Neurocomputing 71 (2008) 2727–2741

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Autonomous intelligent cruise control using a novel multiple-controller framework incorporating fuzzy-logic-based switching and tuning

Rudwan Abdullah [a,*], Amir Hussain [a,1], Kevin Warwick [b], Ali Zayed [c]

[a] Department of Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA, UK
[b] Department of Cybernetics, University of Reading, Reading, RG6 6AY, UK
[c] Seventh April University, Libya

ARTICLE INFO

Available online 16 May 2008

Keywords:
Intelligent adaptive control
Autonomous cruise control
Neural networks
PID control
Zero–pole placement control
Fuzzy switching
Fuzzy tuning
ETC system

ABSTRACT

This paper presents a novel intelligent multiple-controller framework incorporating a fuzzy-logic-based switching and tuning supervisor along with a generalised learning model (GLM) for an autonomous cruise control application. The proposed methodology combines the benefits of a conventional proportional-integral-derivative (PID) controller, and a PID structure-based (simultaneous) zero and pole placement controller. The switching decision between the two nonlinear fixed structure controllers is made on the basis of the required performance measure using a fuzzy-logic-based supervisor, operating at the highest level of the system. The supervisor is also employed to adaptively tune the parameters of the multiple controllers in order to achieve the desired closed-loop system performance. The intelligent multiple-controller framework is applied to the autonomous cruise control problem in order to maintain a desired vehicle speed by controlling the throttle plate angle in an electronic throttle control (ETC) system. Sample simulation results using a validated nonlinear vehicle model are used to demonstrate the effectiveness of the multiple-controller with respect to adaptively tracking the desired vehicle speed changes and achieving the desired speed of response, whilst penalising excessive control action.

1. Introduction

An intelligent controller, based on an expert control concept, as a decision-making element in a feedback control loop that requires as much the same decision-making ability as is needed in other expert systems: however, there are significant differences. One crucial requirement is the need to provide control signals to the process in real-time. The second requirement is that the intelligent controller should not need human interaction to complete its functions. The third is that the intelligent controller must be interfaced directly to the process and be equipped with the means for applying control to the process [5].

An intelligent controller framework based on this type of a controller will need to combine several different control algorithms as well as to tune the parameters of each algorithm according to the desired user specifications. It should also automatically manage the selection between those control algorithms to maintain the control objectives at or near their optimal values for specific process conditions. In emergency situations, where major elements in a system break down, an intelligent controller may manage the reconfiguration of the control algorithm or switch to another more appropriate or robust control algorithm.

Today's automobile effectively encompasses the spirit of mechatronic systems with its abundant applications of electronics, sensors, actuators, and microprocessor-based control systems to provide improved performance, fuel economy, emission levels comfort, and safety [3,33]. For almost 2 decades, autonomous systems have been a topic of intense research. Since the mid-1980s, several research programs have been initiated all over the world, including Advances in Vehicle Control and Safety (AVCS) in Asia, Intelligent Vehicle Highway Systems (IVHS), and Partners for Advanced Transit and Highways (PATH) in the United States. Since 2004, the US Defense Advanced Research Project Agency (DARPA) has started to organise the DARPA Grant Challenge to test automatic-vehicle technology [22]. In Europe, the DRIVE and PROMETHEUS projects have aimed at increasing the safety and efficiency in normal traffic and at reducing the adverse environmental effects of the motor vehicle [8]. In France, projects such as Praxitele and "La rout automatiée" focus on driving in urban environments, as do the EU's Cybercars and CyberCars-2 projects. Another European project, Chauffeur, focuses on truck platoon driving [22]. Thus, many research

* Corresponding authors. Tel.: +44 1786 467433; fax: +44 1786 464551.
E-mail addresses: raa@cs.stir.ac.uk (R. Abdullah), a.hussain@cs.stir.ac.uk (A. Hussain).
[1] Tel.: +44 1786 467437; fax: +44 1786 464551.

0925-2312/$ - see front matter Crown Copyright © 2008 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.05.016

groups are focusing on the development of functionalities for autonomous road vehicles that are able to interact with other vehicles safely and cooperatively [8,6,39].

An important component of adaptive cruise control (ACC) is to design control systems for controlling the throttle and brake so that the vehicle can follow the speed response of the leading vehicle and at the same time keep a safe inter-vehicle spacing under the constraint of comfortable driving [13,14]. Though there are a lot of possible techniques with which to perform ACC; conventional methods based on analytical control generate good results but exhibit high design and computational costs since the application object, a car, is a nonlinear element and a complete mathematical representation is impossible. As a result, other means of reaching human-like speed control have been recently developed, for example, through the application of artificial intelligence techniques [21].

The control of a dynamical system in the presence of large uncertainties is of great interest at the present time. Such problems arise when there are large parameter variations due to failures in the system, or due to the presence of large external disturbances [23]. The application of linear control theory to these problems relies on the key assumption of a small range of operation in order for the (local) linear model assumption to be valid. When the required operating range is large, a linear controller may not be adequate. For this reason, it seems appropriate to use nonlinear extension of generalised minimum variance-based control in order to control complex plants with nonlinear models and with plant/model mismatches. A possible way to achieve this is by incorporating the inherent nonlinearity of the process into the control design process using a *learning model* [37,40,41].

The main contribution of this paper is in the development of a new intelligent multiple-controller methodology, which is applied to an autonomous cruise control problem in order to maintain a desired vehicle speed by controlling the throttle plate angle in the car electronic throttle control (ETC) system. The proposed framework employs a neural network-based generalised learning model (GLM) to minimise the effect of both the nonlinear dynamic behaviour of the ETC system and the process disturbances. The fuzzy-logic-based switching and tuning supervisor automatically selects between a proportional-integral-derivative (PID) controller or a PID structure-based (simultaneous) pole and zero placement controller. Depending on the identified process phase and system behaviour, the fuzzy-logic-based supervisor activates the appropriate control mode. In addition, the supervisor also tunes the parameters of the multiple-controller in an on-line fashion, including the poles and zeros of the (simultaneous) pole–zero placement controller as well as the PID gains. All controllers are designed to operate using the *same* adaptive procedure and a selection between the various controller options is made on the basis of the required performance measure. The supervisor can use any available data from the control system to characterise the system current behaviour so that it knows which controller to activate, the required parameters to tune, and the tuning value for each parameter required to achieve the desired specification.

This paper is organised as follows: an overview of the control law and the GLM framework for the identification of nonlinear plants is presented in Section 2. The fuzzy-logic-based switching and tuning supervisor is illustrated in Section 3, along with some real-time implementation and stability considerations. Section 4 presents the longitudinal vehicle model and the ETC system employed for the ACC control application. Section 5 presents simulation results to assess the effectiveness of the proposed intelligent multiple controller in tracking the desired vehicle speed. Finally, some concluding remarks are given in Section 6. The appendix presents the closed-loop stability analysis of the proposed multiple-controller framework.

2. Description of the control law

Consider the following controlled auto-regressive moving average (CARMA) representation for a nonlinear plant model [37,38,40,41]

$$A(z^{-1})y(t+k) = B(z^{-1})u(t) + f_{0,t}(Y,U) + \xi(t+k), \tag{1}$$

where $y(t)$ is the measured output, $u(t)$ is the control input and $\xi(t)$ is an uncorrelated sequence of random variables with zero mean at the sampling instant $t = 1, 2, \ldots$, and k is the time delay of the process in the integer-sample interval. The term $f_{0,t}(Y,U)$ in Eq. (1) above, is potentially a nonlinear function which accounts for any unknown time-delays, uncertainty and nonlinearity in the complex plant model, and can be conveniently represented by a radial basis function neural network (RBF NN) [37] multi-layered perceptron [38,40].

The overall plant model represented by Eq. (1), is termed the GLM [37], and can be seen as the combination of a linear sub-model and a nonlinear or (learning) sub-model as shown in Fig. 1.

Also, in Eq. (1) above, $y(t) \in Y$, and $u(t) \in U$; $\{Y \in R^{n_a}; U \in R^{n_b}\}$ and $A(z^{-1})$ and $B(z^{-1})$ are polynomials with orders n_a and n_b, respectively, which can be expressed in terms of the backwards shift operator, z^{-1} as

$$A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}. \tag{2a}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b}, \quad b_0 \neq 0. \tag{2b}$$

The following control law for a multiple-controller framework has been derived in [37,38] for the above nonlinear plant model

$$u(t) = \frac{[v\tilde{H}[H(1)]^{-1}F(1)w(t) - vFy(t) + \Delta v H'_N f_{0,t}(\ldots)]}{\Delta q'}, \tag{3}$$

where $w(t)$ is the system set-point, $f_{0,t}(...)$ is a nonlinear function representing the nonlinear dynamics of the plan, v is a user-defined gain, Δ is the integral action required for the PID design, $\tilde{H}$ is a user-defined polynomial which can be used to introduce arbitrary closed-loop zeros for an explicit pole–zero placement controller, $H(1)$ is the value of $\tilde{H}$ at the system output steady state, F is a polynomial derived from the linear parameters of the controlled plant and includes the desired closed-loop poles, $F(1)$ is the value of F at the steady state, and H'_N is a user-defined polynomial. To eliminate the effect of the nonlinear dynamics in the steady state, H'_N can be set to $H'_N = -[B(1)v]^{-1}q'(1)$ [37,40]. The parameter q' is a transfer function used to bring the closed-loop system parameters in the stability unit disc, and is a polynomial in z^{-1} having the following form: $q'(z^{-1}) = 1 + q'_1 z^{-1} + q'_2 z^{-2} + \cdots + q'_{n_{q'}} z^{-n_{q'}}$, where $n_{q'}$ is the degree of the polynomial q'.

It can be seen from the control law in Eq. (3) above that the controller denominator is split into two parts, namely: an integrator action part (Δ) required for PID design; and an arbitrary compensator (q') that may be used for simultaneous pole and zero placement designs. Moreover, the polynomials H'_N and q' can be used to switch on and off the two controlling modes of the multiple controller, both of which are presented next.

2.1. Multiple-controller mode 1: PID controller

In this mode, the multiple controller operates as a conventional self-tuning PID controller, which can be expressed in the most commonly used velocity form [34] as

$$\Delta u(t) = K_I w(t) - [K_P + K_I + K_D]y(t) - [-K_P - 2K_D]y(t-1) - K_D y(t-2). \tag{4}$$

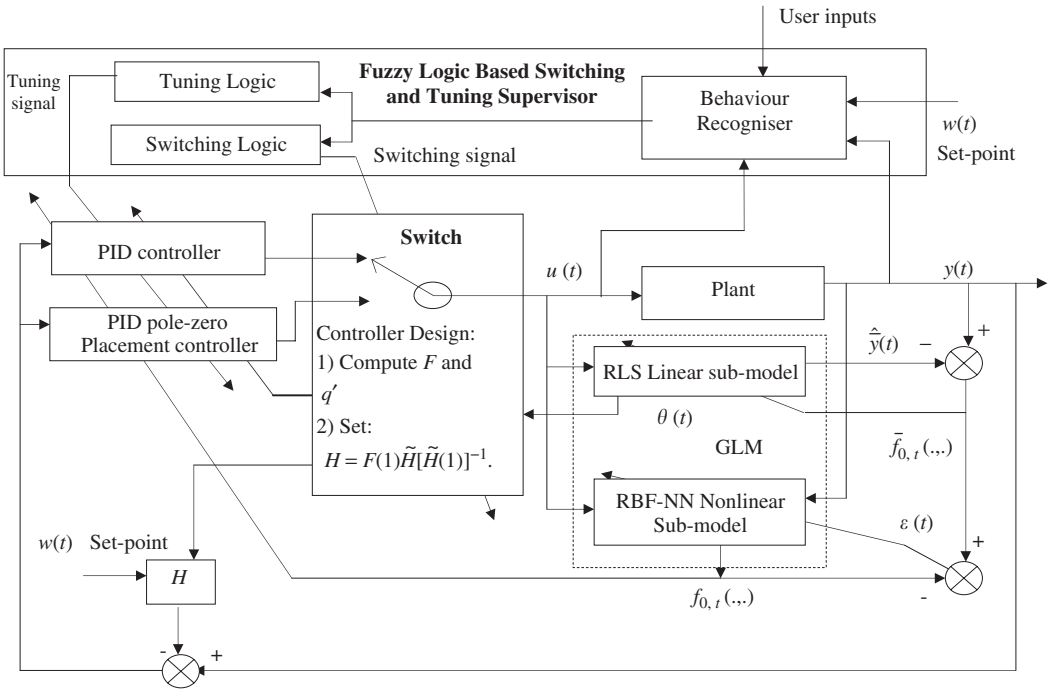


Fig. 1. Nonlinear multiple controller incorporating the GLM and fuzzy supervisory switching and tuning.

