amcs

INDIRECT ADAPTIVE CONTROLLER BASED ON A SELF–STRUCTURING FUZZY SYSTEM FOR NONLINEAR MODELING AND CONTROL

RUIYUN QI *, MIETEK A. BRDYS **,***

* College of Automation Engineering
Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, P. R. China
e-mail: `ruiyun.qi@nuaa.edu.cn`

** Department of Electronic, Electrical and Computer Engineering
University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
e-mail: `m.brdys@bham.ac.uk`

*** Department of Control Systems Engineering, Faculty of Electrical and Control Engineering
Gdańsk University of Technology, ul. Narutowicza 11/12, 80–952 Gdańsk, Poland
e-mail: `mbrdys@ely.pg.gda.pl`

In this paper, a unified nonlinear modeling and control scheme is presented. A self-structuring Takagi-Sugeno (T-S) fuzzy model is used to approximate the unknown nonlinear plant based on I/O data collected on-line. Both the structure and the parameters of the T-S fuzzy model are updated by an on-line clustering method and a recursive least squares estimation (RLSE) algorithm. The rules of the fuzzy model can be added, replaced or deleted on-line to allow a more flexible and compact model structure. The overall controller consists of an indirect adaptive controller and a supervisory controller. The former is the dominant controller, which maintains the closed-loop stability when the fuzzy system is a good approximation of the nonlinear plant. The latter is an auxiliary controller, which is activated when the tracking error reaches the boundary of a predefined constraint set. It is proven that global stability of the closed-loop system is guaranteed in the sense that all the closed-loop signals are bounded and simulation examples demonstrate the effectiveness of the proposed control scheme.

Keywords: fuzzy control, self-structuring fuzzy model, on-line modeling, stability.

1. Introduction

The control of complex nonlinear plants is always one of the most challenging tasks in the control field, especially when it is difficult to obtain the mathematical model of the plant through conventional physical analysis methods or there exist uncertainties in the plant. In recent decades, fuzzy systems have become a powerful tool in nonlinear identification and control due to their advantages in dealing with uncertainties and capturing nonlinearities. Fuzzy systems transfer human knowledge and experience described by a linguistic language into mathematical descriptions in order to utilize them. A fuzzy system consists of a group of fuzzy rules which can either be built by experts or extracted from training data. This characteristic makes them more flexible in the identification of nonlinear plants. There are two main types of fuzzy systems: Mamdani fuzzy systems and Takagi-Sugeno (T-S) fuzzy

systems. Both have been proven to be universal approximators (Hao, 1998; Hao and Shao, 1999), which provides a strong support for their successful application in nonlinear identification. Fuzzy systems can be used in both direct and indirect control of nonlinear plants. In the former, a fuzzy system is used to approximate an ideal control law, while in the latter it is used to approximately represent an unknown nonlinear plant, based on which the control law is designed. In both cases, the first problem to be solved is how to get such a fuzzy system. As we have mentioned, it can be built by employing expert knowledge or from data by using proper training algorithms. However, since both expert knowledge and off-line data are limited and cover only part of the truth of the real plant, the initial fuzzy system obtained using the above methods may only be a local model. When such a fuzzy system is used in a closed-loop control system, it may not represent the nonlinear plant with acceptable accuracy when the parameters

of the plant change due to internal or external uncertainties or the plant moves into a new operating region not covered by off-line training data. In such situations, the fuzzy system needs to be updated on-line to regain good approximation, that is, it should be an adaptive system. An appropriate on-line learning algorithm plays an important role in designing adaptive fuzzy systems.

In the literature, various adaptive fuzzy systems have been proposed (Qi and Brdys, 2008; Park and Cho, 2004; Chien and Yao, 2004). However, in most of them, only the parameters of the fuzzy system are adjusted on-line while the structure, including the number of fuzzy rules and membership functions, is fixed and assumed to be correct. Due to the limited knowledge and data the designer owns, the initial fuzzy rule number may be set too large or too small. A large number of fuzzy rules can cover more situations and operating regions the plant meets, but it will cause more complex computations and make real-time implementations very difficult and even impossible. A small number of fuzzy rules requires fewer computations but may not be capable of approximating the plant accurately when it enters new operating regions which are not covered by the current membership functions.

Moreover, the parameters of the fuzzy system may be unnecessarily tuned to try to compensate the error brought by the incorrect structure, resulting in an even worse model. The structure of a fuzzy system is usually identified through fuzzy clustering. Various clustering mechanisms can be applied to generate the fuzzy rule base, including fuzzy C-means (Bezdek, 1974), Gustafson-Kessel clustering (Gustafson and Kessel, 1979), subtractive clustering (Chiu, 1994), etc. They are very useful in an off-line or batch learning mode, when all I/O data are available. However, not all of them are suitable for on-line clustering.

In (Angelov and Filev, 2004), an approach to on-line identification of the T-S fuzzy model is proposed, which is based on a novel learning algorithm that recursively updates both the T-S model structure and parameters. The model structure is identified through an on-line clustering algorithm which is developed from subtractive clustering. It is pointed out that the learning could start without *a priori* information and with only one data sample, which potentially makes it very suitable for adaptive control. However, in (Angelov and Filev, 2004), this algorithm is only applied to open-loop identification and not to closed-loop control. In (Gao and Er, 2003), a self-organizing fuzzy system is proposed for adaptive control of nonlinear plants. Fuzzy rules can be generated on-line and the self-structuring fuzzy system can be used to approximate either the nonlinear plant or the perfect control law. However, it is assumed that a part function of the nonlinear plant is known, which is not always true in reality and requires the memory of all the past I/O data at each step. Stability analysis considers only a closed-loop system with a fixed structure. In (Phan and Gale, 2008), a direct adaptive fuzzy control scheme is presented and a self-structuring fuzzy system is employed to approximate the ideal control law. Both the structure and parameters of the fuzzy system can be tuned on-line, and the system is shown to be stable when the structure changes.

In our paper, we consider indirect adaptive control based on a self-structuring fuzzy system. The overall controller consists of two parts: the Indirect Adaptive Controller based on the Self-structuring Fuzzy System (IACSFS) is the dominant controller, which maintains the closed-loop stability when the fuzzy system is a good approximation of the nonlinear plant. A supervisory controller is an auxiliary controller, which is activated when the state reaches the boundary of a predefined constraint set. The supervisory controller helps to generate useful data and allows enough time for the fuzzy system to learn and improve on-line. When the fuzzy system regains good approximation through learning and the model based main controller is capable of maintaining system stability, the supervisory controller is idle. It is shown that the overall adaptive control scheme with the IACSFS and the supervisory controller guarantees global stability in the sense that all the closed-loop signals are bounded.

