SEBHA UNIVERSITY CONFERENCE PROCEEDINGS, VOL.04 NO. 1 2025
DOI: 10.51984/SUCP.V4I1.3956

المؤتمر العلمي الأول للتطبيقات الهندسية (ICEA'2024)

The First Scientific Conference on Engineering Applications (ICEA'2024)

Conference homepage: www.icea.ly

Adaptive Control Approach for Optimized Lane Keeping in Autonomous Vehicles

*Ahmed J. Abougarair[a], Ibrahim B. Attawil[b], Abdusalam A Alaktiwi[c] , Mohsen Bakouri[d]

[a]Electrical and Electronics Engineering,University of Tripoli,Tripoli, Libya
[b]Electrical and Electronics Engineering,University of Tripoli,Tripoli, Libya
[c]Biomedical Engineering, University of Tripoli,Tripoli, Libya
[d]Department of Physics, College of Arts, Fezzan University, Traghen 71340, Libya

Keywords:
Adaptive MPC
Lateral Equations
Cost Function
Tracking Term
Path Following Control

ABSTRACT

The section describes a study on the application of Adaptive Model Predictive Control (AMPC) for lane tracking in autonomous vehicles. The study utilizes a 3 Degree-of-Freedom bicycle model and incorporates dynamic adjustments through sensor fusion to enhance the steering control system, which is crucial for autonomous vehicles. The research focuses on the systematic tuning of control weights and real-time adjustments to demonstrate AMPC's effectiveness in various driving scenarios. Simulation results validate the accuracy, safety, and reliability of AMPC, with ongoing objectives including further development and evaluation of sensor fusion, particularly using depth cameras. The use of Unreal Engine 3D simulation highlights the potential for real-time implementation of AMPC in autonomous driving and lane following. Continuous advancements in control algorithms and sensor technologies are expected to improve AMPC performance further.

نهج التحكم التكيفي للحفاظ على المسار الأمثل في المركبات ذاتية القيادة

*أحمد جابر أبوجرير [1] و ابراهيم الطويل [2] و عبد السلام الغديوى [3] و محسن بكورى [4]

[1] قسم الهندسة الكهربائية والالكترونية، جامعة طرابلس، طرابلس، ليبيا
[2] قسم الهندسة الكهربائية والالكترونية، جامعة طرابلس، طرابلس، ليبيا
[3] قسم الهندسة الطبية، جامعة طرابلس، طرابلس، ليبيا
[4] قسم الفيزياء، كلية التربية ، جامعة فزان، تراغن، ليبيا

الكلمات المفتاحية:
MPC التكيفي
معادلات الحركة
دالة التكلفة.
تتبع المسار
التحكم فى تحديد المسار

الملخص

تقدم هذه الورقة حلاً لنظام التحكم في التوجيه، والذي يلعب دورًا حاسمًا في بناء المركبات ذاتية القيادة. تبحث هذه الدراسة في تطبيق التحكم التنبؤي بالنموذج التكيفي (AMPC) لتتبع المسار في القيادة الذاتية. إنه يستخدم نموذج دراجة ثلاثية الدرجات من الحرية ويدمج التعديلات الديناميكية من خلال دمج المستشعرات. يؤكد البحث على الضبط المنهجي لأوزان التحكم والتعديلات في الوقت الفعلي لإظهار فعالية AMPC في سيناريوهات القيادة المختلفة. تثبت نتائج المحاكاة دقة وسلامة وموثوقية AMPC، وتتضمن الأهداف الجارية مزيدًا من التطوير وتقييم دمج المستشعرات، وخاصة مع كاميرات العمق. يعزز استخدام محاكاة Unreal Engine 3D من التطبيق العملي للنهج، مما يسلط الضوء على إمكانية تنفيذ AMPC في الوقت الفعلي في القيادة الذاتية واتباع المسار. ومن المتوقع أن تساهم التطورات المستمرة في خوارزميات التحكم وتقنيات المستشعرات في التحسين المستمر في أداء AMPC.

1. Introduction

In recent years, autonomous vehicle technology has advanced significantly due to the integration of state-of-the-art sensing technologies, artificial intelligence, and sophisticated control systems [1]. These vehicles promise to improve ride comfort, optimize resource use, reduce emissions, and, most importantly, enhance driving safety. Consequently, governments, universities, and businesses have shown considerable interest and investment in this rapidly evolving field [2]. A key challenge in autonomous vehicles is path tracking, which ensures that a vehicle accurately follows a predetermined trajectory set either offline by a navigation system or online through path-planning methods [3]. Designing effective tracking algorithms requires high accuracy and smoothness due to the

*Corresponding author:
E-mail addresses: a.abougarair@uot.edu.ly, (I. B. Attawil)attawil@uot.edu.ly ,(A. A. Alaktiwi) A.Alaktiwi@uot.edu.ly ,(M. Bakouri)m.bakouri@mu.edu.sa

Article History : Received 13 March 2024 - Received in revised form 25 September 2024 - Accepted 15 October 2024

complex, multivariable, and nonlinear nature of the problem. Extensive research has focused on adaptive control techniques for systems without constraints. Traditional control strategies like Proportional Integration Differentiation (PID) control and fuzzy logic control have been used for path tracking in autonomous vehicles [4,5]. However, these methods may not fully meet the diverse control objectives, such as tracking accuracy and driving stability. Recently, the concept of online model adaptation has been extended to include Model Predictive Control (MPC) design [6]. MPC has emerged as a viable alternative because it can simultaneously consider multiple objectives by minimizing a multivariable cost function based on future reference states while adhering to constraints. Researchers have explored various MPC approaches to improve autonomous vehicle performance in terms of tracking accuracy and stability [7]. These strategies often involve adjusting weight values in the cost function to ensure driving stability. While MPC is effective in maintaining high tracking accuracy, it can struggle with changes in vehicle dynamics, especially when the vehicle deviates significantly from its desired path [8,9]. To address this issue, an adaptive-MPC strategy has been proposed. This approach requires an updated dynamics model that interfaces with Adaptive-MPC to account for changes in the vehicle's dynamics, making it a better solution for lane tracking problems. The vehicle is equipped with a depth camera sensor for sensor fusion, capturing environmental data for processing. This processed information is used by the controller for lane tracking. The integrated model and controller produce simulation results that are transferred to the Co-Simulation MATLAB application [10,11]. This application uses Unreal Engine to generate real-time 3D representations, providing a realistic preview of the vehicle's behavior in a simulated scenario [12,13]. This paper introduces an iterative trial-and-error tuning approach for Adaptive-MPC in vehicle lane following. The focus is on thoroughly investigating how various components integrate and their individual and collective impact on system performance. The evaluation emphasizes the importance of both lane-tracking accuracy and driving efficiency in autonomous vehicles. The goal is to understand how these components influence the overall effectiveness of the adaptive-MPC system for vehicle lane following.

Vehicle Model

Vehicle dynamics involves analyzing how vehicles move, particularly how a vehicle's forward motion is influenced by driver actions, propulsion system outputs, environmental conditions, and the state of the road or surface. This field of study is rooted in classical mechanics and is an essential aspect of engineering.

A widely used model in vehicle dynamics is the bicycle model, which simplifies a four-wheeled vehicle to a two-wheeled one by aligning the front and rear wheels along a single track. This model operates with three degrees of freedom, allowing it to capture key aspects of lateral vehicle motion, such as slip angle, cornering force, and yaw rate. However, it falls short in accounting for significant factors like roll motion, tire load transfer, and nonlinear tire behavior. Consequently, more sophisticated models are required for enhanced analysis and simulation.

