# Self-tuning PI Controllers via Fuzzy Cognitive Maps

Engin Yesil, M. Dodurka, Ahmet Sakalli, Cihan Ozturk, Cagri Guzay

# Self-Tuning PI Controllers via Fuzzy Cognitive Maps

Engin Yesil[1], M. Furkan Dodurka[1,2],

Ahmet Sakalli[1], Cihan Ozturk[1], Cagri Guzay[1]

[1]Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control Engineering Department, Maslak, TR-34469, Istanbul, Turkey
[2]GETRON Bilişim Hizmetleri A. Ş., Yıldız Teknik Üniversitesi Davutpaşa Kampüsü, Teknopark Binası B1 Blok, Esenler, 34220, Istanbul, Turkey

{yesileng, dodurkam, sakallia, ozturkci, guzay}@itu.edu.tr
furkan.dodurka@getron.com

**Abstract.** In this study, a novel self-tuning method based on fuzzy cognitive maps (FCMs) for PI controllers is proposed. The proposed FCM mechanism works in an online manner and is activated when the set-point (reference) value of the closed loop control system changes. Then, FCM tuning mechanism changes the parameters of PI controller according to systems' current and desired new reference value to improve the transient and steady state performance of the systems. The effectiveness of the proposed FCM based self-tuning method is shown via simulations on a nonlinear system. The results show that the proposed self-tuning methods performances are satisfactory.

**Keywords:** Fuzzy cognitive maps, PI controllers, self-tuning, supervisory control, optimization.

## 1 Introduction

Although many innovative methodologies have been devised in the past 50 years to handle complex control problems and to achieve better performances, the great majority of industrial processes are still controlled by means of simple proportional-integral-derivative (PID) controllers. PID controllers, despite their simple structure, assure acceptable performances for a wide range of industrial plants and their usage (the tuning of their parameters) is well known among industrial operators. Hence, PID controllers provide, in industrial environments, a cost/benefit performance that is difficult to beat with other kinds of controllers. Aström states that more than 90% of all control loops utilize PID and most of loops are in fact PI [1].

Cognitive maps were introduced for the first time by Axelrod [2] in 1976 in order to signify the binary cause-effect relationships of the elements of an environment. Fuzzy cognitive maps (FCM) are fuzzy signed directed graphs with feedbacks, and they can model the events, values, goals as a collection of concepts by forging a causal link between these concepts [3]. FCM nodes represent concepts, and edges repre-

AIAI 2013, IFIP AICT, 2013.

sent causal links between the concepts. Most widely used aspects of the FCMs are their potential for use in learning from historical data and decision support as a prediction tool. Given an initial state of a system, represented by a set of values of its constituent concepts, an FCM can simulate its evolution over time to learn from history and predict its future behavior. For instance, it may stand for that the system would converge to a point where a certain state of balance would exist, and no further changes would occur.

The main advantages of FCMs are their flexibility and adaptability capabilities [4]. As mentioned in [5], [6] and [7], there is a vast interest in FCMs and this interest on the part of researchers and industry is increasing in many areas such as control. In [8], FCM is studied for modeling complex systems and controlling supervisory control systems. In [9], learning approaches based on nonlinear Hebbian rule to train FCMs that model industrial process control problems is performed. A cognitive–fuzzy model, aiming online fuzzy logic controller (FLC) design and self-fine-tuning is implemented [10]. Fuzzy cognitive network (FCN) is used to the adaptive weight estimation based on system operation data, fuzzy rule storage mechanism to control unknown plants [11]. Besides, FCN is used to construct a maximum power point tracker (MPPT) that operates in cooperation with a fuzzy MPPT controller [12]. By combining topological and metrical approaches, an approach to mobile robot map-building that handles qualitatively different types of uncertainty is proposed [13]. A method for neural network FCM implementation of the fuzzy inference engine using the fuzzy columnar neural network architecture (FCNA) is proposed [14].

In this paper, a novel procedure is proposed to design a self-tuning PI controller via FCM particularly for nonlinear systems. Because of nonlinear systems' dissimilar characteristics at different operating points, fixed PI controllers cannot perform successive behaviors. In the proposed self-tuning method, FCM is used to supervise the control system and decide to change the controller parameters when the operating point changes.

