Vehicle System Dynamics
International Journal of Vehicle Mechanics and Mobility

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/nvsd20

Effective adoption of vehicle models for autonomous vehicle path tracking: a switched MPC approach

Mohammad Rokonuzzaman, Navid Mohajer & Saeid Nahavandi

To cite this article: Mohammad Rokonuzzaman, Navid Mohajer & Saeid Nahavandi (2023) Effective adoption of vehicle models for autonomous vehicle path tracking: a switched MPC approach, Vehicle System Dynamics, 61:5, 1236-1259, DOI: 10.1080/00423114.2022.2071300

To link to this article: https://doi.org/10.1080/00423114.2022.2071300

Published online: 07 May 2022.

VEHICLE SYSTEM DYNAMICS
2023, VOL. 61, NO. 5, 1236–1259
https://doi.org/10.1080/00423114.2022.2071300

IAVSD

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

Check for updates

Effective adoption of vehicle models for autonomous vehicle path tracking: a switched MPC approach

Mohammad Rokonuzzaman, Navid Mohajer and Saeid Nahavandi

Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Geelong, VIC, Australia

ABSTRACT

Efficient path tracking plays a key role in the overall ride experience of Autonomous Vehicles (AVs). Model Predictive Control (MPC) is one of the most competent techniques which is capable of handling multiple variables and constraints. The performance of this controller heavily relies on the proper choice of the vehicle model used to predict future states. This work proposes a novel MPC framework for the effective adoption of vehicle models to achieve a compromise between MPC's performance and computational cost. To this aim, a Switched MPC (SMPC) using vehicle models with different levels of complexity and fidelity is developed. The SMPC uses a novel supervision scheme to adopt the appropriate vehicle model based on the models' prediction error and MPC's solution time. Two different configurations of SMPC are implemented: (1) A Fuzzy Logic System (FLS), and (2) An adaptive switching supervisor. The outcomes show that both approaches can perform path tracking accurately on roads with varying curvatures for different AV speeds. Moreover, comparisons with the conventional MPCs show that the proposed controllers can perform comparably with selective use of a complex vehicle model. Notably, they benefit from a higher computation efficiency from sidestepping the most complex models during path tracking.

ARTICLE HISTORY
Received 3 September 2021
Revised 11 April 2022
Accepted 16 April 2022

KEYWORDS
Autonomous Vehicle (AV); path tracking controller; vehicle dynamics

Nomenclature

a_x	longitudinal acceleration
a_y	lateral acceleration
C_f	cornering stiffness of the front wheel
C_b	cornering stiffness of the rear wheel
F_x	longitudinal force
F_y	lateral force
$J(\cdot)$	cost function
l_f	distance between front wheel and the centre of gravity
l_r	distance between rear wheel and the centre of gravity
N_p	MPC prediction horizon
N_c	MPC control horizon
r	yaw rate

CONTACT Mohammad Rokonuzzaman m.pappu@research.deakin.edu.au

$\mathbf{u}$	input vector
v_x	longitudinal velocity
v_y	lateral velocity
x, y	vehicle position
x_{ref}, y_{ref}	reference point position
$\mathbf{x}$	state vector
α	lateral wheel slip angle
α_f	front wheel slip angle
α_b	rear wheel slip angle
δ	steering angle
ψ	heading angle/yaw angle
ψ_{ref}	reference point heading angle

1. Introduction

Path Tracking Controller (PTC) is a decisive part of an Autonomous Vehicle (AV) interfacing the path planner unit and vehicle dynamic components. Over the last few decades, different path tracking approaches of AVs have been proposed [1]. The controllers that only use the vehicle and road geometry, such as Pure pursuit [2] and Stanley [3], are quite popular due to their simplicity. In addition to these, controllers such as Lyapunov-based controller [4], Feedback Linearisation-based Controller [5] and Sliding Mode Controller [6] have been developed. It has been shown that the path tracking task in urban road conditions can be efficiently implemented using the optimal controllers based on Linear Quadratic Regulator [7] and Model Predictive Control (MPC) [8,9].

MPC is highly desirable for AV path tracking tasks considering its capability to handle multiple inputs, states and constraints. In the MPC approach, an optimisation problem is solved based on the prediction of the states using a vehicle model. The choice of a proper prediction model is crucial for a high-dimensional system such as AV. Vehicle models with different fidelity levels can be used for state prediction. A simpler model such as the kinematic model reduces the computational effort; however, it introduces uncertainty due to the lack of dynamic terms, especially at higher vehicle speed and larger steering angles. The kinematic model may be sufficient when AV moves at a low speed on a straight road without applying a large steering angle. For manoeuvres like sharp turns or obstacle avoidance, a more complex model involving vehicle dynamics is required. The dynamic model can be formulated with different complexity levels. For example, a linear relationship between the tyre-road force and slip angle can be established for the small slip angles. For larger slip angles, nonlinear tyre models are recommended to ensure the accuracy and fidelity of path tracking [10]. It is noteworthy that a complex and high-fidelity vehicle dynamic model is computationally demanding and increases the execution time of the MPC task.

Switched MPC (SMPC) is one of the most recent improvements of MPC. In this technique, a switching criterion is used to shift between different subsystems of switched system. In SMPC, the controlled system may change between different plant dynamics or performance indices based on specific criteria. These switching criteria can be formulated based on time or the system's state space region [11]. However, state space-based formulation of the SMPC is more common. In these implementations, the operating regions are divided into several areas and separate control law is generally designed for each region according to the requirements of each region. In addition, appropriate switching laws are designed to switch between these control laws based on the switching surfaces. Using variations of the cost function for the different regions is one of the popular SMPC design approaches. In this case, different cost functions or different parameter values for the same cost function is used for different operation region. For example, in [12], weighting matrices of the cost function are changed based on the state of the system. The authors aimed to achieve optimal use of actuators and to avoid or reduce the time the system stays in a specific operating region. A similar approach was adopted in [13,14] where multi-objective MPCs were designed to allow the use of different control criteria on a different region of state space. In addition, a switch between MPC and another control technique has also been proposed. In [15], switching between an MPC and a Lyapunov-based bounded robust controller is proposed where the switching was conducted based on stability criteria on different operation regions of the system.

In the context of AV, the SMPC approach has been occasionally used for path tracking tasks. In [16], a driver model for a race car was designed where two different modes of operations were used by using two different cost functions. The objective of this work was to switch between two different driving modes, where one tries to drive to complete the track with minimum time and the second one tries to drive with maximum velocity. The switching laws were designed such that it mimics a human race car driver more appropriately. In addition, SMPC-based path tracking has been designed based on the different formulations of tracking errors and vehicle dynamic conditions. The authors used the heading deviation from the reference path and the vehicle's side-slip angle for calculating the tracking error. Two different operation mode was designed based on the use of steady-state sideslip angle and the real-time sideslip angle. A switching logic was designed to switch between these two modes based on the vehicle's transient and steady-state operating conditions. Moreover, in [17], the predicted tracking error for two different available models. Based on the difference between the actual tracking error and the prediction of these models, the model with a smaller difference was chosen. In this work, only the tracking error is considered. Another common implementation of SMPC-based path tracking where the switching surfaces and their corresponding switching laws are designed based on the vehicle dynamics such as tyre force and sideslip angle. In [18], a piecewise affine state estimation model is formulated where the complexity of the model is varied based on sideslip angle and the tyre forces. Here, the authors used the combination of sideslip angle and the tyre forces to separate the state space into different segments such as linear and saturation regions. Then, a separate MPC law was designed for each region.

The main contribution of this work is a novel control framework for adopting the most efficient vehicle model from a set of available models during the path tracking task of an AV. The efficacy of a model is evaluated based on prediction accuracy and fidelity, as well as computational complexity. A unique switching cost function is proposed based on the model prediction error and the corresponding solution time of the MPC for each model. Using this switching cost, we design and compare two different solutions of the SMPC. In the first approach, a Fuzzy Logic System (FLS) is designed to supervise the switching to an appropriate model from the set of available vehicle models. In this case, prior knowledge of the switching cost's range of each model is required. In the second approach, an adaptive switching rule is proposed where no previous knowledge of model performances in different operating conditions is required. Three vehicle models, including the kinematic model, linear dynamic model and nonlinear dynamic model, are utilised. The objective is to improve the MPC's performance and reduce the computational cost by choosing an appropriate vehicle model during the implementation of the tracking task.

The remainder of the paper is organised as follows. In Section 2, a preliminary discussion on the MPC formulation and the details of the prediction models used for designing the SMPC is given. In Section 3, a formal formulation of the problem is reported, and the corresponding approaches for solving the problems are discussed. The design of the switching supervisor, which is responsible for choosing the appropriate model, is discussed in Section 4. Moreover, a comparison of prediction errors for each model is provided, and two different approaches for designing switching supervisor is discussed in this section. Section 5 presents the discussion on the design of the proposed SMPC approaches. The implementation step and the performance evaluation of the controller are reported in Section 6. In Section 7, a discussion on the novelty and the performance of the proposed controllers is presented. In addition, Section 8 outlines the proposed future works. The conclusion of the work is finally drawn in Section 9.

2. Preliminaries

In this section, the MPC design is discussed and the mathematical formulation of three different vehicle models used in this work is explained.

