World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:4, No:3, 2010

Design Neural Network Controller for Mechatronic System

Ismail Algelli Sassi Ehtiwesh, and Mohamed Ali Elhaj

Abstract**—**The main goal of the study is to analyze all relevant properties of the electro hydraulic systems and based on that to make a proper choice of the neural network control strategy that may be used for the control of the mechatronic system.

A combination of electronic and hydraulic systems is widely used since it combines the advantages of both. Hydraulic systems are widely spread because of their properties as accuracy, flexibility, high horsepower-to-weight ratio, fast starting, stopping and reversal with smoothness and precision, and simplicity of operations. On the other hand, the modern control of hydraulic systems is based on control of the circuit fed to the inductive solenoid that controls the position of the hydraulic valve. Since this circuit may be easily handled by PWM (Pulse Width Modulation) signal with a proper frequency, the combination of electrical and hydraulic systems became very fruitful and usable in specific areas as airplane and military industry.

The study shows and discusses the experimental results obtained by the control strategy of neural network control using MATLAB and SIMULINK [1]. Finally, the special attention was paid to the possibility of neuro-controller design and its application to control of electro-hydraulic systems and to make comparative with other kinds of control.

Keywords**—**Neural-Network controller; Mechatronic; electro-hydraulic

I. INTRODUCTION

THE widespread use of hydraulic circuitry in machine tool applications, aircraft control systems, and similar operations occurs because of such factors as positiveness, accuracy, flexibility, high horsepower-to-weight ratio, fast starting, stopping, and reversal with smoothness and precision, and simplicity of operations.

The operating pressure in hydraulic systems is somewhere between 145 and 500 *lb/in²* (between 1 and 35 MPa) [2]. In some special applications, the operating pressure may go up to 10,000 *lb/in²* (70 MPa). For the same power requirement, the weight and size of the hydraulic unit can be made smaller by increasing the supply pressure. With high pressure hydraulic systems, very large force can be obtained. Rapid-acting, accurate positioning of heavy loads is possible with hydraulic systems. A combination of electronic and hydraulic systems is widely used because it combines the advantages of both electronic control and hydraulic power. Moreover, the operating conditions of, and the disturbance acting on, hydraulic systems vary in a complicated fashion; for instance the valve, oil and load parameters may vary significantly.

Normally these parameters are not precisely known or time-variant for a great variety of reasons, *e.g.*, temperature-dependent behavior. All these properties and facts make the control design and tuning difficult. The main objectives for closed-loop control of hydraulic servo-systems are [3]:

- Linearized input-output behavior, which is consistent over the whole operating range.
- Sufficient damping in order to get better step response.
- Control bandwidth improvement, as much as allowed by the dynamics of the hydraulic system and the robust stability requirements imposed by unmodelled dynamics, as well as by parameter variations and disturbances.
- The size of the mechanical components and the flow rates should be kept at least unchanged.

An ideal controller would thus be robust against parameter and disturbance variations, and lead to best performance simultaneously. In practice, however, a *trade-off* has to be decided depending on the application at hand.

Many industrial controllers for an HSS achieve high bandwidth with fixed gain control laws by over-sizing the cylinder diameter in order to increase the effective stiffness of the fluid in the cylinder. This requires larger and more costly components and higher fluid flow rates in order to move a load at a given speed. A better approach to obtaining a fast response is to model the dominant dynamics of the system, and then to use an approach is that, to achieve a given bandwidth, the mechanical components are smaller, the required flow rates are less, and the overall system is therefore much less expensive.

The main purpose of the study is to analyze the most relevant properties of the electro-hydraulic servo system shown in the Figure 1, and to make an analysis of the control strategies that may be used for the control of these types of servomechanism. The idea is to evaluate, through detailed simulation, neural network controllers that may be used for this purpose.

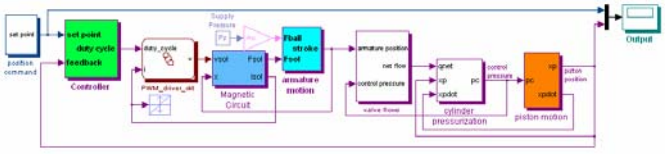

Fig. 1 Electro-hydraulic servomechanism

II. NEURAL NETWORK BASED CONTROL

The neural network based control technique has represented an alternative method to solve the problems in control

Ismail Algelli Sassi Ehtiwesh is with the Department of Mechanical Engineering – 7th of April University, Surman, Libya (ismaeil_ehtiwesh@yahoo.com)

Mohamed Ali Elhaj is with the Department of Electrical Engineering - 7th of April University, Tripoli, Libya (mohamedelhaj@hotmail.com).

engineering. The most useful property of neural networks in control is their ability to approximate arbitrary linear or nonlinear mapping through learning. It is because of the above property that many neural network based controllers have been developed for the compensation for the effects of nonlinearities and system uncertainties in control systems so that the system performance such as the stability and robustness can be improved.

