Self-Organizing Adaptive Network-based Fuzzy Intelligent Controller (SOANFIC)

Hamid Reza Saghir
Department of Mechanical Engineering,
Islamic Azad University, Shiraz Branch, Young Researchers Club,
Shiraz, Iran
E-mail: Fight4bluesky@gmail.com

Saeed Bagheri Shouraki
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
E-mail: bagheri-s@sharif.edu

Abstract— **Self-Organizing controller is a very good tool for providing a system with adaptive abilities. However there are some problems that are drawbacks of this controller. Firstly, although the process model needed for an SOC is an incremental one, it is still difficult to find and calculate such models in practice, and furthermore if an SOC is to be used, there is always a need for the knowledge of time lags in the process under control. The second problem is that in an SOC the process of updating the rules, assumes that there is only one sample in the past that is responsible for the current poor performance, while in fact more than one sample may be responsible and require correction. And the third problem is that if a fuzzy controller is used in the lower level of an SOC, the process of updating control rules would be a very demanding task that ends up in replacement of each rule with three new rules for each sample. So in addition to the need for a garbage collection algorithm to eliminate the redundant rules, the task of correction would be a time consuming process. To overcome these problems we introduce SOANFIC that is a model free controller and solves these problems. In addition to this SOANFIC is a very easily applicable controller that can be applied to MIMO systems as well as SISO ones.**

Keywords-SOC, Fuzzy controller, ANFIS, Intelligent controller…

I. INTRODUCTION

During the last few years, the theory of Self-Organizing Systems has spread over various fields of science. It is applied in Computer Science, Mathematics, Physics, Chemistry, Biology and even Psychology. The self-organizing control (SOC) technique has been developed and investigated over the past decade by Mamdani and his associates [1-3]. Form that time different researches has been done on this topic [4-6]

In those days, the distinction between the terms *Self-Organizing* and *adaptive* was unclear, but today the SOC will be categorized under adaptive controllers; broadly defined, an adaptive controller is a controller with adjustable parameters and a mechanism for adjusting the parameters. Actually, judging from the block diagram of the SOC, it may even be sub-categorized as a model-reference adaptive system; this is an adaptive system in which the performance specifications are given by a model. The model tells how the plant output ideally should respond to the control signal. The SOC uses a desired response, rather than a model of the plant, and it adjusts its parameters according to a predefined performance measure.

This need for some sort of a process model in the SOC has somehow limited its usages. The goal here is to develop a controller based on the concept of self-organization that can bypass these limitations and perform the tasks of system identification and control in a more powerful fashion. This work provides a powerful intelligent control technique that can be applied to many SISO and MIMO processes that include uncertainties.

II. SELF-ORGANIZING CONTROLLER

The SOC has a hierarchical structure in which the lower level is a table-based or a fuzzy controller and the higher level is the adjustment mechanism.

Lower level, At the lower level there is an incremental controller, where the control signal is added to the previous control signal, modeled as an integrator in the figure. The two inputs to the controller are the error e and the change in error ce. These are multiplied by two gains, GE and GCE respectively, before the rule base in **F**.

Higher level, The idea behind self-organization is to let an adjustment mechanism update the values in table **F**, based on the current performance of the controller. Basically, if the performance is poor, the responsible table value or fired rules should be punished, such that next time that cell of the table is visited, the control signal will be better.

The higher level monitors error and change in error and it modifies the table **F** through a modifier algorithm **M** when necessary.

A. Credit assignment

To apply the reinforcement, we have to translate the changes in outputs of a system into the changes in its inputs. To do that, three main questions should be answered:

1. How should we calculate the input deviations from the knowledge of output deviations?
2. If the process is a multivariable one, which input should be corrected and by how much?
3. Which previous control actions has led to current poor performance, i.e. which samples in the past should be punished?

