We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Enhancing Fuzzy Controllers Using Generalized Orthogonality Principle

Nora Boumella, Juan Carlos Figueroa and Sohail Iqbal

Additional information is available at the end of the chapter

1. Introduction

In the early days, the parameters of the fuzzy logic systems were fixed arbitrary, thus leading to a large number of possibilities for FLSs. In 1992, it has been shown that linguistic rules can be converted into Fuzzy Basis Functions (FBFs), and numerical rules and its associated FBFs must be extracted from numerical data training. Since that time, a multitude of design methods to construct a FLS are proposed. Some of these methods are intensive on data analysis, some are aimed at computational simplicity, some are recursive and others are offline, but all based on the the same idea: *tune the parameters of a FLS using the numerical training data*. Methods for designing FLSs can be classified into two major categories: A first category where shapes and parameters of the antecedent MFs are fixed ahead of time and training data are used for tuning the consequent parameters, and a second category that consists of fixing the shapes of the antecedent and consequent MFs using training data to tune the antecedent and the parameters of the consequent.

Two kinds of FLSs, the Mamdani and the Takagi-Sugeno-Kang (TSK) FLSs are widely used and they are currently adopted by the scientific community. They solely differ in the way the consequent structure is defined. The fact that a TSK FLS does not require a time-consuming defuzzification process makes it far more attractive for most of applications.

In this chapter, we consider the first category to design a TSK FLS basing on alinear method. Our design approach requires a set of input-output numerical data training pairs. Given linguistic rules of the FLS, we expand this FLS as a series of FBFs that are functions of the FLS inputs. We use the input training data to compute these FBFs. Therefore, the system becomes linear in the FLS consequent parameters, and we consider each set of FBFs as a basis vector which is easy to be optimized. Then follows the consequent parameters optimization via a minimizing process of the error vector - *the output training data minus the FBFs vectors weighted by the consequent parameters* - norm. This minimzation can be obtained by applying the *Generalized Orthogonality Principle* (GOP). Optimization process is carefully analyzed in this chapter and its applications in two major areas of concern are demonstrated including

robotics and dynamic systems. Firstly, we shall show the improved results with analysis upon the application of GOP in the Fuzzy Logic Controller (FLC) for an inverted pendulum. Secondly, we show how a FLS based on this principle enhances the performance of forecaster for the chaotic time series.

2. Fuzzy Logic Systems (FLS) basic concepts

2.1. Fuzzy sets

A *Fuzzy Set* (FS), $F \in X$ is a set of ordered pairs of a generic element x and its degree, namely *Membership Function* (MF), $\mu_F(x)$. Any FS can be represented as follows:

$$F = \{(x, \mu_F(x)) \, | \forall x \in X\} \tag{1}$$

where the membership degree of x, $\mu_F(x)$, is constrained to be betwwen 0 and 1 for all $x \in X$.

2.2. Mamdani FLS

An FLS is an intuitive and numerical system that maps crisp (deterministic) inputs to a crisp output. It is composed of four elements which are depicted in Figure 1. To completly describe this FLS, we need a mathematical formula that maps the crisp input **x** into a crisp output $y = f(\mathbf{x})$, we can obtain this formula by following the signal **x** through the fuzzifier to the inference block and into the defuzzifier. We explain, in this section, the working principle of this formula.

2.2.1. Rules

The FLS is associated with a set of *IF-THEN* rules with meaningful linguistic interpretations. The lth rule of a FLS having p inputs $x_1, ..., x_p$ and one output $y \in Y$, *Multiple Input Single Output* (MISO), is expressed as:

$$R^l : If \;\; x_1 \;\; is \;\; F_1^l \;\; and, ... , and \;\; x_p \;\; is \;\; F_p^l \;\; THEN \;\; y \;\; is \;\; G^l \tag{2}$$

where F_i^l $(i = 1, 2, ..., p)$ are fuzzy antecedent sets wich are represented by their MFs $\mu_{F_i^l}$, and G^l is a consequent set where $l = 1, ..., M$ (M is the number of rules in the FLS).
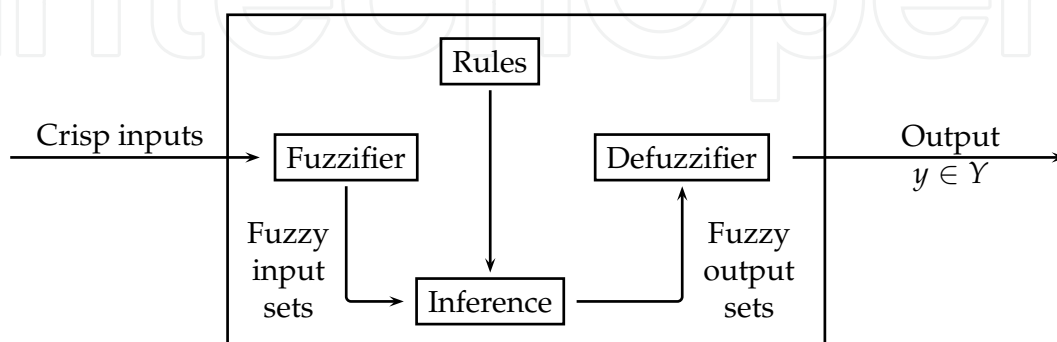

Figure 1. Block diagram of a fuzzy logic system

2.2.2. Fuzzifier

A fuzzifier maps any crisp input $\mathbf{x} = (x_1, ..., x_p)^T \in X_1 \times \cdots \times X_p \equiv \mathbf{X}$ into a fuzzy set $F_\mathbf{x}$ in $\mathbf{X}$ [8].

2.2.3. Inference

A fuzzy inference engine combines rules from the fuzzy rule base and gives a mapping from input fuzzy sets in $\mathbf{X}$ to output sets in Y. Each rule is interpreted as a fuzzy implication, i.e., a fuzzy set in $\mathbf{X} \times Y$, and can be expressed as:

$$R^l : \ F_1^l \times ... \times F_p^l \longrightarrow G^l = A^l \longrightarrow G^l \quad l = 1, ..., M \tag{3}$$

Usually in Mamdani FLS, the implication is replaced by a *t-norm*, i.e. (product or min). Multiple antecedents are connected by a t-norm, so a rule can be expressed by its MF as follows:

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{F_1^l \times F_2^l \times ... \times F_p^l}(x_1, x_2, .., x_p) \star \mu_{G^l}(y)$$

$$= \left[T_{i=1}^p \mu_{F_i^l}(x_i) \right] \star \mu_{G^l}(y) \tag{4}$$

where T and $\star$ are $t - norm$ operators (*product* or *min*). The p-dimensional input to R^l is given by the fuzzy set $A_\mathbf{x}$ whose MF is expressed as [8]

$$\mu_{A_\mathbf{x}}(\mathbf{x}) = \mu_{X_1}(x_1) \star ... \star \mu_{X_p}(x_p) = T_{i=1}^p \mu_{X_i^l}(x_i) \tag{5}$$