It can be seen from the recent development of the neural network based control systems that, by suitably choosing neural network structures, training methods, and sufficient past input and output data, the neural networks can be well trained to learn the system forward dynamics to predict the future behavior of the systems for the predictive control and model control. A neural network derives its computing power through, first, its massively parallel distributed structure and, second, its ability to learn and therefore generalize; generalization refers to the neural networks producing reasonable outputs for inputs not encountered during training.

The use of neural networks offers the following useful properties and capabilities:

1- *Nonlinearity:* A neuron is basically a nonlinear device, Consequently, a neural network, made up of an interconnection of neurons.
2- *Input-Output mapping:* A popular paradigm of learning called supervised learning involves the modification of the synaptic weights of the neural network by applying a set of labeled training.
3- *Adaptivity*: Neural networks have a built-incapability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions.
4- *Evidential response*: In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also about the confidence in the direction made. This latter information may be used to reject ambiguous patterns, should they arise, and thereby, improve the classification performance of the network.
5- *Contextual Information:* Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other naturally by a neural network.
6- *Fault tolerance:* A neural network, implemented in hardware form, has the potential to be inherently fault tolerance in the sense that its performance is degraded gracefully under adverse operating conditions.
7- *Uniformity of analysis and design.* Basically, neural networks enjoy universality as information processors. We say this in the sense that the same notation is used in all the domains involving the application of neural network. This feature manifests itself in different way:

- Neurons, in one form or another, represent an ingredient common to all neural networks.
- This commonality makes it possible to share theories and learning algorithms in different applications of neural networks.
- Modular networks can be built through a seamless integration of modules.

Most control design approaches, which can be used for fuzzy control, are also suitable to be applied in neural network based control schemes. There are two fundamentally different approaches to neural control design:

- Direct Control System Design: The controller itself is a neural network.
- Indirect Control Systems Design: A neural network is used as a system model in a more conventional control design.

III. Experimental Results Obtained By Neural Network Controller

To design the neural network controller, the following procedure had to be performed:

- The first step was to collect the relevant signals that may be used for training of neural network. The question was what relevant signals may be used to train network to behave properly and what behavior may be considered as proper. The idea was to force the network to perform similarly as well tuned PID controller.
- The second step was to choose the type of the reference signal fed to the close loop system with tuned PID controller, in order to generate the training set. It was important to design that type of signal that contains and involves steady state properties of the system, but also basic dynamical characteristics had to be involved. So, the signal presented in figure 2 was chosen to represent both requirements. This signal is lasting 15 seconds with sampling period of 20 *ms*, and the first half of the signal is stair case with intention to represent the capabilities of the system to track the constant reference, while the second half of the signal contains sine function to check the possibility of the system to follow the dynamic reference.

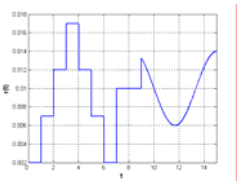

Fig. 2 Reference signal

- The following step was to decide how many and what inputs should be fed to the neural network. The logic was following: if we look at the structures given by figure 3, the error signal $e[kT]$ must be one of the inputs to the neural controller since it is the input to the classical PID [4]. But, PID is a dynamical system and it is capable of generating the integral and first derivative of the error signal, while neural network is a static system without dynamics and it is not able to reconstruct these signals. To avoid this lack of information, it was necessary to define some additional inputs to the neuro-controller. The idea was the second input to be delayed error signal $e[(k-1)T]$ and neural network should be forced to be able to 'reconstruct' the first derivative of the error signal. Finally, instead of the error integral, it is decided the third input to be the output of the system $y[kT]$.

Now, when the nature of neuro-controller inputs and outputs were determined, it became easy to generate the

training set. The reference signal given by figure 2 is fed to the closed-loop system with PID controller given by figure 3 and the signals {*e*[*k*],*e*[*k-1*], *y*[*k*]}, *k*=1,2,…,*N* are collected as training input signals and the signal {*m*[*k*]}, *k*=1,2,..,*N* is collected as training target signal [5].