To effectively describe the motion of the bicycle model, two reference frames are employed. The body frame is beneficial for illustrating the forces and torques acting on the model, as these forces typically align with the model's principal directions. The inertia frame, in contrast, is fixed and does not rotate or accelerate, making it suitable for representing the model's position and velocity in space. Utilizing both frames allows for a comprehensive understanding of the model's dynamics and kinematics. The choice of coordinate system for expressing vehicle motion depends on the specific goals of the study or control design, with each system offering its own set of advantages and drawbacks. A hybrid coordinate system combines elements from both frames to streamline the system equations. Modeling begins with mathematical derivations tailored to describe the vehicle system in question and the updated model intended for control. These equations of motion will express the relationship between steering, angular acceleration, and lateral acceleration. The lateral dynamics are represented by equations that account for all forces in the x and y directions, as well as the rotational forces represented by momentum (M). Starting from the math equation that represent the motion of the Lateral dynamics which F represent all the forces in x, y and M represents the momentum which is the rotation forces, as [14]:

$$\sum F = ma$$

And

$$\sum M = I\ddot{\alpha}$$

Lateral equations:

$$\ddot{y} = \frac{2(C_{lr}+C_{lf}\cos(\delta))}{m\,\dot{x}}\dot{y} + \left(\frac{2(-C_{lr}l_r - C_{lf}\ l_f\cos(\delta))}{m\,\dot{x}} - \dot{x}\right)\dot{\Psi} + \frac{2C_{lf}\ \cos(\delta)}{m}\delta \quad (1)$$

$$\ddot{\Psi} = \frac{2(-C_{lr}l_r - C_{lf}\ l_f\cos(\delta))}{I_z\dot{x}}\dot{y} + \frac{(-C_{lr}l_r^2 - C_{lf}\ l_f^2\cos(\delta))}{I_z\dot{x}}\dot{\Psi} + \frac{2C_{lf}\, l_f\cos(\delta)}{I_z}\delta \quad (2)$$

Where, $\ddot{y}$ is the lateral acceleration and $\ddot{\Psi}$ is the angular acceleration. Longitudinal equations as:

$$\ddot{x} = -\frac{\dot{x}}{\tau} \quad (3)$$

In this context, Where, longitudinal acceleration is $\ddot{x}$, $\dot{x}$ is longitudinal velocity, x longitudinal position, and τ is the Longitudinal acceleration tracking time constant [15]. The system parameters as shown in Fig. 1 are summarized in Table 1. Below are the approximations used in our model:

- $\tan(\theta r) \approx \theta r$ and $\tan(\theta f) \approx \theta f$
- Only lateral and longitudinal dynamics are included.
- Longitudinal practical limitation and factors neglected.
- We assume that the model does not go for high speed above 60 m/h and there are no hard maneuvers.

Table 1: Symbols and abbreviation meaning

Symbol	Meaning	Symbol	Meaning
l_f	Front length	ψ	Inertial frame yaw
l_r	Rear length	$\dot{\psi}$	Inertial frame
δ_f	Front steering	$\ddot{\psi}$	Inertial frame
δ_r	Rear steer angle	θ_{vr}, θ_{vf}	Velocity angle
β	Steer angle from	r	Radius
C_f	Front tire corner	θ	Angle
C_r	Rear tire corner	i_z	Inertia

State space representation is a method for describing a system through its state variables and their dynamics over time. The state variables form the essential set of variables needed to capture the system's behavior. This representation comprises two key equations: the state equation and the output equation [16, 17].

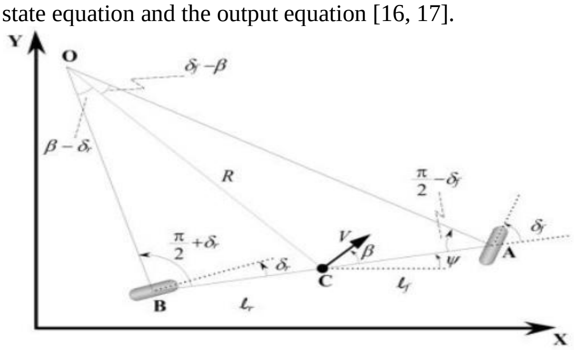

Fig. 1. The model with frames [17]

In matrix form, the state space representation can be expressed as:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

In this notation, x is the state vector, u is the input vector, y is the output vector, and A,B,C, and D are constant matrices. Utilizing this framework, we can model the three degrees of freedom (3DOF) vehicle dynamics as follows:

$$\begin{bmatrix} \ddot{y} \\ \ddot{\psi} \\ \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 \\ \ddot{0} & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{\psi} \\ \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} B_{11} & 0 \\ B_{21} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ a \end{bmatrix} \quad (4)$$

$$A_{11} = \frac{2(C_{lr} + C_{lf}\cos(\delta))}{m\,\dot{x}}$$

$$A_{22} = \frac{(-C_{lr}l_r^2 - C_{lf}\ l_f^2\cos(\delta)\ )}{I_z\dot{x}}$$

$$A_{12} = \frac{2(-C_{lr}l_r - C_{lf}\ l_f\cos(\delta))}{m\,\dot{x}}$$

$$B_{11} = \frac{2C_{lf}\ \cos(\delta)}{m}$$

$$A_{21} = \frac{2(-C_{lr}l_r - C_{lf}\ l_f\cos(\delta)\ )}{I_z\dot{x}}$$

$$B_{21} = \frac{2C_{lf}\,l_f\cos(\delta)}{I_z}$$

2. Sensor Fusion

Sensor fusion refers to the integration of data from multiple sources, usually sensors that measure different parameters such as acceleration, distance, or orientation. This integration is crucial for improving understanding and reliability in autonomous systems. It plays a key role in the four main stages of operation: sensing, perceiving, planning, and acting. While single-sensor fusion aims to enhance data quality, the real advantage comes from combining data from multiple sensors to provide a more complete view of the environment. The sensing stage involves gathering information from various sensors, each offering distinct characteristics. The perceiving stage focuses on interpreting the raw sensor data, utilizing methods like filtering, calibration, and feature extraction. In this research, we concentrate on lane detection using a depth camera, highlighting the importance of accurately understanding the road, lane markings, and potential obstacles. This understanding is essential for enabling the system to make informed decisions [18, 19].

Depth camera

Depth cameras offer valuable information regarding the distance of objects within a scene, enhancing the accuracy of lane tracking. This capability is particularly beneficial in low-light conditions or when lane markings are not clearly visible. Additionally, depth cameras can be employed to monitor the positions of other vehicles on the road, aiding in collision prevention [9].

The primary purpose of the camera is to detect lanes; therefore, we should configure the parameters to focus solely on capturing the necessary data. As a result, only a few parameters will be adjusted, while the rest will remain at their default settings [20, 21].

Processing

The depth camera utilized provides data on lane boundaries, which we can analyze to determine the desired lane center for the vehicle. The parameters employed in calculating the lane center include:

Curvature, Curvature derivative, Lateral offset and Heading angle (yaw).

The processing involves using this data to compute the error that the Adaptive Model Predictive Control (MPC) system aims to minimize for effective lane tracking. This is achieved through straightforward calculations involving curvature, its derivative, lateral offset (which indicates how much the vehicle deviates from the lane boundary), and the yaw angle (representing the angle between the vehicle's orientation and its direction of travel). The decision-making process is based on three scenarios:

1- When both lane lines are visible: In this scenario, all four parameters from both sides are accessible. The data from the left and right lane boundaries are averaged to determine the lane centre. As the vehicle approaches the lane, this estimation is continuously corrected in each iteration to ensure accurate tracking.

$$Curvature\ = 0.5(Right.Curvature + Left.Curvature)$$

$$dCurvature = 0.5(Right.Curvature + Left.Curvature)$$

$$Lateraloffset\ = 0.5(Right.Lateral\ offset + Left.Lateraloffset)$$

$$HeadingAngleyaw = 0.5(Right.HeadingAngle + Left.HeadingAngle)$$

2- When only the right lane line is visible: Given that the lane width is 3.6 meters, we estimate the lane centre by using half of the lane width to calculate the position relative to the right line. This approach allows us to make an informed estimation of the lane centre based on the available data.

$$Curveture = \ Right\ curvature\ /(1 + \ (Right\ curvature) * \ guard)$$

$$dCurveture = \ Right\ curvatureer\ divative\ /(1 + \ (Right\ curvature) * (\ guard)^2)$$

$$HeadingAngle\ yaw = Right.HeadingAngleyaw$$
$$Lateral\ offset\ = Right.Lateraloffset$$
$$Where, the\ gard = 1.8$$

3- When only the left lane line is visible: In this case, we apply a similar approach as for the right line. Since the lane width is 3.6 meters, we estimate the lane centre by subtracting half of the lane width from the position of the left line. This adjustment helps us accurately determine the lane centre in the opposite direction.

$$Curveture = \ left\ curvature(1 - (left\ curvature) * \ guard)$$

$$dCurveture = \ left\ curvature\ derivative\ /(1 - (left\ curvature) * (guard)^2)$$

$$HeadingAngle\ yaw = Left.HeadingAngleyaw$$

The purpose of these three processing scenarios is to estimate the lane center, constrained by the lane boundaries, with the ultimate goal of keeping the vehicle within the lane. Additionally, this processing occurs in each iteration, requiring only a small sample of data to maintain accuracy.