## 2 A Brief Overview of Fuzzy Cognitive Maps

A fuzzy cognitive map F is a 4-tuple (N, W, C, f) [15] where; $N = \{N_1, N_2, \ldots, N_n\}$ is the set of n concepts forming the nodes of a graph. W: $(N_i, N_j) \rightarrow w_{ij}$ is a function of $N \times N$ to K associating $w_{ij}$ to a pair of concepts $(N_i, N_j)$, with $w_{ij}$ denoting a weight of directed edge from $N_i$ to $N_j$ if $i \neq j$ and $w_{ij}$ equal to zero otherwise. Therefore, in brief, $W(N \times N) = (w_{ij}) \in K^{n \times n}$ is a connection matrix. C: $N_i \rightarrow C_i$ is a function that at each concept $N_i$ associates the sequence of its activation degrees such as for $t \in N$, $C_i(t) \in L$ given its activation degree at the moment t. $C(0) \in L^n$ indicates the initial vector and specifies initial values of all concept nodes and $C(t) \in L^n$ is a state vector at certain iteration t. f: $R \rightarrow L$ is a transformation function, which includes recurring relationship on $t \geq 0$ between $C(t + 1)$ and $C(t)$.

The sign of $w_{ij}$ expresses whether the relation between the two concepts is direct or inverse. The direction of causality expresses whether the concept $C_i$ causes the concept $C_j$ or vice versa. Thus, there are three types of weights [16]:

$W_{ij} > 0$, indicates positive causality,
$W_{ij} < 0$, indicates negative causality,
$W_{ij} = 0$, indicates no relation.

Values of concepts change as simulation goes on are calculated by the following formula [17]:

$$C_j(t+1) = f\left(\sum_{\substack{i=1 \\ i\neq j}}^{n} C_i(t) w_{ij}\right) \tag{1}$$

where $C_i(t)$ is the value of ith node at the $t^{th}$ iteration, $e_{ij}$ is the edge weight (relationship strength) from the concept $C_i$ to the concept $C_j$, t is the corresponding iteration, N is the number of concepts, and f is the transformation (transfer) function.

In general, there are two kinds of transformation functions used in the FCM framework. The first one is the unipolar sigmoid function, where $\lambda > 0$ decides the steepness of the continuous function *f* and transforms the content of the function in the interval [0,1].

$$f(x) = \frac{1}{1+e^{-\lambda x}} \tag{2}$$

The other transformation function, hyperbolic tangent, that has been used and which transforms the content of the function is in the interval [-1,1],

$$f(x) = tanh(\lambda x) = \frac{e^{\lambda x} - e^{-\lambda x}}{e^{\lambda x} + e^{-\lambda x}} \tag{3}$$

where $\lambda$ is a parameter used to determine proper shape of the function. Both functions use $\lambda$ as a constant for function slope.

## 3 PID Controllers

In industrial environments, PID controllers provide a cost/benefit performance that is difficult to beat with other kinds of controllers. It should be pointed that PID controllers actually possess characteristics of both PI and PD controllers. However, because of their simple structure, PID controllers are particularly suited for pure first or second order processes, while industrial plants often present characteristics such as high order, long time delays, nonlinearities and so on. For these reasons, it is highly desirable to increase the capabilities of PID controllers by adding new features; in this way, they can improve their performances for a wide range of plants while retaining their basic characteristics.

Åström and Hägglund [1] have stated that most of PID controllers that using in industry is PI controller, which means that D term is set to zero. Because of noises and disturbances on the process, derivative actions may affect control performances adversely. In order to avoid this drawback in industrial applications, PID controllers are mostly used as PI controllers without losing steady state performances. However, in some cases, control signal might be generated out of the operating range of actuators due to high-valued integral term. Because of these limitations and saturations, closed loop control structure might be broken for a while. Accordingly, system begins following set point with a short-term steady state error which is called wind-up [18].

