Fuzzy Control of a Helio-Crane

Andrej Zdešar, Otta Cerman, Dejan Dovžan, Petr Hušek & Igor Škrjanc

Journal of Intelligent & Robotic Systems
with a special section on Unmanned Systems

ISSN 0921-0296
Volume 72
Combined 3-4

J Intell Robot Syst (2013) 72:497-515
DOI 10.1007/s10846-012-9796-0

Springer

J Intell Robot Syst (2013) 72:497–515
DOI 10.1007/s10846-012-9796-0

Fuzzy Control of a Helio-Crane

Comparison of Two Control Approaches

Andrej Zdešar · Otta Cerman · Dejan Dovžan · Petr Hušek · Igor Škrjanc

Received: 11 July 2012 / Accepted: 5 December 2012 / Published online: 23 January 2013

Abstract In this paper we present a comparison of two fuzzy-control approaches that were developed for use on a non-linear single-input single-output (SISO) system. The first method is Fuzzy Model Reference Learning Control (FMRLC) with a modified adaptation mechanism that tunes the fuzzy inverse model. The basic idea of this method is based on shifting the output membership functions in the fuzzy controller and in the fuzzy inverse model. The second approach is a 2 degrees-of-freedom (2 DOF) control that is based on the Takagi-Sugeno fuzzy model. The T-S fuzzy model is obtained by identification of evolving fuzzy model and then used in the feed-forward and feedback parts of the controller. An error-model predictive-control approach is used for the design of the feedback loop. The controllers were compared on a non-linear second-order SISO system named the helio-crane. We compared the performance of the reference tracking in a simulation environment and on a real system. Both methods provided acceptable tracking performance during the simulation, but on the real system the 2 DOF FMPC gave better results than the FMRLC.

Keywords Fuzzy Model Reference Learning Control · Takagi-Sugeno fuzzy model · Evolving fuzzy model · 2 DOF control · Model predictive control

Mathematics Subject Classifications (2010) 91C20 · 62H86

A. Zdešar (✉) · D. Dovžan · I. Škrjanc
Laboratory of Modelling, Simulation and Control, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia
e-mail: andrej.zdesar@fe.uni-lj.si

O. Cerman · P. Hušek
Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27, Prague 6, Czech Republic

1 Introduction

The fuzzy logic is known to be able to solve control problems that are hard to solve with the traditional approaches. The main motivation for this paper comes from the fact that the fuzzy logic can solve problems that are highly non-linear. Furthermore, many different fuzzy control algorithms were developed over the years, and this motivated us to make some comparison between different fuzzy control approaches. The goal of this paper is to evaluate the performance of two different fuzzy control approaches and to determine whether the different fuzzy controllers can solve the same control problem. In this paper we compared the FMRLC with a modified adaptation algorithm and the 2 DOF FMPC. The main contribution of this paper is the comparison of the performance of two different fuzzy control approaches. The fuzzy control approaches were compared not only in simulation environment, but also on the real system. The paper shows how different fuzzy control methods can be used to control SISO non-linear systems.

The outline of the paper is as follows. Section 2 gives an overview of the existing work. Section 3 briefly introduces the FMRLC and the FMRLC with a modified adaptation mechanism that tunes the fuzzy inverse model. Section 4 gives a brief overview of the second fuzzy-control scheme, the 2 DOF FMPC, and presents the method of evolving fuzzy modelling. This is followed by Section 5, which presents the helio-crane system that was used for the comparison of both the presented fuzzy-control algorithms. In Section 6 the experimental results are presented. Both the control approaches were tested in the simulation and real environments. The performances of both control algorithms were evaluated using several different criteria. Finally, Section 7 gives a descriptive comparison of the presented control approaches based on the experimental results and then draws some conclusions.

2 Existing Work

Since 1965, when the fuzzy-set theory was proposed by Zadeh [1], fuzzy logic has been successfully applied to a diverse range of applications, mostly in the fields of control and artificial intelligence. The heart of a fuzzy system is the inference engine that is responsible for the rule processing. The inference engine operates on a set of linguistic variables, so the crisp input variables must be described in terms of a set of linguistic variables (e.g., error is small/big, change is fast/slow), a process known as fuzzification. After the fuzzy-rules processing, a process of defuzzification is necessary to transform the linguistic variables back to the crisp values, so that the result can be applied to a real system.

A fuzzy controller becomes self-organizing, self-learning or self-tuning when it is able to adjust the control rules based on past experience without any human intervention. The first self-organizing fuzzy controller was worked out in [2] and was further elaborated in [3]. Unfortunately, the usability of that approach was limited to a small class of plants because of the difficult design for tracking reference signals that were different from a step signal. This drawback was eliminated by the Fuzzy Model Reference Learning Control (FMRLC) algorithm, which was introduced for the first time in [4]. The main idea is based on the conventional Model Reference Adaptive Control approach (MRAC) [5].

The self-learning concept of the FMRLC seems to have great potential which is supported by some successful applications in aeronautics [6, 7], hydraulic control units [8, 9], the control of manipulators [10, 11], induction machines and generators [12, 13]. Other areas associated with the use of FMRLC are analyzed, e.g., in [14–19]. Some improvements to the method can be found in [20] with an application in [21]. Other approaches based on the FMRLC method are described in [22, 23].

Fuzzy model represents a convenient way to describe the system behaviour. Furthermore, Takagi-Sugeno fuzzy models are thought of as universal approximators, since every system can be represented to an arbitrary precision in the form of a T-S fuzzy model [24, p. 77].

To identify the T-S model the structure and the parameters of the local models must be identified [25]. A structure identification includes an estimation of the cluster centers (antecedent parameters), which is usually done by fuzzy clustering. Then for each cluster the sub-model's parameters are estimated, which is usually done with a least-squares method [26].

The identification can be made off-line or on-line. The on-line learning of the fuzzy model has made significant progress in the past few decades. A range of on-line identification procedures was developed. Some of them are based on fuzzy logic (eTS [27], exTS [28, 29], simple_TS [30], +eTS [31], FLEXFIS [32], FLEXFIS+ [33]), others use neural networks that realize the behavior of the fuzzy model (EFuNN [34, 35], DENFIS [36], ANFIS [37], GANFIS [38], SOFNN [39], SAFIS [40], SCFNN [41], NFCN [42], D-FNN [43], GD-FNN [44], SONFIN [45], NeuroFAST [46], RAN [47], ESOM [48], Neural gas [49], ENFM [50], and GAP-RBF [40]).

The fuzzy models that are normally used by the methods are first-order Takagi-Sugeno (ANFIS, SONFIN, D-FNN, GD-FNN, DENFIS, eTS, NeuroFAST, SOFNN, etc.), zero-order Takagi-Sugeno (SCFNN, SAFIS, GAP-RBF, EFuNN) or a generalized fuzzy model (GANFIS).

The methods also differ in their ability to adapt the fuzzy model and its structure. Some of the methods require an initial fuzzy-model structure, which is then adapted. The adaptation includes only the adaptation of the consequent and premise parameters (*adaptive methods* [37, 38]). Some of the methods include a mechanism for adding new clusters to the model structure (*incremental methods* [27]). Recently proposed methods also include mechanisms for merging, removing and splitting clusters. The methods use different clustering algorithms, such as ECM [36], recursive subtractive clustering [27], Gath-Geva clustering [50] and others. The local model parameters' identification is usually done with some version of the least-squares algorithm. In this paper the evolving fuzzy-model method (eFuMo) will be used for the fuzzy-model identification. The method is based on the recursive Gustafson-Kessel clustering algorithm [51, 52] and recursive fuzzy least squares [27]. It employs evolving mechanisms for adding, removing, merging and splitting the clusters. This method was also used in [53] for constructing the adaptive fuzzy predictive functional controller for a semi-batch reactor.

The model of a system can be used to make predictions about the system behaviour, thereafter the model be used to determine the optimal control actions that take the system dynamics and constraints into account, the approach known as *predictive control*. Over the years many different predictive control algorithms have been developed: Generalized Predictive Control (GPC) [54], Model Algorithmic Control (MAC) [55], Predictive Functional Control (PFC) [56], Model-based Predictive Control (MPC) [57] etc. Predictive controllers were originally designed for linear systems, but the idea has since been extended to non-linear systems. Many different fuzzy-control approaches have been proposed: predictive functional control based on a fuzzy model [58, 59], Fuzzy Model-Based Predictive Control (FMBPC) [60], etc.

Historically, the main emphasis in system-control design has been on the feedback loop; however, recently, the research interest in feedforward control has been growing [61, 62]. A feedforward controller alone can never achieve an accurate tracking performance, but with the addition of a feedback extension this deficiency can be eliminated. The combination of feedforward and feedback control loops is known as *2 DOF control*. The feedforward part based on the fuzzy inverse model should provide a fast reaction to reference changes and drive the output into the vicinity of a reference. A feedback part based on the fuzzy model should eliminate the reference tracking errors that occur due to disturbances, drift, noise, imprecise system modelling, etc. The idea of 2 DOF control has, in recent years, received a lot of attention in the control community and has been successfully implemented in a diverse range of applications [61, 63–68].

3 FMRLC with a Modified Adaptation Mechanism

3.1 Classical FMRLC

First, a classical FMRLC method will be described briefly. More details can be found in [24]. The basic scheme of the FMRLC with a fuzzy PD controller and single-input single-output (SISO) system is shown in Fig. 1. It consists of four main parts: the fuzzy controller to be tuned, the plant, the reference model and the learning (adaptation) mechanism. Let us describe the role of each component in more detail.

For a better and more intuitive explanation the following nomenclature was established. The inputs to the controller before the scaling and the output from the controller after the scaling are labelled without a lower index, see Fig. 1. The inputs to the controller after the scaling and the output from the controller before the scaling include the lower index r. All the gains begin with the letter g followed by the lower index labelling signal that belongs to the relevant gain.

