

A TSK Neuro-Fuzzy Approach for Modeling Highly Dynamic Systems

Giovanni Acampora
Department of Computer Science
University of Salerno
Fisciano, Salerno 84084
Email: gacampora@unisa.it

Abstract—This paper introduces a new type of TSK-based neuro-fuzzy approach and its application to modeling highly dynamic systems. In details, our proposal performs an adaptive supervised learning on a collection of time series in order to create a so-called Timed Automata Based Fuzzy Controller, i.e. an evolvable fuzzy controller whose dynamic features yield high performances in variable structure systems representation. The adaptive learning is accomplished by merging together theories from the area of times series analysis such as the Adaptive Piecewise Constant Approximation method, with a well-known neuro-fuzzy framework, the Adaptive Neuro Fuzzy Inference System. As will be shown in our experiments, where our proposal has been tested on a Fuzz-IEEE 2011 Fuzzy Competition dataset, this approach reduces the output error measure and achieves a better performance than a standard application of the ANFIS algorithm when applied to highly dynamic systems.

Index Terms—Dynamic Systems Modeling, Neuro-Fuzzy Systems, Time Series Approximation, Variable Structure Systems, Timed Automata based Fuzzy Controllers.

I. INTRODUCTION

Traditional systems modeling exploits conventional mathematical methods, such as differential equations, that are not able to deal with complex systems characterized by high levels of imprecision and uncertainty. By contrast, the *fuzzy control theory* [1][2] uses collections of linguistic *if...then* rules in order to model the above mentioned systems by considering qualitative aspects of human knowledge and reasoning processes without employing a precise quantitative analysis[3]. Thanks to their versatility, fuzzy systems have found a great variety of applications ranging from control engineering, qualitative modeling, pattern recognition, signal processing, information processing, machine intelligence, decision making, management, finance, medicine, motor industry, robotics, and so on [4].

However, in spite of their wide applicability, fuzzy systems suffer from some design weaknesses:

- 1) there are no recognized techniques for capturing human knowledge or experience through a fuzzy rule base;
- 2) some effective methods for learning and/or tuning the membership functions are necessary to minimize the output error measure or maximize performance index;
- 3) fuzzy control paradigm does not allow the modeling of systems characterized by a discontinuous nonlinear behavior or, in other words, it cannot directly represent

highly dynamic systems and Variable-Structure Systems (VSSs)[5].

In order to solve these drawbacks, this paper introduces an adaptive supervised learning algorithm, based on a neuro-fuzzy approach, able to infer the different behaviors that a system exhibits during its life-cycle, and return a so-called *Timed Automata based Fuzzy Controller (TAFC)*, i.e. a novel kind of evolvable fuzzy inference engine whose dynamic features yield high performances in dynamic systems representation. The main idea underlying the learning algorithm is to identify the time ranges where the system shows a behavior characterized by soft oscillations and, for each range, to compute the most appropriate TSK fuzzy controller [6] modeling the system behavior in that range.

More in detail, the proposed neuro-fuzzy approach learns a suitable TAFC for modeling a given system by performing two sequential subtasks. The first subtask, that is based on the *Adaptive Piecewise Constant Approximation (APCA)* method [7], analyzes a collection of time series representing the temporal evolution of the system variables and identifies a set of disjointed temporal ranges where such variables are simultaneously characterized by soft oscillations or, in other words, by a low variance. This choice will allow the following subtask to work on a more reliable data collection. The second subtask takes as inputs the temporal ranges computed by the first subtask and, for each range, uses the *Adaptive Neuro-Fuzzy Inference Engine (ANFIS)* in order to compute the most suitable TSK controller that models the system behavior in that range. Finally, the computed temporal ranges and TSK controllers will be used to determine the main components of a TAFC modeling the whole system life-cycle.

As will be shown in experiments, where our proposal has been tested on a Fuzz-IEEE 2011 Fuzzy Competition dataset, the TAFC obtained by means of the proposed neuro-fuzzy approach reduces the output error measure error and achieves performances better than a conventional application of the ANFIS algorithm, when applied to highly dynamic systems.

II. BASIC CONCEPTS AND BACKGROUND

The following subsections will be devoted to present the main methodologies used by our approach: TAFC, APCA and ANFIS. Successively, the section III will formally introduce the proposed supervised learning algorithm.

A. Timed Automata based Fuzzy Controllers

A complete and theoretical discussion about the TAFC definition can be found in [8]. However, in this section a brief description about the most important features characterizing this evolvable inference engine is presented.

TAFCs are evolvable fuzzy systems mainly useful for modeling variable structure systems. In particular, they extend standard fuzzy controllers (Mamdani or TSK) by using three additional concepts: *control eras*, *control configurations* and *control time*. By means of these concepts, the life-cycle of a given system can be considered as composed of a sequence of disjointed time intervals (the control eras) where each era is characterized by a well-defined fuzzy controller (the control configuration) suitable for controlling the system during that era; the control time is the mechanism that allows the system to switch between adjacent control eras; in other words, the control time set the current control configuration of the system and the duration of the control eras.