2.1. Model Predictive Control

MPC is one of the highly suitable control frameworks for designing PTC for AVs. MPC has the inherent ability to withstand disturbances, in addition to handling multiple variables and constraints. In conventional MPC design, a suitable vehicle model is used to predict the vehicle's future states. An optimiser uses these predicted states to provide an optimised control sequence until a specific prediction horizon.

Let's consider the vehicle dynamic system can be expressed as

$$\mathbf{x}(t+1) = f(\mathbf{x}(\mathbf{t}), \mathbf{u}(t)). \tag{1}$$

Where, $\mathbf{x} \in R^n$ is the states and $\mathbf{u} \in R^m$ inputs and $f : R^n \times R^m \rightarrow R^n$ is the state transition function. The objective of the MPC approach is to find the optimised control sequence for the system that minimises a performance index or a cost function. The formulation of the proposed MPC for the path tracking task of the AV can be expressed as

$$\arg\min_{U} \sum_{k=0}^{N_p-1} J(\hat{\mathbf{x}}(k|t), \mathbf{u}(k|t)), \tag{2a}$$

where,

$$\hat{\mathbf{x}}(k+1|t) = f_m(\hat{\mathbf{x}}(k|t), \mathbf{u}(k|t)). \tag{2b}$$

Here, N_p is the prediction horizon, and J is the stage cost. In addition, $\hat{\mathbf{x}}(k|t)$ represents the predicted state of the vehicle at step k based on the current measured state at time t. Besides, f_m is the selected vehicle model. At each iteration, a sequence of optimised control action $U^* = [\mathbf{u}_t^*, \mathbf{u}_{t+1}^*, \ldots \mathbf{u}_{t+N_p}^*]$ is generated and only the first element of the sequence is sent to the system. The same process is repeated for the next time step.

2.2. Prediction models

Three vehicle models with different degrees of complexity are selected. A switching supervisor is designed to select the most appropriate vehicle model based on a designed switching rule. The switching supervisor selects one of these models $f_m(m = \{1, 2, 3\})$ based on a novel switching cost $\sigma(f_m)$. The formulations of these vehicle models are briefly discussed here. For each model, $\mathbf{x} = [x, y, \psi]$ is the state and $\mathbf{u} = [\delta, a_x]$ is the input. It is noted that x and y is the vehicle's position in the global coordinate frame, ψ is the yaw angle, v_x is the longitudinal velocity, a_x is the longitudinal acceleration and δ is the steering angle.

2.2.1. Kinematic model

The kinematic model is one of the most popular vehicle models for model-based tracking controller design for AVs. This model is simple, easy to implement and does not need heavy computation. The kinematic model of an AV can be expressed as:

$$\begin{aligned} \dot{x} &= v_x \cos\psi, \\ \dot{y} &= v_x \sin\psi \ , \\ \dot{\psi} &= \frac{v_x \tan\delta}{L}. \end{aligned} \tag{3a–d}$$

where L is wheelbase.

2.2.2. Dynamic model

Unlike the kinematic model, a dynamic model includes force components in the AV motion. The dynamic model is capable of simulating the motion of a vehicle effectively [19–21]. Assuming the vehicle with mass m and moment of inertia I_z measured at the centre of gravity, the vehicle motion can be expressed as

$$\begin{aligned} \dot{x} &= v_x cos\psi - v_y sin\psi \\ \dot{y} &= v_x sin\psi + v_y cos\psi \\ \dot{v}_x &= a_x + v_y r \\ \dot{v}_y &= \frac{1}{m}[F_{yf}\cos\delta + F_{yr} - F_{xf}\sin\delta] - v_x\dot{\psi}, \\ \dot{r} &= \frac{1}{I_z}[F_{yb}l_b + (F_{xf}\sin\delta - F_{yf}\cos\delta)l_f] \\ \dot{\psi} &= r \end{aligned} \tag{4a–f}$$

where F_x is the longitudinal force, F_y is the lateral force, δ is the steering angle, θ is the yaw angle, and l_f and l_b are the distances between the front and rear wheel and the centre of gravity, respectively. In these equations, b and f denote rear and forward wheels. Figure 1 shows the geometry of a dynamic bicycle model.

2.2.2.1. Linear tyre model. In the dynamic model, accurate representation of the tyre-road interaction is extremely important. Applying some specific assumptions in the state-space of the vehicle system, the linear tyre model can reflect an acceptable level of accuracy for small slip angles. In this region, a linear relationship between the generated tyre force and slip angle can be established. This relationship can be expressed as [22],

$$F_{yf} = -C_f\alpha_f, \tag{5}$$

$$F_{yb} = -C_b\alpha_b, \tag{6}$$

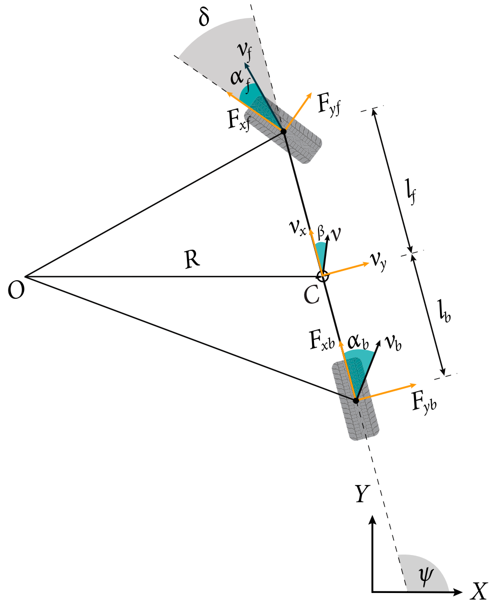

Figure 1. Geometry of dynamic bicycle model.

where α_f and α_b are the front and rear wheel slip angles, and C_f and C_b are the cornering stiffness of the front and rear wheel, respectively.

2.2.2.2. Nonlinear tyre model. Nonlinear tyre models have been introduced to accommodate a more accurate representation of the vehicle motion at larger slip angles. Several representations of nonlinear tyre models are available in the literature. These include empirical models such as the Pacejka model and analytical models such as the brush model [22,23]. Here, a modified brush model is used to represent the nonlinear tyre-road interaction. In this model, the tyre forces are a function of the normal force (F_z) and lateral slip angle (α) of the wheel.

The tyre forces can be expressed as [23]

$$F_y = \begin{cases} (\mu F_z[3\theta\sigma - 3(\theta\sigma)^2 + (\theta\sigma)^3] & \text{if } \sigma \le \sigma_s \\ \mu F_z & \text{otherwise} \end{cases} . \tag{7}$$

where, $\sigma = \tan\alpha$, μ is the friction coefficient of the road surface, σ_s is the isotropic tyre parameter expressed as

$$\sigma_s = \frac{1}{\theta} = \frac{3\mu F_z}{2C_p a^2}. \tag{8}$$

Here C_p denotes the total stiffness of tread elements per unit length and a is half of the contact length of the tyre.

3. Problem formulation

In this section, we formally formulate the problems and discuss approaches to solve the problems. The SMPC problem can be stated as

Problem-1: Given a set of vehicle models f_m with various fidelity and nonlinearity, where $m \in \mathcal{M}$ and $\mathcal{M} = \{1, \dots n\}$, design a model predictive controller which is capable of switching between these models and perform the path tracking task.

To solve this problem we propose a SMPC framework. For the system described in (1), the proposed SMPC solves the following problem

$$\begin{aligned}
&\arg\min_{U} \sum_{k=0}^{N_p-1} J(\hat{\mathbf{x}}(k|t), \mathbf{u}(k|t)) \\
&\text{subjected to,} \\
&\hat{\mathbf{x}}(k+1|t) = f_m(\hat{\mathbf{x}}(k|t), \mathbf{u}(k|t)) \\
&f_m = h(k, \sigma(m,k)) \\
&\mathbf{u}_k = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k \\
&\mathbf{x}(0|t) = \mathbf{x}(t) \\
&\mathbf{u}(k) \in \mathcal{U} \qquad \forall k \in [t, t+N_p] \\
&\hat{\mathbf{x}}(k) \in \mathcal{X} \qquad \forall k \in [t, t+N_p]
\end{aligned}, \tag{9a–g}$$

Here, the state and input constraints are represented by sets $\mathcal{X}$ and $\mathcal{U}$. In addition, $\hat{\mathbf{x}}(k|t)$ represents the predicted state of the vehicle at step k based on the current measured state at time t. Besides, $\sigma(\cdot)$ is the switching cost and $h(\cdot)$ is the switching function. The MPC uses the estimated states up to the time horizon N_p using a prediction model to calculate an optimised control sequence. Here, f_m represents the active prediction model where $m \in \mathcal{M}$ is the model number.

Problem-2: Given a set of vehicle models with various fidelity and nonlinearity f_m, where $m \in \mathcal{M}$ and $\mathcal{M} = \{1, \dots n\}$, design a switching supervisory function $h(t, \sigma(m,t))$ that chooses the vehicle model with the minimum switching cost $\sigma(m,t)$ at each time step t.

A novel switching cost based on the prediction error and the MPC optimisation time is designed to solve this problem. Then, using this switching cost, two different approaches for designing cost function are proposed. In the first approach, an FLS based switching supervisory system is designed. Next, an adaptive logic-based switching function is created.

