A New Self-Tuning Fuzzy PI Controller for Integrating Processes with Dead-Time

Ritu Rani De (Maity)
Dept. of Electronics & Instrumentation Engineering
Dr. B.C. Roy Engineering College
Durgapur, INDIA
e-mail : *ritu_maity_8@yahoo.co.in*
* *Author for correspondence*

*Rajani K. Mudi
Dept. of Instrumentation & Electronics Engineering
Jadavpur University, Salt-lake Campus,
Salt-lake, Kolkata – 700 098, INDIA,
e-mail : *rkmudi@yahoo.com*

Abstract — **We propose a robust non-fuzzy self-tuning scheme for fuzzy PI controllers (FPIC) for a class of systems. The output scaling factor (SF) of the proposed non-fuzzy self-tuning FPIC, termed as NF-STFPIC, is modified on-line based on the normalized change of error of the controlled variable. Performance of the proposed controller is evaluated for a pure integrating plus dead-time (IPD) and first-order integrating plus dead-time (FOIPD) processes with a wide variation in dead-time under both set-point change and load disturbance. Detailed performance comparison with conventional PI (both fuzzy and non-fuzzy) controllers as well as a self-tuning fuzzy PI controller (STFPIC) reported in the leading literature is provided with respect to a number of performance indices. Unlike STFPIC, which uses 49 expert's defined self-tuning fuzzy rules our NF-STFPIC uses a single deterministic rule. Experimental results justify the effectiveness of the proposed NF-STFPIC.**

Keywords **– Fuzzy logic controller, Self-tuning FLCs, Scaling factor, Integrating process.**

I. INTRODUCTION

Fuzzy logic controllers (FLCs) are being popular in process control applications due to their inherent capabilities of handling linear as well as highly non-linear and high-order systems. In a number of difficult processes they have been successfully used and proved to be superior to the conventional non-fuzzy controllers [1, 2]. Even they are found to be less sensitive to variations in process parameters than conventional controllers [3]. Usually two types of FLC structures have been considered so far; one is PD-type FLC (FPDC) which generates control action (u) from error (e) and change of error (Δe) of the controlled variable, and the other is PI-type FLC (FPIC), which generates incremental control action (Δu) from e and Δe. PI-type FLCs are most common and practical. However, like conventional PI-controllers, performance of PI-type FLCs is found to be poor for high-order systems, specially, systems with integrating element due to large overshoot and excessive oscillation [4-6].

Conventional FLC is designed using a number of fuzzy *If-Then* rules defined on e and Δe of the controlled variable. The membership functions (MFs) of the input and output linguistic variables are usually defined on a common normalized domain. For the successful design of FLCs proper selection of input and output scaling factors (SFs) and/or tuning of the other controller parameters are crucial jobs which in many cases are done through trial or based on experimental data [7, 8]. Most of the practical processes are nonlinear systems with dead time, and their parameters change with time and ambient conditions. Thus, a conventional FLC with a limited number of rules using simple MFs, may not provide satisfactory control performance. In order to overcome such limitations many research works on tuning of FLCs have been reported where either the input-output SFs or the definitions of MFs and sometimes the control rules are tuned (on-line or off-line) to achieve the desired performance [7-14]. Though many authors have proposed several tuning schemes for improving the performance of FLCs, unlike traditional controllers a standard and systematic method for tuning of FLCs is yet to be developed.

The self-tuning fuzzy PI-type controller (STFPIC) [9] is tuned by dynamically adjusting its output SF in each sampling instant by a gain updating factor (α), which is further augmented by a fixed multiplicative factor chosen empirically. The value of α is determined by 49 fuzzy rules defined on e and Δe, and derived from the knowledge of process control engineering. Instead of 49 expert's defined fuzzy rules, here, we propose a non-fuzzy self-tuning scheme for fuzzy PI-type controller (NF-STFPIC). In the proposed NF-STFPIC, its output SF is continuously modified by a single heuristic rule defined on the normalized change of error, *i.e.*, Δe_N, and the number of fuzzy input partitions. We know that Δe_N actually indicates the instantaneous process trend in terms of speed of response. Thus, the on-line adjusted output SF of the proposed NF-STFPIC is expected to improve the close-loop performance, since it incorporates the dynamics of the process under control. Such knowledge and information have been embedded while developing improved auto-tuning PI/PID controllers [15-17]. The performance of NF-STFPIC is tested by simulation experiments on a number of integrating processes with different values of dead-time. Results in each case show a significantly improved performance of the proposed NF-STFPIC compared to its conventional fuzzy (FPIC) and non-fuzzy PI/PID controllers, and better than or comparable with that of STFPIC [9].

