https://doi.org/10.31436/iiumej.v24i1.2341

PERFORMANCE ANALYSIS OF PREDICTIVE FUNCTIONAL CONTROL FOR AUTOMOBILE ADAPTIVE CRUISE CONTROL SYSTEM

MOHAMED AL-SIDEQUE ZAINUDDIN[1,2], MUHAMMAD ABDULLAH[1*], SALMIAH AHMAD[1], MOHD SUHAIMI UZAIR[3] AND ZAID MUJAIYID PUTRA AHMAD BAIDOWI[4]

[1]*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, Melaka, Malaysia*
[3]*Testing & Development, Engine Development, Powertrain R&D, Powertrain Division, Proton Holdings Berhad, Shah Alam, Malaysia*
[4]*Centre of Foundation Studies, Universiti Teknologi MARA, Cawangan Selangor, Kampus Dengkil, 43800 Dengkil, Selangor, Malaysia*

[*]*Corresponding author: mohd_abdl@iium.edu.my*

(Received: 28th February 2022; Accepted: 21st June 2022; Published on-line: 4th January 2023)

ABSTRACT: This paper presents the performance analysis of Predictive Functional Control (PFC) for Adaptive Cruise Control (ACC) application. To cope with multiple driving objectives of modern ACC systems such as passenger comfort, safe distancing, and fast time response, an advanced optimal controller such as Model Predictive Control (MPC) is often used. Nevertheless, MPC requires a high computation load due to its complex formulation and may overload the processing power of a microcontroller. Thus, the prime objective of this work is to propose a PFC algorithm as an alternative controller, while providing a formal comparison between MPC and the traditional Proportional Integral (PI) controller. A standard kinematic model for vehicle longitudinal dynamics was modelled and used to derive the control law of PFC. Since the open-loop dynamic of the derived transfer function is not stable, the second objective is to propose a pre-stabilized loop or cascade PFC structure for the system. A complete tuning procedure and analysis were presented. The simulation result shows that although MPC performance is the best for the ACC application with Root Mean Square Error (RMSE) of 1.4873, PFC has shown a promising response with RMSE of 1.5501, which is better compared to the PI controller with RMSE of 1.6219. All the imposed driving constraints such as maximum acceleration, maximum deceleration and safe distance were satisfied in the car following application. Thus, the findings from this work can become a good initial motivation to further explore the capability of the PFC algorithm for future ACC development.

ABSTRAK: Kajian ini adalah berkenaan analisis prestasi Kawalan Fungsi Ramalan (PFC) aplikasi Kawalan Mudah Suai (ACC). Bagi memenuhi pelbagai keperluan objektif sistem pemanduan moden ACC seperti keselesaan penumpang, penjarakan selamat dan tindak balas pantas, kawalan optimum terbaru seperti Model Kawalan Ramalan (MPC) sering digunakan. Walau bagaimanapun, MPC memerlukan beban pengiraan tinggi kerana rumusnya yang kompleks dan mungkin mengakibatkan beban berlebihan kuasa pemprosesan mikrokawalan. Oleh itu, matlamat utama kajian ini adalah bagi mencadangkan algoritma PFC yang mempunyai pengiraan mudah sebagai kawalan alternatif, sementara menyediakan perbandingan formal antara MPC dan kawalan tradisional Berkadar Keseluruhan (PI). Oleh kerana model ini tidak stabil, objektif kedua adalah mencadangkan penggunaan struktur PFC berlapis bagi menstabilkan sistem terlebih dahulu sebelum algorithma kawalan digunakan atau dengan menggunakan struktur PFC secara berturut pada sistem. Prosedur lengkap dan terperinci untuk analisis PFC dibentangkan. Dapatan simulasi kajian menunjukkan walaupun prestasi MPC adalah baik bagi aplikasi ACC dengan Ralat Punca Min Kuasa Dua (RMSE) bernilai 1.4873, namun PFC menunjukkan tindak balas baik dengan RMSE bernilai 1.5501 berbanding kawalan PI yang mempunyai RMSE sebanyak 1.6219. Kesemua kekangan seperti pecutan dan nyahpecutan maksima, dan penjarakan selamat bertepatan dengan aplikasi kenderaan ini. Dengan itu, penemuan ini adalah motivasi awal yang baik bagi meneroka lebih jauh keupayaan algoritma PFC bagi membangun ACC pada masa hadapan.

KEY WORDS: *predictive functional control; model predictive control; PID; adaptive cruise control*

1. INTRODUCTION

Adaptive Cruise Control (ACC) is one of the basic features for an Advanced Driving Assistant System (ADAS), where its function is to regulate the speed of a vehicle while retaining a safe following distance. Compared to the conventional Cruise Control system, ACC has two modes of operation: speed control and space control. With the advancement of sensor and microprocessor technologies, an energy efficient ACC system which can satisfy multiple driving objectives such as passengers' comfort, fuel efficiency, and safe distancing with acceptable time response are currently being developed. This type of ACC system requires a more sophisticated control algorithm and structure as each of the driving objective functions needs to be optimized to get the optimum control action. It is also well noted that the vehicle longitudinal dynamics for full range operation is highly nonlinear and thus a hierarchal control structure is required where the upper-level system is controlled to provide a desired signal for the lower-level system to track [1]. At the same time the switching between space and speed control also needs to be considered to ensure the safety and comfort of the passenger.

