An Auto-Tuning Method for the Scaling Factors of Fuzzy Logic Controllers with Application to SISO Mechanical System

Gamal Abdel Nasser, Abdel Badie Sharkawy, and M-Emad S. Soliman

***Abstract*—In this paper, a PD-like self-tuning fuzzy controller based on tuning of scaling factors (STFC) by gradient descent method is presented. The tuning scheme allows the tuning of the scaling factors to be on-line. Tuning scaling factors is more effective and simpler than tuning all the parameters of standard fuzzy logic controller (FLC). The aim is to obtain good performance parameters, such as the rise time, the overshoot, the steady-state error. Experimental results of an inverted pendulum system with STFC controller show a better performance in the transient and steady state phases than other classical controllers like PD, PID, auto-tuned PID controller (PID-AT), and linear quadratic regulator (LQR).**

***Index Terms*—Fuzzy logic controller (FLC), scaling factors, gradient decent method, performance indices.**

I. Introduction

For enhancing control systems, there are two important information sources: sensors, which provide measurements of variables, and human experts who give linguistic instructions and descriptions about the system. Fuzzy logic controller (FLC) was created to combine these two different types of information by handling information coming from human operators. The main advantage of the FLC is that it can be applied to plants that are difficult to model mathematically [1]. Self-tuning of a FLC aims to adapt the controller to different operating conditions [2]. For successful design of a FLC, proper selection of input and output scaling factors and/or the tuning of other controller parameters, such as the representation and construction of the rule base or the determination of the position and shape of the membership functions, are conclusive jobs [2]. Basically, there are two different tuning approaches to achieve optimal parameters for a FLC: on-line and off-line tuning [2]. Off-line tuning scaling factor using genetic algorithm optimization method can be found in [3]. On-line tuning membership function using gradient descent optimization method is discussed in [4]-[6]. On-line tuning scaling factor using fuzzy tuner is demonstrated in [7]-[9]. The authors of [10] have used Neuro-Fuzzy tuner and in [1], [11], [12] used the gradient descent optimization method.

In this work, we introduce an auto-tuning mechanism for the scaling factors of a PD-type FLC. The gradient descent method is employed to optimally determine them on-line. This algorithm along with other classical controllers is tested experimentally using an inverted pendulum mounted on a cart. The inverted pendulum is a highly nonlinear and open-loop unstable system [13], [14]. Inverted pendulum system is often used as a benchmark for verifying the performance and effectiveness of control algorithms because of the simplicity of its structure [15]. Results show the effectiveness of the proposed control system.

The paper is organized as follows. Section II introduces the mathematical model of the inverted pendulum. Section III gives a brief description about classical controllers which are experimentally tested in this work. In Section IV, we derive the auto-tuning algorithm for the scaling factors of the FLC. Section V describes the experimental setup. Section VI discusses the experimental results and Section VII offers our concluding remarks.

II. Inverted Pendulum Model

This Section provides description of the inverted pendulum used in this study. Fig. 1 shows the free body diagram. Using Newton's second law, it can be easily shown that the dynamic equations of motion are as follow:

$$(M+m)\frac{d^2x}{dt^2}+\beta\frac{dx}{dt}=F_V+ml\sin\theta\left(\frac{d\theta}{dt}\right)^2-ml\cos\theta\frac{d^2\theta}{dt^2} \quad (1)$$

$$(I+ml^2)\frac{d^2\theta}{dt^2}=mgl\sin\theta-ml\qquad\theta\frac{d^2x}{dt^2} \quad (2)$$

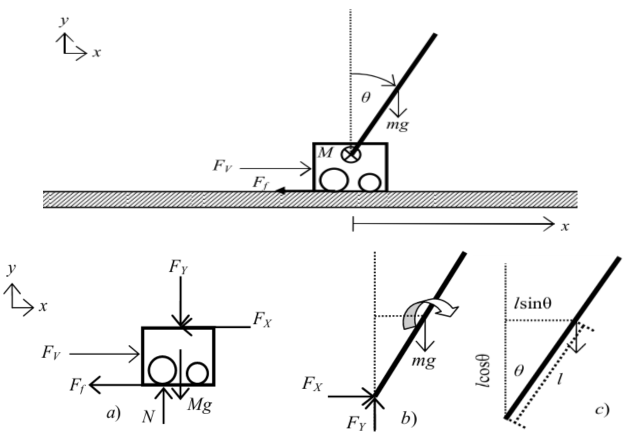

Fig. 1. Free body diagrams of (a) the cart and (b) the pendulum. (c) determining the required distances.

The symbols used in (1) and (2) and their numerical values are defined (see Table I).

Manuscript received January 4, 2014; revised July 23, 2014.
The authors are with the Mechanical Engineering Department, Assiut University, 71516 Assiut, Egypt (e-mail: gamalengineer1@yahoo.com).

