
Article:
Abdullah, M., Zainuddin, M.A.-S., Ahmad, S. et al. (2023) Performance comparison between centralized and hierarchical predictive functional control for adaptive cruise control application. International Journal of Automotive and Mechanical Engineering, 20 (4). 10808 -10820. ISSN: 2229-8649

https://doi.org/10.15282/ijame.20.4.2023.01.0836

INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOL. 20, ISSUE 4, 2023, 10808 – 10820
DOI: https://doi.org/10.15282/ijame.20.4.2023.01.0836

RESEARCH ARTICLE

Performance Comparison Between Centralized and Hierarchical Predictive Functional Control for Adaptive Cruise Control Application

M. Abdullah[1]*, M.A.S. Zainuddin[2], S. Ahmad[3], and J.A. Rossiter[4]

[1,2,3]Department of Mechanical and Aerospace Engineering, International Islamic University Malaysia, Jalan Gombak, 53100, Kuala Lumpur, Malaysia
[2]Department of Automotive Engineering Technology, Kolej Kemahiran Tinggi MARA, Masjid Tanah, 78300 Melaka, Malaysia
[3]College of Engineering and Technology, University of Doha for Science and Technology, Doha, Qatar
[4]Department of Automatic Control and System Engineering, The University of Sheffield, Mappin Street, S1 3JD, UK

ABSTRACT - This paper presents a performance comparison between two different control structures of Predictive Functional Control (PFC), namely centralized and hierarchical architectures for an Adaptive Cruise Control (ACC) system. Centralized PFC utilizes a linearized model of the vehicle longitudinal dynamics to compute the control law, while hierarchical PFC uses a simple kinematic model with inverse vehicle longitudinal dynamics. Based on the simulation results, both control structures produced a satisfactory performance. However, in the presence of disturbances such as road slopes and wind gusts, the centralized PFC becomes more sensitive and conservative than the hierarchical PFC. The main reason is that the linearization model used in centralized PFC is only effective around the selected nominal operating points. Since the decision-making process depends on the internal model performance, centralized PFC cannot compensate for this effect. Besides, it also takes a longer computation time than hierarchical PFC since more mathematical operations must be solved than the kinematic model, especially when physical and operational constraints are considered. The standard performance parameters such as Root Mean Squared Error (RMSE), settling time, and rise time are also used for the analysis. These findings can become a solid justification for using the hierarchical PFC structure in designing an ACC system for future work.

ARTICLE HISTORY
Received : 14th Mar. 2023
Revised : 20th Sep. 2023
Accepted : 8th Nov. 2023
Published : 26th Dec. 2023

KEYWORDS
Cruise control
Predictive control
Model predictive control
PID control
Tuning performance

1.0 INTRODUCTION

Adaptive Cruise Control (ACC) is a system that automates the acceleration and deceleration of a vehicle according to the driver's desired speed while maintaining a safe following distance with the lead car, even in the presence of uncertainties such as steep terrain, changing passenger weight or strong wind gust. This system can provide many benefits to a user, such as reducing body fatigue, improving fuel consumption, and preventing accidents [1]. With the advancement of microprocessor and sensor technologies, more features are being added, such as stop-and-go function, terrain estimation, and control optimization. According to the Society of Automotive Engineers (SAE), ACC is the most basic function required by an autonomous vehicle, which is considered as level 1 [2]. Thus, it is essential to keep improving this system for the future development of autonomous vehicles.

One of the critical areas for development is the control algorithm, which is responsible for computing the best control action for the decision-making process. Commercialized ACC systems often use traditional control algorithms such as PID controller and Fuzzy Logic Controller (FLC) [3]. Nevertheless, there are several limitations to these controllers. For example, a conventional PID controller can only be tuned based on the current measurement; thus, no future prediction is incorporated into the decision-making process. Therefore, a switching strategy is often proposed either for speed control or distance control [4]. As for the FLC, a customized logic rule is needed to design the decision-making process. Although it works in most applications, the rule cannot be generalized to other types of vehicles [3]. In general, these traditional controllers have a robustness issue, where special modifications or integration between the two controllers is often proposed to improve their performance [5]. Additionally, the tuning method and control structure are not that straightforward, as many trial-and-error procedures must be tested [6,15].

Most of the recent proposals on ACC concepts use Model Predictive Control (MPC) [3] since it can offer better response and systematic design procedures. MPC is an optimal control method that computes a control action based on a future prediction of the internal model [16]. The difference between prediction and the desired target is typically optimized using a quadratic cost function to get the optimal control input. Besides, this controller can also consider multiple control objectives, which is beneficial for the ACC system, such as fast response, safe distancing, and good fuel consumption [5]. Many works have demonstrated this concept [3,5,7], but most are still in the simulation phase, and only a few managed to implement it in a prototype system. There is a potentially high price for implementing MPC, where a higher computation load is needed to solve many mathematical operations in the algorithm [8]. In return, a more expensive microprocessor is required to cope with the demand. In the future, a full-spec Autonomous car must consider more

*CORRESPONDING AUTHOR | M. Abdullah | ✉ mohd_abdl@iium.edu.my

complex functions such as lane changes, image processing, mapping, and others. If the computation time for the basic function of the ACC system can be kept to a minimum, it will bring a considerable advantage to the entire system.

Indeed, many efforts are proposed to reduce the computation load of MPC either by modifying the algorithm or the control structure [16]. Several authors have demonstrated by utilizing the Laguerre function in the control law, where a large control horizon can be represented using a smaller horizon to reduce the computation load [9]. Nevertheless, an additional tuning parameter is needed. Conversely, in ACC applications, a hierarchical structure can be used. The control law of MPC can be formulated based on a simple kinematic model with time delay, while an inverse kinetic model can be used to compute the control law [3]. Thus, all the prediction and optimization will be simplified while reducing the computational load. However, even with this modification, the computation is still considered heavy, especially when constraints on the vehicle, such as acceleration limits and safe distancing, are considered.

Alternatively, several works have proposed using Predictive Functional Control (PFC), a simplified MPC subvariant, to design an ACC system [9-10]. PFC uses a similar concept to MPC, except that it minimizes a single target error with direct inversion of equality between target trajectory and output prediction at a specific coincidence horizon instead of using a quadratic cost function. Hence, the controller only needs minimum computation effort, yet due to the simplicity of the objective, one can argue that the control input is suboptimal. However, for the single input single output application, the performance of PFC is acceptable considering its low computation time and often giving a better response than the traditional PID controller [9]. Nevertheless, there is a lack of clarity over which method should be used with the PFC, either the centralized or hierarchical structures, where both approaches have their advantages and disadvantages.