4. Switching supervisor

The proposed SMPC is designed to switch between different vehicle models using a switching supervisor. First, a discussion on the calculation of prediction error is provided. Then, using these prediction errors, the design of a novel switching cost function is discussed. Finally, two different approaches for designing switching supervisors using two different switching rules are reported.

4.1. Model Prediction Error

Prediction accuracy of the used vehicle model plays an essential role in the efficacy of the MPC. As it is not possible to capture the full dynamics of a vehicle, the presence of prediction error is inevitable for any model-based control design. Figure 2 shows a depiction of the prediction error between an actual vehicle and a vehicle model. This model mismatch introduces uncertainty to the system. MPC has inherent robustness against uncertainties;

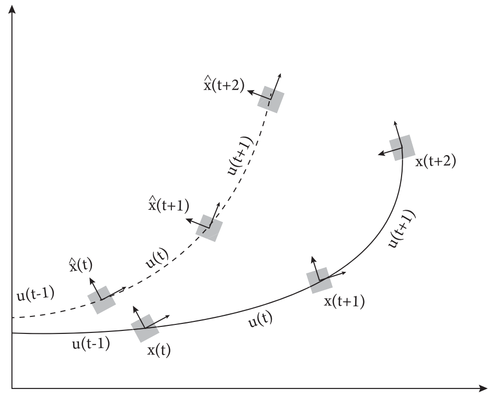

Figure 2. Depiction of the prediction error of a vehicle model. The dotted line shows the predicted trajectory using a model f_m and the solid line represents the trajectory of the controller vehicle for same control sequence. Here, $\hat{\mathbf{x}}$ represents the predicted state and $\mathbf{x}$ represents the measured vehicle states.

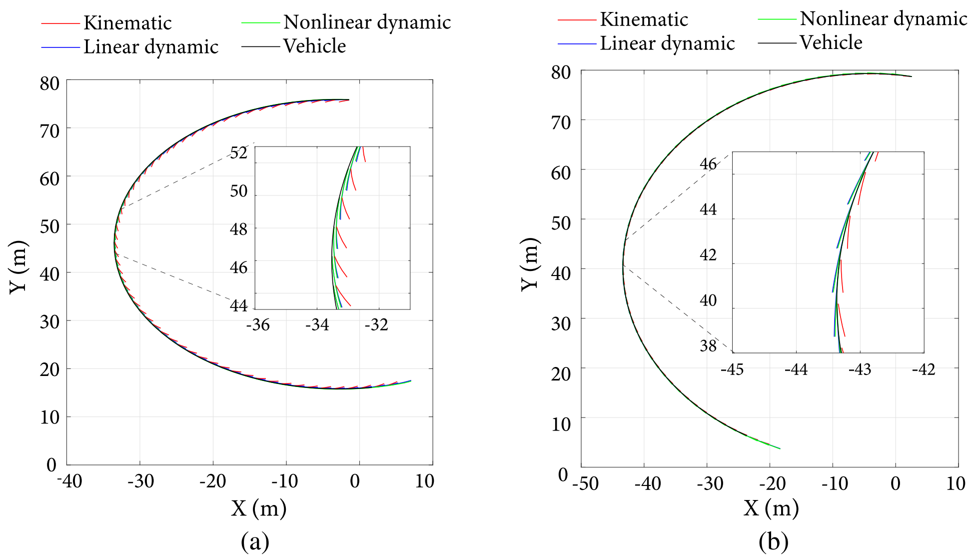

Figure 3. k-step ahead prediction comparison for (a) steering angle = 20 deg, velocity = 60 km/h, and (b) steering angle = 5 deg, velocity = 60 km/h.

however, the prediction accuracy has a significant effect on its performance. In some cases, a lower complexity model such as a kinematic model may provide an acceptable prediction error for the MPC to be effective. However, it may not be suitable for a wider range of operating regions. For those cases, a higher-order vehicle model is required.

In the context of MPC, another important aspect is the prediction error until a certain prediction horizon. The prediction model estimates the vehicle's state for a certain horizon to calculate the control actions. The feedback from the system is collected, and the prediction model is reinitialised with the current system state. If the vehicle model reflects good accuracy for an arbitrary k-step ahead, the model can significantly improve the performance of MPC. Here, we compare the k-step ahead state prediction for the three vehicle models. Figure 3(a) shows the vehicle coordinates for the k-step ahead state prediction for three vehicle models and the trajectory of the controlled vehicle for a steering angle of 20 deg and vehicle speed of 60 km/h. Similarly, Figure 3(b) shows the prediction error for much smaller steering angle of 5 deg with the same vehicle speed. In these figures, each line segment represents the predicted trajectory for the next k steps. After $k = 8$ time steps (same as the prediction horizon of MPC), all prediction models are reinitialised using the state of the controlled vehicle. From the observation of these figures, it is evident that the lower complexity vehicle model, such as the kinematic model, performs reasonably well for small steering angles, even at higher vehicle speed. However, a more complex model such as a nonlinear dynamic model generates more accurate results at larger steering angles.

The simulation of the controlled vehicle's motion has been performed in MATLAB/Simulink. The detail of this vehicle is discussed in Section 5.

4.2. Switching cost

The prediction error of each model is considered the most important switching criterion. The prediction errors are calculated based on the vehicle's measured and predicted states of each model. Two different prediction errors are calculated for all models. These include (i) position deviation error and (ii) heading error between the prediction model and the vehicle. These prediction errors are calculated as

$$\begin{aligned} \xi_d(f_m, t) &= \frac{1}{N_e} \sum_{t=1}^{N_e-1} (\|\hat{x}(t) - x(t)\|^2 + \|\hat{y}(t) - y(t)\|^2) \\ \xi_\psi(f_m, t) &= \frac{1}{N_e} \sum_{t=1}^{N_e-1} (\|\widehat{\psi}(t) - \psi(t)\|^2) \end{aligned} . \tag{10a–b}$$

Here, x and y are the vehicle's measured position coordinates, where as $\hat{x}$ and $\hat{y}$ are the vehicle's predicted position coordinates using the vehicle model $f_m (m = \{1, 2, 3\})$ for a specific time step.

Similarly, ψ is the measured vehicle yaw angle and $\widehat{\psi}$ is the predicted yaw angle. In addition, we use the average error for N_e number of time steps. At each time step, the most recent value is added, and the oldest value is discarded.

The other important aspect considered here is the solution time for each controller. When the controller switches to a model used for the state predictions by MPC, the solution time at each occurrence is recorded. The average value of these solution times is used for designing the final switching signal as

$$\xi_s(f_m, t) = \frac{1}{N_i} \sum_{t=1}^{N_i-1} T(f_m, t), \tag{11}$$

where $T(f_m, t)$ is the solution time of the MPC optimisation problem for a chosen vehicle model f_m at time t, and N_i is the number of time steps in which the vehicle model f_m was active. The switching cost for each vehicle model can be eventually calculated as

$$\sigma(f_m, t) = \theta_d \xi_d(f_m, t) + \theta_\psi \xi_\psi(f_m, t) + \theta_s \xi_s(f_m). \tag{12}$$

Here, θ is the corresponding weight of each cost term. These weights provide the flexibility for designing the appropriate switching cost. For example, a higher value of θ_s can

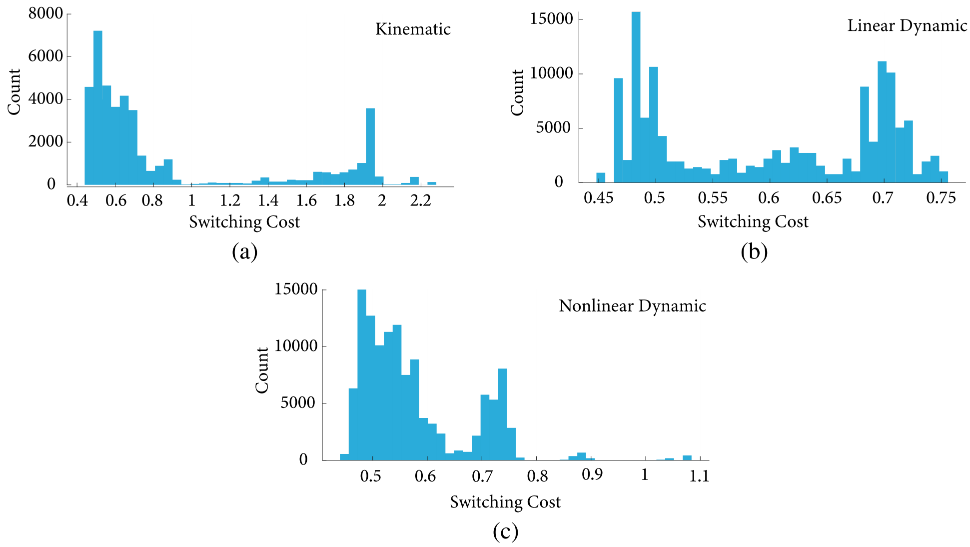

Figure 4. Switching cost for different models considered. (a) Kinematic model, (b) Linear dynamic model, and (c) Nonlinear dynamic model.

provide more emphasis on the solution for switching. For each vehicle model, this cost function is used to calculate the switching cost at each time step. Using these costs, appropriate switching criteria are defined. In this work, a trial-and-error method has been used to find the appropriate values of these weights.