TABLE I: PARAMETERS OF THE INVERTED PENDULUM

Symbol	Parameter	Value	Unit
M	Mass of the cart	0.5	kg
m	Mass of the pendulum	0.2	kg
β	Viscous damping coefficient	0.1	N/m/s
$2l$	Length of the pendulum	0.3	m
I	Mass moment of inertia of the pendulum	0.0015	$\mathrm{kg.m^2}$
g	Acceleration of gravity	9.8	$\mathrm{m/s^2}$
F_V	Input force	-	N

The linearized model equations are

$$(M+m)\frac{d^2x}{dt^2}+\beta\frac{dx}{dt}=F_V+ml\theta\left(\frac{d\theta}{dt}\right)^2-ml\frac{d^2\theta}{dt^2} \quad (3)$$

$$(I+ml^2)\frac{d^2\theta}{dt^2}=mgl\theta-ml\frac{d^2x}{dt^2} \quad (4)$$

We used equations (3) and (4) for solving the Riccati equation in LQR controller to get the optimal state feedback gains.

III. CLASSICAL CONTROL ALGORITHMS

A. LQR (Liner Quadratic Regulator)

The LQR method is a robust technique for designing controllers for complex systems that have strict performance requirements aiming at finding the optimal controller [16]. The aim of the stabilizing controller is to balance the inverted pendulum around vertical position. The state space model has been determined using the linearized model (3) and (4) with system parameter values as given in Table I. Then the Riccati equations are solved and a feedback gain is determined off-line. This gain will lead to optimal results evaluated from the defined performance index [3], [5], [6]. LQR method followed in this work is based on the following state-space model for

$$x_1=\theta\ ,x_2=\dot{\theta}\ ,\ x_3=x\ ,x_4=\dot{x}$$

$$\dot{x_1}=x_2$$

$$\dot{x_2}=62.43x_1+0.91x_4-9.09u$$

$$\dot{x_3}=x_4$$

$$\dot{x_4}=-2.67x_1-0.18x_4+1.8u \quad (5)$$

where θ is pendulum angle measured from vertical reference, $\dot{\theta}$ is rotational speed of pendulum, x is cart position, $\dot{x}$ is cart speed, and u is input force. By solving the Riccati equation the optimal gain vector is $k=[-83,-8,-32,-22]^T$.

B. The PID Controller Using (Ziegler – Nichols) Tuning Rule

Here, a PID controller for the angle of the inverted pendulum is to be designed, not considering the pivot position control. The control structure of the inverted pendulum with PID controller is given in Fig. 2. The goal of the control design is to stabilize the angle of the inverted pendulum with PID controller [17]-[19]. The parameters of PID controller of the inverted pendulum are estimated by using Ziegler – Nichols tuning rule, first set $T_i=\infty$ and $T_d=0$. Using the proportional control action only, increase the proportional gain k_p from 0 to a critical value K where the output first exhibits sustained oscillations. The critical gain K_{cr} and the corresponding period T_{cr} are experimentally determined. The critical gain was found to be $K_{cr}=110$ and the corresponding period $T_{cr}=0.23sec$. Ziegler and Nichols suggested that we set the values of the parameters K_p, T_i and T_d according to the formula (see Table II) [20].

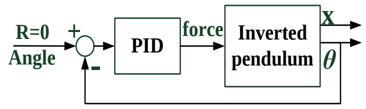

Fig. 2. block diagram of a PID controller.

TABLE II: ZIEGLER–NICHOLS TUNING RULE PARAMETERS

Controller	k_p	T_i	T_d
PID	$0.6k_{cr}$	$0.5p_{cr}$	$0.125p_{cr}$

C. PD Controller

We have made fine tuning on k_p and T_d in order to determine them experimentally. The target of the tuning operation is to get the shortest rise time with suitable overshoot keeping stability of the pendulum in vertical position. The following values have been obtained $k_p=100$, $T_d=0.035$.

D. PID Controller with Fuzzy Self-Tuning of a Single Parameter Optimization (PID-AT)

Because of its simplicity, this strategy has been widely considered [21]. This strategy consists of parameterizing the Ziegler–Nichols formula by means of a single parameter α, then using an online fuzzy tuning to self-tune this single parameter. In this strategy, the three PID parameters can be expressed as

$$k_p=1.2\alpha(t)k_{cr} \quad (6)$$

$$T_i=75\frac{1}{1+\alpha(t)}t_{cr} \quad (7)$$

$$T_d=25T_i \quad (8)$$

where k_{cr} and t_{cr} are the ultimate gain and ultimate period, respectively. The value of $\alpha(t)$ is determined by the following equation:

$$\alpha(t+1)=\begin{cases}\alpha(t)+\gamma h(t)\big(1-\alpha(t)\big) & for\ \alpha(t)>0.5\\ \alpha(t)+\gamma h(t)\alpha(t) & for\ \alpha(t)\le 0.5\end{cases} \quad (9)$$

where $h(t)$ is the output of the fuzzy tuning system and γ is a constant that has to be chosen in the range [0.2 – 0.6]. The fuzzy system has three membership functions for each of the two inputs (e and $\dot{e}$) and three membership functions for the output. The rule-base consists of 9 rules. The initial value of $\alpha(t)$ is set equal to 0.5, which corresponds to the Ziegler–Nichols formula. With respect to the strategy, the tuning of the scaling coefficient of the fuzzy module and of the parameter γ is left to the user [21]. The control structure of inverted pendulum with PID-AT controller with Fuzzy self-tuning of a single parameter optimization is given in Fig. 3.

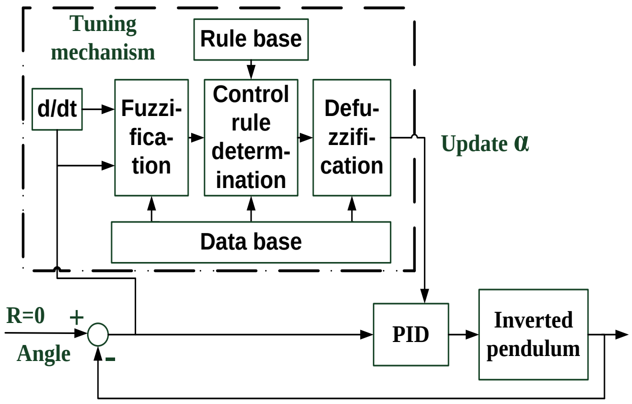

Fig .3. Block diagram of PID controller with fuzzy self-tuning of a single parameter.

IV. Fuzzy Controller with Self Tuning Scaling Factors

The simulation results in [22] show that the FLC with self-tuning scaling factors improves the performance than adjusting the membership function (MF). The scaling factors (SF) are the main parameters used for tuning the FLC [1]. We can find that any change in the SFs results in adjustment of the poles and zeros of the overall transfer function. This is the reason that changes in the SFs have a dramatic effect on the overall dynamics of the closed loop system [7]. We use the PD-like FLC; because it makes quick response with less oscillation than the PI- like FLC [1]. There have been considerable developments in the tuning of parameters in FLC systems using the gradient-descent-based back-propagation (BP) algorithm, like methods in neural networks. The STFC is a four-layer feed forward network; it applies the gradient-descent-based BP algorithm to adjust the SFs. The goal is to minimize a cost function. Table III shows the rule base of a FLC in which N, Z and P denote negative, zero and positive respectively. Fig. 4 shows output membership functions (singletons) of FLC, Fig. 5 shows input membership function of FLC, and Fig. 6 shows the control structure of inverted pendulum with STFC controller.

Here, the back-propagation (BP) algorithm is used to get the updating laws of the scaling factors S_1, S_2 and S_3. The goal is to minimize a cost function E, so that training pattern k is proportional to the square of the difference between the set point (sp) and the plant output $y(k)$ (angle error) and the square of the angle error change [20].

TABLE III: Rule Base of the FLC

Error ,e	Change in error ,Δe: N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

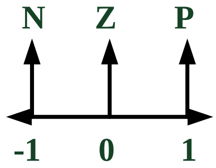

Fig. 4. Singletons of the output membership functions.

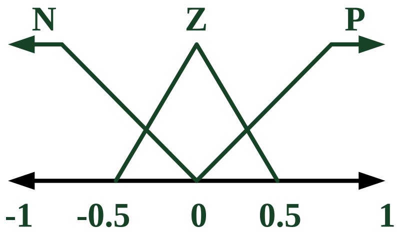

Fig. 5. Input membership functions of the FLC.

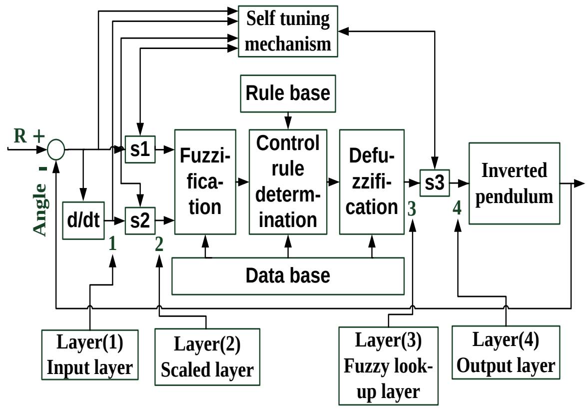

Fig. 6. Block diagram of fuzzy logic controller with self tuning scaling factors.