The prime aim of this work is to provide a formal comparison between the two control structures for the ACC application. The findings can become a solid justification for designing the controller in future applications. The paper is organized such that Section 2 provides the mathematical model for longitudinal vehicle dynamics. Section 3 presents the formulation for Centralized PFC, and Section 4 for Hierarchal PFC. Section 5 discusses the result and analysis. Section 6 concludes the work.

2.0 VEHICLE LONGITUDINAL DYNAMICS

This work will consider a standard vehicle longitudinal dynamic as the plant. The input to the model is the traction force, and the output is the vehicle speed. For simplicity of simulation, the power train and braking dynamics will not be considered, although a simple extension can be implemented. The actual accelerator pedal pressing can be determined from the positive traction force by using specific engine mapping [4]. Similarly, from negative traction force, the actual brake pressing can be calculated using a linear brake equation [4]. Figure 1 shows the free-body diagram representing the car's primary force distribution.

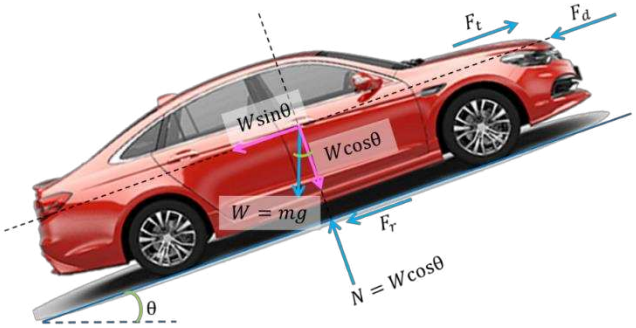

Figure 1. Free body diagram for longitudinal vehicle dynamics [9]

The positive horizontal direction is assumed to be on the right, and the positive vertical direction is upward. The equation of motion is derived based on Newton's second law, where:

$$m\frac{dv}{dt} = F_t - F_w - F_r - F_d \tag{1}$$

Given that m is the vehicle mass, v is the vehicle speed, F_t is the input traction force, F_w is the weight, F_r is the aerodynamic force, and F_d is the rolling resistance force. Each of these forces has its equation, and based on a lumped parameter assumption; they can be represented as:

$$F_w = mg\,sin\,\theta \tag{2}$$

$$F_r = C_r mg\cos\theta \tag{3}$$

$$F_d = 0.5\rho AC_d(v + v_w)^2 \tag{4}$$

Table 1 explains each term in Equations (2), (3), and (4), along with the values that will be used in the simulation, which are based on *Proton Perdana's second-generation* specifications.

Table 1. Parameters for the longitudinal vehicle dynamics

Parameters	Value
Mass of vehicle, m	1535 kg
Front cross-section area, A	1.88 m^2
Slope angel, θ	0^O
Air density, ρ	1.202 kg/m^3
Drag coefficient, C_d	0.31
Gravity acceleration, g	9.81 m/s^2
Rolling coefficient, C_r	0.015

3.0 CENTRALIZED PFC

The centralized PFC uses a linearized model of the kinetic equation in Equation (1) to formulate the control law. Since the step response of this model is stable, a standard PFC formulation can be used to compute the control action and implement the constraints. There are three important formulations: model prediction, control law, and constraint handling. The following subsections will discuss these in detail.

3.1 Model Linearization, Discretization, and Prediction

PFC is a discrete linear controller; thus, the nonlinear equation of motion in Equation (1) needs to be linearized and discretized for prediction and control law formulations [11]. The nominal operating points for the linearization are given in Table 2. The speed of 14 m/s is selected because most of the ACC operation is within this value. The rest of the parameters are measured or calculated based on the steady-state condition.

Table 2. Nominal operating parameters

Parameters	Value
Nominal vehicle speed, v_n	14 m/s
Nominal wind gust speed, v_{wn}	0 m/s
Nominal slope angle, θ_n	0^O
Nominal traction force, F_t	294.53 N

From the Taylor series expansion, the linearized model of Equation (1) can be represented as:

$$m\dot{v}' = F_t{}' - mg(C_r sin\theta_n - cos\theta_n)\theta' - \rho AC_d(v_n - v_{wn})v' \tag{5}$$

Note that the superscript {'} in Equation (5) represents the difference between actual and nominal values of the respective parameters. With a simple algebraic manipulation, Equation (5) can be further simplified into:

$$\tau\dot{v}' + v' = K(F_t{}' + D) \tag{6}$$

where:

$$\tau = \frac{m}{\rho AC_d(v_n - v_{wn})}$$

$$K = \frac{1}{\rho AC_d(v_n - v_{wn})}$$

$$D = mg(C_r sin\theta_n - cos\theta_n)\theta'$$

Applying the Laplace transform, a first-order transfer function can be obtained as:

$$v'(s) = \frac{K}{\tau s + 1}[F_t'(s) + D(s)] \tag{7}$$

The input to transfer function Equation. (7) is the traction force F_t', the output is velocity v', and the term D is considered a disturbance. For the discretization process, a sampling time T_s of 0.1 s is used. Denotes $y(z)$ as the discrete output and $u(z)$ as the discrete input and hence the first-order discrete transfer function can be represented as:

$$y(z) = \frac{b}{z-a}u(z) \tag{8}$$

Equation (8) can then be transformed into a linear prediction equation by noting the term z as a shift operator. Thus, the one-step ahead prediction at the sample k is formulated as:

$$y(k+1|k) = bu(k) + ay(k) \tag{9}$$

By using an independent Model (IM) approach to cater for plant model mismatch and other uncertainties [11], the term $d(k)$ which is the difference between calculated output y and measured output from a plant y_p can be introduced. Thus, Equation (9) can be extended to construct a vector of the nth step ahead output at sample k as:

$$y_n = HU + Fy(k) + d(k) \tag{10}$$

where:

$$y_n = \begin{bmatrix} y_p(k+1|k) \\ y_p(k+2|k) \\ \vdots \\ y_p(k+n|k) \end{bmatrix}, H = \begin{bmatrix} b & 0 & 0 & 0 \\ ab & b & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a^{n-1}b & a^{n-2}b & \dots & b \end{bmatrix},$$

$$U = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n) \end{bmatrix}, F = \begin{bmatrix} a \\ a^2 \\ \vdots \\ a^n \end{bmatrix}$$