With a traditional PI controller, a special switching algorithm is used to ensure a smooth transition between the two modes to avoid jerking [2]. This operation can be done with the help of a look up table. Nevertheless, the control performance is not robust since the decision making is solely based on the current measurement. To improve the robustness, a proper tuning strategy can be implemented, as demonstrated by [3]. However, the control structure has become more complicated and there is no systematic tuning approach for this type of modification leaving only trial and error or use of some optimal algorithm tuning scheme. Several works also have proposed the use of Fuzzy Logic Controller (FLC) for an ACC system. Using a common logic rule, the switching strategy is developed based on the relative distance with the lead vehicle. The strategy is quite straightforward and implantable, yet as reported by [4], in the presence of unmeasured disturbance, the control performance will be deteriorated. Indeed, there are several options to overcome these issues such as improving the accuracy of the fuzzy rule by using a different logic function or combining it with other advanced controllers [5]. However, there is no systematic rule that can be implemented as there are many parameters that need to be tuned and most of the modifications are system dependent.

A predictive controller is another suitable option to control the upper level of an ACC system. A representative kinematic mathematical model is used to predict the future

https://doi.org/10.31436/iiumej.v24i1.2341

velocity and the input acceleration will be optimized to get the desired response while satisfying other driving constraints such as passenger's comfort, fuel efficiency, and safe distancing. As reported in the literature, various authors have managed to prove that this algorithm can effectively control the vehicle speed in satisfying the multiple driving objectives either in simulation or real-time implementation [6-8]. The main reason behind this is the prediction capability where the controller can anticipate what will happen in the future and try to provide the best solution. Hence, no special switching strategy is required and it can be replaced by a more systematic optimization problem. Indeed, there are also several limitations such as a proper selection of the tuning parameters of prediction and control horizons based on certain applications. However, even with suitable parameters, the computation burden will still be heavy to implement in existing vehicle microprocessors. In a short sampling time i.e., 0.1 *s*, the algorithm needs to compute several heavy mathematical operations such as prediction, optimization, and constraints handling using a standard quadratic cost function. Of course, there are several options to reduce the computation such as offline implementation [9] and changing the prediction structures using a special function [7,10], yet still the basic microprocessor in existing vehicles needs to be upgraded. This requirement will increase the cost and, in the future, full implementation of autonomous vehicles would further lead to a greater computation demands.

Due to the above reasons, this work intended to propose an alternative low computation predictive controller known as Predictive Functional Control (PFC), which is novel in the ACC application. In general, the algorithm principle is the same, but the optimization process is simplified without the use of the quadratic cost function. In return, the computation time can be reduced significantly [11]. The PFC algorithm has also been implemented in other engineering fields ranging from chemical, mechanical, and electrical industries [12,13]. However, there is a tradeoff in using PFC, where the optimality of the control solution will be reduced. This is a well-known tradeoff but for a Single Input Single Output (SISO) application such as an ACC system, it may become a good alternative strategy. Besides, there are also a number of works that have improved the existing algorithm such as [14,15]. Thus, the main objective of this work is to investigate and provide a formal performance comparison of PFC with MPC and a traditional PI controller to assess its capability in a vehicle following application.

Another important issue that needs to be highlighted is the application of PFC for marginally stable processes or type 1 processes where there will be an independent integrator in the transfer function's denominator. Since the input to the ACC system is acceleration and the output is speed, if a step input is given, the output will not converge to a steady state value. Thus, when a conventional PFC algorithm is implemented, the control solution will be ill-posed, where the prediction and actual control performance will differ [16]. To cope with this issue, a cascade PFC with feedback compensator will be used where the prediction structure is pre-stabilized before implementing the conventional algorithm [17]. A proper analysis on the tuning process will be discussed and denoted as one of the contributions of this works.

2. METHODOLOGY

2.1 Vehicle Longitudinal Dynamics Modelling

The mathematical model is the most important part of predictive control formulation. In general, the inputs to the system will be the pedal pressure from the throttle and brake, while the output of interest will be velocity. In a traditional cruise control system, the braking effect is not considered, thus the standard kinetic vehicle longitudinal model can be used straight away to derive the control law of PFC as presented in [18]. However, the ACC system needs braking input as well to keep the safe distancing with a lead vehicle. If a full kinetic modelling is used to derive the control law as in [18], the constraint implementation will be quite complicated especially if it is designed for full range dynamic operation where the nonlinearities will be very high due to the changing gear ratio and coefficient of friction between road and tire [2]. Thus, a hierarchal control strategy is often proposed in the literature [5,6] as shown in Fig. 1.

