.016	0.012
Max head.angle err [deg]	17.638	16.656	14.092
RMSE head. angle err [deg]	7.953	7.434	6.324
IACA control input [deg]	12.383	13.490	14.084
Oscillation KPI	1.024	1.068	0.847

Table D.12: KPIs for Lap 3 at velocity 0.8 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.087	0.240	0.193
RMSE lat.err [m]	0.032	0.064	0.070
Max head.angle err [deg]	23.235	30.358	43.255
RMSE head. angle err [deg]	8.678	11.927	14.928
IACA control input [deg]	11.859	15.383	16.995
Oscillation KPI	1.114	0.832	0.932

Table D.13: KPIs for Lap 1 at velocity 0.9 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.083	0.087	0.086
RMSE lat.err [m]	0.033	0.032	0.037
Max head.angle err [deg]	24.065	22.841	22.621
RMSE head. angle err [deg]	9.227	10.475	12.301
IACA control input [deg]	12.709	14.625	14.911
Oscillation KPI	0.936	1.165	1.193

Table D.14: KPIs for Lap 2 at velocity 0.9 m/s, eight trajectory

KPI	**PP with LQR**	**EMRAC**	**EMRAC-NN**
Max lat.err [m]	0.073	0.066	0.060
RMSE lat.err [m]	0.026	0.034	0.019
Max head.angle err [deg]	17.425	23.768	20.344
RMSE head. angle err [deg]	7.610	10.253	7.079
IACA control input [deg]	12.846	14.489	14.050
Oscillation KPI	0.886	1.021	0.714

Table D.15: KPIs for Lap 3 at velocity 0.9 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.109	0.202	0.297
RMSE lat.err [m]	0.039	0.066	0.067
Max head.angle err [deg]	23.917	21.697	22.461
RMSE head. angle err [deg]	8.803	10.155	9.077
IACA control input [deg]	12.058	13.379	13.118
Oscillation KPI	0.945	0.319	0.300

Table D.16: KPIs for Lap 1 at velocity 1 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.102	0.109	0.098
RMSE lat.err [m]	0.048	0.038	0.035
Max head.angle err [deg]	22.121	21.818	23.588
RMSE head. angle err [deg]	10.254	8.866	9.105
IACA control input [deg]	12.882	14.251	14.128
Oscillation KPI	0.943	0.420	0.407

Table D.17: KPIs for Lap 2 at velocity 1 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.088	0.075	0.079
RMSE lat.err [m]	0.039	0.026	0.025
Max head.angle err [deg]	24.573	21.184	20.075
RMSE head. angle err [deg]	9.132	8.739	8.791
IACA control input [deg]	13.358	13.412	13.516
Oscillation KPI	0.876	0.342	0.363

Table D.18: KPIs for Lap 3 at velocity 1 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.229	0.320	0.291
RMSE lat.err [m]	0.074	0.102	0.098
Max head.angle err [deg]	28.661	34.034	30.174
RMSE head. angle err [deg]	9.972	14.083	12.453
IACA control input [deg]	12.594	14.475	12.709
Oscillation KPI	0.962	0.351	0.291

Table D.19: KPIs for Lap 1 at velocity 1.2 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.236	0.116	0.179
RMSE lat.err [m]	0.115	0.047	0.056
Max head.angle err [deg]	32.317	28.034	34.791
RMSE head. angle err [deg]	14.999	12.119	14.495
IACA control input [deg]	14.455	14.890	15.851
Oscillation KPI	0.949	0.557	0.709

Table D.20: KPIs for Lap 2 at velocity 1.2 m/s, eight trajectory

KPI	PP with LQR	EMRAC	EMRAC-NN
Max lat.err [m]	0.203	0.139	0.145
RMSE lat.err [m]	0.091	0.051	0.051
Max head.angle err [deg]	31.582	23.641	24.384
RMSE head. angle err [deg]	12.627	10.422	11.477
IACA control input [deg]	14.401	14.555	14.614
Oscillation KPI	0.958	0.540	0.566

Table D.21: KPIs for Lap 3 at velocity 1.2 m/s, eight trajectory

Bibliography

[1] R. Bacon, "Roger bacon," XIII. https://en.wikipedia.org/wiki/Roger_Bacon.