However, there are some limitations to consider:

1. **Sensor Limitations**: Depth cameras may struggle in low-light conditions, adverse weather, or when lane markings are obscured, potentially leading to inaccurate data.
2. **Dynamic Environments**: Changes in road conditions, such as construction or temporary lane markings, can affect the reliability of lane detection.
3. **Computational Constraints**: Real-time processing of data may be limited by the system's computational power, which could impact responsiveness and accuracy.
4. **Assumptions in Estimation**: The method assumes consistent lane widths and clear boundaries, which may not hold true in all scenarios.
5. **Noise in Data**: Variability or noise in sensor data can lead to fluctuations in the estimated lane center, affecting tracking performance.

Addressing these limitations is essential for improving the robustness and reliability of lane detection and tracking systems.

$$-0.15 \geq Curveture \geq 0.15 \quad , \quad -0.06 \geq dCurveture \geq 0.06$$

$$-0.06 \geq Lateral\ offset \geq 0.06 \quad , \quad -0.6 \geq HeadingAngle\ yaw \geq 0.6$$

Additionally, the heading angle (yaw) and lateral offset must be adjusted in polarity when calculating for the left side, since the sensor initially detects the left boundary, as outlined in MATLAB documentation [9, 22]. To correct for this, we multiply these values by -1. The final step involves creating a previewed curvature as follows:

$$previewed\ Curveture = Curvature + dCurvature\ Vx\ .t$$

$$t = \sum_{i=0}^{prediction\ horizon+1} (k+i)Ts \quad (5)$$

In this context, V represents the longitudinal acceleration, while t is a time vector constructed using the prediction horizon and the sampling time Ts, both defined by the controller design. It is important to note that the lateral offset error and yaw angle error are utilized as measured outputs for the adaptive-MPC. Conversely, the previewed curvature, when multiplied by the longitudinal velocity Vx, serves as a measured disturbance. This disturbance indicates how much the road or path deviates to the left or right, expressed in terms of angular velocity [23, 24].

3. Model Predictive Control (MPC)

This research conducts an in-depth exploration of model-based control principles and predictive control strategies. The study aims to develop and assess a Model Predictive Control (MPC) framework, as illustrated in Fig. 2. This approach facilitates a comprehensive analysis of advanced control techniques and their application in relation to the research objectives.

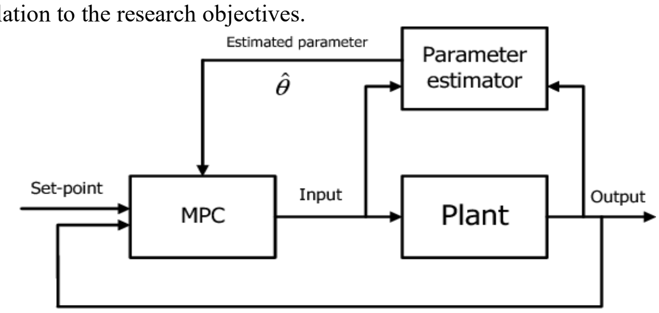

Fig. 2: MPC Control

As shown in Fig. 3, model-based control is essential for our lane-following application, offering a reliable and accurate representation of the system's dynamic behavior. For the control strategy to succeed, it is crucial that the model effectively captures the vehicle's key characteristics and accurately reflects its interactions with the surrounding environment. This solid foundation is vital for creating a successful control system for lane-following applications. MPC integrates model-based control and predictive control to formulate effective strategies. MPC employs a model of the system dynamics to forecast future behavior and optimize control actions over a defined prediction horizon. This methodology is particularly beneficial in lane-following scenarios, where precise trajectory tracking is critical. In MPC, a quadratic cost function is often used to strike a balance between tracking accuracy and control effort [9].

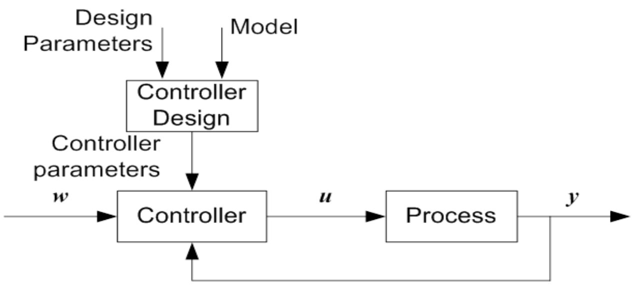

Fig. 3: Model-based control

The cost function penalizes any deviation between the predicted output and the desired reference signal, thereby prioritizing precise tracking. It also includes a term that discourages excessive control effort, fostering energy-efficient and smooth control actions. By minimizing this quadratic cost function, Model Predictive Control (MPC) optimizes the control inputs to ensure accurate tracking while balancing the trade-off between tracking performance and control effort [10].

$$Cost\ function = Tracking\ Term + effort\ term$$

$$J = \frac{1}{2}\vec{e}_{k+n}^{T} S\ \vec{e}_{k+n} + \frac{1}{2}\sum_{i=0}^{n-1}\vec{e}_{k+i}^{T} Q\ \vec{e}_{k+i} + \frac{1}{2}\sum_{i=0}^{n}\vec{\delta}_{k+i}^{T} R\ \vec{\delta}_{k+i} \quad (6)$$

Where, $\vec{e}_k$ is the error to minimize and δ is the steer angle and (S, Q, R) are weight diagonal matrices. The cost function can be spirited to two main terms:

$$Tracking\ Term = \frac{1}{2}\vec{e}_{k+n}^{T} S\ \vec{e}_{k+n} + \frac{1}{2}\sum_{i=0}^{n-1}\vec{e}_{k+i}^{T} Q\ \vec{e}_{k+i} \quad (7)$$

$$control\ effort = \frac{1}{2}\sum_{i=0}^{n}\vec{\delta}_{k+i}^{T} R\ \vec{\delta}_{k+i}$$

(8)

The form of tracking and control effort as:

$$\frac{1}{2}(error\ vector)^{T}(weight\ diagonal\ matrix)(error\ vector) = \frac{1}{2}(weight_1)(error\ state_1)^2 + \frac{1}{2}(weight_2)(error\ state_2)^2 \quad (9)$$

Which a quadratic formation of the errors in shape of matrices, Also the fraction $\frac{1}{2}$ is exist due the gradiant solution to find the optimal point and it will cancel out in the final step, so it is just a part of the formation. In the addition, steering angle change Δδ should not treated as vector anymore because it is one steering angle in bicycle model, using it as vector will be only in prediction form [9, 22].

Fig. 4 illustrates the control algorithm used in Model Predictive Control (MPC). Predictive control is particularly beneficial for lane-following applications, which require precise tracking of the desired path. This tracking can be influenced by various factors, including road conditions, vehicle dynamics, and surrounding traffic. Predictive control techniques excel in addressing these challenges.

By incorporating predictions of the vehicle's future behavior and optimizing control actions based on a cost function, predictive control enables the vehicle to adapt to changing road conditions and manage uncertainties effectively. This capability allows for accurate adherence to the desired lane trajectory, significantly enhancing the safety, precision, and reliability of the lane-following control system. MPC is a widely used predictive control approach that employs optimization techniques to iteratively solve a cost function and generate optimal control actions. The effective application of optimization in predictive control has significantly contributed to the broad adoption of MPC across various fields. This method provides improved performance and advanced control capabilities, making it a favored choice for numerous control applications. The iterative nature of MPC, along with its optimization-driven approach, enables more refined and adaptive control strategies compared to conventional methods.

