

Intelligent Control of an Inverted Pendulum by Self-Tunable Fuzzy PI-type Controller

Djamel Eddine Chaouch¹, Hichem MAAREF²

¹Faculty of Sciences and Technology, University of Mascara BP 276, Mascara – Algeria

²IBISC Laboratory, University of Evry Val d'Essonne, 40 rue du Pelvoux, 91020 Evry, France
dj_chaouch@yahoo.fr, hichem.maaref@ibisc.univ-evry.fr

Abstract: The work describes an automatically on-line Self-tunable PI-type fuzzy inference system with application of control approach to nonlinear system. In this paper, we describe a neuro-fuzzy controller as a STFIS (Self Tuning Fuzzy Inference System) optimized online using the rules of Takagi-Sugeno type, and a back propagation learning algorithm which is used to minimize a cost function that is made up of a quadratic error term and a weight decay term that prevents an excessive growth of parameters. Then STFIS PI-type controller is synthesized. The application is done on a system of inverted pendulum to follow a predefined reference trajectory in the presences of disturbance. Some simulations with robustness tests are performed, which demonstrates the feasibility of the proposed control strategy. Results of simulations containing tests of robustness are presented and realized in MATLAB environment.

Keywords: Self-tuning fuzzy inference system, Adaptive neuro-fuzzy control, Inverted pendulum.

1. Introduction

The common methods of the automatic that were largely implemented have many problems of industrial regulation. However, the majority of the physical systems have nonlinearities and their parameters are often little known about and/or variable in time. For the control of such classes of systems, the conventional methods of the automatic have shown their limits in terms of stabilization and performances. With the considerable development of the digital computers, the automation experts have adopted more new approaches such as the predictive control, the robust control, as well as the techniques based on the artificial intelligence.

The past few years have witnessed a rapid growth in the use of fuzzy logic controllers for the control of processes, which are complex and badly defined. Most fuzzy controllers developed till now have been of the rule-based type, where the rules in the controller attempt to model the operator's response to particular process situation.

Recently, the resurgence of interest in the field of artificial neural networks has injected a new driving force into the fuzzy literature. The back-propagation learning rule, which drew little attention till its applications to artificial neural networks was discovered, is actually an

universal learning paradigm for any smooth parameterized models, including fuzzy inference systems. As a result, a fuzzy inference system can now not only take linguistic information from human experts, but also adapt itself using numerical data (input/output pairs) to achieve better performance. This gives fuzzy inference systems an edge over neural networks, which cannot take linguistic information directly. In autonomous wheeled robot, many developed learning techniques have arisen in order to generate or to tune fuzzy rules. Most of them are based on the so-called "Neuro-Fuzzy learning algorithms" as proposed by [3], [7], [8]. These methods are well for constructing an optimal fuzzy system model which is used to identify the corresponding practical system. Among these approaches; the "control by Neuro-Fuzzy", this control is currently one of the concerns of the researchers in the world.

The work presented in this article relates to the implementation of this technique of control with neuro-fuzzy and specially STFIS PI-type with the physical system: "inverted pendulum". The main objective is to determine online interpretation and optimization of rule base, with minimal time to adapt to the system.

The present paper is organized in the following manner: Section 2 describes the neuro-fuzzy and proposes self-tuning fuzzy inference system (STFIS) then STFIS PI-type architecture and learning algorithm, section 3 presents the application of STFIS PI-type controller to the inverted pendulum. Finally, conclusions are summarized in Section 4.

2. Neuro-Fuzzy

In this part of work, we present an intelligent and innovative method, combining the two techniques intelligent (fuzzy and neural) are designed to pursuing a path of a nonlinear system and they have proved their efficiency, speed and ease of implementation [4]. The system can be interpreted like a special neural network with vague parameters or like a vague system application in a parallel distributed form [7].

2.1. Self-tuning Fuzzy Inference System STFIS

The formal analogy between a fuzzy inference system and a multilayer neural network associated with optimization algorithms is used from the retro-propagation gradient algorithm have winded up in what is called a STFIS network [1], [8].