2. Problem statement

Assume that an SISO nonlinear plant can be accurately represented by the following discrete-time input-output nonlinear dynamics described in the state-space form:

$$
\begin{aligned}
x_j(k+1) &= x_{j+1}(k), \quad j = 1, 2, \ldots, n-1, \\
x_n(k+1) &= f(x(k)) + g(x(k))u(k), \\
y(k) &= x_n(k),
\end{aligned} \tag{1}
$$

where $y \in \mathbb{R}$, $u \in \mathbb{R}$ are the plant output and input, respectively, $x(k) = [x_1(k), x_2(k), \cdots, x_n(k)]^T \triangleq [y(k-n+1), \cdots, y(k)]^T$ is the vector of past outputs that can be measured (n is assumed to be known), and f and g are unknown nonlinear functions which satisfy the following assumption.

Assumption 1 (Wang, 1994). Functions $f^U(x)$, $g^U(x)$ and $g_L(x)$ can be determined such that $|f(x)| \leq f^U(x) < \infty$ and $0 < g_L(x) \leq g(x) \leq g^U(x) < \infty$.

The control objective is to force the plant output to track the reference $y_r(k)$ while all the closed-loop signals remain bounded.

Assumption 2. The reference trajectory $y_r(k)$ satisfies

$$
|y_r(k)| \leq U_r, \qquad \forall k > 0, \tag{2}
$$

where U_r is a known bound.

3. Control algorithm based on a T-S fuzzy model

3.1. Nonlinear plant representation with a T-S fuzzy model.
Since in (1) f and g are unknown functions, a T-S fuzzy model is employed to represent the nonlinear plant with the following N fuzzy rules:

$$Rule\ i : IF\ x_1(k)\ is\ M_1^i\ and\ \cdots\ and\ x_n(k)\ is\ M_n^i;$$
$$THEN\ \hat{y}^i(k+1) = \theta_0^i + \theta_1^i x_1(k) + \cdots + \theta_n^i x_n(k)$$
$$+\theta_{n+1}^i u(k), \tag{3}$$

where N is the number of rules and M_j^i denotes both a fuzzy set and its Gaussian membership grade

$$M_j^i(x_j(k)|c_j^i, \sigma_j^i) = \exp\left\{-\frac{(x_j(k) - c_j^i)^2}{2(\sigma_j^i)^2}\right\}, \quad (4)$$

$i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, n$. The global T-S fuzzy model is inferred as follows:

$$\begin{aligned}\hat{y}(k+1) &= \sum_{i=1}^N \bar{\omega}^i(x(k)|c, \sigma)\theta^{iT}[1\ x^T(k)\ u(k)]^T \\ &= \sum_{i=1}^N \bar{\omega}^i(x(k)|c, \sigma)\theta_f^{iT}[1\ x^T(k)]^T \\ &\quad + \sum_{i=1}^N \bar{\omega}^i(x(k)|c, \sigma)\theta_g^i u(k) \\ &= \hat{f}(x(k)|c, \sigma, \Theta_f) + \hat{g}(x(k)|c, \sigma, \Theta_g)u(k),\end{aligned} \tag{5}$$

where

$$\begin{aligned}\theta_f^i &= [\theta_0^i, \theta_1^i, \theta_2^i, \ldots, \theta_n^i]^T, \\ \theta_g^i &= \theta_{n+1}^i, \\ \theta^i &= [\theta_0^i, \theta_1^i, \theta_2^i, \ldots, \theta_{n+1}^i]^T, \\ \Theta_f &= [\theta_f^{1T}, \theta_f^{2T}, \ldots, \theta_f^{NT}]^T, \\ \Theta_g &= [\theta_g^1, \theta_g^2, \ldots, \theta_g^N]^T, \\ \bar{\omega}^i(\cdot) &= \frac{\omega^i(\cdot)}{\sum_{i=1}^N \omega^i(\cdot)}, \\ \omega^i(\cdot) &= \prod_{j=1}^n M_j^i(\cdot).\end{aligned} \tag{6}$$

To proceed with control design and stability analysis, first let us define

$$\begin{aligned}&c^*, \sigma^*, \Theta_f^*, \Theta_g^* \\ &= \underset{\substack{\Theta_f \in O_{\Theta_f}, \Theta_g \in O_{\Theta_g} \\ c \in O_c, \sigma \in O_\sigma}}{\arg\min}\left[\sup_{x \in X, u \in U_c}|\hat{f}(x|c, \sigma, \Theta_f) \right. \\ &\quad \left. + \hat{g}(x|c, \sigma, \Theta_g)u - (f(x) + g(x)u)|\right],\end{aligned} \tag{7}$$
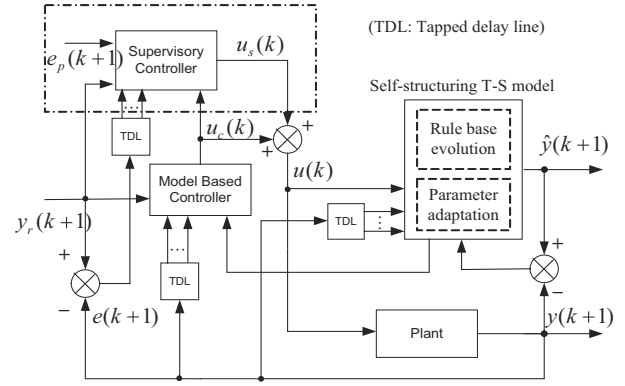

Fig. 1. Overall control structure.

where $X, U_c, O_c, O_\sigma, O_{\Theta_f}$ and O_{Θ_g} are constraint sets for $x, u, c, \ \sigma, \ \Theta_f$ and Θ_g, respectively, specified by the designer. Then the T-S model (5) with the best parameters $\left\{c^*, \sigma^*, \Theta_f^*, \Theta_g^*\right\}$ is

$$\begin{aligned}\hat{y}(k+1) &= \hat{f}(x(k)|c^*, \sigma^*, \Theta_f^*) \\ &\quad + \hat{g}(x(k)|c^*, \sigma^*, \Theta_g^*)u(k).\end{aligned} \tag{8}$$

Next, let us define the *modeling error* as

$$\begin{aligned}w &= \hat{f}(x(k)|c^*, \sigma^*, \Theta_f^*) + \hat{g}(x(k)|c^*, \sigma^*, \Theta_g^*)u(k) \\ &\quad - [f(x(k)) + g(x(k))u(k)],\end{aligned} \tag{9}$$

which is the minimum approximation error that can be achieved with (7).

Assumption 3. The modeling error is negligible.