4.3. Fuzzy logic-based switching

A fuzzy logic-based switching supervision system is proposed for effective switching between the vehicle models in SMPC. We use a Fuzzy Logic System (FLS), taking the switching costs of each model in (12) as input to provide a suitable model as output. For clarity, we denote the cost of the kinematic model, linear dynamic model and nonlinear dynamic model as σ_{kin}, σ_{lin} and σ_{nlin}, respectively. Besides, the output of the FLS is the vehicle model index m. Based on the input of FLS, first, a number of fuzzy sets are designed, and each of them is expressed using a Membership Function (MF). These fuzzy sets are connected by a set of fuzzy rules to provide an output. To define the fuzzy sets for input and their corresponding MFs, the knowledge of input values' ranges is initially required. To create a knowledge database, the vehicle is driven on three different roads, including (1) straight road (highway), (2) curved road (racetrack) and (3) city loop (a mixture of different turns). For each road, different manoeuvres such as single lane and double lane changes are performed frequently. The vehicle is driven with a speed between 30–60 km/h. At each time step, the switching cost of each model is calculated and recorded using (12). Figure 4 shows the range of cost values for each model.

Based on the knowledge base, the appropriate MFs for each input variable are designed. In this case, to define fuzzification, a trapezoidal function is chosen as the MF for all three

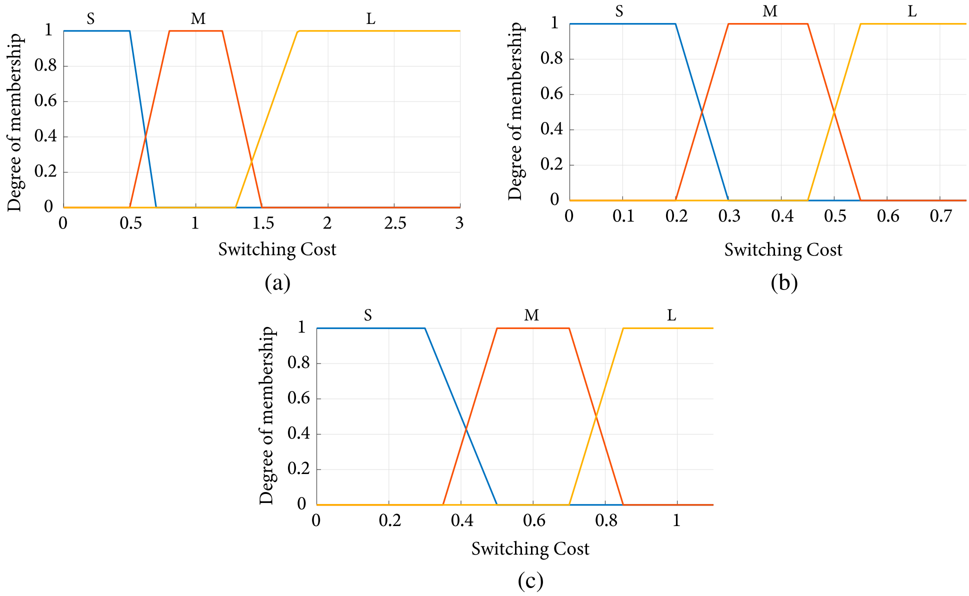

Figure 5. Membership functions for each input variable.

inputs. The membership function can be expressed as:

$$f_z(\sigma_m; a, b, c, d) = \max\left(\min\left(\frac{\sigma_m - a}{b - a}, 1, \frac{d - \sigma_m}{d - c}\right), 0\right). \tag{13}$$

Here, a, b, c and d are the parameters of the trapezoid.

For designing the FLS, the input for each model is expressed using three different fuzzy sets with various ranges. Each of these fuzzy sets is defined using the mentioned MF. The ranges of these sets are chosen based on the created knowledge base. Here, each input is separated into three regions using three membership functions where the regions include S = 'Small', M = 'Medium' and L = 'Large'. Figure 5 shows the input fuzzy sets with their degree of membership.

A set of rules are defined for the FLS to provide suitable output. In this case, the output is the vehicle model index. A total of 27 rules are created for the proposed FLS. These rules are designed to use the least complex model as frequently as possible. Few examples of the rules are shown following

- If σ_{kin} = S and σ_{lin} = S and σ_{nlin} = S then model = 1
- If σ_{kin} = M and σ_{lin} = S and σ_{nlin} = S then model = 2
- If σ_{kin} = L and σ_{lin} = M and σ_{nlin} = S then model = 3

Based on these rules, the FLS provides an output value between 1 and 3. Figure 6 shows the output of the FLS for various combinations of switching cost inputs. One of the inputs is kept fixed in these figures, and the other two values are varied. For model index output value less than 1, the kinematic model is selected. Similarly, for output between 1 and 2, the linear dynamic model is selected, and for output higher than 2, the nonlinear dynamic

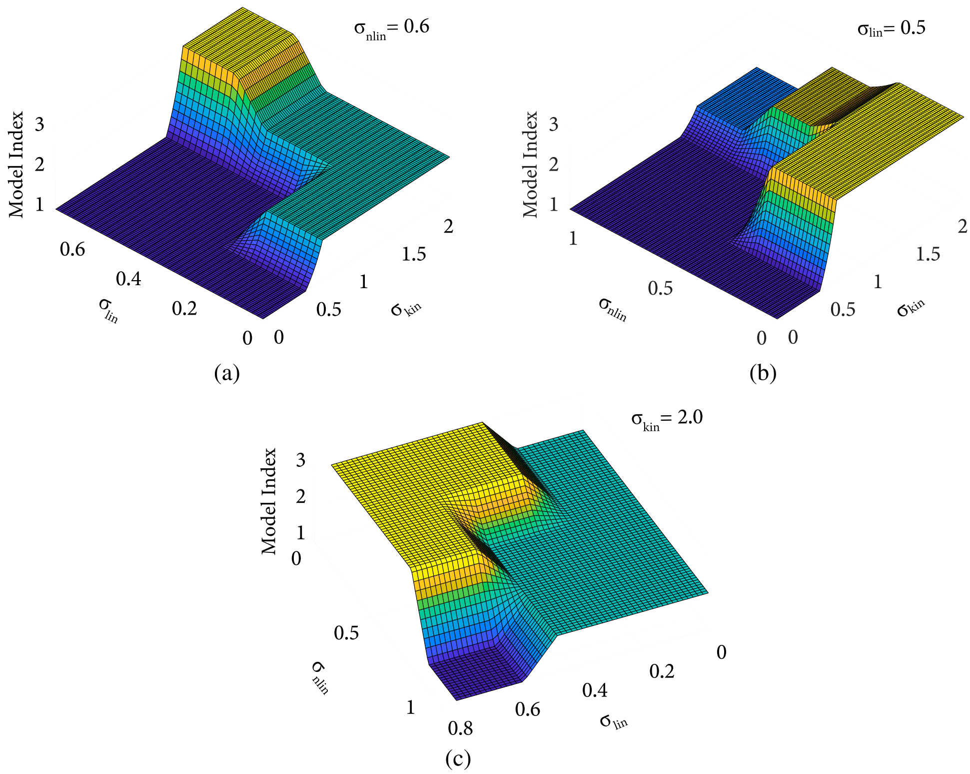

Figure 6. FLS output for different combination of inputs. Here, for each figure, one of the switching cost inputs is kept fixed, and other two are varied. Appropriate vehicle model is chosen based on model index value.

model is selected. We use a Sugeno Fuzzy inference system with AND rules. The final output of the FLS can be expressed as

$$h(\sigma) = \frac{\sum_{i=1}^{R} z_j \prod_{j=1}^{K} f_z(\sigma_m)}{\sum_{i=1}^{R} \prod_{j=1}^{K} f_z(\sigma_m)}, \tag{14}$$

where R is the number of rules, K is the number of inputs, z_j^l is the output value of j^{th} rule.

4.4. Adaptive switching

In the second approach, the switching cost for each vehicle model is calculated and compared to adopt the most suitable option online. The objective is to choose the least complex model based on the cost. The complexity of the vehicle models increases in this order: kinematic, linear dynamic model and nonlinear dynamic model. As a less complex model provides a shorter solution time, the objective is to use them as much as possible without significant degradation in the tracking performance.

In this approach, the vehicle operation starts with the least complex model and switches to a higher-order vehicle model only if the higher-order model's cost is less than a threshold

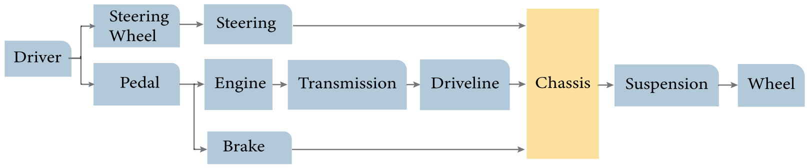

Figure 7. Complex vehicle model used for vehicle simulation.

value ϵ from the cost of the currently active model. If a lower-order model is available, then if the difference between the cost of the currently active model and the lower-order model is within a threshold value, the controller switches to a lower-order model even if the higher-order model has the lower cost.

At each time step, the currently active model is referred to as A, and the model providing minimum switching cost is denoted as M. For example, if the currently active model is the kinematic model, then $A = 1$. A switch to a higher-order only occurs when $\sigma(A) - \sigma(M) > \epsilon$. Here, ϵ is the threshold value assigning the switching operation. With a proper choice of ϵ, frequent switching can be avoided. An additional condition of $\sigma(A) - \sigma(M) \leq \tau$ and $\epsilon > \tau$ is also considered. In this case, if model $M < A$, then M is selected as the active model.