The answer to these questions comes from the knowledge of process's dynamics. Therefore, there is a need for an incremental model of the system. For example in the case of a two-input two-output system:

$$\dot{X} = F(X, U, V)$$
$$\dot{Y} = G(Y, U, V)$$

For small changes in input, the output derivative will be:

$$\begin{pmatrix} \delta\dot{X} \\ \delta\dot{Y} \end{pmatrix} = \begin{pmatrix} \partial F/\partial U & \partial F/\partial V \\ \partial G/\partial U & \partial G/\partial V \end{pmatrix} \begin{pmatrix} \delta U \\ \delta V \end{pmatrix}$$

After a sample time, T:

$$\begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix} \approx \begin{pmatrix} T\delta\dot{X} \\ T\delta\dot{Y} \end{pmatrix} = TJ \begin{pmatrix} \Delta U \\ \Delta V \end{pmatrix} = M \begin{pmatrix} \Delta U \\ \Delta V \end{pmatrix}$$

Where J is the Jacobian matrix and the matrix M can be regarded as the incremental model of the process. Therefore, if the corrections required at the output of the system were $p_1(nT)$ and $p_2(nT)$ then the necessary input corrections $r_1(nT)$ and $r_2(nT)$ would be:

$$\begin{pmatrix} r_1(nT) \\ r_2(nT) \end{pmatrix} = M^{-1} \begin{pmatrix} p_1(nT) \\ p_2(nT) \end{pmatrix}$$

As it was seen, the point about the SOC is that it needs some knowledge of the system's dynamics, which should be presented to the controller through the modifier algorithm. It also needs some information about the time lags in the plant to be able to apply the reinforcements in the correct order. For nonlinear and non-monotonic processes, the matrix M would be dependent on the process states too. Therefore, it is usually not easy to determine the matrix M.

B. *Performance measure*

An adaptive or learning controller needs some form of a measure to be able to assess its own performance. Generally, there are two types of performance measures available:

- Global criterion
- Local criterion

The difficulty with using a global criterion like integral square error is firstly to choose an appropriate criterion and secondly to relate a change in this figure of merit to a set of control actions. To overcome this, local criteria are chosen. However, proper local performance measures are also still difficult to devise.

Considering these difficulties Procyk and Mamdani [1] suggested another method for assessment of the performance of a system based on the premise that a control engineer (or whoever the control system is to serve) has a definite picture in mind of the type of response from the plant he is willing to tolerate.

According to this they developed a table-based performance measure based on the fact that the response of an output can be conveniently monitored by its error, *e(nT),* and change in error, *c(nT).* The performance measure then takes the form of a decision maker which issues the output correction required from a knowledge of *e(nT)* and *c(nT).*

		CE												
		−6	−5	−4	−3	−2	−1	0	1	2	3	4	5	6
	−6	−6	−6	−6	−6	−6	−6	−6	0	0	0	0	0	0
	−5	−6	−6	−6	−6	−6	−6	−6	−3	−2	−2	0	0	0
	−4	−6	−6	−6	−6	−6	−6	−6	−5	−4	−2	0	0	0
	−3	−6	−5	−5	−4	−4	−4	−4	−3	−2	0	0	0	0
	−2	−6	−5	−4	−3	−2	−2	−2	0	0	0	0	0	0
	−1	−5	−4	−3	−2	−1	−1	−1	0	0	0	0	0	0
E	0	−4	−3	−2	−1	0	0	0	0	0	1	2	3	4
	1	0	0	0	0	0	0	1	1	1	2	3	4	5
	2	0	0	0	0	0	0	2	2	2	3	4	5	6
	3	0	0	0	0	2	3	4	4	4	4	5	5	6
	4	0	0	0	2	4	5	6	6	6	6	6	6	6
	5	0	0	0	2	2	3	6	6	6	6	6	6	6
	6	0	0	0	0	0	0	6	6	6	6	6	6	6

Figure 1. Procyk and Mamdani's Perf. Table

The table in figure 2 is the one introduced by Procyk and Mamdani. There are also some other slightly different tables introduced by some other researchers in the literature.