2.1.1 Presentation of STFIS

The method provides a complete structural system with fuzzy inference for zero order Takagi-Sugeno [4]. The architecture of a neuro-fuzzy Controller is shown in figure1, this fuzzy inference system (FIS) is a network consisting of four layers [ZEM 06]:

- 1) *Layer 1*: it receives inputs.
- 2) *Layer 2*: calculate the membership degrees of these inputs to their fuzzy subset. The weights of the network between the first layer and this layer correspond to the parameters defining the membership functions.
- 3) *Layer 3*: calculate the values of truth. The weights between two hidden layers define the AND operator chosen. In fact, it is enough to choose an operator called a T-norm, min is most optimistic of the T-norms [11].
- 4) *Layer 4*: The fourth layer is the output layer. The weight w_i of the network between the third and fourth layer correspond to the parties conclusion of the rules.

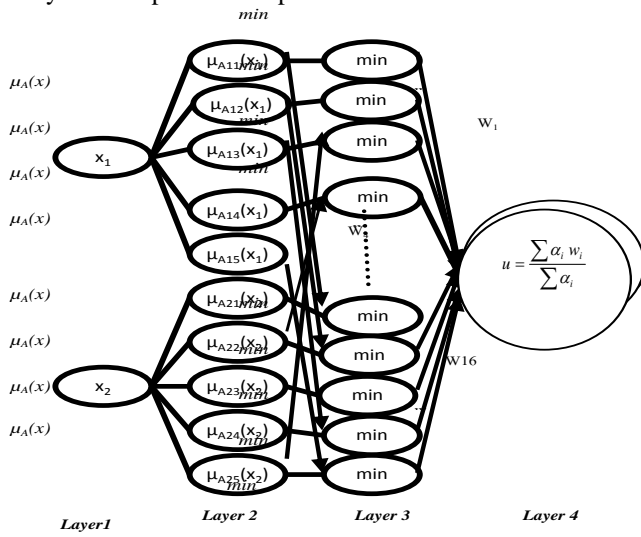


Fig. 1: Self-Tunable Fuzzy Inference System.

2.1.2 Architecture and learning Algorithm

For the control of the nonlinear system, Jordan [2] proposes the distal control method, which is used under the name of JEAN (Jordan method Extended for Adaptive Neural control). This architecture figure.2a needs the used of two STFIS networks:

- 1) *A first network*: to identify the system (Model).
- 2) *A second network*: to control the system (Controller).

Another architecture known as the "mini -JEAN" as illustrated in the figure.2.b. This architecture doesn't require an emulator net-work. It uses only one network as a controller, the learning of which is done directly by the back propagation of the output error [3].

Compared to the architecture JEAN, some equivalent performances are obtained for the mean error in generalization. On the other hand, the computing time favors clearly mini-JEAN. Optimization of adjustable parameter is accomplished with a version of the classic gradient retro-propagation algorithm adapted to net structure of fig. 2.

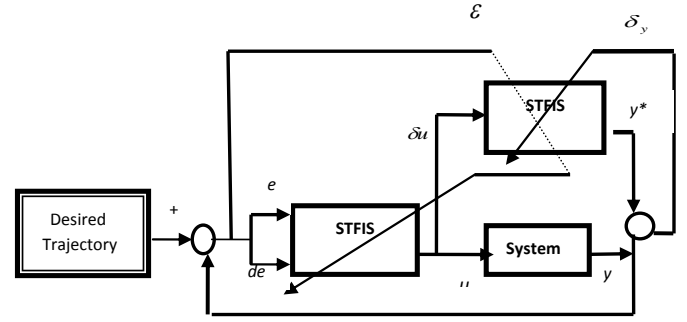


Fig. 2.a: JEAN learning architecture.

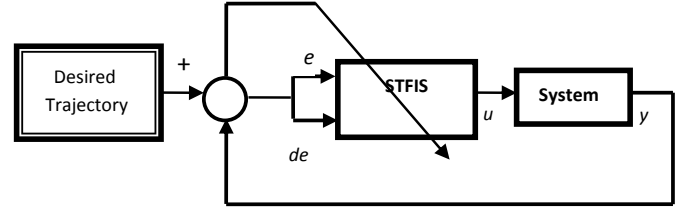


Fig. 2.b: Control architecture mini-JEAN.