5. Switched model predictive control

In this section, the design and implementation process of the SMPC is discussed in detail. Here, the designed SMPC is implemented in a simulated environment using a complex vehicle dynamic system model. A complex 14 DoF vehicle model is used to represent the controlled vehicle to evaluate the controller. This simulated vehicle body has six spatial DoF. Furthermore, it has four wheels and each with two DoF (vertical and rolling). This vehicle simulation block also includes a number of other subsystems such as driveline, engine, hydraulic components and steering system. To be noted that, this vehicle model is too complex for it to be used by the MPC. The block diagram of Figure 7 shows the components of the complex vehicle model representing an actual vehicle to be controlled by the SMPC.

For the controlled vehicle discussed here, the states x, y, ψ, v_x, v_y values are measured. For the implementation of the proposed SMPC of (9), a cost function capable of combined longitudinal and lateral control of the AV is designed. Along with the conventional tracking error of position and heading deviation, the longitudinal velocity is involved in the path tracking task. The cost function for the SMPC is expressed as

$$\begin{aligned} J = \sum_{k=1}^{N_p-1} & w_1\|\hat{x}(t+k|t) - x_{ref}(t+k|t)\|^2 k \\ & + w_2\|\hat{y}(t+k|t) - y_{ref}(t+k|t)\|^2 \\ & + w_3\|\widehat{\psi}(t+k|t) - \psi_{ref}(t+k|t)\|^2 \\ & + w_4\|\hat{v}_x(t+k|t) - v_{ref}(t+k|t)\|^2 \\ & + w_5\|\Delta\delta\|^2 \end{aligned} \quad . \tag{15}$$

Here, $\hat{x}$ and $\hat{y}$ are estimated position coordinates, $\widehat{\psi}$ is the estimated yaw angle, $\hat{v}_x$ is the estimated velocity of the vehicle using the prediction model and δ is the steering angle

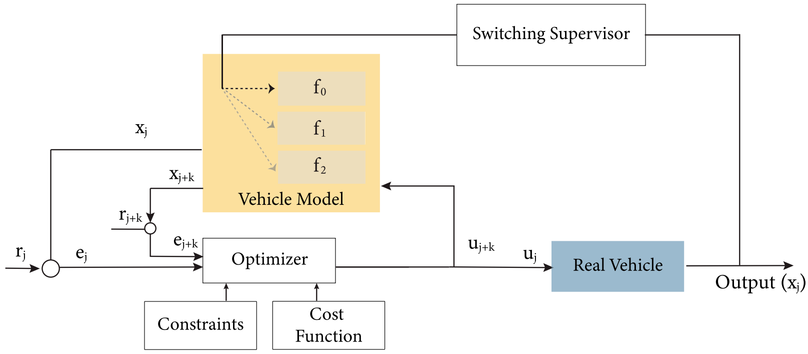

Figure 8. Schematics of the proposed SMPC.

input. Likewise, x_{ref} and y_{ref} are the reference position coordinates, ψ_{ref} is the reference yaw angle and v_{ref} is the reference velocity. In addition, $\mathbf{w} = [w_1, w_2, w_3, w_4, w_5]$ is the weight vector for each term in the cost function. Figure 8 shows the schematics of the SMPC approach.

In the SMPC implementation, three vehicle models discussed in Section 2.1 are used. The switching cost for each model is calculated using (12). As previously mentioned, two different approaches are proposed for designing the switching function. In the first approach, the SMPC uses the FLS-based switching function discussed in (4.3). Here, we refer to the approach as 'FL-SMPC'. In this case, the switching cost for each vehicle is initially fuzzified. Next, the FLS system of (14) is used to get the model index value. Finally, based on the model index value, an appropriate vehicle model is selected. In the second approach, the adaptive switching rule discussed in Section 4.4 is used by the SMPC. The algorithm for the adaptive switching based SMPC is shown in Algorithm 1. This approach is referred to as 'ASMPC'.

For the implementation of these SMPCs, the same fixed prediction horizon of $N_p = 8$ and the control horizon of $N_c = 8$ is used. As both of the control and prediction horizon lengths affects the computation time required for MPC solution, we chose a fixed value for each of them. Here, we chose the maximum control horizon possible (which is equal to the prediction horizon). This allows to have a more consistence comparison of results between the proposed and conventional MPCs without considering the effect of prediction and variations of control horizons.

A number of different control parameters have been tuned for the implementation of the SMPC. A complete list of control parameters and their corresponding tuned value is shown in Table 1. The implementation of the MPC is conducted with the combination of MATLAB and Simulink environment.

6. Results

The performance of the FL–SMPC is initially evaluated. As the path profile has a significant effect on vehicles' handling performance [24], for a proper comparison, a road with varying curvatures is considered. We assume that there are several obstacles on the road and the path planner unit provides a path with changing curvature to avoid obstacles. Figure 9

Algorithm 1: Proposed algorithm for adaptive switching rule-based SMPC

Input: Initial state $\mathbf{x}_0$ (vehicle state measurement), set of vehicle models f_m where $m = \{1,2,3\}$, prediction horizon, N_p, cost function $J(\cdot)$, switching cost function $\sigma(\cdot)$

```
Form the optimisation problemin(9).
while MP Cisrunning do
    Measure current state x(t);
    calculate switching cost σ(f_m) for each model f_m;
    Find model M with minimum switching cost where current active modelis A;
    if M != A then
        if σ(A) − σ(M) > ϵ then
            Switch to model M;
        end
        if σ(A) − σ(M) ≤ τ then
            if M<A then
                Switch to model M;
            end
        end
    end
    Start the optimization problem;
    while optimization is running do
        for i=1: N_p do
            Estimate next state x̂(t + i) using active f_m;
        end
        Find optimal control sequence;
    end
    Apply only u(t|t);
    t=t+1;
end
```

shows the tracking performance of the vehicle using the FL–SMPC approach for a reference velocity of 60 km/h. In addition, the choice of models in the different sections of the trajectory during the tracking task is shown in Figure 10. This result is shown for three different velocities. The colour map of the selected vehicle model is shown where blue is the kinematic model, green is the linear dynamic model and yellow is the nonlinear dynamic model.

The performance of the ASMPC is additionally evaluated. Figure 11 shows the trajectories of the vehicle for different, forward speeds using the ASMPC. Here, the path tracking

Table 1. List of tuned parameter values.

Parameter	Tuned value
N_p	8
N_e	10
w_1	0.5
w_2	0.5
w_3	10
w_4	2
w_5	0.1
ϵ	0.04
τ	0.015
θ_d	1
θ_ψ	5
θ_s	3.5

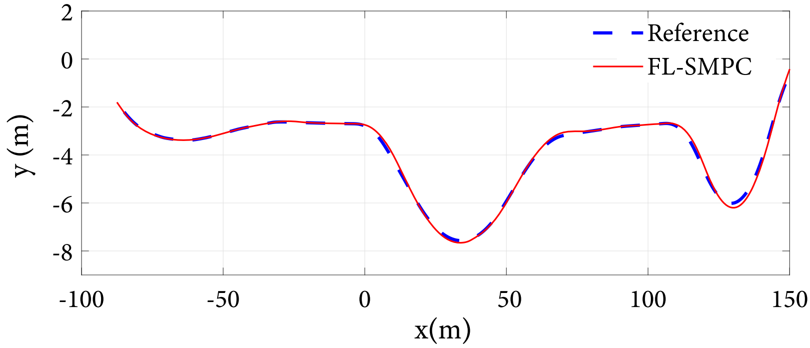

Figure 9. Tracking performance of the vehicle using the FL–SMPC approach.

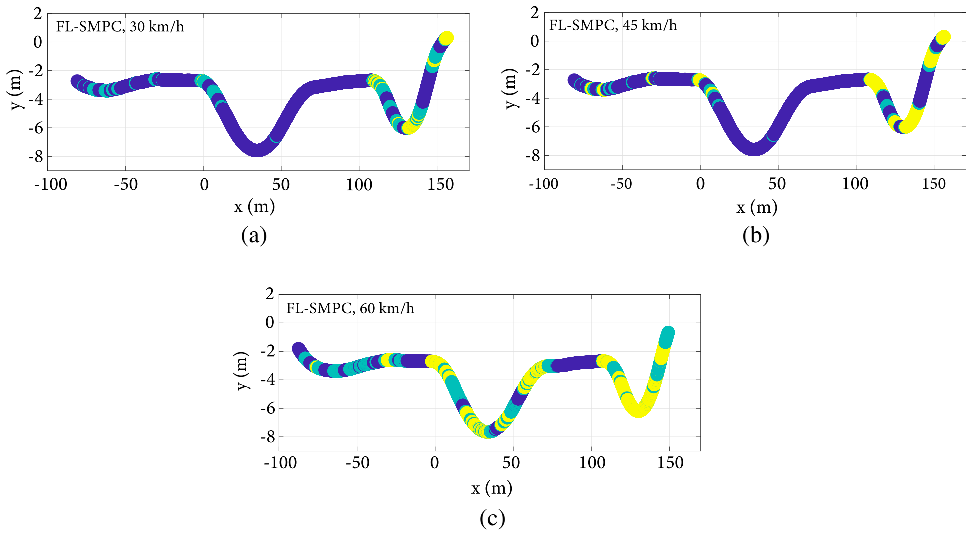

Figure 10. Trajectories of AV with a colour map of vehicle model using FL–SMPC. In these plots, blue shows the kinematic model, green shows the linear dynamic model and yellow shows the nonlinear dynamics model. The reference velocity of the vehicle is (a) 30 km/h, (b) 45 km/h and (c) 60 km/h.

task is performed for three different speeds of 30, 45 and 50 km/h. Moreover, Figure 12 shows the active models used by the ASMPC for generating the trajectories.

