


AUTHOR QUERY FORM


 ELSEVIER	Journal: ASOC	Please e-mail or fax your responses and any corrections to:
	Article Number: 2104	E-mail: corrections.esch@elsevier.thomsondigital.com
	Fax: +353 6170 9272	

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult <http://www.elsevier.com/artworkinstructions>.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

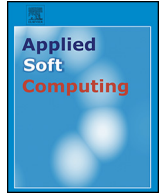
Location in article	Query / Remark: click on the Q link to go Please insert your reply or correction at the corresponding line in the proof
Q1 Q2	<p>Please confirm that given names and surnames have been identified correctly.</p> <p>Please check the telephone number of the corresponding author, and correct if necessary. </p> <div style="border: 1px solid black; padding: 10px; margin-top: 20px;"> <p>Please check this box or indicate your approval if you have no corrections to make to the PDF file <input type="checkbox"/></p> </div>

Thank you for your assistance.



Contents lists available at ScienceDirect

Applied Soft Computing

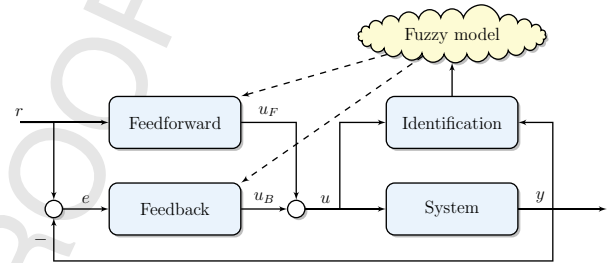
journal homepage: www.elsevier.com/locate/asoc

Graphical Abstract

Self-tuning of 2 DOF control based on evolving fuzzy model

A. Zdešar*, D. Dovžan, I. Škrjanc

Applied Soft Computing xxx (2013) xxx–xxx



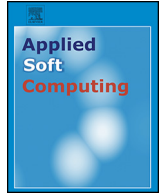
UNCORRECTED PROOF



ELSEVIER

Contents lists available at [ScienceDirect](#)

Applied Soft Computing

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

Highlights

Self-tuning of 2 DOF control based on evolving fuzzy model

Applied Soft Computing xxx (2013) xxx–xxx

A. Zdešar*, D. Dovžan, I. Škrjanc

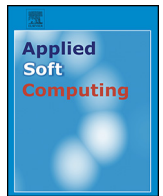
- The feedforward and feedback loops of the control algorithm are self-tuned based on Takagi-Sugeno fuzzy model.
- The evolving fuzzy model employs evolving mechanisms for adding, removing, merging and splitting the clusters.
- The experimental results confirm that the proposed control approach is capable of fast and accurate reference tracking.
- The simulation and real experiments show that the closed-loop step response of the presented approach is invariant to the operating point.

UNCORRECTED PROOF



Contents lists available at ScienceDirect

Applied Soft Computing

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

Self-tuning of 2 DOF control based on evolving fuzzy model

Q1 A. Zdešar*, D. Dovžan, I. Škrjanc

Laboratory of Autonomous Mobile Systems, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:

Received 12 October 2012
Received in revised form 10 October 2013
Accepted 12 October 2013
Available online xxx

Keywords:

Takagi-Sugeno fuzzy model
Evolving fuzzy model
Gustafson-Kessel clustering
recursive fuzzy least squares
2 DOF control
model predictive control
NARX

ABSTRACT

In this paper we present a self-tuning of two degrees-of-freedom control algorithm that is designed for use on a non-linear single-input single-output system. The control algorithm is developed based on the Takagi-Sugeno fuzzy model, and it consists of two loops: a feedforward loop and feedback loop. The feedforward part of the controller should drive the system output to the vicinity of the reference signal. It is developed from the inversion of the T-S fuzzy model. To achieve accurate error-free reference tracking a feedback part of the controller is added. A time-varying error-model predictive controller is used in the feedback loop. The error-model is obtained from the T-S fuzzy model. The T-S fuzzy model of the system, required in the controller, is obtained with evolving fuzzy modelling, which is based on recursive Gustafson-Kessel clustering algorithm and recursive fuzzy least squares. It employs evolving mechanisms for adding, removing, merging and splitting the clusters.

The presented control approach was experimentally validated on a non-linear second-order SISO system helio-crane in simulation and real environment. Several criteria functions were defined to evaluate the reference-tracking and disturbance rejection performance of the control algorithm. The presented control approach was compared to another fuzzy control algorithm. The experimental results confirm the applicability of the approach.

© 2013 Published by Elsevier B.V.

1. Introduction

Fuzzy-set theory was proposed by Zadeh [1], and has since been successfully applied to a diverse range of applications, mostly in the fields of control and artificial intelligence. The use of fuzzy data instead of crisp-valued data has proved to be a convenient way of solving problems that are highly non-linear or are hard to solve with traditional approaches. The way to use fuzzy logic to derive a controller was first shown by Procyk and Mamdani [2], where fuzzy sets were used to describe inputs to the controller as well as output actions. The Takagi-Sugeno (T-S) fuzzy system [3] that was proposed later, uses only some mathematical function of the input variables to describe an output.

Fuzzy model represents a convenient way to describe the system behaviour. Furthermore, Takagi-Sugeno fuzzy systems are thought of as universal approximators [4, p. 77], since every system can be represented to an arbitrary precision in the form of a T-S fuzzy model.

To identify the T-S model the structure and the parameters of the local models must be identified [3]. 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 [5].

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 were developed. Some of them are based on fuzzy logic (eTS [6], exTS [7,8], simple.TS [9], +eTS [10], FLEXFIS [11], FLEXFIS+[12], eTS based on switching to neighbouring models [13]), others use neural networks that realize the behaviour of the fuzzy model (EFuNN [14,15], DENFIS [16], ANFIS [17], GANFIS [18], SOFNN [19], SAFIS [20], SCFNN [21], NFCN [22], D-FNN [23], GD-FNN [24], SONFIN [25], NeuroFAST [26], RAN [27], ESOM [28], Neural gas [29], ENFM [30], and GAP-RBF [31]).

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 generalized fuzzy model (GANFIS).

The methods also differ in their ability of adapting 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 consequent and premise parameters (*adaptive methods* [18,17]). Some of the methods include a mechanism for adding new clusters to the model structure (*incremental methods* [6]). Recently proposed methods also include

Q2 * Corresponding author. Tel.: +386 14768950.
E-mail addresses: andrej.zdesar@fe.uni-lj.si (A. Zdešar),
dejan.dovzan@fe.uni-lj.si (D. Dovžan), igor.skrjanc@fe.uni-lj.si (I. Škrjanc).

mechanisms for merging, removing and splitting clusters. The methods use different clustering algorithms, such as ECM ([16]), recursive subtractive clustering ([6]), Gath-Geva clustering ([30]) 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) is used for the fuzzy-model identification. The method is based on the recursive Gustafson-Kessel clustering algorithm [32,33] and recursive fuzzy least squares [6]. It employs evolving mechanisms for adding, removing, merging and splitting the clusters. This method was also used in [34] for constructing the adaptive fuzzy predictive functional controller for a semi-batch reactor.

A model of a system in an appropriate form can represent a valuable insight into the system internal behaviour. The control algorithms that take the model of a system into account implicitly belong to a class of internal model control [35]. The model of a system can be used to make predictions about the system behaviour, and can though be used to determine the optimal control actions that take the system dynamics and constraints into account, the approach known as *predictive control* [36–38]. Over the years many different predictive control algorithms have been developed: generalized predictive control (GPC) [39], model algorithmic control (MAC) [40], predictive functional control (PFC) [41], model-based predictive control (MPC) [42], 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 [43,44], fuzzy model-based predictive control (FMBPC) [45], 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 [46–49]. 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 *two degrees-of-freedom* (2 DOF) control [35]. The feedforward part should provide a fast reaction to reference changes and drive the output into the vicinity of the reference. The feedback part 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. The 2 DOF control structure was proposed for a class of integral processes [50], a class of MIMO non-linear systems [51], a class of hybrid systems [46], among others. The 2 DOF control approach was successfully implemented in a diverse range of applications: trajectory tracking [52–54], swing-up of the double pendulum on a cart [55], locomotion control of a hydraulically actuated hexapod robot [56], motion of a magnetic suspension positioning stage [57], control of a fire-rescue turntable ladder [58], temperature control of industrial refineries furnaces [59], temperature control in a batch reactor [46], etc.

One of the goals of the paper is to present the usefulness of the evolving fuzzy modelling for solving the control problems. In this paper we present a 2 DOF fuzzy control design that is self-tuning based on fuzzy model of the system. This means that the control structure and parameters are determined implicitly based on the given fuzzy model of the system. Therefore, self-tuning can be achieved. Another goal is to achieve set-point invariant reference tracking. In this paper we consider that both parts of the 2 DOF controller are designed based on the (inverse) Takagi-Sugeno fuzzy model. The feedback part of the control algorithm is designed in the model predictive scheme based on the fuzzy model. The control algorithm is developed for SISO non-linear systems. We also present the evolving mechanisms for adding, removing, merging and splitting the clusters in the evolving fuzzy model.

The outline of the paper is as follows. Section 2 presents a fuzzy model in T-S form, and Section 3 describes the method of evolving fuzzy modelling. This is followed by Section 4, which presents the 2 DOF control scheme. Section 5 gives a description and mathematical modelling of a helio-crane system, the system that was used for evaluation of the presented control algorithm. Section 6 presents experimental results made in simulation environment and on real system, and gives a comparison of the proposed control algorithm to a fuzzy predictive functional controller. Afterwards, a discussion on the results is given in Section 7 and finally, Section 8 concludes the paper with some ideas for future development.

2. Fuzzy model

In this section the notation used to describe a fuzzy system is presented. An arbitrary system can be described with a set of K fuzzy rules $\{\mathcal{R}^j\}_{j=1,\dots,K}$ in Takagi-Sugeno fuzzy form, where the rule \mathcal{R}^j is defined as:

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

The antecedents of the rules (IF parts) describe the fuzzy regions in the space of input variables. For the description of the input-output dynamics in THEN parts of the rules (consequences) the nonlinear auto regressive model with exogenous inputs (NARX) is selected, but any other model could be used as well. The NARX model predicts the next output based on the past inputs and outputs. Furthermore, we assume that every THEN part of each fuzzy rule can be approximated with an affine NARX model:

$$f_j(k) = \theta_j^T \psi(k) \quad (2)$$

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

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

$$y(k+1) = \beta^T(k) \Theta^T \psi(k) \quad (3)$$

where $\beta^T(k)$ represents the normalized degrees of fulfillment for the whole set of fuzzy rules $\{\mathcal{R}^j\}_{j=1,2,\dots,K}$ in the current time step, written in the vector form $\beta^T(k) = [\beta_1(k) \ \beta_2(k) \ \dots \ \beta_K(k)]$. We assume the normalized degrees of fulfillment, which are generally time dependent, comply with Eq. (4) for every time step k .

$$\sum_{j=1}^K \beta_j(k) = 1 \quad (4)$$

In Eq. (3) the matrix $\Theta \in \mathbb{R}^{1+m+n} \times \mathbb{R}^K$ contains all the parameters of the fuzzy model for the whole set of rules $\{\mathcal{R}^j\}_{j=1,2,\dots,K}$: $\Theta = [\theta_1 \ \theta_2 \ \dots \ \theta_K]$.

3. Evolving fuzzy model

The evolving fuzzy model is based on recursive Gustafson-Kessel 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 (\mathbf{x}_f) (clustering data vector) and the other is for local model-parameter estimation (\mathbf{x}_r) (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 change of the center position is calculated as:

$$\Delta v_i(k+1) = \frac{(\mu_{ik+1})^\eta (\mathbf{x}_f(k+1) - v_i(k))}{s_i(k+1)} \quad (5)$$

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

$$s_i(k+1) = \gamma_v s_i(k) + \mu_{ik+1}^\eta \quad (6)$$

The initial $s_i(0)$ is usually set to one. With the initialization of $s_i(0)$ to a higher value, the initial speed of adaptation slows down. The $\gamma_v \in [0, 1]$ is the forgetting factor (value of one turns off forgetting and values lower than one increase the forgetfulness). The forgetting factor can be set using the rule of thumb Eq. (21). The centers are then moved for $\Delta v_i(k+1)$:

$$v_i(k+1) = v_i(k) + \Delta v_i(k+1) \quad i = 1, \dots, c. \quad (7)$$

The index i is the index of the cluster and c is the number of clusters. The membership μ_{ik+1} of the current clustering vector \mathbf{x}_f is calculated as:

$$\mu_{ik+1} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik+1}}{d_{jk+1}} \right)^{\frac{2}{\eta-1}}} \quad (8)$$

where d_{ik+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 is defined as:

$$d_{ik+1}^2 = (\mathbf{x}_f(k+1) - v_i(k))^T \mathbf{A}_i (\mathbf{x}_f(k+1) - v_i(k)) \quad (9)$$

where \mathbf{A}_i depends on the distance that we use. For the Euclidean distance \mathbf{A}_i is the unit matrix (fuzzy c -means algorithm), and using the Mahalanobis distance \mathbf{A}_i is equal to the inverse fuzzy covariance matrix. For the distance used with the GK clustering algorithm the \mathbf{A}_i is equal to the fuzzy inverse matrix and the determinant:

$$\mathbf{A}_i = \rho_i \det(\mathbf{F}_i)^{1/z} \mathbf{F}_i^{-1} \quad (10)$$

where ρ_i is usually set to a value of one and z is the number of variables (number of elements of \mathbf{x}_f).

