Journal of Control, Automation and Electrical Systems (2019) 30:9–15
https://doi.org/10.1007/s40313-018-00422-1

Adaptive Cruise Control with a Customized Electronic Control Unit

Mateus Mussi Brugnolli[1] · Bruno Silva Pereira[1] · Bruno Augusto Angélico[1] · Armando Antônio Maria Laganá[1]

Received: 10 May 2018 / Revised: 20 July 2018 / Accepted: 24 September 2018 / Published online: 8 October 2018

Abstract

This paper aims the development of adaptive cruise control (ACC) system, a technology of advanced driver assistance systems (ADAS), in an embedded application. This research has at its disposal a vehicle with a customized electronic control unit, with functions to support the study of ADAS. An ACC module was produced to allow the insertion of embedded controllers, communicating with the vehicle network and with a radar for future on-road applications. The dynamic model of the vehicle was estimated using the system identification theory, and a model validation was performed. The control system was divided in a cascade control loop, being the outer loop controller responsible for computing the cruise speed and the inner loop for tracking such speed. The outer loop controller was performed using a switching logic between cruise control and ACC modes. The inner loop controller was designed with the Dahlin control theory. The validation of the controllers was performed using a safe and controlled environment, with a dynamometer.

Keywords Adaptive cruise control · Digital control · Electronic control unit · Dynamometer

1 Introduction

Nowadays, one of the most discussed research areas of the automotive engineering is the application of autonomous driving. Due to its high complexity, the study of autonomous driving can be split in several systems, for example, navigation systems (Lima and Pereira 2013) and advanced driver assistance systems (ADAS).

The ADAS are technologies developed to help a driver to handle a vehicle more easily and safely. One of these technologies is the cruise control (CC), a system that controls the longitudinal speed of the vehicle to achieve a desired cruise speed.

The cruise control has been improved in the last decades, creating the adaptive cruise control (ACC) systems, which aims to compute a new cruise speed to maintain a safe distance to leading vehicles (Shakouri et al. 2015). The traffic behavior does change while using CC or ACC systems, and it is already being examined (MarkVollrath 2011). Furthermore, the energy consumption can also be influenced by using ACC systems (Khayyam et al. 2012).

Usually in ACC researches, the control design is divided into two separate loops, an inner and an outer, as can be seen in Fig. 1. For inner loop control, there are two input signals: cruise speed and system (vehicle) speed. This loop outputs are the signals for the actuators of the system. There are three input signals for outer loop control: user-defined speed, vehicle speed and all important radar data, for example distance from the closest vehicle and its relative speed.

Many control theories have been applied in ACC systems, for example, intelligent controllers (Kuyumcu and Şengör 2016), sliding mode control (Ganji et al. 2014) and model predictive control (MPC) (Magdici and Althoff 2017; Li et al. 2017). Among these controllers, the MPC has been one of the most discussed ones. This is due to its capability to insert constraints to improve the performance (Luo et al. 2016) and the robustness (Filho et al. 2017).

However, in ACC researches, most of the papers do not consider validation with practical experiments. Those that do that, implement the controller using a computer and a servomechanism for accelerating and/or breaking using the pedals.