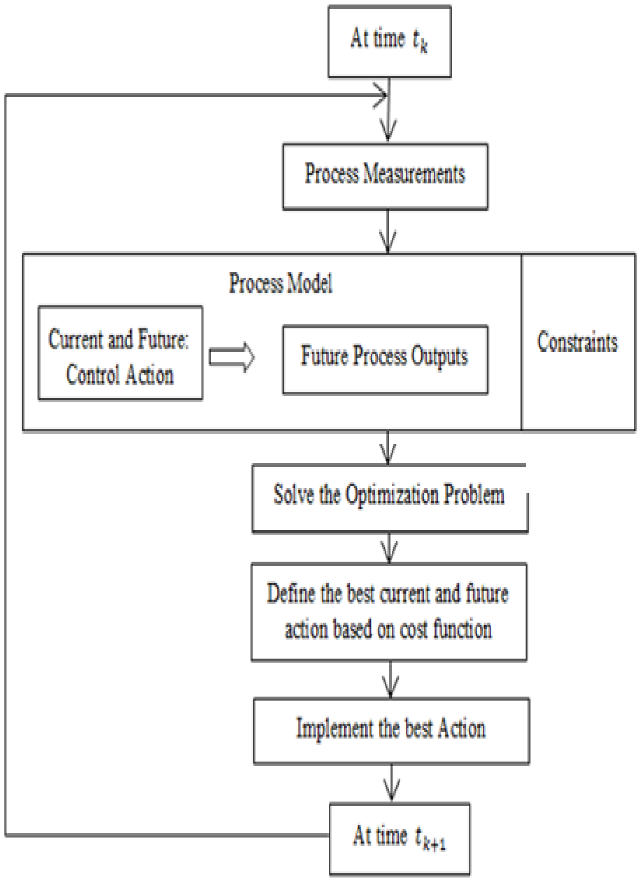

Fig. 4: Structure of MPC [9]

4. Adaptive-MPC

Incorporation adaptivity into Model Predictive Control requires transitioning to a discrete version to enhance responsiveness to changes in the system. This discrete framework facilitates more nuanced real-time adaptations. To effectively integrate adaptivity, modifications to the model are necessary to align with the designated cost function, ensuring an accurate representation of system dynamics. This refined approach, based on discrete MPC, highlights the sophistication and effectiveness of the proposed control framework [9, 20]. As illustrated in Fig. 5, Adaptive MPC is especially effective for lane-following tasks due to its ability to incorporate adaptivity while addressing vehicle dynamics. Although it operates on the same fundamental principles as traditional MPC, Adaptive MPC enhances performance through additional features. The key difference is its capability to continuously update and fine-tune model parameters based on real-time data and observations of the vehicle's behavior.

This adaptive nature allows the system to consider fluctuations in vehicle dynamics, changes in road conditions, and various measured disturbances. By integrating these updated states into each control loop, Adaptive MPC ensures more accurate predictions and control actions, resulting in improved trajectory tracking even amid dynamic variations and external disturbances. The system's real-time adaptability makes it particularly well-suited for the complex and ever-changing conditions typical in lane-following applications.

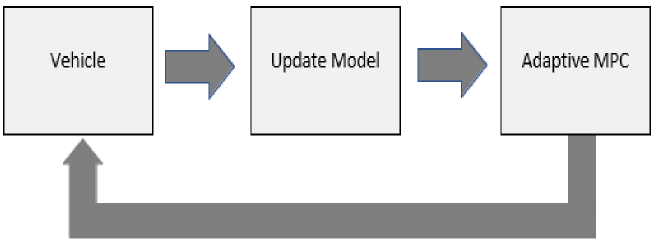

Fig. 5: Adaptive MPC update mechanism

Model modification

To get the Δδ as the control effort we need to modify the state space representation as the following:

$$\vec{x}_{(k+1)} = A\vec{x}_k + B\delta_k$$

$$\Delta\delta = \delta_k - \delta_{k-1}$$

$$\vec{x}_{(k+1)} = A\vec{x}_k + B\vec{\delta}_{(k-1)} + B\Delta\delta_k$$
$$\delta_k = \delta_{k-1} + \Delta\delta_k$$
$$\begin{bmatrix}\vec{x}_{(k+1)}\\ \delta_k\end{bmatrix} = \begin{bmatrix}A & B\\ 0 & I\end{bmatrix}\begin{bmatrix}\vec{x}_k\\ \delta_{k-1}\end{bmatrix} + \begin{bmatrix}B\\ I\end{bmatrix} \tag{10}$$

$$\overline{x}_{(k+1)} = \overline{A}\overline{x}_k + \overline{B}\Delta\delta \tag{11}$$

We also need to augment the measured disturbance and use it difference as the measured disturbance.

$$Then\ \tilde{x}(k) = [\overline{x}^T(k), d^T(k)]^T$$

$$\begin{bmatrix}\vec{x}_{(k+1)}\\ \delta_k\\ d_k\end{bmatrix} = \begin{bmatrix}A & B & B_d\\ 0 & I & 0\\ 0 & 0 & I\end{bmatrix}\begin{bmatrix}\vec{x}_k\\ \delta_{k-1}\\ d_{k-1}\end{bmatrix} + \begin{bmatrix}B\\ 0\\ 0\end{bmatrix}\Delta\delta + \begin{bmatrix}B_d\\ 0\\ 0\end{bmatrix}\Delta d \tag{12}$$

$$\tilde{x}_{(k+1)} = \tilde{A}\tilde{x}_k + B\Delta\delta + B_d\Delta d$$
$$and$$
$$\tilde{x}_{(k+1)} = \tilde{A}\tilde{x}_k + \tilde{B}\,\Delta\tilde{u}$$

$$\tilde{y}_k = \tilde{C}\tilde{x}_k + \tilde{D}\,\Delta\tilde{u}\ , \tilde{C} = [C\ 0\ 0]$$

Starting of expanding what the errors really are:

$$\vec{e}^{\,T}_{k+i} = ref_{k+5} - y_{k+5} = ref_{k+5} - Cx_{k+5} \tag{13}$$

as i =5, ref is the reference and y are the measured output.

$$J = \frac{1}{2}(ref_{k+n} - Cx_{k+n})^T S\,(ref_{k+n} - Cx_{k+n}) +$$
$$\frac{1}{2}\sum_{i=0}^{n-1}(ref_{k+n} - Cx_{k+n})^T Q\,(ref_{k+n} - Cx_{k+n}) +$$
$$\frac{1}{2}\sum_{i=0}^{n}\Delta\delta^T_{k+i}\,R\ \Delta\delta_{k+i} \tag{14}$$

To sum up, we augment our steering and measured disturbance so we can solve for the steering increments ending with modifying our state space representation to be ready for optimization and prediction. Simplify the cost function by eliminating irrelevant constant terms arising from feature-weight multiplication. Adjust constraints based on vehicle dynamics for accuracy. Express the cost function in matrix form using vectors $\vec{\mathbf{x}}_{\mathbf{G}}$ for future states and $\vec{\mathrm{r}}_{\mathrm{G}}$ for reference values.

$$\vec{x}_G^T\begin{bmatrix}\tilde{C}^TQ\,\tilde{C} & 0 & 0 & 0 & 0\\ 0 & \tilde{C}^TQ\,\tilde{C} & 0 & 0 & 0\\ 0 & 0 & \tilde{C}^TQ\,\tilde{C} & 0 & 0\\ 0 & 0 & 0 & \ddots & \vdots\\ 0 & 0 & 0 & \dots & \tilde{C}^TS\,\tilde{C}\end{bmatrix}\vec{x}_G -$$
$$\vec{r}_G\begin{bmatrix}Q\,\tilde{C} & 0 & 0 & 0 & 0\\ 0 & Q\,\tilde{C} & 0 & 0 & 0\\ 0 & 0 & Q\,\tilde{C} & 0 & 0\\ 0 & 0 & 0 & \ddots & \vdots\\ 0 & 0 & 0 & \dots & S\,\tilde{C}\end{bmatrix}\vec{x}_G +$$
$$\frac{1}{2}\overrightarrow{\Delta\delta}_G^T\begin{bmatrix}R & 0 & 0 & 0 & 0\\ 0 & R & 0 & 0 & 0\\ 0 & 0 & R & 0 & 0\\ 0 & 0 & 0 & \ddots & \vdots\\ 0 & 0 & 0 & \dots & R\end{bmatrix}\overrightarrow{\Delta\delta}_G$$
(15)

$$J = \frac{1}{2}\,\vec{x}_G^T\,\bar{\bar{Q}}\,\vec{x}_G - \vec{r}_G\,\bar{\bar{T}}\,\vec{x}_G + \frac{1}{2}\,\overrightarrow{\Delta\delta}_G^T\,\bar{\bar{R}}\,\overrightarrow{\Delta\delta}_G \tag{16}$$