Each rule detemines a fuzzy set B^l in Y which is derived from the sup $-\star$ composition. Then, the MF of this output set is expressed as [8]

$$\mu_{B^l}(y) = \mu_{A_X \circ R^l}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left[\mu_{A_\mathbf{x}}(\mathbf{x}) \star \mu_{R^l}(\mathbf{x}, y) \right] \tag{6}$$

$$\mu_{B^l}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left[T_{i=1}^p \mu_{X_i}(x_i) \star \left(\left[T_{i=1}^p \mu_{F_i^l}(x_i) \right] \star \mu_{G^l}(y) \right) \right] \tag{7}$$

Finally, the *l*th rule is expressed as follows

$$\mu_{B^l}(y) = \mu_{G^l}(y) \star \left[T_{i=1}^p \mu_{F_i^l}(x_i) \right] \quad y \in Y \tag{8}$$

2.2.4. Defuzzifier

As we pointed out before, the main idea of a Mamdani FLS is to use crisp inputs to make fuzzy inference and finally find a crisp output which represents the behavior of the FLS. The process of finding a crisp output after fuzzification and inference is called *Deffuzification*. This final step consist on find an operation point given the results of the inference process of the FLS, which results on a fuzzy output set, so we need to use a mathematical method which returns a crisp measure of the behavior of the FLS.

There are many types of defuzzifiers, but we consider in this paper the *Height Defuzzifier* which replaces each rule output fuzzy set by a singleton at the point having maximum membership

in that output set, $\overline{y}^l$, then it calculates the centroid of the resultantF set of these singletons. The crisp output of this defuzzifier is expressed as:

$$y(\mathbf{x}) = f(\mathbf{x}) = \frac{\sum_{l=1}^{M} \overline{y}^l \mu_{B^l}(\overline{y}^l)}{\sum_{l=1}^{M} \mu_{B^l}(\overline{y}^l)} \tag{9}$$

where $\overline{y}^l$ is the point having maximum membership in the output set [8].

2.3. Takagi-Sugeno-Kang (TSK) FLS

A TSK FLS is a special FLS which is also characterized by IF-THEN rules, but its consequent is a polynomial. Its output is a crisp value obtained from computing the polynomial output, so it does not need a defuzzification process. The l_{th} rule of a first order type-1 TSK FLS having p inputs $x_1 \in X_1, ..., x_p \in X_p$ and one output $y \in Y$ is expressed as:

$$R^l : \text{IF } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \text{ and...and } x_p \text{ is } F_p^l$$

$$\text{THEN} \quad y^l(\mathbf{x}) = c_0^l + c_1^l x_1 + ... + c_p^l x_p \tag{10}$$

where $l = 1, ..., M$, $c_j^l (j = 0, .., p)$ are the consequent parameters, $y^l(\mathbf{x})$ is the output of the lth rule, and F_k^l $(k = 1, ..., p)$ are type-1 antecedent fuzzy sets.

The output of a TSK FLS is obtained by combining the outputs from the M rules in the following form:

$$y_{TSK}(\mathbf{x}) = \frac{\sum_{l=1}^{M} f^l(\mathbf{x}) \left(c_0^l + c_1^l x_1 + ... + c_p^l x_p \right)}{\sum_{l=1}^{M} f^l(\mathbf{x})} \tag{11}$$

where $f^l(\mathbf{x})$ $(l = 1, ..., M)$ are the rule firing levels and they are defined as:

$$f^l(\mathbf{x}) = T_{k=1}^{p} \mu_{F_k^l}(x_k) \tag{12}$$

where T is a $t-norm$ operation, i.e. minimum or product operation (Mendel [8]), and $\mathbf{x}$ is the vector of inputs applied to the TSK FLS.

2.4. Fuzzy basis functions

For Mamdani FLSs, assuming that all consequent MFs are normalized, i.e., $\mu_{G^l}\left(\overline{y}^l \right) = 1$, and using singleton defuzzification, max-product composition and product implication, then the output of the height defuzzifier (9) becomes:

$$y(\mathbf{x}) = f(\mathbf{x}) = \frac{\sum_{l=1}^{M} \overline{y}^l T_{i=1}^{p} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} T_{i=1}^{p} \mu_{F_i^l}(x_i)} \tag{13}$$

The FLS in (13) can be expressed as:

$$y(\mathbf{x}) = f(\mathbf{x}) = \sum_{l=1}^{M} \overline{y}^l \phi_l(\mathbf{x}) \tag{14}$$

where $\phi_l(\mathbf{x})$ is called a *Fuzzy Basis Function* (FBF) of the *lth* rule [11], and it is defined as:

$$\phi_l(\mathbf{x}) = \frac{f^l}{\sum_{l=1}^{M} f^l} \qquad l = 1, ..., M \tag{15}$$

where f^l is given in (12).

This linear combination allows us to view an FLS as series expansions of FBFs [11], [1], [4] and [10] which has the capability of providing a mix of both numerical and linguistic information.

2.5. Weighted FBF

The crisp output of the TSK FLS in (11) can be expressed as:

$$y_{TSK}(\mathbf{x}) = \sum_{l=1}^{M} \phi_l(\mathbf{x}) \sum_{k=0}^{p} c_k^l x_k \tag{16}$$

It can also be expressed as:

$$y_{TSK}(\mathbf{x}) = \sum_{l=1}^{M} \sum_{k=0}^{p} \phi_k^l(\mathbf{x}) c_k^l \tag{17}$$

where $\phi_k^l(\mathbf{x})$ is the *kth* Weighted Fuzzy Basis Function (WFBF) of the *lth* rule which is expressed as [2]:

$$\phi_k^l(\mathbf{x}) = x_k \phi_l(\mathbf{x}), \quad l = 1, ..., M; k = 0, ..., p \tag{18}$$

This linear combination allows us to view the FLS as series expansions of WFBFs [2]. The WFBFs have also a capability of providing a combination of both numerical and linguistic information.

3. Orthogonality principle

We explain in this section how we can obtain, graphically, the optimal scalar that minimizes the norm of an error vector [9].Suppose that we have a set of N measurements collected in a N-vector, $\overrightarrow{y}$, gathered for different values collected in another N-vector, $\overrightarrow{\phi}$. The problem is to find :

$$\min_{\theta} \left\| \overrightarrow{y} - \theta \overrightarrow{\phi} \right\| \tag{19}$$

As shown in Figure 2, we can see that the optimal scalar θ that minimizes the norm of the error vector, $\left\| \overrightarrow{e} = \overrightarrow{y} - \theta \overrightarrow{\phi} \right\|$, is obtained when $\overrightarrow{e} \perp \overrightarrow{\phi}$. This can be expressed as follows :

$$\overrightarrow{\phi} \cdot \left(\overrightarrow{y} - \theta \overrightarrow{\phi} \right) = 0 \tag{20}$$

