Methoden

Goran Andonovski*, Bruno Sielly Jales Costa, Sašo Blažič and Igor Škrjanc

Robust evolving controller for simulated surge tank and for real two-tank plant

Robuster Evolutionsregler für einen simulierten Ausgleichsbehälter und für eine reelle Zweitankanlage

https://doi.org/10.1515/auto-2018-0024
Received March 2, 2018; accepted June 26, 2018

Abstract: This paper presents a robust evolving cloud-based controller named RECCo. The controller has an evolving fuzzy structure and the rules are represented by data clouds. The evolving part of the algorithm allows adding of new rules (clouds) and moreover, the robust adaptive law using the steepest (gradient) descent method adapts the PID-R parameters of each cloud. There are also some protective mechanisms introduced which improve the robustness of the algorithm. The effectiveness of the controller was tested on the simulated surge tank model and on the real two tank plant. Both plants have quite similar structure but they have different nonlinear dynamics. Using the same initializing procedure the RECCo controller efficiently control both plants.

Keywords: robust control, evolving fuzzy system, adaptive controller, surge tank, two-tank plant

Zusammenfassung: In dieser Veröffentlichung wird ein robuster Evolutions- Cloud-basierter Regler mit dem Namen RECCo vorgestellt. Der Regler hat eine entwickelnde Fuzzy-Struktur und die Regeln werden durch Datenclouds dargestellt. Der entwickelnde Teil des Algorithmus erlaubt das Hinzufügen neuer Regeln (Wolken) und darüber hinaus passt das robuste adaptive Gesetz, das die Methode des steilsten (Gradienten-) Abstiegs anwendet, die PID-R-Parameter jeder Wolke an. Es werden auch einige Schutzmechanismen eingeführt, die die Robustheit des Algorithmus verbessern. Die Wirksamkeit des Reglers wurde am simulierten Ausgleichsbehältermodell und an einer realen Zweitankanlage getestet. Beide Anlagen haben eine sehr ähnliche Struktur, aber eine unterschiedliche nichtlineare Dynamik. Bei einem gleichen Initialisierungsverfahren steuert der RECCo Regler beide Anlagen effizient.

Schlagwörter: Robuste Regelung, Evolutions-Fuzzy-System, Adaptiver Regler, Ausgleichsbehälter, Zwei-Tank-Anlage

1 Introduction

The concept of evolving fuzzy rule-based (FRB) systems and its application to control problems has gained significant attention and contribution since it was introduced in [1, 2]. One can mention the work of the authors presented in [3] and [4], which focus on the control of twin-rotor systems and [5] with control of a prosthetic hand. Moreover, [6] also have proposed an evolving FRB approach to control of non-linear dynamic systems and, more recently, to control and navigation of quad-copters [7]. The authors in [8–12] presented different types of evolving controllers which use an eFuMo [13] method for partitioning the nonlinear space. In [14] was presented a new approach to design of experiments (DoE), based on an evolving fuzzy model structure. There are also some methods that tackle the problem of the optimal tuning of the membership functions of fuzzy PID controller [15–17].

In this paper a robust evolving cloud-based controller (RECCo) is used for controlling the real two-tank pilot plant. The RECCo is a fuzzy type of controller which uses data clouds [18] to define the membership functions (fuzzy rules) in the antecedent part. The consequent part consists of adaptive local PID-R (Proportional Integral Derivative with Compensation of the operating point) controllers for each rule with additional compensation of the operating point.

***Corresponding author: Goran Andonovski,** Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia, e-mail: goran.andonovski@fe.uni-lj.si
Bruno Sielly Jales Costa, Campus Natal – Zona Norte, Federal Institute of Rio Grande do Norte, Natal, Brazil; and Research and Innovation Center, Ford Motor Company, Palo Alto, CA, USA, e-mail: bruno.costa@ifrn.edu.br
Sašo Blažič, Igor Škrjanc, Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia, e-mails: saso.blazic@fe.uni-lj.si, igor.skrjanc@fe.uni-lj.si

The fuzzy structure of the controller is not fixed, but evolves during the control of the plant. According to the maximal local density, the current data point is associated with one of the existing clouds. If this density is lower than the defined threshold a new cloud (rule) is added (evolving part). Furthermore, the PID parameters of the local controllers are updated using the stable Lyapunov approach of steepest descent. There were already some attempts proposed of using RECCo algorithm for controlling real and simulated processes [19–21].

The effectiveness of the proposed control algorithm was tested on a simulated surge tank and on a real two-tank pilot plant. The RECCo controller was set with default values of the design parameters and it was initialized with the first data point received. Furthermore, the PID-R parameters for the first local controller were set to zero and were adapted in an online manner.

The real two tank plant used for validation in this work has been object of study of different authors. Since it has multiple real-world and industrial-grade sensors, actuators and controllers, we have to handle different factors from real-world applications that are very difficult to simulate, such as noise, process and inertia. A few authors have recently proposed different control approaches to the plant. A self-evolving cloud-based controller was proposed by [19]. The work focus on an AnYa architecture, which introduces the concept of data clouds as granular structures (rather than traditional clusters). Previously, [22] had proposed a multi-stage hierarchical fuzzy controller for the referred pilot plant, that requires minimal manual parameterization and it is able to cope with the plant non-linearities. Furthermore, many studies on the field of fault detection and diagnosis were performed on the pilot plant. [23] proposes real-time fault detection based on the concept of recursive density estimation as a metric for detection of multi-signal anomalies. The concept was later extended to encompass fault classification [24, 25]. Fault detection was again assessed in [26] with the concept of typicality and eccentricity data analytics, where abnormal (eccentric) data samples are automatically detected. Finally, a self-evolving classification method based on the eccentricity metric was introduced in [27] with very promising results.