Expanding the predicted states

As we saw in prediction model section, we represent every future state with the present state as the following form:

$$\begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{x}_{k+2} \\ \tilde{x}_{k+3} \\ \tilde{x}_{k+4} \\ \tilde{x}_{k+5} \end{bmatrix} = \begin{bmatrix} \tilde{A} \\ \tilde{A}^2 \\ \tilde{A}^3 \\ \tilde{A}^4 \\ \tilde{A}^5 \end{bmatrix} \vec{\tilde{x}}_k + \begin{bmatrix} \tilde{B} & 0 & 0 & 0 & 0 \\ \tilde{A}\tilde{B} & \tilde{B} & 0 & 0 & 0 \\ \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} & 0 & 0 \\ \tilde{A}^3\tilde{B} & \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} & 0 \\ \tilde{A}^4\tilde{B} & \tilde{A}^3\tilde{B} & \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} \end{bmatrix} \begin{bmatrix} \Delta\delta_k \\ \Delta\delta_{k+1} \\ \Delta\delta_{k+2} \\ \Delta\delta_{k+3} \\ \Delta\delta_{k+4} \end{bmatrix} \quad (17)$$

$$\vec{\tilde{x}}_G = \hat{\bar{A}}\,\vec{\tilde{x}}_k + \bar{\bar{C}}\,\overrightarrow{\Delta\delta}_G \quad (18)$$

Expanding what $\vec{\tilde{x}}_G$ equal which is the prediction of the states depending on the increment of the steering $\overrightarrow{\Delta\delta}_G$ going to give us an equation with one variable to solve for.

$$J = \frac{1}{2}\left(\hat{\bar{A}}\,\vec{\tilde{x}}_k + \bar{\bar{C}}\,\overrightarrow{\Delta\delta}_G\right)^T \bar{\bar{Q}}\left(\hat{\bar{A}}\,\vec{\tilde{x}}_k + \bar{\bar{C}}\,\overrightarrow{\Delta\delta}_G\right) - \vec{r}_G\,\bar{\bar{T}}\left(\hat{\bar{A}}\,\vec{\tilde{x}}_k + \bar{\bar{C}}\,\overrightarrow{\Delta\delta}_G\right) + \frac{1}{2}\overrightarrow{\Delta\delta}_G^T\,\bar{\bar{R}}\,\overrightarrow{\Delta\delta}_G$$
(19)

After multiplying prentices and neglecting constant terms, we will end up with our final equation which is the following equation:

$$J = \frac{1}{2}\overrightarrow{\Delta\delta}_G^T\left(\bar{\bar{C}}^T\,\bar{\bar{Q}}\,\bar{\bar{C}} + \bar{\bar{R}}\right)\overrightarrow{\Delta\delta}_G + \begin{bmatrix}\vec{\tilde{x}}_G^T & \vec{r}_G^T\end{bmatrix}\begin{bmatrix}\hat{\bar{A}}\,\bar{\bar{Q}}\,\bar{\bar{C}} \\ -\bar{\bar{T}}\,\bar{\bar{C}}\end{bmatrix}\overrightarrow{\Delta\delta}_G \quad (20)$$

$$J = \frac{1}{2}\overrightarrow{\Delta\delta}_G^T\,\bar{\bar{H}}\,\overrightarrow{\Delta\delta}_G + \begin{bmatrix}\vec{\tilde{x}}_G^T & \vec{r}_G^T\end{bmatrix}\bar{\bar{F}}^T\,\overrightarrow{\Delta\delta}_G \quad (21)$$

Here, H represents the Quadratic Coefficient Matrix, and F denotes the Linear Coefficient Matrix. Due to the complexity of our model, which involves iterative cost function minimization with constraints, traditional gradient methods for determining the optimal steering angle are not feasible. Instead, we utilize a quadratic programming solver that is capable of managing linear constraints to obtain an optimal solution [9, 16].

Constrains

Model Predictive Control (MPC) is a method for managing a system by utilizing a model to predict its future behavior and selecting the optimal inputs. MPC incorporates constraints to restrict optimization and predictions within the model's limitations, ensuring system safety. It can accommodate various types of constraints, including physical limitations, input bounds, and output specifications. These constraints are crucial for maintaining the system's safety, feasibility, and performance [9]. The constraints in our model include ranges for acceleration and steering. We set the acceleration at 0.5 to allow the vehicle to gradually increase speed, enabling the controller to adjust its position within the lane. Additionally, we specify a deceleration of -1, allowing the vehicle to slow down quickly if it exceeds the desired speed [14].

For steering, the allowable range is from -0.26 to 0.26 radians, which keeps the control within the linear range of tan(0)\tan(0) tan(0). Typically, the steering angle for a bicycle falls within -45 to 45 degrees, representing left and right turns. Thus, the cost function minimization is subjected to these constraints.

$$-1 \geq acceleration \geq 0.5\ \frac{m}{s^2} \quad , -0.26 \geq steering \geq 0.26\ rad$$

Weights

MPC weights are coefficients that define the relative importance of the output, input, and input rate terms within the MPC cost function. This cost function is a quadratic function that quantifies the deviations of the outputs from their reference values, the inputs from their target values, and the variations of the inputs [9].

TABLE 2: Tuning process

Number	Yaw error weight
1	0.19
2	0.49
3	0.88
4	1.08
5	1.43

5. Simulation Results

The Path Following Control System block simulates a path-following control (PFC) system that ensures a safe distance from a lead vehicle while tracking a specified velocity and keeping the ego vehicle centered on a straight or curved path (see Fig. 7). The controller achieves this by adjusting the front steering angle and longitudinal acceleration of the ego vehicle. It employs adaptive MPC to determine the optimal actions while adhering to constraints related to safe distance, velocity, acceleration, and steering angle. In the lane-following application, the system receives longitudinal velocity and steering angle inputs from the vehicle, which are processed by the Simulink update model (see Fig. 8). This update model utilizes the information to dynamically adjust the system matrices A, B, and C to account for real-time changes in the vehicle's dynamics [25].

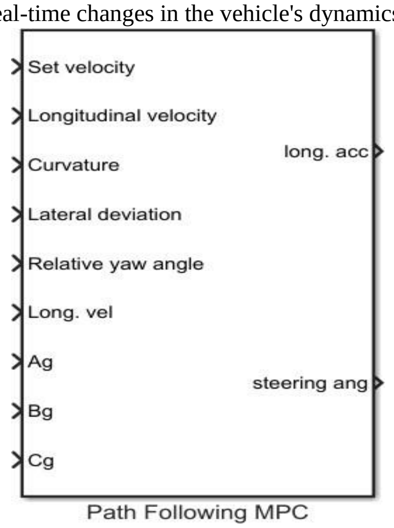

Fig.7: Path Following MPC [25]

The controller can then adjust and optimize its control actions based on the current vehicle status and environmental variables by incorporating these updated matrices into the MPC algorithm.

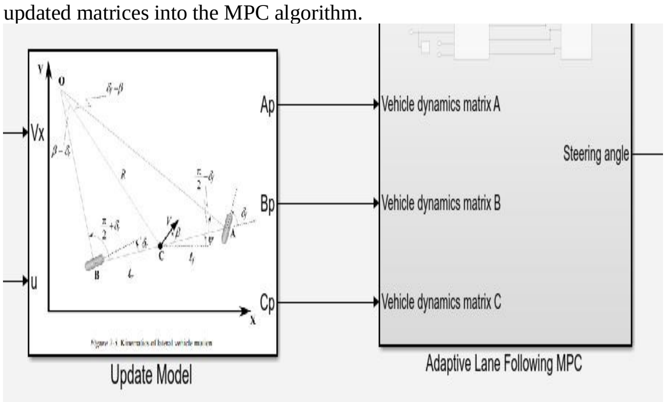

Fig. 8: Update Model [25]

The Simulink model is a mathematical depiction of the longitudinal, lateral, and yaw motion of a vehicle. The vehicle's dynamics and control under various inputs and conditions can be studied and designed thanks to the model. But this model is too complicated for our goal, which is to concentrate on the vehicle's lateral dynamics. In order to create a simpler model for the MPC controller, we will take the pertinent data from the 3DOF model (Fig. 9). Together with the 3DOF model, which depicts the vehicle's plant system, this simplified model—which we refer to as the update model—represents the dynamics on which the control is based.