The original performance tables were built by hand, based on trial and error. Presumably, if the numbers in the table are small, it will be necessary to do many updates to reach useful control rules; if the numbers are large, the convergence will be faster, but maybe also unstable.

The following analysis leads to a deeper understanding of the mechanism. It seems that the Procyk & Mamdani' s table (Fig. 2) holds the zeros in a more or less diagonal band. Because the zeros indicate no penalty, those states must be admissible.

Focusing on the zero diagonal, it expresses the relation:

$$GE \times e + GCE \times \frac{de}{dt} = 0$$

This is an ordinary differential equation, and can be solved as:

$$e(t) = e(0)\exp\left(-\frac{t}{\frac{GCE}{GE}}\right)$$

In other words, a first order exponential decay with a time constant GCE/GE; assuming e(0)=1 the error e will gradually die out, and after t=GCE/GE seconds it has dropped 0.63 units. To interpret, the modifier **M** in an original model based SOC tries to push the system towards a first order transient response.

Instead of a performance table a simple penalty equation can be used:

$$\Delta P = G_p(e_n + \tau \times ce_n) \times T_s$$

The learning rate G_p affects the convergence rate and T_s is the sample period. This penalty equation is an incremental one because the output is a change to an existing value. In order to keep the update rate independent of the choice of sample period the penalty equation is multiplied by T_s.

III. Adaptive Network Based Fuzzy Inference System (ANFIS)

An adaptive network, as its name implies, is a network structure consisting of nodes and directional links through which the nodes are connected. Moreover, part or all of the nodes are adaptive, which means their outputs depend on the parameter(s) pertaining to these nodes, and the learning rule specifies how these parameters should be changed to minimize a prescribed error measure.

In an ANFIS this adaptive network models the structure of a fuzzy system. In the present work a TSK fuzzy system is the subject because of more simplicity.

Layer 1: every node in this layer can be an adaptive one and with the node function:

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

With μ being the membership functions for example:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}}$$

Or,

$$\mu_{A_i}(x) = exp\left\{-\left(\frac{x - c_i}{a_i}\right)^2\right\}$$

Layer 2: every node in this layer are fixed ones that multiply the inputs and send the output out or in other words each node output represents the firing strength of a rule:

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), i = 1,2$$

Layer 3: each node in this layer is a fixed node, which calculates the ratio of the firing strength of each rule to all rule's firing strengths:

$$\overline{w_i} = \frac{w_i}{w_1 + w_2}, i = 1,2$$

Layer 4: each node in this layer can be an adaptive node which constructs the consequent part of a fuzzy if-then rule:

$$O_i^4 = \overline{w_i}\, f_i = \overline{w_i}\,(p_i x + q_i y + r_i)$$

Layer 5: the nodes in this layer are fixed ones that calculate the total outputs of the network as the summation of all incoming signals:

$$O_i^5 = overall\ output = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

IV. SOANFIC

The self-organizing controller is a very useful adaptive tool; however, it has several problems:

1. Although the model needed for an SOC is an incremental one, it is still difficult to find and calculate such models in practice, and furthermore if an SOC is to be used, there is always a need for the knowledge of time lags in the process.
2. In an SOC the process of updating the rules, assumes that there is only one sample in the past that is responsible for the current poor performance, while in fact more than one sample may be responsible and require correction.
3. If a fuzzy controller is used in the lower level of an SOC, the process of updating control rules would be a very demanding task that ends up in replacement of each rule with three new rules for each sample. So in addition to the need for a garbage collection algorithm to eliminate the redundant rules, the task of correction would be a time consuming process.

To solve these problems we took several steps that are explained below:

A. Step 1:

This credit assignment and the concept of performance measures can be used with any type of controller in the lower level of an SOC, but there are two reasons for using a fuzzy controller in the lower hierarchical level:

1. The controller consists of a set of linguistic rules, which lend themselves to simple manipulation.
2. The controller can always be primed with a set of initial crude set of control rules obtained from experience

Considering only one sample to be responsible for poor performance, SOC modifies the fuzzy relations in that sample to apply the correction. This contains the extraction of the bad rule and insertion of the new reinforced rule. This process is a demanding task, which is explained in [1].