II. DESIGN OF THE PROPOSED CONTROLLER – NF-STFPIC

Block diagram of the NF-STFPIC is shown in Fig. 1. The output SF of the controller is modified by a non-fuzzy relation, which is shown above the FLC block (Fig. 1). Various design aspects of the NF-STFPIC are discussed below.

MFs for inputs, (*i.e.*, e_N and Δe_N) and output, (*i.e.*, Δu_N) of the controller (shown in Fig. 2) are defined on the common normalized domain [-1, 1]. Except for the MFs at the two extreme ends, symmetric triangles are used with equal base and 50% overlap with neighboring MFs. The relationships between the SFs (G_e , $G_{\Delta e}$ and G_u), and the input and output variables of the NF-STFPIC are as follows:

$$e_N = G_e \times e, \quad (1)$$

$$\Delta e_N = G_{\Delta e} \times \Delta e, \quad (2)$$

$$\text{and } \Delta u = (\beta G_u) \times \Delta u_N \quad (3)$$

$$\text{Where, } \beta = K_1[1/m + |\Delta e_N|] \quad (4)$$

In Eqn. 4, β is the on-line gain updating factor for the output SF (G_u), K_1 is a constant, which will make required variation in β, and m is the number of uniform input (e_N and Δe_N) fuzzy partitions. In our study, m = 7 as shown in Fig. 2. Unlike fuzzy PI controllers (FPIC), which uses only G_u to generate the incremental output (*i.e.*, $\Delta u = G_u \times \Delta u_N$), the actual output ($\Delta u$) for NF-STFPIC is obtained by using the effective SF (βG_u) as shown in Fig. 1. Note that, unlike STFPIC [9] with 49 fuzzy self-tuning rules, here β is computed on-line using a *single* model independent non-fuzzy relation defined by Eqn. (4).

A PI-type FLC in its velocity form can be described by

$$u(k) = u(k\text{-}1) + \Delta u(k). \quad (5)$$

In Eqn. (5) k is the sampling instance and Δu is the incremental change in controller output.. The rule-base for computing Δu is shown in Fig. 3, which is derived following the principle of sliding mode control [18]. The gain updating factor (β) is calculated using relation (4), which is formulated according to the rule-base in Fig. 3 with the following idea : *when the process is moving rapidly towards it's set-point, control action should be reduced to prevent possible large overshoot and/or undershoot; on the other hand, when the process is moving away very fast from the set-point, control action should be increased to restrict such deviations for a quick recovery of the process to its desired value.*

Observe that in NF-STFPIC the controller output ($\Delta u_{NF\text{-}STFPIC}$) is generated by modifying the output of a simple FLC (Δu_{FPIC}).

$$i.e., \ \Delta u_{NF\text{-}STFPIC} = \beta\,(\Delta u_{FPIC}), \quad (6)$$

From Eqn. (6) we can say that the NF-STFPIC is equivalent to a simple PI-type FLC (FPIC) with a dynamic output SF, *i.e.*, controller gain.

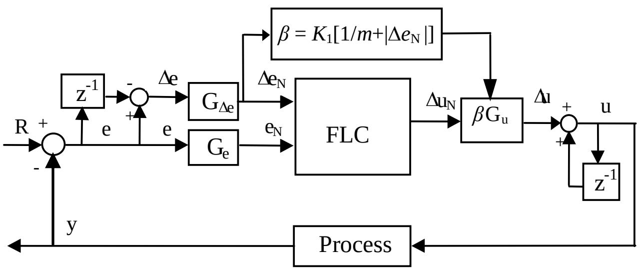

Fig. 1. Block Diagram of the proposed NF-STFPIC

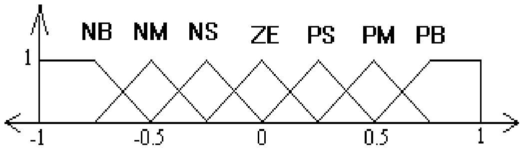

Fig. 2. MFs of e_N, Δe_N and Δu_N.

Δe/e	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NM	NS	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PM	PB
PB	ZE	PS	PS	PM	PB	PB	PB

Fig. 3. Fuzzy Rules for Computation of Δu.