PID controller in parallel form includes sum of proportional, integral, and derivative terms of the error signal. It can be written in time-domain as given as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \quad (4)$$

Laplace form of the (4) is simpler than the time-domain. Transfer function of a parallel connected PID controller is written in (5) as below

$$F(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (5)$$

where $K_p$ is the proportional, $K_i$ is the integral, $K_d$ is the derivative gain, $T_i$ is the integral $T_d$ is the derivative time constants. The tuning methods of PID controllers for the both linear and nonlinear systems have different approaches.

A system is said to be linear if it obeys the two fundamental principles of homogeneity and additivity. If a given process does not satisfy with the two principles, it can be said that the process is a nonlinear. The controller design for a linear system is more straight-forwardly in comparison with a nonlinear system. It is an obvious fact that the most PID controller design methods are focused on linear systems up to the present. The linear system is not dependent on initial conditions or operating point. [19] states that the essential disadvantage of existing design methods of PI or PID controllers is that desire transient responses cannot be assured for nonlinear systems especially parameter variations and unknown external disturbances. It is a novel idea to overcome this problem that self-tuning methods which determine the controller parameters might be used. As mentioned above, controlling the nonlinear systems with linear PID controllers is not a convenient strategy. Some studies show that combining of the PID controller with gain scheduling gives fine result [20]. The auto-tuning of a PID controller procedure needs gradually less exertion than gain schedule strategy [1]. In these mechanisms, PID parameters are the functions of error (and derivative of error) or/and process states [21]. There are varied prominent studies in self-tuning PID field. Supervisory control or self-tuning can be executed via different methods, for instance optimization based [21-23], fuzzy-logic mechanism [24, 25], neural networks [26, 27]. In this paper, for the first time in the literature and beside of mentioned self-tuning PID strategies, the self-tuning PI controller via FCM method is proposed.

## 4 Self-tuning Method Based on FCM for PI Controllers

In this section, the proposed self-tuning PI controller via fuzzy cognitive map, which is illustrated in Fig. 1, is presented particularly for the nonlinear systems. When the reference value (r(t)) of the closed-loop control system is changed, FCM tuning mechanism is triggered. Then, FCM self-tuning mechanism changes control parameters, which are the static gain ($K_p$) and the integral time constant ($T_i$) of PI controller according to system's current state and destination. Since systems are nonlinear, its dynamics and static characteristics are not same for different working points. So, as the process state changes the optimal parameters of PI control will naturally change nonlinearly. The nonlinear behavior of changing PI control parameters are mimicked with the help of FCM. Therefore, main objective is designing a FCM mechanism that represents nonlinear changing behavior of PI control parameters for different operating conditions. Since $K_p$ and $T_i$ are independent from each other, FCM design can be separated into two sub FCM design. The design methodology is the same for each controller parameter, so only one of the sub-FCM is discussed in details.

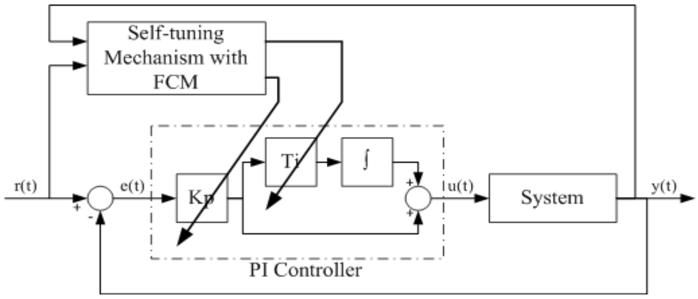


**Fig. 1.** Self-tuning PI controller structure

In a FCM design, the first step is determining the concepts. In the tuning mechanism, the system's current output and the desired set-point are the input nodes, given as $C_1$ and $C_2$. In addition, the output concept is $K_p$ (or $T_i$), as $C_N$. To represent nonlinearity of the self-tuning strategy a number of extra inner concepts $C_n$, which represent the nonlinear behavior of the parameter change, are needed. Here, $n = 3, 4, \dots, N$-1 and $N$ is total of the concepts depending on the system.