This paper is organized as follows. In Section 2 the RECCo evolving adaptive controller is presented. Next, in Section 3 two plans are explained, the surge tank (Subsection 3.1) and the real two-tanks pilot plant (Subsection 3.2). The experimental results of both plants and the efficiency of the algorithm are presented in Section 4. Finally, the conclusions are drawn in Section 5.

2 Evolving adaptive controller RECCo

In this section the robust evolving cloud-based controller (RECCo) will be described. Theoretically, the controller could be initialized from the first data sample received. But of course, any existing information about the controlled process can be used to suitably initialize the design parameters. After the phase of initialization finishes, the structure evolves with every received data sample (when some criteria are satisfied). Furthermore, the controller's parameters are adapted in online manner.

The RECCo is a type of ANYA fuzzy rule-based (FRB) system with non-parametric (cloud-based) antecedents [18]. This controller applies the concept of local density in normalized problem space to define the membership of the current data to the existing clouds. Incoming data samples are analyzed in an online manner and each sample is associated with all of the clouds with a certain membership but only the parameters of the nearest cloud (maximal density) are updated.

The structure of the RECCo controller is based on the ANYA FRB system and has the following form:

$$\mathcal{R}^i : \text{IF} \quad (\mathbf{x}_k \sim X^i) \quad \text{THEN} \quad (u^i) \tag{1}$$

where the number of rules $\mathcal{R}^i$ is equal to the number of existing clouds c, which changes during controlling the process (evolving part). The fuzzy antecedent part is defined with the operator ~ which could be linguistically expressed as '*is associated with*'. That means that the current data $\mathbf{x}_k = [x_1, x_2, \ldots, x_n]^T$ is related to the i-th cloud $X^i \in \mathbb{R}^n$ with a certain membership. The consequent part consists c local controllers (u^i), one for each rule. To calculate the final control action the weighted average method is used (defuzzification).

The degree of association between the data sample $\mathbf{x}_k$ and corresponding cloud X^i is measured by the normalized relative density as follows:

$$\lambda_k^i = \frac{\gamma_k^i}{\sum_{j=1}^{c} \gamma_k^j} \quad i = 1, \ldots, c \tag{2}$$

where γ_k^i is the local density of the i-th cloud for the current data $\mathbf{x}_k$. The local density takes in consideration all the previous data samples associated with a certain cloud and is calculated as follows:

$$\gamma_k^i = \frac{1}{1 + \frac{\sum_{j=1}^{M^i} \|\mathbf{x}_k - \mathbf{x}_j^i\|^2}{M^i}} \tag{3}$$

where M^i is the number of points associated with the i-th cloud. In this case the Euclidean distances between the new data $\mathbf{x}_k$ and all data points of the i-th cloud is used. But again, any other type of distance measure could also be used (e. g., Mahalanobis in [28–30]). From the practical point view and for easier implementation, local density (3) can be recursively rewritten as follows:

$$\gamma_k^i = \frac{1}{1 + \|\mathbf{x}_k - \mu_k^i\|^2 + \sigma_k^i - \|\mu_k^i\|^2} \tag{4}$$

where μ_k^i is the mean value and σ_k^i is the mean-square length of the data points in the i-th cloud. Both of them can be recursively calculated as follows:

$$\mu_k^i = \frac{M^i - 1}{M^i}\mu_{k-1}^i + \frac{1}{M^i}\mathbf{x}_k \tag{5}$$

$$\sigma_k^i = \frac{M^i - 1}{M^i}\sigma_{k-1}^i + \frac{1}{M^i}\|\mathbf{x}_k\|^2 \tag{6}$$

The initial conditions when a new cloud is added ($M^i = 1$) are $\mu_1^i = \mathbf{x}_1$ and $\sigma_1^i = \|\mathbf{x}_1\|^2$ for the mean value and mean-square length, respectively.

The evolving law in this paper consists only a mechanism for adding new clouds (rules). We decide to use just adding mechanism due to simplicity of the implementation and because it is sufficient for control the plants presented in Section 3. The adding mechanism relies on the local density γ_k^i of the current data sample with the existing clouds. Once a new data sample arrive we need to calculate c different local densities between the sample and all the existing clouds. According to the maximal local density ($\max_i \gamma_k^i$) the data sample is associated with that cloud and the properties are updated (equations (5) and (6)). But, if the maximal local density ($\max_i \gamma_k^i$) is lower than the threshold value γ_{max} a new cloud is added. Due to the problem space normalization [31] the threshold can be fixed $\gamma_{max} = 0.93$. Beside this some other criteria need to be fulfilled before adding a new cloud (e. g., certain time n_{add} has passed from the last change).

The reference model defines the desired trajectory y^r that the controlled variable y should follow. Suggestions for selecting the structure of reference model are less or equal order of the plant [32]. Moreover, the time constants have to be similar (usually slightly shorter) to the dominant time constant of uncontrolled process. According to the controlled plant we define simple first order linear reference-model as:

$$y_{k+1}^r = a_r y_k^r + (1 - a_r) r_k \quad 0 < a_r < 1 \tag{7}$$

where r_k is the reference signal the parameter a_r can be approximated by $(1 - \frac{T_s}{\tau})$, where T_s is the sampling period of the process and τ is the time constant which is slightly shorter than the estimated time constant of the controlled plant. The goal of the controller, is to provide efficient performance and to ensure that the tracking error is as small as possible. The tracking error represents the deviation of the plant output from the desire trajectory and is defined as follows:

$$\varepsilon_k = y_k^r - y_k \tag{8}$$

Once we defined reference model and the tracking error, we can define the data point x_k in a normalized data space as follows [20]:

$$\mathbf{x}_k = \left[\begin{array}{cc} \frac{\varepsilon_k}{\Delta\varepsilon}, & \frac{y_k^r - r_{min}}{\Delta r} \end{array}\right]^T = [\varepsilon_{k,norm}\ y_{k,norm}^r]^T \tag{9}$$

where $\Delta r = r_{max} - r_{min}$ and $\Delta\varepsilon = \frac{\Delta r}{2}$.

