

Neuro-Adaptive Robust Control Configurations Based on Variable Structure Control

Marco Antonio Alves¹, Eurípedes Nóbrega²

¹T-Systems do Brasil Ltda

²Departamento de Mecânica Computacional

Faculdade de Engenharia Mecânica

Universidade Estadual de Campinas

E-mails: marco.alves@t-systems.com.br, egon@fem.unicamp.br

Abstract

Among the several control methods developed during the last two decades, robustness has been the most emphasized characteristic in general. Recently, the compromise between robustness and performance has motivated new studies, considering the natural opposition between them. Structure variable control is a distinguished method because of its remarkable simplicity. Despite it has been developed long ago, it has attracted the attention recently due to its inherently robustness, and also because it is equally applied to linear and nonlinear systems. The basic idea is to restrict the state space of a given plant through a so called sliding surface, whose dynamics is simpler than the original plant dynamics. Enforcing a state space trajectory from the initial conditions to reach the surface, once there the plant dynamics is substituted by the surface dynamics. For adequately designed surfaces, they present the invariance property, guaranteeing an intrinsic robustness, because the new dynamics does not depend on the plant parameters. Associating this method to artificial neural networks, some of the proposed configurations may present simultaneously performance and robustness. In this work, a new configuration is proposed, implementing a neuro-adaptive control method using the variable structure approach to adjust the neural network weights, and consequently presenting also robustness. The central idea is to add a second control signal to a regular controller, generated through a neural network, whose online learning is also robust. It is expected that the controller performance will be maintained, independently of perturbations caused by structural or parametric variations. The configuration is explored through three different cases. These cases and also the robust learning technique are presented. Numerical simulations presenting good results justify the expectations for the configuration.

1. Introduction

It is highly desirable for any control method to present high performance and good robustness, but these features are hard to coexist and in general it is necessary to achieve a compromise between them. Two of the most successful control strategies are based on robust and

adaptive methods. Adaptive control and robust control methods are complementary on dealing with model uncertainty. The adaptive approaches uses online identification to adjust the controller to plant model errors, and is suitable to a wide range of parameter variations. Robust control methods aims to make the system insensitive to uncertainties, but they present in general a fixed-structure controller. Recently, hybrid configurations using both approaches have been presented as capable of presenting the best characteristics of each one. A new hybrid configuration using an adaptive neural network in a robust double loop is proposed here, intended to perform well also to nonlinear plant control.

Structure variable control (SVC) is probably the most successful adaptive method whose robustness has been studied during the last two decades (DeCarlo et al, 1988) (Slotine and Li, 1991) (Hung et al, 1993). It is based on the design of a restraining sliding-mode surface, reducing the order of the plant dynamics and indeed substituting it, but causing chattering to the control input due to the switching to remain at the predefined state-space subspace. Another weaknesses of the conventional SVC is the necessary assumption of known uncertainty bounds (Yu and Lloyd, 1997). Nevertheless, the switching mode presents invariance properties to the plant model uncertainties, making SVC a good option to control uncertain nonlinear systems.

To overcome some of these limitations, several authors (Sun et al, 2000) (Jung and Hsia, 2000) (Topalov and Kaynak, 2001) (Barambones and Etxebarria, 2002) adopted as a design philosophy to approximate the uncertain nonlinearity with a neural network and to compensate the approximation errors and external disturbances with a robust controller. Using artificial neural networks (ANN) to improve SVC is a good choice, since it is a well-known fact that they can approximate nonlinear mappings to any desired accuracy (Funahashi, 1989). The ANN modules are used to learn highly nonlinear functions representing the plant direct or inverse dynamics, or some other desired function. The ANN may be trained offline, presenting a fixed input-output relation, as in Haykin (1998). When this capability is otherwise explored online, it is appropriately called an adaptive artificial neural network (AANN). Barambones and Etxebarria, (2000) present a