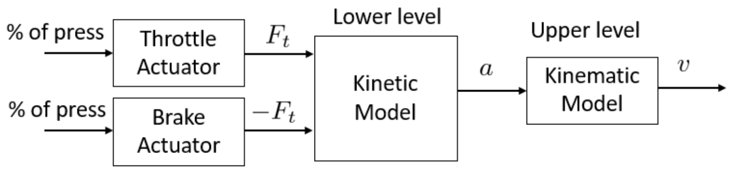

Fig. 1: Hierarchical structure for vehicle longitudinal dynamics.

In the hierarchal structure, the throttle and brake actuator correspond to powertrain and brake line dynamics respectively. As for the lower-level dynamic, it is responsible for the relationship between traction force F_t and acceleration, a, by Newton's second law. The details of these models are available in many references including academic textbooks [2]. Since the lower-level model is nonlinear in nature, the PFC controller will be derived based on the upper-level system that represents the kinematic dynamic between acceleration and velocity where the transfer function is given as:

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

The additional time constant τ in Eq. (1) corresponds to the estimation of the first order lag of the lower-level controller [2]. In this case, it is expected that the car will track the velocity imperfectly and thus the nominal time constant τ is approximately equal to 0.5 s [2]. In this work, it is assumed that the lower-level controller can retain the nominal value in every operating speed, although in reality, it will be changed due to nonlinearities, especially for low-speed operation where the gear ratio is not constant. The investigation of the impact of nonlinearities on the overall control performance of an ACC system and the possibility of using other robust control methods will denote future work.

2.2 Cascade PFC Prediction

Since the PFC algorithm works in the digital domain, the transfer function in Eq. (1) needs to be discretized a with sampling time of 0.1 s [6]. From here on, a standard symbol for input u will be used for acceleration a and output y for velocity v. The discrete transfer function can be represented as:

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

Another important part that needs to be considered is the nature of the transfer function in Eq. (1) and Eq. (2), where it is a type 1 system with independent integrator in the denominator. Since the output response will not converge when given a bounded input, it needs to be pre-stabilized before deriving the control law. If not, it will lead to an ill-posed solution as reported in the literature [14]. There are two ways to overcome this issue [14,17], one is by splitting Eq. (2) into two separate transfer functions that run in parallel, one using partial fractions and the other using a proportional gain, K, to stabilize the

https://doi.org/10.31436/iiumej.v24i1.2341

system as shown in Fig. 2. This work utilizes the second option since only a minor modification in the computed input is required when implementing the control law.

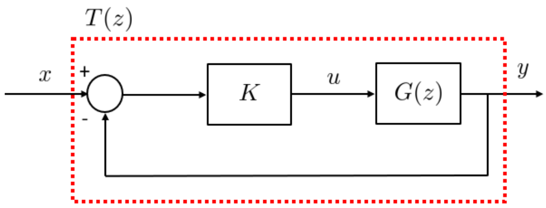

Fig. 2: Proportional feedback loop for plant pre-stabilization.

The inner loop $T(z)$ will be used as a prediction model instead of $G(z)$ as in Eq. (2) to get a stable response and it is computed as:

$$T(z) = \frac{y(z)}{x(z)} = \frac{G(z)K}{1+G(z)K} \tag{3}$$

where, $x(z)$ is the modified controlled input. A suitable value of proportional gain K can be selected based on trial and error or by using a more systematic approach as presented in [17]. As in this work, the MATLAB PID tuner is utilized. Using the function *pidtune(sys,type),* it provides the proportional value of $K = 1.147$. According to the function description in MATLAB help desk, the given value is computed based on the balance performance between response time and robustness.

A linear prediction structure based on the superimposed principle for transfer function in Eq. (3) can be formed, and since the derivation is standard [23-26], only the final form is presented. The i-step ahead prediction at the k sampling is presented as:

$$y(k+i|k) = HX + PX_0 + QY + d \tag{4}$$

The dimension of matrix H, P, and Q depends on the model parameters, and for a standard second order transfer function, the parameters X, X_0 and Y are:

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

The term d in Eq. (4) corresponds the correction term to get an unbiased prediction using an independent model structure as shown in Fig. 3. Technically, it is the difference between the actual measured output form a plant y_p and the calculated output y from $T(z)$ in Fig. 2.

$$d = y_p - y \tag{5}$$

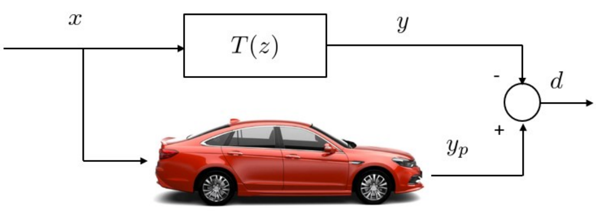

Fig. 3: Independent model structure for PFC prediction.