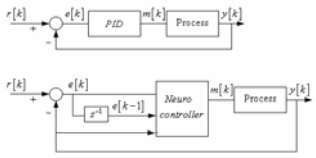

Fig. 3 The structures of closed-loop system

▪ The following step to design the structure of network. The first attempt was to use the simple network with three inputs, one output, and one hidden layer. The activation function for the nodes in the hidden layer was '*tansig*' while the output node was with 'pure line' activation function. Having in mind some experience with the training of neural network, it was decided to use '*Levenberg-Marquard*' algorithm for the back-propagation error method. Finally, it was necessary to choose the number of nodes in the hidden layer. It was clear that the 'optimal' number can be found by 'try and error' approach [6]. In this context *optimal* means the minimum number of nodes that provides satisfied results (good fitting of target signal). So, the first try was with 5 nodes and the obtained result that shows the fitting of obtained and target signal is given by figure 4.

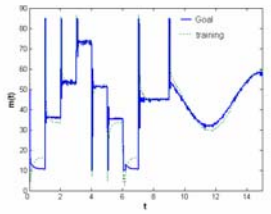

Fig. 4 Obtained and target signal

Of course, it was not possible to get this type of result immediately, since the convergence of mean-square-error depends significantly on the starting point (network initialization). Adopted number of training epochs was 2000. The obtained result presented in figure 4 was not satisfied since there was a significant difference between the target and generated signal. This difference is particularly significant during the steady-state phases of the control sequence *m[k]*. It was obvious that this kind of neuro-controller was not able to generate good results in the closed-loop. In order to make the proper selection of nodes number in the hidden layer, the following experiment has been performed. The number of the nodes was changed from 1 to 20, and for each of these structures the network has been training for enough number of epoch (1000 epochs) and the best results were saved. These results are presented in figure 5. As it is expected, this curve is monotony decreasing, except the value of 14. This irregularity appears because the quality of neural network training is very sensitive to the initial conditions. Initial conditions understand starting point values of the network weights and biases. To overcome this sensitivity, the experiment of network training was repeated 100 times with different initial conditions and after that choosing the best guess. So, this best guess for some number of nodes was really the best one, for some other it was close to the best. Based on the figure 5, it is possible to conclude that significant decreasing of the criterion is obvious for the nodes number less than 6.

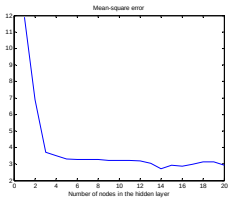

Fig. 5 The mean square error

After that, with further increasing of nodes number, network is trying to learn some details about the controller output and these details cannot improve the behavior of controller significantly. So, it was decided to adopt number of nodes in the hidden layer to be 8, since this value is close to the knee of the characteristics given in figure 5. The following experiment is performed with 8 nodes in the hidden layer, 'tansig' function as activation function of the nodes, three inputs, one output and Levenberg-Marquardt algorithm for back-propagation error network training. The obtained result, showing the fitting between the target signal and network output is presented in figure 6.

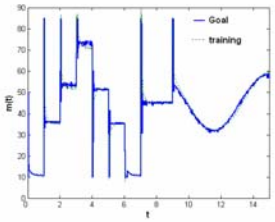

Fig. 6 Target and obtained signals

It became clear that the obtained result was much better and that the difference between what we wanted and what we've got is small enough.

▪ The final step in design of a network was to check if another hidden layer can help. It was decided to introduce another hidden layer, while the number of nodes in the first layer remained 8 and the number of nodes in the second layer was chosen to be 3. The training of this type of network became much more complex and the training process took significantly longer time. The obtained results are given in figure 7, while figure 8 shows the history of mean-square-error during the process of training. The figure 8 shows that the mean-square error after 5000 epochs became 3.34. That means that the relative deviation of the obtained control respect to desirable one is $\sqrt{3.34}/100 = 1.82\%$. This result is pretty acceptable, especially looking at the figure 7 where very good fitting of desired controller output is obtained.

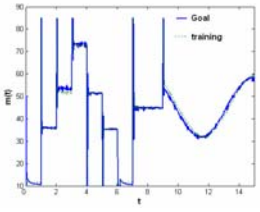

Fig. 7 Target and obtained results

In order to check if the increasing of number of nodes in the second hidden layer can improve the performance of neuro-controller, we again made the experiment where the number of nodes in the first hidden layer was kept to be 8 and we changed the number of nodes in the second hidden layer starting from 1 and ending in 10.