The next step is to determine which concepts are connected to each other. In proposed sub FCM, it is assumed that input concepts are affecting to the whole other concepts, moreover output concepts are affected by the whole other concepts. The inner nodes are affected from the previous other nodes, and affects to next nodes with a one iteration delay. Therefore, one of the proposed inner concept,$C_n$, is affected from the previous concepts $C_{n-1,}C_{n-2}, \dots, C_1$ representing nonlinearity. In a similar way, $C_n$ is affecting the further inner concepts $C_{n+1,}C_{n+2}, \dots, C_N$ with a one iteration delay.

The third step is to determine the transformation functions $f$ given in (3). In the proposed method all concepts expect inner nodes have their own transformation functions. By putting all together the change of $C_n$ provided that $n > 2$ at (t+1)th iteration will be calculated as follows:

$$C_n(t+1) = f_n(C_1(t) + C_2(t) + \sum_{\substack{i=3 \\ i \neq n}}^{N-1} C_i(t-(i-2))) \tag{6}$$

where $f_n$ is corresponding transformation function of $C_n$ with $\lambda_n$.

The last step is to determine connection matrix $W_{NxN}$ and $\lambda_n$ for transformation functions with an appropriate FCM learning methods [28-30] where $n = 3,4 \ldots, N$. Then the built two sub-FCMs are merged into a single FCM to construct the self-tuning mechanism.

## 5 Simulation Results

A second order nonlinear process with time delay is chosen in order to demonstrate the effectiveness of proposed self-tuning PI controller via FCM. As given in [31], the nonlinear process can be described by the following differential equation. In practical studies, time delay constant ($L$) is fixed to 5 seconds.

$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} + 0.25y^2 = u(t - L) \quad (7)$$

For this simulation example, the number of inner concepts that represent the non-linear relations in self-tuning is chosen as 6; therefore a FCM with 16 concepts is design. The FCM designed for self-tuning is given in Fig. 2. Concepts $C_1$ and $C_2$ are chosen as input concepts systems current state point and destination state point, which is the new reference value, respectively. Moreover, $C_9$ and $C_{16}$ are chosen as output concepts which represent $K_p$ and $T_i$.

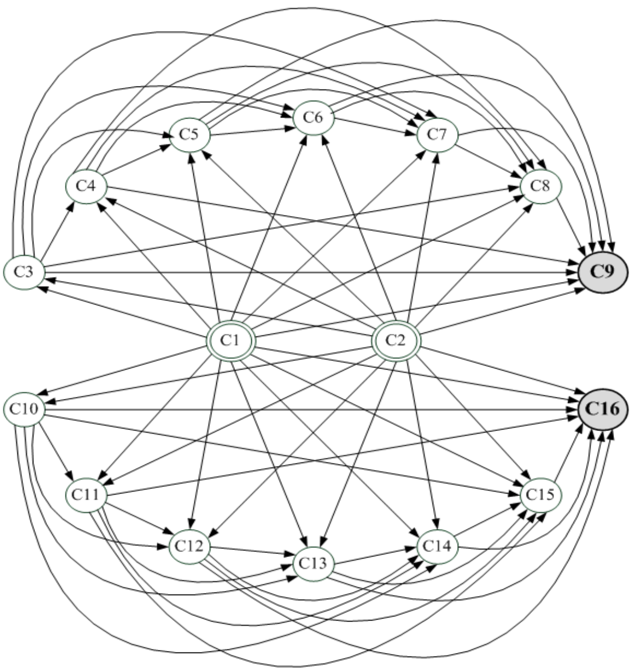


**Fig. 2.** Illustration of designed FCM for self-tuning mechanism

As a first step, optimal PI controller parameters for different operating points have been gathered using Big Bang – Big Crunch (BB-BC) algorithm [32] via many simulations. Integral square error (ISE) is chosen as the cost function in determination of the optimal PI parameters and historical data for learning of FCM is obtained. Then, all of the concepts values normalized in order to fall within the range [-1, 1] by dividing their maximum values respectively. For determination of the weights between the concepts of the proposed FCM, also, BB-BC learning methodology is used [29] with FCM-GUI [33]. The weight matrix obtained at the end of BB-BC learning is as follows:

$$
\mathrm{W} = \begin{bmatrix}
0 & 0 & -0.86 & -1.00 & -1.00 & 0.36 & -0.98 & 0.69 & 0.55 & 0.91 & 0.52 & -0.2 & 0.56 & 0.01 & 0.45 & -0.09 \\
0 & 0 & 0.74 & 0.86 & -0.44 & 0.79 & -0.47 & -0.96 & 0.92 & 0.62 & 0.59 & 1 & 0.12 & -0.97 & -0.19 & 0.5 \\
0 & 0 & 0 & -0.67 & -0.39 & -0.26 & 0.41 & 0.81 & -0.21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.97 & 0.16 & 0.54 & -0.42 & -0.91 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.19 & -0.75 & 0.24 & -0.71 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.35 & 0.21 & 0.63 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.74 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.33 & 0.2 & -0.70 & 1 & 0.52 & -0.91 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.01 & -0.48 & 0.22 & 0.03 & 0.92 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.34 & 0.07 & -0.07 & -0.95 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.84 & 0.02 & -0.87 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.37 & 0.37 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.41 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{8}
$$

The $\lambda$ values obtained for the concepts are tabulated in Table 1.

**Table 1.** Found $\lambda_n$ values of transformation function of $C_n$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Consept | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ |
| $\lambda$ Value | 0.17 | 4.41 | 5.00 | 3.58 | 1.41 | 1.11 | 0.88 |
| Consept | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ | $C_{16}$ |
| $\lambda$ Value | 0.88 | 0.09 | 4.31 | 4.84 | 0.00 | 1.11 | 0.97 |

After obtaining the connection matrix W, four different operating conditions are chosen in order to test the performances of the proposed self-tuning PI controller. While the process is operating at steady state value which is 2, the set point values are changed orderly as 1.8, 2.2, 1.9 and 2.1. Hence, there are four different set points and control regions which means that the process should operate for each operating conditions whether it has different dynamic behaviors. FCM determines the PI controller parameters for each new set point value and triggers controller to changes the controller parameters for optimal behavior at recent operating condition and process dynamics. Table 2 shows the optimal PI parameters and the PI parameters obtained by FCM based self-tuning mechanism. System and controller output due to set point changes are illustrated in Fig. 3. As can be seen from Fig. 3, FCM Self-Tuning PI mechanism determines optimal PI controller parameters when set point changes. The controller parameters produced by the proposed FCM is very close to the optimal parameters. Therefore, the simulations show that the tuned PI controller can perform well in new working condition of system.

**Table 2.** Comparison of optimal controller parameters and proposed FCM parameters

| Change in set-point | 2.0-1.8 | | 1.8-2.2 | | 2.2-1.9 | | 1.9-2.1 | |
|---|---|---|---|---|---|---|---|---|
| | $K_P$ | $T_i$ | $K_P$ | $T_i$ | $K_P$ | $T_i$ | $K_P$ | $T_i$ |
| Optimal PI | 0.91 | 5.48 | 3.43 | 3.82 | 0.92 | 5.55 | 1.07 | 4.09 |
| FCM PI | 0.96 | 6.125 | 1.13 | 1.15 | 0.95 | 6.09 | 1.08 | 4.10 |