2.3 Cascade PFC Control Law

To compute the control input, the prediction in Eq. (4) at n-step ahead is forced to coincide with a first order setpoint trajectory r, which is given as [11]:

$$r(k+n|k) = (1-\lambda^n)R + \lambda^n y_p(k) \tag{6}$$

where R is the desired set point speed. In this stage, there are two tuning parameters that need to be specified. The first one is the coincidence horizon n, where the prediction in Eq. (4) is forced to match with the target trajectory in Eq. (6). The second one is λ, which corresponds to the desired Closed-loop Time Response (time taken to reach 95 % form the steady state value):

$$\lambda = e^{-3T_s/CLTR} \tag{7}$$

The term T_s in Eq. (7) denotes the sampling time. For a second order system, the selection of n has a general tuning guideline as presented in [17]. It should be selected between 40% and 80% of the response to avoid prediction mismatch. Once all the tuning parameters have been selected, the general control law of PFC can be derived. Using its standard assumption that the future modified control input will be constant i.e., $x(k) = x(k+1) = \cdots . x(k+n)$ [25], the n-th row of matrix H_n will reduce to a single value in the form of:

$$h_n = H_n \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{8}$$

Extracting the n-th prediction form each matrix and equating the predictions in Eq. (4) and Eq. (6) gives:

$$h_n x(k) + P_n X_0 + Q_n Y + d = (1-\lambda^n)R + \lambda^n y_p(k) \tag{9}$$

The compensated input x can be computed as:

$$x(k) = h_n^{-1}\left[(1-\lambda^n)R + \lambda^n y_p(k) - P_n X_0 - Q_n Y - d\right] \tag{10}$$

It should be noted that since a cascade structure is used to pre-stabilized the response as explained in Section 2.2. By referring to Fig. 4, the actual input that will be sent to the plant is computed as:

$$u(k) = K\left[x(k) - y_p(k)\right] \tag{11}$$

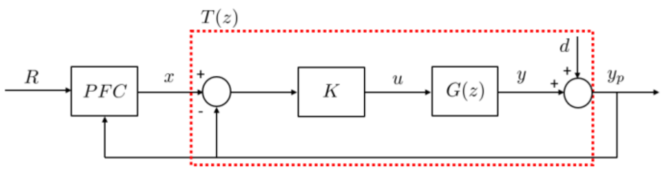

Fig. 4: Cascade PFC structure.

2.4 Constraint Handling for Passenger Comfort and Safe Distancing

For the predictive controller framework, the conventional switching strategy from speed mode to space mode can be implemented by formulating a constraint control problem. Fig. 5 shows the schematic diagram for vehicle following application and how the ACC controller is implemented.

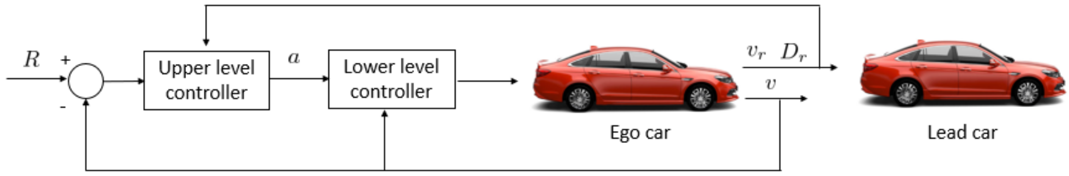

Fig. 5: ACC schematic diagram for vehicle following application.

With the help of a distance sensor such as LIDAR, a relative distance D_r or relative velocity v_r with a lead vehicle can be measured. These two parameters can be used to regulate the acceleration to retain a safe following distance if necessary. A standard safe following distance equation as recommended in [2] is used, where:

$$D_{safe}(k) = D_{default} + T_{gap}v(k) \tag{12}$$

The default distance $D_{default}$ is set to 10 m, it means that if the velocity of vehicle is 0 *m/s* (both lead and ego vehicle are not moving or approaching a complete stop), the relative distance should not be less than $D_{default}$. The safe time gap T_{gap} between the vehicle is set to 1.4 *s*. Based on Eq. (12), the higher the velocity of a vehicle, the larger the safe distance needed to be maintained because it will take more time to slow down the car. Based on the output velocity prediction in Eq. (4), 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

$$D_r(k+1) = [v_l - v(k)]T_s + D_r(k) \tag{13}$$

Using superposition, at each sampling time, the maximum velocity to keep the safe distance can be formed as:

$$v_{max} = \left(v_l T_s + D_r(k) - D_{default}\right)/\left(T_{gap} + T_s\right) \tag{14}$$

Then a normal PFC output constraints formulation can be implemented by selecting a suitable validation horizon n_i. The algorithm is given as below:

Algorithm (A). *At each sample time*:

(1) Compute the unconstrained compensated input x as in Eq. (10).

(2) A simple 'for' loop is used to check the constraint violation while updating Eq. (13) and Eq. (14)

(3) If $v > v_{max}$*, then:*

$$x(k) = HL_i^{-1}[v_{max}(i) - P_i U_0 - Q_i Y - d] \tag{15}$$

(4) Else, the value of x is retained.