Solving for θ we have:

$$\theta_{opt} = \frac{\overrightarrow{y}^T \overrightarrow{\phi}}{\overrightarrow{\phi}^T \overrightarrow{\phi}} \tag{21}$$
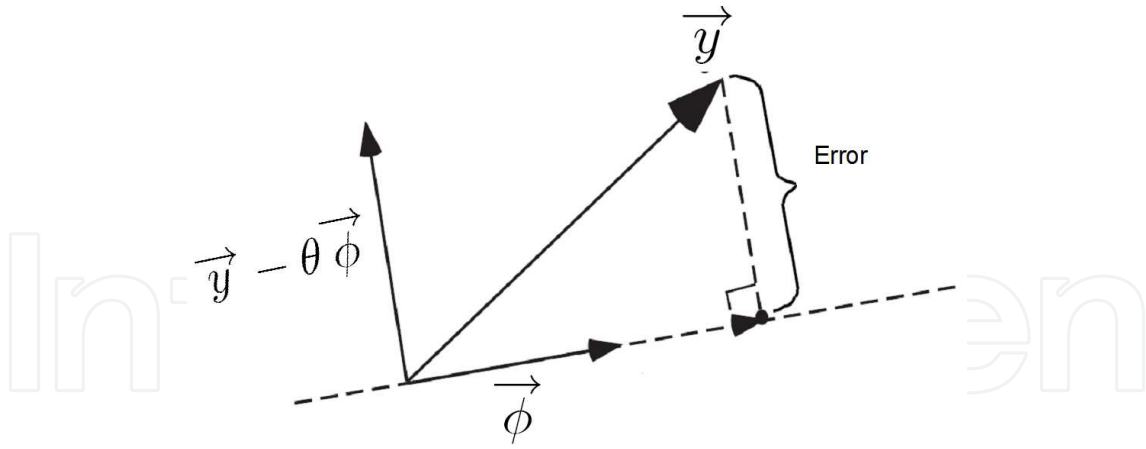
Figure 2. Basic Idea of Orthogonality Principle

4. FLS design based on GOP

GOP is an optimization principle which can be applied to both Mamdani and TSK FLSs. Under the premise of fixed shapes and the parameters of the antecedent MFs over the time, then a training dataset is used to tune the consequent parameters. The consequent parameters are c_k^l ($l = 1, ..., M; k = 0, ..., p$) in (11) for a TSK FLS, and $\overline{y}^l$ ($l = 1, ..., M$) in (9) for a Mamdani FLS.

4.1. Mamdani FLS design

Given a collection of N input-output numerical data training pairs

$$\left(\mathbf{x}^{(1)} : y^{(1)}\right), \left(\mathbf{x}^{(2)} : y^{(2)}\right), ..., \left(\mathbf{x}^{(N)} : y^{(N)}\right)$$

where $\mathbf{x}^{(i)}$ and $y^{(i)}$ are respectively the vector input and scalar output of the FLS given by (13). We have to tune the $\overline{y}^l$ ($l = 1, ..., M$) using these data training. Firstly, we compute the FBFs with training input vectors, then we apply the orthogonality principle on these FBFs and the training output vector.

Equation (14) can be decomposed as follows:

$$
\begin{cases}
y(\mathbf{x}^{(1)}) = f(\mathbf{x}^{(1)}) = \overline{y}^1 \phi_1(\mathbf{x}^{(1)}) + ... + \overline{y}^M \phi_M(\mathbf{x}^{(1)}) \\
y(\mathbf{x}^{(2)}) = f(\mathbf{x}^{(2)}) = \overline{y}^1 \phi_1(\mathbf{x}^{(2)}) + ... + \overline{y}^M \phi_M(\mathbf{x}^{(2)}) \\
\qquad\qquad\qquad\quad \vdots \\
y(\mathbf{x}^{(N)}) = f(\mathbf{x}^{(N)}) = \overline{y}^1 \phi_1(\mathbf{x}^{(N)}) + ... + \overline{y}^M \phi_M(\mathbf{x}^{(N)})
\end{cases}
\tag{22}
$$

So we have

$$y(\mathbf{x}^{(i)}) = f(\mathbf{x}^{(i)}) = \sum_{l=1}^{M} \overline{y}^l \phi_l(\mathbf{x}^{(i)}) \quad i = 1, ..., N \tag{23}$$

Now, if each FBF is considered as a basis function, we can compose the following vector:

$$\overrightarrow{\phi}_j = \begin{pmatrix} \phi_j(\mathbf{x}^{(1)}) \\ \phi_j(\mathbf{x}^{(2)}) \\ \vdots \\ \phi_j(\mathbf{x}^{(N)}) \end{pmatrix}, \quad j = 1, 2, ..., M \qquad (24)$$

where M is the number of rules. We now collect all the N training output data in the same vector $\overrightarrow{y}$:

$$\overrightarrow{y} = \begin{pmatrix} y(\mathbf{x}^{(1)}) \\ y(\mathbf{x}^{(2)}) \\ \vdots \\ y(\mathbf{x}^{(N)}) \end{pmatrix} \qquad (25)$$

and the parameters of the consequent in a vector $\overrightarrow{\theta}$:

$$\overrightarrow{\theta} = \begin{pmatrix} \overline{y}^1 \\ \overline{y}^2 \\ \vdots \\ \overline{y}^M \end{pmatrix} \qquad (26)$$

By considering the N equations, a FLS can be expressed in vector-matrix format as follows:

$$\overrightarrow{y} = \mathbf{\Phi} \, \overrightarrow{\theta} \qquad (27)$$

where the fuzzy basis function matrix $\mathbf{\Phi}$ is given by:

$$\mathbf{\Phi} = [\overrightarrow{\phi}_1, \overrightarrow{\phi}_2, ..., \overrightarrow{\phi}_M] \qquad (28)$$

To find the optimal vector $\overrightarrow{\theta}$ and because of fitting with basis sets, we generalize the presented orthogonality principle to a multi-dimensional basis leading to a GOP. The error vector should be perpendicular to all of the basis fuzzy vectors, as shown in Figure 2.

In a matrix form, we obtain:

$$\mathbf{\Phi}^T \cdot \left(\overrightarrow{y} - \mathbf{\Phi} \, \overrightarrow{\theta} \right) = 0 \qquad (29)$$

Solving for $\overrightarrow{\theta}$, we have:

$$\overrightarrow{\theta}_{opt} = \begin{pmatrix} \overline{y}^1 \\ \overline{y}^2 \\ \vdots \\ \overline{y}^M \end{pmatrix} = \left[\mathbf{\Phi}^T \mathbf{\Phi} \right]^{-1} \mathbf{\Phi}^T \overrightarrow{y}$$