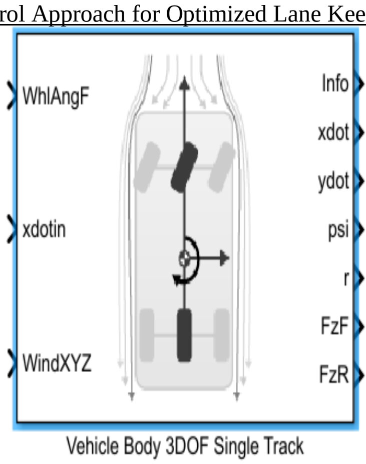

Fig. 9: Vehicle 3dof Simulink model

We chose the Curved Road map, a sizable map with a mix of hard twists, mild turns, and straight roads, for our paper (Fig. 10). Our lane following method is perfect for testing on this map. Since they weren't pertinent to our goal, we didn't include any additional vehicles or items in the scene.

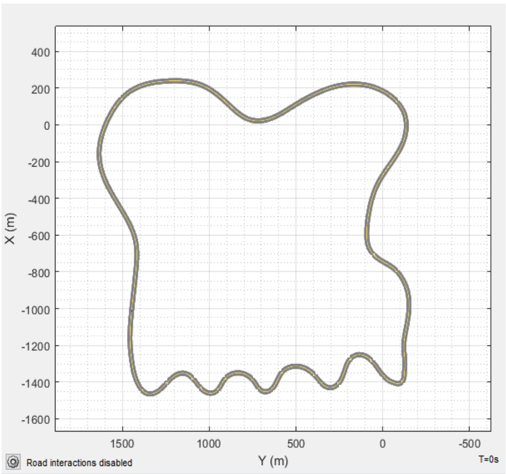

Fig. 10: Environment Map

In some situations, the detection range angle must balance between enhancing visibility and preventing the detection of irrelevant objects, such as lanes on adjacent roadways. Consequently, the processing unit must either manage this parameter effectively or limit it to the target lane (see Fig. 11). The data processing equation is implemented in Simulink using blocks that carry out the necessary calculations and transformations. The processed variables then serve as crucial inputs for the Lane Following Adaptive MPC, ensuring accurate vehicle positioning within the lane (see Fig. 12).

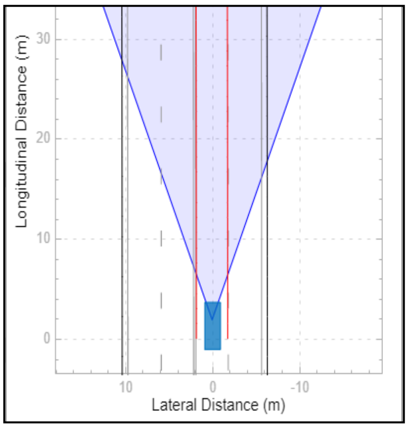

Fig. 11: Depth Camera Lane Detection

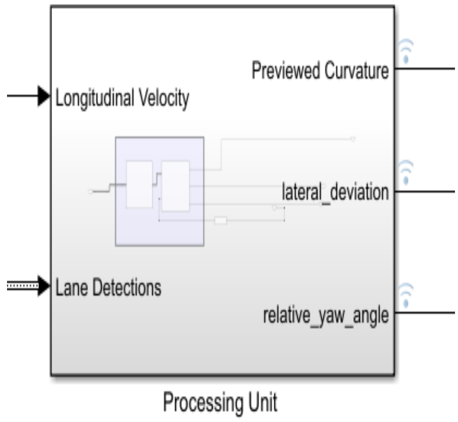

Fig. 12: Processing unit Inputs and Outputs [25]

After completing the weight tuning experiments, the resulting weight values for vehicle control have demonstrated significant improvements in lane tracking, as shown in Fig. 13. Here, the orange dashed line represents the lane center, while the green dotted line indicates the vehicle's position. It is noted that there are sections where the lines are not perpendicular, indicating that the adaptive MPC faces challenges keeping the vehicle centered in the lane—especially during sharp turns or when the vehicle initially strays from the lane.

Fig. 14 illustrates the lateral error at the beginning, where the vehicle is approximately 50 centimeters off the lane, and this error decreases as the vehicle progresses, as depicted in Fig. 15. For the controller to effectively track the lane, it must make predictions, which are highlighted in both Fig. 14 and Fig. 15 by the various colored signals surrounding the dotted green vehicle position line. These predictions are used to generate the next optimal step, represented by the subsequent green dot, calculated by the Adaptive MPC [24-27].

As shown in Fig. 16, the vehicle is able to track the lane effectively. A critical aspect to clarify is whether our system modeling with adaptive MPC is reliable. The answer lies in examining the highest and lowest outputs of the cost function, as illustrated in Fig. 17.

The cost function reaches zero at various points, indicating that we achieve the optimal steering and acceleration necessary for the vehicle to reach the desired tracking.

The quadratic programming algorithm status chart in Fig. 18 shows a maximum result of 3 and a minimum of zero, suggesting that the cost function is solvable, which indicates that our model representation is efficient. The effectiveness of the proposed algorithm is confirmed through simulations conducted in MATLAB/Simulink [26-30]. The simulation results reveal that the algorithm improves reference tracking performance by efficiently allocating control efforts to the steering system.

While traditional MPC has limitations related to model accuracy and uncertainty management, which can impact its performance in lane-following applications, Adaptive MPC (AMPC) strategies facilitate online estimation and adaptation of model parameters. This enhances the robustness and accuracy of the control system when confronted with varying dynamics and uncertainties. AMPC provides an effective solution to address the shortcomings of conventional MPC, leading to improved control performance in lane-following and similar tasks.

However, AMPC also has certain limitations, including increased computational complexity and sensitivity to model errors. The heightened computational demands can be particularly challenging for systems with fast dynamics. Additionally, AMPC relies heavily on an accurate system model to effectively adapt the controller parameters. Model inaccuracies can lead to suboptimal control performance or even instability in the closed-loop system. Significant model errors may hinder the adaptive algorithms' ability to accurately estimate or adjust model parameters. Furthermore, the adaptation process itself might introduce additional dynamics or time delays, potentially affecting control performance [31-34].

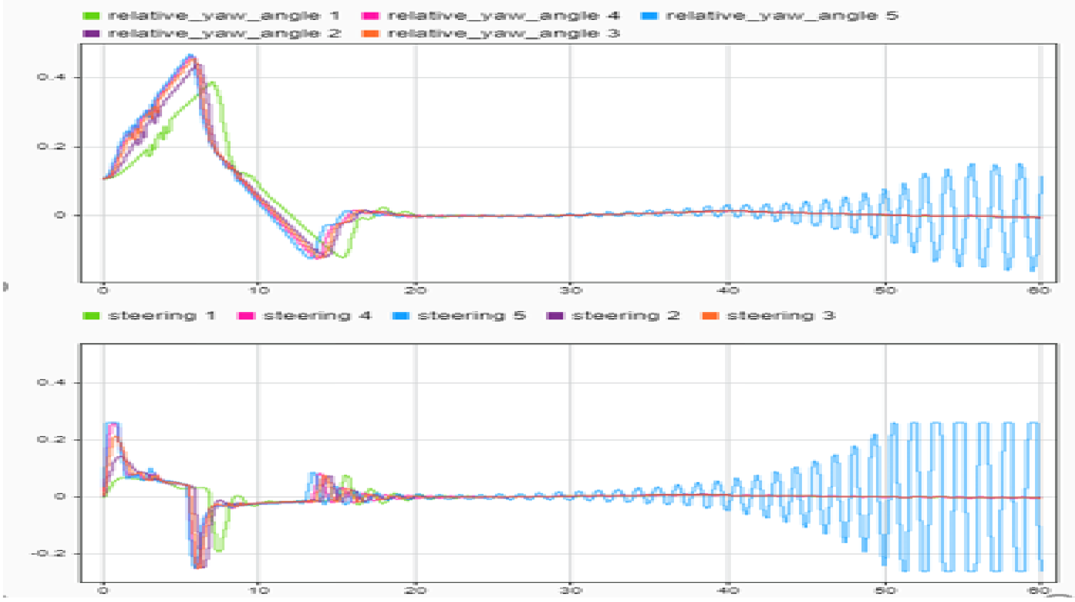

Fig. 13: Weight Tuning [25]