3.2 Centralized PFC Algorithm

In the PFC framework, at each sample time k, a new target trajectory will be calculated based on a desired steady-state target R. A first order response dynamics from the measured output y_p to R is often considered [12], where the nth step ahead trajectory can be defined as:

$$r(k+n|k) = (1-\lambda^n)R + \lambda^n y_p(k) \tag{11}$$

There are two tuning parameters that need to be selected. First is the desired closed-loop pole λ, which corresponds to the desired Closed-Loop Time Response (CLTR): the required time to reach 95% from the steady state value. The relationship between λ and CLTR is given as:

$$\lambda = e^{-3T_s/CLTR} \tag{12}$$

The second tuning parameter is the coincidence horizon n: the point where the future prediction of a system in Equation (10) is forced to match with the first order target trajectory in Equation (11). Based on many PFC references [12-13], it is noted that for a first order model as in Equation (8), a coincidence horizon of n=1, is considered the best choice. Thus, the equality can be represented as:

$$bu(k) + ay(k) + d(k) = (1-\lambda)R + \lambda y_p(k) \tag{13}$$

For the ACC application, the CLTR is set to be 15 s, which corresponds to λ = 0.9. The final control law is derived just by a simple algebraic manipulation of Equation (13), where:

$$u(k) = b^{-1}[(1-\lambda)R + \lambda y_p(k) - ay(k) - d(k)] \tag{14}$$

3.3 Centralized PFC Constraints Handling

One of the advantages of any model-based predictive control is its ability to systematically handle system constraints. In the traditional ACC system, a switching strategy is often used to switch between speed control or space control [4]. However, since a predictive controller can predict a future outcome, the control law can be formulated so that the constraints are satisfied automatically within the control law.

For an ACC system, there are several constraints that need to be imposed. For safety and comfort, the maximum acceleration is limited to a_{max} = 2 m/s^2 and maximum deceleration to a_{min}= -3m/s^2 [4]. These two limits can be considered as output constraints by formulating:

$$y_{max}(k+1) = a_{max}T_s + y(k) \tag{15}$$

$$y_{min}(k+1) = a_{min}T_s + y(k) \tag{16}$$

Figure 2. Schematic of centralized PFC for ACC application [8]

It should be noted that the value of $y(k)$ in Equations (15) and (16) need to be updated for the next prediction by using Equation (10). For the safe distancing, (refer to Fig. 2), a distance sensor is needed to measure the relative distance and the value will be compared with the standard safe following distance [4] given as:

$$D_{safe}(k) = D_{default} + T_{gap}y(k) \tag{17}$$

The default distance is often set to 10 m for the complete stop and the safe time gap T_{gap} between the vehicle is set to 1.4 s [4]. Based on the output velocity prediction in Equation (10), the future relative position between the car D_r can be estimated by assuming the future velocity of the lead car is constant at the instantaneous sampling:

$$D_r(k+1) = [v_l - y(k)]T_s + D_r(k) \tag{18}$$

By using superposition, at each sampling time, the maximum velocity to keep the safe distance can be formed by equating the safe distance in Equation (17) and relative distance in Equation (18) as:

$$y_{max} = \left(v_l T_s + D_r(k) - D_{default}\right)/\left(T_{gap} + T_s\right) \tag{19}$$

The following algorithm shows how all the constraints are computed in the control law:

Algorithm 1: At each sample time:

a) Compute the unconstrained input $u(k)$ as in Equation (14)

b) A simple for loop with suitable validation horizon ni is used for future prediction.

c) At each horizon, the controller will predict the future velocity by using Equation (10) by assuming the future input dynamics in vector U is constant, thus:

$$y_i = H_i L u(k) + F_i y(k) + d(k) \tag{20}$$

where L = [1, 1.,1] T depends on the size of ni and subscript 'i' representing the ith row of the respective matrix or vector.

1) If $y > y_{max}$ / $y < y_{min}$, then:

$$u(k) = H_i L_i^{-1}[v_{max}(i) - F_i y(k) - d] \tag{21}$$

2) Else, the value of $u(k)$ is retained.

3) Update all the constraint values in Equations (15), (16) and (19) and repeat the algorithm.

For the output constraint, the number of validation horizons is very important to ensure smooth control effort. If it is too short, the controller will provide a more aggressive input. Conversely, if it is too long, more computation will be used, and the control effort will be too conservative [11-12]. For an ACC application, the best choice is to select the number that can cover most of the transient period which is around n_i=11, where the algorithm is not usually sensitive to small changes in this choice.

4.0 HIERARCHICAL PFC

In a hierarchal structure, the vehicle dynamics are separated into two parts, namely: low level and upper level as shown in Fig. 3. The upper-level system consists of the vehicle kinematic model which is simpler, and the lower level consists of the nonlinear vehicle longitudinal dynamics. A detailed discussion will be given in the following subsections.

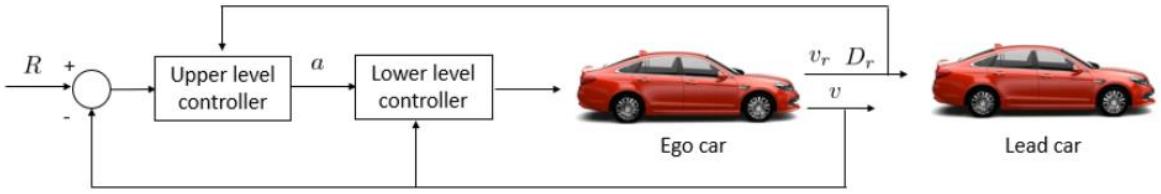

Figure 3. Schematic of hierarchical PFC for ACC application [10]

4.1 Upper-level Control

For hierarchical PFC, the control law will be derived based on the upper-level system by using a simple kinematic dynamic between acceleration and velocity where the transfer function can be represented as:

$$v(s) = \frac{1}{s(\tau s + 1)} a(s) \tag{22}$$

The input to the system is the acceleration $a(s)$, while the output will be velocity $v(s)$. In Equation (22), an estimated time constant τ is used to compensate a delay that corresponds to powertrain dynamics with regard to gear ratio and other components [4]. For this work, it is assumed that the car will track the velocity given an acceleration with a delay of 0.5s [4]. Like the earlier developments, Equation (22) needs to be discretized with a sampling time of 0.1 s, where the discrete transfer function can be represented as:

$$G(z) = \frac{y(z)}{u(z)} = \frac{0.009365z^{-1} + 0.008762z^{-2}}{1 - 1.819z^{-1} + 0.819z^{-2}} \tag{23}$$

The term u represents the acceleration input and the term y is the output velocity. Since the step response of the transfer function in Equation (23) is nonconverging due to the existence of an independent integrator in the denominator, it needs to be pre-stabilized before deriving the control law to avoid an ill-posed solution [13]. A simple solution by cascading a proportional gain, K in the control law is used as shown Fig. 4 [14].