For taking care of passenger comfort and fuel efficiency, the input acceleration is constrained in between u_{min}= -3 to u_{max}= 2 m/s^2. As discussed in many PFC papers [16-17] a simple clipping strategy is enough to cater for input constraint such that if $x < y(k) + u_{min}/K$, then:

$$x(k) = y(k) + u_{min}/K \tag{16}$$

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

$$x(k) = y(k) + u_{max}/K \tag{17}$$

3. SIMULATION RESULT

3.1 Analysis of Tuning Parameter of PFC

The first step in tuning the PFC is to select a suitable compensator gain to stabilize the open-loop behavior of the vehicle dynamics. Since the open-loop transfer function is type 1 (with one integrator in the denominator), a single proportional gain is enough to stabilize the system. As discussed in Section 2.2, an autotune PI controller in MATLAB is used to find an optimum proportional gain K = 1.147 that will provide a balanced performance between response time and robustness [17]. Figure 5 shows the open-loop response of the pre-stabilized system output y against the desired output trajectory R with CLTR of 5s (λ = 0.9418).

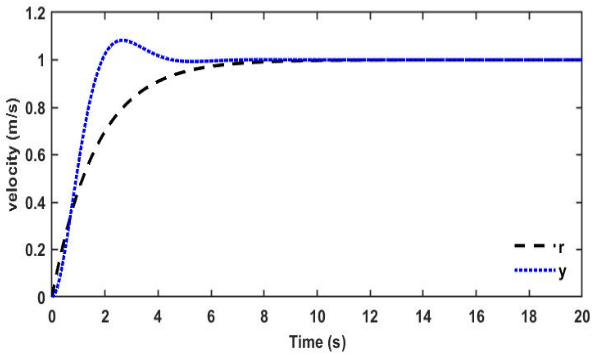

Fig. 5: Pre-stabilized open-loop response compared to the desired trajectory.

The objective of the PFC is to force the system to match the desired target trajectory at a specific coincidence horizon, n. In this stage, a proper coincidence horizon needs to be selected carefully. In general, a smaller coincidence horizon will provide more aggressive response and input, while higher coincidence horizon will be less aggressive and slow. The effect of coincidence horizon may be varied for different systems since the dynamics are different. Several coincidence horizons are simulated to track a desired velocity of 5 m/s with 0.1 s sampling time, as shown in Fig. 6.

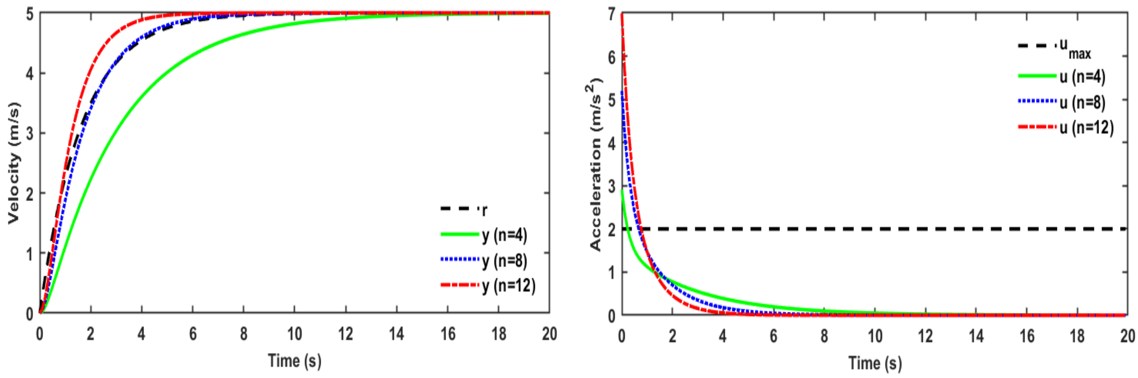

Fig. 6: Closed-loop response with varying coincidence horizon (n).

It can be observed that a smaller horizon will lead to slower response and a larger value will lead to faster response than the desired one (black dashed line). Based on the tuning guide proposed by [17], a suitable value should be selected between 40% and 80% rise of step response to its steady state value. In this case n=8 provides the closest response compared to the desired trajectory. A similar effect can be seen in the control

https://doi.org/10.31436/iiumej.v24i1.2341

effort where large horizon will need larger over actuation compared to the smaller horizon value. Nevertheless, all the control efforts violate the maximum allowable input acceleration, which is 2 m/s^2.

To satisfy the driving constraint, the algorithm (A) in section 2.4 is implemented. As expected, Fig. 7 shows the response where the implied constraint is satisfied systematically, yet with a slower response in tracking the desired target trajectory.