To obtain a self-tuning PID controller, the degree of $F(z^{-1})$ is set to 2 so that

$$F(z^{-1}) = f_0 + f_1 z^{-1} + f_2 z^{-2} \tag{5}$$

and both the pole-placement polynomial q' and zero placement polynomial $\tilde{H}$ are switched off by setting

$$\left.\begin{aligned} q'(z^{-1}) &= 1, (\text{i.e. } q'_1 = q'_2 = \cdots = q'_{n_{q'}} = 0), \\ \tilde{H} &= 1, (\text{i.e. } \tilde{h}_1 = \tilde{h}_2 = \cdots = \tilde{h}_{n_{\tilde{h}}} = 0). \end{aligned}\right\}. \tag{6}$$

Therefore, the self-tuning PID controller is structured as follows:

$$\Delta u(t) = [vF(1)w(t) - v(f_0 + f_1 z^{-1} + f_2 z^{-2})y(t) + \Delta v H'_N f_{0,t}(.,.)], \tag{7}$$

$$K_p = -v[f_1 + 2vf_2], \tag{8}$$

$$K_I = v[f_0 + f_1 + f_2], \tag{9}$$

$$K_D = vf_2. \tag{10}$$

It can be seen from the above Eqs. (7)–(10) that the PID control parameters K_p, K_I and K_D depend on the polynomial matrix $F(z^{-1})$ and the gain v [34].

The main disadvantage of PID self-tuning-based minimum variance control designs, in general, is that the tuning parameters must be selected using a trial and error procedure. That said, the use of general heuristics can provide reasonable closed-loop performance. Alternatively, the tuning parameters can also be automatically and implicitly set on-line by specifying the desired closed-loop poles [17,32].

2.2. Multiple-controller mode 2: simultaneous zero and pole placement controller based on a PID structure

To derive a simultaneous zero and pole-placement controller with a PID structure, the transfer function of the closed-loop system must be derived. Substituting for $u(t)$ given by Eq. (3) into the process model described by Eq. (1), the closed-loop system is obtained as follows:

$$(\tilde{A}q' + z^{-1}\tilde{B}F)y(t) = z^{-1}Bv\tilde{H}[H(1)]^{-1}F(1)w(t) + \Delta(Z^{-1}BvH'_N + q')f_{0,t} + \Delta q', \tag{11}$$

where $\tilde{A} = \Delta A$ and $\tilde{B} = vB$.

The following Diophantine equation is used to place poles of the system in the required position [37]

$$(q'\tilde{A} + z^{-1}F\tilde{B}) = T, \tag{12}$$

where T represents the desired closed-loop poles and q' is the controller polynomial. For Eq. (12) to have a unique solution, the order of the regulator polynomials and the number of the desired closed-loop poles can be set as [4,24,32]

$$\left.\begin{aligned} n_{f'} &= n_{\tilde{a}} - 1 = n_a \\ n_{q'} &= n_{\tilde{b}} + k - 1 \\ n_t &\leqslant n_{\tilde{a}} + n_{\tilde{b}} + k - 1 \end{aligned}\right\}, \tag{13}$$

where $n_{\tilde{a}}$, $n_{\tilde{b}}$ and $n_{q'}$ are the orders of the polynomials $\tilde{A}$, $\tilde{B}$ and q', respectively, and n_t denotes the number of desired closed-loop poles. Also, $n_{\tilde{b}} = n_b$ and $n_{\tilde{a}} = n_a + 1$.

Combining Eqs. (11) and (12) and setting

$$H'_N = -[B(1)v]^{-1}q'(1), \tag{14}$$

$$T(z^{-1}) = 1 + t_1 z^{-1} + t_2 z^{-2} + \cdots + t_{n_t} z^{-n_t}, \tag{15}$$

$$\tilde{H}(z^{-1}) = 1 + h_1 z^{-1} + h_2 z^{-2} + \cdots + h_{\tilde{h}} z^{-\tilde{h}}, \tag{16}$$

where $n_{\tilde{h}}$ and n_t represent orders of the polynomials $\tilde{H}(z^{-1})$ and $T(z^{-1})$, respectively. An arbitrary desired zeros polynomial can be used to reduce excessive control action, which can result from set-point changes when pole placement is used.

The final closed-loop function for the zero and pole-placement controller then becomes

$$Ty(t) = z^{-1}Bv[\tilde{H}(1)]^{-1}\tilde{H}F(1)w(t) + \Delta(z^{-1}Bvq'(1) \times[-(B(1)v)^{-1}] + q')f_{0,t} + \Delta q'\xi(t). \tag{17}$$

Note that, in practice, the order of $T(z^{-1})$ and $\tilde{H}(z^{-1})$ are most of the time selected to equal 1 or 2 [34]. It can be seen from Eq. (17) that the closed-loop poles and zero are placed at their desired positions which are pre-specified by using the polynomials $T(z^{-1})$ and $\tilde{H}(z^{-1})$.

In the next section, the identification of complex nonlinear plants using the GLM framework is briefly discussed.

2.3. GLM for the identification of nonlinear plants

A wide range of nonlinear dynamic plants can be described by a discrete time equation [41] $y(t+1)=f(Y,U)$ where $f(Y,U)\rightarrow R^n$; $\{Y\in R^{n_y}; U\in R^{n_u}; n=n_y+n_u\}$ is a complicated nonlinear function, and $y(t)\in Y$ and $u(t)\in U$ are the plant output and input signals, respectively, at discrete times $t\in 1, 2,\ldots$. To control such a nonlinear plant, the generalised parametric time-varying plant model described in Eq. (1) is considered [37,38,40,41]. In addition, it is assumed that the parameters associated with $A(z^{-1})$ and $B(z^{-1})$ (in Eq. (1)) are either time invariant or slowly time varying. On the other hand, $f(Y,U)\rightarrow R^n$ is potentially a time-varying nonlinear function. Therefore, the equivalent plant model is a combination of a linear time-varying sub-model plus a nonlinear time-varying sub-model (or an error agent), which have been collectively termed the GLM [37]. The linear sub-model is used to approximate the dominant linear dynamics of the nonlinear dynamic plant around its operating point. On the other hand, the 'learning' error agent is used to learn the errors from the linear sub-model that are due to nonlinearities, uncertainties, disturbances and model mismatch in the controlled plant.

The GLM model is illustrated in Fig. 1, where a recursive least-squares algorithm is initially used to estimate the parameters A and B (Eq. (1)) of the linear sub-model, and an RBF NN-based learning model is subsequently used to approximate the nonlinear part $f_{0,t}(.,.)$.

It can be seen from Eq. (1) that the measured output $y(t)$ can be obtained as follows [40]:

$$y(t+1)=\varphi^T(t)\theta+f_{0,t}(y,u), \tag{18}$$

$$\left.\begin{aligned}\theta&=[-a_1,\ldots,-a_{n_a},b_0,\ldots,b_{n_b}]\\ \varphi^T&=[y(t-1),\ldots,y(t-n_a),u(t-1),\ldots,u(t-n_b)]\end{aligned}\right\}. \tag{19}$$

From Fig. 1 we can also see that $\bar{f}_{0,t}(.,.)$ can be expressed as

$$\bar{f}_{0,t}(.,.)=f_{0,t}(.,.)+\varepsilon(t). \tag{20}$$

Using the above Eq. (20), and as shown in Fig. 1, an NN can be effectively used for estimating the nonlinear error function $f_{0,t}$, with the identification error $\varepsilon(t)$ used to update the weights and thresholds of the learning neural network model. The neural network model employed in the proposed control scheme here is chosen to be the linear-in-parameters RBF NN as opposed to the computationally more expensive multi-layer perceptron (MLP) previously used in [40]. The RBF NNs can improve the system damping and dynamic transient stability more effectively than the MLP NNs [26]. In addition, the RBF requires a lower computational complexity and elapsed time to train the network on-line, compared to the MLP [26]. The RBF NNs ability to uniformly approximate smooth functions over compact sets is well documented in the literature, see for example [30].

Using the RBF NN, the nonlinear function $f_0(.,.)$ is adaptively estimated by using the following equations [37]:

$$f_{0,t}(.,.)=\sum_{j=1}^{n} w_j g_j b, \tag{21}$$

$$\delta_w=\eta g_{j,t} f_{0,t}[\bar{f}_{0,t}-f_{0,t}][1-\bar{f}_{0,t}], \tag{22}$$

$$w_j(t)=w_j(t-1)+\delta_w, \tag{23}$$

$$g_{0,t}(\ldots)=\exp\left(-\sum_{i=1}^{l}\frac{\|x_i-c_j^i\|^2}{(2\sigma_j^i)^2}\right), \tag{24}$$

where w_j represents the hidden layer weights, β is the output layer threshold, δ_w is the change in weights, η is the learning rate, x_i is the inputs, c_j^i is the centre of Gaussian basis function of the jth hidden unit, $l=n_a+n_b$, and g_j is the output of the hidden layer. The variance of the Gaussian units σ_j^i is dependent on the input dimension because the RBF inputs are scaled differently [18,27]. RBF weights can be adapted using various algorithms, as good overview of which can be found in [10].

2.4. Discussion on selection of controller type and switching issues

Due to the robustness, simplicity of structure, ease of implementation, and remarkable effectiveness in regulating a wide range of processes, the conventional self-tuning PID controller (multiple-controller mode 1) proposed in Section 2.1 should normally be the first choice to obtain satisfactory closed-loop system performance. If, however, a better closed-loop performance based on, say, a desired damping ratio, rise time, settling time overshoot, system bandwidth or the actual minimum variance performance index-is required, or if the complex system to be controlled is difficult to tune using a conventional self-tuning PID controller—due to, for example, an excessive control (input) action resulting from set-point changes, added disturbances and/or plant nonlinearity—then the PID structure-based (simultaneous) pole and zero placement controller (multiple-controller mode 2) discussed in Section 2.2 can be used as a second choice. However, this will be at the expense of a greater computational effort required for implementation of the (simultaneous) pole and zero placement controller [37,38]. Note that whilst only two control modes have been considered here, the proposed framework is flexible enough to allow additional controllers to be employed in order to further improve the closed-loop performance, if required. In the context of switching between the above two controller modes, the type of discriminating (system performance) criteria mentioned above can also be used to design an intelligent fuzzy-logic-based switching and tuning supervisor, which is presented in the next section.