To benefit from these two advantages, we need a fuzzy controller in the lower level, but also in order to remove the third aforementioned problem, we can use a TSK fuzzy system instead of a normal one. TSK fuzzy systems benefit from providing an output as a linear combination of the inputs. Therefore, the modification of the rules would be much easier if we use TSK fuzzy systems.

The format of rules in a fuzzy TSK system is as follows:

If (x_1 is A_1) and … and (x_k is A_k) then ($y=p_0+p_1x_1+\ldots+p_kx_k$)

B. Step 2:

This modified version still has the first two problems, to solve the first problem we need to eliminate the need for the modifier algorithm, i.e. make this controller a model free one that performs a more powerful system identification. To do that, we need to plug some sort of learning into this system.

C. Step 3:

To solve the second problem we have to make the process of correction, an adaptive process that would adjust itself to the conditions of the plant under control. This means that there needs to be some sorts of learning in the process of correction too. Basically, an adaptive network can provide such abilities due to its inherent learning and adaptive nature. Therefore, according to what has been said, we need a lower level controller that has the following characteristics:

- Fuzzy
- Model free
- Network-based and adaptive

Therefore, the answer to our problem is an adaptive network-based fuzzy inference system, or shortly ANFIS (figure 2). This controller is called the self-organizing adaptive network based fuzzy intelligent controller or shortly SOANFIC.

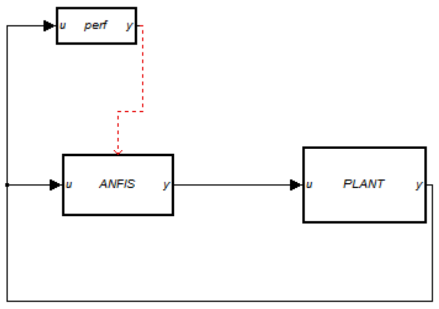

Figure 2. SOANFIC Structure

D. Learning in SOANFIC:

In supervised learning methods, to construct the network, there is always a target or a desired value available for each dataset member that the learning algorithms use. In SOANFIC, we are actually providing the desired values through a performance measure as it was discussed before. This performance measure provides a number at each step, explaining how satisfactory the response is, and not the desired response itself. This means that we need to modify the learning method in a way that eliminates the need for the desired response and replaces it with the knowledge of how satisfactory the response is; i.e. performance signal.

On the other hand, ANFIS and generally networks have two types of learning, offline or batch learning and online learning. In the offline learning mode, a dataset need to be provided for the network so that it can adapt its parameters to the dataset's pattern, but in the online mode there is no such dataset available and data is provided through time. SOANFIC has to use the online learning method because of how the performance data is provided.

In an ANFIS the performance signals in different samples act as the entries of a data-set in an online learning task. Through back-propagation the ANFIS adapts the parameters inside its network in a way to make the reinforcement signal zero.

There are several learning algorithms available for ANFIS but Gradient Descent is a suitable candidate due to the fact that in SOANFIC we don't have the desired values and instead, a measure of performance is provided. The objective function for Gradient Decent is the performance table or the penalty function derived from such table:

$$\Delta P = G_p(e_n + \tau \times ce_n) \times T_s$$

In normal back propagation, the error signal at the output layer of the network is:

$$E_p = \frac{1}{2}(Desired - Output)^2$$

But we know that:

$$(Output) + (Perfomance\ signal) = Desired$$

So:

$$E_p = \frac{1}{2}(O + P - O)^2 = \frac{P^2}{2}$$

Where O and P are the output and the performance signals respectively. Thus, the derivative of the error function is obtained as:

$$\frac{\partial E_p}{\partial O_n^5} = -2(target - output) = -p$$

Using the chain rule:

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \cdot \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k}$$