The fuzzy covariance is calculated using the following equation:

$$\mathbf{F}_i(k+1) = \gamma_c \frac{s_i(k)}{s_i(k+1)} \mathbf{F}_i(k) + \frac{\mu_{ik+1}^\eta}{s_i(k+1)} \mathbf{D}_{F_i} \quad (11)$$

$$\mathbf{D}_{F_i} = (\mathbf{x}_f(k+1) - v_i(k+1)) (\mathbf{x}_f(k+1) - v_i(k+1))^T \quad (12)$$

When using the GK clustering algorithm there is a need for a calculation of the recursive inverse fuzzy covariance matrix and the determinant. The equation for the inverse fuzzy variance matrix is the following:

$$[\mathbf{F}_i(k+1)]^{-1} = \frac{1}{\gamma_c} \frac{s_i(k+1)}{s_i(k)} \left[[\mathbf{F}_i(k)]^{-1} - \frac{\mathbf{B}}{\mathbf{C}} \right] \quad (13)$$

$$\mathbf{B} = [\mathbf{F}_i(k)]^{-1} \mathbf{D}_{F_i} [\mathbf{F}_i(k)]^{-1} \quad (14)$$

$$\mathbf{C} = \gamma_c \frac{s_i(k)}{\mu_{ik+1}^\eta} + d_{F_i}^T [\mathbf{F}_i(k)]^{-1} d_{F_i} \quad (15)$$

$$d_{F_i} = \mathbf{x}_f(k+1) - v_i(k+1) \quad (16)$$

The equation for the recursive determinant calculation is given as:

$$\det(\mathbf{F}_i(k+1)) = \left(\gamma_c \frac{s_i(k)}{s_i(k+1)} \right)^z \det(\mathbf{F}_i(k)) (1 + A) \quad (17)$$

$$A = \frac{1}{\gamma_c} \frac{\mu_{ik+1}^\eta}{s_i(k)} (d_{F_i}^T [\mathbf{F}_i(k)]^{-1} d_{F_i}) \quad (18)$$

where z is the number of elements of the clustering vector. A more detailed derivation of these equations can be found in [33].

Once the clusters are updated the fuzzy recursive least squares are used to update the linear local models parameters. There are different algorithms proposed [60,63,61], and these are based on weighted recursive least squares. The equation for adaptation based on [61] are:

$$\psi_i(k+1) = [1, \mathbf{x}_k(k+1)^T]^T \quad (19)$$

$$y(k+1) = y(k+1) \quad (20)$$

$$\mathbf{P}_i(k+1) = \frac{1}{\lambda_r} \left(\mathbf{P}_i(k) - \frac{\beta_i \mathbf{P}_i(k) \psi_i(k+1) \psi_i^T(k+1) \mathbf{P}_i(k)}{\lambda_r + \beta_i \psi_i^T(k+1) \mathbf{P}_i(k) \psi_i(k+1)} \right) \quad (20)$$

$$\theta_i(k+1) = \theta_i(k) + \mathbf{P}_i(k) \beta_i \psi_i(k+1) (y(k+1) - \psi_i^T(k+1) \theta_i(k)) \quad (21)$$

The parameters of the i th sub-models are denoted as θ_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. For setting the forgetting factor λ_r one can use the rule of thumb presented in [62]:

$$\lambda = 1 - \frac{2}{N}, \quad (21)$$

where N is the number of samples that have an influence on estimation. The same rule can be used for setting the forgetting factors for the clustering algorithm γ_c and γ_v . It was found that setting γ_c equal to γ_v gives more true approximation of fuzzy covariance matrix. For β_i the membership degrees that are calculated during the cluster update can be used. However, in general they do not have a smooth transition. When identifying a process with a smooth nonlinearity it is better to recalculate the membership degrees using a Gaussian function. For this we can use radial base functions:

$$\mu_{ij} = e^{-\frac{d_{ij}^2}{2\eta_m \sigma_{ij}^2}} \quad (22)$$

where d_{ij} is the distance between the j th component of the i th cluster and the j th component of the clustering data vector:

$$d_{ij}^2 = (x_{f_j} - v_{i_j})^2 \quad (23)$$

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). The membership degree β_i is calculated as a product of the membership degree of each component:

$$\beta_i = \prod_{j=1}^z \mu_{ij} \quad (24)$$

where z is the number of components of the clustering vector \mathbf{x}_f . Note that these membership degrees β_i should be normalized as in Eq. (4). The settings for the parameters are given in [34] and [33]. For the fine tuning of the local model parameters the concept of instrumental variables can be used, as presented in [61].

$$\mathbf{P}_i(k+1) = \frac{1}{\lambda_r} \left(\mathbf{P}_i(k) - \frac{\beta_i \mathbf{P}_i(k) \psi_{m_i}(k+1) \psi_{m_i}^T(k+1) \mathbf{P}_i(k)}{\lambda_r + \beta_i \psi_{m_i}^T(k+1) \mathbf{P}_i(k) \psi_{m_i}(k+1)} \right) \quad (25)$$

$$\theta_i(k+1) = \theta_i(k) + \mathbf{P}_i(k) \beta_{m_i} \psi_{m_i}(k+1) (y(k+1) - \psi_{m_i}^T(k+1) \theta_i(k)) \quad (25)$$

where $\psi_{m_i}(k+1)$ is the regression vector, where the delayed process outputs (y) are replaced by the delayed model outputs (y_m): $\psi_{m_i}(k+1) = [1 \ u(k-m+1) \ \dots \ u(k) \ y_m(k-n+1) \ \dots \ y_m(k)]^T$. The β_{m_i} are the membership degrees of the clustering vector, where the delayed process outputs are replaced by the delayed model outputs in the same way as with the regression vector.

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. Also, a condition is added to prevent the moving of clusters to which the membership degree of the current clustering vector is lower than a certain threshold (the beta-cut condition). This threshold is set somewhere between 0.1 and 0.3. The evolving mechanisms are briefly described in the following subsections.

3.1. Adding of clusters

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. In general, different distances can be used. In our case we used the normalized Euclidean distance for each component of the clustering vector. The normalized distance is defined as:

$$d_{ijnorm} = \frac{d_{ij}}{k_n \sqrt{f_{ij}}} \quad (26)$$

$$d_{ij} = |x_{f_j} - v_{ij}|$$

where i is the index of the closest cluster and j is the index of the component. The normalization constant k_n is user defined and is, in general, set to 2 and f_{ij} is the j th diagonal element of i th cluster's fuzzy covariance matrix. In our case the normalized distances were then summed together:

$$D_{sum} = \sum_{j=1}^z d_{ijnorm} \quad (27)$$

A new cluster can be added if the value of D_{sum} is greater than z , where z is the number of clustering-vector components.

To prevent the adding of clusters based on outliers an additional condition [63] was introduced. For the cluster to be added at least N consecutive samples must satisfy the distance adding condition. The number N is normally set from 3 to 10, depending on the data. This condition lowers the possibility of adding the cluster based on outliers.

3.1.1. Initialization of a new cluster

The new cluster center is positioned at the current clustering vector. The fuzzy covariance matrix is initialized as a diagonal matrix where the distances to the closest cluster are considered. The diagonal elements are defined as:

$$f_{new_{jj}} = -\frac{d_{ij}^2}{2 \ln(\epsilon_\beta)} \quad (28)$$

where ϵ_β is a user-defined constant that is normally set to 0.15. If the distance d_{ij} is smaller than the standard deviation ($\sqrt{f_{ij}}$), then a diagonal element is equal to the diagonal element of the closest cluster ($f_{new_{jj}} = f_{ij}$). In the same manner an initial fuzzy covariance for the first cluster can be set by considering the expected number of clusters and data range. Next, the parameters of a new local model can be initialized using a weighted mean:

$$\theta_{i+1_j} = \frac{\sum_{i=1}^c \omega_i \theta_{ij}}{\sum_{i=1}^c \omega_i} \quad (29)$$

where i is the index of the cluster and j is the parameter index. The weights ω_{ij} can be equal to the normalized membership degrees of the other clusters or combined with the variance of the parameters:

$$\omega_{ij} = \beta_i \frac{1}{\sigma_{p_{ij}}^2} \quad (30)$$

where $\sigma_{p_{ij}}^2$ is the j th diagonal element of the least-squares covariance matrix of the i th cluster. The index c defines the number of cluster that satisfy the beta-cut condition.

3.2. Removing of clusters

The removing of clusters in our case is based on their support. Two conditions were implemented. The first one is minimal existence condition. This condition is implemented to remove the clusters, that were added based on outliers. In order for the cluster to stay in the structure it must gather a certain amount of support samples N_{strh} after a certain period after its creation k_{delay} . The N_{strh} is usually set to 15 and k_{delay} to 5 [10]. The support N_{s_i} is usually defined as the number of samples that have the maximum membership of a certain cluster. The second condition is based on cluster's support/age coefficient (S_n). The condition is meant to remove old clusters, that are no longer valid, from the structure. The cluster is removed if its support normalized with its age is lower than a fraction of the mean support/age coefficients. The age a_i is defined as the number of samples from the creation k_i and the current sample k :

$$a_i = k - k_i \quad (31)$$

$$S_{n_i} = \frac{N_{s_i}}{a_i} \quad (32)$$

Both conditions for removing can be written as:

$$\text{IF } S_{n_i} < \epsilon \text{mean}_{i=1, \dots, c}(S_{n_i}) \text{ OR } (N_{s_i} < N_{strh} \text{ AND } k > k_i + k_{delay}) \quad (33)$$

THEN remove i th cluster.

where S_{n_i} is the i th cluster's support/age coefficient, $\text{mean}_{i=1, \dots, c}(S_{n_i})$ is the mean value of support/age coefficient of all clusters and ϵ is the user defined constant lower than one (usually 0.1).

3.3. Merging of clusters

The merging algorithm is meant to merge the clusters that are either close or have the same local model parameters. The eFuMo method considers merging based on the membership degrees of clusters to each other [30] (unsupervised) and merging based on the correlation method [12] (supervised).

In eFuMo the unsupervised merging is based on a normalized Mahanalobis distance:

$$d_{iknorm} = \frac{((v_k - v_i)^T F_i^{-1} (v_k - v_i))^{0.5}}{(\mathbf{d}_{inorm}^T F_i^{-1} \mathbf{d}_{inorm})^{0.5}} \quad (34)$$

$$\mathbf{d}_{inorm} = [\sqrt{f_{i11}} \quad \sqrt{f_{i22}} \quad \dots \quad \sqrt{f_{izz}}]^T \quad (35)$$

The clusters are merged if the distances d_{iknorm} and $d_{ki norm}$ are below a threshold k_d and the ratio between the distances is lower than the threshold k_r .

$$\text{IF } d_{iknorm} < k_d \text{ AND } d_{ki norm} < k_d \text{ AND } |1 - \frac{d_{iknorm}}{d_{ki norm}}| < k_r \quad (36)$$

THEN merge i th and k th cluster.

The threshold k_d is defined based on the allowed maximum membership degree ϵ_β of the k th cluster to the i th cluster and vice versa:

$$k_d = \sqrt{-2 \ln(\epsilon_\beta)} \quad (37)$$

The k_r should be set to around 0.1 and the ε_β to around 0.85. This rule only merges close clusters.