3. Fuzzy-logic-based switching and tuning supervisor

The fuzzy-logic-based switching and tuning supervisor is situated at the highest level of the multiple-controller framework. Following [28], the fuzzy supervisor can use any available data from the control system to characterise the system current behaviour so that it knows which controller to choose, which parameters to tune, and the tuning value for each parameter that is required to ultimately achieve the desired specification. The main idea behind the fuzzy-logic supervisor approach here is to employ logic-based switching and tuning among the family of the candidate controllers. The need for switching stems from the fact that typically no single controller can guarantee the desired behaviour when connected with a poorly modelled process, and particularly so for the case of complex processes exhibiting significant nonlinearity, nonstationarity, uncertainty and/or multi-variable interactions [37]. Such switching schemes can provide an alternative to more traditional continuously tuned adaptive control algorithms.

Following [1,11,28], the supervisor employed in this work comprises two sub-systems: a *behaviour recogniser* and a *switching and tuning logic* (see Fig. 1). The supervisor aims to recognise when

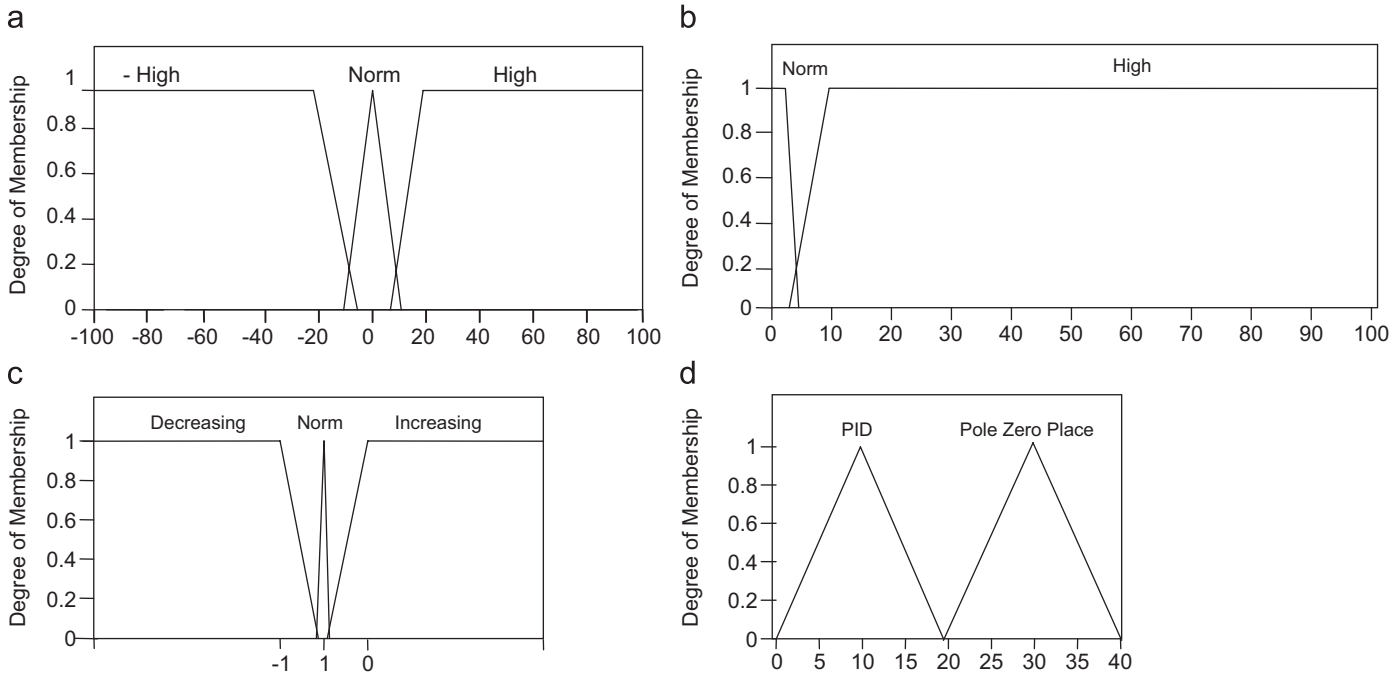


Fig. 2. (a) Membership function of the *overshooting* input parameter, (b) membership function of the *input signal variance* input parameter, (c) membership functions of reference signal changes input parameter and (d) membership function of the *controller selection* output parameter.

the system requires selection of another controller, or when a selected controller is not properly tuned, and then seeks to switch to the candidate controller and/or adjust the controller parameters to obtain improved performance. The whole supervisor is implemented using simple fuzzy-logic-based switching and tuning rules where the premises of the rules form part of the behaviour recogniser and the consequent form the switching and tuning decision. In this way, a simple fuzzy system is used to implement the entire supervisory control level.

3.1. Behaviour recogniser

The behaviour recogniser seeks to characterize the current behaviour of the plant in a way that will be useful to the switching and tuning logic sub-systems [29]. The behaviour of the system is characterized through the on-line estimation of five parameters namely: the *overshoot* of the closed-loop system output signal, the *variance* of the control input signal, rise and fall times of the output signal (used for tuning purposes), reference signal (i.e. set-point) changes, and the system steady-state error. In order to prevent the so-called *shattering* problem (Zeno behaviour) caused by wrongly designed switching rules [11], which can result in an infinite number of switchings between controller modules, the average of the current and last two values of the overshoot measurements are used as an output parameter from the behaviour recogniser. The output signal rise and fall times represent the amount of time for a signal to change state. To measure rise and fall times, the behaviour recogniser tracks 10% and 90% values of the output signal. The PID controller steady-state error is used to tune the gain v (Eq. (3)).

3.2. Switching logic

The key task of the switching logic sub-system is to generate a switching signal which determines, at each instant of time, the candidate controller module that is to be activated [1,11]. The switching logic is implemented using fuzzy-logic rules where the premises of the rules use the output of the behaviour recogniser as input parameters, and the consequents of rules form the controller selection decision (output parameter). Fig. 2a–c, respectively, shows the shape of the membership functions of the system output overshoot, control input-variance and reference signal changes, which are all input parameters for the switching decision. Fig. 2d shows the membership function of the controller selection output parameter. The complete rule set for the fuzzy supervisor switching logic is given below:

If *Overshoot* is High OR *Control input-variance* is High *Then Controller* is Pole-Zero Placement
If *Overshoot* is High OR *Control input-variance* is High *Then Controller* is Pole-Zero Placement
If *Reference signal* is Not Norm *Then Controller* is Pole-Zero Placement
If *Overshoot* is Norm OR *Control input-variance* is Norm *Then Controller* is PID

Depending on the fuzzified value of the two input parameters, the switching logic sub-system will switch either to the conventional PID controller, or PID structure-based (simultaneous) pole and zero placement controller. The middle-of-max approach is used for de-fuzzification in order to identify the selected controller.

3.3. Tuning logic

The key task of the tuning logic sub-system is to tune the parameters of the multiple-controller on-line, including poles and zeros of the (simultaneous) pole-zero placement controller in addition to the PID gains. The tuning facility aims to make the system achieve a desired speed of response and/or minimize the control input action. Using the measurements of the rise time and fall time of the output signal, which are measured by the *behaviour recogniser* sub-system, and detected changes in the reference signal, the fuzzy-logic-based supervisor will specify

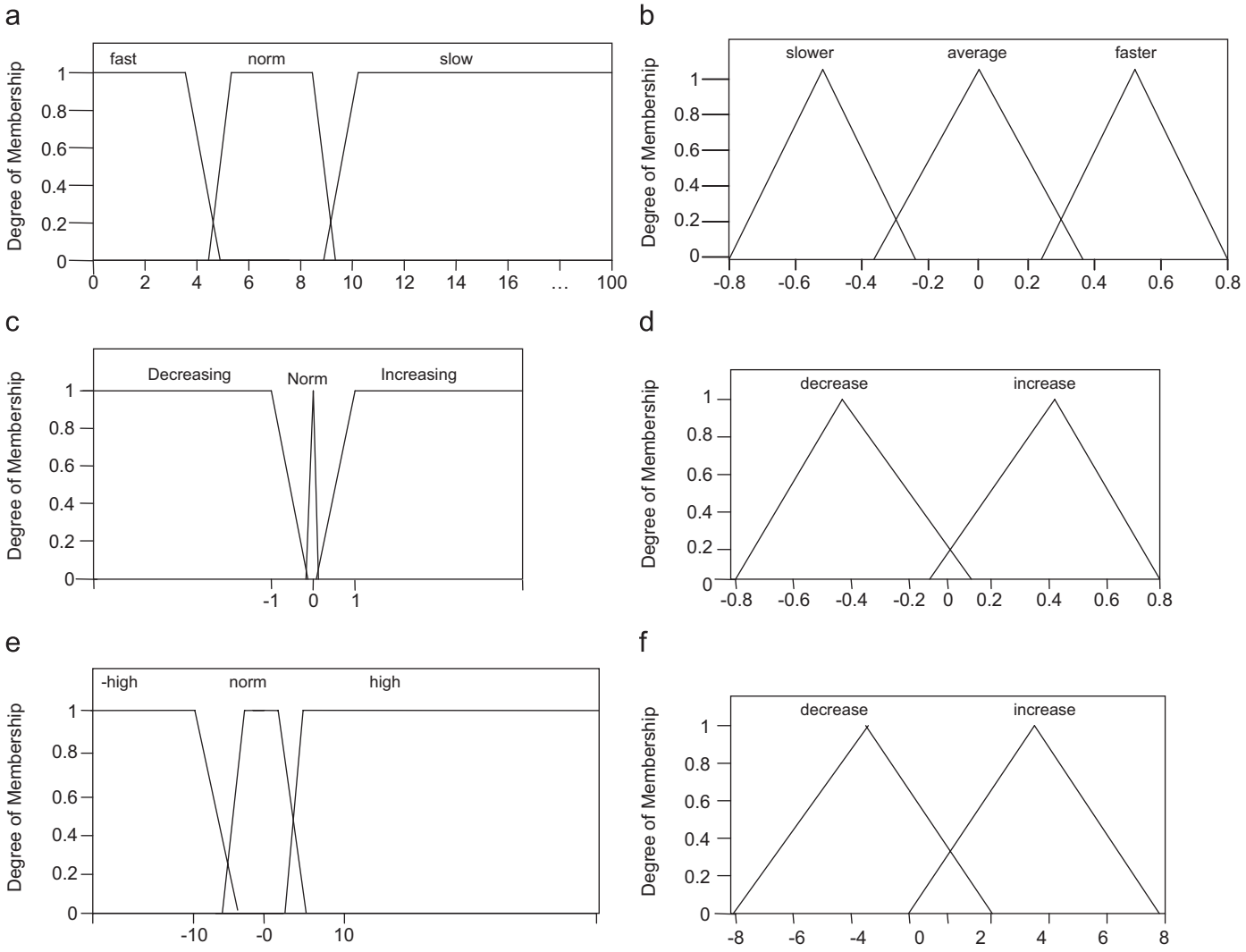


Fig. 3. (a) Membership function of the *rise time* parameter, (b) membership function for *poles tuning* output parameter, (c) membership functions of set-point changes input parameter, (d) membership functions *zeros tuning* parameter, (e) membership function of the *PID steady-state error* parameter and (f) membership function for *PID gain tuning* output parameter.

the tuning values for the active/selected controller's parameters. The complete fuzzy rule set for parameter tuning is given below:

IF Rising-time is norm *AND controller* is Pole-Zero Placement *THEN poles-tuning* is no-change
IF Rising-time is fast *AND controller* is Pole-Zero Placement *THEN poles-tuning* is slower
IF Rising-time is slow *AND controller* is Pole-Zero Placement *THEN poles-tuning* is faster
IF set-point is increasing *AND controller* is Pole-Zero Placement *THEN zeros-tuning* is decrease
IF set-point is decreasing *AND controller* is Pole-Zero Placement *THEN zeros-tuning* is increase
IF Control input-variance is high *AND controller* is PID *THEN gain* is decrease
IF steady-state-error is high *AND controller* is PID *THEN gain* is decrease
IF steady-state-error is high *AND controller* is PID *THEN gain* is increase
IF steady-state-error is norm *AND controller* is PID *THEN gain* is no-change

The fuzzy membership functions used for tuning the controller parameters are shown in Fig. 3a–f.