To effectively evaluate the performance of the controller, a longer and more practical path with different curvatures is chosen. The accuracy of the kinematic model reduces when the vehicle is operating at a high velocity with a large steering angle. In this case, a more complex model, such as a linear or a nonlinear dynamic model, is capable of capturing the vehicle's behaviour more accurately. The curvature of the road plays an important role as a larger steering angle is required for the vehicle to accurately follow a high curvature road. Therefore, it is more realistic to evaluate the performance of the controller on the road with various curvatures.

The chosen reference road provides the opportunity to test the performances of the proposed controller for different road curvatures. An efficient controller should provide

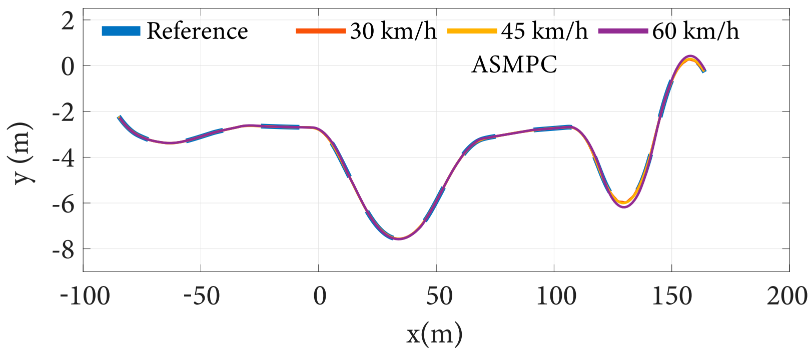

Figure 11. Tracking performance of the vehicle using the ASMPC approach.

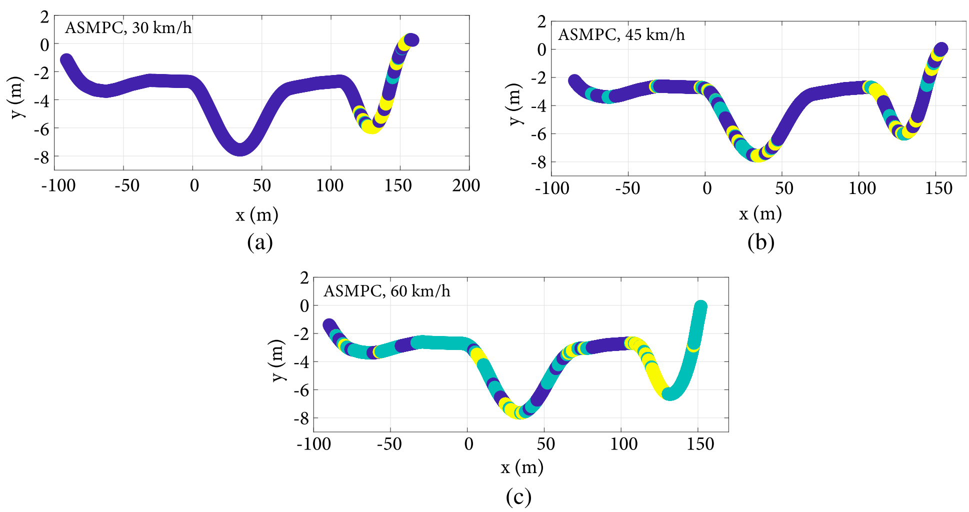

Figure 12. Trajectory of the AV with a colour map of vehicle model used by ASMPC. Blue indicates the kinematic model, green indicates the linear dynamic model and yellow represents the nonlinear dynamic model. The reference velocity of the vehicle is (a) 30 km/h, (b) 45 km/h and (c) 60 km/h.

accurate tracking performances for different road curvatures. The selected road has different segments, and each segment has a specific curvature. This path can be partitioned into three different segments based on the curvature. Segment-2 has smaller curvatures than segment-1, where as the curvatures for segment-3 are much larger than other segments. Figure 13(a) shows the tracking performance of the FL–SMPC and the corresponding choice of different vehicle models for the different segments of the test path. Similarly, in Figure 13(b) the performances and the model choice are shown for the ASMPC.

The controller's performance is also compared to the conventional MPCs, where only a single vehicle model is used. Here, three different MPCs are formulated based on the three models used in this study. We refer to the kinematic model-based MPC as 'KMPC' (using the model described in Equation (3)), the linear dynamic model-based MPC as 'LMPC' (using model described in Equation (4–6)), and the nonlinear dynamic model-based MPC as 'NMPC' (using model described in Equation (4,7,8)). Figure 14 shows the performance

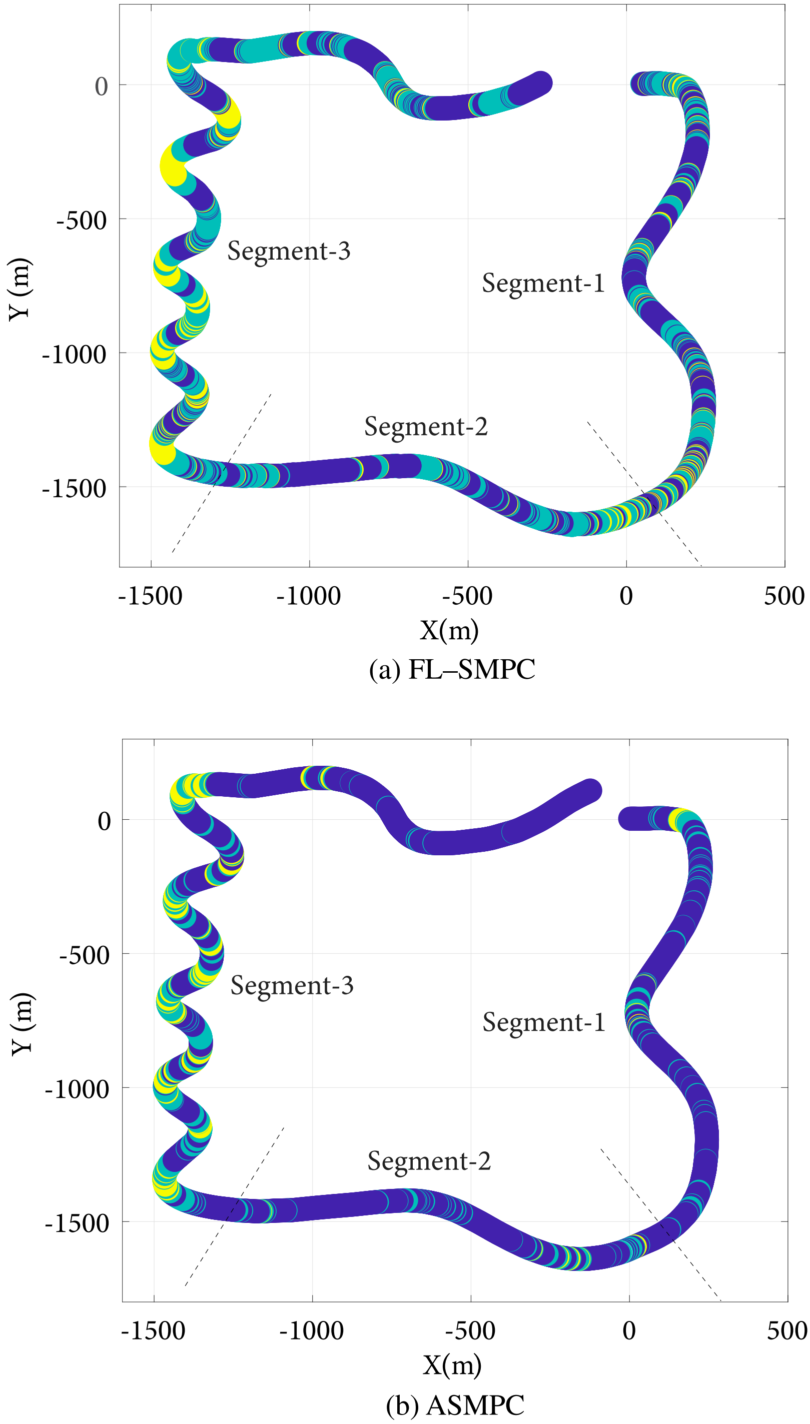

Figure 13. Trajectory of the AV with a colour map of vehicle model used by (a) FL–SMPC and (b) ASMPC. Blue indicates the kinematic model, green indicates the linear dynamic model and yellow represents the nonlinear dynamic model. The reference velocity of the vehicle is 60 km/h.