This gives the derivative of layer 4 of the five layer ANFIS system as:

$$\frac{\partial E}{\partial O_1^4} = \frac{\partial E}{\partial O^5} \cdot \frac{\partial O^5}{\partial O_1^4} = -p \cdot \frac{\partial(\sum_{i=1}^{2} \overline{w}_i \cdot f_i)}{\partial(\overline{w}_1 \cdot f_1)} = -p$$

That gives :

$$\frac{\partial E}{\partial O_1^4} = -p\,, \quad \frac{\partial E}{\partial O_2^4} = -p$$

The derivatives with respect to adaptive parameters and other layers are also calculated the same way from the chain rule.

V. THE INVERTED PENDULUM

The inverted pendulum [10, 11 and 12] system is a standard problem in the area of control systems. The system is nonlinear so it is useful in illustrating some ideas in the nonlinear control field.

The inverted pendulum system inherently has two equilibria, one of which is stable while the other is unstable. The unstable equilibrium corresponds to a state in which the pendulum points strictly upwards and, thus, requires a control force to maintain this position.

By drawing the free body diagram of the system (figure 3) and doing a force and momentum analysis the system equation setup can be derived:

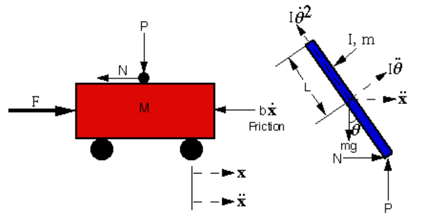

Figure 3. FBD of the System

the system's equation set in state space form is derived as:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{U + (ml\sin(x_3))x_4^{\,2} - bx_2 - \dfrac{(ml\cos(x_3))(mgl\sin(x_3))}{I + ml^2}}{(M + m) - \dfrac{(ml\cos(x_3))^2}{I + ml^2}}$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{U + (ml\sin(x_3))x_3^{\,2} - bx_2 - g\tan(x_3)}{ml\cos(x_3) - \dfrac{(M + m)(I + ml^2)}{ml\cos(x_3)}}$$

VI. SIMULATION RESULTS:

Parameters values used in simulation are:

M	mass of the cart	2 kg
m	mass of the pendulum	0.2 kg
b	friction of the cart	0 N/m/sec
l	length to pendulum center of mass	0.3 m
I	inertia of the pendulum	0.003 kg*m^2

The simulation starts from the scratch and the number of rules in the lower level ANFIS is considered to be 25, but it may present better results if this number is tuned. One can find an optimal response by further tuning of the parameters in SOANFIC.

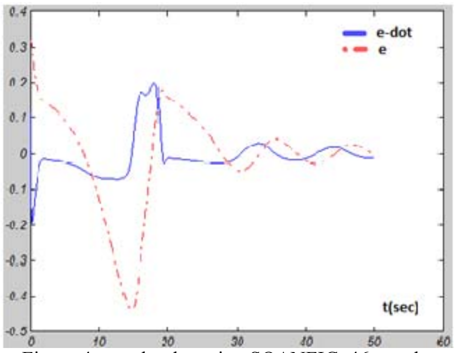

Figure 4. e and e-dot using SOANFIC–46 epochs

This controller can first stabilize the plant after 46 epochs as it is shown in figure 4. The response is not satisfactory yet, thus, the simulation continues.

As it is shown in figure 5, SOANFIC can stabilize the plant in a rather short time without any previous information about the plant.