In order to extend the fuzzy control vision with above said concepts, a modified version of Timed Automata has been used. A timed automaton is a standard finite-state automaton extended with a finite collection of real-valued clocks providing a straightforward way to represent time related events. Conventional automata-based approaches cannot offer this feature. The clocks can be reset to 0 (independently of each other) with the transitions of the automaton, and keep track of the time elapsed since the last reset. The transitions of a timed automaton are labeled with a *guard* (a condition on clocks), an *action* or *symbol* on alphabet Σ , and a *clock reset* (a subset of clocks to be reset). Intuitively, a timed automaton starts execution with all clocks set to zero. Clocks increase uniformly with time while the automaton is within a node. A transition may be taken only if the current values of the clocks satisfy the associated constraints. By taking the transition, all clocks in the clock reset will be set to zero, while the remaining keep their values.

In our vision, TAFCs use the nodes of a timed automaton to implement control era and control configuration, whereas the clocks and transitions realize the control time idea. In particular, when a TAFC transits from a state S_1 to a state S_2 through a transition T_1 , an opportune control configuration (or fuzzy controller) C is activated to manage the system until a successive transition T_2 will be taken. The time that elapses between the two transitions is the duration of the control era related to the control configuration C . The control configuration C is obtained by transforming a previous control configuration related to the state S_1 by means of particular transformation operators that are related to the transition T_1 . In details, transformation operators change a TAFC's control configuration by means of the following operations: adding ($\oplus$) or deleting ($\ominus$) a variable; adding ($\boxplus^k$), removing ($\boxminus^k$) or changing ($\boxdot^k$) k rules in the rule base; changing implication method of the rule base ($\sim$); adding ($\otimes$), deleting ($\oslash$) or changing ($\odot$) a term to a variable; changing defuzzify method ($\propto$), aggregation method ($\bowtie$) or default value ($*$) of an output variable; changing lower bound ($\ltimes$) or upper bound ($\rtimes$) of

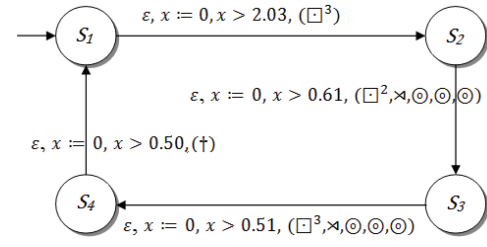


Fig. 1. A Sample of Timed Automata based Fuzzy Controller: an evolution scheme for the Matlab™ Tipper

the universe of discourse of a variable. This process is iterated for each transition taken. A sample of TAFC is shown in Fig. 1. This TAFC evolves the classical tipper controller used as sample of fuzzy system in Mathworks Matlab™. The Tipper system manages the following issue: what is the right amount to tip your waitperson working in a restaurant placed in a sea village? The system determines the amount of tip by analyzing two data: the quality of service and the quality of served food. Although, it may seem a very simple system, the Tipper can be considered as a dynamic system since it changes its structure and behaviour over the time. Indeed, usually, the tip amount is higher in high season than low season and, as consequence, the fuzzy variable modeling this value should be gradually modified among the four year seasons. In details, it is possible to individuate four periods where the Tipper can show different behavior: from october 10th to april 30th, from may 1th to june 30th, from july 1th to august 20th and from august 21th to october 9th.

As proved in [9], where a comparison between a standard TSK controller and a TAFC has been accomplished, a TAFC results more efficient when it is devoted to control a dynamic system such as the Tipper.

In following sections, an introduction to APCA and ANFIS will be provided in order to define a supervised learning algorithm capable of analyzing a collection of time series related a given system and learning the control eras, the control configurations and the control time of the TAFC most suitable to model a system.

B. Adaptive Piecewise Constant Approximation

The APCA is a method for time series approximation originally designed for solving the problem of similarity search in large time series databases. In particular, APCA is a method for approximating a time series by dividing it into M varying-length segments and recording the mean value of the datapoints that fall within the segment. In our context, this approach is used to identify the collection of temporal ranges where a system shows a soft oscillation behavior. Hereafter, an introduction to APCA is provided starting from the time series definition.

A time series is a sequence of data points, measured at successive times spaced at uniform time intervals. A sample of time series is shown in Fig. 2. A common formal notation

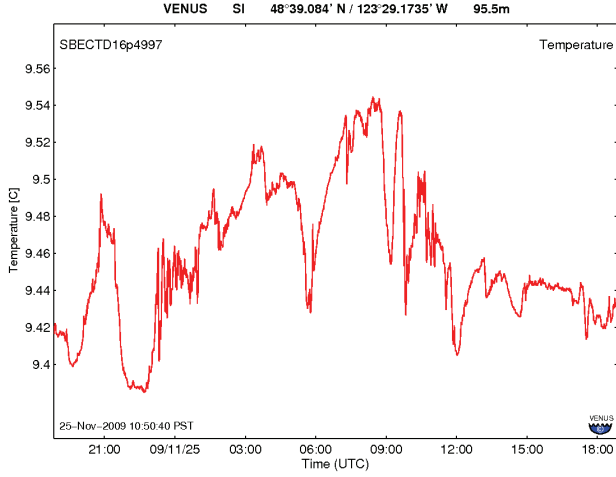


Fig. 2. A Sample of Time Series for Representing Temperature Variations.