where $\overrightarrow{\theta}_{opt}$ is a vector which contains the parameters of the consequent, i.e., $\overline{y}^l$ in (3).
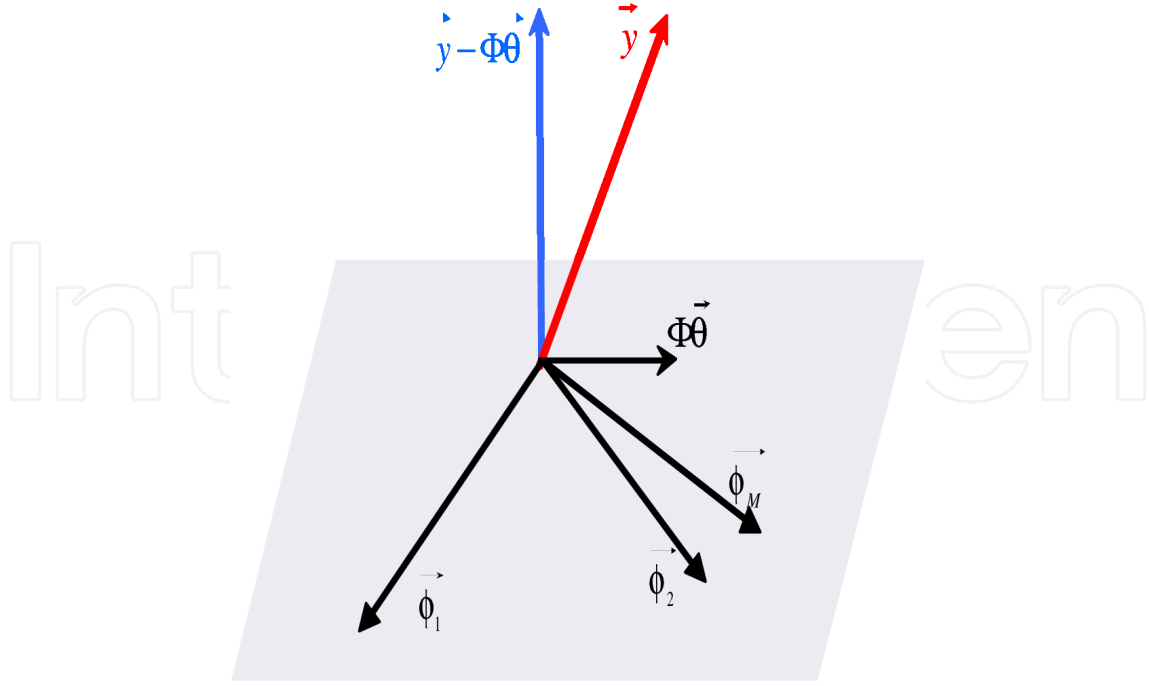
Figure 3. Basic Idea of Generalized Orthogonality Principle. The error vector should be perpendicular to all of the basis fuzzy vectors.

4.2. TSK FLS design

In the same way, the consequent parameters of a TSK FLS are tuned. The design approach is related to the following problem:

Given a collection of N input-output numerical training data pairs:

$$\left(\mathbf{x}^{(1)} : y^{(1)}\right), \left(\mathbf{x}^{(2)} : y^{(2)}\right), \ldots, \left(\mathbf{x}^{(N)} : y^{(N)}\right)$$

where $\mathbf{x}^{(i)}$ is the $(p+1) - dimensional$ input vector ($p+1$ inputs with $x_0 \equiv 1$) and $y^{(i)}$ is the scalar output of the FLS given by (11). We have to tune the c_k^l ($l = 1, ..., M; k = 0, ..., p$) using these data training.

The WFBF vectors are computed using the training input data, then the GOP is applied to the $(p+1)$ combinations of WFBF vectors and the $(p+1)$ of $N-dimensional$ training output vector.

Using the elements of the input-output training pairs, the TSK output given in (17), can be rewritten as follows:

$$y_{TSK}(\mathbf{x}^{(i)}) = \begin{cases} \begin{bmatrix} \phi_0^1(\mathbf{x}^{(i)}) \\ \cdots \\ \phi_p^1(\mathbf{x}^{(i)}) \end{bmatrix}^T \begin{bmatrix} c_0^1 \\ \vdots \\ c_p^1 \end{bmatrix} \\ + \cdots + \\ \begin{bmatrix} \phi_0^M(\mathbf{x}^{(i)}) \\ \cdots \\ \phi_p^M(\mathbf{x}^{(i)}) \end{bmatrix}^T \begin{bmatrix} c_0^M \\ \vdots \\ c_p^M \end{bmatrix} \end{cases} \qquad (30)$$

where $\mathbf{x}^{(i)} = \left[1, x_1^{(i)}, ..., x_p^{(i)} \right]^T$. Collecting the N equations we obtain:

$$\overrightarrow{y_{TSK}} = \begin{cases} \begin{bmatrix} \phi_0^1(\mathbf{x}^{(1)}) & \cdots & \phi_p^1(\mathbf{x}^{(1)}) \\ & \ddots & \\ \phi_0^1(\mathbf{x}^{(N)}) & \cdots & \phi_p^1(\mathbf{x}^{(N)}) \end{bmatrix} \begin{bmatrix} c_0^1 \\ \vdots \\ c_p^1 \end{bmatrix} \\ \qquad\qquad + \cdots + \\ \begin{bmatrix} \phi_0^M(\mathbf{x}^{(1)}) & \cdots & \phi_p^M(\mathbf{x}^{(1)}) \\ & \ddots & \\ \phi_0^M(\mathbf{x}^{(N)}) & \cdots & \phi_p^M(\mathbf{x}^{(N)}) \end{bmatrix} \begin{bmatrix} c_0^M \\ \vdots \\ c_p^M \end{bmatrix} \end{cases} \tag{31}$$

By taking each set of N WFBFs as a Weighted Fuzzy Basis Vector, WFBV:

$$\overrightarrow{\phi_k^l} = \begin{pmatrix} \phi_k^l(\mathbf{x}^{(1)}) \\ \phi_k^l(\mathbf{x}^{(2)}) \\ \vdots \\ \phi_k^l(\mathbf{x}^{(N)}) \end{pmatrix}, \quad \begin{cases} l = 1, ..., M \\ k = 0, ..., p \end{cases} \tag{32}$$

and each set of N outputs as a vector, the output vector can be expressed as follows :

$$\overrightarrow{y_{TSK}} = \begin{cases} \begin{bmatrix} \overrightarrow{\phi_0^1} & \cdots & \overrightarrow{\phi_p^1} \end{bmatrix} \begin{bmatrix} c_0^1 \\ \vdots \\ c_p^1 \end{bmatrix} \\ \qquad\qquad + \cdots + \\ \begin{bmatrix} \overrightarrow{\phi_0^M} & \cdots & \overrightarrow{\phi_p^M} \end{bmatrix} \begin{bmatrix} c_0^M \\ \vdots \\ c_p^M \end{bmatrix} \end{cases} \tag{33}$$

Now we have to tune $p + 1$ parameters for each rule, i.e., M vectors of dimension $(p + 1)$.

$$\overrightarrow{c^l} = \begin{pmatrix} c_0^l \\ \vdots \\ c_p^l \end{pmatrix}, \quad l = 1, ..., M \tag{34}$$

If we define the l_{th} element of Φ_{TSK} as $\Phi_{TSK,l}$, we have:

$$\Phi_{TSK,l} = \begin{bmatrix} \overrightarrow{\phi_0^l}, & \cdots & , \overrightarrow{\phi_p^l} \end{bmatrix}, l = 1, .., M \tag{35}$$

the output vector (33) becomes :

$$\overrightarrow{y_{TSK}} = \Phi_{TSK,1} \overrightarrow{c^1} + \cdots + \Phi_{TSK,M} \overrightarrow{c^M} \tag{36}$$

In a matrix form, (36) becomes :

$$\overrightarrow{y_{TSK}} = \Phi_{TSK} \left[\overrightarrow{c^1} \cdots \overrightarrow{c^M} \right]^T \tag{37}$$