III. RESULTS

The performance of NF-STFPIC is compared with STFPIC [9] and conventional fuzzy (FPIC) and non-fuzzy PI/PID controllers, *i.e.*, Ziegler-Nichols (ZN) and Modified Ziegler-Nichols (TL) tuned controllers ZNPIC, TLPIC, ZNPID, and TLPID [15, 16, 19, 20], for a pure integrating plus dead-time (IPD) and first-order integrating plus dead-time (FOIPD) processes with a wide variation in dead-time under both set-point change and load disturbance. The performance of the controllers are evaluated in terms of peak overshoot (%OS), settling time (t_s), rise time (t_r), and integral absolute error (IAE). To establish the robustness of the proposed scheme we use the same MFs (Fig.2) and same rule-base (Fig.3) for all the processes used for simulation study. In Eqn. 4, β is determined using K_1 = 4.0, which is empirically found through simulation study on a large number of integrating processes with dead-time (L). We have used Mamdani type inferencing and Height method of defuzzification [2,9].

Here we report the simulation results for the following three integrating processes.

$$IPD: \quad G_P(s) = e^{-LS}/s \quad (7)$$

$$FOIPD\text{-}1: \quad G_P(S) = e^{-LS}/[s(s+1)] \quad (8)$$

$$FOIPD\text{-}2: \quad G_P(s) = e^{-LS}/[s(2s+1)] \quad (9)$$

For the pure integrating plus dead-time (IPD) process of (7), we have considered different values of dead time (*i.e.*, L = 0.2, 0.3. 0.4, and 0.5). Responses of this process with L = 0.2 and 0.5 due to both set point change and load disturbance under various FLCs (*i.e.*, NF-STFPIC, STFPIC, and FPIC) and non-fuzzy controllers ZNPIC and TLPIC are shown in Figs. 4a-4c. Detailed performance analysis in terms of various performance indices are provided in Table I. From the results (Figs. 4a and 4b, and Table I) it clearly reveals that FPIC fails to provide acceptable performance when the process is subjected to a large change in dead-time. But in the same situation, our proposed NF-STFPIC maintains satisfactory performance, which justifies the effectiveness of our proposed scheme.

Moreover, the STFPIC that uses additional 49 fuzzy rules for gain adjustment exhibits almost similar performance. Note that for this integrating process even PID controllers (ZNPID and TLPID) show very poor performance as revealed by Table I and Fig. 4d. Similar to the IPD process, we have also considered a wide variation in L (*i.e.*, L = 0.2, 0.3, 0.4, and 0.5) for first-order integrating plus dead-time processes, FOIPD-1 in (8) and FOIPD-2 in (9). Response characteristics of (8) are depicted in Fig. 5 for different values of L under various controllers. Table II provides the detailed performance comparison. Like previous results, NF-STFPIC exhibits improved performance compared to others. Responses of the marginally stable process FOIPD-2 in (9) are shown in Fig. 6, and various performance indices are recorded in Table III. In this case also, we see that the performance of NF-STFPIC is significantly improved over FPIC, ZNPIC, and TLPIC, and better than that of STFPIC.

TABLE I. PERFORMANCE COMPARISON FOR THE IPD PROCESS IN (7).

L	Controller	%OS	t_s	t_r	IAE
0.2	ZNPIC	60.96	28.8	1.4	8.56
	TLPIC	44.29	31.4	2.9	14.45
	ZNPID	71.88	26.7	1.0	7.22
	TLPID	39.33	25.3	2.3	9.37
	FPIC	45.45	15.3	1.5	5.38
	STFPIC	15.6	11.7	1.4	3.78
	NF-STFPIC	20.7	14.1	1.4	3.78
0.3	ZNPIC	56.72	29.5	1.8	5.84
	TLPIC	39.63	36.6	3.3	9.87
	FPIC	52.02	16.1	1.4	6.2
	STFPIC	18.42	11.0	1.5	3.07
	NF-STFPIC	19.45	13.5	1.3	3.9
0.4	ZNPIC	56.96	32.7	1.9	11.62
	TLPIC	37.94	31.8	3.7	20.52
	FPIC	61.7	22.2	1.3	8.1
	STFPIC	28.22	16.1	1.2	3.61
	NF-STFPIC	26.93	16.3	1.3	4.5
0.5	ZNPIC	55.7	31.4	2.0	13.03
	TLPIC	35.94	35.5	4.1	23.41
	FPIC	74.77	34.8	1.3	12.4
	STFPIC	40.87	21.2	1.2	4.9
	NF-STFPIC	39.06	22.3	1.3	5.4