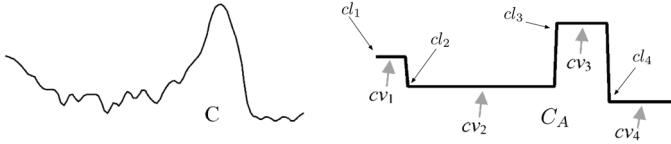


Fig. 3. A time series C and its APCA representation C_A , with $M = 4$.

specifying a time series C is:

$$C = \{c_1, c_2, c_3, \dots, c_n\} \quad (1)$$

where c_i is the i^{th} datapoint in time series C sampled at the times $\{t_i, i = 1, \dots, n\}$, with $t_i - t_{i-1} = \delta$ and $\delta \in \mathbb{R}$ (δ is the fixed sampling step).

Given a time series $C = \{c_1, \dots, c_n\}$, an APCA approximation is represented as:

$$C_A = \{\langle cv_1, cl_1 \rangle, \dots, \langle cv_M, cl_M \rangle\} \quad (2)$$

where cv_i is the mean value of datapoints in the i^{th} segment (i.e., $cv_i = \text{mean}(c_{cl_i}, \dots, c_{cl_{i+1}-1})$) and cl_i the left endpoint of the i^{th} segment. The length of the i^{th} segment can be calculated as $(cl_{i+1} - cl_i)$. Fig. 3 illustrates this notation.

In our neuro-fuzzy context, the APCA is used to analyze the temporal behavior of the variables related to a given system and compute a collection of temporal ranges enabling the proposed learning algorithm to learn the set of desired TAFC control eras.

C. Adaptive Neuro Fuzzy Inference System

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is a class of adaptive networks that are functionally equivalent to conventional fuzzy inference systems. An adaptive network is a multilayer feed forward network in which each node performs a particular function (node function) on incoming signals using a set of parameters specific to this node [3]. The form of node functions may vary from node to node, and the choice of each node function depends on the overall

function, which the adaptive network is designed to implement. Adaptive networks are composed by two different classes of nodes: *adaptive nodes* and *fixed nodes*. Adaptive nodes are characterized by a collection of modifiable parameters, named *parameter set*, whereas, the fixed nodes only deal with an unmodifiable parameters. Adaptive networks work in order to achieve a desired input-output by updating the parameter sets according to given training data and a gradient-based update procedure [10]. ANFIS exploit this updating feature to learn and adapt the parameters of a given TSK fuzzy inference system.

As example, Fig. 4 shows an adaptive network that models a first-order TSK fuzzy inference systems composed by two rules:

- 1) If x is A_1 and y is B_1 Then $f_1 = p_1 \cdot x + q_1 \cdot y + r_1$
- 2) If x is A_2 and y is B_2 Then $f_2 = p_2 \cdot x + q_2 \cdot y + r_2$

To reflect the ANFIS adaptive capabilities, in Fig. 4 the square nodes represent the adaptive nodes, whereas the circle nodes represent the fixed nodes.

Each ANFIS layer implements a particular fuzzy inference subtask. In detail:

Layer 1 Every node i in this layer implements the node function:

$$O_i^1 = \mu_{A_i}(x) \quad (3)$$

where x is the input to node i , and A_i is a linguistic label. From a fuzzy inference point of view, O_i^1 is the membership function related to the fuzzy set A_i , and it specifies the degree to which the input x satisfies A_i . Typically, A_i is a bell-shaped function characterized by a set of three parameters $\{a, b, c\}$ corresponding to the parameter set of the node i .

Layer 2 The nodes labeled with Π multiply their input signals:

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \text{ with } i = 1, 2; \quad (4)$$

where w_i represents the firing strength of a rule.

Layer 3 The node in this layer calculate the ratio of the i^{th} rules firing strength to the sum of all rules firing strengths:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2} \text{ with } i = 1, 2. \quad (5)$$

The value w_i represents the *normalized firing strenght*.

Layer 4 The adaptive node i in this layer computes the following function:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (6)$$

where the collection $\{p_i, q_i, r_i\}$ is the parameter set of the node i and corresponds to the TSK consequent parameters.