For the consequent part of the RECCo algorithm the PID-R controller is used [21] with the following form:

$$u_k^i = P_k^i \varepsilon_k + I_k^i \Sigma_k^\varepsilon + D_k^i \Delta_k^\varepsilon + R_k^i, \quad i = 1, \ldots, c \tag{10}$$

where P_k^i, I_k^i, D_k^i are controller gains while R_k^i is compensation of the operating point. Σ_k^ε and Δ_k^ε in (10) are discrete-time integral and derivative of the tracking error, respectively, and can be calculated as follows:

$$\Sigma_k^\varepsilon = \begin{cases} \Sigma_{k-1}^\varepsilon + \varepsilon_k, & u_{min} < u(k) < u_{max} \\ \Sigma_{k-1}^\varepsilon, & u(k) = u_{min} \textbf{ or } u(k) = u_{max} \end{cases} \tag{11}$$

$$\Delta_k^\varepsilon = \varepsilon_k - \varepsilon_{k-1} \tag{12}$$

As we said above, in this paper we introduce an anti-windup mechanism for protecting the integral explosion.

The vector of the PID-R parameters is denoted as $\theta_k^i = \left[P_k^i, I_k^i, D_k^i, R_k^i\right]^T$. The vector of the first cloud is initialized with zeros $\theta_0^1 = [0, 0, 0, 0]^T$, while all later added clouds are initialized with mean value of the parameter's vector of all previous clouds as follows:

$$\theta_0^c = \frac{1}{c-1}\sum_{j=1}^{c-1}\theta_k^j \tag{13}$$

where c is the index of the newly added cloud.

The adaptation of the PID-R parameters is made in an online manner and only the parameters of the cloud with maximal local density are updated while others are kept constant:

$$\theta_k^i = \theta_{k-1}^i + \Delta\theta_k^i \tag{14}$$

and the adaptation of the PID parameters was introduced in [21], but in this article we proposed an improved version as follows:

$$\begin{aligned}\Delta P_k^i &= \alpha_P\, G_{sign}\lambda_k^i \frac{|e_k\varepsilon_k|}{1+r_k^2}\\ \Delta I_k^i &= \alpha_I\, G_{sign}\lambda_k^i \frac{|e_k\Delta_k^\varepsilon|}{1+r_k^2}\\ \Delta D_k^i &= \alpha_D\, G_{sign}\lambda_k^i \frac{|e_k\Delta_k^\varepsilon|}{1+r_k^2}\\ \Delta R_k^i &= \alpha_R\, G_{sign}\lambda_k^i \frac{\varepsilon_k}{1+r_k^2}\end{aligned} \tag{15}$$

where $\alpha_P, \alpha_I, \alpha_D, \alpha_R$ are the adaptation gains of the controller parameters, $G_{sign} = \pm 1$ is the known sign of the process gain, $e_k = r_k - y_k$ is the control error. The default value of the adaptation gains is 0.1 and is used when the range of the control variable is ($u_{min} = 0/4, u_{max} = 20$). When the range is different, the value of the parameters is rescaled as follows:

$$\alpha_{new} = \frac{u_{max} - u_{min}}{20} \cdot 0.1 \tag{16}$$

For example, if the range is from $u_{min} = 0$ to $u_{max} = 100$ the new value of the adaptive gains will be $\alpha_{new} = 0.5$.

The absolute values in (15) are used only in the starting phase of the control performance (five time constants is enough) and after that they are omitted from the adaptation law. Please refer to [33] for more details about the absolute values in (15).

Finally, for the defuzzification the weighted average is used (but not limited to this form) and therefore, the control variable becomes:

$$u_k = \sum_{i=1}^{c} \lambda_k^i u^i = \frac{\sum_{i=1}^{c} \gamma_k^i u^i}{\sum_{i=1}^{c} \gamma_k^i} \tag{17}$$

where u^i denotes the contribution of the i-th local controller.

In the following four mechanisms for improving the robustness and to minimize the negative influence of disturbances (parasitic dynamics) will be introduced:

2.1 Dead zone in the adaptation law

To improve the robustness under the unknown bounded disturbances and modeling errors, the RECCo controller includes a dead-zone in adaptation law. The general idea behind the dead-zone mechanism, in case of bounded disturbances, is to turn off the adaptation algorithm when the absolute value of the tracking error is smaller than a certain threshold [34]:

$$\Delta\bar{\theta}_k^i = \begin{cases} \Delta\theta_k^i & |\varepsilon_k| \geq d_{dead} \\ 0 & |\varepsilon_k| < d_{dead} \end{cases} \quad i = 1, \ldots, c \tag{18}$$

The parameter d_{dead} should be chosen slightly larger than the process noise to improve the effectiveness of the adaptive law. A larger threshold implies a shorter adaptation period and larger tracking error, while smaller value can lead to parameter drift.

2.2 Parameter projection

Parameter projection mechanism is used to guarantee that the estimation of the parameters will stay within finite known region [35]. In the case of the positive plant gain all the parameters should be bounded by 0 from bellow while upper bound may or may not be provided. The adaptive law in (14) is generalized as follows:

$$\theta_k^i = \begin{cases} \theta_{k-1}^i + \Delta\theta_k^i & \underline{\theta} \leq \theta_{k-1}^i + \Delta\theta_k^i \leq \overline{\theta} \\ \underline{\theta} & \theta_{k-1}^i + \Delta\theta_k^i < \underline{\theta} \\ \overline{\theta} & \theta_{k-1}^i + \Delta\theta_k^i > \overline{\theta} \end{cases} \quad i = 1, \ldots, c \tag{19}$$

In our case we chose $\underline{\theta} = 0$ and $\overline{\theta} = \infty$ for the controller gains P_k, I_k, and D_k, while for the compensation of the operating point R_k the lower bound was $\underline{\theta} = -\infty$.