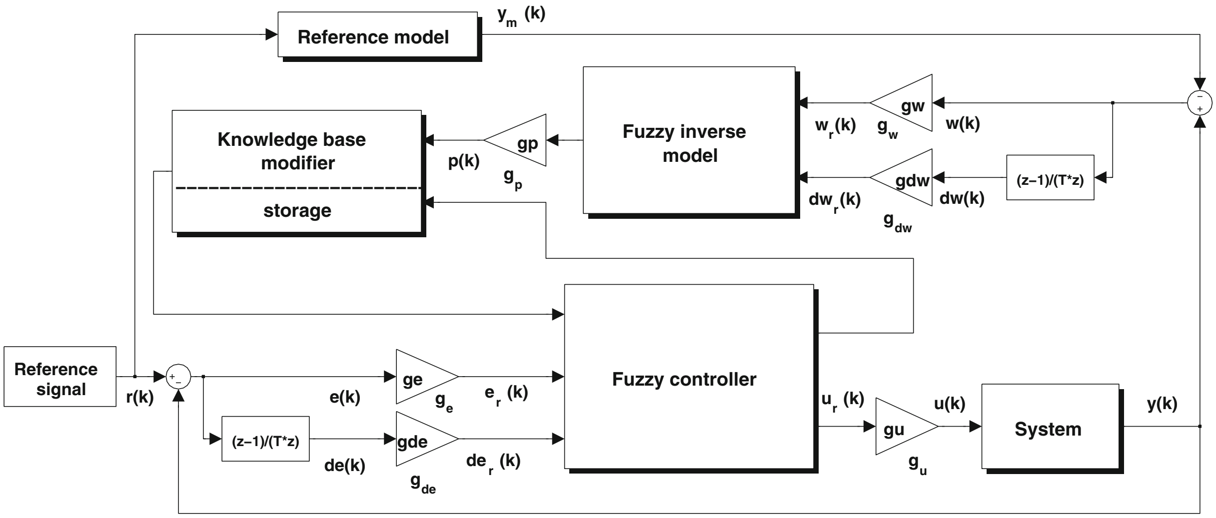


Fig. 1 Basic structure of FMRLC

3.1.1 The Fuzzy Controller

For the explanation a fuzzy PD controller is used; therefore, the inputs to the controller are the control error $e(k) = r(k) - y(k)$ and the change in the error $de(k) = \frac{e(k)-e(k-1)}{T_s}$, and the output from the controller is $u(k)$, where T_s is the sampling time and k is the time sample, see Fig. 1. Generally, the structure and the inputs to the fuzzy controller can be chosen arbitrarily.

For fuzzy-controller design, the scaling gains are used to normalize the universe of the discourse, see Fig. 1. These gains are normally tuned within the overall FMRLC initialization.

The input fuzzy sets are chosen to characterize the premises of the controller rules, while their shape and position remain fixed during the whole control process. In the fuzzy controller, uniformly distributed, symmetric triangular fuzzy sets are normally used for both input universes of the discourse. The position of the output fuzzy sets is assumed to be unknown and will be synthesized or tuned automatically. Although arbitrary shapes of the output fuzzy sets can be chosen the singletons are sufficient in most applications. At the beginning of the adaptation the position of all of them is normally supposed to be at zero, but a different initialization can be set up. To complete the specification of the fuzzy controller a standard center-of-gravity defuzzification technique is used.

3.1.2 Reference Model

The reference model defines the closed-loop specifications (such as stability, rise time, overshoot, etc.) and generates the desired trajectory. Similar to the conventional MRAC, the learning mechanism modifies the fuzzy controller so that the closed-loop system behaves like the given reference model. The reference model has to be chosen with care because if the requirements are too strong the adaptation mechanism shifts the output fuzzy sets by large steps, which can cause a loss of stability.

3.1.3 Learning Mechanism

The learning mechanism consists of two parts (see Fig. 1): *a fuzzy inverse model* and *a knowledge-base modifier*. The purpose of the fuzzy inverse model is to determine the changes $p(k)$ in the process inputs $u(k)$ for pushing the deviation $w(k)$ to zero.

The rule design in the fuzzy inverse model is based on the fact that most often a process operator can roughly characterize the behaviour of the process. The inverse model may be designed in such a way, for example, if the error is small, then the adjustments to the fuzzy controller should be small, and if the error is small, but the rate of error increase is high, then the adjustments should be larger. Several examples and ways of choosing these rules can be found, e.g., in [24]. Given the information about the necessary changes $p(k)$ in the control input $u(k)$ to force the error w to zero, the knowledge-base modifier shifts the output membership functions of the pertinent rules that acted with the highest degree of fulfillment at the time instant $kT_s - T_d$, where T_d is the delay of the plant. Alternative knowledge-base modifiers, described, e.g., in [69], can be used for a more effective adaptation process in the presented FMRLC approach too. Note that analogously to the fuzzy controller, the fuzzy inverse model contains normalizing scaling factors, i.e. g_w, g_{dw}, g_p, for $w(k)$, $dw(k)$ and $p(k)$.

3.2 FMRLC with a Modified Adaptation Mechanism

FMRLC with a modified adaptation mechanism [70] contains an adaptation mechanism for the rules in the inverse model. The proposed algorithm proceeds in such a way that if the consequence of a rule is a worse reference-signal tracking from the control error $e_r(k)$ and its difference $de_r(k)$ point of view, the output of the rule is modified.

When a rule in the controller is activated again (after the time interval $T_m = mT_s$) the values of the relative change of the error $e_m(k) = e_r(k) - e_r(k-m)$ and the relative change of the error difference $de_m(k) = e_r(k) - e_r(k-m)$ are tested. In the case that the actual values of $e_m(k)$ and $de_m(k)$ are both positive or both negative, the consequence of the rule in the inverse model is changed.

For this purpose a new fuzzy system for the modification of the inverse model is established that maps $e_m(k)$ and $de_m(k)$ to the necessary changes in the inverse model output $p(k)$. Note that similar to the fuzzy controller and the inverse model this fuzzy system contains the scaling gains g_{m_e}, $g_{m_{de}}$ and g_{m_u}. The input membership functions cover the whole input space of the fuzzy system.

Table 1 Rules in the fuzzy system for modification of the inverse model

MF $e(k)$	MF $de_m(k)$	Output MF
Negative	Negative	Negative big
Negative	Zero	Negative medium
Negative	Positive	Negative small
⋮	⋮	⋮
Positive	Negative	Positive small
Positive	Zero	Positive medium
Positive	Positive	Positive big

A part of the rules in the fuzzy system for the modification of the inverse model is depicted in Table 1. The fuzzy system has three input triangular membership functions that are uniformly spaced over the interval $[-1, 1]$ on both inputs. The membership functions are labelled by the linguistic values Negative, Zero and Positive. On the output there are nine singletons that are uniformly spaced on the interval $[-1, 1]$ labelled by the linguistic values Negative big, Negative medium, Negative small, Zero negative, Zero, Zero positive, Positive small, Positive medium and Positive big.

The fuzzy system for the modification of the inverse model has rules of the following form:

$$\textbf{IF } e_m \text{ is } E^j \textbf{ AND } de_m \text{ is } C^k, \ \textbf{THEN } u_m \text{ is } U^l, \quad (1)$$

where e_m and de_m denote the linguistic variables associated with the controller inputs $e_m(k)$ and $de_m(k)$, respectively, u_m denotes the linguistic variable associated with the fuzzy system output $u_m(k)$ and E^j, C^k and U^l are fuzzy sets.

To obtain the final value of $p_m(k)$ for the adaptation of the rule in the inverse model the output $u_m(k)$ from the fuzzy system is multiplied by the value $\mathrm{sign}(p(k-m))$. This fuzzy system brings the change into effect by shifting the output membership function of the rule in the inverse model by $p_m(k)$.

4 2 DOF FMPC

The main goal of the presented control algorithm is to provide precise reference-signal tracking. The control scheme depicted in Fig. 2 consists of a fuzzy system identification block and a 2

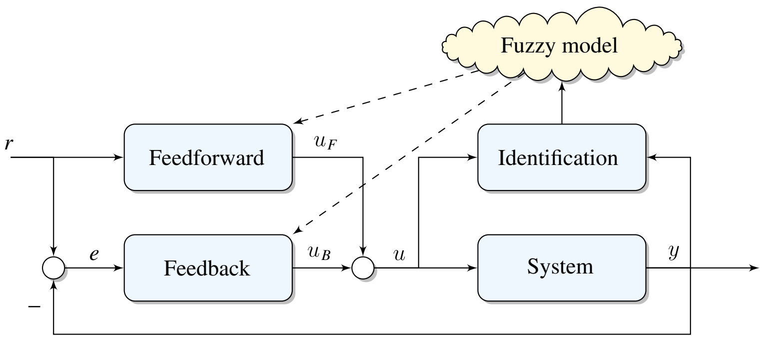


Fig. 2 Control structure of 2 DOF FMPC

DOF FMPC: a combination of feedforward and feedback loops. Once the fuzzy-system model is identified, it can be fed to the controller and the controller can be switched on. In the following section the basic structure of the controller is presented briefly.

4.1 Fuzzy Modelling

An arbitrary system can be described with a set of K fuzzy rules $\{\mathcal{R}^j\}_{j=1,\ldots,K}$ in Takagi-Sugeno fuzzy form, where the rule $\mathcal{R}^j$ is defined as:

$$\begin{aligned}&\textbf{IF } y(k-n+1) \text{ is } A_n^j \textbf{ AND } \ldots \textbf{ AND } y(k) \text{ is } A_1^j,\\&\textbf{THEN } y(k+1) = f_j(u(k-m+1), \ldots, u(k),\\&\quad y(k-n+1), \ldots, y(k)) .\end{aligned} \tag{2}$$

Antecedents of the rules (**IF** parts) describe the fuzzy regions in the space of input variables. A_i^j represent fuzzy sets characterized by their membership functions. For the description of the input-output dynamics in **THEN** parts (consequences) we chose the Nonlinear Auto Regressive model with eXogenous inputs (NARX), but any other model could be used as well. The NARX model predicts the next output based on past inputs and outputs. Furthermore, we assume that every **THEN** part of each fuzzy rule can be approximated with the affine NARX model:

$$f_j(k) = \boldsymbol{\theta}_j^T \boldsymbol{\psi}(k) , \tag{3}$$

where $\boldsymbol{\theta}_j^T = [r_j \;\; b_{m,j} \ldots b_{1,j} \;\; a_{n,j} \ldots a_{1,j}]$ contains all the parameters that apply to the rule $\mathcal{R}^j$. In the vector $\boldsymbol{\psi}^T(k) = [1 \; u(k-m+1) \ldots u(k) \; y(k-n+1) \ldots y(k)]$ the previous inputs and outputs are gathered.