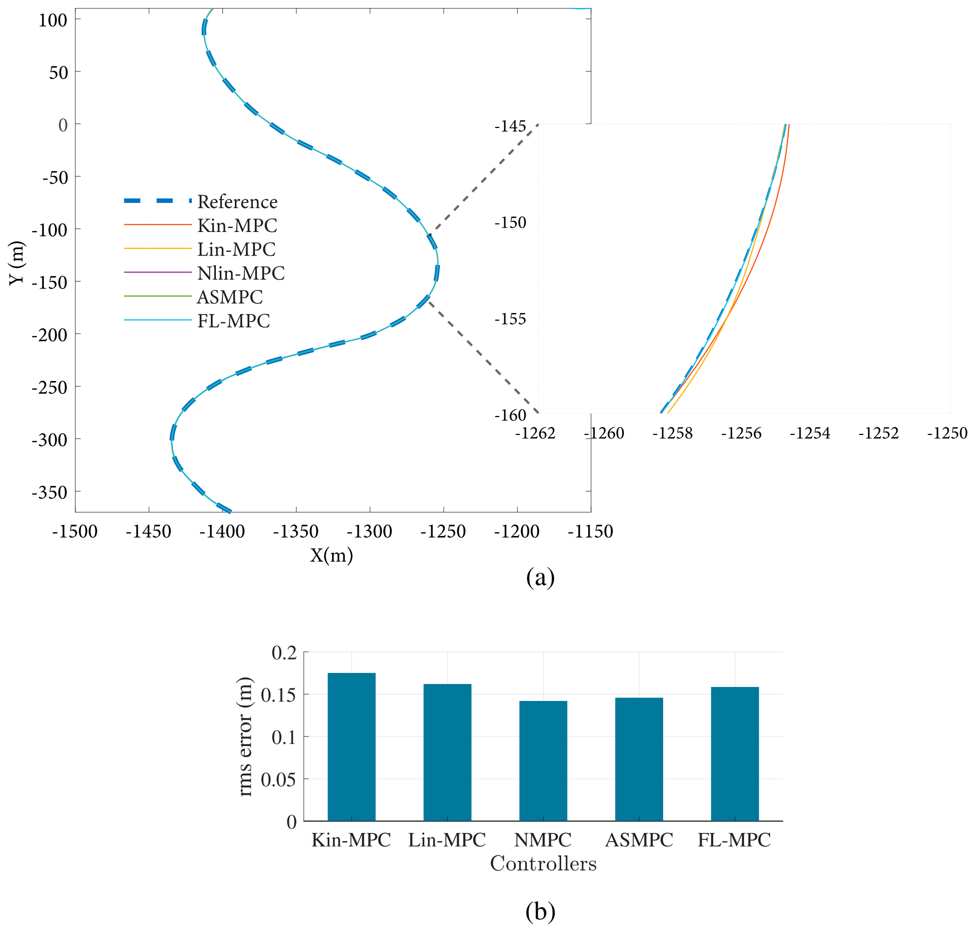

Figure 14. Comparison of proposed FL–SMPC and ASMPC with conventional MPCs. (a) Trajectories generated by different controllers. Only a portion of the trajectory is shown for clarity. (b) Tracking distance error for different controllers.

comparison of these controllers with the proposed SMPC controller. In Figure 14(a), a comparison of the trajectories generated by the proposed and conventional MPCs are shown. In addition, Figure 14(b) shows the tracking error comparison of these controllers. Finally, a comparison of MPC solution time for the proposed FL–SMPC, ASMPC with the NMPC is shown in Figure 15. Here, the solution time is recorded at each time step. For each tracking task, the test path shown in Figure 13(a) with the longitudinal velocity of 60 km/h is chosen. This velocity is chosen based on the review report of the Victorian Government, Australia, where it is reported that 60 km/h is the most common speed limit for Australian roads with little to no pedestrian activities with a high number of access points [25].

Finally, the proposed controllers have been compared with a conventional SMPC. We implemented the conventional SMPC using the design approach of [17]. Here, only two vehicle models, a kinematic and a dynamic vehicle model, were used. The switching law was designed based on the vehicle speed and steering angle. Figure 16 shows the performance

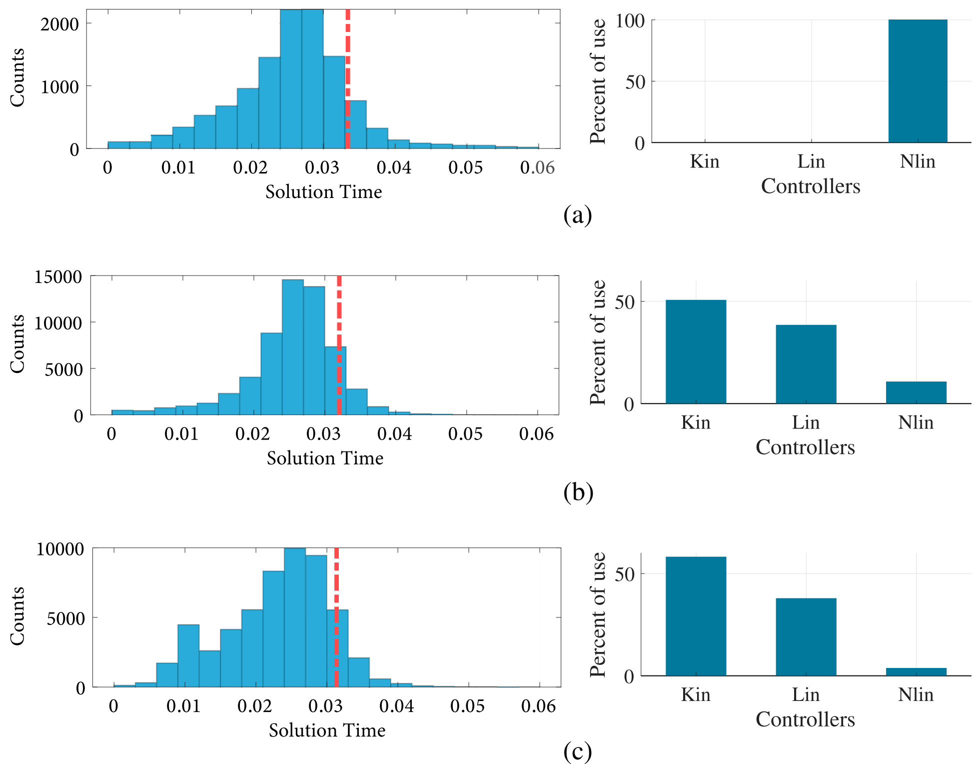

Figure 15. MPC solution time and the percentage of time that a vehicle model was chosen by the controller during the same tracking task: (a) conventional NMPC, (b) FL–SMPC and (c) ASMPC. 90% of the solution time samples are below the dotted red line.

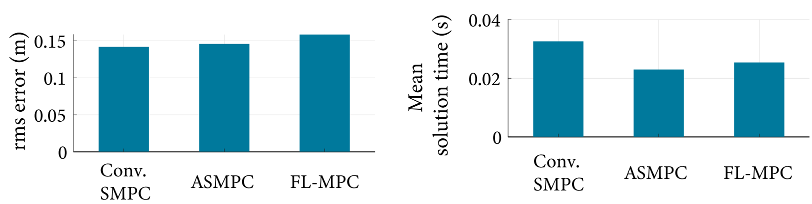

Figure 16. Comparison of proposed SMPCs with conventional SMPC.

comparison. In Figure 16(a), the proposed controllers' tracking error are compared with the conventional SMPC.

In Figure 16(b), a comparison of the average MPC solution time for all controllers is shown.

7. Discussion

The main objective of this work is to develop an MPC framework that can effectively utilise vehicle prediction models with various levels of complexity. As a complex vehicle model increases the solution time in the MPC, the selective use of the more complex models can improve the computational efficiency of the MPC. In addition, we designed a novel switching cost, which provides more flexibility in terms of trade-off between tracking error and increased computational efficiency. We proposed two solutions to accomplish this. The use of an FLS-based supervisor provides an effective way to switch between prediction models. From the results of Figures 9 and 10, it is evident that this technique is capable of accomplishing tracking tasks on the road with various curvatures. In addition, these results indicate the frequent use of the least complex kinematic model and selective use of the most complex nonlinear dynamic model.

The primary objective of this work is to accommodate appropriate switching between vehicle models with different fidelities during the path tracking operation. Our proposed technique leads to the selection of the less complex model more often than, the more complex models without any significant increase in tracking error. Based on the performance comparison, it is evident that the average MPC solution time is shorter than the conventional MPCs for the proposed controller. This also reflects the fact that the less complex models are used more frequently.

Compared to other reported works, the contribution of our is twofold. Firstly, most of the previous approaches depend on the system's operating regions and different control laws used in these regions. In the context of path tracking tasks, SMPC has been dominantly used based on the vehicle dynamic responses such as tyre force or operation mode. In our proposed work, a single control law, which does not need to be changed based on the switching surface, is sufficient. Secondly, in all reported works, knowledge of the switching surface and performances at different operation regions must be known before starting the system's operation. This is normally formulated during the controller's design process. However, in our proposed ASMPC approach, knowledge of the switching surface and performances at different operation regions is not required. This allows flexibility for adjusting the weight between the tracking error and solution.