2.3 Leakage in the adaptation law

The use of leakage in the adaptation law is a very known approach for improvement of robustness of adaptive control [36].

Including the leakage in the adaptation law results in:

$$\theta_k^i = (1 - \sigma_L)\theta_{k-1}^i + \Delta\theta_k^i \quad i = 1, \ldots, c \tag{20}$$

where σ_L defines the extent of the leakage. The value of leakage used in this paper is $\sigma_L = 10^{-6}$.

2.4 Interruption of adaptation

In the RECCo algorithm we first calculate the adaptation of the PID parameters ($\Delta\theta_k^i$) and then the control variable u_k. In some cases this two steps can be in conflict, which means that the adaptation causes control signal which is outside the limits $[u_{min}, u_{max}]$. In such case the adaptive law should be interrupted in the following manner:

$$\Delta\bar{\theta}_k^i = \begin{cases} \Delta\theta_k^i & u_{min} \leq u_k \leq u_{max} \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, c \tag{21}$$

3 Problem description

In this section two experiments are presented. First experiment is a simulated surge tank [37] and the second one is a real plant of two tanks developed by DeLorenzo [38]. In both cases the main goal is to control the liquid level in the tank manipulating the input liquid flow.

3.1 Simulation example: Surge tank

The simulation example of surge tank, presented in [37], can be described with the following differential equation:

$$\frac{dy(t)}{dt} = \frac{-c\sqrt{2gy(t)}}{A(y(t))} + \frac{1}{A(y(t))}u(t) \tag{22}$$

where $y(t)$ is the liquid level in meters ($0 \div 10$ m); $u(t)$ is the input flow (control signal), which can be positive or negative in the range $[u_{min}, u_{max}] = [-50, 50]$; $g = 9.8\,\mathrm{m/sec}^2$ is gravity; $c = 1$ is the known cross-sectional area of the output pipe; and $A(y(t)) = ay(t)^2 + b$ is cross-sectional area of the tank;

As suggested in [37] we choose $a = 1$ and $b = 2$ as the nominal plant parameters. After the discretization and using an Euler approximation the differential equation (22) can be rewritten:

$$y_{k+1} = y_k + T\left[\frac{-\sqrt{19.6y_k}}{y_k^2+2} + \frac{1}{y_k^2+2}u_k\right] \tag{23}$$

where the sampling time is $T = 0.1$. According to [39] the time constant of the process was estimated to $\tau = 10$ s.

The control objective of the surge tank is to control the liquid level in the tank ($y(t) = h(t)$) by manipulating the input liquid flow $u(t)$.

3.2 Real example: Two-tank pilot plant

The plant used for testing the proposed algorithm is a two tank pilot plant for industrial control, developed by DeLorenzo [38]. The plant (see Fig. 2a) is physically located in the Automation Laboratory of the Federal Institute of Rio Grande do Norte – Campus Natal/Zona Norte, Brazil. The plant allows various experiments on continuous processes based on four typical variables – pressure, temperature, flow and level. The communication between the controller and the plant is granted by an OPC interface [40].

The pilot plant includes: sensors for temperature, pressure, flow and level; indicators that convert physical into electrical signals, to be processed by a PLC; one terminal bus, where all existing electrical signals are made available for the controller; a SCADA software for parameter configuration and plant monitoring; a physical panel with a PLC and all electrical components for plant control; two pressurized vessels: the top one made of acrylic, T_1, and the bottom one, made of stainless steel, T_2; two directional valves, V_1 and V_2; a centrifugal pump for recirculation, controlled by a frequency inverter; a heater and heat exchange system. Fig. 2b illustrates the didactic scheme of the pilot plant.

As easily seen in Fig. 2b, the tanks are connected through a piping system and enable one directional water flow and liquid transfer in both tanks. The transfer from T_1 to T_2 is made by gravity, while the transfer from T_2 to T_1 is made by pressure generated by the centrifugal pump. The valves V_1 and V_2 are used to control the flow in each part of the process. For this study, temperature was not assessed.

The control objective of the two tank plant is to control the water level in tank T_1 ($y_k = L_{T1}$) by manipulating the pump speed ($u_k = V_{pump}$). The valves V_1 and V_2 are opened to 100 %.

4 Results

In this section the practical results of the RECCo controller for simulated surge tank and for real two-tank pilot plant are presented. The reference signal r_k (7) is chosen to cover the majority of the process's range ($[y_{min}, y_{max}]$) to see the ability of learning in different operating points. The advantage of the proposed algorithm is that it requires basic knowledge of the controlled process (input and output range, time constant and sampling time). The Table 1 shows all the design parameters for both processes. First

Table 1: Design (control) parameters for controlled plants.

	Parameter	Surge Tank	Two-Tank Plant
Plant properties	u_{min}	-50	0
	u_{max}	50	100
	y_{min}	0	0
	y_{max}	10	100
	τ	10	40
	T_s	0.1	1
Evolving	$a_r \approx 1 - T_s/\tau$	0.99	0.975
	γ_{max}	0.93	0.93
	c_{max}	100	100
	n_{add}	20	20
Adaptation	α_{new} (15), (16)	0.5	0.5
	d_{dead} (18)	0.5	0.5
	$[\underline{\theta}, \overline{\theta}]$ (19)	$[0, \infty]$	$[0, \infty]$
	σ_L (20)	10^{-6}	10^{-6}