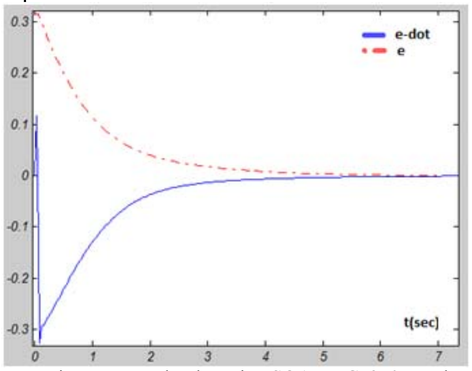

Figure 5. e and e-dot using SOANFIC–250 epochs

VII. CONCLUSION

SOANFIC is developed based on some ideas from Self-Organizing Controllers and Network-based Fuzzy systems. It was discussed that SOCs have three main drawbacks:

- Although the model needed for an SOC is an incremental one, it is still difficult to find and calculate such models in practice, and furthermore if an SOC is to be used, there is always a need for the knowledge of time lags in the process under control.
- In an SOC the process of updating the rules, assumes that there is only one sample in the past that is responsible for the current poor performance, while in fact more than one sample may be responsible and require correction.
- If a fuzzy controller is used in the lower level of an SOC, the process of updating control rules would be a very demanding task that ends up in replacement of each rule with three new rules for each sample. So in addition to the need for a garbage collection algorithm to eliminate the redundant rules, the task of correction would be a time consuming process.

We showed that these problems of SOCs are solved in SOANFIC. Furthermore, to test the applicability of SOANFIC, the nonlinear model of an inverted pendulum was used as a test-bed. SOANFIC showed that it can stabilize the system while starting from scratch and that it can provide a satisfactory response.

REFERENCES

[1] Procyk T J, Mamdani EH, "a *Linguistic Self Organizing Process Controller*", Automatica, Vol.15, No.1, Pp. 15-30, 1979

[2] YAMAZAKI, T, and MAMDANI, E.H.: 'On the performance of a rule-based self-organizing controller'. Proceedings of IEEE conference on applications of adaptive & multivariable control, Hull, UK, 1982

[3] ASSILIAN, S., and MAMDANI, E.H.: 'An experiment in linguistic synthesis with a fuzzy logic controller', *Int. J. Man-Mach. Stud.,* 1974, 7, pp. 1-13

[4] Jantzen Jan, *"The Self-Organising Fuzzy Controller",* Technical University of Denmark, Department of Automation, Bldg 326, DK-2800 Lyngby, DENMARK. Tech. report no 98-H 869 (soc), 19 Aug 1998.

[5] Jin-Maun Ho, Shoou-Ren Lin, "A Learninc Algorithm For Fuzz!' Sclf-Organizing Controller", IEEE International Workshop on Intelligent Motion Control, Bogaziqi University, Istanbul, 20-22 August 1990.

[6] Sang-Ho So, Dong-Jo Park, "Design of Gradient Descent Based Self-organizing fuzzy Logic Controller with Dual Outputs", 1999 IEEE International Fuzzy Systems Conference Proceedings *August* 22-25, *1999, Seoul, Korea*

[7] Jang J S R, "*ANFIS: Adaptive Network Based Fuzzy Inference System*", IEEE Transactions, Man & Cybernetics 1991

[8] Jyh-Shing R. Jang, *"Self-Learning Fuzzy Controllers Based on Temporal Back Propagation*", IEEE Transactions on Neural Networks, Sep 1992.

[9] J.-S. Roger Jang, "*Fuzzy Controller Design without Domain Experts*", IEEE International Conference on Fuzzy Systems, 1992.

[10] Jyh-Shing Roger Jang, Chuen-Tsai Sun, "*Neuro-Fuzzy Modeling and Control*", Proceedings of the IEEE, Mar 1995.

[11] J yh-S hing Roger Jang, "*Input Selection for ANFIS Learning*", Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 1996.

[12] R.J. Stonier, A.J. Stacey, C. Messom, "*Learning fuzzy control laws for the inverted pendulum*",ISCA 98

[13] Jan Jantzen, "*Analysis Of A Pendulum Problem*", Technical University of Denmark, Department of Automation, Bldg 326, DK-2800 Lyngby, DENMARK.Tech. report no 98-E 863 (cartball), 19 Aug 1998.

[14] Http://Www.Engin.Umich.Edu/Group/Ctm/Examples/Pend/Invpen.Html. Example: Modeling An Inverted Pendulum.