In general, the clusters can be further apart, but still have the same shape and the parameters of the local model. To merge these kinds of clusters the supervised merging algorithm is used. A correlation between the past membership degrees ($C_{beta_{ik}}$) of the clustering vectors is monitored, as in [12]. If the correlation is higher than a threshold value (usually 0.9), the clusters are considered for merging. This condition is combined with the condition of the angles between the parameters (ϕ_{ik}) and support for the clusters from the time of the last change in the structure (Nsl_i). This means that each time some of the evolving mechanisms change the number of clusters the correlation value and the support Nsl_i are reset. The two clusters are merged if they have Nsl_i larger than a user defined threshold, the angles between the local models parameters are small and there are no other clusters between them. The angles of local models are calculated differently as in [12]. Using the angle condition as presented in [12] we found that in many cases clusters, that do not have similar parameters are often merged together. Therefore we implemented a different version of the condition. First the algorithm searches all local models for the absolute maximal value of each local model's parameter. The parameters of the two local models that are considered for merging are normalized and the angle difference is calculated for each parameter.

$$\alpha_j = |\arctan(\theta_{n_{kj}}) - \arctan(\theta_{n_{ij}})|, \quad (38)$$

where k and i are the indexes of the two clusters considered for merging, j is the index of the parameter and θ_n is the normalized parameter. If angle differences α_j for all parameters are below the user defined threshold (usually 2 degrees), the clusters are merged together.

The initialization of a new cluster position and variance is made as suggested in [30].

3.4. Splitting of clusters

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-covariance matrix. The new positions are calculated as:

$$\begin{aligned} v_{i1} &= v_i + \mathbf{d}f_{i_{norm}\lambda} \\ v_{i2} &= v_i + \mathbf{d}f_{i_{norm}\lambda} \end{aligned} \quad (39)$$

This approach to initialization is also presented in [64]. Initialization can also be based on the eigenvectors of the fuzzy covariance matrix [65]. This fuzzy covariance matrix is divided by two, and the threshold is time decaying. The user specifies the maximum and minimum values and the decay constant for the threshold, the same as in [23]. The error is set to zero when the number of clusters changes.

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. The scheme of the algorithm is shown in Fig. 1.

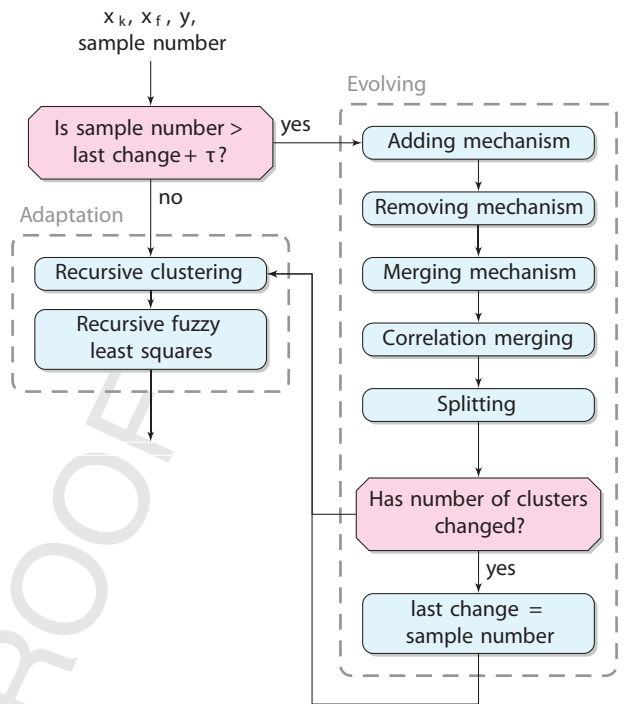


Fig. 1. Flowchart of a single step in the identification of evolving fuzzy model.

4. 2 DOF fuzzy model predictive control

The main goal of the presented control algorithm is to provide precise error-free reference-signal tracking. The control structure is designed in a way that all the system dependent parameters of the control algorithm can be obtained from the fuzzy model of the system. Since the fuzzy model can be obtained with the identification algorithm presented in Section 3, the control algorithm is considered to be self-tuned. The control structure uses a two degree-of-freedom concept: a combination of feedforward and feedback loops; and fuzzy model is used in the design of both control loops. An overview of the control structure is depicted in Fig. 2. Once the fuzzy model of the system is identified, it can be fed to the controller and the controller can be switched on. The identification is made on the open-loop system, and once the controller is switched on the identification is switched off, and the evolving fuzzy model is frozen.

4.1. Feedforward control

To implement a feedforward reference tracking controller the inverse model of the system needs to be known. The derivation of an inverse model may not be an easy task and the solution may not even be feasible. In this section we present a summary of an

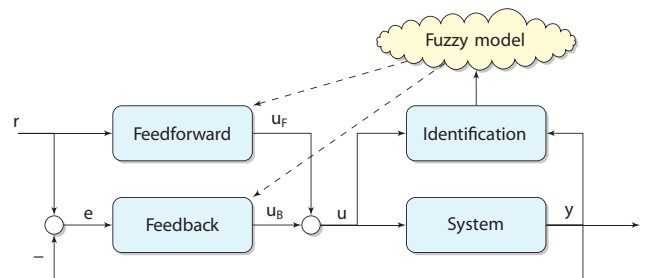


Fig. 2. Structure of 2 DOF fuzzy model predictive control.

approach to the inversion of a T-S fuzzy model introduced by Karer et al. [46].

First, we break apart the matrix of model parameters $\Theta^T = [\theta_r \ \Theta_u^T \ \theta_{u,1} \ \Theta_y^T]$ and the regression vector $\psi^T(k) = [1 \ \psi_u^T(k) \ u(k) \ \psi_y^T(k)]$ of a fuzzy model Eq. (3), where $\theta_r \in \mathbb{R}^K$, $\Theta_u \in \mathbb{R}^{m-1} \times \mathbb{R}^K$, $\theta_{u,1} \in \mathbb{R}^K$, $\Theta_y \in \mathbb{R}^n \times \mathbb{R}^K$ and $\psi_u \in \mathbb{R}^{m-1}$, $\psi_y \in \mathbb{R}^n$. Notice the slight abuse of notation, since θ in this section does not correspond to the θ defined in Section 2. In the new extended notation the fuzzy system Eq. (3) is rewritten as:

$$y(k+1) = \beta^T(k) \left(\theta_r + \Theta_u^T \psi_u(k) + \theta_{u,1} u(k) + \Theta_y^T \psi_y(k) \right) \quad (40)$$

We assume the model Eq. (40) is capable of following the reference signal. Then, there should exist a time-series of feedforward inputs $u_F(k)$ that achieves an error-free reference signal tracking of the model $y(k) = r(k)$ for all the $k > 0$. If we substitute in Eq. (40) all the outputs y for the appropriate reference values r and all the inputs u for the previous feedforward inputs u_F , then the feedforward input $u_F(k)$ can be calculated:

$$u_F(k) = \frac{r(k+1) - \beta_r^T(k) (\theta_r + \Theta_u^T \psi_{u_F}(k) + \Theta_y^T \psi_r(k))}{\beta_r^T(k) \theta_{u,1}} \quad (41)$$

where $\psi_{u_F}^T(k) = [u_F(k-m+1) \ \dots \ u_F(k-1)]$ and $\psi_r^T(k) = [r(k-n+1) \ \dots \ r(k)]$. We have denoted the fulfillment vector with a subscript r to emphasize that the vector belongs to the model of the reference signal. The feedforward input $u_F(k)$ does not depend on the system measurements, but only on the undisturbed reference signal and previous feedforward inputs. The fulfillment vector $\beta_r(k)$ is calculated based on the reference signal.

The Eq. (41) clearly violates the causality constraint, since the optimum input in time-step k is dependent on the future reference signal at time step $k+1$. This means that the feedforward control has an inherent delay of one sample time.

An important role in the calculation of the feedforward input $u_F(k)$ is played by the shape of the reference signal r . It should be noted that some reference signals (too frequency rich) may push the solution of the system Eq. (41) 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 in the reference signal. When selecting the filter cut-off frequency we can take into account the system dynamics and the upper and lower signal bounds of the input signal.

Although the fuzzy model may be stable and invertible, the inverse of the fuzzy model could become unstable. This happens when the model has non-minimum phase. As a consequence, the system cannot be driven to the desired state by the feedforward controller alone. Naturally, the feedforward control cannot be implemented on inherently unstable systems. Therefore, in a case of unstable and non-minimum phase systems the feedforward control approach is not suitable. The feedforward approach could still be used on unstable systems, but the system must further be enclosed in a closed loop that is responsible for ensuring system stability. Note, that the feedback controller should be able to take the feedforward actions into account when determining the feedback corrections.

4.2. Model-based predictive feedback control

In a 2 DOF control design, the control action is a sum of the feedforward and feedback signals, $u(k) = u_F(k) + (-u_B(k))$. Here, we make an assumption that the fulfillment vectors of the reference signal and system model are equal $\beta_r(k) = \beta(k)$ for all the steps $k > 0$. This means that we do the clustering either on the reference signal or on the system output for the calculation of the feedforward and

feedback actions. Substituting Eq. (41) into Eq. (40) the following relation can be expressed:

$$e(k+1) = \beta^T(k) \left(\Theta_u^T \psi_{u_B}(k) + \theta_{u,1} u_B(k) + \Theta_y^T \psi_e(k) \right) \quad (42)$$

where we have introduced new variables: the predicted error $e(k+1) = r(k+1) - y(k+1)$, the vector of the past feedback actions $\psi_{u_B}(k) = \psi_{u_F}(k) - \psi_u(k)$ and the vector of the past errors $\psi_e(k) = \psi_r(k) - \psi_y(k)$. The fuzzy error-model Eq. (42) can be transformed into a time-varying state-space form:

$$\begin{aligned} \mathbf{x}_e(k+1) &= \mathbf{A}_e(k) \mathbf{x}_e(k) + \mathbf{b}_e(k) u_B(k) \\ e(k) &= \mathbf{c}_e^T(k) \mathbf{x}_e(k) \end{aligned} \quad (43)$$

We can conclude that the dynamic behaviour of the feedback controller depends on the dynamics of the system model. When transforming the fuzzy error model into the state-space error model, the selection of the state-space variables should be according to the following scheme:

$$\mathbf{x}_e(k) = \begin{bmatrix} e(k) \\ e(k) - e(k-1) \\ e(k) - 2e(k-1) + e(k-2) \\ \vdots \\ u_B(k-1) \\ u_B(k-1) - u_B(k-2) \\ u_B(k-1) - 2u_B(k-2) + u_B(k-3) \\ \vdots \end{bmatrix} \quad (44)$$

where every element in $\mathbf{x}_e \in \mathbb{R}^{n+m-1}$ except for the first and the $(n+1)$ th is calculated as a difference of the previous element at two consecutive past time steps.

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

$$\boldsymbol{\epsilon}(k+1) = \begin{bmatrix} \mathbf{A}_e(k) & \mathbf{0} \\ \mathbf{c}_e^T(k) & 1 \end{bmatrix} \boldsymbol{\epsilon}(k) + \begin{bmatrix} \mathbf{b}_e(k) \\ 0 \end{bmatrix} u_B(k) = \mathbf{A}(k) \boldsymbol{\epsilon}(k) + \mathbf{b}(k) u_B(k) \quad (45)$$

where $\mathbf{A} \in \mathbb{R}^N \times \mathbb{R}^N$, $\mathbf{b} \in \mathbb{R}^N$, $N = n + m$.

The error model Eq. (45) is used in the development of the predictive feedback control. In order to obtain the optimal control algorithm, we define a criterion function that penalizes the error and control action on a finite prediction horizon $h \in \mathbb{N}$:

$$J(u_B) = \sum_{i=1}^h \xi^T(k+i|k) \mathbf{Q} \xi(k+i|k) + R u_B^2(k+i-1|k) \quad (46)$$

where we have introduced $\xi(k+i|k) = \boldsymbol{\epsilon}_r(k+i|k) - \boldsymbol{\epsilon}(k+i|k)$ as the difference between the reference error $\boldsymbol{\epsilon}_r^T(k+i|k)$ and the predicted error $\boldsymbol{\epsilon}(k+i|k)$ for all the $i \in \{1, 2, \dots, h\}$ time-steps ahead of the current time-step k . In Eq. (46), the $\mathbf{Q} \in \mathbb{R}^N \times \mathbb{R}^N$ and $R \in \mathbb{R}$, and $\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} \geq 0$ for every $\boldsymbol{\alpha} \in \mathbb{R}^N$, and $R \geq 0$. The development of the control law by minimization of the cost function Eq. (46) has previously been shown in [53]. Here, we give a summary of the approach that is rewritten for the SISO systems.