3.2. Model-based controller.
If $\{c^*, \sigma^*, \Theta_f^*, \Theta_g^*\}$ are known, a certainty equivalent control law can be designed based on the T-S fuzzy model (8):

$$u_c(k|p^*) = \frac{1}{\hat{g}(x(k)|p_g^*)}\left[-\hat{f}(x(k)|p_f^*) + v(k)\right], \quad (10)$$

where

$$\begin{aligned}p^* &\triangleq \left\{c^*, \sigma^*, \Theta_f^*, \Theta_g^*\right\}, \\ p_f^* &\triangleq \left\{c^*, \sigma^*, \Theta_f^*\right\}, \\ p_g^* &\triangleq \left\{c^*, \sigma^*, \Theta_g^*\right\},\end{aligned}$$

and $v(k)$ is a new input chosen as

$$v(k) = y_r(k+1) + k^T e(k), \tag{11}$$

where $e(k) = y_r(k) - y(k)$ and

$$\begin{aligned}e(k) &= \ [e_1(k), \ldots, e_{n-1}(k), e_n(k)]^T \\ &\triangleq \ [e(k-n+1), \ldots, e(k-1), e(k)]^T, \quad (12)\end{aligned}$$

Using (22), the values of the Lyapunov function along the tracking error trajectory (20) in the closed-loop system satisfy

$$
\begin{aligned}
\Delta V(e(k)) &= \frac{1}{2}e^T(k+1)Pe(k+1) - \frac{1}{2}e^T(k)Pe(k) \\
&= -\frac{1}{2}e^T(k)Qe(k) + V_S(k),
\end{aligned}
$$
(24)

where $V_S(k)$ consists of all the elements on the right side of (24) except $-\frac{1}{2}e^T(k)Qe(k)$.

To achieve $\Delta V(e(k)) \le 0$, we may make $V_S(k) \le 0$ by properly designing $u_s(k)$. Let $e_p(k+1)$ be the prediction of $e(k+1)$ at k. We choose the *supervisory controller* action $u_s(k)$ as

$$
\begin{aligned}
&u_s(k) \\
&= \varsigma(k)\mathrm{sgn}(e_p^T(k+1)Pb)\frac{1}{g_L(x)}\Big(g^U(x)|u_c(k|\hat{p})| \\
&+ f^U(x) + |y_r(k+1)| + |k^Te(k)|\Big).
\end{aligned}
$$
(25)

The first term in (25) is activated by the variable $\varsigma(k)$ if the tracking error is outside the desired accuracy ball. Let ς denote a desired tracking error bound. We have $\varsigma(k) = 1$ if $\|e(k)\| > E$ and $\varsigma(k) = 0$ if $\|e(k)\| \le E$ for all k. All the other elements in (25) can be determined as well. Therefore, $u_s(k)$ can be implemented. We proceed to show in the following subsection that closed-loop boundedness can be guaranteed with the supervisory controller (25).

3.4. Closed-loop stability. Before we proceed with stability analysis, we make the following assumption.

Assumption 5. The sign of $e_p^T(k+1)Pb$ is the same as that of $e^T(k+1)Pb$.

Meeting the above assumption requires prediction accuracy not necessarily with respect to the value but with respect to the sign only.

Lemma 1. *If P is a positive definite symmetric matrix and p_{nn} is its (n, n)-th element, then $p_{nn} > 0$.*

Proof. See Appendix A. ∎

Theorem 1. *Consider the nonlinear plant represented by (1). If the supervisory controller is designed as in (25) and Assumptions 1–5 are satisfied, then it can be guaranteed that all the signals in the closed-loop system are bounded.*

Proof. See Appendix B. ∎

4. On-line self-structuring algorithm for model adaptation

In Section 3, a certainty equivalent controller and a supervisory controller are designed, which are proven to be able to guarantee the boundedness of all signals in the closed-loop system. However, with only the non-adaptive certainty equivalent controller and the supervisory controller, the resulting tracking performance may not be satisfactory. As introduced in Section 1, the model obtained through off-line training may only cover a part of the truth of the real plant dynamics because of the limitations of expert knowledge and training data. Therefore, the model needs to be updated on-line to provide an improved model by using the latest on-line information. There are few approaches in the literature to vary the model structure in the closed-loop system because they may cause instability and there are, to the best of our knowledge, no complete stability results on model structure adaptation in the closed-loop fuzzy control system. However, the supervisory controller can be considered as a kind of safeguard to maintain boundedness in the closed-loop system while updating the model structure. In this section, we will present how to update both the model structure and parameters on-line under the protection of supervisory control.

In off-line identification of a T-S fuzzy model, the model structure, including the number of fuzzy rules and the parameters of membership functions, can be identified through well-established fuzzy clustering methods such as fuzzy C-means (Bezdek, 1974), the G-K clustering method (Gustafson and Kessel, 1979) and the subtractive clustering method (Chiu, 1994). These methods are successful when all the input-output data are available. However, when the data are collected continuously on-line, they cannot be applied directly. An approach to identify T-S fuzzy models on-line is proposed by (Angelov and Filev, 2004), and is developed from a subtractive clustering method. The authors illustrate the viability and efficiency of the on-line identification method to generate effective T-S fuzzy models when used with a limited amount of initial information through open-loop identification simulation examples and suggest that this approach is very promising in applications to on-line modeling and control of complex processes.

The on-line identification procedure includes two phases: on-line clustering and on-line parameter identification.

4.1. On-line clustering. In this approach, each coming data point ($z(k) = [x^T(k-1), y(k)]^T$) is viewed as a candidate prototype cluster center. The online clustering procedure starts with the first data point as the focal point of the first cluster. Its coordinates are used to form the antecedent part the first fuzzy rule and its potential is set to 1. From the second data point, the potential of new data

points is calculated recursively using the potential function

$$P_k(z(k)) = \left(1 + \frac{1}{(k-1)} \sum_{m=1}^{k-1} \sum_{j=1}^{n+1} \left(d_{mk}^j\right)^2\right)^{-1}, \quad (26)$$

where $P_k(z(k))$ denotes the potential of the data point $z(k)$ at time k; $d_{mk}^j = z_j(m) - z_j(k)$ denotes the projection of the distance between two data points, $z_j(m)$ and $z_j(k)$, on the z_j axis.