So the *Weighted Basis Function Matrix* (WBFM) Φ can be defined as:

$$\Phi_{TSK} = \left[\Phi_{TSK,1}, \ldots, \Phi_{TSK,M} \right] \tag{38}$$

The optimal parameters of the consequent conforms a vector, $\overrightarrow{c^l}$ in (34) are obtained when the error vector, $\left(\overrightarrow{y_{TSK}} - \Phi_{TSK} \left[\overrightarrow{c^1} \cdots \overrightarrow{c^M} \right]^T \right)$, must be perpendicular to all the weighted fuzzy basis vectors, $\overrightarrow{\phi_k^l}$ ($k = 0, \ldots, p$ and $l = 1, \ldots, M$), which are the columns of the WBFM Φ_{TSK}, as shown in Figure 4.
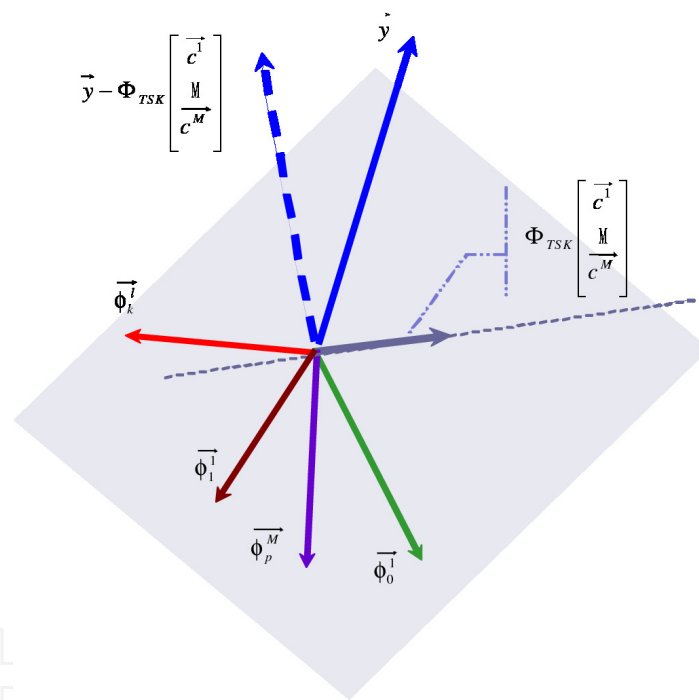

Figure 4. Extended Generalized Orthogonality Principle. The error vector $\overrightarrow{y} - \Phi \left[\overrightarrow{y_C^1} \cdots \overrightarrow{y_C^M} \right]^T$ should be perpendicular to all the fuzzy basis vectors, $\overrightarrow{\phi_k^l}$

This may be expressed directly in terms of the WBFM Φ as follows:

$$\Phi^T \overrightarrow{y} - \Phi^T \Phi \left[\overrightarrow{y_C^1} \cdots \overrightarrow{y_C^M} \right]^T = 0 \tag{39}$$

Solving for $\left[\overrightarrow{y_C^1} \cdots \overrightarrow{y_C^M} \right]^T$ provides the following

$$\left[\overrightarrow{y_C^1} \cdots \overrightarrow{y_C^M} \right]_{opt}^T = \left[\Phi \cdot \Phi^T \right]^{-1} \Phi \overrightarrow{y}$$

5. FLC design for controlling an inverted pendulum on a cart

5.1. Description of the system

Schematic drawing of an *Inverted pendulum On a Cart* (IPOC) system is depicted in Figure 5. where x is the position of the cart, θ is the angle of the pendulum with respect to the vertical direction and $\vec{F}$ is the external acting force in the $x - direction$. In order to keep the pendulum upright, we design a Fuzzy Logic Controller (FLC) using the GOP.
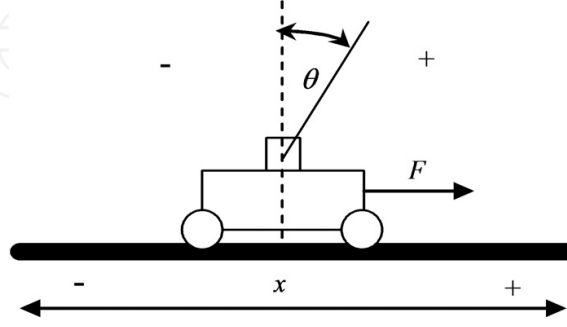

Figure 5. A schematic drawing of the inverted pendulum on a cart

The Lagrange equation for the position of the pendulum, θ, is given by:

$$\left(\frac{ml^2}{4} + J\right)\ddot{\theta} + \frac{ml}{2}(\ddot{x}\cos\theta - g\sin\theta) = 0 \tag{40}$$

The Lagrange equation for the position of the cart, x, is given by:

$$(M_1 + m)\ddot{x} + \frac{ml}{2}(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta) = F(t) \tag{41}$$

where J is the moment of inertia of the bar. The masses of the cart and the rod are $M_1 = 2Kg$ and $m = 0.1Kg$, respectively. The rod has a length $l = 0.5m$.

Since the goal of the control system is to keep the pendulum upright the equations can be linearized around $\theta = 0$. We chose $\mathbf{x} = \begin{bmatrix} \theta & \dot{\theta} & x & \dot{x} \end{bmatrix}^T$ as the state vector, where $\dot{\theta}$ is the pendulum angle variation and $\dot{x}$ is the cart position variation. The state representation is given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{6}{l(m+4M_1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-3g \cdot m}{m+4M_1} & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{6}{l(m+4M_1)} \\ 0 \\ \frac{4}{m+4M_1} \end{bmatrix} F(t) \tag{42}$$

$$\mu_{F_i^l}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - m_{F_i^l}}{\sigma_{F_i^l}}\right)^2\right] \tag{43}$$

5.2. FLC structure and design

We try to keep the pendulum upright regardless the cart's position, i.e., *Pure Angular Position Control System* (PAPCS). Then, the two inputs of the Fuzzy Logic Controller FLC are the angular pendulum position, θ, and its derivative, $\dot{\theta}$, i.e., $\mathbf{x}_1 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T$ and its output is the applied force to the system $y = force$.
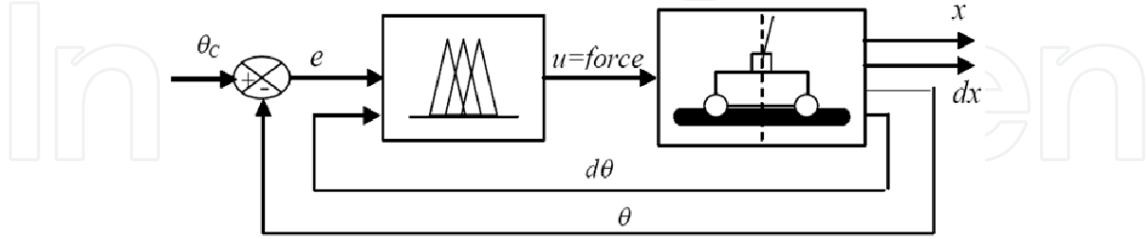
Figure 6. Fuzzy control system of the PAPCS

In this case, we use a Mamdani FLS with four rules. We use gaussian MF to fuzzify the two controller's inputs (44) and triangular MF to fuzzify the controller output.

$$\mu_{F_i^l}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - m_{F_i^l}}{\sigma_{F_i^l}}\right)^2\right] \tag{44}$$

where $m_{F_i^l}$ and $\sigma_{F_i^l}$ are respectively the centers and standard deviations of these MFs.