In our type of controller we use back propagation gradient algorithm to adjust the weights of the last layer of the network. The general principle of this algorithm is as follows: at each iteration, we modify the weights of output layer; this change will make the weight in the opposite direction to the gradient of the cost function. The process is repeated until the weight of the output layer has converged, that is to say that the gap between the network output and the desired output becomes acceptable.

The aim is to minimize cost function E:

$$E = \frac{1}{2} \varepsilon^2 \quad (1)$$

Where ε is the difference between set point process outputs. The basic equations of the algorithm are:

$$w_{ij}^n(t) = w_{ij}^n(t-1) + \Delta w_{ij}^n(t) \quad (2)$$

$$\Delta w_{ij}^n(t) = -\eta \delta_i^n \alpha_j^{n-1} + b \Delta w_{ij}^n(t-1) \quad (3)$$

Where

$w_{ij}^n(t)$: i^{th} parameter between i of layer n and j^{st} unit of layer $n-1$.

η : learning gain.

t : training iteration.

b : moment parameter .

δ_i^n : error term(i^{th} neuron of layer n).

α_j^{n-1} : output of j^{st} unit of layer $n-1$.

The quality of solution obtained using this algorithm depends on input learning signals, algorithm control parameters and learning duration (number of iterations).

2.1.3 Algorithm of Modification Weights Regression

The procedure is entirely done on-line on the engine. The table of rules (weights w_i) can be initially empty or filled with an a priori knowledge. The engine acquires by its systems output measures, calculates the error to back-propagated, updates the triggered rules on-line. The weights of the table of decision are then adjusted locally and progressively. The cost function is given by [8]:

$$J = E + \lambda \sum w_i^2 \quad (4)$$

Where E is the classic quadratic error, w_i are the parameters (weights) to optimize parameters and λ is a constant that controls the growth parameters. The second term in J is known as weight decay and used usually in the context of classification problems. This technique has been analyzed in the framework of leaning theory and it was shown that is a very simple manner to implement a regularization method in a neural network in order to optimize the compromise between the learning error and the generalization error. Thanks to the classic back-propagation algorithm, the parameters are modifying as:

$$w(t+1) = w(t) + \eta \left(\frac{-\partial J}{\partial w} \right) \quad (5)$$

This algorithm easily includes the effect of the second term of the cost function J and by taking $\beta = 2\lambda\eta$ (regression coefficient) we obtain:

$$w(t+1) = w(t) + \eta \left(\frac{-\partial E}{\partial w} \right) - \beta w(t) \quad (6)$$

Since a fuzzy inference system is concerned, we adapt this formula by multiplying β by the firing term of the rule, namely $\alpha_i / \sum \alpha_i$, α_i is the truth value of the premise part of the triggered rule.

If we limit the optimization only on the conclusions parameters w_{1j}^4 then we get:

$$\Delta w_{1j}^4(t) = -\eta \delta_1^4 \alpha_j^3 + b \Delta w_{1j}^4(t-1) - \alpha_j^3 \beta w_{1j}^4(t-1) / \sum \alpha_j^3 \quad (7)$$

With

$$\delta_1^4 = y_1 - y / \sum \alpha_j^3 \quad (8)$$

Where:

y_1 : effective output value.

y : desired output.

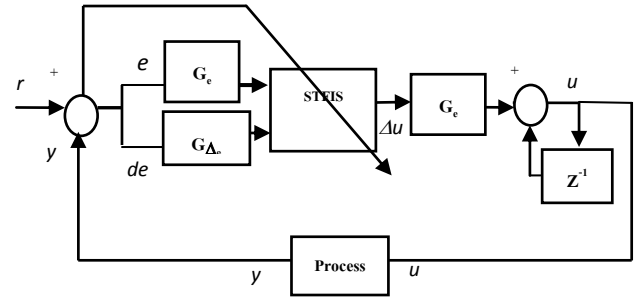


Fig. 3: Self-Tunable Pi-type Fuzzy Inference System control architecture.

3. The proposed controller

The block diagram of the STFIS PI-type controller is shown in fig. 3. The output of the controller is modified by a self-tuning mechanism which is shown in the last sections.