The predicted output of a fuzzy model can be given in a compact matrix form as follows:

$$y(k+1) = \boldsymbol{\beta}^T(k) \boldsymbol{\Theta}^T \boldsymbol{\psi}(k) . \tag{4}$$

Here, $\boldsymbol{\beta}^T(k)$ represents the normalized degrees of fulfillment for the whole set of fuzzy rules $\{\mathcal{R}^j\}_{j=1,2,\ldots,K}$ in the current time step, written in the vector form $\boldsymbol{\beta}^T(k) = [\beta_1(k) \;\; \beta_2(k) \ldots \beta_K(k)]$. We assume the normalized degrees of fulfillment, which are generally time dependent, comply with Eq. 5 for every time step k.

$$\sum_{j=1}^{K} \beta_j(k) = 1 \tag{5}$$

In Eq. 4 the matrix $\boldsymbol{\Theta} \in \mathbb{R}^{1+m+n} \times \mathbb{R}^K$ contains all the parameters of the fuzzy model for the whole set of rules $\{\mathcal{R}^j\}_{j=1,2,\ldots,K}$: $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1 \; \boldsymbol{\theta}_2 \; \ldots \; \boldsymbol{\theta}_K]$.

Next, a brief overview of the identification method is presented.

4.2 Evolving Fuzzy Model

The evolving fuzzy model is based on recursive Gustafson-Kessel (GK) clustering. The algorithm starts with one cluster and adds clusters if necessary. The first data sample is taken as an initial center of the first cluster. The method considers two different regression vectors. One is for clustering ($\boldsymbol{x}_f$) (clustering data vector) and the other is for local model-parameter estimation ($\boldsymbol{x}_k$) (the regression vector). The y in the following equations denotes the output of the process.

To cluster the input-output space the positions of the cluster centers and the variance of the data around the clusters should be calculated. Using the fuzzy c-means recursive algorithm this can be done using the following equations. First, the new center position of the cluster $i \in \{1, 2, \ldots, c\}$ is calculated as:

$$\boldsymbol{v}_i(k+1) = \boldsymbol{v}_i(k) + \frac{\left(\mu_{i,k+1}\right)^{\eta}\left(\boldsymbol{x}_f(k+1) - \boldsymbol{v}_i(k)\right)}{s_i(k+1)}, \tag{6}$$

where $\boldsymbol{v}_i(k)$ is the center position for the previous sample, $\boldsymbol{x}_f(k+1)$ is the current clustering data vector and $s_i(k+1)$ is the sum of the past membership degrees calculated as:

$$s_i(k+1) = \gamma_v s_i(k) + \mu_{i,k+1}^{\eta}. \tag{7}$$

The initial $s_i(0)$ is usually set to one. The γ_v is the forgetting factor. The membership $\mu_{i,k+1}$ of the current clustering vector $\boldsymbol{x}_f$ is calculated as:

$$\mu_{i,k+1} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{i,k+1}}{d_{j,k+1}}\right)^{\frac{2}{\eta-1}}}, \tag{8}$$

where $d_{i,k+1}$ is the distance of the clustering vector to the i-th cluster and η is the fuzziness (in most cases $\eta = 2$). The distance used with the GK clustering algorithm is defined as:

$$d_{i,k+1}^2 = \left(\boldsymbol{x}_f(k+1) - \boldsymbol{v}_i(k)\right)^T \boldsymbol{A}_i \left(\boldsymbol{x}_f(k+1) - \boldsymbol{v}_i(k)\right), \tag{9}$$

where $\boldsymbol{A}_i = [\rho_i \det(\boldsymbol{F}_i)]^{1/p}\boldsymbol{F}_i^{-1}$. The parameter ρ_i is usually set to a value of one and p depends on the number of variables (number of elements of $\boldsymbol{x}_f$).

The fuzzy covariance is calculated using the following equation:

$$\boldsymbol{F}_i(k+1) = \gamma_c \frac{s_i(k)}{s_i(k+1)}\boldsymbol{F}_i(k) + \frac{\mu_{i,k+1}^{\eta}}{s_i(k+1)}\boldsymbol{D}_{F_i} \tag{10}$$

$$\begin{aligned}\boldsymbol{D}_{F_i} &= \left(\boldsymbol{x}_f(k+1) - \boldsymbol{v}_i(k+1)\right) \\ &\quad \times \left(\boldsymbol{x}_f(k+1) - \boldsymbol{v}_i(k+1)\right)^T.\end{aligned} \tag{11}$$

A detailed description for the calculation of the recursive inverse fuzzy covariance matrix and the determinant can be found in [52].

Once the clusters are updated the fuzzy recursive least squares are used to update the linear sub-models' parameters. There are different algorithms proposed [27, 50, 71, 72], and these are based on weighted recursive least squares. The equations for adaptation based on [72] are:

$$\begin{aligned}\boldsymbol{\psi}_i(k+1) &= [1, \boldsymbol{x}_k(k+1)^T]^T \\ y(k+1) &= y(k+1)\end{aligned} \tag{12}$$

$$\begin{aligned}&\boldsymbol{P}_i(k+1) \\ &= \frac{1}{\lambda_r}\left(\boldsymbol{P}_i(k) - \frac{\beta_i \boldsymbol{P}_i(k)\boldsymbol{\psi}_i(k+1)\boldsymbol{\psi}_i^T(k+1)\boldsymbol{P}_i(k)}{\lambda_r + \beta_i\boldsymbol{\psi}_i^T(k+1)\boldsymbol{P}_i(k)\boldsymbol{\psi}_i(k+1)}\right) \\ &\boldsymbol{\theta}_i(k+1) \\ &= \boldsymbol{\theta}_i(k) + \boldsymbol{P}_i(k)\beta_i\boldsymbol{\psi}_i(k+1) \\ &\quad \times \left(y(k+1) - \boldsymbol{\psi}_i^T(k+1)\boldsymbol{\theta}_i(k)\right)\end{aligned} \tag{13}$$

The parameters of the i-th sub-models are denoted as $\boldsymbol{\theta}_i$, the forgetting factor is denoted with λ_r and β_i denotes the membership degree of the current clustering vector to the i-th cluster. In general, the β_i the membership degrees that are calculated during the cluster update do not have a smooth transition. Therefore, when identifying a process with a smooth nonlinearity it is better to recalculate the membership degrees using a Gaussian function:

$$\beta_i = \prod_{j=1}^{z} e^{-\frac{\left(x_{f_j} - v_{ij}\right)^2}{2\eta_m\sigma_{ij}^2}}, \tag{14}$$

where σ_{ij}^2 is the j-th diagonal element of the fuzzy covariance matrix and η_m is the overlapping factor (set between 0.25 to 1), and z is the number of components of the clustering vector $\boldsymbol{x}_f$. Note that these membership degrees β_i should be normalized as in Eq. 5. The settings for the parameters are given in [53] and [52].

The above equations represent the adaptation algorithm of the eFuMo method. To achieve the evolving nature of the method the mechanisms for adding, removing, splitting and merging the clusters must be included. The scheme of the algorithm is shown in Fig. 3. The evolving mechanisms are only briefly described in the following

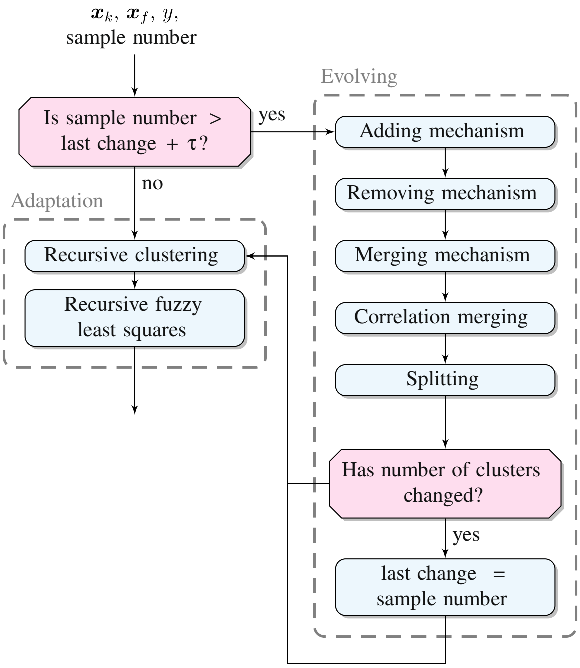


Fig. 3 Flowchart of a single step in the identification of evolving fuzzy model

paragraphs, since the topic is out of the scope of this paper.

The adding of clusters is usually done by some distance measure of the current clustering vector to existing clusters or by membership degree. If a current sample has a low membership degree a new cluster is added with the center in the current clustering vector. The eFuMo adding criterion is based on the normalized distance of the current clustering vector to the existing clusters.

The removing of clusters in our case is based on their support. The support is, in this case, defined as the number of samples that have the maximum membership to a certain cluster. The cluster is removed if, in a certain user-defined number of samples, it does not receive a certain number of support samples [31]. The removing of clusters is also made based on the age of the clusters.

The merging algorithm is meant to merge the clusters that are either close or have the same sub-model parameters. The eFuMo method considers merging based on the membership degrees of clusters to each other [50] and merging based on the correlation method [33].

The splitting of clusters is currently meant to fine tune the fuzzy model. It can add clusters in the input-output space where the output-model error is higher than a predefined threshold. The eFuMo method tracks the mean error for each cluster. For each sample, if the sample does not satisfy the distance condition for adding, the output error of the current model is calculated. Then the model error is divided among the clusters, depending on the membership degrees of the current clustering sample. If the error of one of the clusters exceeds the defined threshold, this cluster is split. The parameters of the model stay the same, and the centers of the clusters are positioned based on a fuzzy-variance matrix.

The eFuMo method also considers a general time delay (τ) for the evolving mechanisms. This delay is user specified. If a change in the number of clusters occurs, the evolving mechanisms are stopped for this specified delay.