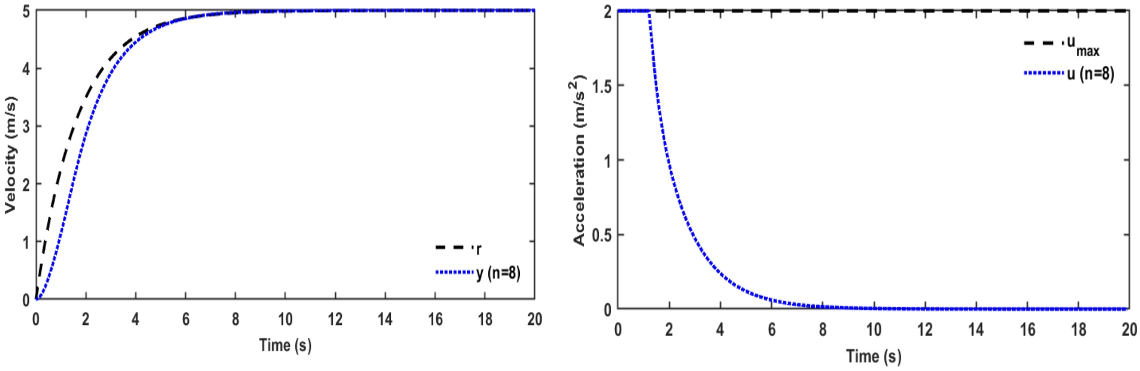

Fig. 7: Closed-loop response when input constraint is activated.

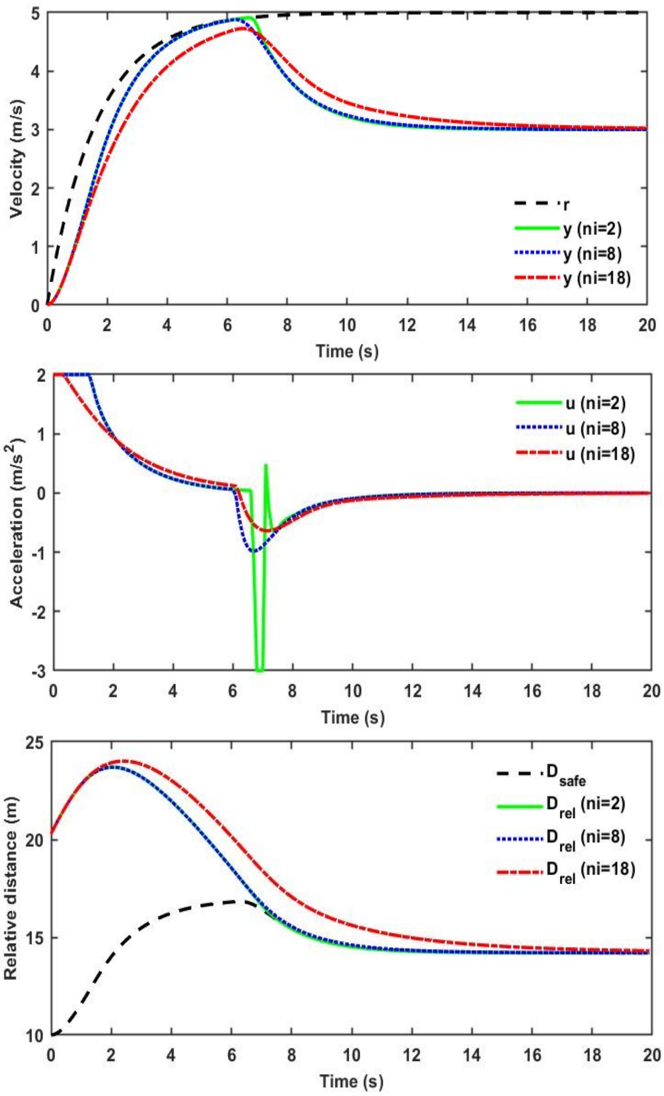

Fig. 8: Closed-loop response for vehicle following application.

For the *car following application*, the output velocity also needs to be constrained to ensure that the safe following distance is always respected. With the prediction capability of PFC, it can anticipate the future relative distance and change the input accordingly. Nevertheless, a suitable validation horizon (how far ahead the constraint is implemented) needs to be selected. Figure 8 overlays different selections of validation horizons when following a lead vehicle with a constant velocity of 3 *m/s* and initial relative distance of 21 m. As can be observed, all three responses managed to prevent the relative distance from going lower than the safe following distance and respecting all the input constraints. Nevertheless, there is a clear tradeoff in the selection of validation horizon, if it is too short (n_i=2 green solid line), an aggressive control effort is needed. Nevertheless, if it is too large (n_i=18, red dash dotted line), the response may be too conservative, and the computation burden of the controller will increase as it needs to compute more mathematical operation. For this case, a validation of n_i=8 (blue dotted line), which is equal to the coincidence horizon, is selected since it respects all the constraints with optimum control effort. This feature is very important to ensure the safety and comfort of the passengers.

3.2 Performance Comparison with PI and MPC

To analyze the control performance of PFC, its closed-loop response is compared with two benchmark controllers: PI and MPC. The reason the PI controller is used instead of the PID is because it is a type 1 system. Figure 9 shows the comparison of the input acceleration, output velocity and relative distance of the three controllers. The lead vehicle (black-dashed line) is assumed to operate with sinusoidal acceleration signal with amplitude of 0.35 m/s^2 and frequency of 0.1 *rad/s*. The ego vehicle (the controlled vehicle) is set to track a desired velocity of 30 *m/s* while respecting the safe following distance. For predictive controllers (PFC and MPC), the safe following distance is treated as a constraint, while for PI controller, a simple switching strategy is used to change between speed and space mode. The gains are selected based on an auto tune function in MATLAB with P = 0.8, I = 0.001. At the same time, input acceleration is constrained between -3 m/s^2 to 2 m/s^2 for passenger comfort.