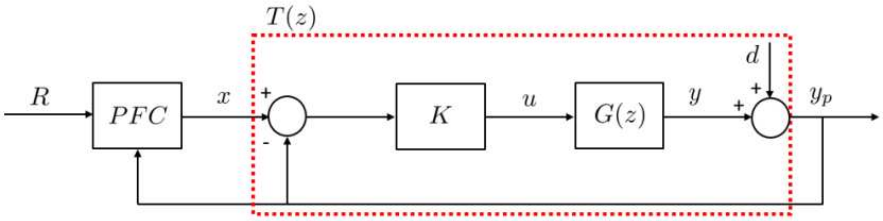

Figure 4. Inner cascade hierarchical PFC structure [10, 14]

By reducing the block diagram in Fig. 4, the inner loop transfer function $T(z)$ can be derived and used to compute a stable prediction and control law as:

$$T(z) = \frac{y(z)}{x(z)} = \frac{G(z)K}{1+G(z)K} \tag{24}$$

The term $x(z)$ in Equation (24) represents a modified controlled input. At this stage, the value of K is crudely selected by utilizing the MATLAB PID tuner which proposes a value of K=1.147 [10]. Since the focus of this paper is on algorithm development, the effect of this value will not be considered in depth and will be part of future work. Equation (24) also can be used to construct the n-step ahead prediction at k samples by using the superimposed principle as:

$$y(k+n|k) = HX + PX_0 + QY + d \tag{25}$$

where:

$$X = \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+i) \end{bmatrix}, X_0 = x(k-1), Y = \begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix}$$

The control law is computed by equating Equation (25) with the target trajectory in Equation (11). Since the internal model is a second order system, according to [13], the number of coincidence horizons needs to be more than 1. A standard PFC framework assumes that the future input is constant ie. $x(k) = x(k+1) = \ldots = x(k+n)$ [13]. Thus, the summation of the n-th row of matrix H_n will be h_n and the final equality is given as:

$$h_n x(k) + P_n X_0 + Q_n Y + d = (1 - \lambda^n)R + \lambda^n y_p(k) \tag{26}$$

The modified input x is then computed as:

$$x(k) = h_n^{-1}\left[(1 - \lambda^n)R + \lambda^n y_p(k) - P_n X_0 - Q_n Y - d\right] \tag{27}$$

Since a cascaded structure is used for this controller, by referring to Fig. 4, the actual input can be computed as:

$$u(k) = K\left[x(k) - y_p(k)\right] \tag{28}$$

4.2 Lower-level Control via Inverse Model

The lower-level controller is responsible for tracking the desired acceleration that is computed by the upper-level controller by sending a suitable input command to the plant (refer to Fig. 3). In the actual implementation, the input will be the percentage of the pedal press. However, to keep the simulation simple, only the traction force F_t from vehicle longitudinal dynamics is considered as an input and it can be computed by inversing Equation (1) as:

$$F_t = ma + F_w + F_r + F_d \tag{29}$$

With this structure, the nonlinear equation can be used directly where it has less sensitivity to the nonlinearities. In this work, the information such as wind gust and slope angle will remain constant. However, for future work, this value can be estimated using a suitable sensor and fed directly to update the inverse model for more accurate calculation.

4.3 Hierarchical PFC Constraint Handling Formulation

For constraint handling, the implementation of the Hierarchical PFC is a bit different from the Centralized PFC since the internal model is simpler. The safe distancing constraints Equation (19) will be imposed as output constraints by using almost a similar algorithm as before:

Algorithm 2: At each sample time:

a) Compute the unconstrained input $u(k)$ as in Equation (27)

b) A simple for loop with suitable validation horizon n_i is used for future prediction.

c) At each horizon, predict the future velocity by using Equation (25)

d) If $y > y_{max}$ / $y < y_{min}$, then:

$$x(k) = h_i^{-1}[y_{max}(i) - P_i U_0 - Q_i Y - d] \tag{30}$$

e) Else, the value of $x(k)$ is retained.

f) Update all the constraint values in Equation (19) and repeat the algorithm for the next horizon.

As for the constraints in the passenger comfort Eq. (15) and (16), it will be formulated as an input constraint since the input to the upper-level system is the acceleration. According to PFC literature [11,12,15] a simple clipping strategy is sufficient, such that if $x < y(k) + u_{min}/K$, then:

$$x(k) = y(k) + u_{min}/K \tag{31}$$

Similarly, if $x > y(k) + u_{max}/K$, then:

$$x(k) = y(k) + u_{max}/K \tag{32}$$

Finally, once the compensated input value $x(k)$ is finalized, the actual input should be calculated as in Equation (28).

5.0 SIMULATION RESULTS

This section presents the simulation results to compare the performance between the two PFC control structures. The two control algorithms are applied to the vehicle plant, consisting of the nonlinear longitudinal dynamic described in Equation (1). Standard performance parameters are used for the analysis, such as root mean square error and settling time, except for the case when all the constraints are activated. The PFC with a hierarchical structure is denoted as PFCH, and the PFC with a centralized structure is denoted as PFCC.

5.1 Tracking Desired Speed

Both controllers are set with the initial speed of 20 m/s and tuned to track the desired speed of 30 m/s from 0 – 50 s and 14 m/s from 50 – 100 s. No constraints are considered in this stage, and both controllers are set to have a CLTR of 15 s to reach the steady state value. Fig. 5 shows the closed-loop responses of both controllers. It can be observed that PFCH (red solid line) managed to track the desired speed more accurately compared to PFCC (blue dotted line) in both set points. The reason is that PFCC uses the linearized internal model, where it can only produce accurate tracking when the set point is near the nominal operating point (refer to Table 2, the nominal velocity is 14 m/s). As the set point becomes far from the operating point, for example, in the case of tracking 30 m/s desired speed, it takes longer to settle in the steady state region due to the plant model mismatch. Of course, several solutions exist to improve the issue, such as introducing gain scheduling, but this would require more testing and accurate data to design a suitable mapping table.