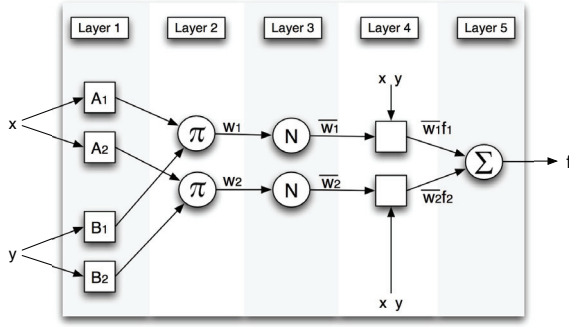


Fig. 4. The Adaptive Neuro Fuzzy Inference System.

Layer 5 The single and fixed node in this layer, labeled Σ , computes the overall network output as the summation of all incoming signals:

$$O_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{w_i} \quad (7)$$

In our learning context, ANFIS is used to compute a collection of TSK controllers, each one useful to control a given system during a particular control era previously computed by the APCA approach.

III. A SUPERVISED LEARNING ALGORITHM FOR MODELING HIGHLY DYNAMIC SYSTEMS THROUGH TAFCS

This section formally introduces our proposal of a neuro-fuzzy approach for learning the TAFCS most suitable to modeling the behavior of a given system. The proposed algorithm achieves this goal by taking as input a collection of time series representing the temporal evolution of the different variables of the system. The main idea underlying the learning algorithm is to identify the time ranges where the system shows a behavior characterized by soft oscillations and, for each range, to compute the most appropriate TSK controller modeling the system behavior in that range. Starting from the time ranges and TSK controllers the algorithm returns the control eras, the control configurations and the control time of the TAFCS that better models the system under modeling.

The algorithm is essentially based on two steps, the *temporal learning step* and *qualitative learning step* (see Fig. 5). The first step uses the APCA in order to examine the time series and computes the sequence of aforementioned time intervals; the second step, taking as input the sequence of time intervals, returns the appropriate (one for each time interval) TSK controller, computed by means of ANFIS.

A. Temporal Learning Step

Let S be a system characterized by n_i input variables and n_o output variables belonging, respectively, to the following sets:

$$\begin{aligned} V_i &= \{v_j, j = 1, \dots, n_i\} \\ V_o &= \{v_j, j = n_i + 1, \dots, n_i + n_o\} \end{aligned} \quad (8)$$

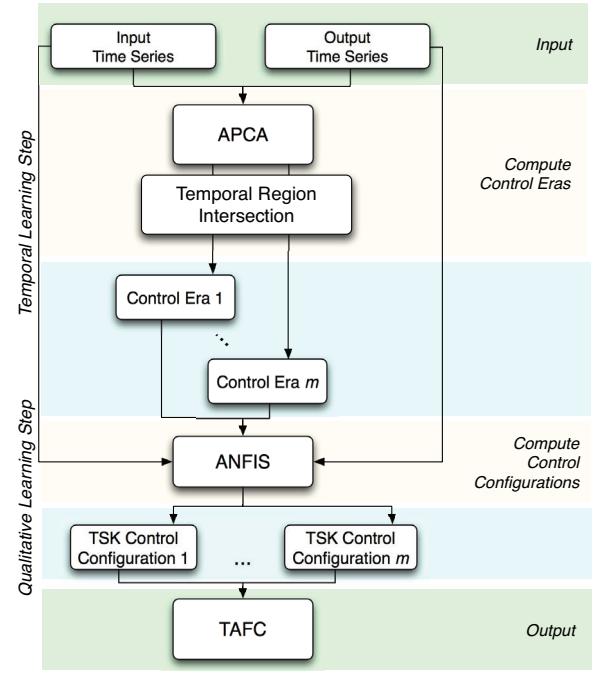


Fig. 5. The Supervised Learning Algorithm for modeling Highly Dynamic Systems: a Flow Chart

For each variable $v_j \in V_i \cup V_o$, let $C^j = \{c_1^j, c_2^j, \dots, c_n^j\}$ be a time series containing the values of the variable v_j sampled at the time t_i with $i = 1, \dots, n$ by a fixed sampling step δ .

The main aim of the temporal learning step is to analyze, for each output variable $v' \in V_o$, the time series C^j in order to identify the temporal ranges where the variables $v_j \in V_i \cup \{v'\}$ are simultaneously characterized by soft oscillations. In order to achieve this goal, the APCA method is applied. More in detail, the APCA method is executed on each time series C^j in order to compute the following time series approximation:

$$C_A^j = \{\langle cv_1^j, cl_2^j \rangle, \dots, \langle cv_{M_j}^j, cl_{M_j}^j \rangle\}$$

where cv_i^j is the mean value of variable v_j in the i^{th} segment and $cl_i^j \in \{t_1, t_2, \dots, t_n\}$ is the left endpoint of the i^{th} segment. Successively, the algorithm extracts from C_A^j a collection of M_j time instants, $T^j = \{cl_1^j, cl_2^j, \dots, cl_{M_j}^j\}$, corresponding to the left endpoints of the time ranges where the variable v_j is characterized by soft oscillations. A sample of this process, with $M = 9$, is shown in Fig. 6.