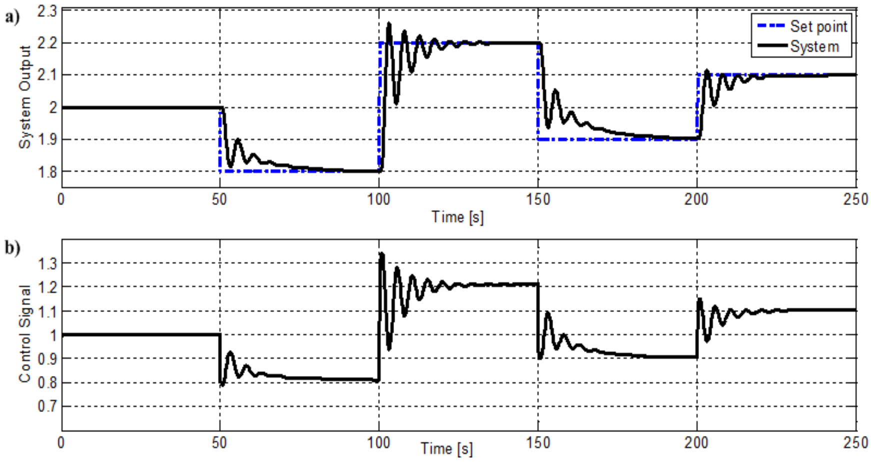


**Fig. 3.** The illustration of (a) system response, (b) control output

## 6 Conclusion

In this study, a novel self-tuning PI controller using fuzzy cognitive map is proposed for the first time in the literature. The proposed FCM mechanism tunes the PI controller parameters using system's current and desired operation point (reference value) when a change in set point occurs. For this study, various simulations are studied and one of the simulations based on a nonlinear system is presented. The obtained results show that proposed FCM adopts the controller according to system's new working points. The system output shows that PI controller performance is fairly for various operation conditions since the proposed FCM represent nonlinear self-tuning strategy efficiently. FCM is a resourceful tool for self-tuning mechanism of a PI controller for nonlinear systems.

For the future work, the self-tuning PI with FCM mechanism will be extended for PID controllers. Then, the proposed method will be implemented on a real-time experimental system.

## References


1. Åström, K.J., Hägglund, T.: The future of PID control. Control Engineering Practice, pp.1163-1175 (2000)
2. Axelrod, R.: Structure of Decision: the Cognitive Maps of Political Elites. Princeton University Press, Princeton, New Jersey (1976)
3. Kosko, B.: Fuzzy cognitive maps. International Journal of Man-Machine Studies, vol. 24, pp. 65–75 (1986)
4. Aguilar, J.: A survey about fuzzy cognitive maps papers. International Journal of Computational Cognition, vol. 3, no. 2, pp. 27–33 (2005)
5. Papageorgiou, E. I.: Review study on fuzzy cognitive maps and their applications during the last decade. in Proceedings of the 2011 IEEE International Conference on Fuzzy Systems, IEEE Computer Society, Taipei, Taiwan, pp. 828-835 (2011)
6. Papageorgiou, E. I.: Learning algorithms for fuzzy cognitive maps: A review study. IEEE Trans. Syst., Man Cybern. C Appl. Rev., vol. 42, no. 2, pp.150 -163 (2011)
7. Papageorgiou, E. I., Salmeron, J. L.: A Review of Fuzzy Cognitive Map research during the last decade. accepted for publication in IEEE Transactions on Fuzzy Systems, vol. 21, no. 1, pp. 66-79 (2013)
8. Stylios, C. D., Groumpos, P. P.: Modeling Complex Systems Using Fuzzy Cognitive Maps. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 34, no. 1, pp. 155-162, (2004)
9. Papageorgiou, E. I., Stylios, C., Groumpos, P.: Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. International Journal of Human-Computer Studies, vol. 64, pp. 727–743 (2006)
10. Gonzalez, J. L., Aguilar, L. T., Castillo, O. A.: cognitive map and fuzzy inference engine model for online design and self fine-tuning of fuzzy logic controllers, International Journal of Intelligent Systems 24 (11), pp. 1134-1173 (2009)
11. Kottas, T. L., Boutalis, Y. S., Christodoulou, M. A.: Fuzzy Cognitive Networks: Adaptive network estimation and control paradigms, Studies in Fuzziness and Soft Computing 247, pp. 89-134 (2010a)
12. Kottas, T. L. Boutalis, Y. S., Christodoulou, M. A.: Fuzzy Cognitive Networks for Maximum Power Point Tracking in Photovoltaic Arrays Studies in Fuzziness and Soft Computing, vol. 247, 231- 257 (2010b)
13. Beeson, P., Modayil, J., Kuipers, B.: Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy, International Journal of Robotics Research vol. 29, no. 4, pp. 428-459 (2010)
14. Ismael, A., Hussien, B., McLaren, R. W.: Fuzzy neural network implementation of self tuning PID control. In: Proc. IEEE Int. Symp. Intelligent Control, pp. 16 -21 (1994)
15. Khan, M. S., Chong, A.: Fuzzy cognitive map analysis with genetic algorithm. In: Ind. Int. Conf. Artif. Intell. (2003)
16. Parsopoulos, K. E., Papageorgiou, E. I., Groumpos, P. P., Vrahatis, M. N.: A first study of fuzzy cognitive maps learning using particle swarm optimization. In: Proc. IEEE Congr. Evol. Comput, pp. 1440–1447 (2003)
17. Stach, W., Kurgan, L., Pedrycz, W.: A survey of fuzzy cognitive map learning methods In: P. Grzegorzewski, M. Krawczak, and S. Zadrozny, (Eds.), Issues in Soft Computing: Theory and Applications, Exit, pp. 71-84 (2005)
18. Yesil, E., Ozturk, C., Cosardemir, B., Urbas, L.: MATLAB Case-Based Reasoning GUI application for control engineering education, IEEE Int. Conf. Information Technology Based Higher Education and Training (2012)