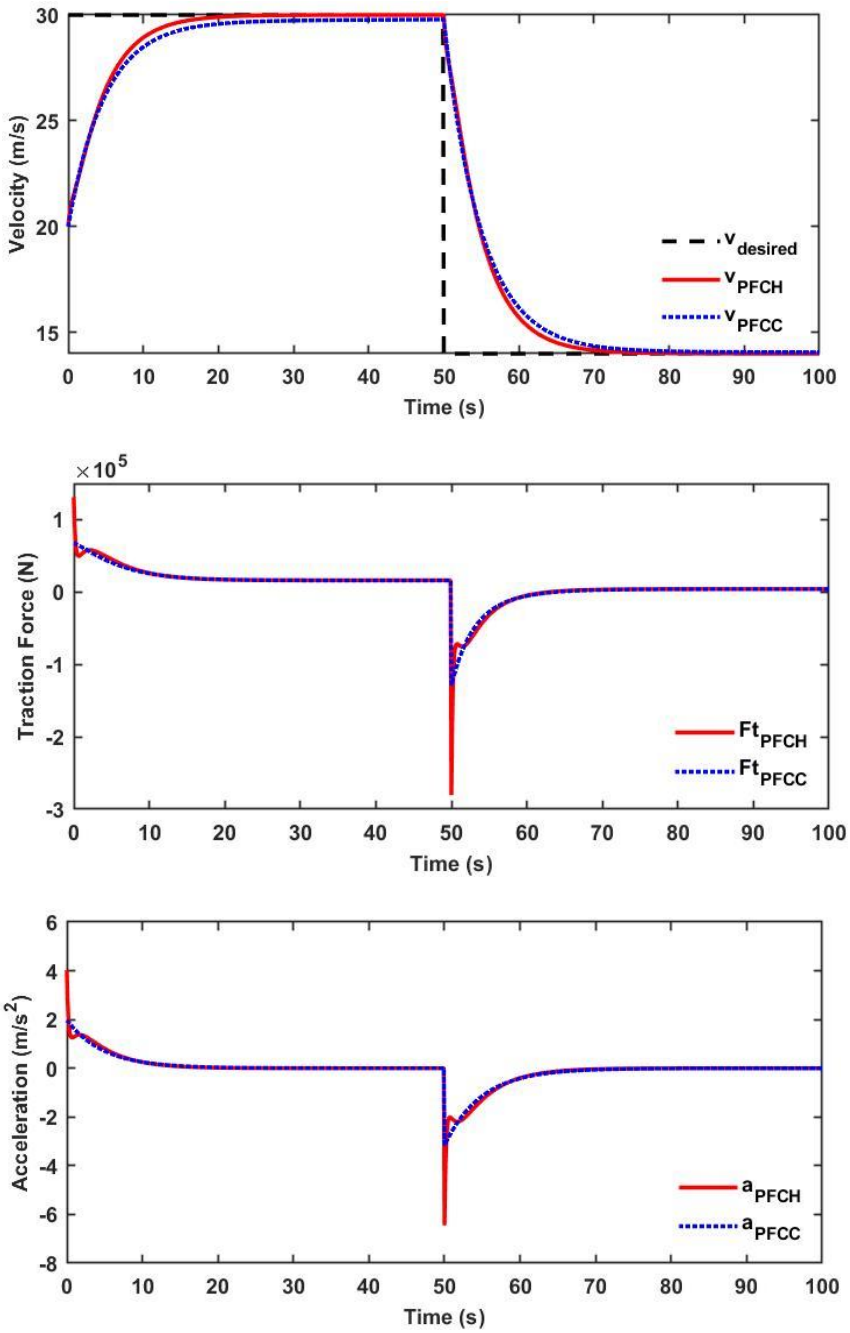

Figure 5. Performance index of PFCH and PFCC in tracking the desired set-point

Table 3 provides the standard performance criteria for both controllers. The PFCH produces a lower root mean squared error and faster settling time than PFCC. Nevertheless, it should be noted that the accuracy of PFCH comes with a hefty price as more aggressive input excitation is needed, as shown in Fig. 5. This is because the accuracy of PFCH prediction forces the response to converge faster than the PFCC controller. In return, a higher traction force is needed to satisfy the control objective. In return, this high over-actuation in acceleration may provide an uncomfortable ride for the passenger and consume more fuel than necessary. As mentioned earlier, a constraint on acceleration needs to be implemented to overcome the issue.

Table 3. Performance index of PFCH and PFCC in tracking the desired set-point

Performance Index	PFCH	PFCC
Settling time, s	67.0778	69.6655
Rise time, s	9.2838	11.0593
RMSE	2.9459	2.9970

5.2 Acceleration Constraints

In this subsection, the acceleration and deceleration of both cars are constrained to consider passenger comfort. The constrained PFCC is implemented by using Algorithm 1 and PFCH by using Algorithm 2. Fig. 6 shows the constrained performance of both controllers, where it is noted that both controllers managed to satisfy the implemented constraints (a_{max} = 2 m/s^2 and a_{min}= -3 m/s^2). However, as can be observed, PFCH provides better control performance, where it retains the accuracy in tracking the desired speed with some delay in the settling time due to the constraint implementation. Conversely, PFCC produces a longer settling time to track the desired velocity, especially when the set point is far away from the nominal value.