Using the model Eq. (45) we can make a prediction of the error $\boldsymbol{\epsilon}(k+i|k)$ as a function of the current error $\boldsymbol{\epsilon}(k)$ and the unknown

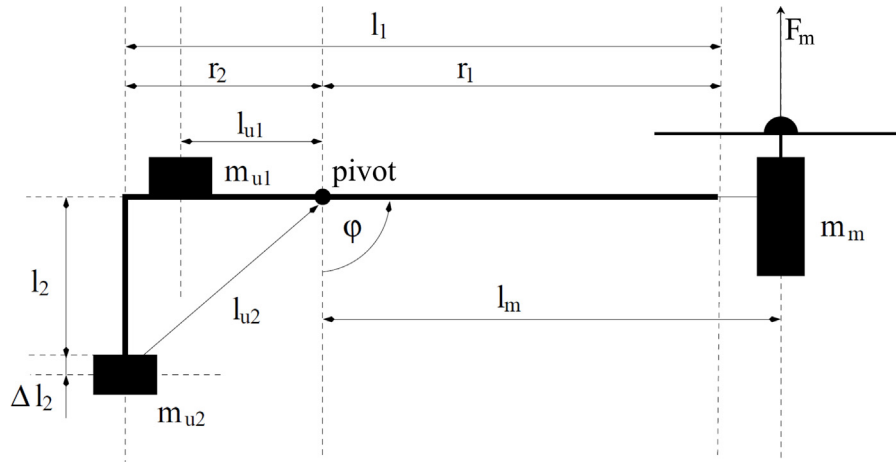


Fig. 3. Schematic model of a helio-crane.

feedback inputs $u_B(k+i-1|k)$ for $i=1, \dots, h$:

$$\epsilon(k+i|k) = \mathbf{A}_{i,0}(k)\epsilon(k) + \sum_{j=1}^i \mathbf{A}_{i,j}(k)\mathbf{b}(k+j-1|k)u_B(k+j-1|k) \quad (47)$$

where we have defined $\mathbf{A}_{i,j}(k) = \mathbf{A}(k+i-1|k) \dots \mathbf{A}(k+j+1|k)\mathbf{A}(k+j|k)$. It should be noted that $\mathbf{A}_{i,j}(k) = \mathbf{I}$ for all the $j \geq i$, where \mathbf{I} is the unity matrix. All the predicted states can be gathered in an augmented vector $\bar{\epsilon}^T(k+1) = [\epsilon^T(k+1|k) \ \epsilon^T(k+2|k) \ \dots \ \epsilon^T(k+h|k)]$, and all the future inputs in another augmented vector $\bar{\mathbf{u}}_B^T(k) = [u_B(k|k) \ u_B(k+1|k) \ \dots \ u_B(k+h-1|k)]$. Now, the following extended system can be written:

$$\bar{\epsilon}(k+1) = \bar{\mathbf{A}}(k)\bar{\epsilon}(k) + \bar{\mathbf{B}}(k)\bar{\mathbf{u}}_B(k) \quad (48)$$

where the matrix $\bar{\mathbf{A}}(k) \in \mathbb{R}^{hN} \times \mathbb{R}^{hN}$ is defined as:

$$\bar{\mathbf{A}}^T(k) = [\mathbf{A}_{1,0}^T(k) \ \mathbf{A}_{2,0}^T(k) \ \dots \ \mathbf{A}_{h,0}^T(k)] \quad (49)$$

and the matrix $\bar{\mathbf{B}}(k) \in \mathbb{R}^{hN} \times \mathbb{R}^h$ is defined as:

$$\bar{\mathbf{B}}(k) = \begin{bmatrix} \mathbf{b}(k|k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_{2,1}(k)\mathbf{b}(k|k) & \mathbf{b}(k+1|k) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{h,1}(k)\mathbf{b}(k|k) & \mathbf{A}_{h,2}(k)\mathbf{b}(k+1|k) & \dots & \mathbf{b}(k+h-1|k) \end{bmatrix} \quad (50)$$

The cost function Eq. (46) can be rewritten in a more compact form:

$$J(\bar{\mathbf{u}}_B) = \bar{\xi}^T(k+1)\bar{\mathbf{Q}}\bar{\xi}(k+1) + \bar{\mathbf{u}}_B^T(k)\bar{\mathbf{R}}\bar{\mathbf{u}}_B(k) \quad (51)$$

where we have introduced $\bar{\xi}(k+1) = \bar{\epsilon}_r(k+1) - \bar{\epsilon}(k+1)$ as the difference between the reference error $\bar{\epsilon}_r^T(k+1) = [\epsilon_r^T(k+1|k) \ \epsilon_r^T(k+2|k) \ \dots \ \epsilon_r^T(k+h|k)]$ and the predicted error $\bar{\epsilon}^T(k+1)$ on the prediction horizon h . The weight matrices are given as $\bar{\mathbf{Q}} = \text{oplus}_{i=1}^h \mathbf{Q}$ and $\bar{\mathbf{R}} = \text{oplus}_{i=1}^h \mathbf{R}$, and are both required to be positive semi-definite. The operator $\text{oplus}_i(\cdot)$ denotes the direct matrix sum.

The reference error is usually defined as an exponentially decreasing function in the form of a state-space model: $\epsilon_r(k+i|k) = \mathbf{A}_r^i \epsilon(k)$ for $i \in \{1, 2, \dots, h\}$. The model of the reference error on the

entire prediction horizon h is then $\bar{\epsilon}(k+1) = \bar{\mathbf{A}}_r(k)\bar{\epsilon}(k)$, where a new augmented matrix has been introduced:

$$\bar{\mathbf{A}}_r^T(k) = [\mathbf{A}_r^T(k) \ (\mathbf{A}_r^2(k))^T \ \dots \ (\mathbf{A}_r^h(k))^T] \quad (52)$$

To obtain the optimum control law, we need to find the minimum of the objective function Eq. (51):

$$\frac{\partial J(\bar{\mathbf{u}}_B)}{\partial \bar{\mathbf{u}}_B} = -2\bar{\mathbf{B}}^T \bar{\mathbf{Q}} (\bar{\mathbf{A}}_r - \bar{\mathbf{A}}) \bar{\epsilon} + 2(\bar{\mathbf{B}}^T \bar{\mathbf{Q}} \bar{\mathbf{B}} + \bar{\mathbf{R}}) \bar{\mathbf{u}}_B = \mathbf{0} \quad (53)$$

From the Eq. (53) the optimum control inputs are obtained:

$$\bar{\mathbf{u}}_{B,opt.}(k) = (\bar{\mathbf{B}}^T(k)\bar{\mathbf{Q}}\bar{\mathbf{B}}(k) + \bar{\mathbf{R}})^{-1} \bar{\mathbf{B}}^T(k)\bar{\mathbf{Q}} (\bar{\mathbf{A}}_r(k) - \bar{\mathbf{A}}(k)) \bar{\epsilon}(k) \quad (54)$$

Since the term in the last parentheses in Eq. (53) is positive semi-definite, the obtained control input Eq. (54) is at the minimum of the objective function Eq. (51), and therefore really the optimum control input. According to the receding horizon strategy, at the time step k only the $u_{B,opt.}(k|k) = [1 \ 0 \ \dots \ 0]\bar{\mathbf{u}}_{B,opt.}(k)$ is added to the feedforward input signal $u_F(k)$, and, in spirit of the predictive control, during the next time step the whole procedure is repeated again.

The control algorithm can be tuned with several different parameters. The weight matrix \mathbf{Q} and scalar R determine how strictly should the predicted error 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 \mathbf{A}_r and the length of the prediction horizon h . These parameters define the desired response dynamics and power consumption.

The presented control algorithm takes the system dynamics into account inherently, in terms of the given fuzzy model. Since the model of the system is obtained by identification of evolving fuzzy model, the control approach is considered to be self-tuned. The other tunable parameters presented in the paragraph above determine the desired goals of the control algorithm, and are relatively easy to tune. When the system goes through the input-output space, these additional parameters do not need to be changed and still the behaviour of the closed-loop system should stay the same regardless of the operating point. This means that even when we are dealing with non-linear systems, the closed-loop response should be invariant to the operating point, provided a good enough model of the plant is available.

The stability study of fuzzy systems is a difficult task. Even if the fuzzy model has locally asymptotically stable subsystems, the system can be still globally unstable [66]. The presented control algorithm can achieve stable closed-loop response only on a certain

class of non-linear SISO systems. The approach was designed to be used on non-linear SISO systems with simple dynamics. The system can be inherently unstable, but it must not have non-minimum phase, or else some additional measures must be taken to achieve stable behaviour. In a case of non-minimum phase system, the fuzzy model of the system used in the feedforward loop must be converted to appropriate minimum-phase system, otherwise the closed-loop system becomes unstable. This is only a necessary condition for closed loop stability and appropriate system behaviour. Although, the stability of the closed loop could be checked with the method of linear matrix inequalities (LMI) as in [67], this is not a simple task, since a complex set of equations has to be solved. Furthermore, if the set of conditions for stability cannot be solved, the system could still be stable, since the approach of LMI only gives sufficient conditions for stability.

5. Helio-crane

The system that was chosen for the evaluation of the fuzzy-control algorithm is composed of a rigid metal rod on a pivot that can swing in a single vertical plane like a pendulum (Fig. 3). The swing of one end of the rod is physically restricted to a vertical half-plane, so the ends of the rod can 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 0V to 10V. 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 0V to 10V. 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:

$$J\ddot{\varphi} = T(\varphi) - f\dot{\varphi} \quad (55)$$

where φ is the inclination of the rod, T is the sum of all 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 physical dimensions of the helio-crane, applying some basic knowledge for the calculation of the partial moments of inertia. According to Fig. 3, the moment of inertia J is:

$$J = \rho_d \left(\frac{1}{3} l_1 (r_1^2 - r_1 r_2 + r_2^2) + l_2 \left(\frac{l_2^2}{3} + r_2^2 \right) \right) + m_m l_m^2 + m_{u1} l_{u1}^2 + m_{u2} l_{u2}^2 \quad (56)$$

and the torque T is:

$$T = F_m l_m + g \sin(\varphi) \left(\rho_d \left(\frac{r_2^2}{2} - \frac{r_1^2}{2} \right) + m_{u1} l_{u1} - m_m l_m \right) + \rho_d l_{2d} g \sin(\varphi + \psi_1) + m_{u2} g l_{u2} \sin(\varphi + \psi_2) \quad (57)$$

where $l_{2d} = l_2 \sqrt{\frac{l_2^2}{4} + r_2^2}$, $\tan \psi_1 = \frac{l_2}{2r_2}$ and $\tan \psi_2 = \frac{l_2 + \Delta l_2}{r_2}$. The variables r_1 , r_2 , l_1 , l_2 , Δl_2 , l_{u1} , l_{u2} , l_m represent the lengths as denoted in Fig. 3; m_m , m_{u1} and m_{u2} are the masses of the motor and the two weights, respectively; g is the gravitational acceleration; and

Table 1

Helio-crane (a) model parameters (see Fig. 3), (b) input and (c) output characteristic function data.

Symbol	Value	Units
g	9.81	m s^{-2}
ρ_d	1.13	kg m^{-1}
m_m	0.13	kg
m_{u1}	0.145	kg
m_{u2}	0.02	kg
r_1	0.34	m
r_2	0.2	m
l_{u1}	0.165	m
l_{u2}	0.2402	m
l_m	0.38	m
l_1	0.54	m
l_2	0.128	m
Δl_2	0.0051	m
l_{2d}	0.0269	m^2
ψ_1	0.3097	rad
ψ_2	0.5872	rad
f	0.15	$\text{kg m}^2 \text{s}^{-1}$
J	0.0483	kg m^2

u [V]	f_u [N]
0.0	0.0000
4.6	0.0000
4.7	0.1558
5.0	0.2727
5.5	0.4286
6.0	0.5844
7.0	0.8182
7.5	0.9351
8.0	1.0130
8.5	1.0909
10.0	1.0909

φ	f_y [V]
14°	6.00
90°	3.52
177°	0.83

ρ_d is the linear mass density of the rods. The damping factor f can be heuristically determined and is defined in Table 1(a), where all the other model parameters are also gathered. The output characteristic function $y=f_y(\varphi)$ is approximately linear (Table 1(c)), but the input characteristic function $F_m=f_u(u)$ is highly non-linear (Table 1(b)). We can conclude that the helio-crane is a SISO non-linear system with second-order dynamics, as Eq. (55) suggests.