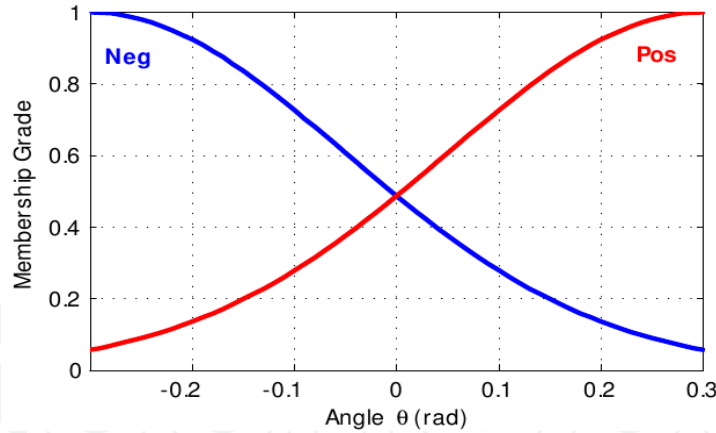The MFs of the antecedents are depicted in Figures 7 and 8.

Figure 7. Membership functions for the first controller input θ

Figure 9 shows the 56 data training and the optimal fitting given by the GOP method.

The obtained optimal consequent parameters are

$$\left(\overline{y}^1, \overline{y}^2, \overline{y}^3, \overline{y}^4\right)_{opt} = (-14.3, -14.23, 9.61, 18.96)$$

Figure 10 shows the response of the pendulum system controlled by the designed FLC to a reference $\theta_{ref} = 0$ with its response at the same reference when it is controlled by untuned FLC. The initial state vector is $\mathbf{x}_0 = \begin{bmatrix} \theta_0 & \dot{\theta}_0 & x_0 & \dot{x}_0 \end{bmatrix}^T = \begin{bmatrix} 0.1 & 0.2 & 0 & 0 \end{bmatrix}^T$.
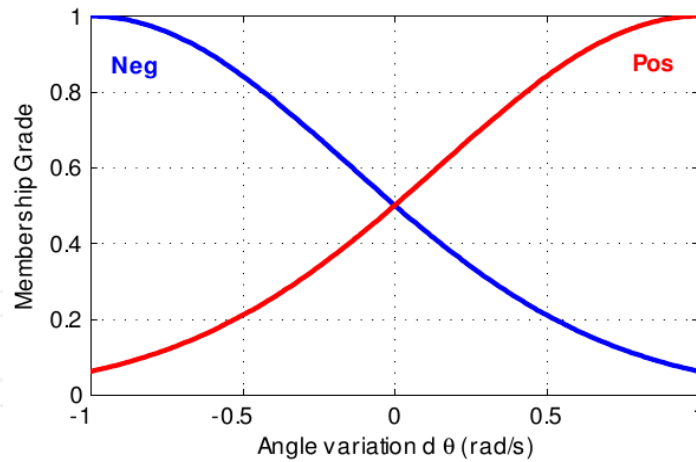
Figure 8. Membership functions for the second crontroller input $\dot{\theta}$
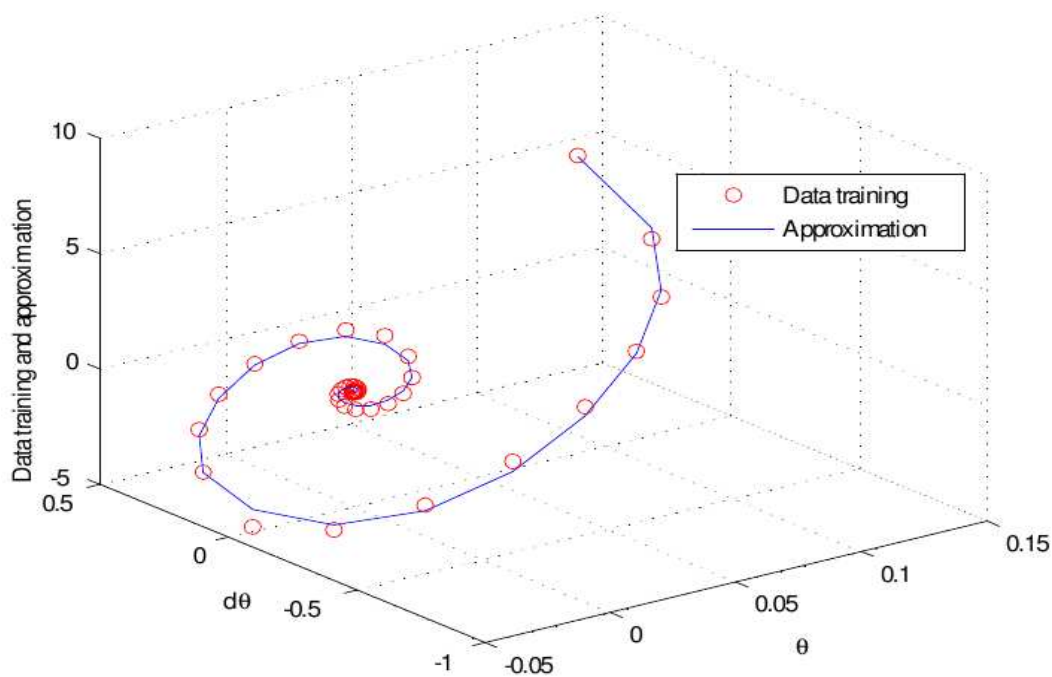

Figure 9. Data training and its approximation based on GOP

We evaluate the proposed design by using its error rate. For quantifying the errors, we use three different performance criteria to analyze the rise time, the oscillation behaviour and the behaviour at the end of transition period. These three criteria are: *Integral of Square Error* ($ISE = \int_0^\infty [e(t)]^2\,dt$), *Integral of the Absolute value of the Error* ($IAE = \int_0^\infty |e(t)|\,dt$) and *Integral of the Time multiplied by Square Error* ($ITSE = \int_0^\infty t\,[e(t)]^2\,dt$)

Table 1 summarizes the obtained values of *ISE*, *IAE* and *ITSE* of PAPCS using FLC, when tuning and no tuning are used.

We notice from this table that the errors obtained when tuning is used are all smaller than those obtained with untuned FLC. Fig. 11, 12, 13 show the different quantified errors.