✉ Armando Antônio Maria Laganá
lagana@lsi.usp.br

Mateus Mussi Brugnolli
mateus_mmb@usp.br

Bruno Silva Pereira
bruno.sp@usp.br

Bruno Augusto Angélico
angelico@lac.usp.br

[1] Escola Politécnica da Universidade de São Paulo, São Paulo, SP, Brazil

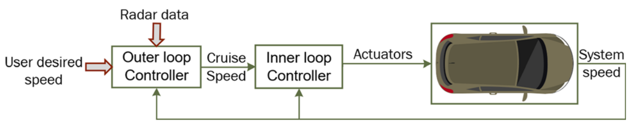

Fig. 1 Representation of an ACC System

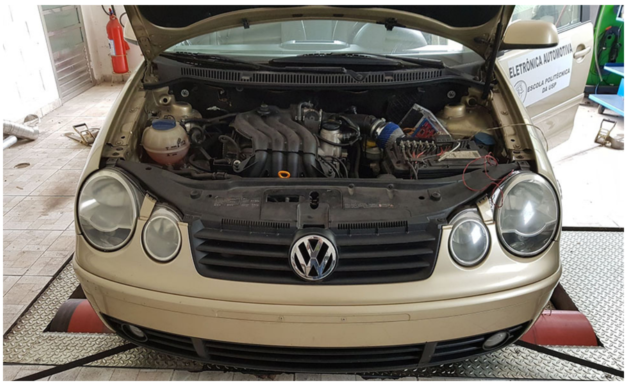

Fig. 2 Photograph of Polo vehicle

Fig. 3 Photograph of customized ECU

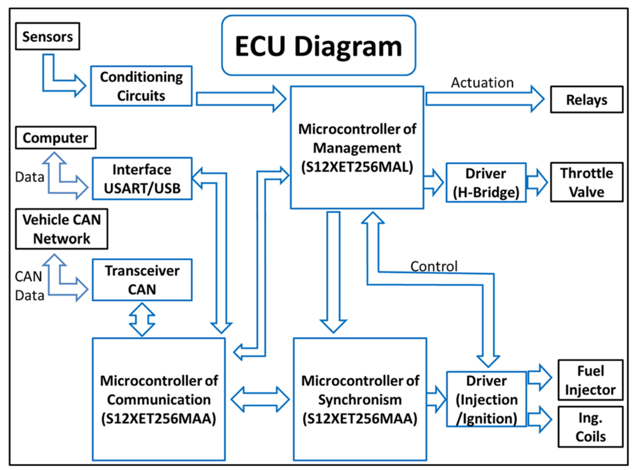

Fig. 4 Diagram of the ECU

This work aims to create an ACC system and carry out performance tests in a safe and controlled test environment. After obtaining a suitable model for the vehicle, the controller is designed and its algorithm is embedded in a customized electronic control unit (ECU).

The vehicle and the test environment are described in Sect. 2. The method to obtain a suitable model for the vehicle and the design of the controllers are addressed in Sect. 3. In Sect. 4, the practical results of the controllers are featured. Lastly, some conclusions and final considerations are provided in Sect. 5.

2 Vehicle and Test Setup

The platform available for this research consists of the following components: a Volkswagen vehicle , Polo Sedan model with spark-ignition engine 2.0 L, shown in Fig. 2, controlled by an open-source ECU; an inertial dynamometer from NAPRO company; a module for ACC management; and a long-range radar, ARS300 model from Continental, for future on-road applications.

The open-source ECU, shown in Fig. 3, has been developed by the Automotive Electronics Group from Escola Politécnica da Universidade de São Paulo (EPUSP) aiming the research of ADAS. There are other researches that also developed customized ECU, such as a study for reducing the NOx emissions of a diesel engine (Ferreira et al. 2014).

The ECU was developed using a decentralized architecture of three microcontrollers, shown in Fig. 4. Each microcontroller operates at 50 MHz and has a distinct functionality, being for management, synchronism and communication. The microcontrollers communicate with each other using SPI (Serial Peripheral Interface) at 12.5 MHz. The communication with the customized ECU is available via CAN network of the vehicle and via USB for the supervisory software on a computer.

This ECU has been improved in the last 5 years providing power and torque performance and driving characteristics equivalent to the vehicle factory ECU. To achieve such results, several works were realized: open-loop torque control, stoichiometric control of air and fuel mixture, electronic throttle valve position control and engine idling speed control with phase lead compensation for the ignition.