4.3 Feedforward Control

In this section we present a short summary of an approach to model inversion introduced by Karer [61]. First, we break apart the matrices $\boldsymbol{\Theta}^T = [\boldsymbol{\theta}_r\ \boldsymbol{\Theta}_u^T\ \boldsymbol{\theta}_{u,1}\ \boldsymbol{\Theta}_y^T]$ and $\boldsymbol{\psi}^T(k) = [1\ \boldsymbol{\psi}_u^T(k)\ u(k)\ \boldsymbol{\psi}_y^T(k)]$ of a fuzzy model Eq. 4, where $\boldsymbol{\theta}_r \in \mathbb{R}^K$, $\boldsymbol{\Theta}_u \in \mathbb{R}^{m-1} \times \mathbb{R}^K$, $\boldsymbol{\theta}_{u,1} \in \mathbb{R}^K$, $\boldsymbol{\Theta}_y \in \mathbb{R}^n \times \mathbb{R}^K$ and $\boldsymbol{\psi}_u \in \mathbb{R}^{m-1}$, $\boldsymbol{\psi}_y \in \mathbb{R}^n$. Notice the slight abuse of notation, since $\boldsymbol{\theta}$ in this section does not correspond to the $\boldsymbol{\theta}$ defined in Section 4.1. The input $u(k)$ that influences the next output $y(k+1)$ can now be expressed as:

$$u(k) = \frac{y(k+1) - \boldsymbol{\beta}^T(k)(\boldsymbol{\theta}_r + \boldsymbol{\Theta}_u^T \boldsymbol{\psi}_u(k) + \boldsymbol{\Theta}_y^T \boldsymbol{\psi}_y(k))}{\boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{u,1}}. \tag{15}$$

To calculate the optimum feedforward input $u_F(k)$ we substitute in Eq. 15 all the outputs y for the appropriate reference values r and all the inputs u for the previous feedforward inputs u_F.

Equation 15 clearly violates the causality constraint, since the optimum input in time step k is dependent on the future output at time step $k+1$. This does not pose a problem if the reference signal r is known in advance.

An important role in the calculation of the optimum input is played by the shape of the reference signal. It should be noted that some reference signals (too frequency rich) may push the solution of the system Eq. 15 outside the area of physically feasible solutions (e.g., an infinite input impulse). To overcome this problem, the reference signal must be chosen carefully. It is recommended that some kind of filtering is considered in order to suppress the high frequencies. When selecting the filter cut-off frequency we can take into account the system time constant and the upper and lower bounds of the input signal.

4.4 Model-Based Predictive Feedback Control

To eliminate the error $e(k)$ we design an error-model predictive controller. We follow the 2 DOF error-model tracking control design presented by Klančar and Škrjanc [64, 65]. We assume the control action is a sum of the feedforward and feedback parts $u(k) = u_F(k) + u_B(k)$. According to time-freeze theory, we can transform the fuzzy model Eq. 4 into a time-varying, affine, state-space form and then linearise it around the reference signal to obtain a linear, time-varying error model:

$$\begin{aligned} \boldsymbol{e}(k+1) &= \boldsymbol{A}(k)\boldsymbol{e}(k) + \boldsymbol{b}(k)u_B(k)\,, \\ e(k) &= \boldsymbol{c}^T(k)\boldsymbol{e}(k)\,. \end{aligned} \tag{16}$$

Here we have assumed that the error model has the same dynamics as the system model. The selection of the state-space variables should be according to the following scheme:

$$\boldsymbol{e}(k) = \begin{bmatrix} e(k) \\ e(k) - e(k-1) \\ e(k) - 2e(k-1) + e(k-2) \\ \vdots \end{bmatrix}, \tag{17}$$

where every element, except for the first, is the difference of a previous element for two consecutive previous time steps.

To ensure the integral action of the controller, an additional state-space variable v is introduced that integrates the output error $v(k+1) = e(k) + v(k)$. Defining the augmented vector $\boldsymbol{\xi}^T(k) = [\boldsymbol{e}^T(k)\ \boldsymbol{v}^T(k)]$, the extended system with the new state can be written:

$$\boldsymbol{\xi}(k+1) = \begin{bmatrix} \boldsymbol{A}(k) & \boldsymbol{0} \\ \boldsymbol{c}^T(k) & 1 \end{bmatrix} \boldsymbol{\xi}(k) + \begin{bmatrix} \boldsymbol{b}(k) \\ 0 \end{bmatrix} u_B(k)\,. \tag{18}$$

Using the model Eq. 18 predictions of the error $\boldsymbol{\xi}(k+i|k)$ over a finite horizon h can be made as functions of the current error $\boldsymbol{\xi}(k)$ and the unknown feedback inputs $u_B(k+i-1|k)$ for $i = 1, \ldots, h$. Next, a criterion function is defined:

$$\begin{aligned} J = \sum_{i=1}^{h} & \boldsymbol{\epsilon}^T(k+i|k)\,\boldsymbol{Q}\boldsymbol{\epsilon}(k+i|k) \\ & + u_B(k+i-1|k)\,R u_B(k+i-1|k)\,, \end{aligned} \tag{19}$$

where we have introduced $\boldsymbol{\epsilon}(k+i|k) = \boldsymbol{\xi}_r(k+i) - \boldsymbol{\xi}(k+i|k)$ as the difference between the reference error $\boldsymbol{\xi}_r(k+i)$ and the predicted error $\boldsymbol{\xi}(k+i|k)$. The reference error is usually defined as an exponentially decreasing function in the form of a state-space model: $\boldsymbol{\xi}_r(k+1) = \boldsymbol{A}_r\boldsymbol{\xi}(k)$. With an analytical differentiation of Eq. 19 with respect to the unknown inputs $\{u_B(k+i-1|k)\}_{i=1,\ldots,h}$, the optimum input signals $\{u_{B,opt.}(k+i-1|k)\}_{i=1,\ldots,h}$ over the prediction horizon can be found [64]. According to the receding horizon strategy, at time step k only $u_{B,opt.}(k|k)$ is added to the feedforward input signal $u_F(k)$ and during the next time step the whole procedure is repeated again.

The control algorithm can be tuned with several different parameters. The weighting semi-positive definite matrices $\boldsymbol{Q} \geq 0$ and $R \geq 0$ determine how strictly the predicted error should follow the desired reference error and how energy rich input signal is allowed, respectively. The controller can also be tuned by selecting the error reference model $\boldsymbol{A}_r$ and the length of the prediction horizon h. These parameters define the desired response dynamics and the power consumption.

5 Helio-crane

The system that was chosen for the comparison of the fuzzy-control algorithms is composed of a rigid metal rod on a pivot that can swing in a single vertical plane like a pendulum (Fig. 4). The swing of one end of the rod is physically restricted to a vertical half-plane, so the ends of the rod can

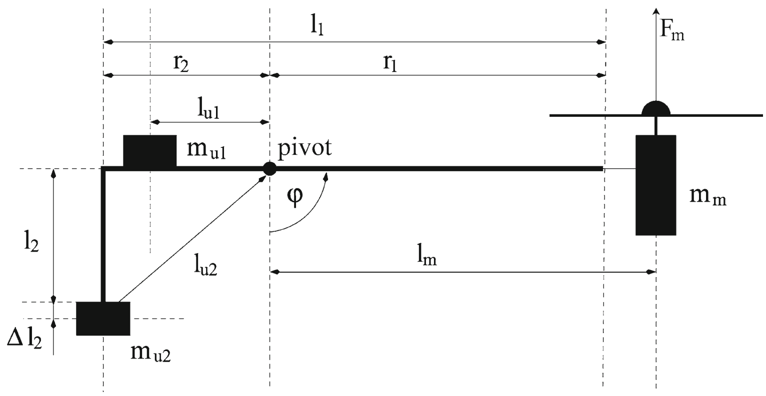


Fig. 4 Schematic model of a helio-crane

freely move up and down. At the end of one end of the swinging rod a motor with a lightweight plastic propeller is placed perpendicular to the rod, so the rod can be raised or lowered by changing the propeller's thrust F_m. When the motor is not turned on, the end of the rod with the motor is at the bottom position. The motor can only rotate in one direction, so the thrust always points in the same direction with respect to the motor. However, applying some thrust to the motor can only raise the rod, and the rod is lowered passively by gravitational force. To the main rod some additional weights are attached that influence the behaviour of the system. The interaction with the system is made through an additional electronic circuit. The speed of the motor (system input) is voltage controlled in the range from 0 V to 10 V. The inclination of the rod (system output) is measured with a resistive sensor for measuring the angle that returns the voltage, also in range from 0 V to 10 V. Since the system is composed of a motor with a propeller, like the one in a helicopter, mounted on a swinging rod, which can lift a weight like a crane, the system name was coined helio-crane.

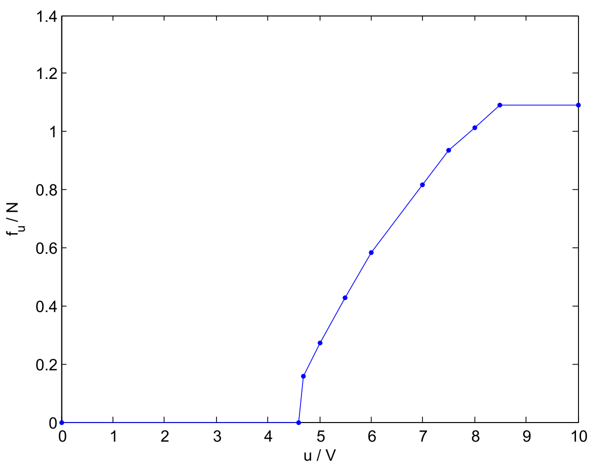
The system can be mathematically modelled by writing down the basic physics equation for rotating objects:

$$J\ddot{\varphi} = T(\varphi) - f\dot{\varphi}\,, \tag{20}$$

where φ is the inclination of the rod, T is the sum of all the torques on the system, J is the moment of inertia and f is the damping factor. The moment of inertia J can be determined from the

Table 2 Helio-crane (a) model parameters, (b) input and (c) output characteristic function data

a)			b)		c)	
Symbol	Value	Units	u [V]	f_u [N]	φ	f_y [V]
g	9.81	$\mathrm{m\,s^{-2}}$	0.0	0.0000	14°	6.00
ρ_d	1.13	$\mathrm{kg\,m^{-1}}$	4.6	0.0000	90°	3.52
m_m	0.13	kg	4.7	0.1558	177°	0.83
m_{u1}	0.145	kg	5.0	0.2727		
m_{u2}	0.02	kg	5.5	0.4286		
l_2	0.128	m	6.0	0.5844		
r_1	0.34	m	7.0	0.8182		
r_2	0.2	m	7.5	0.9351		
l_{u1}	0.165	m	8.0	1.0130		
l_m	0.38	m	8.5	1.0909		
l_{2d}	0.0269	$\mathrm{m^2}$	10.0	1.0909		
f	0.15	$\mathrm{kg\,m^2\,s^{-1}}$				
l_1	0.54	m				
ψ_1	0.3097	rad				
ψ_2	0.5872	rad				
J	0.0483	$\mathrm{kg\,m^2}$				
l_{u2}	0.2402	m				