Let E be defined by

$$E = e^2 + \mu \Delta e^2 \qquad (10)$$

$$e = S_P - Y(K) \qquad (11)$$

where

E Cost function

μ Error change weight

s_p Set point

$Y(k)$ System output

e Error

Learning rule is [1]:

$$S_i(k+1) = S_i(k) - \eta_i \frac{\partial E}{\partial S_i} + \alpha_i \Delta S_i(k) \qquad (12)$$

where

S_i adjusted Scaling factor (SF) i =1, 2, 3

η_i the learning rate

α_i the momentum parameter

$$\alpha_i, 0 < \alpha_i < 1$$

$$\Delta S_i(k) = S_i(k) - S_i(k-1) \quad (13)$$

The learning law for each layer in the feedback direction is as follow;

Layer 4:

The gradient of E in (10) with respect to an arbitrary weighting w

$$\frac{\partial E}{\partial w} = 2\, e\,(k)\, \frac{\partial e(k)}{\partial w} + 2\,\mu\,\Delta\, e\,(k)\, \frac{\partial \Delta e(k)}{\partial w} \quad (14)$$

For set point = 0

$$\frac{\partial E}{\partial w} = -2\, e\,(k)\, \frac{\partial Y(k)}{\partial w} - 2\,\mu\,\Delta\, e\,(k)\, \frac{\partial \Delta Y(k)}{\partial w} \quad (15)$$

$$\frac{\partial E}{\partial w} = -2e\,(k)\, \frac{\partial Y(k)}{\partial U(k)}\, \frac{\partial U(k)}{\partial W(k)} - 2\,\mu\,\Delta\, e\,(k)\, \frac{\partial \Delta Y(k)}{\partial U(k)}\, \frac{\partial U(k)}{\partial w} \quad (16)$$

where $U(k)$ is the output of STFC after S_3 tuning, and $\frac{\partial Y(k)}{\partial U(k)}$ is the plant sensitivity.

From (16) we can derive the propagation error term given by the output node

$$\delta^n = -\frac{\partial E}{\partial net^n} \quad \text{where n is the number of layers}$$

$$\delta^4 = -\frac{\partial E}{\partial net^4} = 2\, e\,(k)\, \frac{\partial Y(k)}{\partial net^4} + 2\,\mu\,\Delta\, e\,(k)\, \frac{\partial \Delta Y(k)}{\partial net^4}$$

$$\delta^4 = 2\, e\,(k) \frac{\partial Y(k)}{\partial U(k)} \frac{\partial U(k)}{\partial net^4} + 2\,\mu\,\Delta\, e\,(k)\, \frac{\partial \Delta Y(k)}{\partial U(k)}\, \frac{\partial U(k)}{\partial net^4}$$

$$\delta^4 = 2\, e\,(k) \frac{\partial Y(k)}{\partial U(k)} + 2\,\mu\,\Delta\, e\,(k) \frac{\partial \Delta Y(k)}{\partial U(k)}$$

Then, we have

$$-\frac{\partial E}{\partial S_3} = -\frac{\partial E}{\partial Y^4}\, \frac{\partial Y^4}{\partial net^4}\, \frac{\partial net^4}{\partial S_3} \quad (17)$$

$$\frac{\partial net^4}{\partial S_3} = Y^4 = U^* - \frac{\partial E}{\partial S_3} = \delta^4 Y^3 = \delta^4\, U^* \quad (18)$$

Hence, by (12) the scaling factor S_3 is updated by

$$S_3(k+1) = S_3(k) + \eta_3 \delta^4(k)\, Y^3(k) + \alpha_3\, \Delta\, S_3(k) \quad (19)$$

Layer 3: the error term δ^3 is derived as follows

$$\delta^3 = -\frac{\partial E}{\partial net^3} = -\frac{\partial E}{\partial Y^3}\, \frac{\partial Y^3}{\partial net^3} = -\frac{\partial E}{\partial Y^3} \quad (20)$$

$$\delta^3 = -\frac{\partial E}{\partial Y^4} \frac{\partial Y^4}{\partial Y^3}$$

$$\delta^3 = \delta^4\, S_3 \quad (21)$$

Layer 2: the error term is computed

$$\delta_j^2 = -\frac{\partial E}{\partial net_j^2} = -\frac{\partial E}{\partial Y_j^2}\, \frac{\partial Y_j^2}{\partial net_j^2} \quad (22)$$

$$\delta_j^2 = -\frac{\partial E}{\partial Y^3} \frac{\partial Y^3}{\partial Y_j^2} = \delta^3 \frac{\partial Y^3}{\partial Y_j^2}$$

$$\Delta_1^2 = \delta^3 \frac{\partial U^*}{\partial e^*} \quad for\ j = 1$$

$$\delta_2^2 = \delta^3 \frac{\partial U^*}{\partial \Delta e^*} \quad for\ j = 2 \quad (23)$$