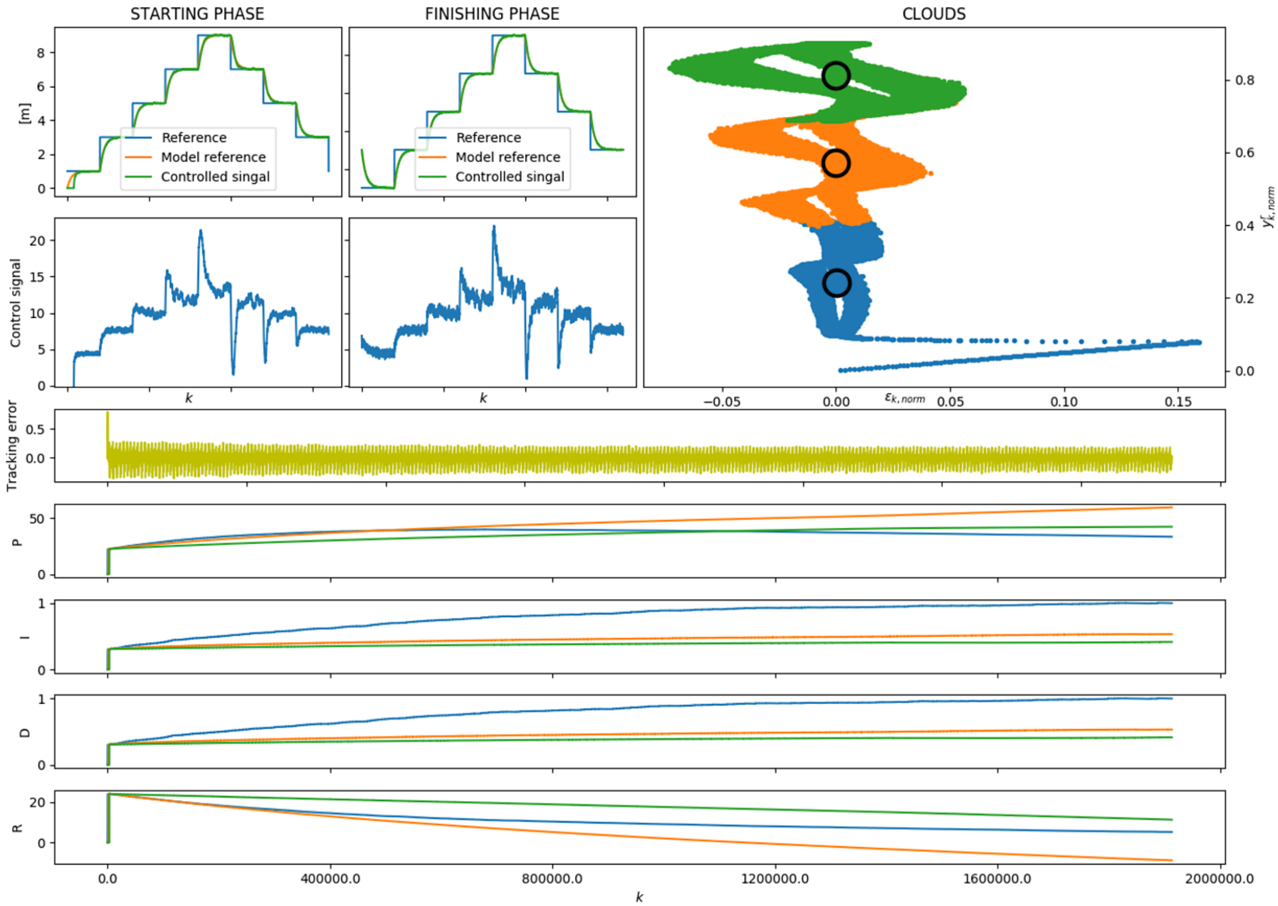


Figure 1: Simulation results of the surge tank.

group of parameters are the *plant's properties*, second group are the *evolving* and the third group are the parameters of the *adaptive law*.

We should to note that the controller starts with empty set of rules (1) and the first rule is generated (initialized) with the first data point received. Also the controller's gains (10) are initialized with zeros at the beginning of the experiment.

4.1 Simulation results of surge tank model

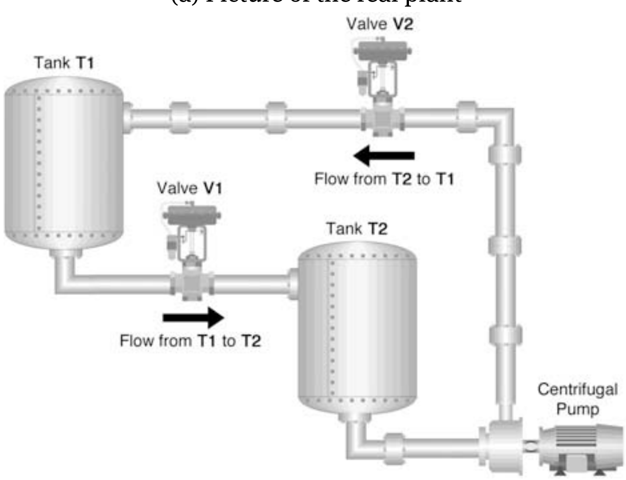
Using the parameters presented in Table 1 the simulation results of surge tank were provided which are presented in Fig. 1. The reference, model reference, controlled and control signals are shown for the starting and the finishing phase of the experiment (see first four plots on left top corner in Fig. 1). On the top right corner the clouds are presented. In this case the evolving algorithm added three clouds. The tracking error is shown for the length of whole experiment. We can notice slight and stable decreasing of the tracking error. The last four plots in Fig. 1 show the adaptive parameters (P, I, D, and R). Again, we can notice that the parameters converge through time.

4.2 Experimental results of the real two-tank plant

As we said above the control objective in case of two tank plant is to control the water level in tank T_1 ($y_k = L_{T1}$) by manipulating the pump speed ($u_k = V_{pump}$). The valves V_1 and V_2 are opened at 100 %. In this case we chose the length of the experiment approximately to 140 h (251 repetition of the chosen reference signal).