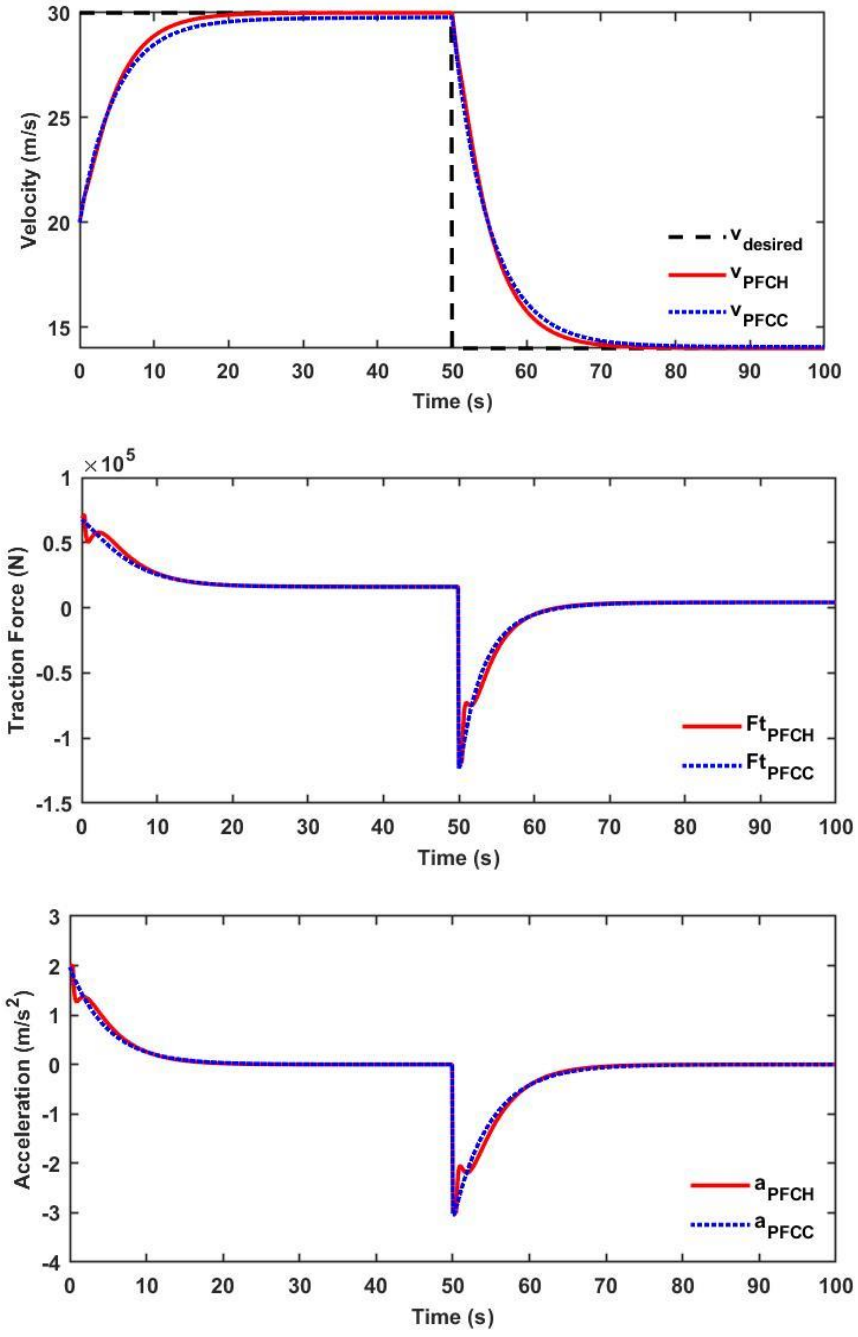

Figure 6. Constrained closed-loop performance in tracking the set-point

There is also some interesting remark: although both controllers are implemented with a validation horizon of 1 for acceleration limits, referring to Table 4, PFCH is approximately two times faster than PFCC in satisfying the acceleration limits. The maximum elapsed time is recorded using MATLAB's *tic* and *toc* functions. The main difference between the two algorithms is how the constraints are implemented. The PFCC considers the acceleration limit as an output constraint; thus, the controller needs to solve more mathematical equations as given in Algorithm 2 since output prediction is necessary to check and satisfy the constraints. Conversely, the PFCH treats the acceleration limit as an input constraint, which can be implemented directly as given in Equations (31) and (32). Since the future input prediction in PFC is assumed to be constant, only the first sample is needed to check and satisfy the constraints.

Table 4. Computation time for implementing acceleration constraints

Performance Index	PFCH	PFCC
Computation time, s	0.000302	0.000642

5.3 Safe distancing

With regards to safe distancing, both controllers need to consider it as an output constraint. Fig. 7 shows the performance of both controllers for a car following application when all the constraints are implemented. As can be observed from the velocity and acceleration graphs, initially both controllers try to track the desired velocity of 30 m/s, however approximately around 10s, a possible violation of the safe distance constraints is predicted (refer Fig. 8). Thus, the supplied traction force is reduced to retain the safe distancing.

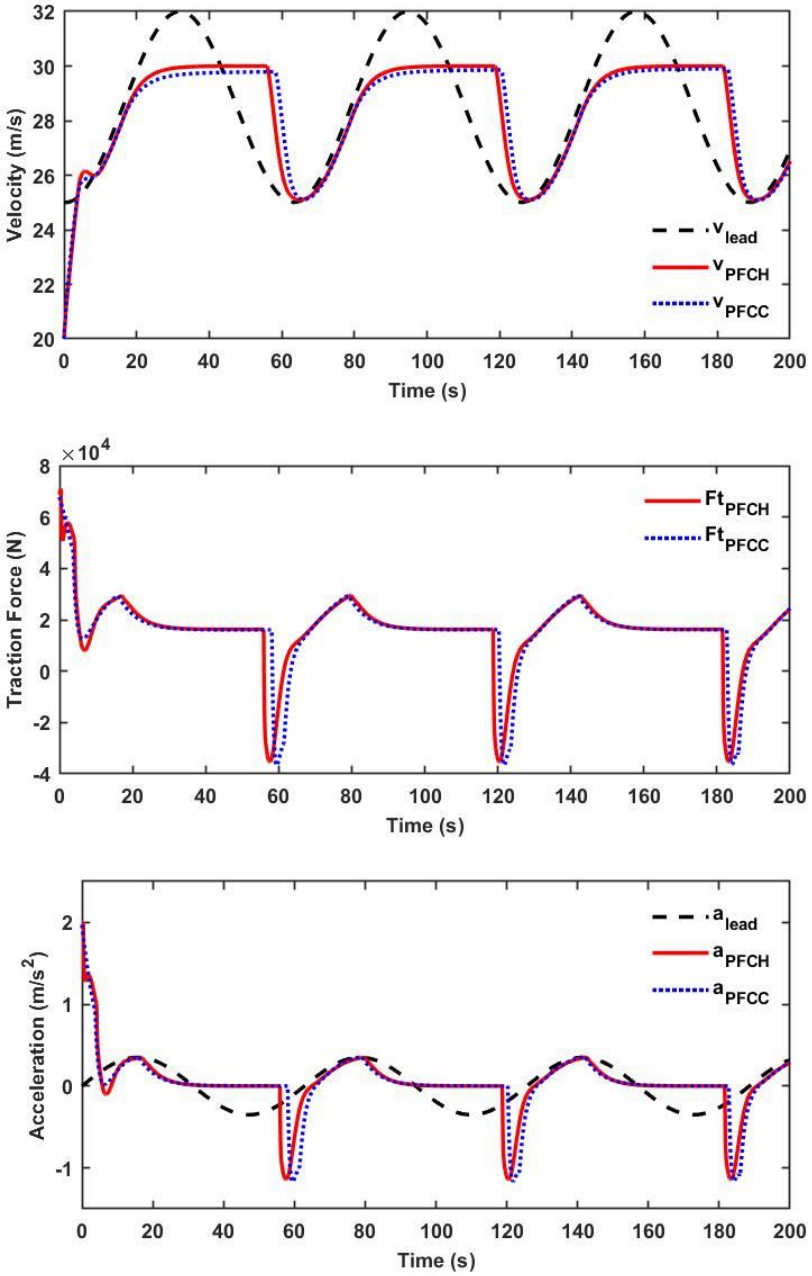

Figure 7. Constrained closed-loop performance PFCH and PFCC in the car following application

Once the lead vehicle speed is increasing, both controllers resume their original goal, which is to track the desired speed of 30 m/s while satisfying all the constraints. However, a similar observation as in the previous case is found where PFCC takes a longer time to reach the desired velocity due to the usage of a linearized model. Another issue that is worth discussing is the selection of the validation horizon. As the constraints-checking algorithm becomes more complicated, a longer validation horizon is required. Thus, both controllers need validation horizons of at least 13 for a smooth control effort. Table 5 shows the required computation time to execute the constraint implementation. The difference is not quite significant since both controllers are using almost the same algorithm. There is also a delay noticeable in PFCC, where it detects the violation one step behind PFCH, as shown in Fig. 8. Nevertheless, both controllers managed to satisfy the safe

distancing. Note that the safe distance for both controllers is different as its formulation in Equation (17) depends on the current speed of a vehicle.