Once APCA has computed the collection T^j for each variable $v_j \in V_i \cup V_o$, it is necessary to identify a further collection of time instants, $\mathbf{T} = \{\tau_1, \tau_2, \dots, \tau_p\}$, corresponding to the left endpoints of the time ranges where the system variables are 'simultaneously' characterized by soft oscillations. This operation is performed by a *temporal intersection operator*. We give an algorithmic representation of this operator (see Algorithm 1) and an example of temporal intersection applied to two sequences $T^1 = \{0, 27, 289, 437, 595, 759\}$ and $T^2 = \{0, 27, 389, 547\}$ whose temporal intersection returns $\mathbf{T} = \{0, 27, 289, 389, 437, 547, 595, 759\}$ (see Fig. 8).

Starting from $\mathbf{T} = \{\tau_1, \tau_2, \dots, \tau_p\}$, the algorithm computes the collection of control eras as follows:

$$E = \{\langle \tau_1, \tau_2 - \delta \rangle, \langle \tau_2, \tau_3 - \delta \rangle, \dots, \}$$

We remind that δ is the fixed sampling step related to time series C^j . In order to compute the TAFC control configurations, the qualitative learning step prepares a collection of datasets, each one of the related to a given control era in E , and successively uses ANFIS to complete the learning process.

Algorithm 1 T-INTERSECTION($T^1, \dots, M^1, \dots$)

Require: The collections $T^1, T^2, \dots, T^{n_i+n_o}$

Ensure: The collection $\mathbf{T} = \{\tau_1, \tau_2, \dots, \tau_p\}$.

```

1:  $\mathbf{T} \leftarrow \emptyset$ 
2:  $k \leftarrow 1$ 
3: for  $i = 1$  to  $|T^1| + |T^2| + \dots + |T^{n_i+n_o}|$  do
4:   for  $j = 1$  to  $(n_i + n_o)$  do
5:     if  $|T^j| > 0$  then
6:        $m_j \leftarrow \min(T^j)$ 
7:     else
8:        $m_j \leftarrow +\infty$ 
9:     end if
10:     $\{m_j \text{ is the minimum value of } T^j\}$ 
11:  end for
12:   $\tau \leftarrow \min(\{m_j, j = 1, \dots, n_i + n_o\})$ 
13:  if  $(\tau \notin \mathbf{T})$  then
14:     $\tau_k \leftarrow \tau$ 
15:     $\mathbf{T} \leftarrow \mathbf{T} \cup \{\tau_k\}$ 
16:     $k \leftarrow k + 1$ 
17:  end if
18: return  $\mathbf{T}$ 

```

B. Qualitative Learning Step

The qualitative learning step applies the ANFIS system to a collection of dataset $D = \{D_i, i = 1, \dots, |E|\}$ where D_i is the dataset related to the control era $\langle \tau_i, \tau_{i+1} - \delta \rangle$. Precisely, D_i is a matrix with $n_i + 1$ rows and $\frac{(\tau_{i+1} - \delta) - \tau_i}{\delta}$ columns where the j^{th} row corresponds to the values taken by variable v_j at time instants:

$$\tau_i, \tau_i + \delta, \tau_i + 2 \cdot \delta, \dots, \tau_i + \frac{(\tau_{i+1} - \delta) - \tau_i}{\delta} \cdot \delta$$

These values are extracted from the time series C^j related to the variable v_j .

Once each dataset D_i has filled, the qualitative learning step uses ANFIS in order to learn the best suitable TSK controller modeling the system S in the i^{th} control era: $\phi_i = \text{ANFIS}(D_i)$. The set $\Phi = \{\phi_i, i = 1, \dots, |E|\}$ is the collection of control configurations related to the TAFC modeling the system S and the time instants $\tau_i + \frac{(\tau_{i+1} - \delta) - \tau_i}{\delta} \cdot \delta$ ($i = 1, \dots, |E| - 1$) are used by the control time to move the TAFC from the state corresponding to the controller ϕ_i to the next state corresponding to the controller ϕ_{i+1} .

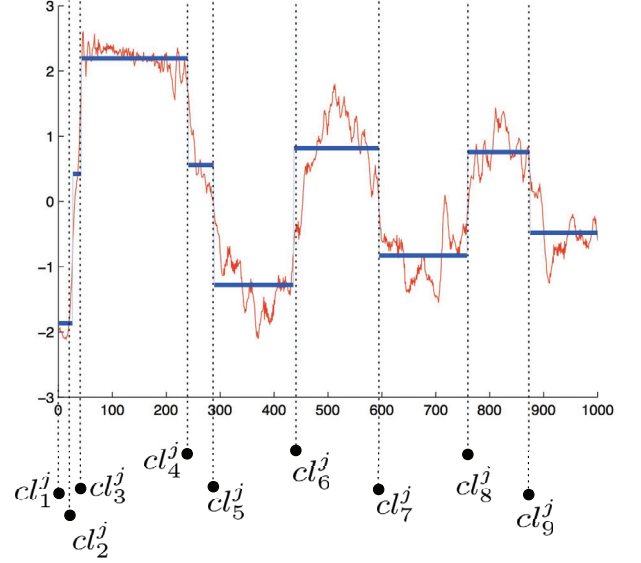


Fig. 6. The time ranges computed by APCA.