All MFs for controller inputs, ie e and Δe are defined on the common normalized domain [-1, 1]. Example of fuzzification in our case we chose the membership functions of Gaussian type and sigmoid. The MFs for both normalized inputs (e_N and Δe_N) and output (Δu_N) of the controller have defined on the common normalized domain [-1, 1]. The relationships between the SFs (G_e , $G_{\Delta e}$ and $G_{\Delta u}$) and the input and output variables of STFIS PI-type controller are as follows:

$$e_N = G_e \cdot e \quad (9)$$

$$\Delta e_N = G_{\Delta e} \cdot \Delta e \quad (10)$$

$$\Delta u_N = G_{\Delta u} \cdot \Delta u \quad (11)$$

The operation of PI-type STFIS can be described by:

$$u(k) = u(k-1) + \Delta u(k) \quad (12)$$

In Eq. (12) k is the sampling instance and Δu is the incremental change in controller output which is determined by the Self tuning Fuzzy Inference System learning Algorithm.

4. Simulation results

Use In this section we apply the theoretical results presented in the preceding section to an inverted pendulum stabilization problem. The stabilization problem is to design a controller to keep the pendulum in its unstable equilibrium point in the presence of disturbances. Because of its dynamically rich structure, inverted pendulum system is widely used in the literature to check validity of control strategies [5].

The dynamic equation of inverted pendulum can be given [6]:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{g \sin x_1 - m l x_2^2 \cos x_1 \sin x_1 / (m_c + m)}{l(4/3 - m \cos^2 x_1 / (m_c + m))} \\ + \frac{\cos x_1 / (m_c + m)}{l(4/3 - m \cos^2 x_1 / (m_c + m))} u(t) + d(t) \end{cases} \quad (13)$$

Where x_1 is angular position and x_2 is the velocity of the pole respectively, $g = 9.8 \text{ m/s}^2$, m_c is the mass of cart and $m_c = 1\text{kg}$, m is the mass of pole and $m = 0.1\text{kg}$, here $l = 0.5\text{m}$ is half lengthen of pole, u is control input, $d(t)$ is external disturbance[9]. Here we take $d(t) = 20\sin(2\pi)$, position tracking reference input was $r(t) = 0.2\sin(\pi + \pi/2)$. This inverted pendulum dynamic model is constructed using MATLAB Simulink software. The initial state of single inverted pendulum is $[\pi/60, 0]$.

A STFIS PI-type controller based on an on-line optimization of a zero order Takagi-Sugeno inference system is successfully applied. It is used to minimize the cost function that is made up of a quadratic error term.

Example of fuzzification in our case we chose the membership functions of Gaussian type and sigmoid. The Memberships functions are normalized and shared in five subsets for all displacement (fig.4). The linguistic labels are defined follows: NB: Negative Big, NS: Negative Small, Z approximately Zero, PS: positive Small and PB: Positive Big.

For the STFIS network parameters, there were five memberships functions of type sigmoid and Gaussians (fig.3) and $\beta = 0.00001$, $\eta = 0.7$ and $b = 0.3$.

The fig.5 shows the STFIS PI-type controller response of the inverted pendulum for the given desired point angle trajectory and fig. 6 shows the Control input

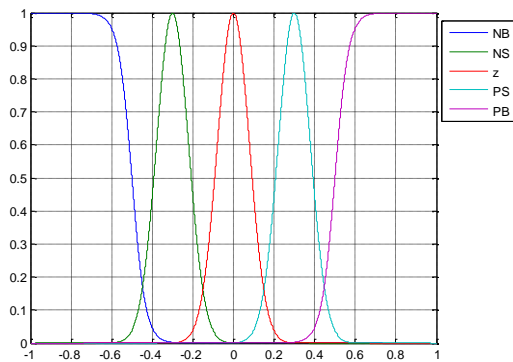


Fig. 4: Memberships function.

for inverted pendulum. It is found that actual trajectories of the inverted pendulum are almost merged with the desired trajectory, with the presence of the perturbations.

Form this inference, it is concluded that STFIS PI-type training is completely satisfied and inverted pendulum tracking error is almost nearly zero fig.7.