Furthermore, many other features have been implemented: identification of the set gear; speed calculation; increase in start engine idling speed; SPI communication security between the blocks of the ECU, using cyclic redundancy check (CRC); a supervisory software for computers; multiple operation modes for the ECU (normal, economic and sports); starting engine control; communication with the dashboard

Table 1 CAN Messages for the ACC System

CAN Message	ID (Hex)	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Input signal	200	"E" Token	"C" Token	"U" Token	Input accelerator pedal (0–100%)				
Configuration	201	"E" Token	"C" Token	"U" Token	Input pedal mode (On or Off)	Set operation mode	Idling speed mode	–	–
Output signal	590	Accelerator pedal (0–100%) factor 0.01		Measured speed (Km/h) factor 0.01		Gear engaged	Read operation mode	–	Error code

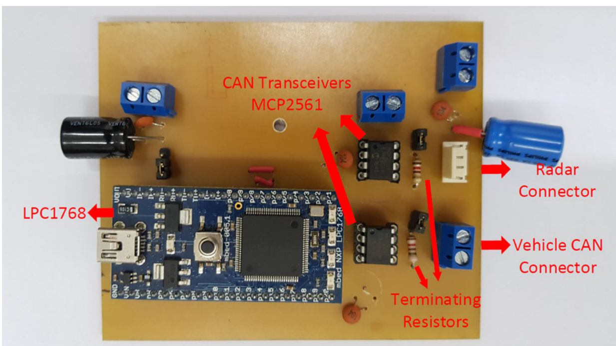

Fig. 5 Photograph of ACC Module

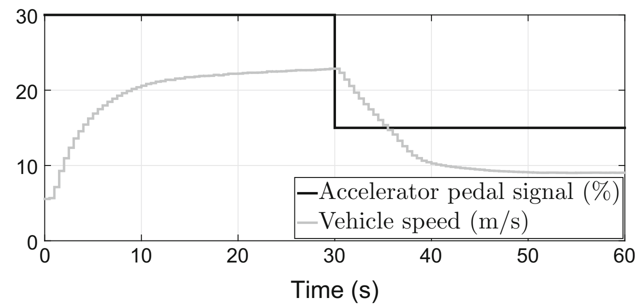

Fig. 6 System step response

of the vehicle; identification and correction of eventual faults; and on-board diagnosis (OBD) (Pereira 2015, 2017).

The CAN messages used for this work are presented in Table 1. These messages were customized to easily operate and read the most relevant variables for the ACC system.

The ACC module, shown in Fig. 5, communicates with the ECU and, in the future, with the ARS300 radar. The microcontroller LPC1768 that is compatible with the Mbed compiler has two CAN channels. These features will be essential because the CAN network frequencies of the vehicle and the radar are different; therefore, the ACC module will act as an intermediary between the two CAN networks. The ACC algorithm is embedded in the LPC1768.

3 System Identification and Control Design

Although the vehicle has a customized ECU, it does not have an electronic brake system. Thus, this research discarded using the brake pedal as a control input. The control input chosen for the vehicle is the accelerator pedal, whose range varies from 0% (not pushed) to 100% (fully pushed). The vehicle output of an ACC system is the translational vehicle speed. Both signals have specific CAN Messages, as shown in Table 1. In this work, the system was considered as a single-input single-output (or SISO).

In order to design the control system, a suitable vehicle model was required. The system identification theory was chosen to obtain a vehicle model given some restrictions (Ljung 1999). The identification and control experiments were established with the vehicle set in the third gear. For safety purposes and due to the capability to change the dynamometer load, the experiments were also performed using the dynamometer.

For an experiment of 60 s, in the first 30 s a CAN message of 30% of accelerator pedal was sent to the ECU. In the last 30 s, this CAN message was changed to 15% of accelerator pedal as input. Using a sampling time of $T_s = 0.5$ s ($f_s = 2$ Hz), the system step response is presented in Fig. 6.

System time constant τ can be estimated as the time interval for the system output to achieve 63% of output variation caused by the step input. However, the step response pointed out that the system has a time delay of one sampling time of $\theta = 0.5$ s. Additionally, the system time constant is approximately $\tau = 5$ s. Then, a sampling time of $T_s = 0.5$ s, which is 10 times faster than τ, is suitable for identification and control of the system (Seborg et al. 2004).