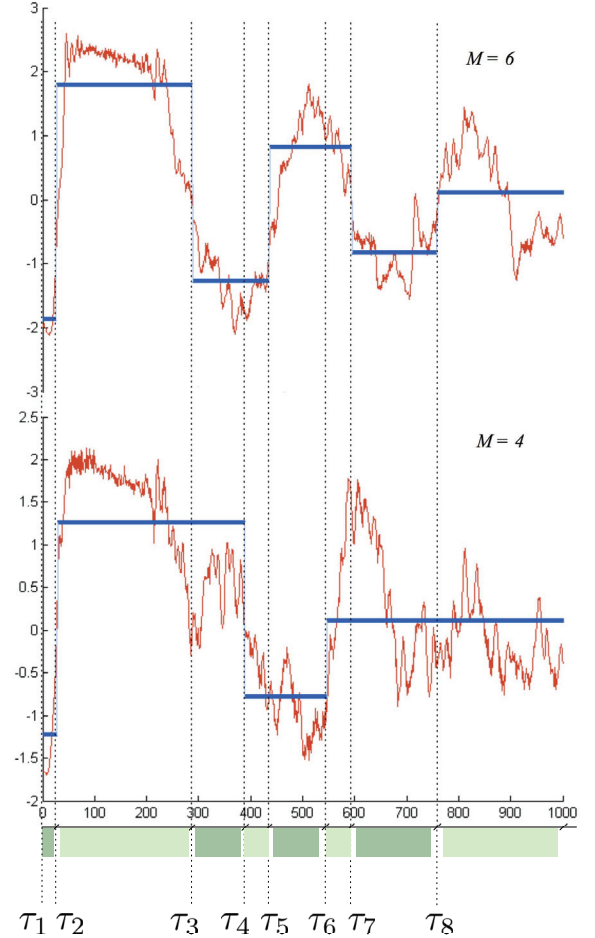


Fig. 7. The time ranges computed by the temporal intersection operator.

IV. EXPERIMENTAL RESULTS

In order to verify and validate the benefits provided by our neuro-fuzzy approach, a dataset from FUZZ-IEEE 2011

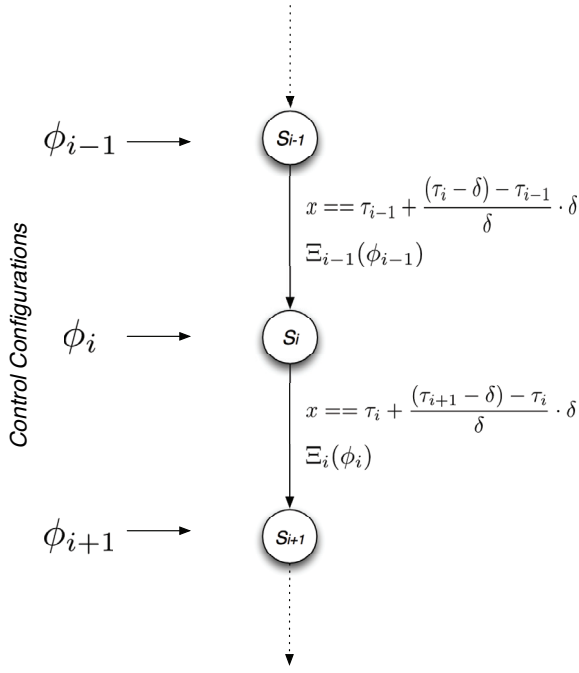


Fig. 8. A portion of TAFC returned by the learning algorithm.

Competition has been used. This dataset is related to a Multi-Input Multi-Output system with output couplings. An important feature of such a system is that each of the system's output variables may depend on all the input variables.

In particular, this competition dataset involves a nonlinear coupling system with four inputs and four outputs. Through data acquisition, 1000 input-output data points (namely, $u_1(n)$, $u_2(n)$, $u_3(n)$, $u_4(n)$, $y_1(n)$, $y_2(n)$, $y_3(n)$, $y_4(n)$, $n = 1, 2, \dots, 1000$) were collected from the system with the sampling period $T_s = 0.0005s$. This data acquisition is used to create eight different time series $C_1, C_2, \dots, C_8$, where the series $C_1, \dots, C_4$ correspond to the four input variables, whereas the time series C_5 and C_8 correspond to the output variables.