It can be shown that:

$$-\frac{\partial E}{\partial s_j} = -\frac{\partial E}{\partial Y_j^2} \frac{\partial Y_j^2}{\partial s_j} = \delta_j^2 \frac{\partial Y_j^2}{\partial s_j} = \delta_j^2 x_j^1$$

for $j = 1,2$

$$-\frac{\partial E}{\partial s_2} = \delta_2^2\, \Delta e\,(k) \quad for\ -1 < Y_2^2 < 1 \quad (24)$$

$$-\frac{\partial E}{\partial s_2} = 0 \quad for\ Y_2^2 = 1\ or\ Y_2^2 = -1$$

$$-\frac{\partial E}{\partial s_1} = \delta_1^2\, e\,(k) \quad for\ -1 < Y_1^2 < 1 \quad (25)$$

$$-\frac{\partial E}{\partial s_1} = 0 \quad for\ Y_1^2 = 1\ or\ Y_1^2 = -1$$

Therefore, the update rules for S_1, S_2 are

$$S_1(k+1) = S_1(k) + \eta_1 \delta_1^2(k)\, e(k) + \alpha_1\, \Delta\, S_1(k) \quad (26)$$

$$S_2(k+1) = S_2(k) + \eta_2 \delta_2^2(k)\, \Delta e(k) + \alpha_2\, \Delta\, S_2(k) \quad (27)$$

V. System Description

In this research, LabVIEW programming language was used for building the code for the different examined controllers. A picture for the experimental setup is given in Fig. 7. A PC with labview programming language is utilized to build STFC, PID-AT, LQR, PID, PD controllers. The angle of the pendulum is measured by one turn angular potentiometer and cart position is measured by 10 turn angular potentiometer. An interface card is used to communicate with the running variables to-and-from the inverted pendulum using labview program. The specifications of this system are described (see Table IV).

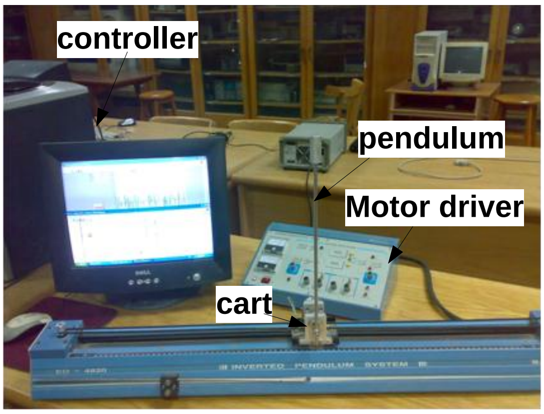

Fig. 7. Experimental setup.

TABLE IV: SPECIFICATIONS OF THE SYSTEM

Elements	Descriptions
PCI-9112	16-CH 12-Bit 110 KS/s Multi-Function DAQ Card
Angle pot.	One turn pot. 10Kω
Position pot.	10 turn pot. 10Kω
DC motor	220 RPM with gear box 24V-1A
Operating system& program	Windows XP and LabVIEW 8.6

VI. EXPERIMENTAL RESULTS

In this Section, the experimental investigation is presented. All the examined controllers have been initiated with initial angle error = 0.23 rad. Experimental results of the LQR and PID controller are illustrated in Fig. 8-Fig. 11. Figure 8 shows cart force, and Fig. 9 shows the error evolution of pendulum using LQR. As it can be noticed, LQR exhibits better performance than PID controller results which are shown in Fig. 10 and Fig. 11 regarding settling time and the overshoot. Both of them perform better results than PD controller which are given in Fig. 12 (cart force) and Fig. 13 (error angle of pendulum). However, the PID-AT controller settling time and rise time are better than LQR controller, PID controller, and PD controller, as shown in Fig.14 (cart force) and Fig. 15 (error angle of pendulum). Results also show that the maximum overshoot in LQR controller is lower than PID-AT controller as given in Table 5. On the other hand, the proposed STFC controller exhibits the best results as shown in Fig. 16 (cart position), Fig. 17 (cart force) and Fig. 18 (error angle of pendulum). The time history of the scaling factors S_1, S_2, S_3 of the STFC controller are illustrated in Fig. 19, Fig. 20, and Fig. 21. Also in force curves of these controllers, LQR is the lowest cart force fluctuation as shown in Fig.8. But the STFC is the lowest cart force overshoot as shown in Fig. 17. Over fluctuations have negative effects on the mechanical structure.