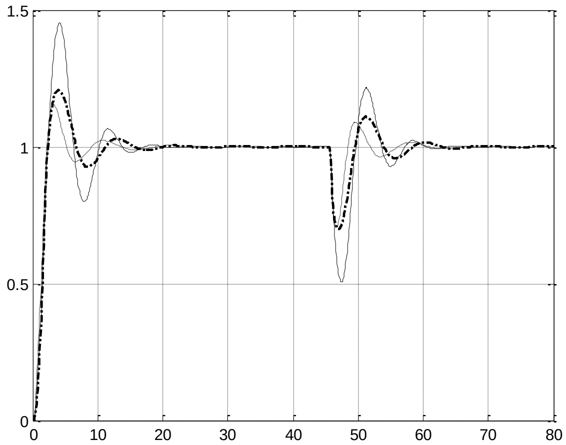

Fig. 4a. Responses of (7) for L=0.2; FPIC(—);STFPIC(– –),NF-STFPIC(– ·).

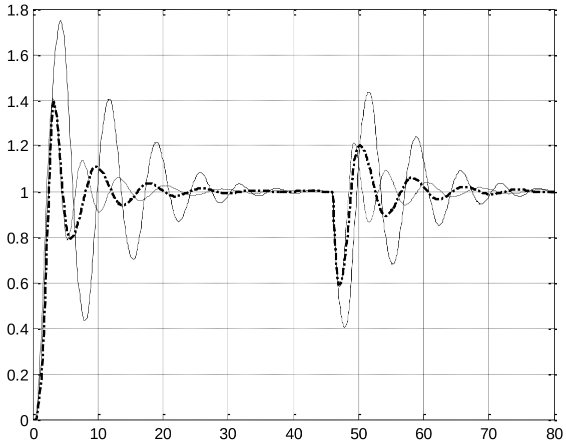

Fig. 4b.Responses of (7) for L=0.5; FPIC(—),STFPIC(– –),NF-STFPIC(– ·).

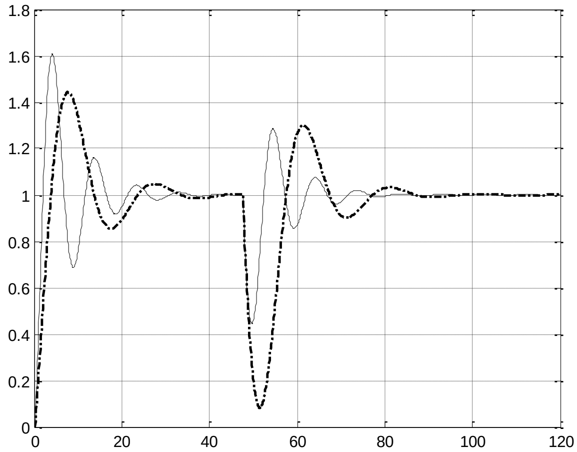

Fig. 4c. Responses of (7) for L=0.2; ZNPIC(—) and TLPIC(– ·).

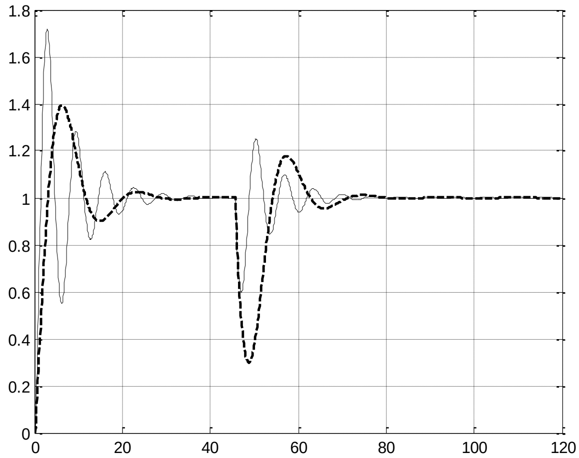

Fig. 4d. Responses of (7) for L=0.2; ZNPIDC(—) and TLPIDC(– ·).