The proposed neuro-fuzzy modeling system approach applies the APCA method to the time series C_i with $i = 1, \dots, 8$ in order to obtain $M_1 = 4$ temporal segments for the variable u_1 , $M_2 = 4$ temporal segments for the variable u_2 , $M_3 = 5$ temporal segments for the variable u_3 , $M_4 = 4$ temporal segments for the variable u_4 , $M_5 = 8$ temporal segments for the variable y_1 , $M_6 = 8$ temporal segments for the variable y_2 , $M_7 = 6$ temporal segments for the variable u_3 and $M_8 = 6$ temporal segments for the variable y_4 . In particular, for each output variable, the approach uses APCA method in order to compute the time instants corresponding to the left endpoints where the variables shows a soft oscillation. For example, considering the output variable y_4 , the following collections

are obtained:

$$\begin{aligned} T^1 &= \{0, 0.126, 0, 0.225, 0.297\} \\ T^2 &= \{0, 0.159, 0.283, 0.379\} \\ T^3 &= \{0, 0.056, 0.192, 0.291, 0.381\} \\ T^4 &= \{0, 0.176, 0.268, 0.334\} \\ T^8 &= \{0, 0.014, 0.17, 0.215, 0.268, 0.335\} \end{aligned}$$

The temporal intersection operator 'merges' together the collection T^1, T^2, T^3, T^4, T^8 and obtains the time instants collection $\mathbf{T}$:

$$\mathbf{T} = \{0, 0.0140, 0.056, 0.126, 0.159, 0.170, 0.176, 0.192, 0.215, 0.225, 0.268, 0.283, 0.291, 0.297, 0.335, 0.379, 0.381\}$$

The collection $\mathbf{T}$ is used to learn 17 TAFC control eras:

$$\begin{aligned} E &= \{\langle 0, 0.0135 \rangle \langle 0.0140, 0.0555 \rangle, \langle 0.056, 0.1255 \rangle, \\ &\langle 0.126, 0.1585 \rangle \langle 0.159, 0.1695 \rangle, \\ &\langle 0.170, 0.1755 \rangle \langle 0.176, 0.1915 \rangle, \\ &\langle 0.192, 0.2145 \rangle, \langle 0.115, 0.2245 \rangle, \\ &\langle 0.225, 0.2675 \rangle \langle 0.268, 0.2825 \rangle, \\ &\langle 0.283, 0.2905 \rangle \langle 0.291, 0.2965 \rangle, \\ &\langle 0.297, 0.3345 \rangle \langle 0.335, 0.3785 \rangle \\ &\langle 0.379, 0.3805 \rangle \langle 0.381, 0.5 \rangle\} \end{aligned}$$

Starting from E , the collection of ANFIS dataset $D = \{D_i, i = 1, \dots, 17\}$ is computed in order to learn the control configurations $\Phi = \{\phi_i, i = 1, \dots, 17\}$ and the control time of the TAFC modeling the behavior of the output variable y_4 . The TAFC modeling the behavior of the systems regarding the output variable y_4 is shown in Fig. 9.

By iterating the learning process for each output variables, the performance of our neuro-fuzzy approach can be evaluated. In particular, this performance has been assessed according to the following error measure:

$$\begin{aligned} \hat{e}_{total} &= \hat{e}_1 + \hat{e}_2 + \hat{e}_3 + \hat{e}_4 = \\ &= \sum_{n=1}^{1000} (y_1(n) - \hat{y}_1(n))^2 + \\ &+ \sum_{n=1}^{1000} (y_2(n) - \hat{y}_2(n))^2 + \\ &+ \sum_{n=1}^{1000} (y_3(n) - \hat{y}_3(n))^2 + \\ &+ \sum_{n=1}^{1000} (y_4(n) - \hat{y}_4(n))^2 \end{aligned}$$

where $\hat{y}_i$ with $i = 1, \dots, 4$, are the output value computed by the TAFC model. We used Matlab to learn the TAFCs related to the FUZZ-IEEE 2011 Competition and we obtained

the following error measures:

$$\begin{aligned}
\sum_{i=1}^{1000} (y_1(n) - \hat{y}_1(n))^2 &= 0.0645 \\
\sum_{i=1}^{1000} (y_2(n) - \hat{y}_2(n))^2 &= 0.0474 \\
\sum_{i=1}^{1000} (y_3(n) - \hat{y}_3(n))^2 &= 0.5906 \\
\sum_{i=1}^{1000} (y_4(n) - \hat{y}_4(n))^2 &= 0.6512
\end{aligned} \tag{9}$$

and, as a consequence:

$$\hat{e}_{total} = 0.0645 + 0.0474 + 0.5906 + 0.6512 = 1.3536 \tag{10}$$

In order to validate the benefit provided by the proposed approach, a comparison between the total error shown in equation 10 and the error $\hat{e}_{total}$ returned by a conventional ANFIS application is given. Indeed, by using the conventional ANFIS approach, the following errors are measured:

$$\begin{aligned}
\hat{e}_{total} &= 455.2466 + 377.2482 + 315.9917 + 300.0843 = \\
&= 1.4486 \times 10^3
\end{aligned} \tag{11}$$

with a percentage improvement:

$$\Delta_E = \frac{(1.4486 \times 10^3 - 1.3536)}{1.4486 \times 10^3} \times 100 = 99.9066\%$$