As expected, it can be observed that from the result in Fig. 9, MPC (green dashed-dotted line) provides the most optimal response with less aggressive input acceleration followed by PFC (blue dotted line) and PI (red solid line). The difference in output velocity is not quite significant except that a large spike response is generated by the PI controller at 10 *s* when the ego car is trying to adjust its speed to match the leading vehicle. As for the safe distancing, both PFC and MPC managed to retain it, while PI needs to violate the safe distancing to satisfy input acceleration constraints at 60 *s*, 120 *s* and 180 *s*.

Besides the qualitative results, some quantitative values are shown in Table 1 for each controller to numerically demonstrate their output velocity performance in terms of rise time τ_r, settling time τ_s and Root Mean Square Error (RMSE). These quantitative values are measured over the duration of 40 *s* for tracking 30 *m/s* from rest and will be the performance indices of the controllers. It can be observed from Fig. 9 and Table 1 that the PI controller for the constrained closed-loop response requires longer settling times (199.5386 *s*) as compared to others although it has the fastest rise time (2.7213 *s*). Another observation is that all the controllers do not produce an overshoot. However, the MPC and PFC provide the best fit response curve to the desired speed of 30 *m/s* with lower RMSE of (1.5501) and (1.4873) respectively compared to PI controller RMSE of (1.6219).

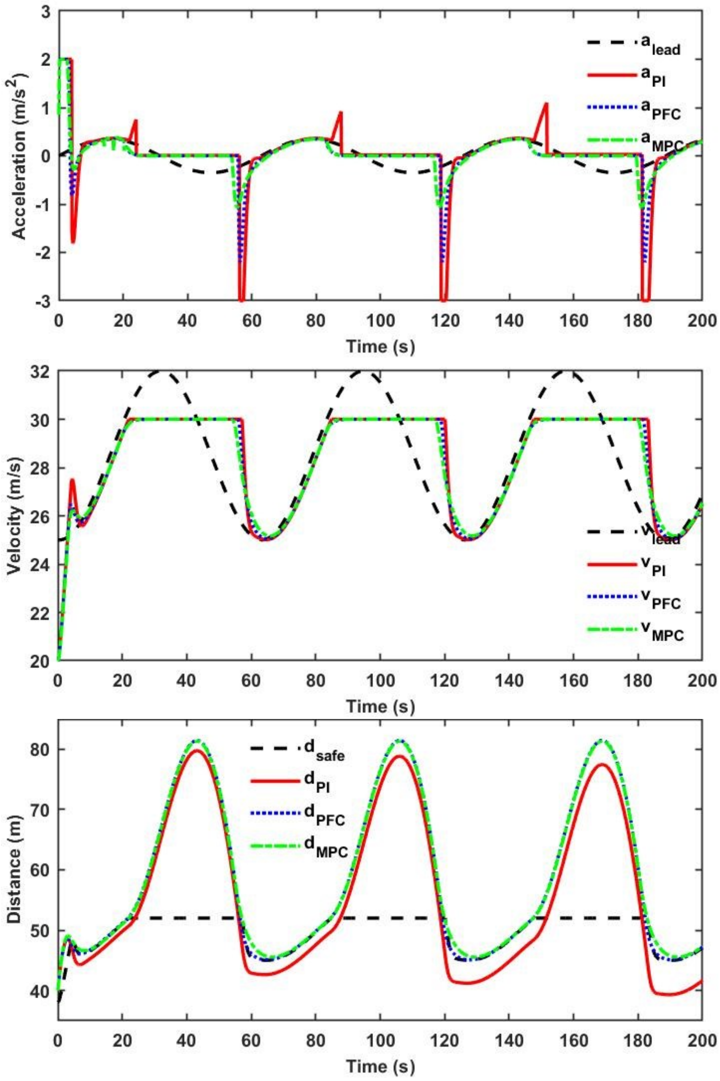

Fig. 9: Closed-loop performance comparison between PI, PFC, and MPC for vehicle following application.

Table 1: Performance indices for all controllers.

Performance Criteria	PI	PFC	MPC
Rise time, τ_r	2.7213	2.7348	2.9551
Settling time, τ_s	199.5386	199.5348	199.5295
RMSE	1.6219	1.5501	1.4873

Based on the analysis, it can be explained that the prediction capability is very important in an ACC system to provide optimal response. By estimating a future behavior, a better control input can be generated while satisfying the constraints systematically. Nevertheless, there is an obvious difference between PFC and MPC because the optimization algorithm is simplified to reduce the computation burden. However, it worth pointing out that PFC performance is better compared to PID. The main reason is because the PI computes the control action based on the current measurement rather than prediction. Indeed, one may argue that if it is tuned properly or a suitable special switching strategy is used, a better performance can be obtained. Nevertheless, it also should be noted that the tuning procedure is not as straightforward as expected since many trial-and-error procedures need to be implemented.