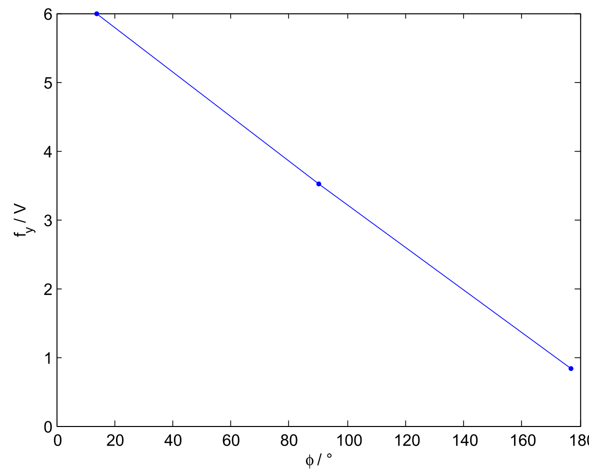


Fig. 5 **a** Input and **b** output characteristic function

physical dimensions of the helio-crane, applying some basic knowledge for the calculation of the partial moments of inertia. According to Fig. 4, the moment of inertia J is:

$$J = \rho_d \left(\frac{1}{3} l_1 (r_1^2 - r_1 r_2 + r_2^2) + l_2 \left(\frac{l_2^2}{3} + r_2^2 \right) \right) + m_m l_m^2 + m_{u1} l_{u1}^2 + m_{u2} l_{u2}^2 \tag{21}$$

and the torque T is:

$$T = F_m l_m + g \sin(\varphi) \left(\rho_d \left(\frac{r_2^2}{2} - \frac{r_1^2}{2} \right) + m_{u1} l_{u1} - m_m l_m \right) + \rho_d l_{2d} g \sin(\varphi + \psi_1) + m_{u2} g l_{u2} \sin(\varphi + \psi_2) \,. \tag{22}$$

The damping factor f can be heuristically determined and is defined in Table 2a, where all the other model parameters are also gathered. The output characteristic function $y = f_y(\varphi)$ is approximately linear (Table 2c, Fig. 5b), but the input characteristic function $F_m = f_u(u)$ is highly non-linear (Table 2b, Fig. 5a). We can conclude that the helio-crane is a SISO non-linear system with second-order dynamics, as Eq. 20 suggests.

6 Simulation and Real Experiment

The control algorithms were compared in a simulation environment and on a real system. Both control algorithms were implemented in a MATLAB simulation environment. The continuous system was simulated using MATLAB function `ode45`. For communication with the real system the MATLAB input-output interface was used. In the following subsections we present the criteria functions used for the comparison, the selected controller parameters for both control approaches and give the simulation and real results.

6.1 Criteria Functions

To compare the control algorithms several criteria functions were defined [73]. Besides the error function $e(k) = r(k) - y(k)$, we used two more integral (cumulative sum) criteria functions: the Sum of the Absolute Error

$$SAE = T_s \sum_k |e(k)| \tag{23}$$

and the Sum of the Squared Error

$$SSE = T_s \sum_k e^2(k) \,. \tag{24}$$

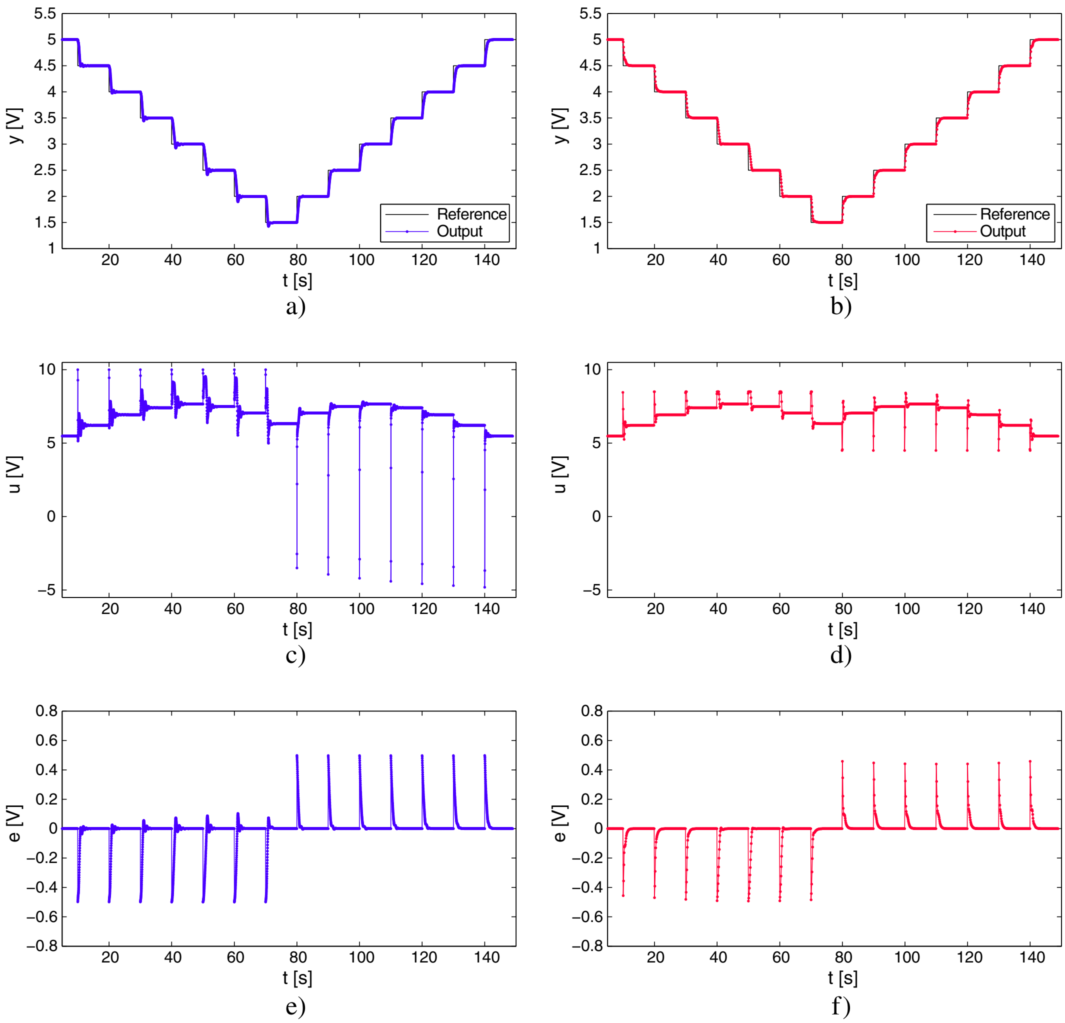


Fig. 6 Simulation. Comparison of FMRLC (*left column*) and 2 DOF FMPC (*right column*); **a**, **b** reference signal tracking, **c**, **d** control action and **e**, **f** error

The introduction of the sample time T_s into the criteria functions Eqs. 23 and 24 is necessary to allow for the case when the sampling times of the compared control algorithms are not equal. To evaluate the control effort we have taken a closer look at the change of the input action $\Delta u(k) = u(k) - u(k-1)$ and introduced two more integral criteria functions: the Sum of the Absolute Input differences

$$SAdU = \sum_k |\Delta u(k)| \tag{25}$$

and the Sum of the Squared Input differences

$$SSdU = \sum_k \Delta u^2(k)\,. \tag{26}$$

The results were also compared by means of the settling time $t_{s,\sigma}$ (a minimum time range after which the output stays within a predefined error region σ around the reference trajectory) and the maximum overshoot OS. If there were similar step reference changes at different operating points, the maximum settling time and overshoot

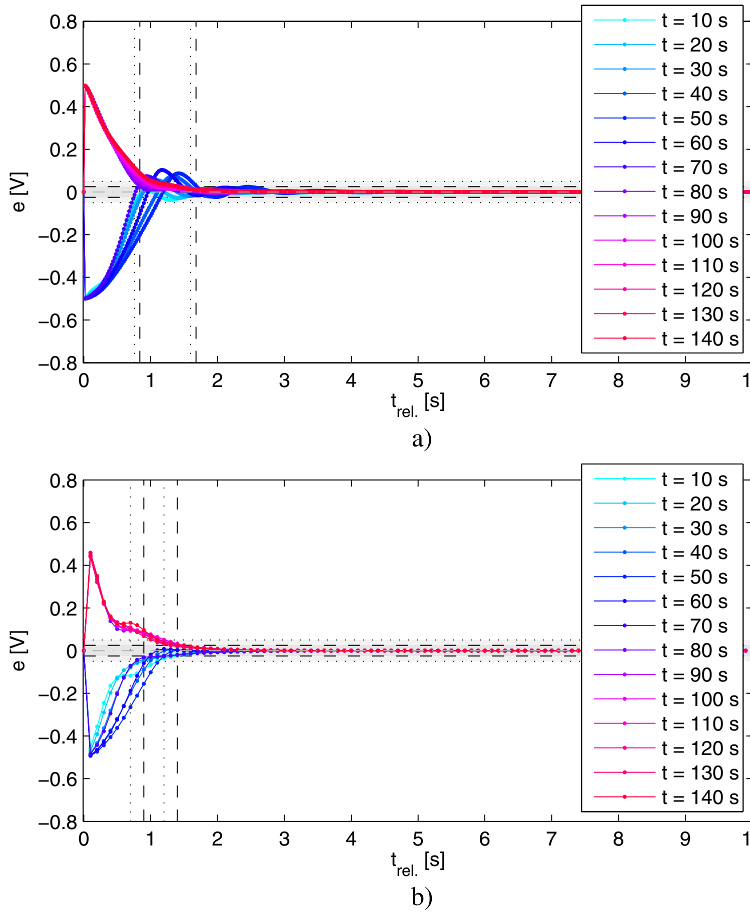


Fig. 7 Simulation. Error signals around all the reference steps plotted in the same time frame for **a** FMRLC and **b** 2 DOF FMPC

from among all the responses at different operating points were selected as the criteria for the comparison, denoted as $\max t_{s,\sigma}$ and $\max OS$, respectively.

6.2 Controllers Parameters

6.2.1 FMRLC

This method uses three triangular membership functions that are uniformly spaced on each input in the controller and three singletons that are initially distributed uniformly on the output corresponding to the allowable range of the control signal. For the inverse model and the fuzzy system for the modification of the inverse model three triangular membership functions that are uniformly spaced on each input are used, respectively, and three singletons that are distributed uniformly on the interval on the outputs, respectively. The transfer function of the reference model was chosen with respect to the dynamics of the controlled system as $\frac{Y_m(s)}{R(s)} = \frac{16}{s^2+8s+16}$.