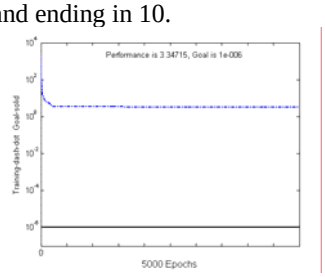

Fig. 8 Mean square error

- Based on that it was clear that the proper structure giving the satisfactory result is a network with 3+8+5+1 structure *(3 input nodes, 8 nodes is first hidden layer, 5 nodes in second hidden layer and 1 node in output layer)*. The results obtained after training of this type of network are presented in figures 9 and 10. Figure 10 shows high quality fitting of the desired controller output in both cases: when the reference is constant and when it is time dependent. Of course, it does not mean that the network really understood what kind of behavior it should perform. It only means that it was able to repeat the behavior contained in the training set. In order to check the property of 'generalization' that the network should posses, some other experiments had to be performed.

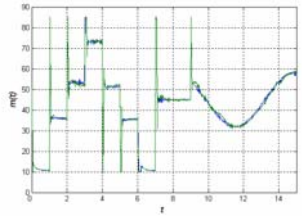

Fig. 9 Target and obtained results

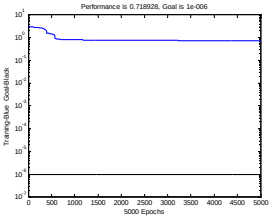

Fig. 10 Mean square error

The figure 10 tells us that the final mean-square-error was 0.71. In other words, relative deviation of the obtained control respect to desirable one is $\sqrt{0.71}/100 = 0.84\%$. This result is quite acceptable. So one can be pleased with the obtained training process, and some analysis of the closed-loop system with neuro-controller had to be performed. Now, when the neuro-controller is designed, it becomes interesting to check what quality of control-loop regulation is possible to expect. The first and most logical test was to prepare neuro-controller in Simulink and to see what output will be obtained if the reference signals is identical as the signal used for neural network training. Figure 11 represents the structure of network prepared in Simulink layout. Since the structure of this controller became complex (not in the numerical sense because the controller performed the number of simple arithmetical operations but the number of these operations became huge) it was necessary to organize simple transfer of the numerical values for each of the network weights and biases. Having in mind the number of nodes in the each of the layers, the number of parameters that had to be fed to the network became:3×8+8×3+5×1=69 for weights and 8+5+1=14 for biases. So 69+14=83 saved parameters were transferred from the disk to the Simulink using by the protocol called 'From Disk to Simulink' available in this programming package.

Now, when the structure of neuro-controller was selected and corresponding parameters were calculated, it was possible to make the close-loop system with the neuro-controller in line. This part of the system that shows the connection of the controller with other parts is given in figure 12. Using neural network controller in the closed-loop electro-hydraulic servo system, with the reference signal used for neural network training gives the output of the system shown in figure 13.

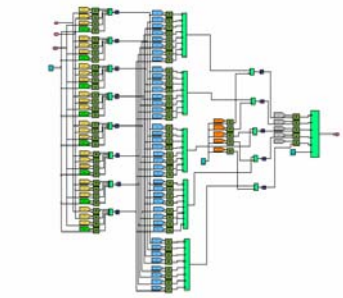

Fig. 11 Structure of neural network controller

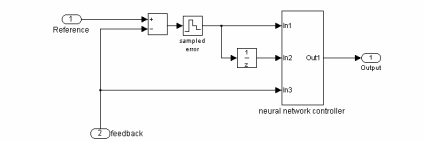

Fig. 12 Neural network controller

Figure 13 also shows the giving reference and the output of the system when well tuned PID is used as a controller. One can conclude that the fitting of the different controllers is almost perfect. The difference between these two outputs is smaller than 1% of the reference. The following experiment had to check the capability of the network to control the system for some other type of reference signal. Making some changes in the input as shown in figure 14, the same experiment has been performed.

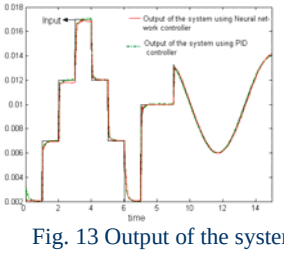

Fig. 13 Output of the system

Figure 14 represents the changed reference signal, the system output obtained by using PID controller and neuro-

controller. In this case the difference between these two outputs becomes obvious. The PID controller showed better performance especially in the regions marked on this figure. The reason for that is simple: the network was not trained for the reference with values between 0.002 and 0.004, and it is able only to interpolate the expected output. To check this reasoning, it was decided to extend the training reference signal and to make a mixture of the references given by figure 14 and figure 13 and to repeat the training procedure.