3.4. Real-time implementation and stability considerations

Two key real-time implementation constraints include minimising the amount of memory used and the time taken to compute the fuzzy outputs from given inputs [28]. Keeping these in view, the proposed fuzzy-logic-based switching and tuning supervisor has been designed with a minimum number of fuzzy rules along with minimum input and output parameters. The switching logic consists of a total of six rules, two input parameters and one output parameter. The tuning logic comprises nine rules, three inputs and three output parameters. From the point of view of computational complexity, a maximum of three membership functions are considered for both the inputs and outputs with no more than two overlapping membership functions. In order to improve the computation time, the triangular and trapezoidal membership functions are used, which have the advantages of simplicity and require minimum rules. To reduce the memory requirements and to enable online operation

(i.e., switching and tuning), the fuzzy supervisor is designed to compute the rule-base at each time instant rather than using a stored one. Future implementation prospects can be improved by using a state-of-the-art microprocessor or signal processing chip. An alternative would be to investigate the advantages and disadvantages of using a fuzzy processor (i.e., a processor designed specifically for implementing fuzzy controllers) as in [28].

Finally, note that a theoretical verification of the stability of the proposed multiple-controller framework is important especially for safety-critical systems such as the target ACC application considered in this work. The closed-loop stability analysis of the proposed controller framework is summarised in Appendix A.

4. Description of the application model employed

One major thrust in automotive technology is the drive-by-wire system, which replaces the direct mechanical linkages with sensors, and actuators under computer control [3]. These systems allow traditional driver-to-vehicle interfaces to be redesigned and offer the potential to supplement the driver's command. In the automotive vehicle cruise control application, the engine control unit for an internal combustion engine receives input from many engine sensors (e.g. manifold air pressure, manifold temperature, engine speed, etc.) to control the fuel and spark delivery process. For the application at hand, the primary focus is the ETC dynamics. The ETC system is a drive-by-wire system in which the direct linkage between the accelerator and the throttle are replaced with sensors and actuators under computer control [3,9]. The ETC system, similar to the driver's foot, generates a control signal proportional to the desired speed by dynamically adjusting the pedal to throttle position transfer function in response to changes in vehicle parameters [7,21]. The ETC system uses a torque motor (DC servo-motor) to regulate the throttle plate angle θ between $0<\theta\pi/2$ radians (i.e., closed to wide-open-throttle) in order to adjust the inlet airflow (see Fig. 4). The servo-motor is controlled by the applied armature voltage e_aV. The nonlinear model of the ETC can be presented as follows [3]:

$$y + a_1\frac{dy}{dt} + a_2\frac{d^2y}{dt^2} = b_1u - c_1\Delta P\cos^2 y - \theta_0, \tag{25}$$

where the input and output for the ETC system are $u = e_a$ and $y = \theta$, respectively, and θ_0 is the pre-tension angle of the spring. The process linear parameters a_1, a_2, b_1 and c_1 are estimated using the input/output relationship in Eq. (25) with the recursive least-squares algorithm (GLM linear sub-model). ΔP is the manifold pressure across the throttle plate. The term $\Delta P\cos^2 y$ is a nonlinear function which represents the spring torque and airflow torque over the throttle plate. These nonlinear dynamics and any other disturbances in ETC system will be estimated by the function $f_{0,t}(.,.)$ using the RBF NN-based learning agent (GLM nonlinear sub-model) described in Section 2.3 [37,41].

The main objective of the control problem is to adjust the throttle plate angular position θ so as to maintain the desired speed v of a longitudinal vehicle model. A longitudinal vehicle system is mainly composed of an engine, transmission, and drive train system. A simple functional description of such a system is shown in Fig. 5. Each block can be considered as a sub-system with various inputs and outputs. The output of the engine sub-system is the engine torque which is a nonlinear function of air/fuel ration, exhaust gas recirculation (EGR), cylinder total mass charge, spark advances, engine speed and drive train load, as well as the throttle angle [39].

A simplified longitudinal vehicle model can be represented using the following equations [7,12,39]:

$$\dot{v} = \frac{1}{m}[-c_v v^2 - c_p v - d_m + f_1(v, T_e)], \tag{26}$$

$$T_e = f_2(\theta), \tag{27}$$

where v is the vehicle speed, m is the vehicle mass, c_v is the coefficient of aerodynamics drag, c_p is the coefficient of friction force, and d_m is the mechanical drag. $f_1(v,T_e)$ is the ideal-tire force which is generally measured by steady-state tests [39] and it depends mainly on the vehicle speed and the engine torque T_e. The engine torque T_e itself is a nonlinear mapping from θ to T_e. Finally,

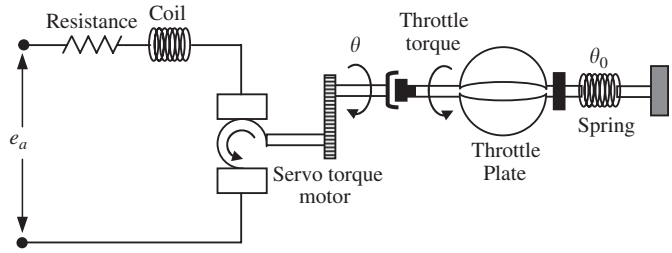


Fig. 4. Electronic throttle schematic diagram [3].

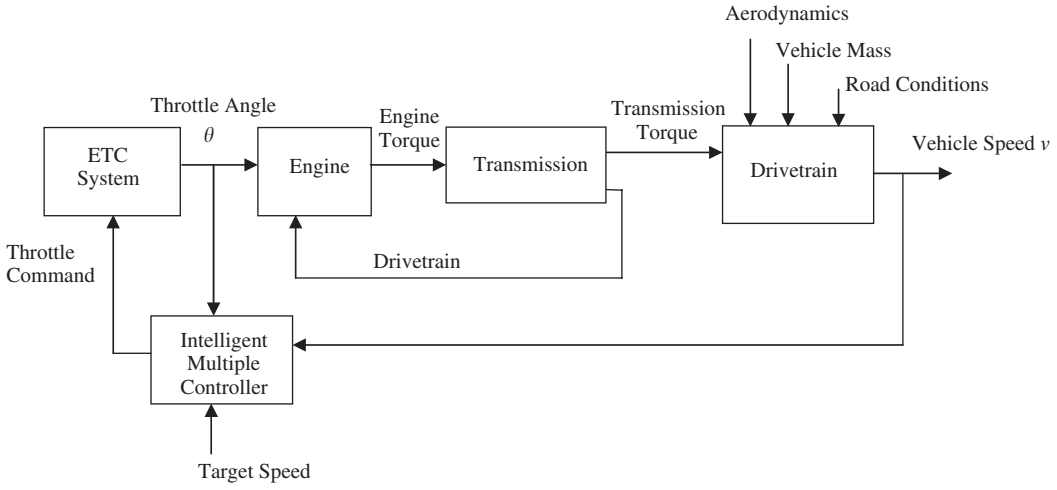


Fig. 5. Longitudinal vehicle model.

$f_2(\theta)$ is the steady-state characteristics of engine and transmission systems, and in this paper it is represented by an RBF NN with θ as its input, and T_e as the output variable. The RBF design parameters were selected on a trial and error basis.

Note that as discussed in [39], the above vehicle dynamics equations (26) and (27) have the following property: for any given desired ideal-tire force and velocity v, there exists a unique throttle angle input θ that achieves the desired ideal-tire force. Further details on the above model can be found in [39].

5. Simulation results

In order to assess the performance of the proposed scheme in controlling the ETC system for an autonomous cruise control application, 3 experiments are performed and presented in this section. All simulation examples are performed over 1000 samples to track a reference signal representing the target vehicle speed changes. The desired closed-loop pole and zero polynomials are initially selected as $T = 1-0.4z^{-1}-0.2z^{-2}$ and $h = 1+0.9z^{-1}+0.6z^{-2}$. In order to achieve the PI self-tuning case for this application, the user-defined parameters are selected as: $P = 1$ and $v = 0.83$. The nonlinear function $f_{0,t}(.,.)$ representing the nonlinear dynamics of the ETC system is obtained using Eqs. (21)–(24), which implement the nonlinear sub-model of the complex ETC system. Using a trial- and -error approach, the RBF NN was designed with eight units in the hidden layer three of which had fixed centres and the remaining five units were selected to have adaptive centres and were fine tuned using the back-propagation method (as in) [25]. The output layer weights were updated using the Delta rule.

The problem of regulating the throttle plate in the ETC system can be divided into three sub-problems: firstly, preventing overshooting in control of the throttle plate angle caused by either reference signal changes, nonlinear system dynamics and/or added disturbances; secondly, minimising any high-voltage inputs applied to the DC servo-motor in the ETC system; and finally, minimising oscillation in the normal system (steady state) operation. The controller that solves best the first and second sub-problems is the PI structure-based pole–zero placement controller (at the expense of a relatively greater computational requirement), whereas the controller that most effectively deals with the third sub-problem is the conventional PI controller. The fuzzy-logic-based switching and tuning supervisor is designed to automatically switch between these two controllers on the basis of the system behaviour detected by the behaviour recogniser sub-system. The tuning logic is used to regulate each controller's parameters in order to achieve the desired (user specified) signal rise and fall times, and to reduce the magnitude of the control input signal introduced to the throttle actuator (i.e., the DC servo-motor). In the following 2 experiments, the ETC system is controlled using each of the two individual multiple-controller modes (with no switching). In the final third experiment, the intelligent multiple-controller is deployed incorporating fuzzy-logic-based switching between the two controller modes. The aim is to justify the need for, and demonstrate the advantages of deploying the proposed intelligent multiple-controller framework incorporating a fuzzy-logic-based switching and tuning supervisor. In all experiments, the nonlinear sub-model of the GLM was set to work with the linear sub-model in order to incorporate $f_{0,t}(.,.)$ in the control law.

5.1. Experiment 1

In this experiment, the multiple-controller was set to control the ETC system using only the conventional PI controller (multiple-controller mode 1).

In the obtained results, Fig. 6a–d, respectively, shows the vehicle speed signal obtained (when the PI controller is used to maintain the target vehicle speed v), the output throttle plate angle θ (required to reach the target speed), the throttle control input signal $u(t)$ applied to the torque motor (to produce the required throttle angle θ), and the system nonlinearities and external disturbances applied to the controlled ETC system.