4. CONCLUSION

In summary, this paper describes a design that proposes the use of PFC in controlling the ACC system of a vehicle. Based on the simulation results, it is found that although the PFC performance with RMSE of 1.4873 is not comparable with the more advanced MPC algorithm with RMSE of 1.5501, it gives a better performance compared to the PI controller with RMSE of 1.6219. Besides, the PFC algorithm is also simple and straightforward to implement and requires less calculation compared to the MPC while retaining its predictive advantage compared to other traditional controllers. The tuning process is also intuitive, being based on the desired time constant and coincidence horizon selection. Indeed, there is a lot of room for improvement and future work will provide a more systematic analysis of the computation requirement of both MPC and PFC controllers and their robustness properties. Hence, it can be concluded that PFC can become a good alternative to MPC and PI by trading off some of the optimality properties for a lower computation burden.

ACKNOWLEDGEMENT

The authors would like to acknowledge the Ministry of Higher Education, Malaysia for funding this work under the Fundamental Research Grant Scheme FRGS/1/2021/TK02/UIAM/02/2 (FRGS21-240-0849). The first author would like to acknowledge Majlis Amanah Rakyat (MARA) for his tuition fee waiver.

REFERENCES

[1] Jiang Y, Deng W, He R, Yang S, Wang S, Bian N. (2017) Hierarchical framework for adaptive cruise control with model predictive control method (No. 2017-01-1963). SAE Technical Paper.

[2] Rajamani R. (2011) Vehicle dynamics and control. Springer Science & Business Media.

[3] Haroon Z, Khan B, Farid U, Ali SM, Mehmood CA. (2019). Switching control paradigms for adaptive cruise control system with stop-and-go scenario. Arabian Journal for Science and Engineering, 44(3): 2103-2113.

[4] Alomari K, Mendoza RC, Sundermann S, Goehring D, Rojas R. (2020). Fuzzy Logic-based Adaptive Cruise Control for Autonomous Model Car. In ROBOVIS (pp. 121-130).

[5] Phan D, Amani AM, Mola M, Rezaei AA, Fayyazi M, Jalili M., ... Khayyam H. (2021). Cascade Adaptive MPC with Type 2 Fuzzy System for Safety and Energy Management in Autonomous Vehicles: A Sustainable Approach for Future of Transportation. Sustainability, 13(18): 10113.

[6] Takahama T, Akasaka D. (2018). Model predictive control approach to design practical adaptive cruise control for traffic jam. International Journal of Automotive Engineering, 9(3): 99-104.

[7] Li SE, Jia Z, Li K, Cheng B. (2014). Fast online computation of a model predictive controller and its application to fuel economy–oriented adaptive cruise control. IEEE Transactions on Intelligent Transportation Systems, 16(3): 1199-1209.

[8] Guo L, Ge P, Sun D, Qiao Y. (2020). Adaptive cruise control based on model predictive control with constraints softening. Applied Sciences, 10(5): 1635.

[9] Borek J, Groelke B, Earnhardt C, Vermillion C. (2019, July). Optimal control of heavy-duty trucks in urban environments through fused model predictive control and adaptive cruise control. In 2019 American Control Conference (ACC) (pp. 4602-4607). IEEE.

[10] Awad N, Lasheen A, Elnggar M, Kamel A. (2022). Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles. ISA transactions, 129: 193-205.

[11] Rossiter JA. (2018). A first course in predictive control. CRC press.

[12] Nasiri Soloklo H. (2018). Predictive Functional Control for Tracking of Core Power Variations in Pressurized Water Reactor based on Laguerre functions and Reduced-Order Model. Modares Mechanical Engineering, 18(1): 299-306.

[13] Li MY, Lu KD, Dai YX, Zeng GQ. (2020). Fractional-Order Predictive Functional Control of Industrial Processes with Partial Actuator Failures. Hindawi Complexity, 2020: 1-25.

[14] Abdullah M, Rossiter JA. (2018). Input shaping predictive functional control for different types of challenging dynamics processes. Processes, 6(8): 118.

[15] Abdullah M, Rossiter JA, Ghaffar AFA. (2021). Improved constraint handling approach for predictive functional control using an implied closed-loop prediction. IIUM Engineering Journal, 22(1): 323-338.

[16] Abdullah M, Rossiter JA. (2021). Using Laguerre functions to improve the tuning and performance of predictive functional control. International Journal of Control, 94(1): 202-214.

[17] Rossiter JA, Aftab MS. (2021). A Comparison of Tuning Methods for Predictive Functional Control. Processes, 9(7): 1140.

[18] Zainuddin MAS, Abdullah M, Ahmad S, Tofrowaih KA. (2022). Performance Comparison Between Predictive Functional Control and PID Algorithms for Automobile Cruise Control System. International Journal of Automotive and Mechanical Engineering, 19(1): 9460-9468.