New reference signal is presented in figure 15. Now, the reference was two times longer and it was logical the training process to last at least two times more. The training process was very successful and the obtained results are given in figures 16 and 17. Figure 16 presents the training control sequence and the output of the network, while figure 17 presents the reference, output of the system with PID controller and output of the system with neuro-controller.

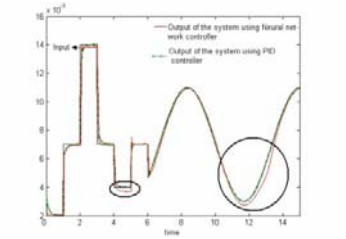

Fig. 14 Output of the system

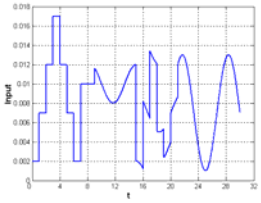

Fig. 15 Training reference signal

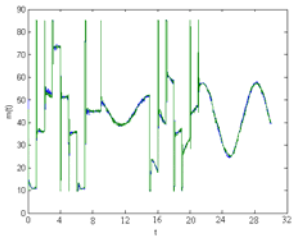

Fig. 16 Training output comparing with the goal

Since the difference between these three signals is almost negligible, the figure 18 presents the difference (error signal) between the given reference and output signal obtained with neuro-controller. The noticeable peaks in this signal have a source in the sharp jumps in reference signal.

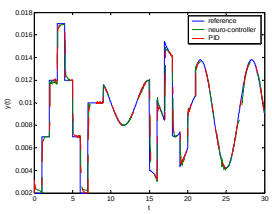

Fig. 17 Output of the system

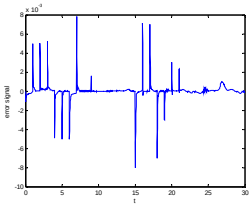

Fig. 18 Error signal for the system

IV. Conclusion

Finally, the special attention was paid to the possibility of neuro-controller design and its application to control of electro-hydraulic systems. Several questions had to be answered in order to design the proper neuro-controller. The first important question was what should be the inputs to such controller. Making several 'try and error' attempts, the answer was that the controller should have the information about the error signal, delayed error signal and system output (piston position). After that it was necessary to decide about the number of hidden layers, the number of nodes in the layers, the activation functions in the nodes and the algorithm for network training. Few analyses regarding the number of nodes in the layers have been accomplished giving the pretty clear answer that the network should contain two hidden layers with eight nodes in the first and five nodes in the second layer.

These answers have been obtained after the long systematic and tedious experiments, where the number of nodes have been changed and checking the actual criteria. The adopted criteria for network training quality were the computed mean square error between the network output and the desired output signal.

Also, the activation function of 'tansig' type was selected and Levenberg-Marquardt back-propagation algorithm for network training. Another important question was how to design the training set for network training. The most simple and most logical choice was to push the network to behave similar as PID controller. So, with the proper reference signal, the output of the PID controller was saved and used as a training (desired) set for the procedure of neural network training. The obtained results were promising, since the training procedure resulted in the mean square error less than 0.8%. In other words, it seemed that the network learnt to behave very similar as classical feedback controller.

Two additional analyses were accomplished. The first one was to check if the network is able to preserve good behavior even if the reference signal is changed. It was concluded that the performances of the closed-loop system is changed in that case, not significantly but noticeable. And it was clear, if the reference signal used for the training of neural network is more reach and longer, although the training procedure takes more time, the quality of regulation becomes improved. The other important performed analysis was related to the appearance of disturbance.

References

[1] K. Singh and G. Agnibori. System Design Through Matlab, Control toolbox and Simulink. Springer, 2000.

[2] Jelali and Kroll. Hydraulic Servo-systems, Modeling, Identification and Control. Springer, UK, 2002.
[3] H. E. Merritt. Hydraulic Control Systems. John Wiley&Sonns, USA, 1967.
[4] K Ogata. Modern Control Engineering. Aeeizh, USA, 2002.
[5] D. Pham and L. Xing. Neural Networks for Identification, Prediction and control. Springer, 1997.
[6] Simon Haykin. Neural Networks. Macmillan, USA, 1994.
[7] K. Astrom and B. Wittenmark. Computer controlled Systems. Prentice-Hall, 1991.
[8] P. Wasserman. Neural computing Theory and Practice. Van Nostrand Reinhold, New York 1991.