The model describing the system, Eq. (55), can be linearised around some operating point (OP). We would like to obtain a linear mapping for different operating points between the input actuator voltage u and the output sensor voltage y . The system Eq. (55) can be rewritten into a non-linear state-space form

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), u(t)) \quad y(t) = H(\mathbf{x}(t), u(t)) \quad (58)$$

where $\mathbf{x} = [x_1 \ x_2]^T$ denotes the internal states, which we choose to be $x_1 = \varphi$ and $x_2 = \dot{\varphi}$. The linearised state-space model is obtained by Taylor expansion of Eq. (58), where all the terms higher than the first order are omitted:

$$\Delta \dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ \frac{b_1}{J} \cos(x_1) - \frac{b_2}{J} \sin(x_1) & -\frac{f}{J} \end{bmatrix} \Bigg|_{OP} \Delta \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{l_m}{J} \frac{\partial f_u}{\partial u} \end{bmatrix} \Bigg|_{OP} \Delta u(t) \quad (59)$$

$$\Delta y(t) = \begin{bmatrix} \frac{\partial f_y}{\partial x_1} & 0 \end{bmatrix} \Bigg|_{OP} \Delta \mathbf{x}(t)$$

where we have introduced two new variables $b_1 = g(\rho_d(\frac{r_2^2 - r_1^2}{2} + l_{2d} \cos(\psi_1)) + m_{u1} l_{u1} - m_m l_m + m_{u2} l_{u2} \cos(\psi_2))$ and $b_2 = g(\rho_d l_{2d} \sin(\psi_1) + g m_{u2} l_{u2} \sin(\psi_2))$.

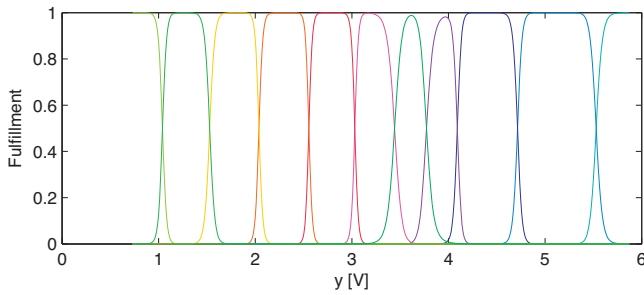


Fig. 4. Simulation. Distribution of the clusters over the system output.

Note that our approach does not require mathematical modelling of the system, since the model required for the controller is obtained by identification of the evolving fuzzy model. The derived analytical model will be used for validation of the presented identification algorithm.

6. Experimental results

The presented self-tuning of the 2 DOF fuzzy model predictive control (2 DOF FMPC) algorithm was experimentally tested on the helio-crane system introduced in Section 5. In this section, we first present the validation of the fuzzy model obtained with evolving fuzzy modelling. Next, we evaluate the performance of the presented control algorithm and show the comparison of the algorithm against another advanced control approach. At the end, experimental results obtained from the real system are presented.

6.1. Simulation

First, we validated the identification of the evolving fuzzy model with simulation of the model Eq. (55). For the identification of the evolving fuzzy model a second-order NARX model was chosen: $n = 2$ and $m = 1$. We selected the clustering variable to be the output of the system y (helio-crane inclination). The parameters for the identification were set to the recommended values presented in Section 3. We drove the helio-crane in open-loop up and down through all the achievable inclinations of the rod. The measured signals were used in the on-line identification of the system. After some time, when the number of clusters in the evolving fuzzy model settled (at eleven clusters), the fuzzy model was frozen. The final distribution of the clusters over the output space is shown in Fig. 4. To evaluate the performance of the identification, a comparison between Bode diagrams at all the operating points of the identified fuzzy model and the linearised model Eq. (59) of the helio-crane was made (Fig. 5).

Once the model of the system was obtained, the controller could be switched on. We selected the reference to be a stairs-line 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 back to 5 V. In this way the helio-crane goes through the whole range of possible inclinations. Note that due to the negative characteristic of the sensor output, a lower sensor voltage means a higher helio-crane inclination (see Table 1(c)).

The controller parameters were set to the following values: the prediction horizon $h = 5$, the reference error-model matrix $A_r = 0.65 I$, the states weight $Q = \text{diag}([10 \ 1 \ 1]^T)$ and the input weight $R = 0.1$, where I is an identity matrix and $\text{diag}(\cdot)$ is a diagonal matrix. The simulation results of the reference tracking are presented in Fig. 6. During the simulation time the degree of fulfillment of each cluster changed with time as it is shown in Fig. 7.

The performance of the presented control approach was compared to a fuzzy predictive functional control (FPFC). A short

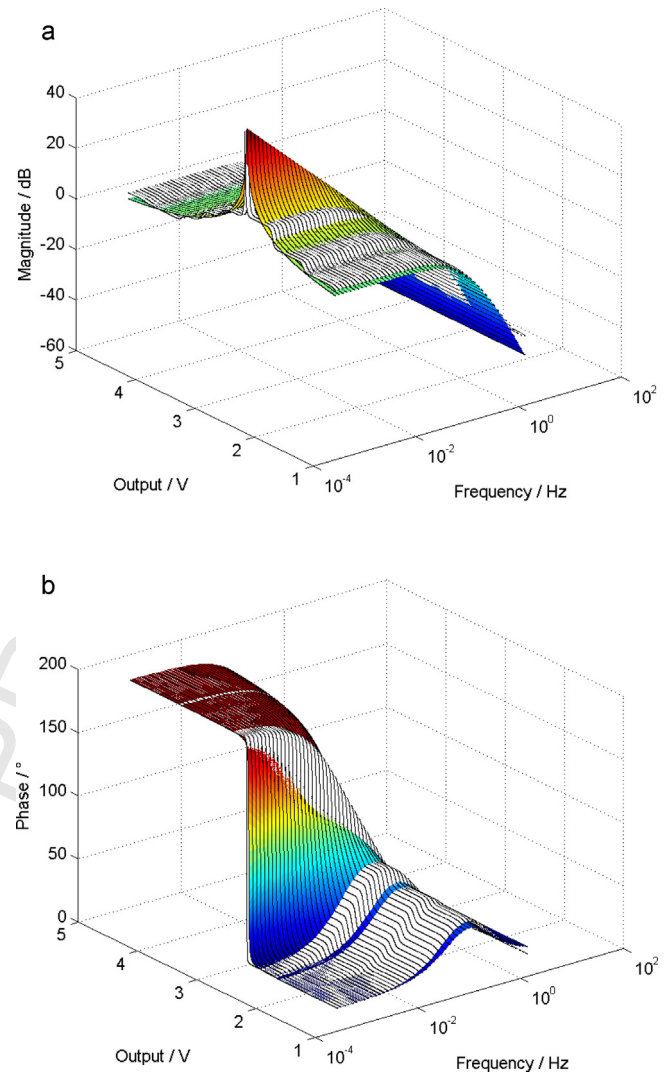


Fig. 5. Simulation. Comparison between the Bode diagrams of the system (coloured) and the identified fuzzy model (white) for different operating points.

summary of the FPFC for a second-order system is given below. The model of a second-order system $G_m(z) = \frac{b_m}{(z-a_{m1})(z-a_{m2})}$ is rewritten into a parallel form $G_m(z) = G_{m1}(z) + G_{m2}(z)$, where prediction of each of the model for the future time $k+h$, $h \in \mathbb{N}$ is as follows:

$$y_{m1}(k+h) = a_{m1}^h y_{m1}(k) + \frac{k_{m1}(1-a_{m1}^h)}{1-a_{m1}} u(k) \quad (60)$$

$$y_{m2}(k+h) = a_{m2}^h y_{m2}(k) + \frac{k_{m2}(1-a_{m2}^h)}{1-a_{m2}} u(k) \quad (61)$$

where $k_{m1} = \frac{b_m}{a_{m1}-a_{m2}}$ and $k_{m2} = \frac{-b_m}{a_{m1}-a_{m2}}$. The predicted output of the model is a sum of both predicted submodel outputs: $y_m(k+h) = y_{m1}(k+h) + y_{m2}(k+h)$. The fuzzy predictive functional control is given as:

$$u(k) = \frac{1-a_r^h e(k) + y_{m1}(k)(1-a_{m1}^h) + y_{m2}(k)(1-a_{m2}^h)}{\frac{k_{m1}(1-a_{m1}^h)}{1-a_{m1}} + \frac{k_{m2}(1-a_{m2}^h)}{1-a_{m2}}} - \frac{r_m}{b_m} \quad (62)$$

where $e(k) = w(k) - y(k) - y_m(k) + y_{md}(k)$ and a_r is the parameter of the reference signal:

$$y_r(k+h) = a_r^h y_r(k) + (1-a_r^h) w(k) \quad (63)$$

where $w(k)$ is the reference signal. The $y_{md}(k)$ is the delayed output of the model. The parameters of the model a_{m1} , a_{m2} , b_m and r_m are

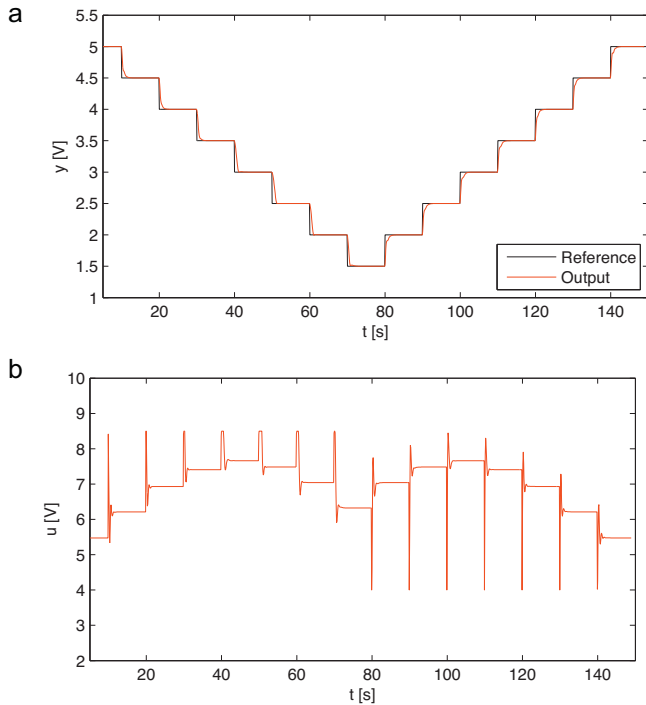


Fig. 6. Simulation of 2 DOF FMPC: (a) reference signal tracking and (b) control action.

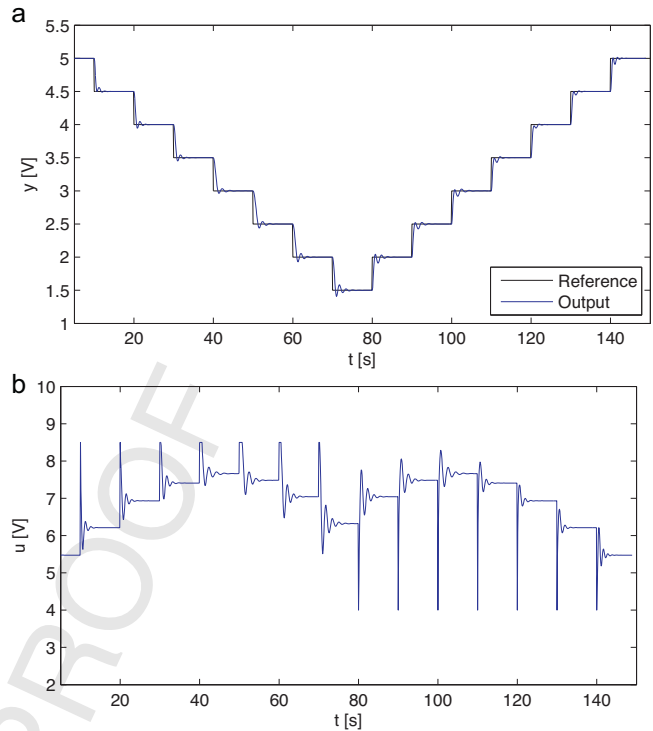


Fig. 8. Simulation of FPFC: (a) reference signal tracking and (b) control action.

obtained from the fuzzy model Eq. (1). For more information on the derivation of the control law the reader is referred to the works [68,69,34].

The design parameters of the fuzzy predictive functional control were selected to be $a_r=0.7$ and $h=5$. The reference tracking performance of the FPFC controller is shown in Fig. 8 and can be compared to the reference tracking of the proposed 2 DOF fuzzy control shown in Fig. 6. For better comparison of the results, the error is plotted in all the consecutive 10 s time windows into the same (relative) time frame (Fig. 9).

To evaluate the performance of the control algorithm several criteria functions were defined [70]. 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

$$SAE = T_s \sum_k |e(k)| \quad (64)$$

and the Sum of the Squared Error

$$SSE = T_s \sum_k e^2(k) \quad (65)$$

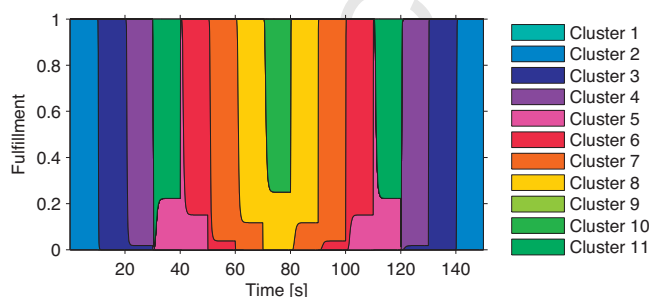


Fig. 7. Simulation. Time progress of the degrees of fulfillment for all the clusters.