The potential of the new data point is recursively calculated as follows:

$$P_k(z(k)) = \frac{k-1}{(k-1)(\vartheta_k + 1) + \sigma_k - 2\upsilon_k}, \quad (27)$$

where

$$\vartheta_k = \sum_{j=1}^{n+1} \left(z_j(k)\right)^2,$$

$$\sigma_k = \sum_{m=1}^{k-1} \sum_{j=1}^{n+1} \left(z_j(m)\right)^2,$$

$$\upsilon_k = \sum_{j=1}^{n+1} z_j(k)\beta_k^j,$$

$$\beta_k^j = \sum_{m=1}^{k-1} z_j(m).$$

In (27), the parameters ϑ_k and υ_k can be calculated from the current data point $z(k)$, while β_k^j and σ_k are recursively updated as

$$\sigma_k = \sigma_{k-1} + \sum_{j=1}^{n+1} \left(z_j(k-1)\right)^2,$$

$$\beta_k^j = \beta_{k-1}^j + z_j(k-1). \quad (28)$$

The new data point influences the potential of the existing cluster centers, which also need to be updated as

$$P_k(z^{l*}) =$$

$$\frac{(k-1)P_{k-1}(z^{l*})}{k-2 + P_{k-1}(z^{l*}) + (k-1)P_{k-1}(z^{l*}) \sum_{j=1}^{n+1} \left(d_{k(k-1)}^j\right)^2}$$

$$(29)$$

where $P_k(z^{l*})$ is the potential at time k of the cluster center.

The potential of the new data point is compared with the updated potential of the existing cluster centers to determine whether to accept this new data point as a new cluster center according to the following criterion.

IF the new data point potential is higher than the potential of the existing centers AND the new data point is close to an old center:

$$P_k(z(k)) > \max_{i=1}^{N} \left\{P_k(z^{i*})\right\}$$

$$\frac{P_k(z(k))}{\max\limits_{l=1}^{N} P_k(z^{l*})} - \frac{\delta_{\min}}{r} \geq 1, \quad (30)$$

where $\delta_{\min}$ is the shortest distance between the new data point and all the existing cluster centers and is the radius of the neighborhood,

THEN the new data point replaces it.

ELSE IF the potential of the new data point is higher than the potential of the existing centers:

$$P_k(z(k)) > \max_{i=1}^{N} \left\{P_k(z^{i*})\right\}, \quad (31)$$

*THEN it is **added** to the rule-base as a new rule center, N=N+1.*

ENDIF.

The above approach is used for adding on-line new rules or replacing existing ones. However, there still remain some problems when applying this on-line approach to closed-loop control.

1. Choice of r

In off-line identification, it is easy to know the data range and normalize the data. In this case, the value of r is usually chosen as $r \in [0.3, 0.5]$. In on-line training, the data are collected continuously in time, so that we may not know the range of all the data in advance. A possible solution is to collect some off-line data to estimate the range of the data for on-line learning. Then it is updated on-line with more data coming. In this case, the data are not normalized and the value of is calculated as

$$r = roi \cdot \left\| \max_{m=1}^{k}\{z(m)\} - \min_{m=1}^{k}\{z(m)\} \right\|, \quad (32)$$

where

$$\max_{m=1}^{k}\{z(m)\} \triangleq \left[\max_{m=1}^{k}\{x_1(m-1)\}, \ldots, \max_{m=1}^{k}\{y(m)\}\right]$$

and $\min\limits_{m=1}^{k}\{z(m)\}$ is defined similarly. Here roi is chosen in the interval $[0.3, 0.5]$. Hence, as new data come, r is adapted rather than fixed.

2. Completeness of the rule base

There is no stability problem in open-loop identification while stability is an important issue in closed-loop control. In indirect control, where the controller is designed based on the model, the quality of the model is essential. For a fuzzy model, the completeness of its rule base is critical for its successful application to designing the model

for a nonlinear plant and for the effectiveness of the corresponding controller designed based on this model. The criterion (30)–(31) gives the condition for replacing an old rule or generating a new rule; however, it does not guarantee the completeness of the rule base. In open-loop identification, the identification procedure can go on with an incomplete rule base, while in closed-loop systems, it may cause a breakdown of the whole system. Therefore, we add another criterion called $\in$-*completeness of fuzzy rules* for generating new rules, that is, for any input within the operating region, there exists at least one rule such that the matching degree (or firing strength) is no less than $\in$. So except for the criterion (30)–(31), the following criterion is also used for rule generation:

IF $\omega^i(\cdot) < \in$ *, THEN a new cluster center is **added***.

The minimum value of $\in$ is usually selected as $\in_{\min} = 0.5$.

3. Compactness of the rule base

Since the range of data is not fixed but adapted with the growth of the data, the cluster centers that seem not close at the starting stage may become close to each other when the range of the data is enlarged. Therefore, during the procedure of rule base evolution, some redundant rules need to be deleted. However, the criterion (30)–(31) does not give the condition when to delete a rule. To provide a more compact rule base, we use the following criterion to *delete a rule*:

IF $d_k^{ij}/d_k^{\max} < \eta$ *AND* $P_k(z_i^*) < P_k(z_j^*)$, *THEN* z_i^* *is **deleted** from the group of cluster centers, and so is its corresponding rule*,

where d_k^{ij} is the distance between two cluster centers z_i^* and z_j^* at time k, $d_k^{\max}$ is the maximum distance between any two cluster centers, and $0 < \eta \leq 1$ is a constant chosen by the designer.

4.2. On-line parameter adaptation. The parameters of the Gaussian membership functions (4) are updated as follows:

$$
\begin{aligned}
\hat{\sigma}_j^i(k) &= \frac{roi}{\sqrt{8}} \cdot \left(\max_{m=1}^{k} \{x_j(m)\} - \min_{m=1}^{k} \{x_j(m)\} \right) \\
\hat{c}_j^i(k) &= z_j^{i*}, \quad i = 1, \dots, N; \quad j = 1, \dots, n.
\end{aligned}
\tag{33}
$$

For a fixed number of rules and membership parameters, the estimation of consequent parameters can be transformed into a least squares problem by reducing the overall T-S model (5) to the following form:

$$
y(k+1) = \Psi_k^T \Theta, \tag{34}
$$

where $\Psi_k = \left[\bar{\omega}^1 x_e^T(k), \ \dots, \ \bar{\omega}^N x_e^T(k) \right]^T$, $x_e(k) = \left[1, \ x^T(k), \ u(k) \right]^T$, $\Theta = \left[\theta^{1T}, \theta^{2T}, \dots, \theta^{NT} \right]^T$, $\theta^i = \left[\theta_0^i, \dots, \theta_n^i, \theta_{n+1}^i \right]^T$.