The input and output of the step response experiment were chosen as identification data, and the model representation

Table 2 FIT (%) for system validation, with several configurations of steps ahead

Steps	1	2	5	10	20	50	∞
FIT	97.19	96.08	93.10	92.23	93.07	93.52	91.08

was the autoregressive with exogenous input (ARX) model (Ljung 1999), with the following structure:

$$y(t) + a_1 y(t-1) + \cdots + a_{na} y(t-na) \\ = b_1 u(t-nk) + \cdots + b_{nb} u(t-nk-nb+1) + e(t) \quad (1)$$

With an input u and an output y, an ARX model (Eq. (1)) calculates all a_i and b_i coefficients to reduce the error $e(t)$. The system time delay is defined as nk, na is the order of the output polynomial, and nb is that of the input polynomial.

Given $T_s = 0.5$ s, the time delay order was defined as $nk = 1$, since the time delay for step response was $\theta = 0.5$ s. Once the system step response represents a typical first-order model, the chosen orders of the polynomials were $na = 1$ and $nb = 1$. Such hypothesis should be checked with a model validation. With the aid of MATLAB®, the computed ARX model is represented in Eq. (2).

$$y(t) - 0.903y(t-1) = 0.085u(t-1) + e(t) \quad (2)$$

To simplify the visualization of the ARX model, its corresponding representation of transfer function in continuous and discrete time are shown in Eq. (3). The time constant $\tau = 4.8983$ was close to the estimated value of 5 s.

$$G_c(s) = e^{-0.5s} \frac{0.8762}{4.8983s + 1}; \quad G_d(z) = z^{-1} \frac{0.085}{z - 0.903} \quad (3)$$

A brief model validation was also performed analyzing the FIT indexes of the ARX model, while predicting the system output. The results are shown in Table 2, pointing out that even in the worst case (prediction with infinite steps ahead), the FIT index had a remarkably high value.

In this research, one controller is designed for the outer loop and another controller for the inner loop, as shown in Fig. 1. The outer loop controller must be able to change between cruise control (CC mode) and adaptive cruise control (ACC mode). For CC mode, the user-defined speed should be the cruise speed, while for ACC mode, a new cruise reference speed must be calculated to maintain a safe distance.

3.1 Outer Loop Controller

The algorithm for the outer loop controller was adapted from Shakouri et al. (2015). The following are considered: the vehicle speed as v, the user-defined speed as v_{user}, the measured distance d with the closest vehicle, a security distance d_{ref} with the closest vehicle, relative speed (v_r) between the closest vehicle and the system vehicle.

The controller for the outer loop must compute d_{ref} to maintain a safe distance to the leading vehicle. In Shakouri et al. (2015), the authors suggest computing d_{ref} as in Eq. (4). The parameter ℓ is the length of the system vehicle, d_s it is an additional distance to avoid crashes, and T_h is known as constant-time headway, which estimates a human driver reaction time. Usually, this value varies between 0.8 and 2 s.

$$d_{\text{ref}} = \ell + d_s + T_h v \quad (4)$$

The same authors suggest computing a reference speed, named as v_{ref}, as shown in Eq. (5) (Shakouri et al. 2015). If the distance d for the closest vehicle is same as the security distance d_{ref}, the system is safe to travel with the same speed of the closest vehicle v_l, commonly described as leading vehicle. Otherwise, it is necessary to change the security speed v_{ref} using a proportional gain K_p.

$$v_{\text{ref}} = v_l - K_p(d_{\text{ref}} - d) \quad (5)$$

An adapted control law is shown in Algorithm 1. Firstly, the controller computes d_{ref} and v_{ref}. Next, it compares whether the reference speed is greater than the user-defined speed. If this statement is true, the cruise speed must be limited to v_{user}, causing the controller to enter in CC mode. Otherwise, the cruise speed will be adjusted to v_{ref}, to keep a safe distance from the leading vehicle (ACC mode).