Figures 10, 11, 12 and 13 show that the system using tuning is less oscillatory, having a rise time and errors at the end of transition period smaller than those obtained by untuned FLC.
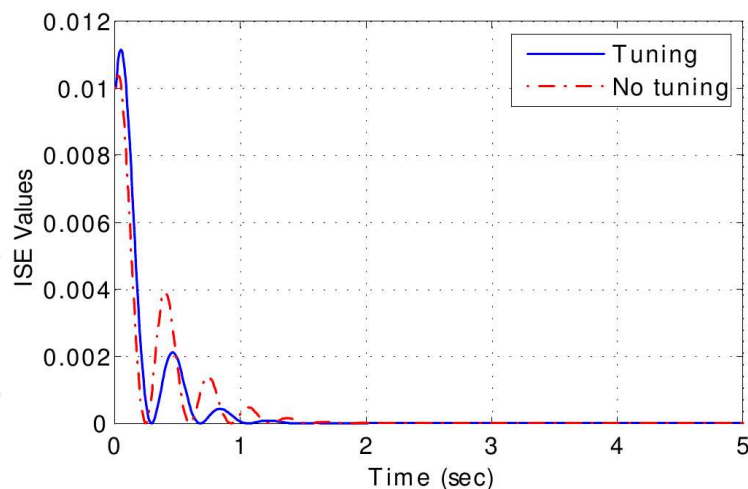
Figure 10. System responses of PAPCS controlled by a tuned and untuned FLC
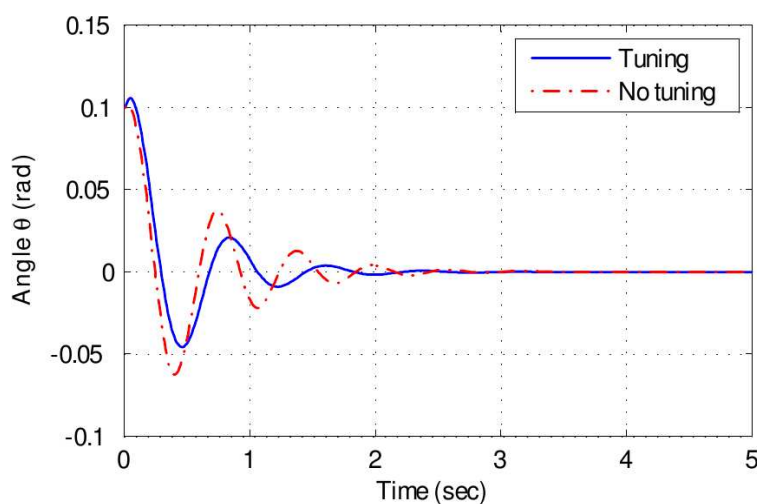

Figure 11. Integral of square error values of the PAPCS of tuned and no tuned consequent parameters. Rise time of the system is shorter for the tuned FLC
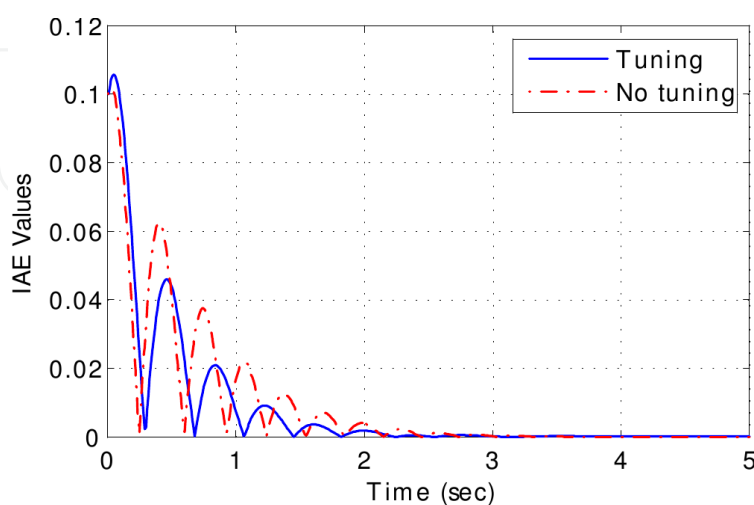

Figure 12. Integral of the absolute value of the error values of PAPCS of tuned and no tuned consequent parameters The system is less oscillatory for the tuned FLC before becoming stable

	No tuning	Tuning
ISE	0.2338	0.2224
IAE	4.7343	4.0403
ITSE	6.7733	4.4278

Table 1. Comparison of performance criteria for of PAPCS using tuned and no tuned FLC.
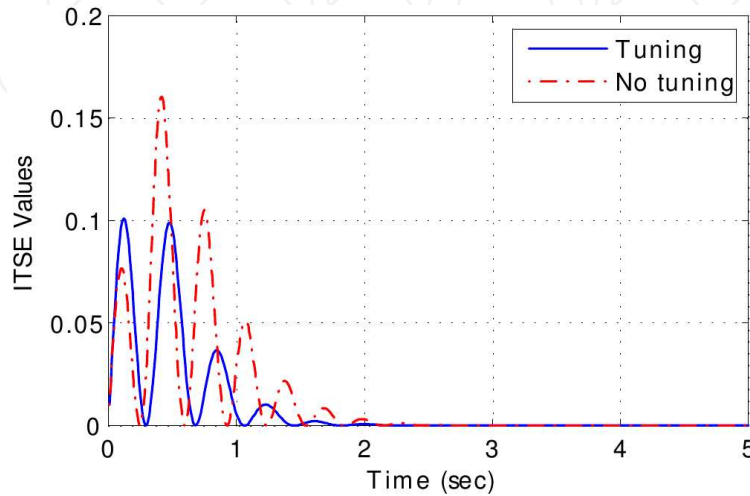

Figure 13. Integral of the time multiplied by square error values of PAPCS of tuned and no tuned consequent parameters. The error at the end of transition period is less important for the tuned FLC

6. FLS design for predicting time series

We apply the GOP to design an FLS which predicts a time series. The FLS has to predict the future value $x(t+6)$ of a Mackey-Glass time series (45) which is volatile. The following four antecedents were used: $x(t-18)$, $x(t-12)$, $x(t-6)$ and $x(t)$, which are known values of the time series ([2], [3]).

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{45}$$

The training data are obtained by simulating (45) for $\tau = 17$. We use the samples $x(1001), \cdots, x(1524)$ to train the IT2 FLS and the samples $x(1501), \cdots, x(2024)$ for testing. We use two Gaussian MFs per antecedent, so we have then 16 rules. The MFs of the antecedents are Gaussian, where its mean and the standard deviation were obtained from the 524 training samples, $x(1001), \cdots, x(1524)$. Table 2 summarizes the consequent parameters per each rule.

Figure 14 displays performance of the FLS in training data, and Figure 15 shows its results on Testing data. Note that the GOP-designed is a better forecaster, since the differences from original data are small in both training and testing data sets.

Some additional analyses should be performed to verify the goodness of fit of the method (See [5], and [6]), but in this case, the proposed GOP has shown good results, so we can recommend its application to real cases. Time series analysis is an useful topic for many decision makers, so the use of optimal and easy-to-be-implemented techniques, as the proposed one has a wide potential.