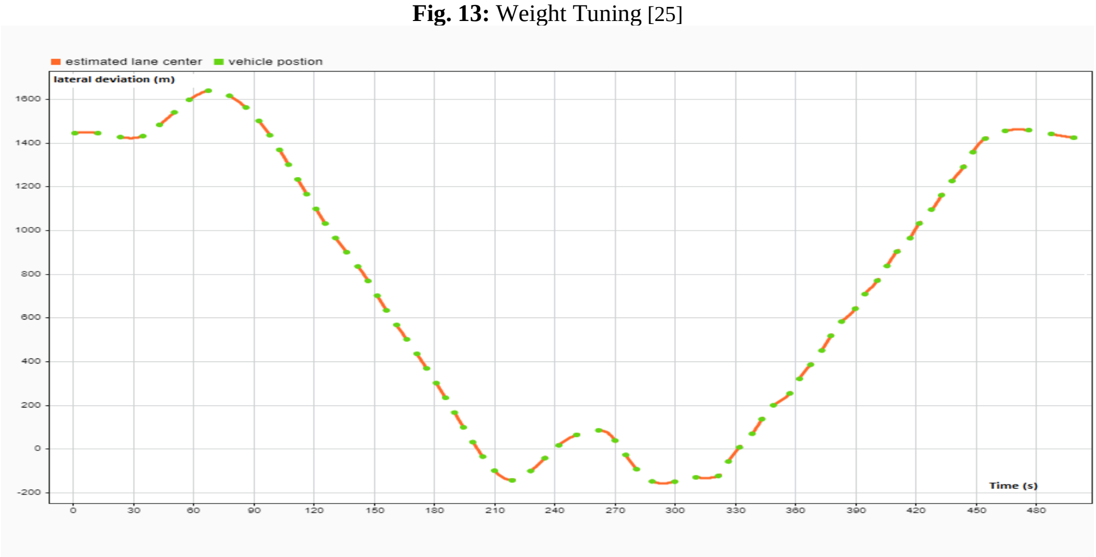

Fig. 14: Lane Tracking

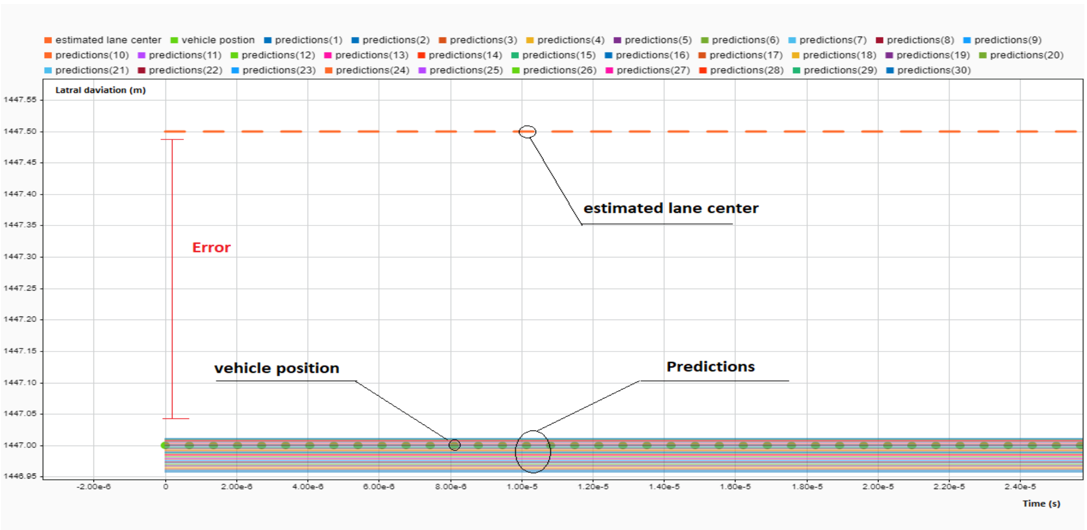

Fig. 15. Lane Tracking with predictions where vehicle far from the lane [25]

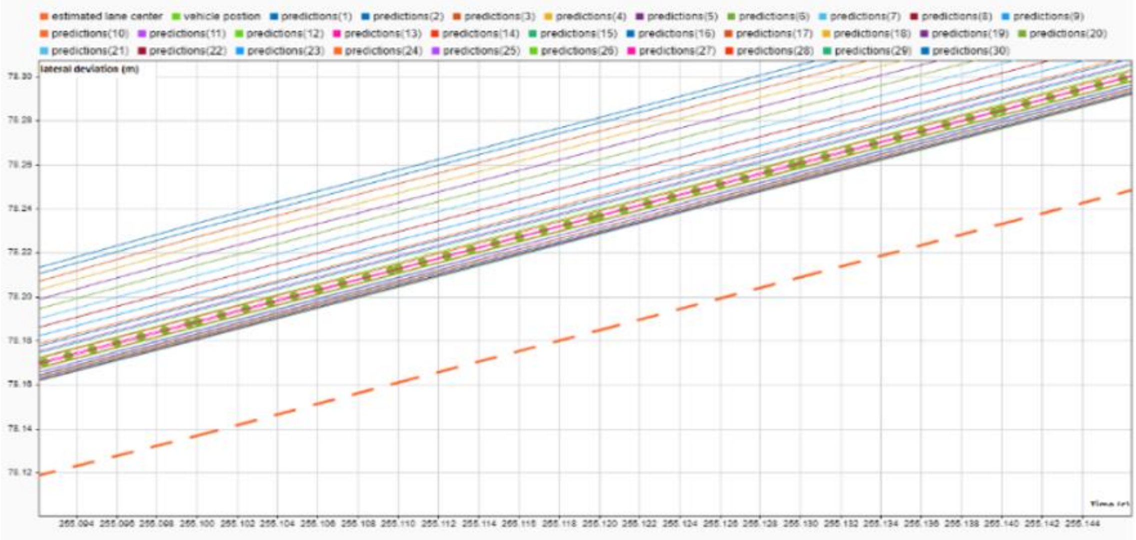

Fig. 16. Lane Tracking with predictions where vehicle in the lane [25]