Define the following objective function:

$$
J_G = \sum_{k=1}^{K} [y(k+1) - \Psi_k^T \Theta]^2, \tag{35}
$$

where K is the number of training data. The rule consequent parameter Θ can be estimated by the Recursive Least Squares Estimation (RLSE) algorithm (Angelov and Filev, 2004):

$$
\hat{\Theta}_{k+1} = \hat{\Theta}_k + C_{k+1} \Psi_k \big(y(k+1) - \Psi_k^T \hat{\Theta}_k \big), \tag{36}
$$

$$
C_{k+1} = C_k - \frac{C_k \Psi_k \Psi_k^T C_k}{1 + \Psi_k^T C_k \Psi_k}, \tag{37}
$$

where $\hat{\Theta}_0 = \left[\hat{\theta}_0^{1T}, \hat{\theta}_0^{2T}, \dots, \hat{\theta}_0^{NT} \right]^T$ and $C_0 = \Omega I$. Here Ω is a positive large number chosen by the designer. Large Ω can speed up parameter convergence, but too large a value may cause instability. It needs to be chosen properly during the parameter estimation.

The optimization problem with the objective function (35) is globally optimal, but it does not guarantee locally adequate behavior of local affine models that form the overall T-S fuzzy model. To obtain locally interpretable models, which means that local models of the T-S fuzzy model can represent local behavior of the plant in the operating regions defined by the rule premises, the following locally weighted objective function is used:

$$
J_{L,i}(\theta^i) = (y - X^T \theta^i)^T W^i (y - X^T \theta^i), \tag{38}
$$

where $\mathbf{y} = [y(1), y(2), \cdots, y(K)]^T$,

$$
\begin{aligned}
X &= \begin{bmatrix} x_e^T(1) \\ \vdots \\ x_e^T(K) \end{bmatrix}, \\
W^i &= \begin{bmatrix} \bar{\omega}^i(x(1)) & & 0 \\ & \ddots & \\ 0 & & \bar{\omega}^i(x(K)) \end{bmatrix}.
\end{aligned}
\tag{39}
$$

The solution θ_{i*} that solves the weighted least squares estimation problem expressed by the objective function (38) can be obtained by applying a weighted pseudo-inversion:

$$
\theta^{i*} = \left(X^T W^i X \right)^{-1} X^T W^i y, \quad i = 1, \cdots, N. \tag{40}
$$

Alternatively, a weighted RLSE (wRLSE) algorithm can be applied to estimate θ^{i*}:

$$
\begin{aligned}
\hat{\theta}_{k+1}^i &= \hat{\theta}_k^i + \frac{\bar{\omega}^i(x(k)) s_k^i x_e(k)[y(k+1) - x_e^T(k)\hat{\theta}_k^i]}{1 + \bar{\omega}^i(x(k)) x_e^T(k) s_k^i x_e(k)},
\end{aligned}
\tag{41}
$$

$$s^i_{k+1} = s^i_k - \frac{\bar{\omega}^i(x(k))s^i_k x_e^{\ T}(k)x_e(k)s^i_k}{1 + \bar{\omega}^i(x(k))x_e^T(k)s^i_k x_e(k)} \qquad (42)$$

with the initial conditions $\hat{\theta}^i_0 = \mathbf{0}$ and $s^i_0 = \Omega I, i = 1, 2, \ldots, N$.

When a rule is **deleted** from the rule base, its corresponding membership functions and parameters are deleted as well. When a new rule is **added** to the rule base, its membership parameters are calculated from (33) while its consequent parameters are set to zero. When a new rule **replaces** an old rule, its membership parameters are calculated from the new cluster center while its consequent parameters are inherited from the old one.

Remark 2. Since stability analysis with the supervisory controller in Section 3 does not depend on the number and parameters of fuzzy rules, it will guarantee closed-loop stability when the T-S fuzzy model is updated with the on-line self-structuring algorithm. However, the supervisory controller is a kind of high gain controller which may bring undesired transient phenomena and hinder tracking error convergence. So it can only be viewed as a safeguard strategy rather than a main controller and should be switched off when the closed-loop system remains bounded. The switching policy used in Section 3 is a kind of hard switch. However, as such, it may cause chattering phenomena around the error boundary. Here, we modify the hard switch policy into a soft switch policy by using a ridge function as follows:

$$\varsigma(k) = \begin{cases} 0, & \|e(k)\| < E_1, \\ \dfrac{1}{2} + \dfrac{1}{2}\sin\left[\dfrac{\pi}{E_2 - E_1} \right. & \\ \left.\left(\|e(k)\| - \dfrac{E_1 + E_2}{2}\right)\right], & E_1 \le \|e(k)\| \le E_2, \\ 1, & \|e(k)\| > E_2, \end{cases} \qquad (43)$$

where E_1 and E_2 define the error boundaries beyond which the supervisory controller is fully switched off or on. When $\|e(k)\|$ remains within $[E_1, E_2]$, the supervisory controller is softly switched on to a certain degree according to the value of the ridge function.

5. Simulation example

In this example, the proposed modeling and control algorithm is applied to control a pendulum, which is a relative-degree-two system described by

$$\ddot{q}(t) + \dot{q}(t) + \sin(q(t)) = u(t). \qquad (44)$$
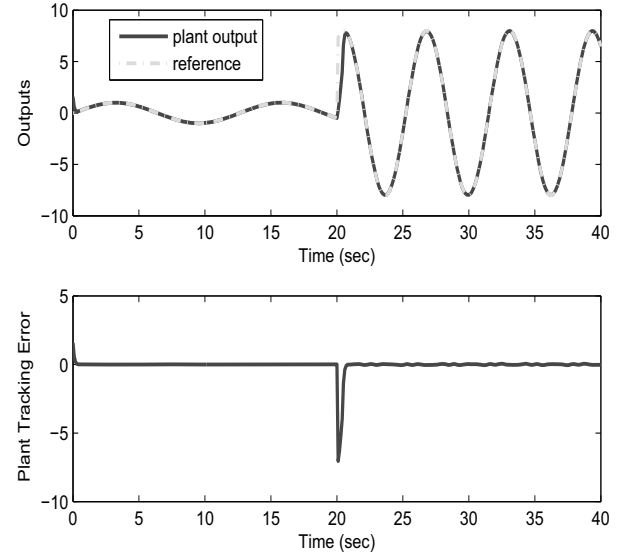
Fig. 2. Plant tracking response and the tracking error.
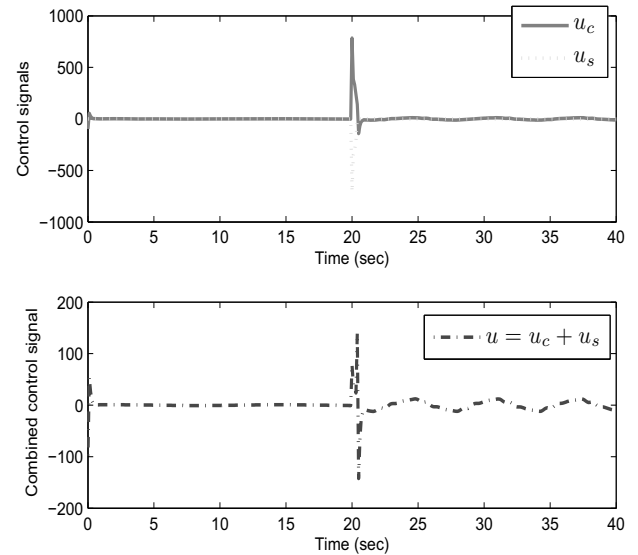
Fig. 3. Control inputs.