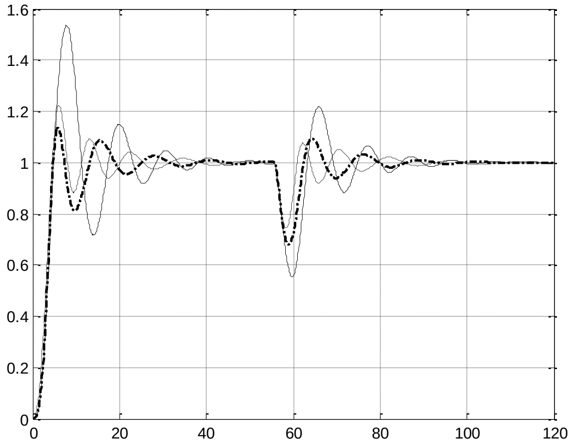

Fig. 5a. Responses of (8) for L=0.2; FPIC(—);STFPIC(– –), NF-STFPIC(– ·).

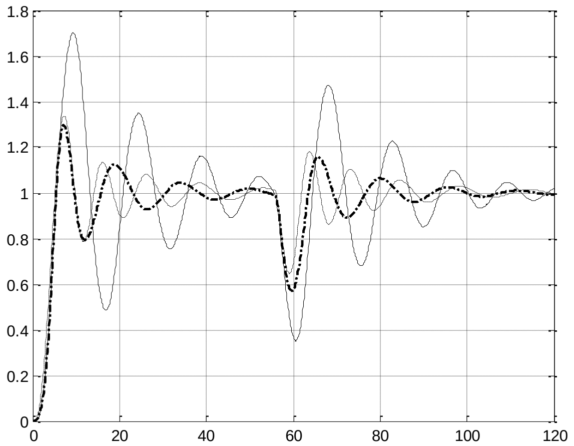

Fig. 5b. Responses of (8) for L=0.4; FPIC(—);STFPIC(– –),NF-STFPIC(– ·).

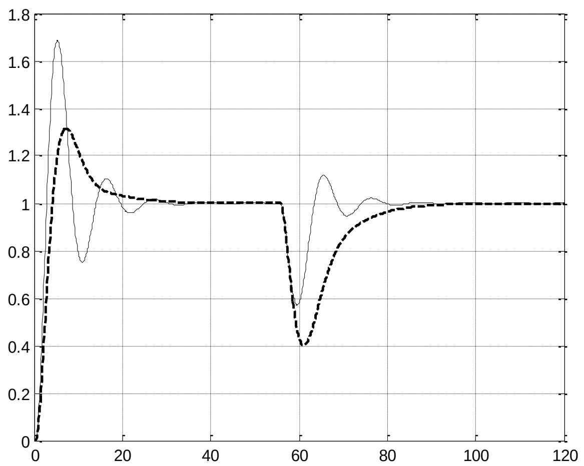

Fig. 5c. Responses of (8) for L=0.4; ZNPIC(—) and TLPIC(– ·).

.

TABLE II. PERFORMANCE COMPARISON FOR THE FOIPD-1 PROCESS IN (8).

L	Controller	%OS	t_s	t_r	IAE
0.2	ZNPIC	88.92	36.6	1.0	8.35
	TLPIC	48.0	13.7	1.4	4.88
	ZNPID	84.18	19.7	0.9	5.17
	TLPID	29.23	13.7	1.3	3.76
	FPIC	53.8	36.7	2.7	11.64
	STFPIC	22.25	29.5	2.8	6.26
	NF-STFPIC	13.6	29.3	2.5	7.07
0.3	ZNPIC	75.92	28.6	1.3	7.46
	TLPIC	37.64	18.2	2.0	7.06
	FPIC	62.77	54.3	2.9	17.18
	STFPIC	27.78	40.2	3.0	8.24
	NF-STFPIC	23.26	34.9	2.7	9.10
0.4	ZNPIC	68.84	23.8	1.7	8.47
	TLPIC	31.5	23.6	2.6	11.27
	FPIC	70.4	71.6	2.9	24.14
	STFPIC	33.77	54.1	3.0	10.32
	NF-STFPIC	29.78	43.9	2.8	10.83
0.5	ZNPIC	69.92	25.3	1.7	9.32
	TLPIC	31.16	25.8	2.8	12.83
	FPIC	77.19	104.4	3.0	31.2
	STFPIC	39.0	69.8	3.1	13.22
	NF-STFPIC	36.35	54.6	2.9	13.08

TABLE III. PERFORMANCE COMPARISON FOR THE FOIPD-2 PROCESS IN (9).