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

$$SAdU = \sum_k |\Delta u(k)| \quad (66)$$

and the Sum of the Squared Input differences

$$SSdU = \sum_k \Delta u^2(k) \quad (67)$$

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 signal) and the maximum overshoot OS. If there were similar step reference changes at different operating points, the maximum settling time and overshoot from among all the responses at different operating points was selected as the criteria for the comparison, denoted as $\max t_{s,\sigma}$ and $\max OS$, respectively. In Table 2 the comparison of the reference tracking quality between the FPFC and 2 DOF FMPC in terms of different criteria is shown.

We also evaluated the input disturbance rejection performance of the presented controller and FPFC controller at three different operating points: 5 V, 3.5 V and 2 V. The results are shown in Fig. 10

Table 2
Simulation. Comparison of reference-tracking performance to different criteria.

Criterion	Units	2 DOF FMPC	FPFC
SAE	V s	3.1360	4.0305
SSE	V ² s	0.8798	1.2032
SAdU	V	86	93
SSdU	V ²	137	123
$\max t_{s,0.025V}$	s	1.50	2.80
$\max t_{s,0.05V}$	s	1.20	2.20
$\max OS$	V	0.0060	0.0932

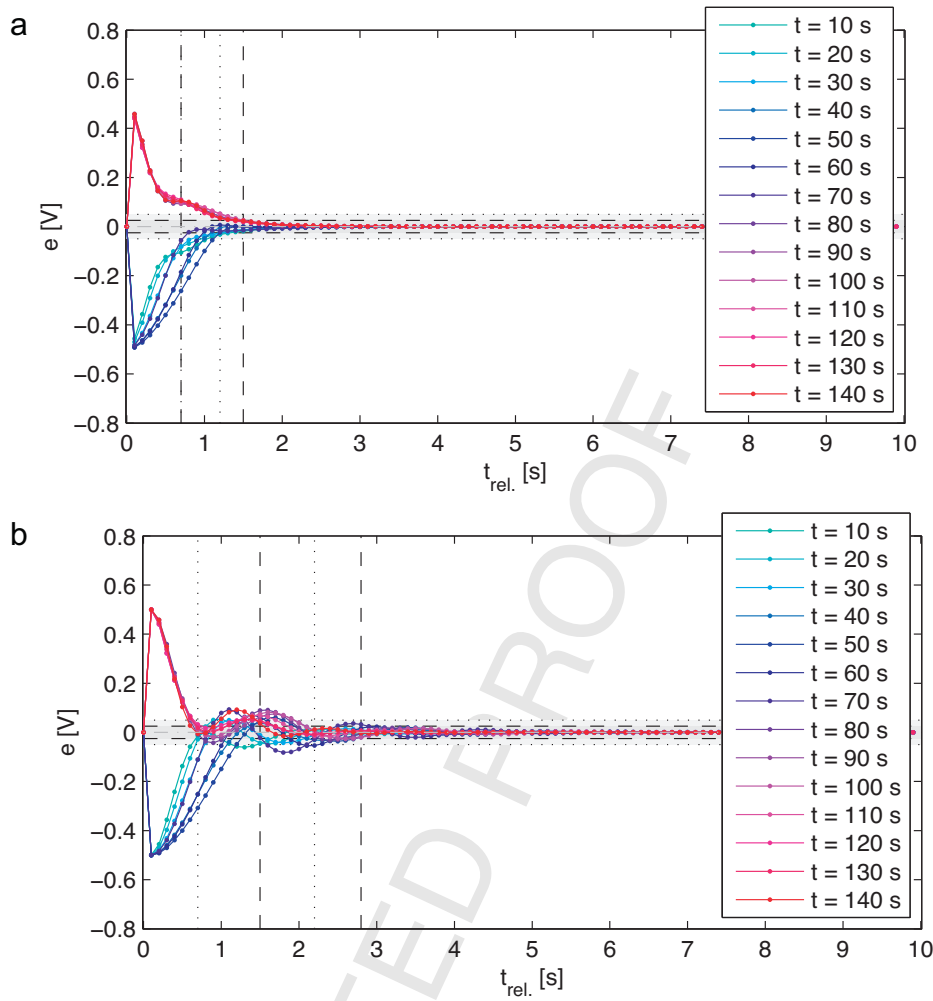


Fig. 9. Simulation. Error signals around all the reference steps plotted in the same time frame for (a) FPFC and (b) 2 DOF FMPC.

, and in Table 3 the performance of the disturbance rejection is evaluated in terms of several criteria.

6.2. Real system

We tested the tracking of the same reference signal as in the simulation environment also on the real system. We used only the presented control algorithm. The results from the real system are shown in figures from Fig. 11 to Fig. 15. Fig. 11 shows the distribution of the clusters over the output space, Fig. 12 shows Bode diagrams of the identified fuzzy model, in Fig. 13 and Fig. 14 the results of the reference tracking are shown, and in Fig. 15 the switching between the clusters during the reference tracking is visualized. The performance of the control algorithm on the real system in terms of different criteria is shown in Table 4.

7. Discussion

The fuzzy model of the system, required in the control algorithm, was obtained with evolving fuzzy modelling. The comparison of Bode diagrams of the identified fuzzy model and linearised analytical model in Fig. 5 confirms that the presented identification method is capable of modelling the system behaviour for the whole range of operating points.

The proposed control law was designed based on the obtained fuzzy model of the system. The reference tracking results in Fig. 9(b) reveal that the closed-loop step response of the proposed approach is invariant to the operating point (inclination of the helio-crane), which was one of the control goals. This means, that the controller is capable of determining appropriate control signals even though the system dynamic behaviour is non-linear. The presented control law takes non-linearity of the system into account implicitly (in

Table 3 Simulation. Comparison of input disturbance rejection to different criteria.

Criterion	Units	2 DOF FMPC	FPFC
SAE	V s	0.1858	0.3541
SSE	V ² s	0.0112	0.0293
SAdU	V	9.48	12.56
SSdU	V ²	5.38	5.12
max t _{s,0.005V}	s	1.60	4.00
max t _{s,0.01V}	s	1.30	3.20
max OS	V	0.1034	0.1497

Table 4 Real system. Performance of reference-tracking to different criteria.

Criterion	Units	2 DOF FMPC
SAE	V s	6.2333
SSE	V ² s	1.2018
SAdU	V	315
SSdU	V ²	403
max t _{s,0.05V}	s	5.50
max t _{s,0.1V}	s	1.60
max OS	V	0.0654

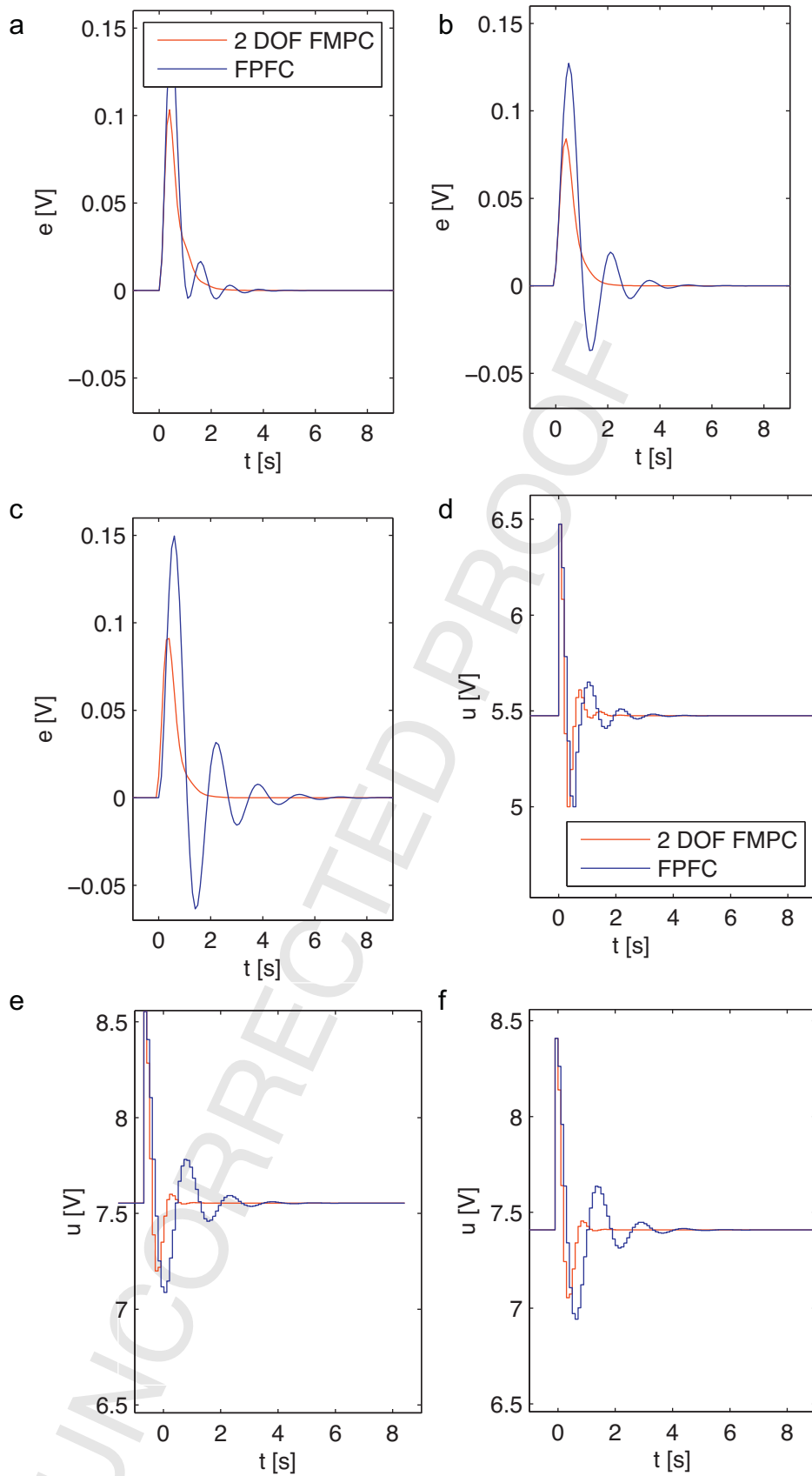


Fig. 10. Simulation. Comparison of input disturbance rejection at three operating points: (a)–(c) outputs and (d)–(f) inputs around the reference values of 5V, 3.5V and 2V from left to right, respectively.

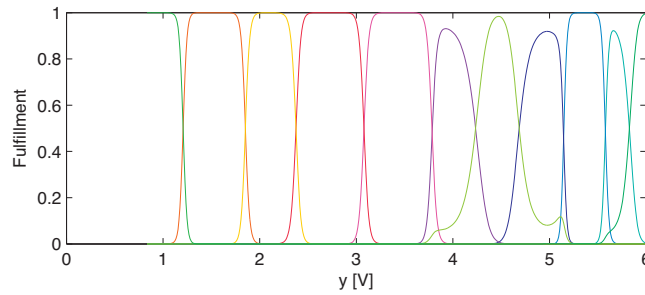


Fig. 11. Real experiment. Distribution of the clusters over the system output.

827 form of a fuzzy model) and adapts the control action accordingly.
 828 The comparison of the presented control law with the fuzzy predic-
 829 tive functional control (Fig. 8) reveals, that the proposed control
 830 law gives a better performance in terms of almost all considered
 831 criteria (Table 2). The output in the case of the FPMC is a little oscil-
 832 latory and has little longer settling time (Fig. 9(a)) than in the case
 833 of 2 DOF fuzzy model predictive control, where the output error
 834 is also monotonically decreasing (Fig. 9(b)). The proposed control
 835 algorithm also give satisfactory results when dealing with input
 836 input disturbance rejection (Fig. 10 and Table 3). This comparison
 837 confirms the applicability of the proposed approach.

838 Since the presented control algorithm is self-tuned based
 839 on the evolving fuzzy model, the mathematical modelling of
 840 the system is not necessary. This simplified the implemen-
 841 tation of the control algorithm not only in the simulation
 842 environment but also on the real system. The main control
 843 goals were also achieved on the real system (Fig. 11 to
 844 Fig. 15). However, the reference tracking (Fig. 14) is not so
 845 precise as in simulation environment. The reason for that is
 846 mainly due to some stiction in the bearing at the pivot point.
 847 Nevertheless, the obtained results on the real system are satisfac-
 848 tory.