Algorithm 1 Outer loop controller

1: $d_{ref} = l + d_s + T_h v$;
2: $v_{ref} = v_l - K_p(d_{ref} - d)$;
3: If $v_{ref} > v_{user}$, $v_{cruise} = v_{user}$ (CC Mode);
4: Otherwise, $v_{cruise} = v_{ref}$ (ACC Mode).

3.2 Inner Loop Controller

Several control theories can be applied for the inner loop control. This work focuses on Dahlin control, also known as Lambda tuning (Seborg et al. 2004). This method specifies a desired closed-loop transfer function ($T_{\text{cl}}(s)$) as follows:

$$T_{\text{cl}}(s) = \frac{v(s)}{v_{\text{cruise}}(s)} = \frac{e^{-\theta s}}{\lambda s + 1} \quad (6)$$

The desired closed-loop transfer function is represented by a system of first order with a time constant equal to λ and with θ seconds of time delay. The control design parameter is

λ, given that θ is the time delay of the system. The identified ARX model had a similar transfer function; consequently, this control technique seems to be suitable. Discretizing the transfer function $T_{\text{cl}}(s)$, the Dahlin controller ($C(z)$) to achieve such performance is described in Eq. (7).

$$\begin{aligned}\frac{C(z)G_{\text{d}}(z)}{1+C(z)G_{\text{d}}(z)} &= T_{\text{cl}}(z)\\ C(z)G_{\text{d}}(z) &= T_{\text{cl}}(z)+T_{\text{cl}}(z)C(z)G_{\text{d}}(z)\\ C(z)G_{\text{d}}(z)(1-T_{\text{cl}}(z)) &= T_{\text{cl}}(z)\\ C(z) &= \frac{1}{G_{\text{d}}(z)}\frac{T_{\text{cl}}(z)}{1-T_{\text{cl}}(z)}\end{aligned} \tag{7}$$

4 Practical Results

Firstly, the tuning of outer loop controller depends on two main variables of the ACC controller: d_{ref} and v_{ref}. The time headway T_{h} is the time of action and reaction of a driver, which varies from 0.8 to 2 s, without any adverse conditions (Shakouri et al. 2015). The security distance d_{ref} (Eq. (4)) was computed with:

$$\ell = 4\,\text{m};\quad d_{\text{s}} = 1\,\text{m};\quad T_{\text{h}} = 2\,\text{s}.$$

The computing of v_{ref} (Eq. (5)) was tuned with $K_{\text{p}} = 0.22$. The tuning process K_{p} was empirically done considering several practical experiments. Higher values of K_{p} increased the oscillations of the cruise speed v_{cruise}, and lower values presented a slower control performance.

It was desired a closed-loop time constant two times faster than τ, resulting in $\lambda = 2.5$ s. The higher the λ, the slower the system response. Moreover, lower values of λ resulted in system oscillations due to the integrator in the controller structure. The parameter $\theta = 0.5$ represents the time delay of the system.

With such tuning parameters, the Dahlin controller ($C(z)$) has been designed as in Eq. (8). Remark the presence of an integrative action within the controller, which assists the reduction of steady-state errors.

$$C(z) = \frac{2.132z(z-0.903)}{(z-1)(z+0.1813)} \tag{8}$$

Both controllers of the outer loop and Dahlin $C(z)$ were implemented using Mbed C++ compiler, targeting the hardware LPC 1768. Two experiments were proposed to analyze the performance of the controllers, using the control structure shown in Fig. 1. As the experiments were established with the dynamometer, the leading vehicle speed v_{lead} was emulated via software.