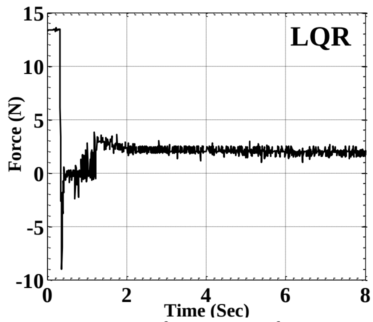

Fig. 8. Cart force curve of LQR.

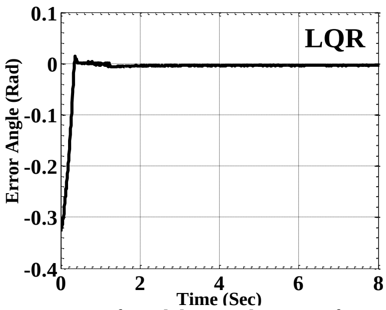

Fig. 9. Error of pendulum angle curve of LQR.

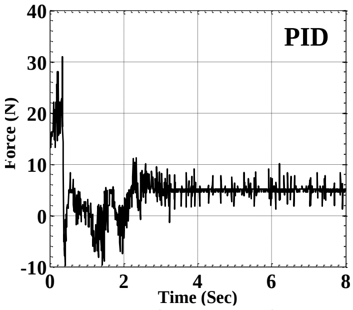

Fig. 10. Cart force curve of PID.

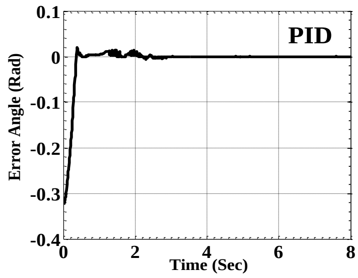

Fig. 11. Error of pendulum angle curve of PID.

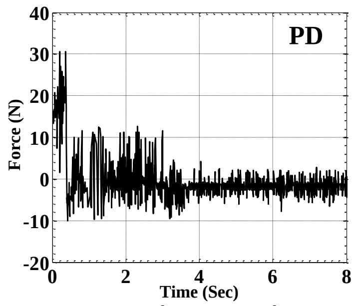

Fig. 12. Cart force curve of PD.

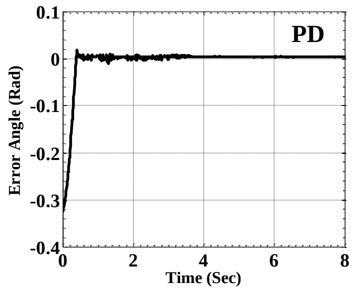

Fig. 13. Error of pendulum angle curve of PD.

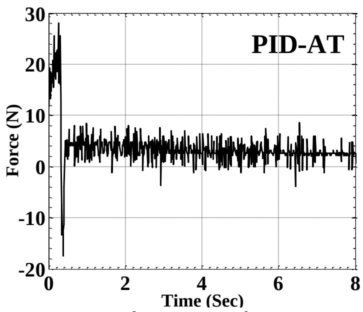

Fig. 14. Cart force curve of tuning PID.

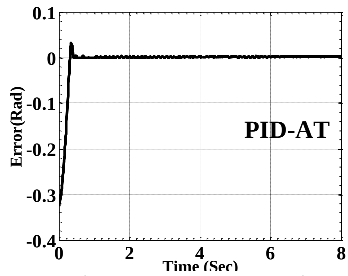

Fig. 15. Error of pendulum angle curve of tuning PID.

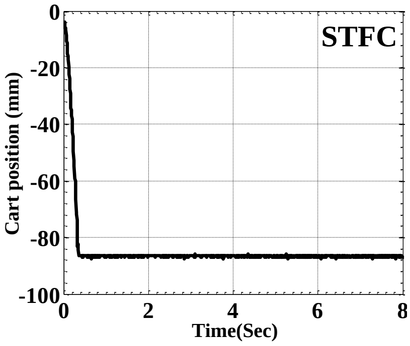

Fig. 16. Cart position curve of scaling factor tuning.

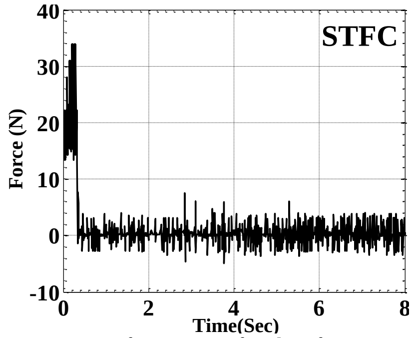

Fig. 17. Cart force curve of scaling factor tuning.

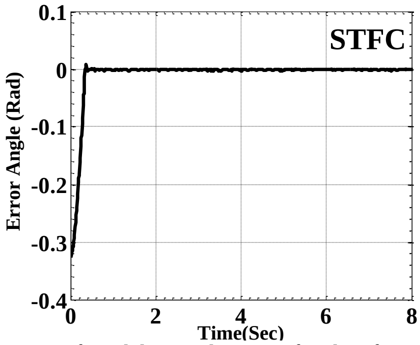

Fig. 18. Error of pendulum angle curve of scaling factor tuning.