L	Controller	%OS	t_s	t_r	IAE
0.2	ZNPIC	94.37	98.2	1.4	18.5
	TLPIC	56.17	28.7	2.0	7.2
	ZNPID	84.47	37.1	1.3	8.34
	TLPID	33.05	19.4	1.9	4.96
	FPIC	45.8	57.8	5.3	18.5
	STFPIC	15.96	47.0	5.6	10.67
	NF-STFPIC	7.47	44.9	5.0	12.08
0.3	ZNPIC	83.86	64.1	1.8	14.62
	TLPIC	46.52	25.8	2.6	9.14
	FPIC	49.2	59.3	5.3	19.9
	STFPIC	18.72	54.0	5.5	11.4
	NF-STFPIC	9.06	53.8	4.9	12.57
0.4	ZNPIC	82.29	61.1	2.1	14.8
	TLPIC	42.96	30.7	3.1	11.52
	FPIC	52.9	60.0	5.2	21.45
	STFPIC	21.92	62.7	5.4	12.49
	NF-STFPIC	11.73	53.6	4.8	13.03
0.5	ZNPIC	75.35	50.6	2.6	14.57
	TLPIC	37.27	36.6	3.8	15.19
	FPIC	56.54	60.0	5.1	24.32
	STFPIC	25.15	69.8	5.1	13.91
	NF-STFPIC	14.5	62.2	4.8	14.06

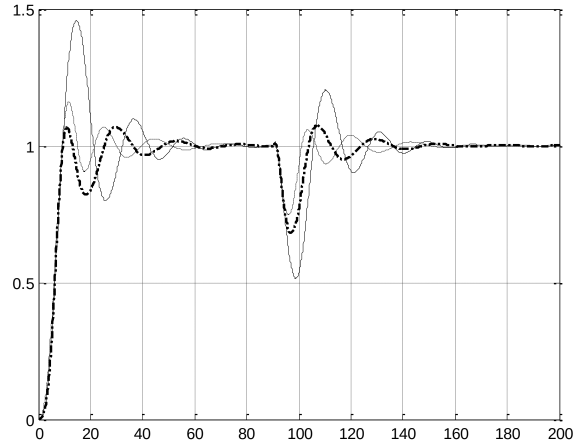

Fig. 6a. Responses of (9) for L=0.2; FPIC(**—**),STFPIC(– –),NF- STFPIC(– ·**).**

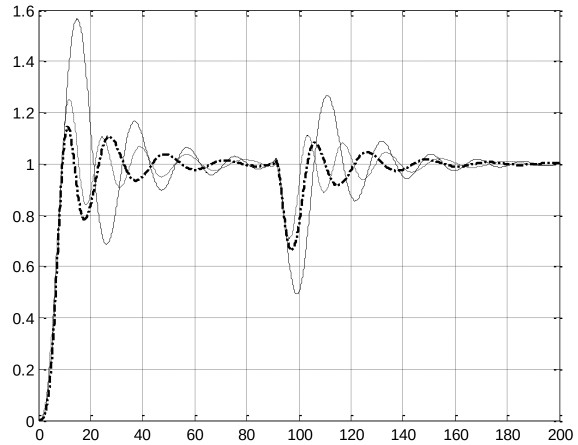

Fig. 6b. Responses of (9) for L=0.5; FPIC(**—**),STFPIC(– –),NF-STFPIC(– ·**).**

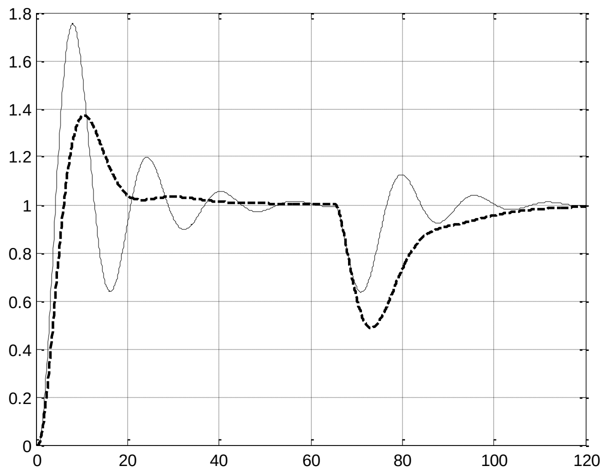

Fig. 6c. Responses of (9) for L=0.5; ZNPIC(**—**) and TLPIC(– ·**).**

IV. CONCLUSION

We proposed a self-tuning fuzzy PI controller whose output SF gets real-time adjustment by a non-fuzzy gain updating parameter defined on the normalized change of error, which indicates the instantaneous process trend. The proposed self-tuning scheme thus became more rational, since it embedded the dynamics of the process under control. Performance of the proposed controller has been tested under both set-point change and load disturbance for a number of integrating processes, for which conventional fuzzy as well as non-fuzzy controllers provide very poor performance. Comparisons with a well known self-tuning FLC (STFPIC) with additional 49 fuzzy self-tuning rules justified the effectiveness of the proposed controller.