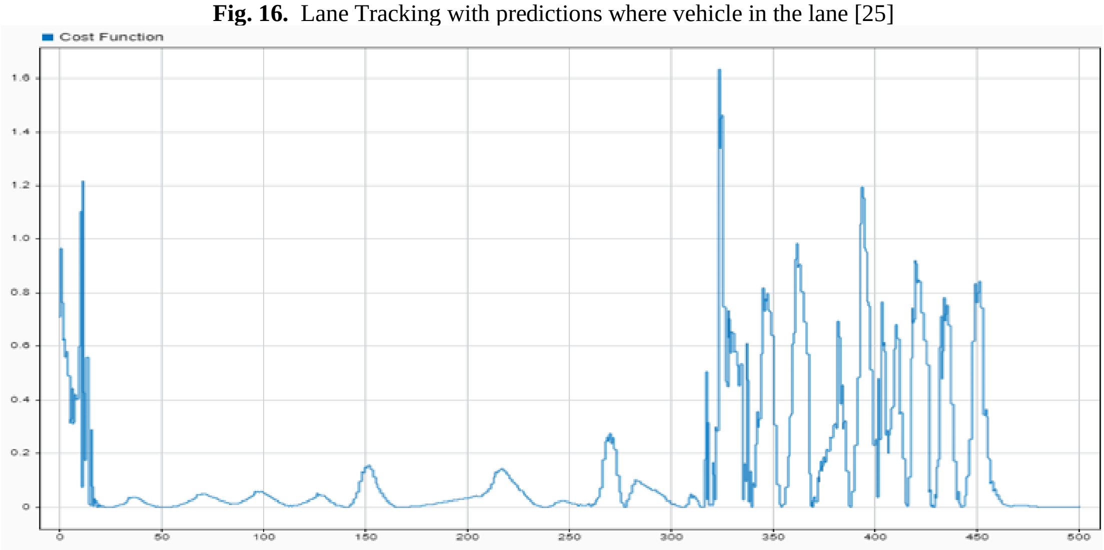

Fig. 17. Cost Function

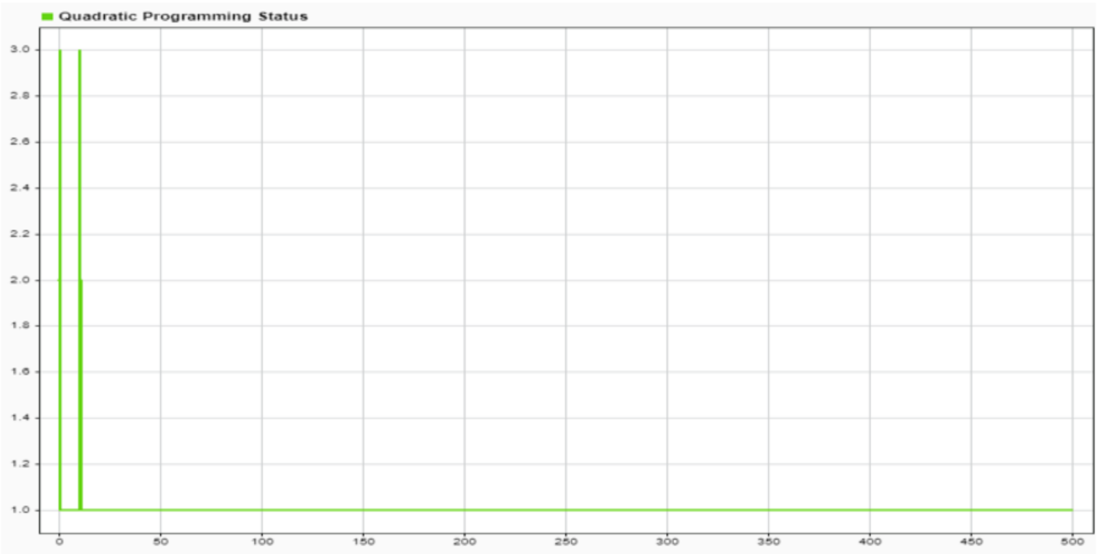

Fig. 18. Quadratic Programming Status

6. Conclusion

In summary, the effective use of lane-following Adaptive Model Predictive Control (AMPC) demonstrates its value in autonomous driving. The system's flexibility, which includes a 3-degree-of-freedom (3-DOF) model and real-time adjustments based on sensor integration, significantly improves accuracy in diverse driving situations. Simulation outcomes confirm the capability of AMPC to handle difficult conditions, with real-time modifications ensuring safety and dependability. The focus on optimization is evident through systematic tuning of control weights, highlighting the significance of a trial-and-error approach for achieving peak performance. Additionally, AMPC's computational efficiency allows for real-time deployment, making it viable for real-world applications. Future goals involve further refining the 3-DOF model and updating it to improve adaptability. We will also explore how sensor fusion, especially from depth cameras, affects AMPC's efficiency. Utilizing Unreal Engine for 3D simulation not only validates the approach's practicality but also acts as a visualization tool. Ongoing improvements in control algorithms and sensor technologies are expected to enhance AMPC's performance in autonomous driving and lane-following tasks. It's crucial to recognize that MPC can be affected by measurement noise or unforeseen disturbances, which may impact its overall effectiveness.

7. References

[1] A. Faisal, and G. Currie, Understanding autonomous vehicles. Journal of transport and land use, 12(1), pp.45-72, 2019.

[2] C. Piao and X. Liu, Lateral control using parameter self-tuning LQR on autonomous vehicle. In Proceedings of the International Conference on Intelligent Computing, Automation and Systems, Chongqing, pp. 913–917, 2019.

[3] M. Aburakhis, Performance of Anti-Lock Braking Systems Based on Adaptive and Intelligent Control Methodologies, Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 2022.

[4] J. Normey, Mobile robot path tracking using a robust PID controller, Control Eng. Pract., vol. 9, no. 11, pp. 1209–1214, 2001.

[5] J. Naranjo, Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver, IEEE Trans. Intell. Transp. Syst., vol. 9, no. 3, pp. 438– 450, 2008.

[6] G. Raffo, A Predictive Controller for Autonomous Vehicle Path Tracking. IEEE Trans. Intell. Transp. Syst. 2009, 10, 92–102.

[7] C. Sun, , Design of a Path-Tracking Steering Controller for Autonomous Vehicles. Energies 2018.

[8] H. Wang, and B. Liu , Path tracking control for autonomous vehicles based on an improved MPC. IEEE Access, 7, pp.161064-161073, 2019.

[9] L. Wang,. Model predictive control system design and implementation using MATLAB. Springer Science & Business Media, 2009.

[10] F. Eduardo , Model predictive controllers. Springer London, 2007.

[11] D. Yeong, Sensor and sensor fusion technology in autonomous vehicles: A review. Sensors, 2021.

[12] M. Swedan, Stabilizing of Quadcopter Flight Model, (MI-STA2023), 21-23 May,2023, Libya.

[13] Design driving scenarios, configure sensors, and generate synthetic data - MATLAB (mathworks.com)

[14] Rajamani, Rajesh. Vehicle dynamics and control. Springer Science & Business Media, 2011.

[15] A. Abougarair, Model Reference Adaptive Control And Fuzzy Optimal Controller For Mobile Robot, Journal of Multidisciplinary Engineering Science and Technology, pp 9722-9728,Vol. 6, Issue 3, 2019.

[16] I. Buzkhar, Modeling and Control of a Two-Wheeled Robot Machine with a Handling Mechanism, (MI-STA2023), 21-23 May, 2023.

[17] C. Garcia, Model predictive control: Theory and practice-A survey. Automatica, 25(3), 1989.

[18] A. Abougarair, Real Time Classification for Robotic Arm Control Based Electromyographic Signal, (MI-STA2022), 23-25 may,2022.

[19] A. Abougarair, Adaptive Neural Networks Based Optimal Control for Stabilizing Nonlinear System, (MI-STA2023), 21-23 May,2023, Benghazi, Libya.

[20] M. Aburakhis, Adaptive Neural Networks Based Robust Output Feedback Controllers for Nonlinear Systems, International Journal of Robotics and Control Systems, Vol. 2, No. 1, pp. 37-56, 2022.

[21] V. Kumar and J. Jerome, Robust LQR controller Design for Stabilizing and Trajectory Tracking of Inverted Pendulum, Procedia Eng. 64 , 169 – 178, 2015.

[22] S. Elwefati, Control of Epidemic Disease Based Optimization Technique", (MI-STA2021),25-27 may,2021, Tripoli, Libya.

[23] A. Abougarair, Integrated Controller Design for Underactuated Nonlinear System, IEEE, Second International Conference on Power, Control and Computer Technologies (ICPC2T-2022), 2022.

[24] N. Aboud, Robust H-Infinity Controller Synthesis Approach for Uncertainties System, (ICSC'2023), December 18-20, 2023, Sousse, Tunisia.

[25] I. Attawil, et al.,, Lateral Control of Autonomous Vehicles through Adaptive Model Predictive Control, , 2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA),Tripoli, Libya, 19-21 May 2024.

[26] Abdulhamid A. Oun , et al., Comparative Evaluation of PID Controller Tuning through Conventional and Genetic Algorithm, (MI-STA),Tripoli, Libya, 19-21 May 2024.

[27] M. Ashread, et al., Model Predictive Control for Optimizes Battery Charging Process, (MI-STA),Tripoli, Libya, 19-21 May 2024.

[28] E. Alfian, et al., Optimizing Light Intensity with PID Control, Control Systems and Optimization Letters, Vol. 1, No 3, 2023, ISSN: 2985-6116, DOI: 10.59247/csol.v1i3.38.

[29] M. Antonio, et al., "Sliding Mode Control Design for Magnetic Levitation System", Journal of Robotics and Control (JRC),Vol. 3, Issue 6, 2022. https://doi.org/10.18196/jrc.v3i6.12389

[30] A. Alduraywish , et al., Optimizing cancer treatment using optimal control theory, AIMS Mathematics, vol. 9,issue 11, 2024.

[31] A. Elmulhi, et. al., Sliding Mode Control for the Satellite with the Influence of Time Delay, 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, Libya.

[32] M. Edardar, et al.,, Lyapunov Redesign of Piezo-Actuator for Positioning Control, 2021 9th International Conference on Systems and Control (ICSC), pp499-503,France.

[33] Abougarair, Adaptive Traffic Light Dynamic Control Based on Road Traffic Signal from Google Maps ICEMIS'21, October 11–13, 2021, Almaty, Kazakhstan.

[34] Alfian Ma'arif, et. al., Analysis of Mobile Accelerometer Sensor Movement Using Machine Learning Algorithm, 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA, Libya.