One of the shortcomings of the FLS-based approach is that it requires a knowledge of the switching cost' range for each model. Under these circumstances, the online adaptive switching law for designing the switching supervisor can provide a suitable alternative. Results of Figure 11 show the effectiveness of the ASMPC approach for path tracking tasks. In addition, results of Figure 12 prove the selective choice of models based on complexity.

From the comparisons of these two approaches, it is apparent that both controllers can complete path tracking tasks for different vehicle speeds. Moreover, from the model choice comparison of these controllers, the kinematic model is sufficient for most of the tracking tasks at lower vehicle speeds (30 km/h). However, dynamic models are required when the path curvature is large. At a higher speed than 45 km/h, the kinematic model is chosen for the majority of the task; however, the other models are chosen more frequently than the lower vehicle speed. Finally, when the vehicle is travelling at 60 km/h, the dynamic models are dominantly in use. It is noteworthy that the FL–SMPC uses the most complex vehicle model more often than the ASMPC, even though they almost provide similar tracking performances.

The performances of both FL–SMPC and ASMPC have been compared to conventional MPC. The tracking error of Figure 14 shows that both the controllers provide similar tracking performances compared to the conventional MPCs, including NMPC. However, these proposed techniques significantly reduce the computational time compared to the NMPC.

From the results of Figure 15, it is evident that both controllers require significantly shorter solution time than the conventional NMPC.

Finally, the proposed FL–SMPC and ASMPC have been compared to a conventional SMPC. From the tracking error comparison in Figure 16(a), both the proposed controllers provide similar performances compared to the conventional SMPC. However, from the solution time comparison in Figure 16(b), it is evident that the proposed controllers significantly reduce the average solution time.

8. Future works

A number of extensions and improvements are in the scope of our future works. Among them, the most important one is to evaluate the feasibility of adaptive prediction and control horizon for the proposed SMPCs. Currently, our approaches reduce the solution time of the MPC and provide an opportunity to use a longer prediction horizon when a simpler vehicle is in effect. In our future effort, a rigorous investigation will be carried out to utilise an adaptive prediction and control horizon, while maintaining the stability of the controller and the vehicle. In addition, we plan to extend the scope of the work by integrating a learning-based vehicle dynamics model adopted in our previous work [26].

In this work, the weight terms in (12) and the values of ϵ and τ have been chosen based on a trial-and-error. As a potential future effort, a learning-based MPC approach [27] will be adopted to estimate the appropriate values of these parameters. As the PTC of an AV can be used to increase the safety and comfort of its passenger [20]. This approach will be used to generate more comfortable motion by improving the handling performance of the vehicle.

9. Conclusion

Path tracking controller (PTC) is a crucial part of the Autonomous Vehicles (AVs) which is responsible for implementing the planned path and controlling the vehicle dynamics. The Model Predictive Control (MPC) has shown to be a capable solution for the PTC of AVs. One of the most important aspects of MPC is the proper choice of the vehicle model to predict the future states of the vehicle. A simple model can potentially reduce the computational cost; however, this will be achieved at the cost of less accuracy and more uncertainty. A more complex model, however, can provide improved performance with increased computational cost.

This work proposes a new framework for effective utilisation of vehicle prediction models in terms of accuracy and computational efficiency. This is achieved by using three vehicle models with different levels of complexity and fidelity in a Switched MPC (SMPC) framework. The switching is performed to shift among these models efficiently. A novel switching cost function is used based on the prediction error of each model and the MPC solution time. Using this switching cost function, two different approaches for designing switching supervisors have been adopted: (1) Fuzzy logic-based switching and (2) adaptive switching.

The proposed controllers' performances have been evaluated in a simulated environment and on roads with varying curvatures for different vehicle speeds. The simulation outcomes show that the controllers are capable of performing accurate path tracking. Furthermore, the controllers' performances have been compared to the conventional MPC. The comparison shows that the tracking performances are comparable to the conventional MPC. Nevertheless, the proposed controllers are more computationally efficient due to the selective use of complex vehicle models.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Rokonuzzaman M, Mohajer N, Nahavandi S, et al. Review and performance evaluation of path tracking controllers of autonomous vehicles. IET Intel Transport Syst. 2021;15(5):646–670.
[2] Campbell SF. "Steering control of an autonomous ground vehicle with application to the DARPA Urban Challenge," Ph.D. dissertation, Massachusetts Institute of Technology, 2007. [Online]. Available: http://dspace.mit.edu/handle/1721.1/42301
[3] Thrun S, Montemerlo M, Dahlkamp H, et al. Stanley: The robot that won the DARPA Grand challenge. In: Buehler M, Iagnemma K, Singh S, editor. The 2005 DARPA Grand challenge. Springer tracts in advanced robotics, vol 36. Berlin: Springer; 2007. p. 1–43.
[4] Alcala E, Puig V, Quevedo J, et al. Autonomous vehicle control using a kinematic lyapunov-based technique with LQR-LMI tuning. Control Eng Pract. 2018;73:1–12. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066117302721
[5] De Luca A, Oriolo G, Samson C. Feedback control of a nonholonomic car-like robot. In: Laumond JP, editor. Robot motion planning and control. New York, NY: Springer-Verlag; 1998. p. 170–253.
[6] Chwa D. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. IEEE Trans Control Syst Technol. 2004;12(4):637–644.
[7] Li L, Jia G, Chen J, et al. A novel vehicle dynamics stability control algorithm based on the hierarchical strategy with constrain of nonlinear tyre forces. Veh Syst Dyn. 2015;53(8):1093–1116.
[8] Falcone P, Tseng HE, Borrelli F, et al. MPC-based yaw and lateral stabilisation via active front steering and braking. Veh Syst Dyn. 2008;46(sup 1):611–628.
[9] Rokonuzzaman M, Mohajer N, Nahavandi S. "NMPC-based controller for autonomous vehicles considering handling performance," in *IEEE 7th International Conference on Control, Mechatronics and Automation*, Delft, Netherlands, Nov. 2019, pp. 266–270.
[10] Mohajer N, Nahavandi S, Abdi H, et al. Enhancing passenger comfort in autonomous vehicles through vehicle handling analysis and optimization. IEEE Intell Transp Syst Mag. 2021;13(3):156–173.
[11] Zhang L, Zhuang S, Braatz RD. Switched model predictive control of switched linear systems: feasibility, stability and robustness. Automatica (Oxf). 2016;67:8–21.
[12] Magni L, Scattolini R, Tanelli M. Switched model predictive control for performance enhancement. Int J Control. 2008;81(12):1859–1869.
[13] Bemporad A, Muñoz de la Peña D. Multiobjective model predictive control. Automatica (Oxf). 2009;45(12):2823–2830.
[14] Müller MA, Allgöwer F. Improving performance in model predictive control: switching cost functionals under average dwell-time. Automatica (Oxf). 2012;48(2):402–409.
[15] Mhaskar P, El-Farra NH, Christofides PD. Robust hybrid predictive control of nonlinear systems. Automatica (Oxf). 2005;41(2):209–217.
[16] Anderson JR, Ayalew B, Weiskircher T. "Modeling a professional driver in ultra-high performance maneuvers with a hybrid cost MPC," in 2016 *American Control Conference (ACC)*, Jul. 2016, pp. 1981–1986, iSSN: 2378-5861.

[17] Zhang K, Sprinkle J, Sanfelice RG. Computationally aware control of autonomous vehicles: a hybrid model predictive control approach. Auton Robots. 2015;39(4):503–517. doi:10.1007/s10514-015-9469-5.

[18] Di Cairano S, Tseng HE, Bernardini D, et al. Steering vehicle control by switched model predictive control. IFAC Proceedings Volumes. 2010;43(7):1–6.

[19] Mohajer N, Abdi H, Nelson K, et al. Vehicle motion simulators, a key step towards road vehicle dynamics improvement. Veh Syst Dyn. 2015;53(8):1204–1226.

[20] Mohajer N, Abdi H, Nelson K, et al. "On the simulation-based objective estimation of road vehicle ride comfort," in 2017 *IEEE International Conference on Mechatronics (ICM)*, 2017, pp. 159–165.

[21] Mohajer N, Asadi H, Nahavandi S, et al. "Evaluation of the path tracking performance of autonomous vehicles using the universal motion simulator," in 2018 *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2018, pp. 2115–2121.

[22] Rajamani R. Lateral vehicle dynamics. In: Rajamani R, editor. Vehicle dynamics and control. Boston, MA: Springer; 2012. p. 15–46.

[23] Pacejka HB. Tire and vehicle dynamics. 3rd ed. Oxford: Butterworth-Heinemann; 2012.

[24] Mohajer N, Rokonuzzaman M, Nahavandi D, et al. "Effects of road path profiles on autonomous vehicles' handling behaviour," in 2020 *IEEE International Systems Conference (SysCon)*, Montreal, Canada, Apr. 2020, p. to appear.

[25] Vic Roads Australia. (2012, August). "Victorian Speed Limit Review". [Accessed Aug. 13, 2021]. [Online]. Available: https://www.warrnambool.vic.gov.au/road-safety

[26] Rokonuzzaman M, Mohajer N, Nahavandi S, et al. Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking. IEEE Access. 2021;9:128233–128249.

[27] Rokonuzzaman M, Mohajer N, Nahavandi S, et al. "Learning-based Model Predictive Control for Path Tracking Control of Autonomous Vehicle," in 2020 *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2020, pp. 2913–2918, iSSN: 2577-1655.