Fig. 6b and c shows high overshooting in both the output throttle angle signal and in the throttle control input signal, respectively, when the reference signal changes. The use of the PI only controller resulted in this overshooting as it worked to produce the required output throttle angle to track the target changes in speed. On the other hand, as can be seen from Fig. 6b, as expected theoretically, the PI controller has the advantage of maintaining relatively smooth steady performance. In addition, Fig. 6b and c also shows relatively low overshooting at sampling time 300 caused by the presence of nonlinear dynamics of the ETC system, and the externally introduced constant and random disturbances. The variance of the control input signal was measured to equal 0.0030, and the variance of the output (throttle angle) signal was 0.0305.

5.2. Experiment 2

In this experiment, the multiple-controller was set to control the ETC system using only the pole–zero placement controller (multiple-controller mode 2).

In the obtained results, Fig. 7a–d, respectively, shows the vehicle speed signal obtained, the output throttle plate angle θ, the throttle control input signal $u(t)$ and the system nonlinearities and external disturbances applied to the controlled ETC system.

In the obtained results, Fig. 7b and c shows relatively higher overshooting at sampling time 300 caused by the presence of nonlinear dynamics of the ETC system, and the externally introduced constant and random disturbances. In addition, continuous sharp oscillations can be seen during the steady state while the pole–zero placement controller worked to produce the required output throttle angle for effectively tracking the desired speed changes. However, as expected theoretically, this pole–zero placement controller exhibits very low overshooting while dealing with reference signal changes (as can be seen from Fig. 7b) as a result of effectively minimising the control input-variance (shown in Fig. 7c). The variance of the control input signal was measured to equal 0.0027 and the variance of the throttle angle signal was 0.0278.

5.3. Experiment 3

In this example, the intelligent multiple controller was set to automatically control the ETC system in order to track the defined vehicle speed reference signal. By appropriately switching between the conventional nonlinear PI controller and the nonlinear pole–zero placement controller (based on the monitored degree of overshooting and input signal variance), the fuzzy-logic-based supervisor works to prevent any overshooting and minimises the steady-state oscillations while controlling the ETC system. The capability of the fuzzy-logic-based supervisor to tune the controller parameters also allows for controlling the rise and fall times of the output signal (throttle angle) according to user-defined requirements.

In the obtained results, Fig. 8a–e, respectively, shows the vehicle speed signal obtained, the output throttle plate angle θ, the throttle control input signal $u(t)$, the selection scheme resulting from use of the fuzzy-logic-based switching supervisor,

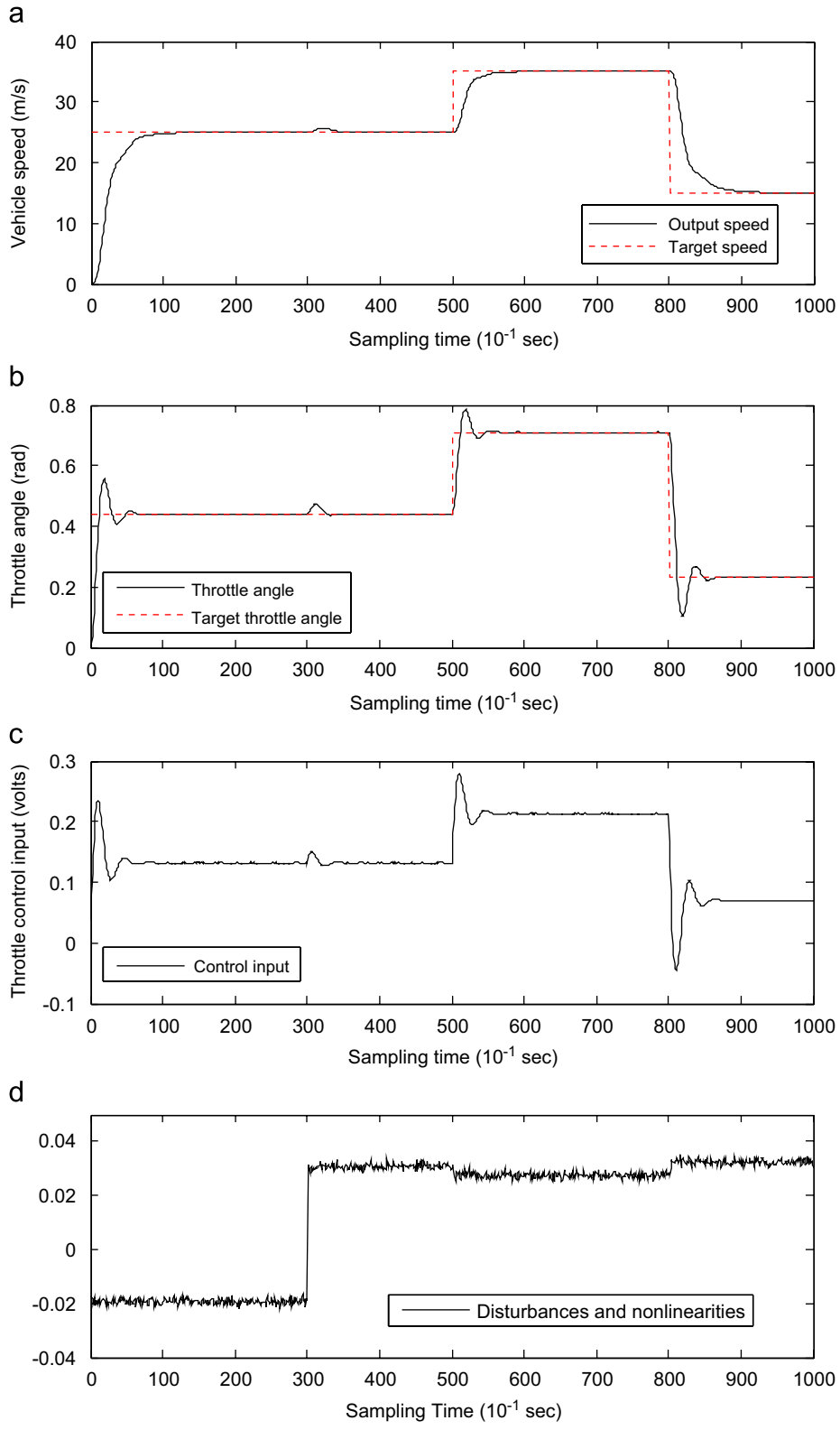


Fig. 6. (a) The speed signal v, (b) the output throttle angle θ, (c) the control input signal $u(t)$ and (d) the nonlinearities and disturbances.

and system nonlinearities and external disturbances applied to the controlled ETC system.

Compared to the results obtained from experiments 1 and 2, it can be seen from Fig. 8(a)–(d) that the fuzzy-logic-based supervisor effectively manages to prevent output signal (throttle angle) overshooting and simultaneously preserves relatively smooth steady-state input and output signals. Another point to note is that the excessive control action (resulting from reference set-point changes) is tuned most effectively when the switching supervisor activates the PI-based pole–zero placement controller.

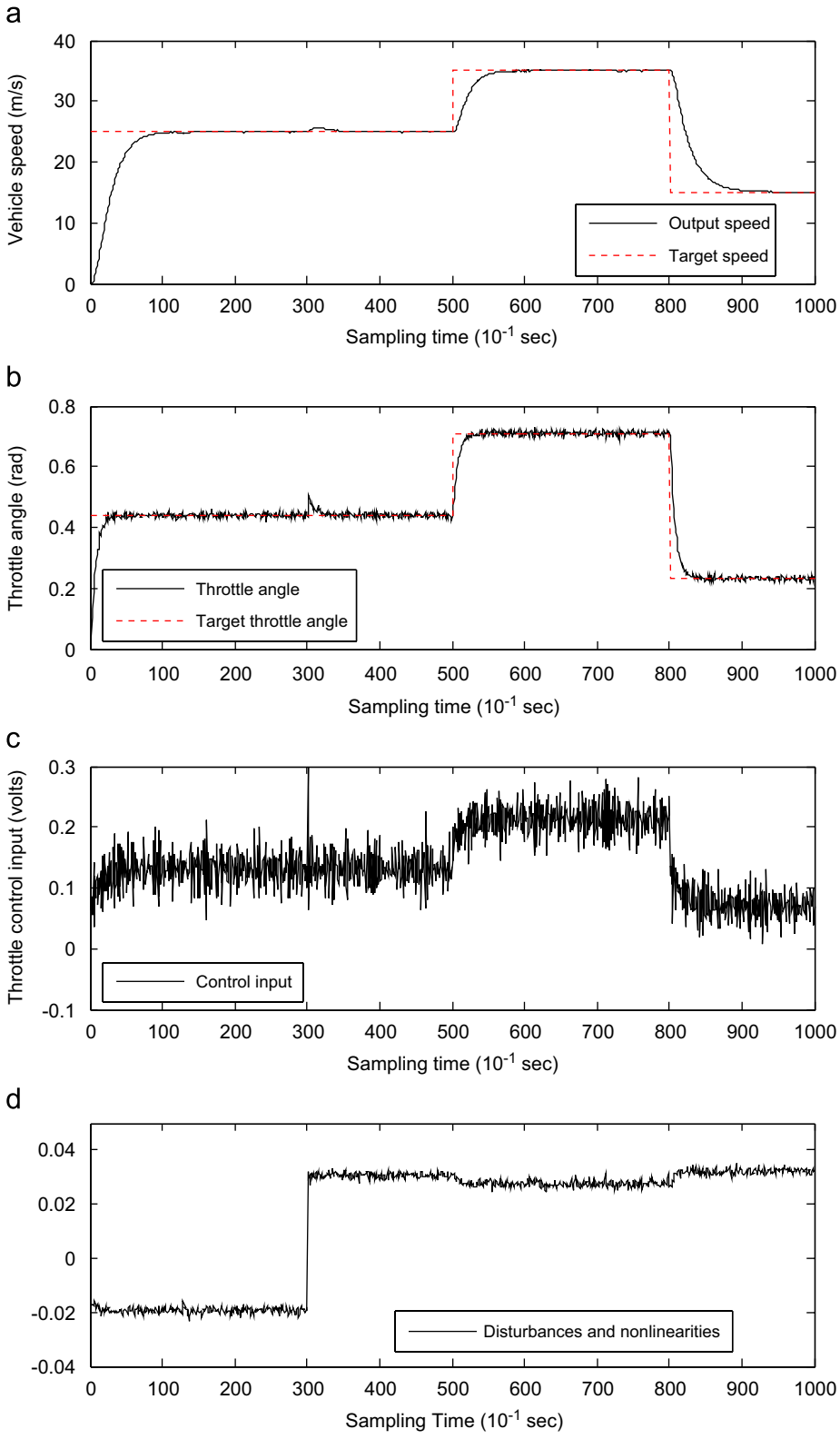


Fig. 7. (a) The speed signal v, (b) the output throttle angle θ, (c) the control input signal $u(t)$ and (d) the nonlinearities and disturbances.

As a result of effectively tuning the poles and zeros of the pole-zero placement controller (in order to maintain the desired rise and fall signal times), it can be seen in Fig. 8b, that at sampling time 500, a faster rising time (for achieving the required throttle angle) also results in a shorter settling time (for reaching the desired vehicle speed).