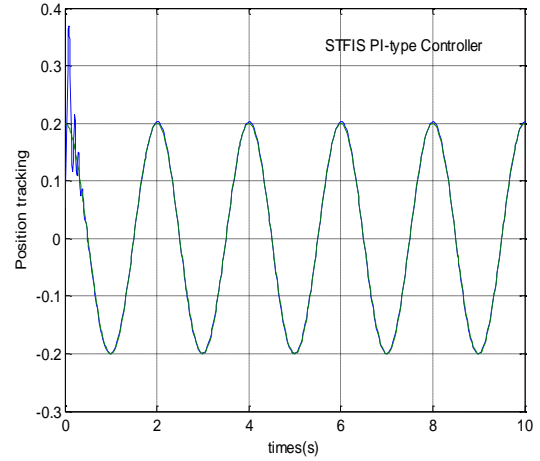


Fig. 5: Trajectory designed and Trajectory tracking of inverted pendulum with STFIS controller for angle.

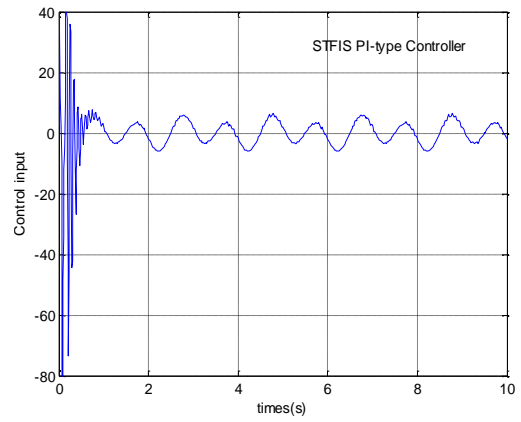


Fig. 6: Control input for inverted pendulum.

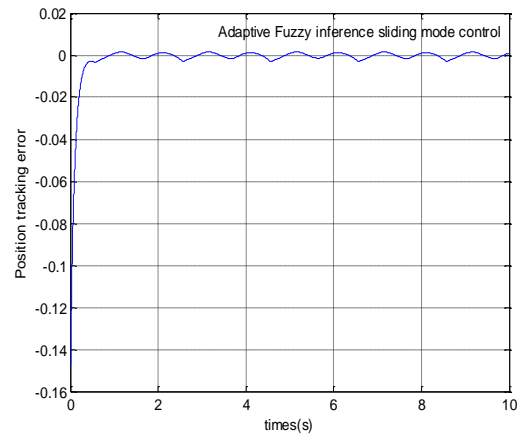


Fig. 7: Position tracking error for inverted pendulum.

Fig.8 represents the convergence of the parameters (weights) of the last layer.

The results of the simulation are reported in the table 1. The optimization phase tends to stable weights (Fig. 8). In these circumstances the outputs linguistic labels could be interpreted as follows: NB: [-20, -5], NS: [-4, -0.002], Z: [-0.0001, 0.0001], B: [-0.002, 4] and PB: [5, 20].

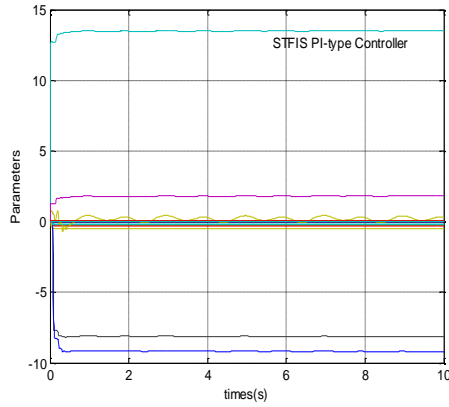


Fig. 8: The parameters (weights).

The table 2 illustrates the linguistic translation of the table obtained by on-line optimization for the inverted pendulum control (table 1).

By comparing the table 2 proposed by learning and the table 3 of Mac Vicard-Whelan [10], we can observe that the two sets of linguistic rules are quite close. The cases (noted with *) are different and they differ from only one linguistic concept (NB instead NS and PB instead PP...).

e de	NB	NS	Z	PS	PB
NB	-0.026	-0.250	-0.283	-0.194	0
NS	-0.473	-8.126	-9.194	0	-0.006
Z	-0.166	0	0.290	0	-0.006
PS	-0.002	0	13.5286	1.8343	-0.002
PB	0	0.0712	0.0712	0.0711	0.0001

Table 1: Evolution of weights.

e de	NB	NS	Z	PS	PS
NB	NS*	NS*	NS	NS	Z
NS	NS*	NB	NB	Z	NS*
Z	NS	Z*	PS*	Z*	NS*
PS	NS	Z	PB*	PB*	NS*
PB	Z	PS	PS	PS*	Z*

Table 2: Terms linguistic of optimized weights.