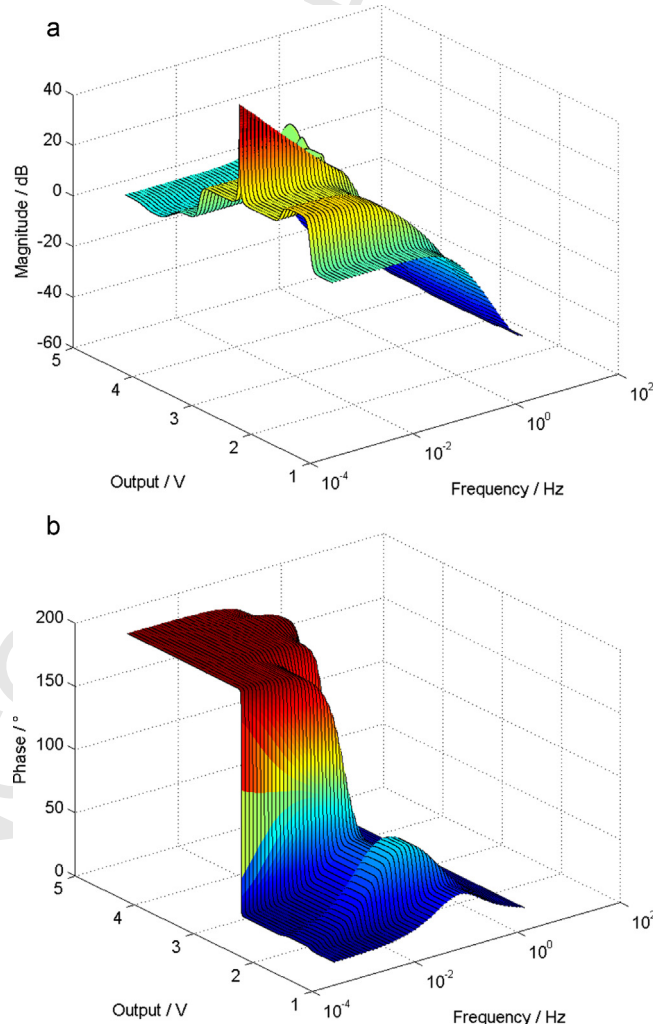


Fig. 12. Real experiment. Bode diagrams of the identified fuzzy model for different operating points.

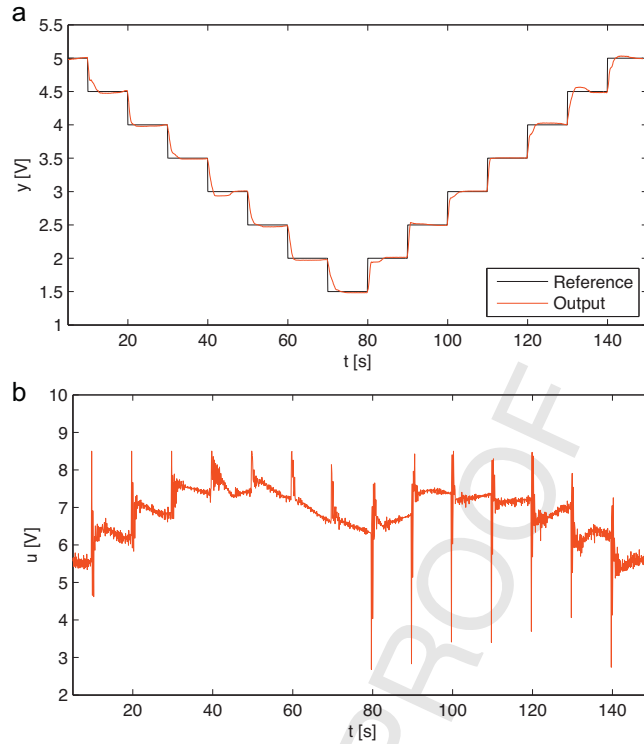


Fig. 13. Real experiment: (a) reference signal tracking and (b) control action.

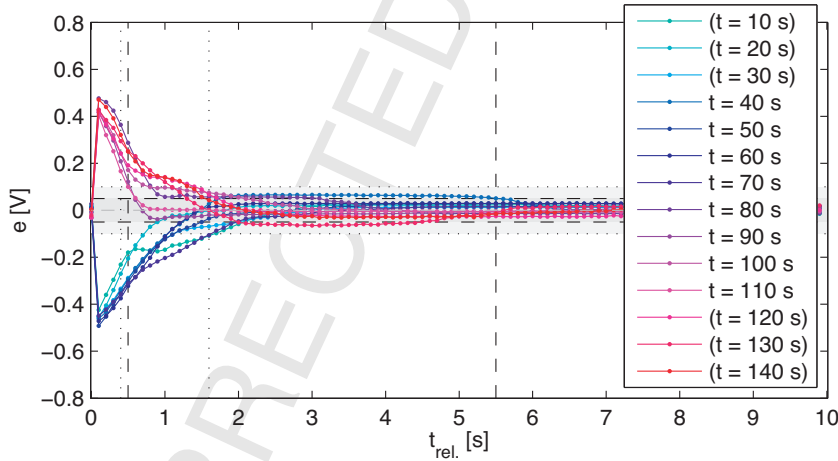


Fig. 14. Real experiment. Error signals around all the reference steps plotted in the same time frame.

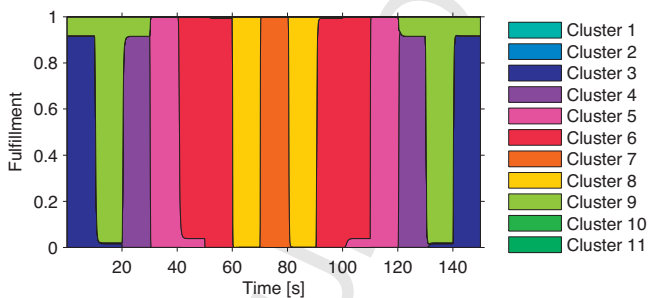


Fig. 15. Real experiment. Time progress of the degrees of fulfillment for all the clusters.

8. Conclusion

In this paper a self-tuning of 2 DOF control based on evolving fuzzy model was presented. We showed the way fuzzy model can be used in the design of the feed-forward and feedback loop of the 2 DOF control algorithm. To achieve self-tuning of the control algorithm, we presented the identification algorithm of evolving fuzzy model which employs mechanisms for adding, merging, splitting and removing of clusters. The results obtained in the simulation environment and real experiments confirm the applicability of the approach. With the presented 2 DOF structure of the control algorithm we were able to achieve good reference tracking that reacts fast to changes in the reference signal and it is able to eliminate the reference-tracking error, even in the presence of disturbances. We showed that the presented control algorithm can

849
850
851
852
853
854
855
856
857
858
859
860
861
862

achieve set-point invariant reference tracking on a non-linear SISO system.

The experimental results have proved the applicability of the presented approach and give solid ground for future enhancements of the presented control algorithm. The control algorithm was developed for a class of SISO non-linear systems, but in the future the approach might be extended to some other classes of systems, e.g. to at least some classes of MIMO systems.