R^l	c_1	c_2	c_3	c_4	c_5
R^1	-3.58	7.94	-9.17	0.03	0.78
R^2	9.92	-10.9	-0.02	1.29	-7.83
R^3	16.05	7.58	12.40	5.25	-33.96
R^4	8.92	-6.78	-12.22	3.68	-4.13
R^5	1.06	4.64	28.42	-40.17	0.16
R^6	22.57	-33.28	-12.43	17.13	-12.76
R^7	-2.93	-7.65	5.73	-2.91	-1.30
R^8	22.88	26.23	-15.79	-6.19	-0.60
R^9	-3.86	4.36	2.04	0.21	4.77
R^{10}	27.72	-45.35	24.63	7.92	6.26
R^{11}	-0.24	-4.99	30.94	-26.54	6.65
R^{12}	2.36	5.34	-26.93	18.21	-8.03
R^{13}	-30.66	13.37	5.27	3.60	1.43
R^{14}	23.62	-21.97	-3.87	6.04	8.01
R^{15}	3.70	-5.07	0.61	-0.76	8.38
R^{16}	-25.05	11.30	-0.42	1.27	4.64

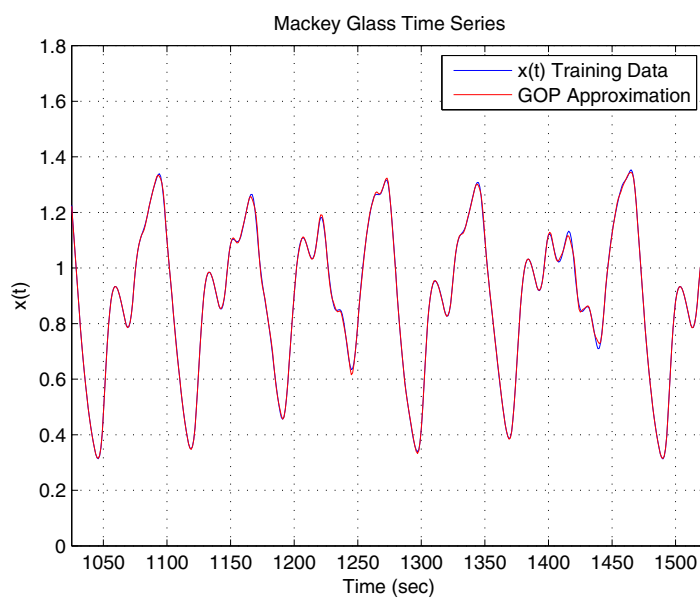**Table 2.** The optimal TSK FLS consequent parameters obtained by GOP design.

Figure 14. Mackey-Glass time series. The samples $x(1001), \cdots, x(1524)$ are used for designing the FLS forecaster
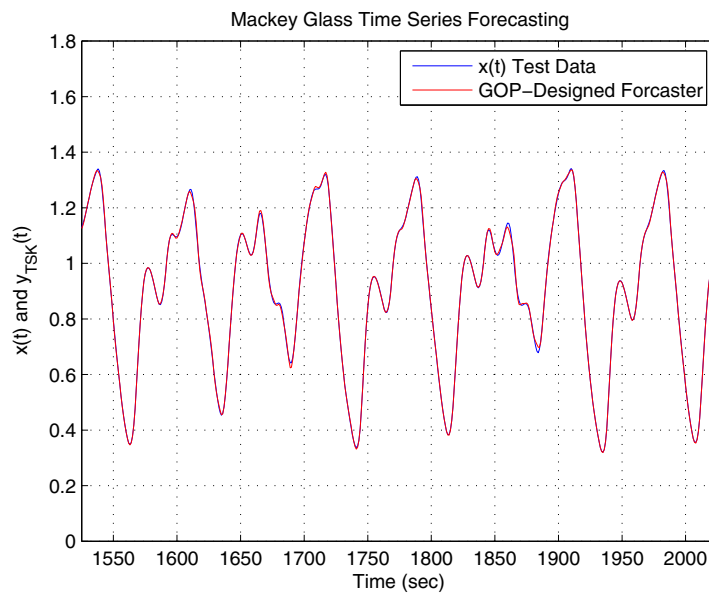
Figure 15. Output of the TSK FLS time-series forecaster. The samples $x(1525), \cdots, x(2024)$ are used for testing the GOP design

7. Concluding remarks

In this chapter we have presented an enhancement method of fuzzy controllers using the generalized orthogonality principle. We applied the method to two different cases: a first one involving control of an inverted pendulum and a second one for fuzzy forecasting. In the first application, numerical rules and their FBFs were extracted from numerical training data. This combination of both linguistic and numerical information simultaneously become FBFs an useful method. Since a specific FLS can be expressed as a linear combination of FBFs, we generalized orthogonality principle on FBFs that results in a better FLS.

In the second study case, we applied the GOP to design a FLS for time series forecasting. The FLS has been applied to a Mackey-Glass time series with better results compared to a non-GOP FLS. The results were validated with simulations.

All the FBFs can be seen as a basis vector, which allows to optimize the parameters of the consequents. This means that the error vectors are orthogonal to these FBFs, resulting in the minimization of the magnitudes of these error vectors, and consequently an optimal FLS.

The proposed method has a wide potential in complex forecasting problems ([5], and [6]). Its application to hardware design problems ([7]) can improve the performance of fuzzy controllers, so its implementation arises as a new field to be covered.

Author details

Nora Boumella
University of Batna, Batna - Algeria

Juan Carlos Figueroa
Universidad Distrital Francisco Jose de Caldas, Bogota - Colombia

Sohail Iqbal
NUST-SEECS, Islamabad - Pakistan

8. References

[1] Berenji, H. & Khedkar, P. [1992]. Learning and tuning fuzzy logic controllers through reinforcements, *IEEE Trans. Neural Networks* 3(1): 724–740.

[2] Boumella, N., Djouani, K. & Boulemden, M. [2011]. On an Interval Type-2 TSK FLS A1-C1 consequent parameters tuning, *in* IEEE (ed.), *Proc. SSCI 2011 T2FUZZ - 2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems*, IEEE, pp. 1–6.

[3] Boumella, N., Djouani, K. & Boulemden, M. [2012]. A robust Interval Type-2 TSK fuzzy logic system design based on chebyshev fitting, *International Journal of Control, Automation, and Systems* 10(4).

[4] Boumella, N., Djouani, K. & Iqbal, S. [2009]. A new design of fuzzy logic controller based on generalized orthogonality principle, *in* IEEE (ed.), *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2009*, IEEE, pp. 497–502.

[5] Figueroa, J. C. [2009]. An evolutive Interval Type-2 TSK fuzzy logic system for volatile time series identification, *2009 Conference on Systems, Man and Cybernetics*, IEEE, pp. 1–6.

[6] Figueroa, J. C., Kalenatic, D. & Lopez, C. A. [2010]. A neuro-evolutive Interval Type-2 TSK fuzzy system for volatile weather forecasting, *Lecture Notes in Computer Science* 6216: 142–149.

[7] Melgarejo, M. & Peña, C. A. [2007]. Implementing Interval Type-2 fuzzy processors I, *Computational Intelligence Magazine* 2(1): Pág. 63–71.

[8] Mendel, J. [2001]. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice Hall.

[9] Strang, G. [1988]. *Linear Algebra and its Applications, Third edition*, Brooks/Cole.

[10] Wang, L. [1992]. Fuzzy systems are universal approximators, *in* IEEE (ed.), *Proc. IEEE Int'l. Conf. on Fuzzy Systems, San Diego, CA*, IEEE, pp. 1163–1170.

[11] Wang, L. & Mendel, J. M. [1992]. Fuzzy basis functions, universal approximation, and orthogonal least squares learning, *IEEE Trans. Neural Networks* 3(5): 807–814.