Table 1 summarises the comparative results of the 3 experiments in terms of computed rise time (when changing speed from 25 m/s to the target 35 m/s), and variance measures of the control-input signal and the output-signal variance, respectively. It can be seen from Table 1 that the intelligent multiple-controller incorporating fuzzy-logic-based tuning and switching results in

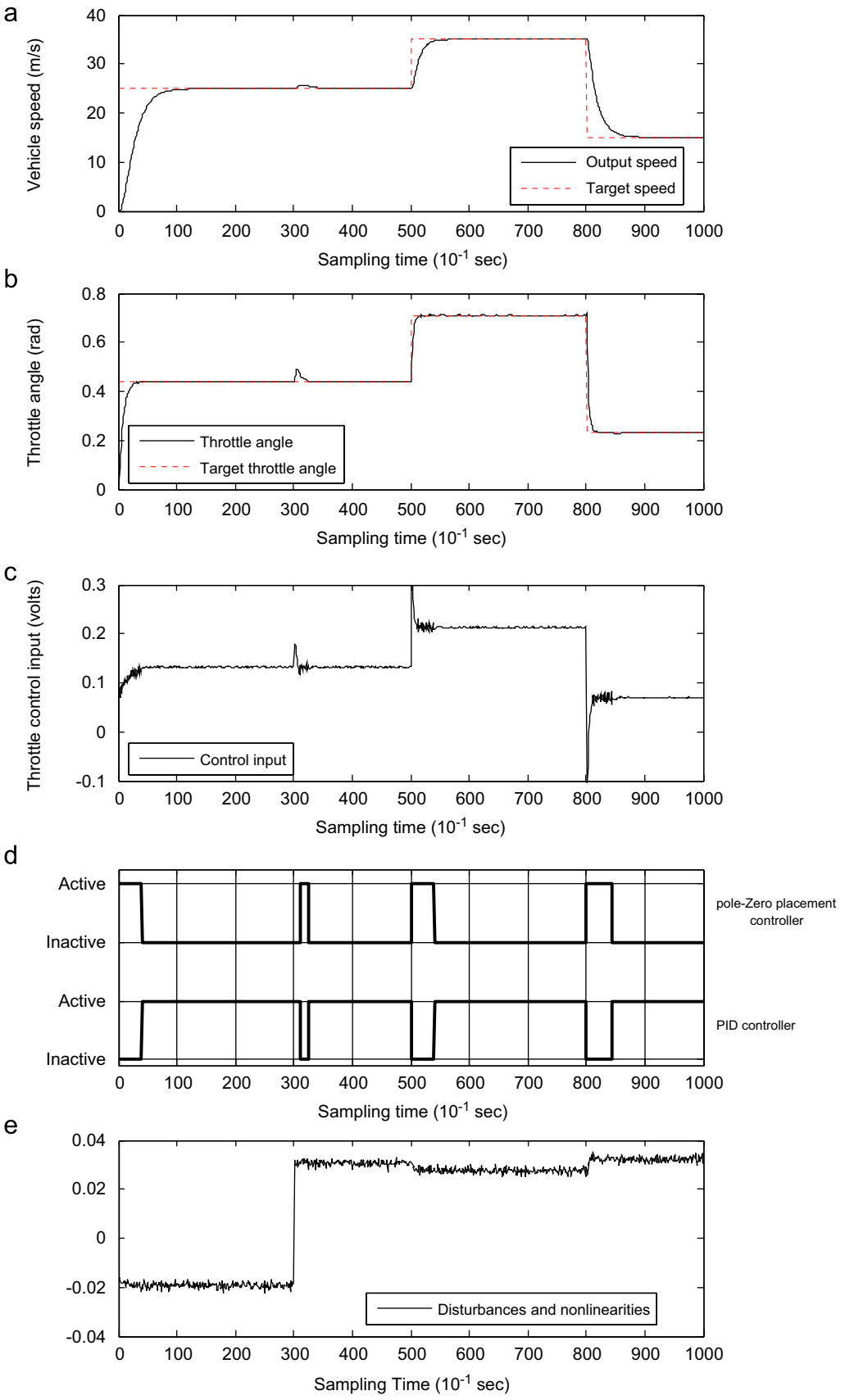


Fig. 8. (a) The speed signal v, (b) the output throttle angle θ, (c) the control input signal $u(t)$, (d) the switching scheme and (e) the nonlinearity and disturbances.

Table 1
Summary of comparative minimum variance performance

	(Control) input signal variance	(Throttle angle) output signal variance	Rise time to target speed (35 m/s) (s)
Experiment 1 (PI controller mode only)	0.0030	0.0305	6.3
Experiment 2 (Pole–zero placement controller mode only)	0.0027	0.0278	8.7
Experiment 3 (multiple-controller with intelligent switching between PI and pole–zero placement controllers)	0.0028	0.0300	4.2

improved performance (both in terms of minimum output-signal variance and rise-times) compared to the stand-alone PI only and pole–zero placement only controllers. These preliminary results demonstrate the need for, and advantages of effectively switching between, and tuning the two multiple controllers using a fuzzy-logic-based supervisor.

Note that another switching issue, which has not been addressed theoretically in this work relates to a potential problem associated with conventional multiple-controller switching paradigms of having to achieve a smooth *bumpless* transfer when switching between controllers [1]. It is always desirable to perform bumpless switching between different controlling scenarios, i.e., switching that does not induce a large transient because of incompatible "initial conditions" of the controller connected to the plant and of the plant itself [36]. On the other hand, as is evident from the sample simulation results above, our proposed framework appears to exhibit the desired bumpless switching between the two controllers. An intuitive explanation of this is that the two multiple-controller modes are derived from a single minimum variance-based cost function, which guarantees compatible initial conditions for both controllers. As a result, the switching decision is made in an integral manner by modifying the common control law (contrary to the case with other classical multiple-controller-switching frameworks) [37].

6. Conclusions

In this paper, a new intelligent nonlinear multiple-controller framework incorporating a fuzzy-logic-based switching and tuning supervisor is developed. The framework integrates the simple fuzzy-rule-based supervisor with the benefits of both the conventional PID and PID-structured pole–zero placement nonlinear controllers along with a GLM framework. In the GLM, the unknown complex process to be controlled is represented by an equivalent stochastic model consisting of a linear time-varying sub-model plus a computationally efficient RBF NN-based learning sub-model. The proposed methodology provides the designer the choice between the conventional PID self-tuning controller, or the PID structure-based (simultaneous) pole and zero placement controller. Both controllers (multiple-controller modes) benefit from the simplicity of having a PID structure, operate using the same adaptive procedure and can be selected on the basis of the required performance measure. The switching decision between the two nonlinear fixed structure controllers, along with on-line tuning of the controller parameters, is made using a fuzzy-logic-based supervisor operating at the highest level of the system. The proposed intelligent framework is applied to the nonlinear ETC system for an autonomous cruise control application. Sample simulation results using a validated nonlinear vehicle model are used to demonstrate the effectiveness of the multiple-controller with respect to adaptively tracking desired vehicle speed changes and achieving the desired speed of response, whilst penalising excessive control action.

For future work, the proposed controller framework will be implemented on a test-bed vehicle in real-time in order to further assess its ability to adapt the vehicle speed to a pre-defined level and additionally, to maintain a safe headway distance (gap) between the controlled vehicle and a preceding vehicle. Also, note that whilst only two controller modes have been considered in this paper, the proposed intelligent switching and tuning supervisory framework readily allows additional controllers to be employed in order to further improve the closed-loop performance. This will also be investigated in future work.

Appendix A. Closed-loop stability analysis of the multiple-controller framework

Stability of the proposed control algorithm for the ACC problem is analysed based on the following assumptions.

Assumption A. Given a positive constant ε and a compact set $S \subset \Re^{n_a \times n_b}$, there exist coefficients W such that $f_0(.,.W)$ approximate the continuous functions $f_0(.,.)$ with accuracy ε, that is

$$\exists W.S.t \max|\hat{f}_0(x, W) - f(x)| \leqslant \varepsilon < \infty; x \in S. \tag{A.1}$$

Assumption B. $f_0(.,.)$ is a bounded quantity [19,40].

Assumption C. The reference signal (desired vehicle speed for the case of the ACC problem) is bounded.

In order to derive the stability of the overall closed-loop system, the stability analysis of each of the two controller modes (pole–zero placement controller and PID controller) is discussed separately in Sections A.1 and A.2, respectively, whereas the stability of the fuzzy switching and tuning supervisory system is discussed in Section A.3.

A.1. Stability analysis of the nonlinear pole– zero placement controller

The closed-loop system transfer function at time t and $(t+k)$ can be described by the following Lemma A.1.1, which is derived from the transfer function of the closed-loop system when the pole–zero placement controller is selected.

Let $H'_0 = \tilde{H}[\tilde{H}(1)]^{-1}F(1)$.

Lemma A.1.1.

$$\begin{bmatrix} B_t P_{n_t} + Q_t P_{d_t} A_t & 0 \\ 0 & B_t P_{n_t} + Q_t P_{d_t} A_t \end{bmatrix} \begin{bmatrix} y(t+1) \\ u(t) \end{bmatrix}$$
$$= \begin{bmatrix} B_k \cdot R_t \\ A_k \cdot R_t \end{bmatrix} w(t) + \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y(t+1) \\ u(t) \end{bmatrix}$$
$$+ \begin{bmatrix} B_k P_{d_t} E_t + Q_t P_{d_t} \\ B_k P_{d_t} E_t - P_{n_t} \end{bmatrix} \xi(t+1) + \begin{bmatrix} B_k \cdot P_{d_t} H_{N_t} + Q_t P_{d_t} \\ A_k \cdot P_{d_t} \cdot H_{N_t} - P_{n_t} \end{bmatrix} \hat{f}_{0,t}(.,.)$$
$$+ \begin{bmatrix} B_k \cdot E_t P_{d_t} \\ A_k \cdot E_t P_{d_t} \end{bmatrix} [f_0(.,.) - \hat{f}_0(.,.)], \tag{A.2}$$

where

$$A_{11} = B_k \cdot (P_{d_t} E_t A_t - P_{d_t} E_t A_k) + (B_t P_{n_t} - B_k \cdot P_{n_t}),$$

$$A_{12} = B_k \cdot (P_{d_t} E_t B_k - P_{d_t} E_t \cdot B_t) + (P_{d_t} Q_t B_t - B_k \cdot P_{d_t} Q_t),$$

$$A_{21} = A_k \cdot (P_{d_t} E_t A_t - P_{d_t} E_t A_k) + (A_t P_{n_t} - A_k \cdot P_{n_t}),$$

$$A_{22} = A_k \cdot (P_{d_t} E_t B_k - P_{d_t} E_t \cdot B_t) + (P_{d_t} A_t Q_t - A_k \cdot P_{d_t} Q_t)$$

and where A_t and A_k denote the estimate values of A at t and $t+k$ moments, respectively, i.e.

$$A_t = A_t(t, z^{-1}), A_k = A_k(t+k, z^{-1}),$$
$$A_t \cdot B_k = A(t, z^{-1}) \cdot B(t+k, z^{-1}) \neq B_k A_t, \tag{A.3}$$

$$A_t \cdot B_t = AB(t, z^{-1}) = B_t A_t. \tag{A.4}$$

Proof of A.1.1. To prove this lemma, reconsider the original CARMA plant model

$$A_k y(t+1) B_k u(t) + f_0(.,.) + C\xi(t+1). \tag{A.5}$$

To simplify the derivation let $C = 1$. □

Multiplying (A.5) by $P_{d_t} E_t$ we obtain

$$P_{d_t} E_t \cdot A_k y(t+1) P_{d_t} E_t \cdot B_k u(t) + P_{d_t} E_t \cdot f_0(.,.) + P_{d_t} E_t \cdot \xi(t+1).$$

Using (A.3) and (A.4) and the identity $P_n = AEP_d + z^{-1}F$ [37]

$$\begin{aligned} P_{n_t} y(t+k) - F_t y(t) - P_{d_t} E_t B_t u(t) - P_{d_t} E_t \hat{f}_{0,i}(.,.) \\ = P_{d_t}(E_t \cdot B_k - E_t B_t) u(t) + P_{d_t}(E_t A_t - E_t \cdot A_k) y(t+k) \\ + P_{d_t} E_t [f_0(.,.) - \hat{f}_0(.,.)] + P_{d_t} E_t^i \cdot \xi(t+1). \end{aligned} \tag{A.6}$$

Then using $P_d(EB+Q)u(t) = [P_d R w(t) - Fy(t) + P_d(H_N - E) f_{0,t}(.,.)]$ [37] and (A.6) gives

$$\begin{aligned} P_{n_t} y(t+k) - P_{d_t} R_t w(t) + P_{d_t} Q_t u(t) - P_{d_t} H_{N_t} \hat{f}_0(.,.) \\ = (P_{d_t} E_t \cdot B_k - P_{d_t} E_t B_t) u(t) \\ + (P_{d_t} E_t \cdot A_t - P_{d_t} E_t \cdot A_k) y(t+k) \\ + P_{d_t} E_t [f_0(.,.) - \hat{f}_0(.,.)] + P_{d_t} E_t \cdot \xi(t+1) \end{aligned} \tag{A.7}$$

The stability and convergence of the algorithm are then as stated below.