Acknowledgements

The authors would like to thank Assoc. 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] L.A. Zadeh, Fuzzy sets, *Information and Control (IANDC)* 8 (3) (1965) 338–353, [http://dx.doi.org/10.1016/S0019-9958\(65\)90241-X](http://dx.doi.org/10.1016/S0019-9958(65)90241-X).
- [2] T. Procyk, E. Mamdani, A linguistic self-organizing process controller, *Automatica* 15 (1) (1979) 15–30, [http://dx.doi.org/10.1016/0005-1098\(79\)90084-0](http://dx.doi.org/10.1016/0005-1098(79)90084-0).
- [3] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man, and Cybernetics SMC-15* (1) (1985) 116–132, <http://dx.doi.org/10.1109/TSMC.1985.6313399>.
- [4] K.M. Passino, S. Yurkovich, *Fuzzy Control*, Addison-Wesley Longman, 1998, ISBN 0-201-18074-X.
- [5] S.L. Chiu, Fuzzy model identification based on cluster estimation, *Journal of Intelligent and Fuzzy Systems* 2 (3) (1994) 267–278.
- [6] P.P. Angelov, D.P. Filev, An approach to online identification of Takagi-Sugeno fuzzy models, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 34 (1) (2004) 484–498, <http://dx.doi.org/10.1109/TSMCB.2003.817053>.
- [7] M.A. Memon, P. Angelov, H. Ahmed, An approach to real-time color-based object tracking, in: *2006 International Symposium on Evolving Fuzzy Systems, IEEE, Ambleside, 2006*, pp. 86–91, <http://dx.doi.org/10.1109/IEFS.2006.251169>, ISBN 0-7803-9718-5.
- [8] P. Angelov, X. Zhou, Evolving fuzzy systems from data streams in real-time, in: *2006 International Symposium on Evolving Fuzzy Systems, IEEE, Ambleside, 2006*, pp. 29–35, <http://dx.doi.org/10.1109/IEFS.2006.251157>, ISBN 0-7803-9718-5.
- [9] P. Angelov, D. Filev, *Simpl.eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models*, in: *14th IEEE International Conference on Fuzzy Systems (FUZZ'05), IEEE, Reno, Nevada, USA, 2005*, pp. 1068–1073, <http://dx.doi.org/10.1109/FUZZY.2005.1452543>, ISBN 0-7803-9159-4.
- [10] P. Angelov, Evolving Takagi-Sugeno fuzzy systems from streaming data (eTS+), in: *IEEE Press Series on Computational Intelligence, John Wiley & Sons, New Jersey, 2010*, pp. 21–50, <http://dx.doi.org/10.1002/9780470569962.ch2>.
- [11] E. Lughofer, E.P. Klement, FLEXFIS: a variant for incremental learning of Takagi-Sugeno fuzzy systems, in: *14th IEEE International Conference on Fuzzy Systems (FUZZ'05), IEEE, Reno, Nevada, USA, 2005*, pp. 915–920, <http://dx.doi.org/10.1109/FUZZY.2005.1452516>, ISBN 0-7803-9159-4.
- [12] E. Lughofer, J.L. Bouchot, A. Shaker, On-line elimination of local redundancies in evolving fuzzy systems, *Evolving Systems* 2 (3) (2011) 165–187, <http://dx.doi.org/10.1007/s12530-011-9032-3>.
- [13] A. Kalhor, B.N. Araabi, C. Lucas, Evolving Takagi-Sugeno fuzzy model based on switching to neighboring models, *Applied Soft Computing* 13 (2) (2013) 939–946, <http://dx.doi.org/10.1016/j.asoc.2012.09.015>.
- [14] N. Kasabov, Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 31 (6) (2001) 902–918, <http://dx.doi.org/10.1109/3477.969494>.
- [15] N. Kasabov, Evolving fuzzy neural networks – algorithms, applications and biological motivation, in: *Methodologies for the Conception, Design and Application of Soft Computing, Japan, 1998*, pp. 271–274.
- [16] N.K. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE Transactions on Fuzzy Systems* 10 (2) (2002) 144–154, <http://dx.doi.org/10.1109/91.995117>.
- [17] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man, and Cybernetics* 23 (3) (1993) 665–685, <http://dx.doi.org/10.1109/21.256541>.
- [18] M. Azeem, M. Hanmandlu, N. Ahmad, Structure identification of generalized adaptive neuro-fuzzy inference systems, *IEEE Transactions on Fuzzy Systems* 11 (5) (2003) 666–681, <http://dx.doi.org/10.1109/TFUZZ.2003.817857>.
- [19] G. Leng, G. Prasad, T.M. McGinnity, An on-line algorithm for creating self-organizing fuzzy neural networks, *Neural Networks* 17 (10) (2004) 1477–1493, <http://dx.doi.org/10.1016/j.neunet.2004.07.009>.
- [20] H.J. Rong, N. Sundararajan, G.B. Huang, P. Saratchandran, Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction, *Fuzzy Sets and Systems* 157 (9) (2006) 1260–1275, <http://dx.doi.org/10.1016/j.fss.2005.12.011>.
- [21] F.J. Lin, C.H. Lin, P.H. Shen, Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, *IEEE Transactions on Fuzzy Systems* 9 (5) (2001) 751–759, <http://dx.doi.org/10.1109/91.963761>.
- [22] C.T. Lin, A neural fuzzy control system with structure and parameter learning, *Fuzzy Sets and Systems* 70 (2–3) (1995) 183–212, [http://dx.doi.org/10.1016/0165-0114\(94\)00216-T](http://dx.doi.org/10.1016/0165-0114(94)00216-T).
- [23] S. Wu, M.J. Er, Dynamic fuzzy neural networks – a novel approach to function approximation, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 30 (2) (2000) 358–364, <http://dx.doi.org/10.1109/3477.836384>.
- [24] S. Wu, M.J. Er, Y. Gao, A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, *IEEE Transactions on Fuzzy Systems* 9 (4) (2001) 578–594, <http://dx.doi.org/10.1109/91.940970>.
- [25] C.F. Juang, C.T. Lin, An online self-constructing neural fuzzy inference network and its applications, *IEEE Transactions on Fuzzy Systems* 6 (1) (1998) 12–32, <http://dx.doi.org/10.1109/91.660805>.
- [26] S.G. Tzafestas, K.C. Zikidis, NeuroFAST: on-line neuro-fuzzy ART-based structure and parameter learning TSK model, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 31 (5) (2001) 797–802, <http://dx.doi.org/10.1109/3477.956041>.
- [27] J. Platt, A resource-allocating network for function interpolation, *Neural Computation* 3 (2) (1991) 213–225, <http://dx.doi.org/10.1162/neco.1991.3.2.213>.
- [28] D. Deng, N. Kasabov, Evolving self-organizing maps for on-line learning, data analysis and modelling, in: S.I. Amari, C.L. Giles, M. Gori, V. Piuri (Eds.), *IJCNN 2000 Neural Networks Neural Computing: New Challenges Perspectives New Millennium, vol. 6, IEEE, New York, 2000*, pp. 3–8.
- [29] B. Fritzke, A growing neural gas network learns topologies, *Advances in Neural Information Processing Systems* 7 (1995) 625–632.
- [30] B.H. Soleimani, C. Lucas, B.N. Araabi, Recursive Gath-Geva clustering as a basis for evolving neuro-fuzzy modeling, *Evolving Systems* 1 (1) (2010) 59–71, <http://dx.doi.org/10.1007/s12530-010-9006-x>.
- [31] G.B. Huang, P. Saratchandran, N. Sundararajan, A recursive growing and pruning RBF (GAP-RBF) algorithm for function approximations, in: *4th International Conference on Control and Automation (ICCA'03), IEEE, Montreal, Canada, 2003*, pp. 491–495, <http://dx.doi.org/10.1109/ICCA.2003.1595070>, ISBN 0-7803-7777-X.
- [32] D. Dovžan, I. Škrjanc, Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes, *ISA Transactions* 50 (2) (2011) 159–169, <http://dx.doi.org/10.1016/j.isatra.2011.01.004>.
- [33] D. Dovžan, I. Škrjanc, Recursive clustering based on a Gustafson-Kessel algorithm, *Evolving Systems* 2 (1) (2011) 15–24, <http://dx.doi.org/10.1007/s12530-010-9025-7>.
- [34] D. Dovžan, I. Škrjanc, Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor, *Control Engineering Practice* 18 (8) (2010) 979–989, <http://dx.doi.org/10.1016/j.conengprac.2010.04.004>.
- [35] C. Brosilow, B. Joseph, Two-degree of freedom internal model control, in: *Techniques of Model-Based Control*, Prentice Hall, 2002, pp. 65–84, ISBN 9780130280787.
- [36] M. Morari, J.H. Lee, Model predictive control: past, present and future, *Computers and Chemical Engineering* 23 (4–5) (1999) 667–682, [http://dx.doi.org/10.1016/S0098-1354\(98\)00301-9](http://dx.doi.org/10.1016/S0098-1354(98)00301-9).
- [37] J.M. Maciejowski, *Predictive Control: With Constraints*, Pearson Education, 2002, ISBN 9780201398236.
- [38] E.F. Camacho, C. Bordons, *Model Predictive Control, 2nd ed.*, Springer, 2004, ISBN 9781852336943.
- [39] J.E. Normey-Rico, J. Gómez-Ortega, E.F. Camacho, A Smith-predictor-based generalised predictive controller for mobile robot path-tracking, *Control Engineering Practice* 7 (6) (1999) 729–740, [http://dx.doi.org/10.1016/S0967-0661\(99\)00025-8](http://dx.doi.org/10.1016/S0967-0661(99)00025-8).
- [40] R. Rouhani, R.K. Mehra, Model algorithmic control (MAC): basic theoretical properties, *Automatica* 18 (4) (1982) 401–414, [http://dx.doi.org/10.1016/0005-1098\(82\)90069-3](http://dx.doi.org/10.1016/0005-1098(82)90069-3).
- [41] J. Richalet, D. O'Donovan, *Predictive Functional Control: Principles and Industrial Applications*, Springer, 2009, ISBN 9781848824935.
- [42] J. Richalet, Industrial applications of model based predictive control, *Automatica* 29 (5) (1993) 1251–1274, [http://dx.doi.org/10.1016/0005-1098\(93\)90049-Y](http://dx.doi.org/10.1016/0005-1098(93)90049-Y).
- [43] I. Škrjanc, D. Matko, Predictive functional control based on fuzzy model for heat-exchanger pilot plant, *IEEE Transactions on Fuzzy Systems* 8 (6) (2000) 705–712, <http://dx.doi.org/10.1109/91.890329>.
- [44] M. Lepetič, I. Škrjanc, H.G. Chiachiarini, D. Matko, Predictive functional control based on fuzzy model: comparison with linear predictive functional control and PID control, *Journal of Intelligent and Robotic Systems* 36 (4) (2003) 467–480, <http://dx.doi.org/10.1023/A:1023685505300>.
- [45] S. Blažič, I. Škrjanc, Design and stability analysis of fuzzy model-based predictive control – a case study, *Journal of Intelligent and Robotic Systems* 49 (3) (2007) 279–292, <http://dx.doi.org/10.1007/s10846-007-9147-8>.
- [46] G. Karer, G. Mušič, I. Škrjanc, B. Zupančič, Feedforward control of a class of hybrid systems using an inverse model, *Mathematics and Computers in Simulation* 82 (3) (2011) 414–427, <http://dx.doi.org/10.1016/j.matcom.2010.10.015>.
- [47] F. Malchow, O. Sawodny, Model based feedforward control of an industrial glass feeder, *Control Engineering Practice* 20 (1) (2012) 62–68, <http://dx.doi.org/10.1016/j.conengprac.2011.09.004>.

- [48] K. Graichen, V. Hagenmeyer, M. Zeitz, A new approach to inversion-based feedforward control design for nonlinear systems, *Automatica* 41 (12) (2005) 2033–2041, <http://dx.doi.org/10.1016/j.automatica.2005.06.008>.
- [49] R. Garduno-Ramirez, K.Y. Lee, Wide range operation of a power unit via feedforward fuzzy control, *IEEE Transactions on Energy Conversion* 15 (4) (2000) 421–426.
- [50] R.E. Precup, S. Preitl, E.M. Petriu, J.K. Tar, M.L. Tomescu, C. Pozna, Generic two-degree-of-freedom linear and fuzzy controllers for integral processes, *Journal of the Franklin Institute* 346 (10) (2009) 980–1003, <http://dx.doi.org/10.1016/j.franklin.2009.03.006>.
- [51] C.S. Chiu, Mixed feedforward/feedback based adaptive fuzzy control for a class of MIMO nonlinear systems, *IEEE Transactions on Fuzzy Systems* 14 (6) (2006) 716–727, <http://dx.doi.org/10.1109/TFUZZ.2006.877357>.
- [52] G. Oriolo, A. De Luca, M. Vendittelli, WMR control via dynamic feedback linearization: design implementation and experimental validation, *IEEE Transactions on Control Systems Technology* 10 (6) (2002) 835–852, <http://dx.doi.org/10.1109/TCST.2002.804116>.
- [53] G. Klančar, I. Škrjanc, Tracking-error model-based predictive control for mobile robots in real time, *Robotics and Autonomous Systems* 55 (6) (2007) 460–469, <http://dx.doi.org/10.1016/j.robot.2007.01.002>.
- [54] I. Škrjanc, G. Klančar, Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves, *Robotics and Autonomous Systems* 58 (1) (2010) 1–9, <http://dx.doi.org/10.1016/j.robot.2009.09.003>.
- [55] K. Graichen, M. Treuer, M. Zeitz, Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation, *Automatica* 43 (1) (2007) 63–71, <http://dx.doi.org/10.1016/j.automatica.2006.07.023>.
- [56] R.K. Barai, K. Nonami, Optimal two-degree-of-freedom fuzzy control for locomotion control of a hydraulically actuated hexapod robot, *Information Sciences* 177 (8) (2007) 1892–1915, <http://dx.doi.org/10.1016/j.ins.2006.10.003>.
- [57] B.J. Kang, L.S. Hung, S.K. Kuo, S.C. Lin, C.M. Liaw, H-infinity 2DOF control for the motion of a magnetic suspension positioning stage driven by inverter-fed linear motor, *Mechatronics* 13 (7) (2003) 677–696, [http://dx.doi.org/10.1016/S0957-4158\(02\)00037-5](http://dx.doi.org/10.1016/S0957-4158(02)00037-5).
- [58] N. Zimmert, A. Pertsch, O. Sawodny, 2-DOF control of a fire-rescue turntable ladder, *IEEE Transactions on Control Systems Technology* 20 (2) (2012) 438–452, <http://dx.doi.org/10.1109/TCST.2011.2116021>.
- [59] A.G. Abilov, Z. Zeybek, O. Tuzunalp, Z. Telatar, Fuzzy temperature control of industrial refineries furnaces through combined feedforward/feedback multivariable cascade systems, *Chemical Engineering and Processing* 41 (1) (2002) 87–98, [http://dx.doi.org/10.1016/S0255-2701\(01\)00119-2](http://dx.doi.org/10.1016/S0255-2701(01)00119-2).
- [60] J. de Jesús Rubio, Stability analysis of an on-line evolving neuro-fuzzy network, in: *IEEE Press Series on Computational Intelligence*, John Wiley & Sons, Hoboken, New Jersey, 2010, pp. 173–199, ISBN 9780470569955.
- [61] S. Blažič, I. Škrjanc, S. Gerkišič, G. Dolanc, S. Strmčnik, M.B. Hadjiski, et al., Online fuzzy identification for an intelligent controller based on a simple platform, *Engineering Applications of Artificial Intelligence* 22 (4–5) (2009) 628–638, <http://dx.doi.org/10.1016/j.engappai.2009.03.002>.
- [62] K.J. Åström, B. Wittenmark, *Adaptive Control*, 2nd ed., Addison-Wesley, USA, 1995, ISBN 9780201558661.
- [63] L. Hartert, M.S. Mouchaweh, P. Billaudel, A semi-supervised dynamic version of fuzzy k-nearest neighbours to monitor evolving systems, *Evolving Systems* 1 (1) (2010) 3–15, <http://dx.doi.org/10.1007/s12530-010-9001-2>.
- [64] E. Lughofer, Evolving fuzzy systems – methodologies, advanced concepts and applications, in: vol. 266 of *Studies in Fuzziness and Soft Computing Series*, 1st ed., Springer, Berlin, Heidelberg, 2011, p. 456, ISBN 9783642180866.
- [65] B. Hartmann, O. Banfer, O. Nelles, A. Sodja, L. Teslić, I. Škrjanc, Supervised hierarchical clustering in fuzzy model identification, *IEEE Transactions on Fuzzy Systems* 19 (6) (2011) 1163–1176, <http://dx.doi.org/10.1109/TFUZZ.2011.2164256>.
- [66] H.O. Wang, K. Tanaka, M.F. Griffin, An approach to fuzzy control of nonlinear systems: stability and design issues, *IEEE Transactions on Fuzzy Systems* 4 (1) (1996) 14–23.
- [67] I. Škrjanc, S. Blažič, Predictive functional control based on fuzzy model: design and stability study, *Journal of Intelligent and Robotic Systems* 43 (2–4) (2005) 283–299, <http://dx.doi.org/10.1007/s10846-005-5138-9>.
- [68] I. Škrjanc, D. Matko, Fuzzy predictive functional control in the state space domain, *Journal of Intelligent and Robotic Systems* 31 (1–3) (2001) 283–297.
- [69] M. Lepetič, I. Škrjanc, H.G. Chiacchiarini, D. Matko, Predictive functional control based on fuzzy model: magnetic suspension system case study, *Engineering Applications of Artificial Intelligence* 16 (5–6) (2003) 425–430, [http://dx.doi.org/10.1016/S0952-1976\(03\)00071-X](http://dx.doi.org/10.1016/S0952-1976(03)00071-X).
- [70] M. Blazinšek, M. Arh, I. Škrjanc, Comparison between the auto-tuning of PID, PFC and modified-PFC controllers, in: *7th Vienna Conference on Mathematical Modelling (MATHMOD)*, Vienna, 2012, pp. 1–6.