19. Yurkeyich, V. D.: PI/PID control for nonlinear systems via singular perturbation technique, Advances in PID Control (2011)
20. Anh, H. P. H., Nam, N. T.: A new approach of the online tuning gain scheduling nonlinear PID controller using neural network, PID Control, Implementation and Tuning (2011)
21. Mhaskar, P., El-Farr,a N.H., Christofides, P. D.: A method for PID controller tuning using nonlinear control techniques, American Control Conference, pp. 2925-2930 (2004)
22. Liu, G. P., Daley, S., Optimal-tuning PID control of hydraulic systems, Control Engineering Practice (8), pp. 1045-1053 (2000)
23. He, S. Z., Tan, S., Xu, F. L., Wang, P. Z.: Fuzzy self-tuning of PID controllers, Fuzzy Sets and Systems vol. 56, pp. 37-46 (1993)
24. Yesil E., Guzelkaya M, Eksin I.: Fuzzy logic based tuning of PID controllers for time delay systems, Artificial Intelligence and Soft Computing, pp.236-241 (2006)
25. Soyguder, S., Karakose, M., Ali, H.: Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system, Expert Systems with Applications, vol. 36, no. 3, pp. 4566-4573 (2009)
26. Fang, M.C., Zhuo, Y.Z., Lee, Z.Y.: The application of the self-tuning neural network PID controller on the ship roll reduction in random waves, Ocean Engineering, vol. 37, no. 7, pp 529-538 (2010)
27. Emilia, G.D., Marra, A., Natale, E.: Use of neural networks for quick accurate auto-tuning of PID controller, Robotics and Computer-Integrated Manufacturing, vol. 23, no. 2, pp. 170-179 (2007)
28. Yesil, E., Dodurka, M. F.: Goal-Oriented Decision Support using Big Bang-Big Crunch Learning Based Fuzzy Cognitive Map: An ERP Management Case Study. In: IEEE Int. Conf. Fuzzy Systems (2013)
29. Yesil, E., Urbas, L.: Big Bang - Big Crunch Learning Method for Fuzzy Cognitive Maps, International Conference on Control, Automation and Systems Engineering (2010)
30. Yesil, E., Ozturk, C., Dodurka, M. F., Sakalli, A.: Fuzzy Cognitive Maps Learning Using Artificial Bee Colony Optimization. In: IEEE Int. Conf. Fuzzy Systems (2013)
31. Mudi, R. K., Pal, N.P.: A robust self-tuning scheme for PI- and PD-type fuzzy controllers, IEEE Transactions on Fuzzy Systems, vol. 7, no.1, pp. 2-16 (1999)
32. Erol, O. K., Eksin, I.: A new optimization method: Big Bang–Big Crunch, Advances in Engineering Software, vol. 37, pp. 106–111 (2006)
33. Yesil, E., Urbas L., Demirsoy, A.: FCM-GUI: A graphical user interface for Big Bang-Big Crunch Learning of FCM, Fuzzy Cognitive Maps for Applied Sciences and Engineering – From Fundamentals to Extensions and Learning Algorithms. Ed. Elpiniki Papageorgiou. Vol. Intelligent Systems Reference Library. Springer (2013)