Defining the output $y(t) = q(t)$ and using Euler's rule, the above equation can be transformed into a discrete-time form

$$y(k+1) = (2 - T)y(k) + (T-1)y(k-1) \\ - T^2\sin(y(k-1)) + T^2 u(k-1), \qquad (45)$$

where T denotes the sampling interval.

To define the certainty equivalent control $u_c(k)$, the following transform is performed by using the method described in (Chen and Khalil, 1995). Firstly, we obtain

the following input-output relationship including $u(k)$ from (45):

$$y(k+2) = (2-T)y(k+1) + (T-1)y(k) \\ - T^2 \sin(y(k)) + T^2 u(k). \tag{46}$$

Replacing $y(k+1)$ in (46) by (45) yields

$$y(k+2) = f(y(k-1), y(k), u(k-1)) + gu(k), \tag{47}$$

where $g = T^2$,

$$f(y(k-1), y(k), u(k-1)) \\ = (2-T)[(2-T)y(k) + (T-1)y(k-1) \\ \quad - T^2 \sin(y(k-1)) + T^2 u(k-1)] \\ \quad + (T-1)y(k) - T^2 \sin(y(k)). \tag{48}$$

Now the function f depends on past inputs and outputs, and thus the control $u_c(k)$ can be defined in terms of f and g. Since f is unknown, a T-S fuzzy model $\hat{f}$ is employed to approximate it.

Initial conditions are chosen as $y(k-1) = y(k) = \pi/2$ and $u(k-1) = 0$. The sampling interval is $T = 0.1$ s. The desired trajectory is formed by two sine waves:

$$y_{\text{ref}}(t) = \begin{cases} \sin(0.5t), & 0 \le t < 20, \\ 8\sin(t), & t \ge 20. \end{cases} \tag{49}$$

The certainty equivalent control law gains in (14) are chosen as $k = [0.04, -0.4]^T$ while the boundary parameters for calculating the supervisory control signal $u_s(k)$ by (25) are selected as $E_1 = 5$, $E_2 = 50$,

$$P = \begin{bmatrix} 44.9206 & -8.8401 \\ -8.8401 & 85.7339 \end{bmatrix}. \tag{50}$$

To obtain the supervisory control signal, we have to determine the bounds $f^U(x)$, g_L and g^U. For this system, we get $g_L = g^U = T^2$,

$$|f(y(k-1), y(k), u(k-1))| \\ = |(T^2 - 3T + 3)y(k) + (2-T)(T-1)y(k-1) \\ \quad - (2-T)T^2 \sin(y(k-1)) - T^2 \sin(y(k)) \\ \quad + (2-T)T^2 u(k-1)| \\ \le |T^2 - 3T + 3||y(k)| + |(2-T)(T-1)||y(k-1)| \\ \quad + (3-T)T^2 + |(2-T)T^2||u(k-1)| \\ = f^U(x). \tag{51}$$

The term $\text{sgn}(e_p^T(k+1)Pb)$ used in (25) determines the direction of the supervisory controller. If the sampling interval T is chosen small enough, it is reasonable to obtain $\text{sgn}(e_p^T(k+1)Pb)$ by using $\text{sgn}(e^T(k)Pb)$.

The parameters for updating model are chosen as $roi = 0.5$, $\eta = 0.25$, $Q = 10^3$. The initial T-S model
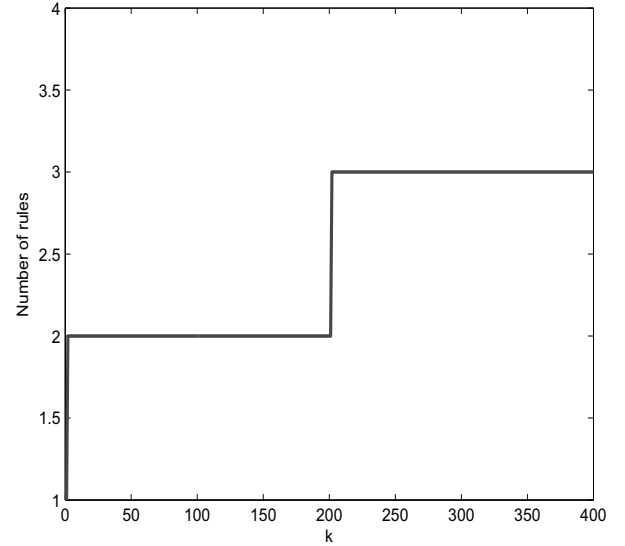
Fig. 4. On-line fuzzy rule evolution.
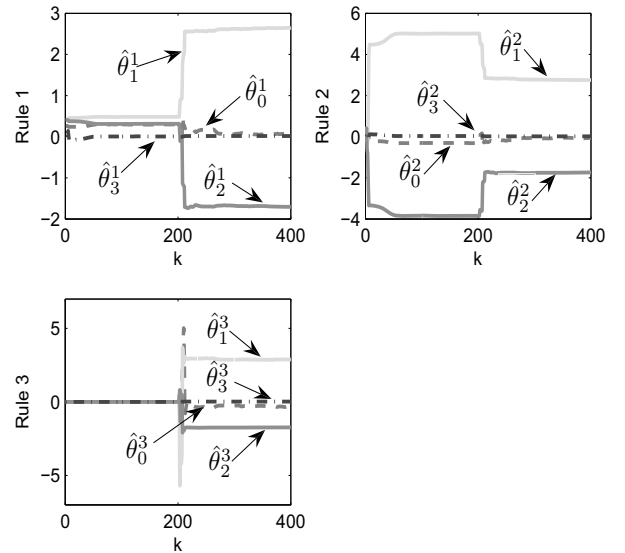
Fig. 5. Parameter adaptation by RLSE.

has only one rule with all its consequent parameters set to zero and its membership parameters chosen as $\hat{\sigma}_j^1(0) = 1$ and $\hat{c}_j^1(0) = 0$, $j = 1, 2, 3$.