Experiment 1 has three main objectives: check the performance of the control system in CC Mode, its dynamics while changing into ACC mode and its sensitivity to disturbances. The procedure of Experiment 1 is shown in Fig. 7. During the start of Experiment 1, the user-defined speed is 15 m/s and it is assumed that at 10 m of distance there is a leading vehicle at constant 15 m/s. The controlled vehicle starts with 10 m/s and set in the third gear. At 30 s of the experiment, there is an Event A. This event is described as the leading vehicle slowly breaks until it reaches the speed of 10 m/s. At 70 s, there is an Event B, in which the load torque of the dynamometer slowly increases from 0 to 15%, representing the inclusion of disturbances.

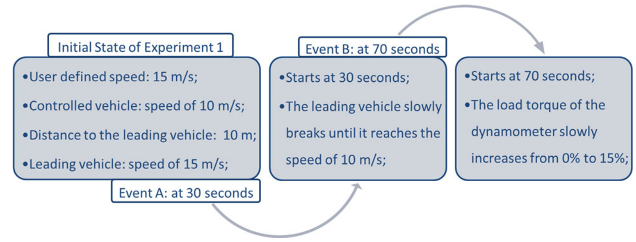

Fig. 7 Schematic of Experiment 1

The practical results of Experiment 1 are shown in Fig. 8, the relevant speeds of the experiment, and Fig. 9, the estimated distance to the leading vehicle, the calculated distance d_{ref} and the control input (accelerator pedal). At the start of the experiment, the outer loop controller goes to CC mode and the actuator takes time to effectively operate the system due to the dynamics of the integrative action in the controller.

After 30 s, Event A started. However, the controller remained in CC mode for few seconds considering that there was a considerable distance to the leading vehicle. Later on, the distance between the vehicles changed the outer loop controller into ACC mode. Therefore, the cruise speed started to change and the Dahlin controller began to track the changing set-point, to maintain a safe distance to the leading vehicle.

At 70 s of experiment (Event B), the load torque of the dynamometer started to increase and the control input changed to maintain the vehicle speed close to the cruise speed. Even with the disturbances, the vehicle speed has barely changed. Most importantly, the ACC system maintained a considerable distance to the leading vehicle.

Experiment 2 has the objective to check the performance of the ACC System with all possible scenarios regarding the leading vehicle. This experiment requires a leading vehicle with lower, equal and higher speed in relation to v_{user}. The procedure of Experiment 2 is shown in Fig. 10. During the start of Experiment 2, the user-defined speed is 15 m/s and it is assumed that at 10 m of distance there is an leading vehicle at constant 10 m/s. The controlled vehicle starts with 10 m/s and set in the third gear. At 40 s of the experiment, there is an Event A. This event is described as the leading vehicle slowly accelerates until it reaches the speed of 15 m/s. At