The controller parameters were set to $g_e = g_w = 0.5$, $g_{de} = g_{dw} = \frac{1}{4}$, $g_u = 10$, $g_p = \frac{1}{5} g_u = 2$.

6.2.2 2 DOF FMPC

For the evolving fuzzy identification a second-order NARX model was chosen: $n = 2$ and $m = 1$ in Eq. 2, so the matrix $\boldsymbol{\Theta}$ with the model parameters takes the form $\boldsymbol{\Theta}^T = [\boldsymbol{\theta}_r \ \boldsymbol{\theta}_{u,1} \ \boldsymbol{\theta}_{y,2} \ \boldsymbol{\theta}_{y,1}]$. The clustering was made on the system output variable $y(k)$. The parameters for the identification

Table 3 Simulation

Criterion	FMRLC	2 DOF FMPC
SAE [V s]	3.9542	3.1606
SSE [V^2 s]	1.2743	0.8856
$SAdU$ [V]	257	80
$SSdU$ [V^2]	1308	110
max $t_{s,0.025\text{V}}$ [s]	1.68	1.40
max $t_{s,0.05\text{V}}$ [s]	1.60	1.20
max OS [V]	0.1028	0.0075

Comparison of reference-tracking quality for different criteria

were set to the recommended values presented in Section 4.2. The identification process returned 11 clusters for both the simulated and the real system. To transform the fuzzy model into Eq. 16 we followed the scheme in Eq. 17 for choosing the state-space variables, so the time-varying state-space matrices are:

$$\boldsymbol{A}(k) = \begin{bmatrix} \boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{y,1} + \boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{y,2} & -\boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{y,2} \\ \boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{y,1} + \boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{y,2} - 1 & -\boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{y,2} \end{bmatrix},$$

$$\boldsymbol{B}(k) = \begin{bmatrix} \boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{u,1} \\ \boldsymbol{\beta}^T(k)\boldsymbol{\theta}_{u,1} \end{bmatrix}. \tag{27}$$

The controller parameters were set to: prediction horizon $h = 5$, reference error-model matrix $\boldsymbol{A}_r = 0.65\boldsymbol{I}$, states weight $\boldsymbol{Q} = \text{diag}([10\ 1\ 1]^T)$ and input weight $R = 0.1$, where $\boldsymbol{I}$ is an identity matrix and diag$(\cdot)$ a diagonal matrix.

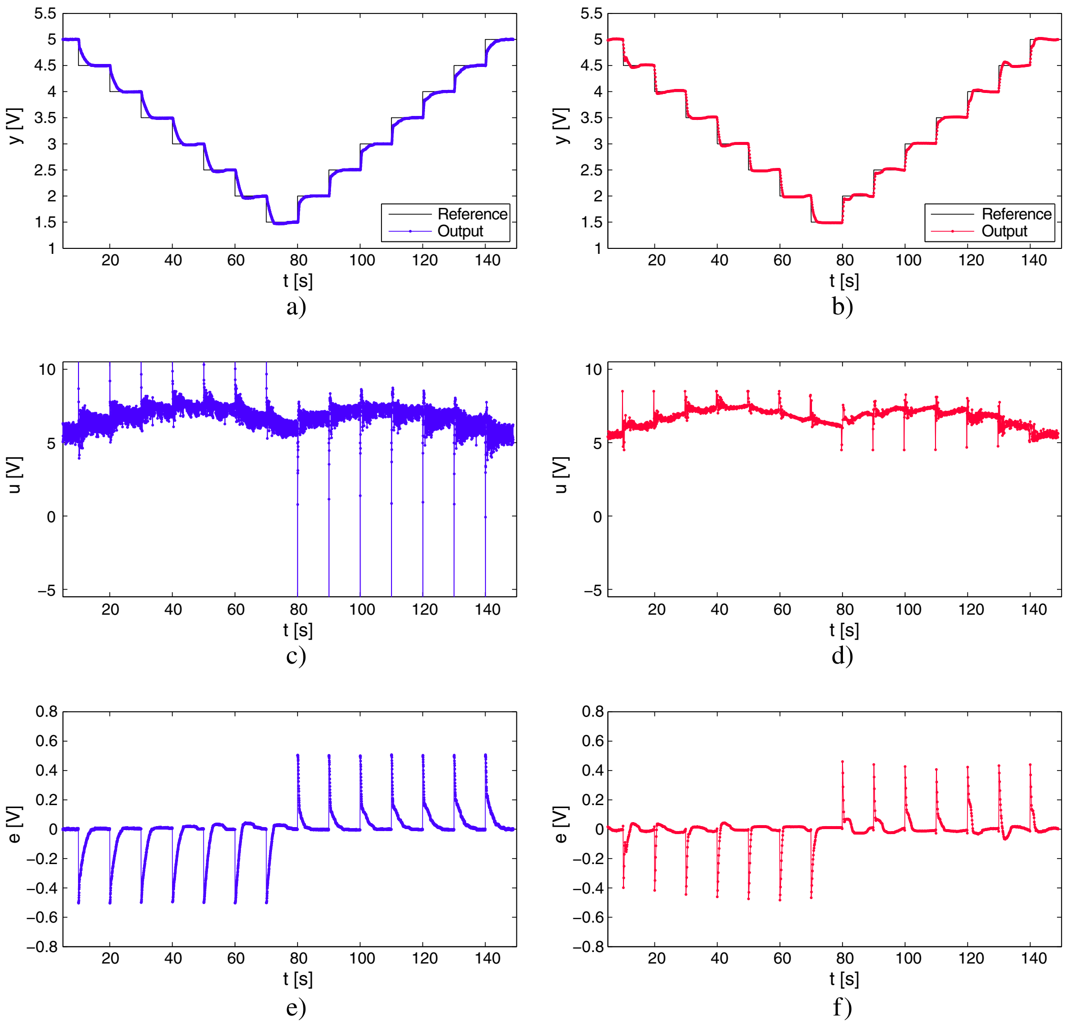


Fig. 8 Real system. Comparison of FMRLC (*left column*) and 2 DOF FMPC (*right column*); **a**, **b** reference signal tracking, **c**, **d** control action and **e**, **f** error

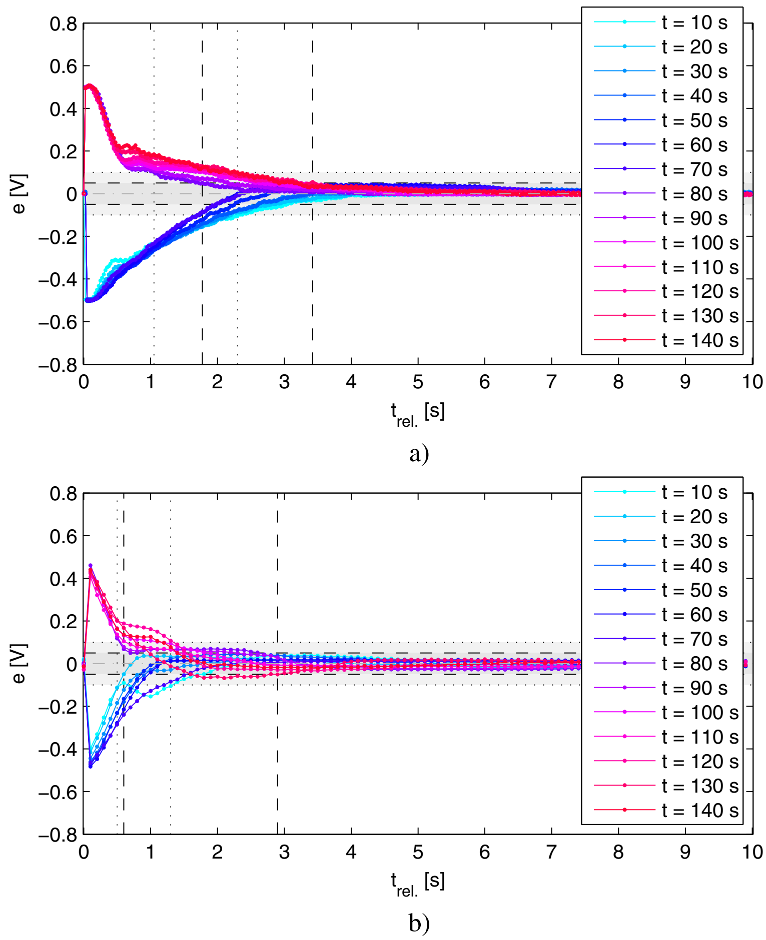


Fig. 9 Real system. Error signals around all the reference steps plotted in the same time frame for **a** FMRLC and **b** 2 DOF FMPC

6.3 Reference Tracking

The reference signal was chosen to be a stairs-like signal with the length of each step equal to 10 s and the height of each step equal to 0.5 V, from 5 V to 1.5 V and then back to 5 V. In this way the helio-crane goes through the whole range of possible inclinations. Due to the negative characteristic of the sensor output, a lower sensor output means a higher helio-crane inclination (see Table 2c).

Figure 6 shows the input and output signals from the simulation experiment for both control algorithms. Around all the reference changes a 10 s time window was selected and the error signal from all the time windows is shown in one relative time frame Fig. 7. In Fig. 7 the minimum and maximum settling times are marked with vertical lines for two different error regions $\sigma \in \{0.025\text{ V},\ 0.05\text{ V}\}$ that correspond to 5 and 10 % of a single reference step height, respectively. The values of all the criteria used for the comparison of the control algorithms are collected in Table 3.

The results from the real system are shown in Figs. 8, 9 and Table 4. The error regions used in the calculation of the settling time were selected to be twice as large as in the simulation $\sigma \in \{0.05\text{ V},\ 0.1\text{ V}\}$.