Table 5. Computation time for implementing all the constraints

Performance Index	PFCH	PFCC
Computation time, s	0.003800	0.004095

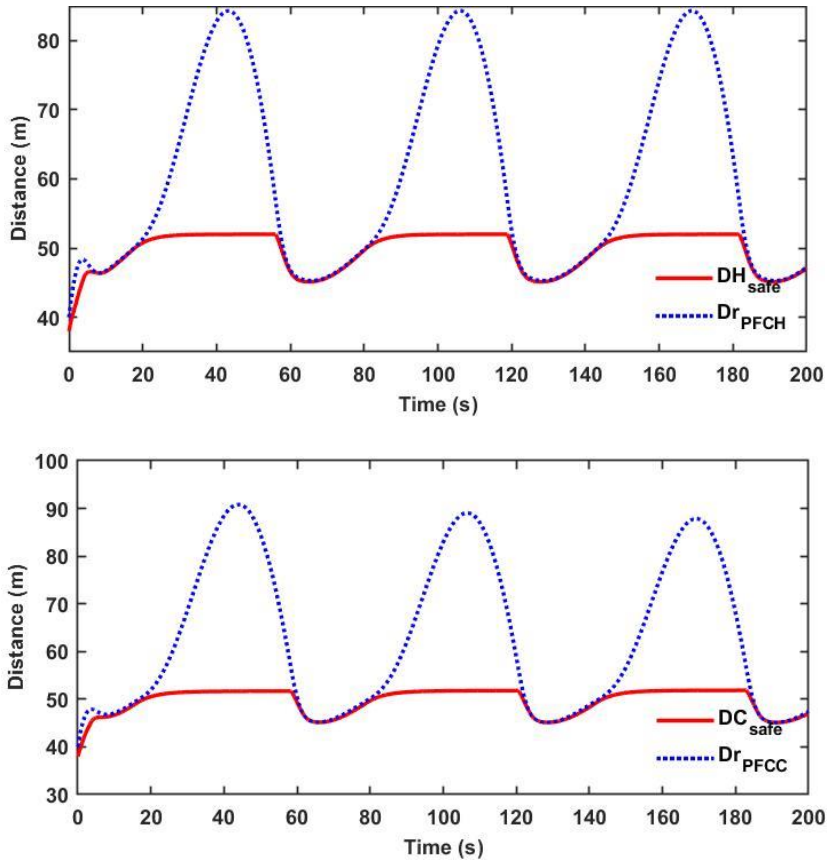

Figure 8. Safe distancing for PFCH and PFCC in the car following application

5.4 Disturbance and Uncertainties

For this test, several input disturbances are inserted into the system by increasing the slope of the road from 0 to 10 degrees after 40 s and wind gusts from 0 to 4 m/s after 140 s. Fig. 9 shows the response of both controllers. In this case, it is obvious that PFCH is better than PFCC in handling disturbances. The PFCC becomes more sensitive to the disturbance and in return, the control performance becomes more conservative. This is very undesirable since in the actual implementation, any delay in response will make the traffic become worse since the vehicle from the next lane can cut in. This scenario could lead to a phantom traffic issue.

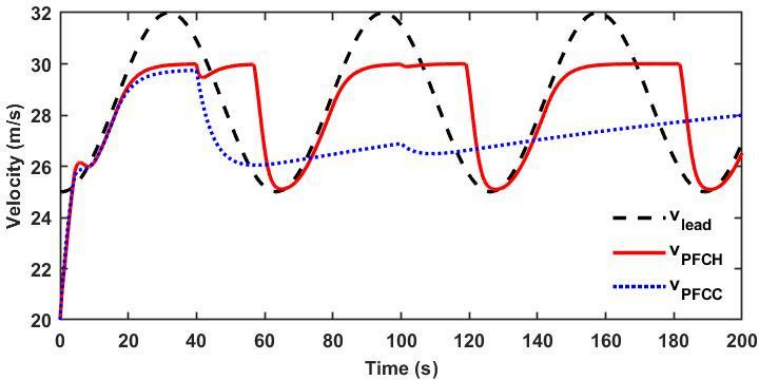

Figure 9. Constrained closed-loop performance PFCH and PFCC in the car following application in the presence of disturbances

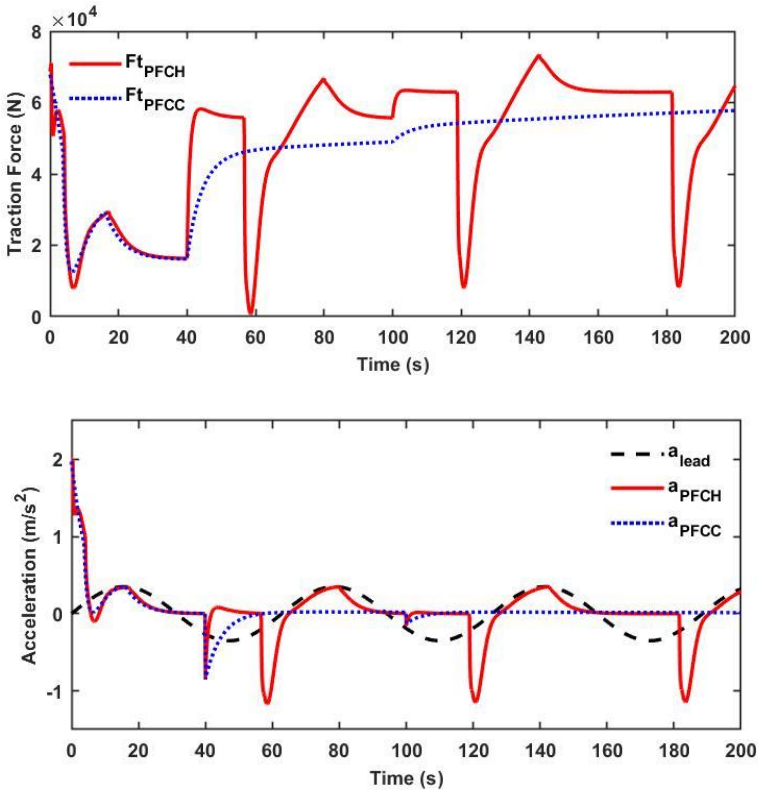

Figure 9. (cont.)

Conversely, since PFCH is developed based on two level structures, it can cope with the modeling uncertainty better than PFCC. The inverse nonlinear model used in lower-level control managed to compensate for the nonlinearities effect even when some of the parameters are assumed constant. If these parameters are updated online, a better control action may be expected. Nevertheless, this issue will be investigated later in future work. Fig. 10 shows that the safe distancing is retained for both controllers, yet the lead vehicle has left the PFCC vehicle far ahead by observing the relative distance.