The values $\hat{e}_{total}$ and Δ_E strongly depend upon the number of control eras returned by the temporal intersection operator. In turn, the number of control eras is related to M_j with $j = 1, \dots, 8$ i.e., the number of segments used by APCA to approximate the time series C_j related to the variable v_j . In this experiment, we have chosen the M_j values in an empirical way in order to minimize the value of the total error $\hat{e}_{total}$ and try to avoid the overfitting issue. In the future works, a more accurate study about overfitting question will be done by validating the learned TAFCS by means of opportune testing datasets.

V. CONCLUSION

This paper introduced a novel of TSK-based neuro-fuzzy approach and its application to dynamic system modeling. This approach used the ANFIS system together with a time series approximation method, the APCA, in order to identify the time ranges where the system shows a behavior characterized by soft oscillations and, for each range, to compute the most appropriate TSK controller modeling the system behavior in that range. Successively, the time ranges and TSK controllers have been used in order to define the control eras, the control configurations and the control time of a TAFCS, an innovative evolvable fuzzy inference engine. As show in experimental results section, our approach reduces the output error measure and achieves a better performance than a standard application of the ANFIS algorithm when applied to a highly dynamic system. In the future, we will test our neuro-fuzzy system on real-world scenarios in order to further validate the provided benefits.

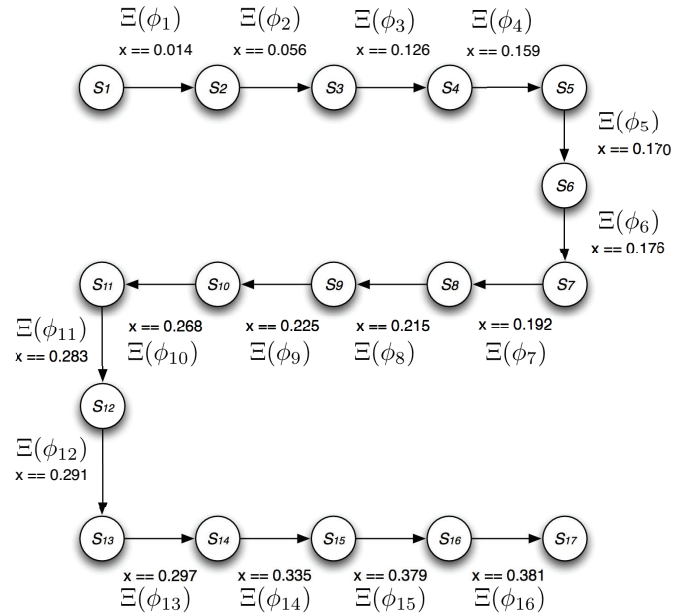


Fig. 9. The TAFCS modeling the FUZZ-IEEE 2011 Competition dataset regarding to the output variable y_4 .

ACKNOWLEDGMENT

The author would like to thank the PhD student, Autilia Vitiello, for her precious support to Matlab programming.

REFERENCES

- [1] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plants," *Proceedings of IEEE*, vol. 121, no. 12, pp. 1585–1588, 1974.
- [2] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [3] Jang, J.-S.R., "ANFIS: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on*, vol.23, no.3, pp.665–685, May/Jun 1993, doi: 10.1109/21.256541.
- [4] G. Feng, "Analysis and Synthesis of Fuzzy Control Systems A Model-Based Approach," CRC Press, 2010.
- [5] S.V. Emelyanov, S.V. Utkin, V.A. Taran, N.E. Kostyleva, A.M. Subladze, V.B. Ezerov, E.N. Dubrovskij, "Theory of Variable Structure Systems", (in Russian), Nauka, Moscow, 1970.
- [6] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions Systems, Man & Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [7] J. Scargle, B. Jackson and J. Norris, "Adaptive Piecewise-constant Modeling of Signals in Multidimensional Spaces," in *Proceedings of PHYSTAT2003*, pp. 157–161, 2003.
- [8] G. Acampora, "Exploiting Timed Automata-based Fuzzy Controllers and data mining to detect computer network intrusions," *Fuzzy Systems (FUZZ)*, 2010 IEEE International Conference on, pp.1–8, July 2010.
- [9] G. Acampora, V. Loia, A. Vitiello, "Hybridizing fuzzy control and timed automata for modeling variable structure fuzzy systems," *Fuzzy Systems (FUZZ)*, 2010 IEEE International Conference on, pp.1–8, July 2010.
- [10] J.S.R. Jang, "Self-learning fuzzy controllers based on temporal back-propagation," *IEEE Trans. Neural Networks*, vol. 3, pp. 714–723, Sept. 1992.