Table 4 Real system

Criterion	FMRLC	2 DOF FMPC
SAE [V s]	8.3072	5.1093
SSE [V^2 s]	2.0584	0.9200
$SAdU$ [V]	3625	329
$SSdU$ [V^2]	5885	233
max $t_{s,0.05\text{V}}$ [s]	3.43	2.90
max $t_{s,0.1\text{V}}$ [s]	2.30	1.30
max OS [V]	0.0450	0.0682

Comparison of reference-tracking quality for different criteria

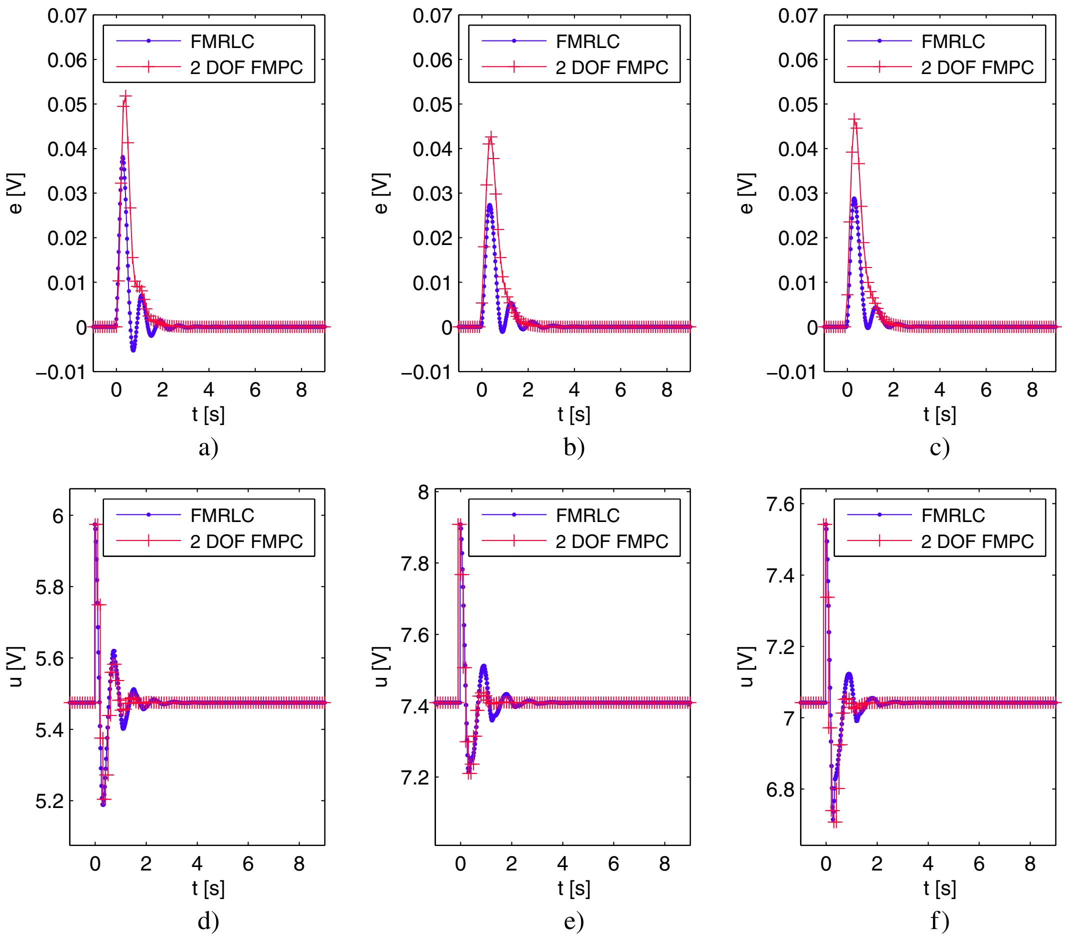


Fig. 10 Simulation. Comparison of input disturbance rejection at three operating points: **a**–**c** outputs and **d**–**f** inputs around reference values of 5 V, 3.5 V and 2 V, respectively.

6.4 Disturbance Rejection

The controllers were also compared to see how good they are at suppressing an input disturbance. To the input signal a constant disturbance of 0.5 V was added at three different operating points: 5, 3.5 and 2 V. The experiments were only conducted in the simulation environment. The error regions used in the calculation of the settling time were selected in accordance with the amplitude of the disturbance response, $\sigma \in \{0.005\ \text{V},\ 0.01\ \text{V}\}$. The results are shown in Fig. 10 and Table 5.

Table 5 Simulation. Comparison of input disturbance rejection by different criteria

Criterion	FMRLC	2 DOF FMPC
SAE [V s]	0.0478	0.0880
SSE [V^2 s]	0.0009	0.0027
$SAdU$ [V]	5.93	5.02
$SSdU$ [V^2]	0.91	1.44
max $t_{s,0.005\ \text{V}}$ [s]	1.32	1.20
max $t_{s,0.01\ \text{V}}$ [s]	0.62	0.90
max OS [V]	0.0381	0.0518

7 Conclusion

In this paper FMRLC with a modified adaptation mechanism was compared with 2 DOF FMPC control on a non-linear second-order SISO system helio-crane.

The simulation results were comparable for both controllers, as seen in Fig. 6. Nevertheless, the 2 DOF FMPC gives better results for all the criteria considered (see Table 3). Even though the output of the 2 DOF FMPC is more damped than in the FMRLC, the maximum settling time is shorter. Both control algorithms perform better when the helio-crane is being lowered (see Fig. 7). A slight overshoot is observed in the case of FMRLC when the reference inclination angle is increasing. In contrast to the input action of FMRLC, which crosses the input constraints of 0 V to 10 V, 2 DOF FMPC does not violate the input constraints.

The simulation results of the disturbance rejection reveal that both control algorithms are equally good at suppressing input disturbances. However, FMRLC gives a slightly lower overshoot. The output overshoot is relatively small (an output change of 0.05 V corresponds to an approximately 2° change in the helio-crane inclination) considering the input disturbance was 0.5 V.

The results obtained on the real system are also in favor of the 2 DOF FMPC in all the criteria considered, except for the maximum overshoot (see Table 4), which is a little smaller in the case of the FMRLC. The comparison of experimental results made on the real system with the results from the simulation environment reveal that the values of some criteria are of different magnitude (see Tables 3 and 4). The increase of the output tracking error is mainly due to the measurement noise and some form of inherent dead zone of the helio-crane, effects that were not considered in the simulation environment. The values of the criteria that evaluate the control effort ($SAdU$ and $SSdU$) are also much larger in the experiments made on the real system, especially in the case of the FMRLC. The ratio of these values between the simulation environment and the real system is mainly connected to the closed-loop system susceptibility to noise. In this aspect the 2 DOF FMPC gives better results. The settling times measured in the simulation and real environment are not comparable, since different error regions were selected. It is clear that the 2 DOF FMPC achieves better reference tracking performance on the real system. Furthermore, the 2 DOF FMPC requires significantly less control effort that the FMRLC (see Fig. 8c and Table 4), and yet it achieves better reference-tracking performance.

The control algorithms were only tested on a SISO system, since the control algorithms were developed to work with this class of the systems. In the future work we will try to extend our control algorithms to at least some classes of multiple-input multiple-output systems. Overall, we can conclude that the 2 DOF FMPC gives better results than the FMRLC. Nevertheless, the presented comparison revealed some open issues that should be addressed in order to further improve the considered control algorithms. The 2 DOF FMPC control gives satisfactory results, but the sampling time was set to only 0.1 s, since the current implementation (object-oriented code in MATLAB) does not allow for much shorter sampling times. The results show that the 2 DOF FMPC is a little worse at suppressing input disturbances than FMRLC, so this could also be improved in the future. The experiments on the real system revealed that the FMRLC algorithm demands high-gain actions in the presence of noise, and this problem needs to be addressed in the future.

Acknowledgements The work was carried out in the frame of the bilateral project between the Czech Republic and Slovenia, Advanced Algorithms for Adaptive Fuzzy Control Systems Design, BI-CZ/11-12-003. The authors would like to thank the Department of Research Infrastructure and International Cooperation, Slovenian Research Agency (ARRS), Slovenia, and the Ministry of Education, Youth and Sports, Czech Republic, for their support. Special thanks go to prof. dr. A. Belič, Faculty of Electrical Engineering, University of Ljubljana, Slovenia, for providing the detailed documentation about the helio-crane, including the system figures, the mathematical model and the measurements of the input-output characteristic functions.

References

1. Zadeh, L.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
2. Procyk, T., Mamdani, E.: A linguistic self-organizing process controller. Automatica **15**(1), 15–30 (1979)
3. Shao, S.: Fuzzy self-organizing controller and its application for dynamic processes. Fuzzy Sets Syst. **26**(2), 151–164 (1988)
4. Layne, J., Passino, K.: Fuzzy model reference learning control. In: First IEEE Conference on Control Applications, pp. 686–691. IEEE (1992)
5. Åström, K.J., Wittenmark, B.: Adaptive Control, 2nd edn. Dover Publications (2008)