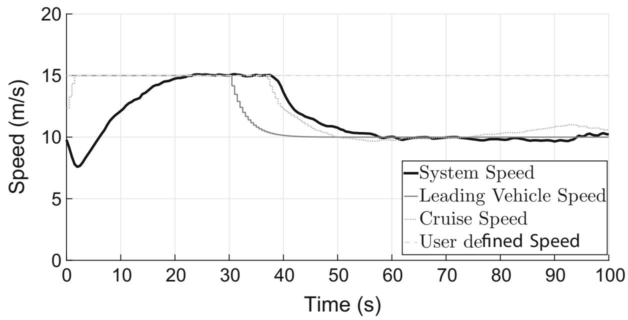

Fig. 8 Speeds for Experiment 1

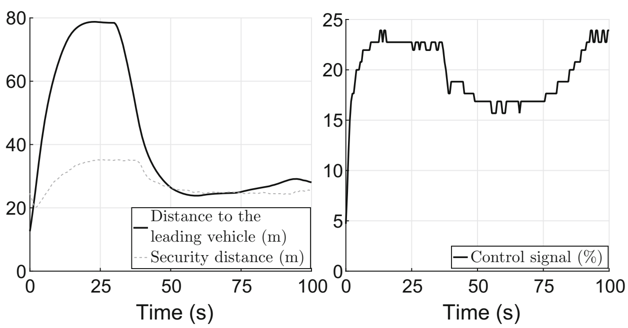

Fig. 9 Distances and Control input for Experiment 1

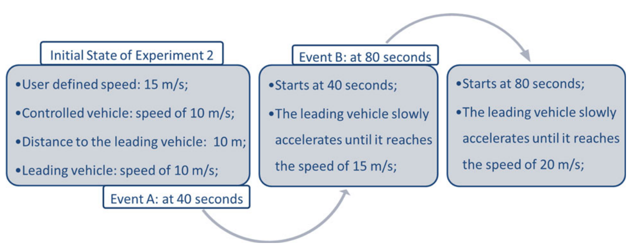

Fig. 10 Schematic of Experiment 2

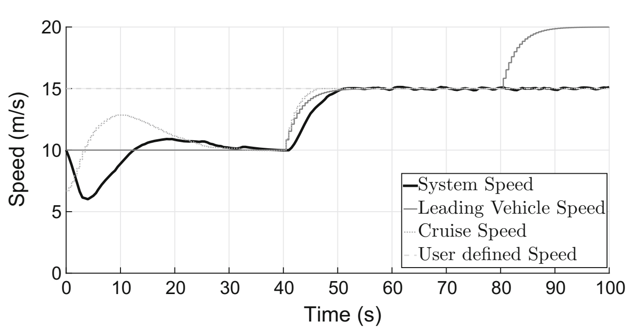

Fig. 11 Speeds for Experiment 2

80 s, there is an Event B, in which the leading vehicle slowly accelerates until it reaches the speed of 20 m/s.

The practical results of Experiment 2 are shown in Fig. 11, the relevant speeds of the experiment, and Fig. 12, the estimated distance to the leading vehicle, the calculated distance d_{ref} and the control input (accelerator pedal).

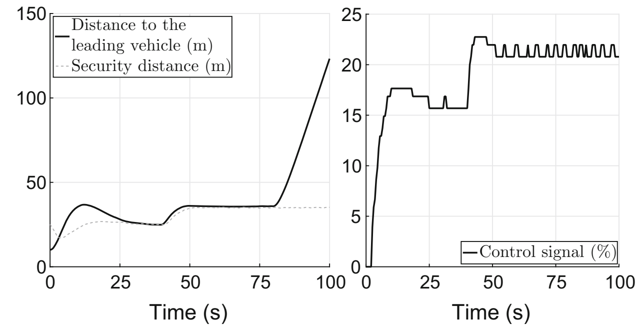

Fig. 12 Distances and Control input for Experiment 2

The ACC system accelerates to keep the distance to the leading vehicle as close as possible to the security distance d_{ref}. Due to the null initial condition of the integrator, the initial performance is somehow poor. The result would be better if the initial condition was set around the initial speed error.

After 40 s, Event A of Experiment 2 started. The leading vehicle accelerates to 15 m/s and, correspondingly, the outer loop controller changes into CC mode. During this event, the integrator has already been in a steady value. The controller reaches the cruise speed after the leading vehicle speed stops accelerating, presenting a good performance.

At 80 s of experiment (Event B), the leading vehicle accelerates even more to 20 m/s; however, the cruise speed remained at 15 m/s, as well as the vehicle speed. This occurred for two reasons: the maximum speed for the outer loop controller is $v_{user} = 15$ m/s and the ACC System is not a vehicle following system, but a speed tracking system while maintaining a safe distance. Therefore, the controlled vehicle did not follow the leading vehicle after Event B, increasing the distance between the two vehicles.

5 Conclusions

A practical ACC application in a real vehicle was presented. The control algorithm was embedded in a customized ECU.

From Experiments 1 and 2, the ACC system displayed satisfying rejection of disturbances and the elimination of steady-state errors. From an automotive perspective, the control system accomplished the design requirements, keeping a safe distance to the leading vehicle and tracking a changing cruise speed.

The designed ACC module allows the availability of several control theories implementations due to the ease of programming and communication with the vehicle. In the future, there is also the opportunity to evaluate the ACC system with ARS300 radar in highway applications.