REFERENCES

[1] M. Sugeno, *Industrial Applications of Fuzzy Control*, Amsterdam: Elsevier Science, 1985.

[2] D. Dirankov, H. Hellendoorn & M. Reinfrank, *An Introduction to Fuzzy Control*, NY : Springer-Verlag, 1993.

[3] C. J. Harris, C. G. Moore & M. Brown, *Intelligent Control - Aspects of Fuzzy Logic and Neural Nets*, Singapore : World Scientific, 1993.

[4] H. Ying, W. Siler, and J. J. Buckley, "Fuzzy Control Theory: A Nonlinear Case," *Automatica*, vol. 26, pp. 513-520, 1990.

[5] S. Boverie, "Fuzzy Logic Control for High-order Systems," *Proc. 2nd IEEE Int. Conf. on Fuzzy Systems,* pp. 117-121, 1993.

[6] J. Lee, "On Methods for Improving Performance of PI-Type Fuzzy Logic Controllers," *IEEE Trans. on Fuzzy Syst*., vol. 1, no. 4, pp. 298-301, 1993.

[7] H. Nomura, I. Hayashi and N. Wakami, "A Self-Tuning Method of Fuzzy Control by Decent Method," *Proc. IFSA '91*, pp. 155-158, 1991.

[8] Hung-Yuan Chung, Bor-Chin Chen and Jin-Jye Lin, "A PI-type Fuzzy Controller with Self-tuning Scaling Factors," *Fuzzy Sets and Syst.*, vol. 93, pp. 23-28, 1998.

[9] R. K. Mudi and N. R. Pal, "A Robust Self-Tuning Scheme for PI and PD Type Fuzzy Controllers," *IEEE Trans. on Fuzzy Systems*, vol.7, no. 1, pp. 2-16, 1999.

[10] R. Palm, "Scaling of Fuzzy Controller Using the Cross-Correlation," *IEEE Trans. on Fuzzy Syst*., vol. 3, no. 1, pp. 116-123, 1995.

[11] R. K. Mudi and N. R. Pal, "A Self-Tuning Fuzzy PI Controller," *Fuzzy Sets and Systems*, vol. 115, no. 2, pp. 327-338, 2000.

[12] H. X. Li and H. B. Gatland, "Conventional Fuzzy Control and Its Enhancement," *IEEE Trans. on Syst., Man, Cybern.,* vol. 26, no. 5, pp. 791-797, 1996.

[13] R.K. Mudi and N. R. Pal, "A Self-Tuning Fuzzy PD Controller," *IETE Journal of Research*, vol., 44, no. 4 & 5 (*Special Issue* on Fuzzy Systems), pp. 177-189, 1998.

[14] A.K.Pal and R.K.Mudi, "Self-Tuning Fuzzy PI controller and its application to HVAC system," IJCC (US), vol.6, no.1, 2008.

[15] R.K. Mudi, C. Dey, and T.T. Lee, 'An improved auto-tuning scheme for PI controllers', ISA Transactions, vol. 47, no. 1, pp. 45-52, 2008.

[16] C. Dey and R.K. Mudi, 'An improved auto-tuning scheme for PID controllers', *ISA Tran actions*, vol. 48, no. 4, pp. 396-409, 2009

[17] R.K. Mudi and C. Dey, " Performance improvement of PI controllers through dynamic set-point weighting,' *ISA Transactions*, vol. 50, no. 2, pp. 220-230, 2011.

[18] R. Palm, " Sliding Mode Fuzzy Control," *Proc. 1st IEEE Int. Conf. on Fuzzy Systems*, pp. 519-526, 1992.

[19] K.J. Åström and T. Hägglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins," Automatica, vol.20, pp. 645-651, 1984.

[20] K.J. Åström and T. Hägglund, The future of PID control, Control Engineering Practice, vol.9, pp 1163–1175, 2001.