Figure 2 shows the plant tracking response and the tracking error. The control inputs are illustrated in Fig. 3. During the initial 20 s, the number of rules evolves from 1 to 2. At $t = 20$ s, the reference signal has a sudden change, which leads to an increase in the values of control inputs in Fig. 3 and also indicates that the plant enters a certain new operating region. This new operat-

Fig. 6. Parameter adaptation by wRLSE.

ing region brings new information that is not covered by the current rules. Hence, a new rule is generated, which can be clearly seen in Fig. 4. The adaptations of consequent parameters by RLSE and wRLSE are shown in Figs. 5 and 6, respectively. Comparing those two figures, it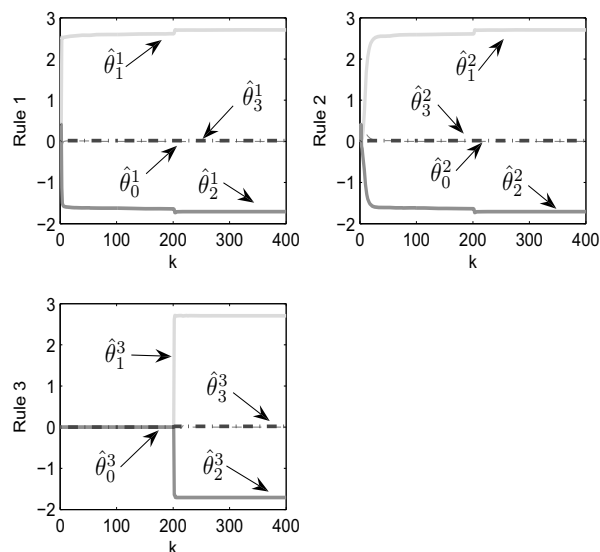 can be observed that the parameters of all the rules are adjusted to compensate for the error by standard RLSE in Fig. 5 while mainly the parameters of Rule 3 are adapted. The reason is that with standard RLSE data are used to adjust all the rules without distinguishing which operating region they belong to. On the contrary, with wRLSE data are weighted by the firing strength of each rule so that they contribute differently to the parameter adaptations of different rules. At 20 s, when the plant enters a new operating region which is governed by Rule 3, the coming data contribute most to the adjustment of the parameters of Rule 3 in Fig. 6.

6. Conclusion

This paper presented an indirect adaptive fuzzy control scheme for a class of SISO nonlinear systems. A T-S fuzzy model was employed as a dynamical model of nonlinear system dynamics. The T-S model is self-structuring in that both the number and the parameters of rules can be adjusted on-line. The number of fuzzy rules can be added, replaced or deleted automatically to provide a flexible structure and computational efficiency. The overall controller consists of a T-S fuzzy model based feedback linearization controller and a Lyapunov based supervisory controller. The supervisory controller is appended to the fuzzy controller to force the tracking error to be within

a bounded set. The stability of the system is established using the Lyapunov approach. Tracking control of a pendulum with relative-degree-two demonstrates the validity and effectiveness of the proposed on-line modeling and control scheme.

Discarding the assumption about the zero modeling error will be considered in the future research. In fact, in the simulation exercise, the modeling error exists at least due to the discretization error and the proposed controller has managed to overcome it. An obvious way to approach the problem is to apply a robust parameter estimation method. The controller ability to handle the modeling error would open a way to consider not affine in control plant dynamics by linearizing a nonlinear control dependent term in these dynamics.

References

Angelov, P. P. and Filev, D. P. (2004). An approach to online identification of Takagi-Sugeno fuzzy models, *IEEE Transactions on System, Man and Cybernetics—Part B: Cybernetics* **34**(1): 484–498.

Bezdek, J. (1974). Comparing different approaches to model error modeling in robust identification, *Journal of Cybernetics* **3**(3): 58–71.

Chen, F. and Khalil, H. (1995). Adaptive control of a class of nonlinear discrete-time systems using neural networks, *IEEE Transactions on Automatic Control* **40**(5): 791–801.

Chien, C.-J., C.-T. H. and Yao, C.-Y. (2004). Fuzzy system-based adaptive iterative learning control for nonlinear plants with initial state errors, *IEEE Transactions on Fuzzy Systems* **12**(5): 724–732.

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation, *International Journal of Fuzzy Systems* **2**: 267–278.

Gao, Y. and Er, M. J. (2003). Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems, *IEEE Transactions on Fuzzy Systems* **11**(4): 462–477.

Gustafson, D. E. and Kessel, W. C. (1979). Global random optimization by simultaneous perturbation stochastic approximation, *Proceedings of the IEEE Control Decision Conference, San Diego, CA, USA*, pp. 761–766.

Hao, Y. (1998). General SISO Takagi-Sugeno fuzzy system with linear rule consequent are universal approximators, *IEEE Transactions on Fuzzy Systems* **6**(4): 582–587.

Hao, Y., Y. D.-S. L. and Shao, S. (1999). Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators, *IEEE Transactions on System, Man and Cybernetics—Part B: Cybernetics* **29**(5): 508–514.

Ogata, K. (1995). *Discrete-time Control System, 2nd Ed.*, Prentice-Hall, Upper Saddle River, NJ.

Park, C.-W. and Cho, Y.-W. (2004). T-S model based indirect adaptive fuzzy control using online parameter estimation, *IEEE Transactions on System, Man and Cybernetics—Part B: Cybernetics* **34**(6): 2293–2302.

Phan, P. A. and Gale, T. J. (2008). Direct adaptive fuzzy control with a self-structuring algorithm, *Fuzzy Sets and Systems* **159**(8): 871–899.

Qi, R. and Brdys, M. A. (2008). Stable indirect adaptive control based on discrete-time T-S fuzzy model, *Fuzzy Sets and Systems* **159**(8): 900–925.

Wang, L. (1994). *Adaptive Fuzzy System and Control: Design and Stability Analysis*, Prentice-Hall, Englewood Cliffs, NJ.

Ruiyun Qi received the B.Eng. degree in automatic control from the University of Science and Technology of China, Hefei, China, in 2001, and the Ph.D. degree in electrical engineering from the University of Birmingham, UK, in 2007. She joined the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, in 2008. Currently she is an associate professor. Her research interests include fuzzy modeling, fuzzy control, and adaptive control.