So, we can claim that the extracted rules are quite logical and coherent. On the other hand, the main advantage of the described technique is the optimization of the controller with respect to the actual characteristics of the engine. The use of a function cost gathering a quadratic error and a term of regression of the weights enabled us to achieve our goal.

e de	NB	NS	Z	PS	PS
NB	NB	NB	NS	NS	Z
NS	NB	NS	NS	Z	PS
Z	NS	NS	Z	PS	PS
PS	NS	Z	PS	PS	PB
PB	Z	PS	PS	PB	PB

Table 3: Expertise Terms linguistic.

5. Conclusion

An on-line Self-tunable PI-type fuzzy inference system is presented to realize the stabilizing tracking control problem for inverted pendulum system. We have presented and implemented an optimization technique allowing an on-line adjustment of the fuzzy controller parameters. The descent gradient algorithms, with its capacities to adapt to unknown situations by the means of its faculties of optimization, and the fuzzy logic, with its capacities of empirical knowledge modeling, are combined to control. Indeed, we have obtained an on-line optimized Takagi-Sugeno type FIS of zero order (STFIS). The STFIS which not require the good knowledge of the model is simple, economical and safe since it is done on control. It leads to very quick and efficient optimization technique. A comparison between the STFIS set rules and that deduced by human expertise, shows the validity of the proposed technique.

The feasibility of STFIS PI-type controller for inverted pendulum has been proved and illustrated by simulation. The STFIS controller presented very interesting tracking features and was able to respond to different dynamic conditions.

References

- [1] Ajith Abraham "Neuro Fuzzy Systems: State-of-the-art Modeling Techniques", School of Computing & Information Technology, Monash University, Australia, 2001.
- [2] M. I. Jordan, D. Rumelhart "Internal word models and supervised learning " *Proceeding of the 8th International Workshop on Machine learning*, pp 70-74, 1991.
- [3] H. Maaref, C. Barret, "Progressive Optimization of a Fuzzy Inference System", *IFSA-NAFIPS'2001*, Vancouver, pp.665-679, 2001.
- [4] D. Nauck, R. Kruse, "What are Neuro-Fuzzy Classifiers?" *Seventh International Fuzzy Systems Association World Congress IFSA '97*, Vol. IV, pp. 228-233, Academie de prague, 1997.
- [5] A. Yazici, A. Karamancioglu, R. N. Gasimov "Nonlinear Programming Based Sliding Mode Control of an Inverted Pendulum " Osmangazi University, Meselik, Eskisehir TURKEY, 2006.
- [6] W. Wang " Adaptive Fuzzy Sliding Mode Control for Inverted Pendulum" *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCST '09)* Huang-

shan, P. R. China, 26-, pp. 231-234, Decemder 2009.

- [7] K. Zemalache, H. Maaref “Intelligent control for a drone by self-tunable fuzzy inference system,” *6th international multi-conference on systems, signals and devises, Djerba, Tunisia*, 23-26 March 2009.
- [8] K. Zemalache, L. Beji, H. Maaref “Control of Drone :Study and Analysis of the Robostness” *Journal of Automation, Mobile Robotics and Intel-ligent Sysyems* ,2008, pp 33-42.
- [9] JIA Nuo, WANG Hui. “Nonlinear Control of an In-verted Pendulum System based on Slinding mode method”, *ACTA Analysis Funcionalis applicata*, 2008, 9(3): pp.234-237.
- [10] P. J. MAC Vicar-Whelan, “ Fuzzy sets for man machine interactions”, *International Journal of Man-Machine Studies*, 1976, vol. 8, pp 687-697.
- [11] T. Takagi, M. Sugeno “Fuzzy identification of sys-tems and Its applications to modeling and control”, *IEEE Transactions on systems Man and Cybernet-ics*, vol 15, no1, pp 116-132, 1985.