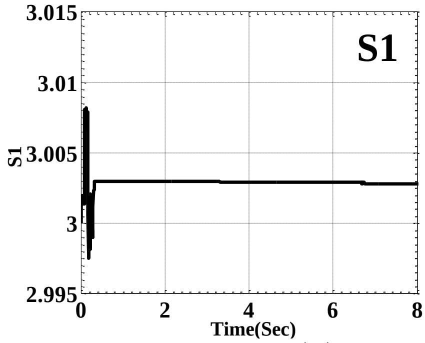

Fig. 19. Time history of (S_1).

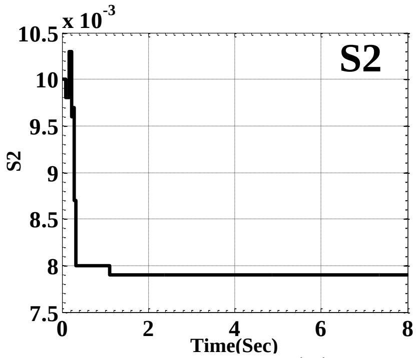

Fig. 20. Time history of (S_2).

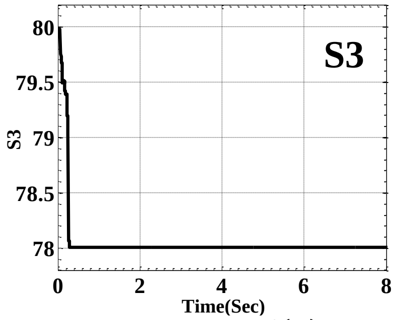

Fig. 21. Time history of (S_3).

Table V presents a numerical comparison of the results of the five controllers for the response of a step input with an initial error of 0.32 rad. As it can be noticed, PID-AT exhibits the maximum overshoot while PD results have the maximum settling time and the maximum rise time. In comparison with PID-AT, PID, PD, and LQR controller, STFC exhibits minimum overshoot, rise time, and settling time.

TABLE V: PERFORMANCE RESULTS OF THE TESTED CONTROLLERS

	STFC	PID-AT	LQR	PID	PD
Max. Over shoot %	2.7	9.09	4.5	6.05	5.3
Settling time, sec	0.37	0.4	0.42	0.43	0.45
Rise time, sec	0.33	0.33	0.35	0.36	0.38

VII. CONCLUSIONS

In this work, an update law for the scaling factors of a PD-like FLC has been derived. The methodology is based on the gradient descent and back-propagation which is widely used in neural networks. The control algorithm, i.e. STFC, has been experimentally verified using an inverted pendulum mounted on a cart. Further we compare the results with classical controllers like PD, LQR, PID, and PID-AT. Experimental results for a step input show that the proposed STFC technique outperforms the other controllers.

REFERENCES

[1] C. Chao and C. Teng, "A PD-like self-tuning fuzzy controller without steady-state error," *Fuzzy Sets Syst.*, vol. 87, pp. 141–154, 1997.

[2] L. Cammarata and L. Yliniemi, *Department of Process Engineering Development of a Self-Tuning Fuzzy Logic Controller for a Rotary Dryer*, Report A No. 10, December 1999.

[3] S.-K. Oh, W. Pedrycz, S.-B. Rho, and T.-C. Ahn, "Parameter estimation of fuzzy controller and its application to inverted pendulum," *Eng. Appl. Artif. Intell.*, vol. 17, no. 1, pp. 37–60, Feb. 2004.

[4] F. Nagi, A. T. Zulkarnain, and J. Nagi, "Tuning fuzzy Bang–bang relay controller for satellite attitude control system," *Aerosp. Sci. Technol.*, vol. 26, no. 1, pp. 76–86, Apr. 2013.

[5] F. Nagi and A. Abidin, "Tuning of a new fuzzy bang? Bang relay controller for attitude control system," *Autom. Control*, vol. 5, no. 2, pp. 97–118, 2011.

[6] A. Habbi and M. Zelmat, "An improved self-tuning mechanism of fuzzy control by gradient descent method," in *Proc. 17th Eur. Simul.*, 2003.

[7] H. Chung, B. Chen, and J. Lin, "A PI-type fuzzy controller with self-tuning scaling factors," *Fuzzy Sets Syst.*, vol. 93, pp. 23–28, 1998.

[8] O. Karasakal and E. Yesil, "Implementation of a new self-tuning fuzzy PID controller on PLC," *Turk J Elec Engin*, vol. 13, no. 2, pp. 277–286, 2005.