The reference $r(t)$, the model-reference $y^r(t)$ and the controlled variable for the starting phase are presented in Fig. 3 (top plot). The control signal $u(t)$ is presented in bottom plot in the same figure. We can notice that the controlled variable $y(t)$ relatively fast reaches the reference value but do not follow the model-reference signal. After the transient period of learning (evolving the structure and adapting the parameters) the controller more efficiently control the plant. The finishing phase of the learning is

(a) Picture of the real plant



(b) A didactic scheme of the plant

Figure 2: Pictures of Two-Tank Plant.

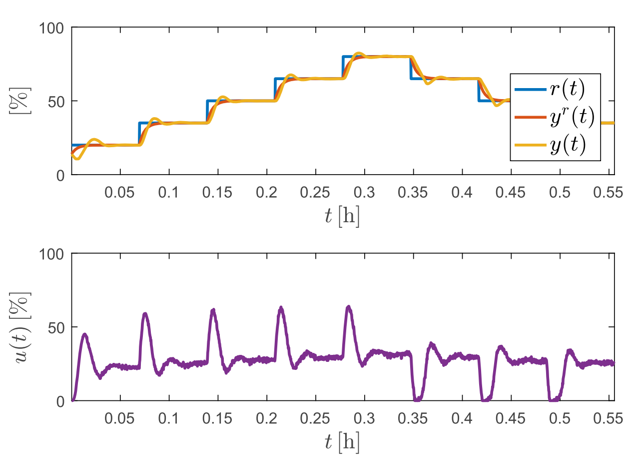


Figure 3: Starting phase. The reference $r(t)$, the model reference $y^r(t)$, and the controlled variable $y(t)$ (top plot) and the control signal $u(t)$ (bottom plot).

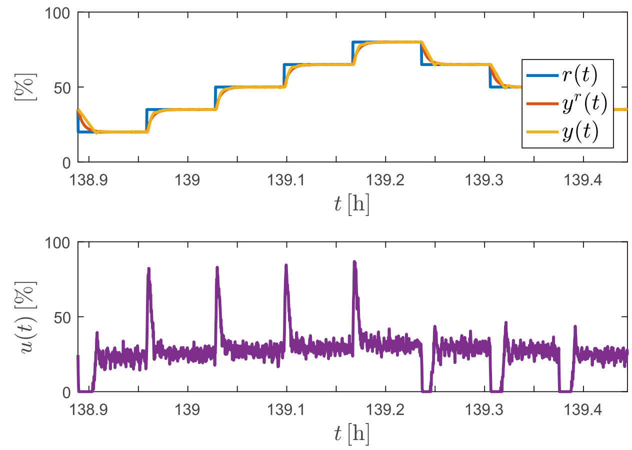


Figure 4: Finishing phase. The reference $r(t)$, the model reference $y^r(t)$, and the controlled variable $y(t)$ (top plot) and the control signal $u(t)$ (bottom plot).

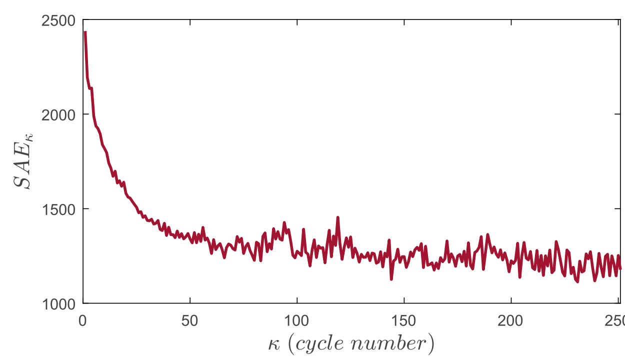


Figure 5: The sum of absolute error *SAE* for each cycle.

presented in Fig. 4, where the reference $r(t)$, the model-reference $y^r(t)$ and the controlled variable are shown.

The sequence presented in Fig. 3 is just one cycle of the learning process. This cycle was repeated 250 times to gain enough knowledge for all operational points. In Fig. 5 values of SAE_κ for each cycle are shown. We can see that during the learning process the sum of absolute tracking error decreases.

During the learning phase 4 clouds were created (see Fig. 6). For each data cloud the adaptive PID-R parameters are shown in Fig. 7.

5 Conclusion

The main purpose of this paper was to show the ability of learning of the RECCo controller. Using the same initial parameters, the RECCo algorithm is capable of controlling

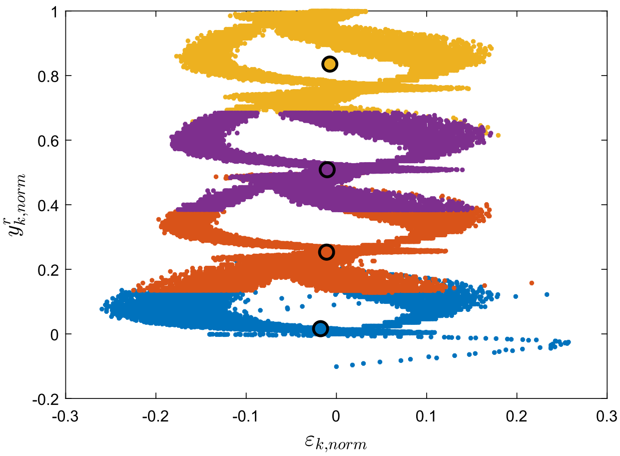


Figure 6: Data clouds (fuzzy rules) created.