References

de Lima, D. A., & Pereira, G. A. S. (2013). Navigation of an autonomous car using vector fields and the dynamic window approach. *Journal of Control, Automation and Electrical Systems*, *24*, 106–116.

Ferreira, V. P., Achy, A. R. A., Pepe, I. M., & Torres, E. A. (2014). A new electronic ethanol injection management system for diesel engines. *Journal of Control, Automation and Electrical Systems*, *25*(5), 566–575.

Filho, C. M., Terra, M. H., & Wolf, D. F. (2017). Safe optimization of highway traffic with robust model predictive control-based cooperative adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, *18*(11), 3193–3203. https://doi.org/10.1109/TITS.2017.2679098.

Ganji, Kouzani, & Khoo and Shams-ZahraeiGanji 2014]slidingacc Ganji, B., Kouzani, A. Z., Khoo, S. Y., & Shams-Zahraei, M., (2014). Adaptive cruise control of a hev using sliding mode control. *Expert Systems with Applications*, *41*(2), 607–615. https://doi.org/10.1016/j.eswa.2013.07.085.

Khayyam, Nahavandi, & DavisKhayyam 2012]energyacc Khayyam, H., Nahavandi, S., & Davis, S., (2012). Adaptive cruise control look-ahead system for energy management of vehicles. *Expert Systems with Applications*, *39*(3), 3874–3885. https://doi.org/10.1016/j.eswa.2011.08.169.

Kuyumcu, A., & Şengör, N. S. (2016). Effect of neural controller on adaptive cruise control. In A. E. Villa, P. Masulli, & A. J. Pons Rivero (Eds.), *Artificial neural networks and machine learning–ICANN 2016* (pp. 515–522). Cham: Springer International Publishing.

Li, S. E., Guo, Q., Xu, S., Duan, J., Li, S., Li, C., et al. (2017). Performance enhanced predictive control for adaptive cruise control system considering road elevation information. *IEEE Transactions on Intelligent Vehicles*, *2*(3), 150–160. https://doi.org/10.1109/TIV.2017.2736246.

Ljung, L. (ed.) (1999). *System identification: Theory for the user*, 2nd Edn. Prentice Hall PTR, Upper Saddle River, NJ.

Luo, L., Chen, J., & Zhang, F. (2016). Integrated adaptive cruise control design considering the optimization of switching between throttle and brake. In *2016 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1162–1167). https://doi.org/10.1109/IVS.2016.7535536.

Magdici, S., & Althoff, M. (2017). Adaptive cruise control with safety guarantees for autonomous vehicles (20th IFAC World Congress). *IFAC-PapersOnLine*, *50*(1), 5774–5781. https://doi.org/10.1016/j.ifacol.2017.08.418.

MarkVollrath, Schleicher, S., & Gelau, C. (2011). The influence of cruise control and adaptive cruise control on driving behaviour—A driving simulator study. *Accident Analysis & Prevention*, *43*(3), 1134–1139. https://doi.org/10.1016/j.aap.2010.12.023.

Pereira, B. C. F. (2017). Evolução de uma unidade de gerenciamento eletrônico de um motor vw 2.0 l e desenvolvimento de controle de cruzeiro: Projeto otto iv. Master's thesis, Universidade de São Paulo.

Pereira, B. S. (2015). Controle da mistura ar/combustível em um motor a combustão interna: sistema em malha fechada. Master's Thesis, Universidade de São Paulo.

Seborg, D., Edgar, T., & Mellichamp, D. (2004). *Process dynamics and control: Wiley Series in Chemical Engineering*. New York: Wiley.

Shakouri, P., Czeczot, J., & Ordys, A. (2015). Simulation validation of three nonlinear model-based controllers in the adaptive cruise control system. *Journal of Intelligent and Robotic Systems*, *80*(2), 207–229. https://doi.org/10.1007/s10846-014-0128-4.