[9] A. A. Aly, A. S. A. El-Lail, K. A. Shoush, and F. A. Salem, "Intelligent PI fuzzy control of an electro-hydraulic manipulator," *Int. J. Intell. Syst. Appl.*, vol. 4, no. 7, pp. 43–49, Jun. 2012.

[10] H. Hasanien, "Speed control of switched reluctance motor using an adaptive neuro-fuzzy controller," in *Proc. World Congr. Eng.*, vol. II, no. 3, pp. 3–6, 2013.

[11] A. Marwan, K. Sahari, S. Hanim, and I. Fadi, "On-line adaptive fuzzy switching controller for SCARA robot," *Syst. Control*, vol. 6, no. 11, pp. 404–416, 2011.

[12] A. Marwan and N. Farrukh, "Real-time on line tuning of fuzzy controller for two-link rigid-flexible robot manipulators," *Control*, vol. 11, no. 12, pp. 381–390, 2012.

[13] W. Li, H. Ding, and K. Cheng, "An investigation on the design and performance assessment of double-PID and LQR controllers for the inverted pendulum," in *Proc. 2012 UKACC Int. Conf. on Control*, September, 2012, pp. 190-196.

[14] L. Bush, "Fuzzy logic controller for the inverted pendulum problem," Rensselaer Polytech Institute, Troy, New York, 2001.

[15] V. K. E and J. Jerome, "Stabilizing controller design for self erecting single inverted pendulum using robust LQR," *Aust. J. Basic Appl. Sci.*, vol. 7, no. 7, pp. 494–504, 2013.

[16] A. Ahmed, M. Ajangnay, and M. Dunnigan, "New approach for position control of induction motor," *Sustech.Edu*, 2010.

[17] J.-J. Wang, "Simulation studies of inverted pendulum based on PID controllers," *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 440–449, Jan. 2011.

[18] A. N. K. Nasir, M. A. Ahmad, R. Ghazali, and N. S. Pakheri, "Performance comparison between fuzzy logic controller (FLC) and PID controller for a highly nonlinear two-wheels balancing robot," in *Proc. First Int. Conf. Informatics Comput. Intell.*, Dec. 2011, pp. 176–181.

[19] A. N. K. Nasir, M. A. Ahmad, M. F. Rahmat, N. Barsoum, S. Uatrongjit, and P. Vasant, "Performance comparison between lqr and pid controllers for an inverted pendulum system," in *Proc. AIP Conf.*, vol. 1052, 2008, pp. 124–128.

[20] K. Ogata and Y. Yang, *Modern control engineering*, vol. 3 no 10, pp.670-680, ch. 13, pp. 912-923. 1997

[21] A. B. Sharkawy, "Genetic fuzzy self-tuning PID controllers for antilock braking systems," *Eng. Appl. Artif. Intell.*, vol. 23, no. 7, pp. 1041–1052, Oct. 2010.

[22] Z. W. Woo, H. Y. Chung, and J. J. Lin, "A PID type fuzzy controller with self-tuning scaling factors," *Fuzzy Sets Syst.*, vol. 115, no. 2, pp. 321–326, Oct. 2000.

Gamal Abdel Nasser got his B.Sc. degree in mechanical engineering from Mechanical Engineering Department, Mechatronics Section, Assiut University, Egypt in 2010. He is currently a demonstrator at the same department. His research interests lie in the field of robotics, intelligent control and identification.

Abdel Badie Sharkawy received the B.Sc. degree in mechanical engineering, the M.Sc. degree in production engineering from Assiut University, Egypt, in 1981 and 1990 respectively. He received the Ph.D degree in control engineering from the Slovak Technical University (STU) in Bratislava in 1999. He was a senior lecturer within the department of mechatronics engineering, the Hashemite University, Jordan for three years (2001-2004) and within the electrical engineering department, Al-Tahady University, Sirte, Libya during the fall semester, 2005. He is currently a professor at mechanical engineering department, Assiut University, Egypt (April 2012-now). His research interests include adaptive fuzzy identification and control, automotive control systems, robotics (modeling and control), and the use of neural networks in the control of mechanical systems.

M-Emad S. Soliman received his B.Sc. and M.Sc. degrees in mechanical engineering (design & production engineering) from Assiut University, Egypt, in 1982 and 1990 respectively. He spent two years (1993-1995) at Northwestern University, IL, USA as a visiting scholar doing research through a joint program with Assiut University towards his Ph.D. He received his Ph.D. degree in systems engineering from Assiut University, Egypt in 2000. He was a senior lecturer within the Department of Mechanical Engineering, Elmergeb University, Khoms, Libya for four years (2005-2009). He is currently an assistant professor at mechanical engineering department, Assiut University, Egypt. His research interests include mechatronic systems, mechanical vibrations, vehicle suspension systems, noise control, and metal forming.