Theorem. If Assumptions A, B and C *are satisfied and hold, the recursive parameter estimation algorithm has the following properties*:

$$\lim_{t\to\infty} [y(t)] < \infty; \quad \lim_{t\to\infty} [u(t)] < \infty, \tag{A.8}$$

$$\lim_{t\to\infty} |\phi(t+1)|^2 < \sigma^2 < \infty. \tag{A.9}$$

The boundedness of $y(t)$ and $u(t)$ in (A.8) can be proved by considering in Lemma A.1.1 that the terms in the parentheses of (A.2) tend to zero at $t \to \infty$ subjected to Assumptions A and B, and the boundedness of $w(t)$. Therefore, the algorithm stability is proven. From (A.6) we have that

$$\begin{aligned} \lim_{t\to\infty} |\phi(t+1)|^2 = \lim_{t\to\infty} |P_{d_t}(E_t \cdot B_k - E_t B_t) u(t) + P_{d_t}(E_t A_t - E_t \cdot A_k) y(t+k) \\ + P_{d_t} E_t \cdot \xi(t+1) + P_{d_t} E_t [f_0(.,.) - \hat{f}_0(.,.)]|^2, \end{aligned}$$

$$\lim_{t\to\infty} |\phi(t+1)|^2 \leqslant (P_{d_t} E_t \varepsilon)^2 = \sigma^2 < \infty.$$

Hence, the convergence of (A.9) is proven.

A.2. Stability analysis of the nonlinear PID controller

To prove the stability of the self-tuning nonlinear PID controller, consider the transfer function of the closed-loop system in Eq. (A.10).

$$\begin{aligned} y(t) = \left[1 + \frac{1}{\Delta} z^{-k} \frac{B}{A} vF\right]^{-1} \left[\left(\frac{1}{\Delta} z^{-k} \frac{B}{A}\right) vH_0\right] w(t) \\ + \left[1 + \frac{1}{\Delta} z^{-k} \frac{B}{A} vF\right]^{-1} \left[\frac{C}{A}\right] \xi(t) \\ + \left[1 + \frac{1}{\Delta} z^{-k} \frac{B}{A} vF\right]^{-1} \left[\frac{1}{A} z^{-k}\right] f_{0,t}(.,.) \\ + \left[1 + \frac{1}{\Delta} z^{-k} \frac{B}{A} vF\right]^{-1} \left[\frac{H_N B}{A} z^{-k}\right] f_{0,t}(.,.) \end{aligned} \tag{A.10}$$

After manipulating the transfer function in (41) the resultant equation is, thus,

$$[z^k \Delta A + BvF] y(t) = BvH_0 w(t) + z^k C \Delta \xi(t) + [1 + H_N B] \Delta f_{0,t}(.,.). \tag{A.11}$$

If we let $x_1(t) = C\xi(t)$, $x_2(t) = [1 + H_N B] f_{0,t}(\dots)$ and let $z^{-1} = d$, then the Eq. (A.11) above becomes

$$[d^{-k} \Delta A + BvF] y(t) = BvH_0 w(t) + d^{-k} \Delta x_1(t) + \Delta x_2(t).$$

Let

$$G' = d^{-k} \Delta A + BvF. \tag{A.12}$$

The transfer function from the reference input $w(t)$ to the output $y(t)$ becomes $G_{wy} = [G']^{-1} BvH_0$, the transfer function from the disturbance $x_1(t)$ to the output $y(t)$ becomes $G_{x_1 y} = [G']^{-1} d^{-k} \Delta$, and the transfer function from the disturbance $x_2(t)$ to the output $y(t)$ becomes $G_{x_2 y} = [G']^{-1} \Delta$.

The poles of the closed-loop system are determined by G' and the zeros are those of the open-loop zeros plus additional zeros provided by the term vH_0, assuming that no pole–zero cancellation occurs providing that the pole–zero placement controller is set offline. The condition for the closed-loop stability is then dependent on G' such that, for stability,

$$\det(G') = 0$$

has all its roots strictly outside the unit circle. The requirement is equivalent to G' having nonzero eigenvalues. Therefore, to prove the stability of the closed-loop system, it is necessary to prove that G' is a complex matrix which has an inverse for $|d| < 1$ [2,20,35]. From the identity $P_n = AEP_d + z^{-1}F$ [37], we can derive F as

$$F = d^{-k}(P_n - AEP_d). \tag{A.13}$$

Substituting (A.13) into the expression for G', we obtain

$$G' = d^{-k} A[\Delta + A^{-1} Bv(P_n - AP_d E)].$$

Knowing that $\Delta = (1-d)I$ and $D' = dI$, then G' can be written as

$$G' = d^{-k} A[I + (1-d) dP_{d_1} - D' + A^{-1} Bv(P_n - AP_d E)].$$

Let

$$G'_1 = (1-d) dP_{d_1} - D' + A^{-1} Bv(P_n - AP_d E), \tag{A.14}$$

then (A.12) becomes $G' = d^{-k} A(I + G'_1)$.

For the stability, using the results of [15,16], $\|G'_1\|$ must be less than 1 for all $|d| < 1$. A further requirement is that A is stable. Recall that P_{d_1} is diagonal matrix and letting p_1 be one of the elements of the matrix, we can choose $-0.5 \leqslant d(1-d)p_1 - 1 \leqslant 0.5$. Therefore, $\|(1-d)dP_1 - d\|$ is less than 0.5 if we select $((-0.5/d)+1/(1-d)) \leqslant p_1 \leqslant ((0.5/d)+1(1-d))$.

Finally, it is necessary to prove that the remaining term in (A.14) has a modulus less that 0.5 with the assumption that the term is bounded. Therefore, we can consider that for stability, v, P_n and P_d can be chosen small enough such that the modulus of the remaining term is less than 0.5. Hence, referring to the triangular

inequality $\|G'_1\| \leqslant \|(1-d)dP_{d_1} - D'\| + \|A^{-1}Bv(P_n - AP_dE)\|$. This makes $\|G_1'\|$ less than 1 and the stability of the closed-loop system is proved. □

A.3. Stability of the fuzzy switching and tuning supervisory system

Having proven the stability of the two control modes 1 and 2, based on the work of [23,31], this section will outline the stability of the fuzzy switching and tuning supervisory system.

Let the system state vector at time instant k be $\bar{x}(k) = [x_1(k)\ldots x_n(k)]^T$, where $x_1(k)\ldots x_n(k)$ are the state variables of the system at time instant k, and the controllers state vector at time k be $\bar{u}(k) = [u_1(k)\ldots u_m(k)]$, where $u_1(k)\ldots u_m(k)$ are controller state variables and m is the number of controllers. Then the fuzzy system is defined by the implications below

$$R^i: \text{IF } (x_1(k) \text{ is } S_1^i, \text{ AND} \ldots \text{AND } x_n(k) \text{ is } S_n^i) \text{ THEN } \bar{u}(k+1) \text{ is } u_g(k), \tag{A.15}$$

for $i = 1\ldots N$ and $g = 1\ldots M$. Here, S_1^i is the fuzzy set corresponding to the state variable x_i and implication R^l. The truth value of the implication R^l a time instant k denoted by $w_i(k)$ is defined as

$$w_i(k) = \wedge(\mu_{S_1^i}(x_1(k)), \ldots, \mu_{S_1^i}(x_n(k))),$$

where $\mu_s(x)$ is the membership function value of the fuzzy set S at the position x and $\wedge$ is an operator satisfying

$$\min(l_1, \ldots, l_n) \geqslant \wedge(l_1, \ldots, l_n) \geqslant 0.$$

Usually $\wedge$ is taken to be the minimum operator which gives the minimum of its operands. Then, at instant k the controller's state vector is updated according to

$$\bar{u}(k+1) = \frac{(\sum_{i=1}^N w_i(k)A_i\bar{x}(k))}{\sum_{i=1}^N w_i(k)} = \sum_{i=1}^N \alpha_i(k)A_i\bar{x}(k); \alpha_i(k) = \frac{w_i(k)}{\sum_{i=1}^N w_i(k)}. \tag{A.16}$$

A fuzzy system is completely represented by the set of characteristic matrices $\bar{A} = [A_1,\ldots A_n]$ and the fuzzy sets S_j^l, $l = 1\ldots N$; $j = 1\ldots n$. Corresponding to this fuzzy system, a corresponding switching system is described below.

The state update at time instant k is given as follows:

$$\bar{x}(k+1) = A\bar{x}(k), \tag{A.17}$$

where $A \in \bar{A}$ (i.e., it is one of the matrices $A_1,\ldots,A_n$).

The following is a definition of global asymptotic stability of the switching and tuning system.

Definition A.3.1. The fuzzy system described in (A.16) is globally stable if

$$\bar{x}(k) \to 0 \text{ as } k \to \infty$$

or equivalently, there exists $\|\cdot\|$, a norm on $\Re^n$

$$\|\bar{x}(x)\| \to 0 \text{ as } k \to \infty$$

for all initial values $\bar{x}(k) \in \Re^n$ and for all possible fuzzy sets S_i^j, $\forall i = 1\ldots N$, $\forall j = 1\ldots n$.

Definition A.3.2. The switching system described in (A.17) is globally asymptotically stable if

$$\bar{x}(k+1) = A(k)\bar{x}(0) \to 0 \text{ as } k \to \infty; \quad \forall \bar{x}(0) \in \Re^n,$$

where $A(k) \in \bar{A}_k$. Equivalently $A(k) \to 0$ as $k \to \infty$; $A(k) \in \bar{A}_k$.

Theorem A.3.1. *A necessary and sufficient condition for the stability as in* Definition A.3.1 *of fuzzy system (A.16) is that the corresponding switching system (A.17) be stable, as in* Definition A.3.2.

The proof of the above theorem is presented in [31].

Finally, the multiple-controller framework was not found to exhibit any transfer problems during the switching mode in any of the simulations. An intuitive explanation for this was given in Section 5.3. In continuous systems, the problem of transition between the various controller modes can be solved by using a hold circuit. In our case, the system is discrete and the hold circuit is not needed and since the controllers exhibit bumpless switching, the stability of the intelligent controller is achieved.