6. Yuan, Y., Feng, Y., Gu, W.: Fuzzy model reference learning control for aircraft pitch autopilot design. In: 8th Control, Automation, Robotics and Vision Conference, ICARCV 2004, vol. 3, pp. 1923–1927. IEEE (2004)
7. Layne, J., Passino, K.: Fuzzy model reference learning control. J. Intell. Fuzzy Syst. **4**(1), 33–47 (1996)
8. Ling, D.: Fuzzy model reference learning control for a nonlinear model of hydrogenerator unit. In: International Workshop on Intelligent Systems and Applications, pp. 1–4. IEEE (2009)
9. Zhong, L.: Application of fuzzy model reference learning control in hydraulic power sets. J. Power Eng. **5**, 680–683 (2006)
10. Sheppard, M., Tarbouchi, M.: Design and implementation of a stable fuzzy model reference learning controller applied to a rigid-link manipulator. In: IEEE International Conference on Industrial Technology, vol. 3, pp. 1186–1191. IEEE (2004)
11. Moudgal, V., Kwong, W., Passino, K., Yurkovich, S.: Fuzzy learning control for a flexible-link robot. IEEE Trans. Fuzzy Syst. **3**(2), 199–210 (1995)
12. Zhen, L., Xu, L.: Fuzzy learning enhanced speed control of an indirect field-oriented induction machine drive. IEEE Trans. Control Syst. Technol. **8**(2), 270–278 (2000)
13. Rehman, H.u., Dhaouadi, R.: A fuzzy learning–sliding mode controller for direct field-oriented induction machines. Neurocomputing **71**(13–15), 2693 (2008)
14. Ismail, A.: Fuzzy model reference learning control of multi-stage flash desalination plants. Desalination **116**(2–3), 157–164 (1998)
15. Oltean, S.E., Grif, H., Duka, A.V.: Electron beam focusing and deflection control systems. In: Interdisciplinarity in Engineering—Scientific International Conference, pp. 1–6 (2007)
16. Zhang, J.l., Ning, N., Zheng, M.y.: Application of fuzzy model reference learning control in dynamic hot-box device. J. Harbin Inst. Technol. **4**(2), 36–47 (2006)
17. Xu, S., Pan, H.: Coke oven temperature control based on fuzzy model reference learning control. Control and Instruments in Chemical Industry **3**(2), 10–16 (2008)
18. Su, Y., Zheng, C., Duan, B.: Fuzzy learning tracking of a parallel cable manipulator for the square kilometre array. Mechatronics **15**(6), 731–746 (2005)
19. Naceri, A., Ramdani, Y., Bounouna, H.: A fuzzy model reference learning controller of ASMES to improve transient power system stability. Mediterranean J. Measure. Control **3**(3), 126–133 (2007)
20. Kwong, W., Passino, K.: Dynamically focused fuzzy learning control. IEEE Trans. Syst. Man Cybern. Part B **26**(1), 53–74 (1996)
21. Dessouky, A., Tarbouchi, M.: Model reference adaptive fuzzy controller for induction motor using auto-attentive approach. In: Proceedings of the 2000 IEEE International Symposium on Industrial Electronics, vol. 2, pp. 719–723. IEEE (2000)
22. Oltean, S., Abrudean, M., Gligor, A.: MRAC and FMRLC for a plant with time varying parameters. In: IEEE International Conference on Automation, Quality and Testing, Robotics, vol. 1, pp. 62–67. IEEE (2006)
23. Kopasakis, G.: Adaptive performance seeking control using fuzzy model reference learning control and positive gradient control. In: 33rd Joint Propulsion Conference and Exhibit, pp. 1–9 (1997)
24. Passino, K.M., Yurkovich, S.: Fuzzy Control. Addison-Wesley (1998)
25. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **15**(1), 116–132 (1985)
26. Chiu, S.L.: Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Syst. **2**, 267–278 (1994)
27. Angelov, P.P., Filev, D.P.: An approach to online identification of Takagi-Sugeno fuzzy models. IEEE Trans. Syst. Man Cybern., Part B **34**(1), 484–498 (2004)
28. Memon, M.A., Angelov, P., Ahmed, H.: An approach to real-time color-based object tracking. In: International Symposium on Evolving Fuzzy Systems, pp. 86–91. IEEE (2006)
29. Angelov, P., Zhou, X.: Evolving fuzzy systems from data streams in real-time. In: International Symposium on Evolving Fuzzy Systems, pp. 29–35. IEEE (2006)
30. Angelov, P., Filev, D.: Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In: 14th IEEE International Conference on Fuzzy Systems, pp. 1068–1073. Reno, Nevada, USA (2005)
31. Angelov, P.: Evolving Intelligent Systems: Methodology and Applications, chap. Evolving Takagi-Sugeno Fuzzy Systems From Streaming Data (eTS+), pp. 21–50. Wiley, New Jersey (2010)
32. Lughofer, E., Klement, E.P.: FLEXFIS: a variant for incremental learning of Takagi-Sugeno fuzzy systems. In: 14th IEEE International Conference on Fuzzy Systems, pp. 915–920. IEEE, Reno, Nevada, USA (2005)
33. Lughofer, E., Bouchot, J.L., Shaker, A.: On-line elimination of local redundancies in evolving fuzzy systems. Evolving Syst. **2**(3), 165–187 (2011)
34. Kasabov, N.K.: Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. IEEE Trans. Syst. Man Cyber. Part B **31**(6), 902–918 (2001)
35. Kasabov, N.: Evolving fuzzy neural networks — algorithms, applications and biological motivation. In: Methodologies for the Conceptation, Design and Application of Soft Computing, pp. 271–274. Japan (1998)
36. Kasabov, N.K., Song, Q.: DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans. Fuzzy Syst. **10**(2), 144 (2002)
37. Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cyber. **23**(3), 665–685 (1993)
38. Azeem, M.F., Hanmandlu, H., Ahmad, N.: Structure identification of generalized adaptive neuro-fuzzy inference systems. IEEE Trans. Fuzzy Syst. **11**(5), 666–681 (2003)
39. Leng, G., Prasad, G., McGinnity, T.M.: An on-line algorithm for creating self-organizing fuzzy neural networks. Neural Netw. **17**(10), 1477–1493 (2004)

40. Rong, H.J., Sundararajan, N., Huang, G.B., Saratchandran, P.: Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. Fuzzy Sets Syst. **157**(9), 1260–1275 (2006)
41. Lin, F.J., Lin, C.H., Shen, P.H.: Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. IEEE Trans. Fuzzy Syst. **9**(5), 751–759 (2001)
42. Lin, C.T.: A neural fuzzy control system with structure and parameter learning. Fuzzy Sets Syst. **70**(2–3), 183–212 (1995)
43. Wu, S., Er, M.J.: Dynamic fuzzy neural networks — a novel approach to function approximation. IEEE Trans. Syst. Man Cyber. Part B **30**(2), 358–364 (2000)
44. Wu, S., Er, M.J., Gao, Y.: A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. IEEE Trans. Fuzzy Syst. **9**(4), 578–594 (2001)
45. Juang, C.F., Lin, C.T.: An online self-constructing neural fuzzy inference network and its applications. IEEE Trans. Fuzzy Syst. **6**(1), 12–32 (1998)
46. Tzafestas, S.G., Zikidis, K.C.: NeuroFAST: on-line neuro-fuzzy art-based structure and parameter learning TSK model. IEEE Trans. Syst. Man Cyber. Part B **31**(5), 797–802 (2001)
47. Platt, J.: A resource-allocating network for function interpolation. Neural Comput. **3**(2), 213–225 (1991)
48. Deng, D., Kasabov, N.: Evolving self-organizing maps for on-line learning, data analysis and modeling. In: Proc. IJCNN 2000 Neural Networks Neural Computing: New Challenges Perspectives New Millenium, vol. 6, pp. 3–8. IEEE, New York (2000)
49. Fritzke, B.: A growing neural gas network learns topologies. Adv. Neural Inf. Process. Syst. **7**, 845–865 (1995)
50. Soleimani-B, H., Lucas, C., Araabi, B.N.: Recursive Gath–Geva clustering as a basis for evolving neuro-fuzzy modeling. Evolving Syst. **1**(1), 59–71 (2010)
51. Dovžan, D., Škrjanc, I.: Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. ISA Trans. **50**(2), 159–169 (2011)
52. Dovžan, D., Škrjanc, I.: Recursive clustering based on a Gustafson–Kessel algorithm. Evolving Syst. **2**(1), 15–24 (2011)
53. Dovžan, D., Škrjanc, I.: Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor. Control Eng. Pract. **18**(8), 979–989 (2010)
54. Normey-Rico, J.E., Gòmez-Ortega, J., Camacho, E.F.: A Smith-predictor-based generalised predictive controller for mobile robot path-tracking. Control Eng. Pract. **7**(6), 729–740 (1999)
55. Rouhani, R., Mehra, R.K.: Model algorithmic control (MAC); basic theoretical properties. Automatica **18**(4), 401–414 (1982)
56. Richalet, J., O'Donovan, D.: Predictive functional control: principles and industrial applications. Advances in Industrial Control. Springer (2009)
57. Richalet, J.: Industrial applications of model based predictive control. Automatica **29**(5), 1251–1274 (1993)
58. Škrjanc, I., Matko, D.: Predictive functional control based on fuzzy model for heat-exchanger pilot plant. IEEE Trans. Fuzzy Syst. **8**(6), 705–712 (2000)
59. Lepetič, M., Škrjanc, I., Chiacchiarini, H.G., Matko, D.: Predictive functional control based on fuzzy model: comparison with linear predictive functional control and PID control. J. Intell. Robot. Syst. **36**(4), 467–480 (2003)
60. Blažič, S., Škrjanc, I.: Design and stability analysis of fuzzy model-based predictive control — a case study. J. Intell. Robot. Syst. **49**(3), 279–292 (2007)
61. Karer, G., Mušič, G., Škrjanc, I., Zupančič, B.: Feedforward control of a class of hybrid systems using an inverse model. Math. Comput. Simul. **82**(3), 414–427 (2011)
62. Malchow, F., Sawodny, O.: Model based feedforward control of an industrial glass feeder. Control Eng. Pract. **20**(1), 62–68 (2012)
63. Oriolo, G., De Luca, A., Vendittelli, M.: WMR control via dynamic feedback linearization: design, implementation, and experimental validation. IEEE Trans. Control Syst. Technol. **10**(6), 835–852 (2002)
64. Klančar, G., Škrjanc, I.: Tracking-error model-based predictive control for mobile robots in real time. Robot. Auton. Syst. **55**(6), 460–469 (2007)
65. Škrjanc, I., Klančar, G.: Optimal cooperative collision avoidance between multiple robots based on Bernstein-Beézier curves. Robot. Auton. Syst. **58**(1), 1–9 (2010)
66. Pao, L.Y., Butterworth, J.A., Abramovitch, D.Y.: Combined feedforward/feedback control of atomic force microscopes. In: American Control Conference, pp. 3509–3515. IEEE (2007)
67. Chiu, C.S.: Mixed feedforward/feedback based adaptive fuzzy control for a class of MIMO nonlinear systems. IEEE Trans. Fuzzy Syst. **14**(6), 716–727 (2006)
68. Wu, Y., Zou, Q.: Robust inversion-based 2-DOF control design for output tracking: piezoelectric-actuator example. IEEE Trans. Control Syst. Technol. **17**(5), 1069–1082 (2009)
69. Passino, K.: Biomimicry for Optimization, Control, and Automation. Springer (2004)
70. Cerman, O., Hušek, P.: Fuzzy model reference learning control with modified adaptation mechanism. In: Seventh International Conference on Natural Computation (ICNC), vol. 1, pp. 123–128. IEEE (2011)
71. de Jesús Rubio, J.: Evolving Intelligent Systems: Methodology and Applications, chap. Stability analysis for an online evolving neuro-fuzzy recurrent network, pp. 173–199. Wiley, Hoboken, New Jersey (2010)
72. Blažič, S., Škrjanc, I., Gerkšič, S., Dolanc, G., Strmčnik, S., Hadjiski, M.B., Stathaki, A.: Online fuzzy identification for an intelligent controller based on a simple platform. Eng. Appl. Artif. Intell. **22**(4–5), 628–638 (2009)
73. Blazinšek, M., Arh, M., Škrjanc, I.: Comparison between the auto-tuning of PID, PFC and modified-PFC controllers. In: 7th Vienna Conference on Mathematical Modelling, MATHMOD 2012, pp. 1–6 (2012)