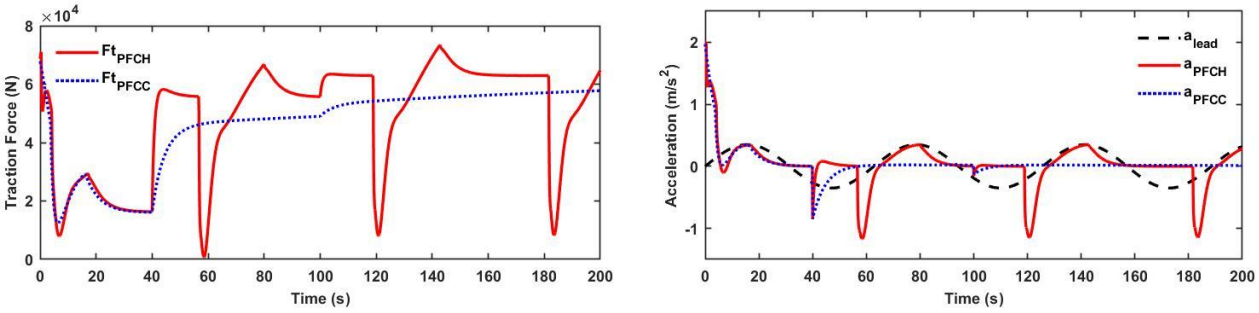

Figure 10. Safe distancing for PFCH and PFCC in the car following application in the presence of disturbance

6.0 CONCLUSIONS

In summary, this paper has compared the control performance of PFC between the centralized and hierarchical structures. Based on the simulation results, it is found that the PFCH provides a more accurate response with a better RMSE of 2.9459 and a quicker settling time of 67.0778 s compared to PFCC with RMSE of 2.9970 and settling time of 69.6655 s in tracking the desired speed. The computation time of 0.000302 s for PFCH is also approximately two times faster compared to PFCC, 0.000642 s, in satisfying the acceleration constraints. However, when satisfying the constraint on safe distancing, the computation time between the two controllers is minimal, where 0.003800 s for PFCH and 0.004095 s for PFCC since almost a similar approach is used in this case. Overall, it is proven that PFCH provides better results in handling the nonlinear effect and disturbance due to the use of separate level control architecture. Thus, the hierarchical structure is more suitable to be implemented with PFC, especially for the ACC application. Future work will consider a more detailed analysis by including the full vehicle longitudinal dynamics, which consists of powertrain and braking dynamics, to assess the controller performance and the effect of updating other model parameters in the lower-level control.

7.0 ACKNOWLEDGEMENT

The authors would like to acknowledge the Ministry of Higher Education Malaysia for funding this work under the Fundamental Research Grant FRGS/1/2021/TK02/UIAM/02/2 (FRGS21-240-0849) and the International Islamic University Malaysia Research Management Center Grant 2020 (RMCG20-010-0010). The second author would like to acknowledge Majlis Amanah Rakyat (MARA) for his tuition fee waiver. The authors also want to acknowledge the Center of Unmanned Technologies (CUTe) IIUM for their support and guidance.

8.0 CONFLICT OF INTEREST

The authors declare that no competing financial interest or personal relationship could have appeared to influence the work reported in this paper.

9.0 REFERENCES

[1] S. H. Lee and D.R. Ahn, "Design and verification of driver interfaces for adaptive cruise control systems," *Journal of Mechanical Science and Technology*, vol. 29, no. 4, pp. 2451-2460, 2015.

[2] B.C. Zanchin, R. Adamshuk, M.M. Santos and K.S. Collazos, "On the instrumentation and classification of autonomous cars," *In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2631-2636, 2017.

[3] L. Yu and R. Wang, Researches on Adaptive Cruise Control system: A state of the art review. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 236, no. 2-3, pp. 211-240, 2022.

[4] R. Rajamani, "Adaptive cruise control," *In: Baillieul, J., Samad, T. (eds) Encyclopedia of Systems and Control.* pp. 20-26, 2015. Springer, London.

[5] T. Takahama and D. Akasaka, "Model predictive control approach to design practical adaptive cruise control for traffic jam," *International Journal of Automotive Engineering*, vol. 9, no. 3, pp. 99-104, 2018.

[6] Z. Haroon, B. Khan, U. Farid, S.M. Ali and C.A. Mehmood, "Switching control paradigms for adaptive cruise control system with stop-and-go scenario," *Arabian Journal for Science and Engineering*, vol. 44, no. 3, pp. 2103-2113, 2019.

[7] Z. Nie and H. Farzaneh, "Adaptive cruise control for eco-driving based on model predictive control algorithm," *Applied Sciences*, vol. 10, no. 15, p. 5271, 2020.

[8] M. Abdullah and M. Idres, "Fuel cell starvation control using model predictive technique with Laguerre and exponential weight functions," *Journal of Mechanical Science and Technology*, vol. 28, no. 5, pp. 1995-2002, 2014.

[9] M.A.S. Zainuddin, M. Abdullah, S. Ahmad and K. A. Tofrowaih, "Performance comparison between predictive functional control and PID algorithms for Automobile Cruise Control System," *International Journal of Automotive and Mechanical Engineering*, vol. 19, no. 1, pp. 9460-9468, 2022.

[10] M.A.S. Zainuddin, M. Abdullah, S. Ahmad, M.S. Uzair and Z.M.P.A Baidowi, "Performance analysis of predictive functional control for Automobile Adaptive Cruise Control System," *IIUM Engineering Journal,* vol. 24, no. 1, pp. 213-225, 2023.

[11] J.A. Rossiter, M.S. Aftab, G. Panoutsos and O. Gonzalez-Villarreal, "A novel approach to PFC for nonlinear systems," *European Journal of Control*, vol. 68, p. 100668, 2022.

[12] M. Abdullah and J.A. Rossiter, "Utilising Laguerre function in predictive functional control to ensure prediction consistency," *UKACC 11th International Conference on Control (CONTROL),* pp. 1-6. 2016.

[13] M. Abdullah and J.A. Rossiter, "Alternative method for predictive functional control to handle an integrating process," *UKACC 12th International Conference on Control (CONTROL),* pp. 26-31, 2018.

[14] Z. Zhang, J. A. Rossiter, L. Xie, and H. Su, "Predictive functional control for integrator systems," *Journal of the Franklin Institute*, vol 357, no. 7, pp. 4171-4186, 2020.

[15] J.A. Rossiter and M.S. Aftab, "A comparison of tuning methods for predictive functional control," *Processes*, vol. 9, no. 7, p. 1140.

[16] M. Abdullah, M. Idres and M. Azan, "Model Predictive Control for Regulating Fuel Cell Stack Temperature and Air Flow Rate," *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, vol. 92, no. 2, pp. 171-181, 2022.