References

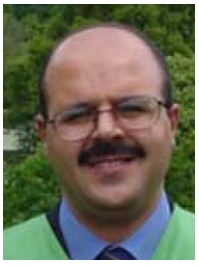
[1] A. Breemen, T. Vries, An agent-based framework for designing multiple-controller systems, in: Proceedings of the Fifth International Conference on the Practical Applications of Intelligent Agents & Multi-Agents Technology, Manchester, UK, 10–11 April 2000, pp. 219–235.
[2] D. Cao, P. He, Stability criteria of linear neutral systems with a single delay, Appl. Math. Comput. 148 (1) (2004) 135–143.
[3] R. Conatser, J. Wagner, S. Ganta, I. Walker, Diagnosis of automotive electronic throttle control systems, Control Eng. Pract. 12 (2004) 23–30.
[4] R. Davies, M. Zarrop, On reduced variance overparametrized pole—assignment control, Int. J. Control 69 (1) (1998) 131–144.
[5] M. Farsi, K. Karam, H. Abdalla, Intelligent multi-controller assessment using fuzzy logic, J. Fuzzy Sets Syst. 79 (1) (1996) 25–41.
[6] A. Girard, J. Hedrick, Real-time embedded hybrid control software for intelligent cruise control applications, IEEE Robotics Automot. Mag., Spec. Issue Intelligent Transp. Syst. 12 (2005) 22–28.
[7] A. Girard, A. Howell, J. Hedrick, Model-driven hybrid and embedded software for automotive applications, Second RTAS Workshop on Model-Driven Embedded Systems, 2004, pp. 25–28.
[8] R. Gregor, M. Lützeler, M. Pellkofer, K. Siedersberger, E. Dickmanns, EMS-Vision: a perceptual system for autonomous vehicles, IEEE Trans. Intelligent Transp. Syst. 3 (1) (2002) 48–59.
[9] P. Griffiths, Embedded software control design for an electronic throttle body, Master of Science Thesis, University of California, Barkeley, USA, 2002.
[10] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan, USA, 1994.
[11] J. Hespanha, D. Liberzon, A. Morse, B. Anderson, T. Brinsmead, D. Bruyne, Multiple model adaptive control. Part 2: switching, Int. J. Robust Nonlinear Control 11 (2001) 479–496.
[12] S. Huang, W. Ren, Use of neural fuzzy networks with mixed genetic/gradient algorithm in automotive vehicle control, IEEE Trans. Ind. Electron. 46 (6) (1999) 1090–1102.
[13] P. Ioannou, Z. Xu, Throttle and brake control systems for automatic vehicle following, IVHS J. 1 (4) (1994) 345–377.
[14] L. Kun, P. Ioannou, Modeling of traffic flow of automated vehicles, IEEE Trans. Intelligent Transp. Syst. 5 (2) (2004) 99–113.
[15] P. Lancaster, Theory of Matrices, Academic Press, New York, 1969.
[16] P. Lancaster, M. Tismenetsky, The Theory of Matrices, Academic Press, Orlando, FL, 1985.
[17] J. Lee, H.K.S. Lee, W.C. Kim, Model-based iterative learning control with a quadratic criterion for time-varying linear systems, Automatica 36 (5) (2000) 641–657.
[18] Y. Li, S. Qiang, X. Zhuang, O. Kaynak, Robust and adaptive backstepping control for nonlinear systems using RBF neural networks, IEEE Trans. Neural Networks 15 (3) (2004) 693–701.
[19] L. Mei-Qin, Stability analysis of neutral-type nonlinear delayed systems: an LMI approach, J. Zhejiang Univ. Sci. A 7 (2) (2006) 237–244.
[20] K. Najim, Control of Continuous Linear Systems, ISTE Ltd, 2006.
[21] J. Naranjo, C. Gonzalez, J. Reviejo, R. Garcia, T. Pedro, Adaptive fuzzy control for inter-vehicle gap keeping, IEEE Trans. Intelligent Transp. Syst. 4 (3) (2003) 132–142.
[22] J. Naranjo, C. Gonzalez, R. Garcia, T. Pedro, Using fuzzy in automated vehicle control, IEEE Intelligent Syst. 22 (1) (2007) 36–45.
[23] K. Narendra, C. Xiang, Adaptive control of discrete-time systems using multiple models, IEEE Trans. Autom. Control 45 (9) (2000) 1669–1686.
[24] D. Nguyen, B. Widrow, Neural networks for self-learning control systems, IEEE Control Syst. Mag. 10 (3) (1990) 18–23.
[25] C. Panchapakesan, M. Palaniswami, D. Ralph, C. Manzie, Effects of moving the centers in an RBF network, IEEE Trans. Neural Networks 13 (6) (2002) 1299–1307.
[26] J. Park, R. Harley, G. Venayagamoorthy, Comparison of MLP and RBF neural networks using deviation signals for indirect adaptive control of a synchronous generator, IEEE-INNS International Conference on Neural Networks, vol. 1, Honolulu, USA, 2002, 919–924.
[27] K. Passino, Biomimcry for Optimization, Control, and Automation, Springer, Berlin, 2005.
[28] K. Passino, S. Yukovich, Fuzzy Control, Addison-Wesley, Reading, MA, 1998.
[29] C. Pous, J. Colomer, J. Melendez, J. de la Rosa, Fuzzy identification for fault isolation. Application to analog circuits diagnosis, A.I. Res. Dev. 1 (2003) 409–420.
[30] R. Sanner, J. Slotine, Gaussian networks for direct adaptive control, IEEE Trans. Neural Networks 3 (6) (1992) 837–863.

[31] M. Thathachar, P. Viswanath, On the stability of fuzzy systems, IEEE Trans. Fuzzy Syst. 5 (1) (1997) 145–151.
[32] M. Tokuda, T. Yamamoto, A neural-net-based controller supplementing a multiloop PID control system, IEICE Trans. Fundam. E 85-A (1) (2002) 256–261.
[33] C. Toy, K. Leung, L. Alvarez, R. Horowitz, Emergency vehicle maneuvers and control laws for automated highway systems, IEEE Trans. Intelligent Transp. Syst. 3 (2) (2002) 109–118.
[34] R. Yusof, S. Omatu, M. Khalid, Self-tuning PID control: a multivariable derivation and application, Automatica 30 (1994) 1975–1981.
[35] R. Yusof, S. Omatu, A multivariable self-tuning PID controller, Int. J. Control 57 (6) (1993) 1387–1403.
[36] L. Zaccarian, A. Teel, A common framework for anti-windup, bumpless transfer and reliable design, Automatica 38 (10) (2002) 1735–1744.
[37] A. Zayed, A. Hussain, R. Abdullah, A novel multiple-controller incorporating a radial basis function neural network-based generalized learning model, Neurocomputing (Elsevier Science B.V.) 69 (16–18) (2006) 1868–1881.
[38] A.S. Zayed, A. Hussain, M.J. Grimble, A nonlinear PID-based multiple controller incorporating a multilayered neural network learning submodel, Int. J. Control Intelligent Syst. 34 (3) (2006) 201–1499.
[39] Y. Zhang, E. Kosmatopoulos, P. Ioannou, Autonomous intelligent cruise control using front and back information for tight vehicle following maneuvers, IEEE Trans. Veh. Tech. 48 (1) (1999) 319–328.
[40] Q. Zhu, Z. Ma, K. Warwick, Neural network enhanced generalised minimum variance self-tuning controller for nonlinear discrete-time systems, IEE Proc. Control Theory Appl. 146 (4) (1999) 319–326.
[41] Q. Zhu, K. Warwick, A neural network enhanced generalized minimum variance self-tuning proportional, integral and derivative control algorithm for complex dynamic systems, J. Syst. Control Eng., Proc. Inst. Mech. Eng. Part I 126 (3) (2002) 265–273.

Rudwan A. Abdullah received his B.Sc. degree in Computer Engineering in 1994 from the Engineering Academy, in Tripoli, Libya, and his M.Sc. degree in Software Engineering in 1998 from the University of Stirling in Scotland, UK. He is currently a Ph.D. candidate and Teaching Assistant in Computing Science at the University of Stirling funded by the Biruni Remote Sensing Centre. His research interests include intelligent modeling and control of complex Systems, neural networks, fuzzy logic, image processing and geographical information systems (GIS).

Amir Hussain obtained his B.Eng. (with First Class Honours) and his Ph.D. in Electronic and Electrical Engineering in 1992 and 1996, respectively, both from the University of Strathclyde in Glasgow, Scotland UK. From 1996 to 1998, he held a post-doctoral research fellowship at the University of Paisley in Scotland, UK. From 1998 to 2000, he held a Research Lectureship in Applied Computing Science at the University of Dundee in Scotland, UK. Since 2000, he is with the University of Stirling in Scotland, UK where he is currently a Senior Lecturer in Computing Science. His research interests include novel interdisciplinary research for modelling and control of complex systems, adaptive nonlinear speech signal processing and computational intelligence techniques and applications. His research activities have been funded by, amongst others, the UK Engineering & Physical Sciences Research Council (EPSRC), the UK Royal Society, the European Commission and industry. These have led to one international patent in neural computation and more than 100 publications to-date in various international journals, books and refereed international conference and workshop proceedings. He is IEEE Chapter Chair for the IEEE UK & RI Industry Applications Society Chapter. He currently serves on the editorial board of a number of international Journals and has acted as an invited Guest Editor for numerous Journals' Special Issues, including the (Elsevier) Neurocomputing Journal Special Issue on IEEE ICEIS'2006. He serves as an independent Expert for the European Commission's 7th Framework Program for RTD, and as a Consultant for the Pakistan Higher Education Commission, Islamabad.

Kevin Warwick is Professor of Cybernetics at the University of Reading, England, where he carries out research in artificial intelligence, control, robotics and biomedical engineering. He is also Director of the University KTP Centre, which links the University with Small to Medium Enterprises and raises over £2 Million each year in research income for the University. Kevin took his first degree at Aston University, followed by a Ph.D. and a research post at Imperial College, London. He subsequently held positions at Oxford, Newcastle and Warwick universities before being offered the Chair at Reading. He has been awarded higher doctorates (D.Sc.s) both by Imperial College and the Czech Academy of Sciences, Prague. He was presented with The Future of Health technology Award from MIT (USA), was made an Honorary Member of the Academy of Sciences, St. Petersburg and received The IEE Achievement Medal in 2004. In 2000, Kevin presented the Royal Institution Christmas Lectures. He is perhaps best known for carrying out a pioneering set of experiments involving the implant of multi-electrodes into his nervous system. With this in place, he carried out the world's first experiment involving electronic communication directly between the nervous systems of two humans.



Ali S. Zayed obtained his B.Sc. in Electronic Engineering from the Higher Institute of Electronics (Libya) in 1989. He then joined Sirte Oil Company as an electronic engineer and was awarded a scholarship to pursue an M.Phil. program in Electrical and Electronic Engineering at the University of Strathclyde, in Glasgow, UK (1996–1997). In 1998, he was promoted to a senior engineer and in 1999, he joined the Department of Electrical Engineering at the University of Seventh April as a Lecturer. He was awarded a scholarship by the Seventh April University in 2001 to pursue his Ph.D. program in Adaptive Control Engineering at the University of Stirling in Scotland, UK. From 2003 to 2004, he worked as a Research Fellow at Stirling University on the UK Engineering and Physical Sciences Research Council (EPSRC) funded Project titled: "Towards Multiple-Model-based Learning Control Paradigms". He obtained his Ph.D. in 2005 and is currently a Faculty member at Seventh April University in Libya. His research interests include adaptive minimum variance-based learning control techniques and applications.