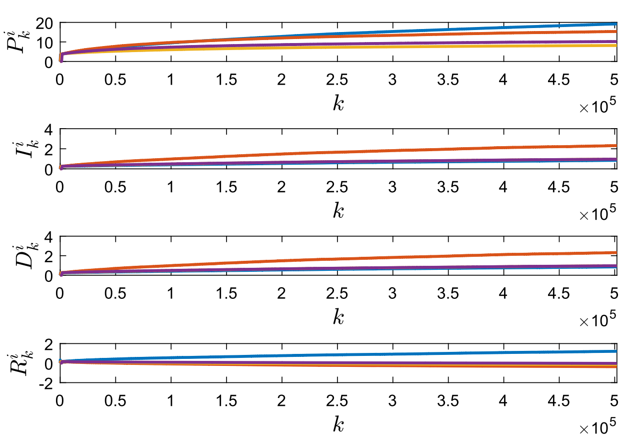


Figure 7: The process of adaptation of the PID-R parameters for each data cloud.

processes with different dynamics. Only the basic knowledge of the controlled process is required (i. e., input and output range, estimated time constant and the sampling time). The performance of the algorithm was tested on simulated surge tank model and on the real two-tank pilot plant. In both cases the algorithm shows that the control efficiency improves through time and the parameters converges. The protective mechanisms also improve the robustness of the adaptive law.

References

1. P. Angelov, "Evolving rules-based control," in *Proceedings of the EUNITE Symposium, Tenerife, Spain*, 2001, pp. 36–41.
2. P. Angelov, "A fuzzy controller with evolving structure," *Information Sciences*, vol. 161, no. 1–2, pp. 21–35, 2004.
3. R.-E. Precup, M.-b.Radac, E. M. Petriu, and R.-C. Roman, "Evolving Fuzzy Models for the Position Control of Twin Rotor Aerodynamic Systems," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 237–242.
4. A. Silva, W. Caminhas, A. Lemos, and F. Gomide, "Real-time nonlinear modeling of a twin rotor MIMO system using evolving neuro-fuzzy network," *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 2014, pp. 1–8.
5. R.-E. Precup and T.-a.Teban, "Evolving Fuzzy Models for Myoelectric-based Control of a Prosthetic Hand," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 72–77.
6. D. Leite, R. M. Palhares, V. C. S. Campos, and F. Gomide, "Evolving Granular Fuzzy Model-Based Control of Nonlinear Dynamic Systems," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 923–938, 2015.
7. D. Domingos, G. Camargo, and F. Gomide, "Autonomous Fuzzy Control and Navigation of Quadcopters," *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 73–78, 2016.
8. A. Zdešar, O. Cerman, D. Dovžan, P. Hušek, and I. Škrjanc, "Fuzzy control of a helio-crane: comparison of two control approaches," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3–4, pp. 497–515, 2013.
9. D. Dovžan, S. Blažic, and I. Škrjanc, "Towards evolving fuzzy reference controller," in *2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, Linz, Austria, 2014, pp. 1–8.
10. A. Zdešar, D. Dovžan, and I. Škrjanc, "Self-tuning of 2 DOF control based on evolving fuzzy model," *Applied Soft Computing*, vol. 19, pp. 403–418, 2014.
11. S. Blažič, D. Dovžan, and I. Škrjanc, "Robust evolving fuzzy adaptive control with input-domain clustering," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 5387–5392, 2014.
12. S. Blažič, I. Škrjanc, and D. Matko, "A robust fuzzy adaptive law for evolving control systems," *Evolving Systems*, vol. 5, no. 1, pp. 3–10, 2014.
13. D. Dovžan, V. Logar, and I. Škrjanc, "Solving the sales prediction problem with fuzzy evolving methods," in *WCCI 2012 IEEE World Congress on Computational Intelligence*, Brisbane, Australia, 2012, pp. 10–15.
14. I. Škrjanc, "Evolving Fuzzy-Model-Based Design of Experiments," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 861–871, 2015.
15. R. E. Precup, M. C. Sabau, and E. M. Petriu, "Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for Anti-lock Braking Systems," *Applied Soft Computing Journal*, vol. 27, pp. 575–589, 2015.
16. A. Sebastião, C. Lucena, L. Palma, A. Cardoso, and P. Gil, "Optimal tuning of scaling factors and membership functions for mamdani type PID fuzzy controllers," in *Proceedings – 2015 International Conference on Control, Automation and Robotics, ICCAR 2015*, 2015, pp. 92–96.
17. H. N. Chaleshtori, S. M. A. Mohammadi, and E. Bijami, "Optimal design of fractional order fuzzy PID controller with simultaneous auto-tuned fuzzy control rules and membership functions," in *2nd Conference on Swarm Intelligence and*