Mietek A. Brdys received the M.Sc. degree in electronic engineering and the Ph.D. and D.Sc. degrees in control systems from the Institute of Automatic Control at the Warsaw University of Technology in 1970, 1974 and 1980, respectively. From 1974 to 1983, he held the posts of an assistant professor and an associate professor at the Warsaw University of Technology. In 1992 he became a full-rank professor of control systems in Poland. Between 1978 and 1995, he held various visiting faculty positions at the University of Minnesota, City University, De Montfort University, and the University Polytechnic of Catalunya. Since January 1989, he has held the post of a senior lecturer at the School of Electronic, Electrical and Computer Engineering of the University of Birmingham, where he heads the Control and Decision Support Research Laboratory. Since 2001 he has held the post of a full professor of control systems at the Department of Automatic Control of the Gdańsk University of Technology. There, in 2008 he founded the Department of Control Systems Engineering and became its head. He has served as a consultant for the Honeywell Systems and Research Centre in Minneapolis, GEC Marconi and Water Authorities in the UK, France, Spain, Germany and Poland. His research is supported by the UK and Polish Research Councils, industry and the European Commission. He is an author and co-author of about 220 peer-reviewed papers and six books. His current research includes intelligent decision support and control of complex uncertain systems, robust monitoring and control, softly switched robustly feasible model predictive control. The applications include environmental systems, technological processes, autonomous intelligent vehicles and defence systems. He is a Chartered Engineer, a member of the IEE, a Senior Member of the IEEE, a Fellow of the IMA and the chair of the IFAC Technical Committee on Large Scale Complex Systems. He was the IPC chair of the 11th IFAC Symposium on Large Scale Complex Systems (Gdańsk, 2007).

Appendix A

Proof. (Proof of Lemma 1). Choosing an $n \times 1$ vector $v = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T$. Since P is a positive definite symmetric matrix, we have

$$v^T P v > 0, \qquad (52)$$

Substituting $v = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T$ into (43) gives $p_{nn} > 0$. ∎

Appendix B

Proof. (Proof of Theorem 1). The following notation is used for analysis simplicity:

$$
\begin{aligned}
\alpha_1 &\triangleq \frac{1}{2} p_{nn} g^2(x), \\
\alpha_2 &\triangleq -\left(e^T(k)\Lambda^T P b + \phi(k)p_{nn} \right) g(x), \\
\alpha_3 &\triangleq \frac{1}{2} p_{nn}\phi^2(k) + e^T(k)\Lambda^T P b \phi(k).
\end{aligned} \qquad (53)
$$

Then $V_S(k)$ in (23) becomes

$$ V_s(k) = \alpha_1 u_s^2(k) + \alpha_2 u_s(k) + \alpha_3. \qquad (54) $$

Consider

$$
\begin{aligned}
\Delta &= \alpha_2^2 - 4\alpha_1\alpha_3 \\
&= g^2(x) \left(e^T(k) \mathrm{A}^T P b \right)^2 \geq 0.
\end{aligned} \qquad (55)
$$

Hence, the equation $V_S(k) = 0$ has two real roots:

$$ \lambda_1(k) = \frac{\phi(k)}{g(x)}, \qquad (56) $$

$$ \lambda_2(k) = \frac{2e^T(k)\mathrm{A}^T P b}{p_{nn}g(x)} + \frac{\phi(k)}{g(x)}. \qquad (57) $$

Define

$$
\begin{aligned}
\lambda_{\min}(k) &= \min\{\lambda_1(k), \lambda_2(k)\}, \\
\lambda_{\max}(k) &= \max\{\lambda_1(k), \lambda_2(k)\}.
\end{aligned}
$$

With Lemma 1 and Assumption 1, $p_{nn} > 0$ and $g^2(x) > 0$, we have $\alpha_1 > 0$. Then the necessary and sufficient condition for $V_S(k) \leq 0$ is

$$ \lambda_{\min}(k) \leq u_s(k) \leq \lambda_{\max}(k). \qquad (58) $$

Define

$$
\begin{aligned}
\Upsilon(k) = \frac{1}{g_L(x)}\big(&\left|f^U(x)\right| + \left|y_r(k+1)\right| + \left|k^T e(k)\right| \\
&+ g^U(x)\left|u_c(k)\right|\big).
\end{aligned} \qquad (59)
$$

With (14), (18) and (56), we have

$$
\begin{aligned}
\lambda_1(k) = \frac{1}{g(x)}\big(&-f(x) + y_r(k+1) - k^T e(k|\hat{p}) \\
&- g(x)u_c(k|\hat{p})\big).
\end{aligned} \qquad (60)
$$

Comparing (60) with (59) and with Assumption 1, we have

$$ -\Upsilon(k) \leq \lambda_1(k) \leq \Upsilon(k). \qquad (61) $$

Let us now consider $\lambda_2(k)$. With (17), the following holds:

$$e^T(k)A^T Pb$$
$$= [e(k+1) - b\phi(k) + bg(x)u_s(k)]^T Pb$$
$$= e^T(k+1)Pb - p_{nn}\phi(k) + p_{nn}g(x)u_s(k).$$
$$(62)$$

Substituting (62) into (57) yields

$$\lambda_2(k) = 2\frac{e^T(k+1)Pb}{p_{nn}g(x)} - \lambda_1(k) + 2u_s(k). \quad (63)$$

With (59) and Assumption 5, the supervisory controller (25) can be written as

$$u_s(k) = \varsigma\,\mathrm{sgn}(e^T(k+1)Pb)\Upsilon(k). \quad (64)$$

The following two cases are considered when $\varsigma = 1$.

1. If $\mathrm{sgn}(e^T(k+1)Pb) = 1$, then $u_s(k) = \Upsilon(k)$ in (64). With Assumption 1 and (61), the following relationship holds:

$$\lambda_1(k) \leq u_s(k)$$
$$\lambda_2(k) = 2\frac{e^T(k+1)Pb}{p_{nn}g(x)}$$
$$+ (u_s(k) - \lambda_1(k)) + u_s(k)$$
$$\geq u_s(k).$$
$$(65)$$

Hence, $u_s(k)$ satisfies $\lambda_1(k) \leq u_s(k) \leq \lambda_2(k)$.

2. If $\mathrm{sgn}(e^T(k+1)Pb) = -1$, then $u_s(k) = -\Upsilon(k)$ in (64). With Assumption 1 and (61), the following relationship holds:

$$\lambda_1(k) \geq u_s(k)$$
$$\lambda_2(k) = 2\frac{e^T(k+1)Pb}{p_{nn}g(x)} + (u_s(k) - \lambda_1(k)) + u_s(k)$$
$$\leq u_s(k).$$
$$(66)$$

Hence, $u_s(k)$ satisfies $\lambda_1(k) \leq u_s(k) \leq \lambda_2(k)$.

Therefore, with the supervisory controller $u_s(k)$ designed in (25), the inequality $\lambda_{\min}(k) \leq u_s(k) \leq \lambda_{\max}(k)$ always holds and is implied. Hence, from (24) we have

$$\Delta V(e(k)) \leq -\frac{1}{2}e^T(k)Qe(k) + V_S(k) \leq 0, \quad (67)$$

and the boundedness of $e(k)$ has now been proven. Since the reference trajectory is bounded with Assumption 2, $x(k)$ is bounded. With bounded $e(k)$, $u(k)$ are bounded as well. Hence all the closed-loop signals are bounded. The proof of Theorem 1 has now been completed. ∎