[2] PwC, "Five trends transforming the automotive industry," 2017. https://www.pwc.com/gx/en/industries/automotive.html.

[3] K. Ogata, *Modern Control Engineering.* Pearson College, 2009.

[4] S. Dixit, U. Montanaro, M. Dianati, A. Mouzakitis, and S. Fallah, "Integral mrac with bounded switching gain for vehicle lateral tracking," *IEEE Transactions on Control Systems Technology*, 2021.

[5] U. Montanaro, S. Martini, Z. Hao, Y. Gao, and A. Sorniotti, "Multi-input enhanced model reference adaptive control strategies and their application to space robotic manipulators," *Wiley*, 2022.

[6] U. Montanaro and J. M.Olm, "Integral mrac wit hminimal controller synthesis and bounded adaptive gains:the continuous-timecase," *Journal oftheFranklinInstitute*, 2016.

[7] U. Montanaro and J. M.Olm, "Discrete-time integral mrac with minimal controller synthesis and parameter projection," *Journal oftheFranklinInstitute*, 2015.

[8] M. di Bernardo, A. di Gaeta, U. Montanaro, J. M. Olm, and S. Santini, "Discrete-time mrac with minimal controller synthesis of an electronic throttle body," *The International Federation of Automatic Control*, 2011.

[9] N. Najva and A. Saleem, "Model reference controller approach for robot arm tracking using neural networks," *Indian Journal of Science and Technology*, 2019.

[10] H. Fu, X. Chen, W. Wang, and M. Wu, "Mrac for unknown discrete-time nonlinear systems based on supervised neural dynamic programming," *Neurocomputing*, 2019.

[11] J. Girish and C. Girish, "Deep model reference adaptive control," *Palais des Congrès et des Expositions Nice Acropolis*, 2019.

[12] N. T. Nguyen, *Model-Reference Adaptive Control.* Springer, 2017.

[13] M. Guiggiani, *The Science of Vehicle Dynamics*, ch. 3. Springer, 2014.

[14] M. Guiggiani, *The Science of Vehicle Dynamics*, ch. 6. Springer, 2014.

[15] R. Rajamani, *Vehicle Dynamics and Control*, ch. 2. Springer, 2011.

[16] C. Klauer, M. Schwabe, and H. Mobalegh, "Path tracking control for urban autonomous driving," *International Federation of Automatic Control (IFAC)*, 2020.

[17] M. Works, "Trapezoidal numerical integration," 2023. https://it.mathworks.com/help/matlab/ref/trapz.html.

[18] M. Works, "Differences and approximate derivatives," 2023. https://it.mathworks.com/help/matlab/ref/diff.html.

[19] M. Works, "1-d data interpolation," 2023. https://it.mathworks.com/help/matlab/ref/interp1.html#d126e830066.

[20] M. Works, "Smooth response data," 2023. https://it.mathworks.com/help/curvefit/smooth.html.

[21] R. S. Sharp, D. Casanova, and P. Symonds, *A mathematical model for driver steering control, with design, tuning and performance results*, vol. 33. Vehicle system dynamics, 2000.

[22] Quanser, "Qcar," 2023. https://www.quanser.com/products/qcar/.

[23] Quanser, 2021. 119 Spy Court Markham, Ontario, L3R 5H6, Canada.

[24] D. Ragone, "Modelling, estimation, and control of scaled fully autonomous vehicles," 2022.

[25] G. Scattolini, "Positioning and path tracking control strategies experimentally validated through fully autonomous scaled vehicle," 2023.

[26] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," 2009.