Evolutionary Computation, CSIEC 2017 – Proceedings, 2017, pp. 100–105.
18. P. Angelov and R. Yager, “Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density,” in *Symposium Series on Computational Intelligence (IEEE SSCI 2011) – IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS 2011)*, 2011, pp. 62–69.
19. B. S. J. Costa, I. Škrjanc, S. Blažič, and P. Angelov, “A practical implementation of self-evolving cloud-based control of a pilot plant,” in *IEEE International Conference on Cybernetics, CYBCONF*, 2013, pp. 7–12.
20. G. Andonovski, P. Angelov, S. Blažič, and I. Škrjanc, “A practical implementation of Robust Evolving Cloud-based Controller with normalized data space for heat-exchanger plant,” *Applied Soft Computing*, vol. 48, pp. 29–38, 2016.
21. I. Škrjanc, S. Blažič, P. Angelov, “Robust Evolving Cloud-based PID Control Adjusted by Gradient Learning Method,” *2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 1–8, 2014.
22. B. S. J. Costa, C. G. Bezerra, and L. A. H. G. De Oliveira, “A multistage fuzzy controller: Toolbox for industrial applications,” in *2012 IEEE International Conference on Industrial Technology, ICIT 2012*, 2012, pp. 1142–1147.
23. B. S. J. Costa, P. P. Angelov, and L. A. Guedes, “Real-time fault detection using recursive density estimation,” *Journal of Control, Automation and Electrical Systems*, vol. 25, no. 4, pp. 428–437, 2014.
24. R.-E. Precup, P. Angelov, B. S. J. Costa, and M. Sayed-Mouchaweh, “An overview on fault diagnosis and nature-inspired optimal control of industrial process applications,” *Computers in Industry*, vol. 74, pp. 75–94, 2015.
25. B. S. J. Costa, P. P. Angelov, and L. A. Guedes, “Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier,” *Neurocomputing*, vol. 150, Part A, pp. 289–303, 2015.
26. C. G. Bezerra, B. S. J. Costa, L. A. Guedes, and P. P. Angelov, “An evolving approach to unsupervised and Real-Time fault detection in industrial processes,” *Expert Systems with Applications*, vol. 63, pp. 134–144, 2016.
27. B. S. J. Costa, C. G. Bezerra, L. A. Guedes, P. P. Angelov, and N. Z. Norte, “Unsupervised Classification of Data Streams based on Typicality and Eccentricity Data Analytics,” in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 58–63.
28. S. Blažič, P. Angelov, and I. Škrjanc, “Comparison of Approaches for Identification of All-data Cloud-based Evolving Systems,” in *2nd IFAC Conference on Embedded Systems, Computer Intelligence and Telematics CESCIT 2015*, Maribor, Slovenia, 2015, pp. 129–134.
29. S. Blažič, D. Dovžan, and I. Škrjanc, “Cloud-based Identification of an Evolving system With Supervisory Mechanisms,” in *2014 IEEE International Symposium on Intelligent Control (ISIC)*, Antibes, France, 2014, pp. 1906–1911.
30. S. Blažič and I. Škrjanc, “Problems of Identification of Cloud-Based Fuzzy Evolving Systems,” in *Artificial Intelligence and Soft Computing. ICAISC 2016. Lecture Notes in Computer Science*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds. vol. 9692. Springer, Cham, 2016, pp. 173–182.
31. G. Andonovski, S. Blažič, P. Angelov, and I. Škrjanc, “Robust Evolving Cloud-based Controller in Normalized Data Space for Heat-Exchanger Plant,” in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Istanbul, Turkey, 2015, pp. 1–7.
32. P. Ioannou and J. Sun, *Robust Adaptive Control*. PTR Prentice-Hall, 1996.
33. G. Andonovski, S. Blažič, P. Angelov, and I. Škrjanc, “Analysis of Adaptation Law of the Robust Evolving Cloud-based Controller,” in *2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2015, pp. 1–7.
34. B. B. Peterson and K. S. Narendra, “Bounded Error Adaptive Control,” *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1161–1168, 1982.
35. G. Kreisselmeier and K. Narendra, “Stable model reference adaptive control in the presence of bounded disturbances,” *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1169–1175, 1982.
36. P. Ioannou and P. Kokotovic, “Instability analysis and improvement of robustness of adaptive control,” *Automatica*, vol. 20, no. 5, pp. 583–594, 1984.
37. K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison-Wesley Longman, Inc., 1998.
38. DeLorenzo, “Didactic process control pilot plant,” in *DeLorenzo Italy: Catalog*, 2009.
39. I. Škrjanc, S. Blažič,and D. Matko, “Direct fuzzy model-reference adaptive control,” *International Journal of Intelligent Systems*, vol. 17, no. 10, pp. 943–963, 2002.
40. W. K. Ho, K. C. Tan, A. Tay, and R. Srinivasan, “Using the OPC standard for real-time process monitoring and control,” *IEEE Software*, vol. 22, no. 6, pp. 54–59, 2005.

Bionotes

Goran Andonovski
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia
goran.andonovski@fe.uni-lj.si

Goran Andonovski received the B.Sc. degree in 2012 from the Faculty of Electrical Engineering, University of Ljubljana. He is currently working on his PhD thesis and he is employed as a researcher at the Laboratory of Modelling, Simulation and Control at the same University. His research interests include adaptive and predictive control of nonlinear processes, evolving fuzzy learning methods and fault detection and diagnosis.

Bruno Sielly Jales Costa
Campus Natal – Zona Norte, Federal Institute of Rio Grande do Norte, Natal, Brazil
Research and Innovation Center, Ford Motor Company, Palo Alto, CA, USA
bruno.costa@ifrn.edu.br

Bruno Costa received the B. Eng., M. Eng. and D. Eng. degrees in Electrical and Computer Engineering from the Federal University of Rio Grande do Norte (UFRN), Brazil, in 2008, 2010 and 2014, respectively. He is currently an Adjunct Professor with the Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN), Brazil. Dr. Costa's latest work is based on computational intelligence techniques applied to computer vision, anomaly detection, controls and fault detection and diagnosis problems.

Sašo Blažič
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia
saso.blazic@fe.uni-lj.si

Sašo Blažič received the B.Sc., M. Sc., and Ph. D. degrees in 1996, 1999, and 2002, respectively, from the Faculty of Electrical Engineering, University of Ljubljana. He is currently a Professor with the University of Ljubljana. His research interests include adaptive, fuzzy and predictive control of dynamical systems and modelling of nonlinear systems. Recently, the focus of his research has moved towards the areas of autonomous mobile systems, mobile robotics, and the control of satellite systems.

Igor Škrjanc
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia
igor.skrjanc@fe.uni-lj.si

Igor Škrjanc received B.S., M.S. and Ph.D. degrees in electrical engineering, in 1988, 1991 and 1996, respectively, at the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia. He is currently a Full Professor with the same faculty and Head of Laboratory for Autonomous and Mobile Systems. He is lecturing the basic control theory at graduate and advanced intelligent control at postgraduate study. His main research areas are adaptive, predictive, neuro-fuzzy and fuzzy adaptive control systems. His current research interests include also the field of autonomous mobile systems in sense of localization, direct visual control and trajectory tracking control. He is Humboldt research fellow, research fellow of JSPS and Chair of Excellence at University Carlos III of Madrid. He also serves as an Associated Editor for IEEE Transaction on Neural Networks and Learning System, IEEE Transaction on Fuzzy Systems, the Evolving Systems journal and International journal of artificial intelligence.