configuration to control a generic robot, implementing an AANN-based feedback linearization and a robust sliding control to compensate for the neural approximation error. Robotic manipulators are hard to control nonlinear systems, with time-varying inertia and gravitational loads, and joint friction model uncertainties. These characteristics make robot control a good candidate to test control strategies, being the preferred example of several authors. Jung and Hsia propose a robust impedance control scheme that uses an ANN to cancel out the uncertainties of the inexact robot dynamic model. Sun et al present an approach for robot trajectory tracking, dropping the common assumption of the known bound on the ANN reconstruction error. A robotic planar two-link arm is here used to numerically simulate the proposed configuration, using three different combinations of design methods.

The following sections present the proposed configuration, the sliding-mode adaptive control method for the neural network controller, the adopted residue generator techniques, the robot modeling, and finally the numerical results obtained through simulations and the conclusions.

2. Proposed configuration

The block diagram depicted in Figure 1 shows the control methodology used. The following description is based on this diagram.

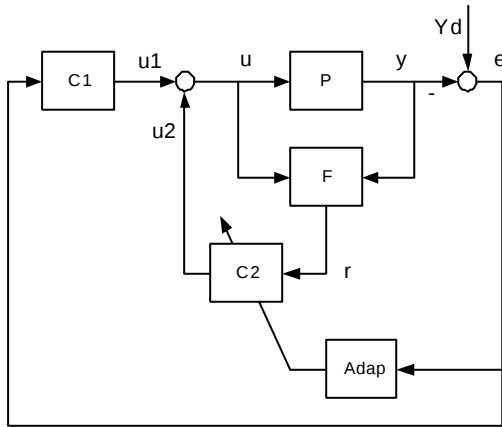


Figure 1 – Control Approach

One can see in Figure 1 that the control signal u results from the sum of the output signals of two different controllers, C_1 and C_2 . The first controller, here called the direct controller, is designed based on a regular control method, and produces the control signal vector u_1 . The second is a neural network controller, producing the control signal vector u_2 . The block F is a residue generator, whose output is the vector r . This signal vector represents the difference between the expected signals from the mathematical model of the plant and the actual measured signals. If there is a match between the

implicit model and the experimental online results, the residuals are very small and the direct controller is sufficient to govern the plant. The residue vector is the input to the neural controller C_2 and if there is a significant difference between the plant model and the actual plant performance, the respectively generated control signal will try to correct it. This difference may be due to an error modeling, but the uncertainty of the model will be corrected also. So far, the configuration is very similar to the scheme proposed by Zhou (2000). But there are two important differences: using a neural network as the second controller, and also the adopted respective sliding-mode adaptive weight adjustment (represented in Figure 1 as the module named *Adap*). The objective here is to guarantee the overall performance through transients. The difference between the response of the plant y and the desired response y_d is the error vector e . This error signal vector is used on-line to adjust the neural network weights, through sliding-mode laws.

Three cases are considered to fully explore the configuration, applied to a nonlinear plant. The first case is a model-based design, except for the neural controller, and the third case is totally neuro-based, implying that the plant model may be unknown. The second is an intermediate case. The direct controller and the residue generator are designed using two different techniques for each one. For the first case the nonlinear model was linearized around a central point of the robot workspace and a residue generator using this linearized state space model of the plant is adopted. For the second case the direct controller is the same used in the first case, but a neural network based residue generator is adopted. For the third case, the direct controller and the residue generator are both based on neural networks. For the first and the second cases a proportional-derivative control law was used, according to:

$$v = \ddot{y}_d + K_d \dot{e} + K_p e, \quad (1)$$

where the tracking error is defined as

$$e = y_d(t) - y(t),$$

and the state-space model of the robot is

$$\dot{x}(t) = f(x) + g(t)u_1(t). \quad (2)$$

Adopting

$$u_1(t) = \frac{1}{g(x)}(-f(x) + v(t)), \quad (3)$$

based on the feedback linearization scheme, the control law is calculated as

$$u_1(t) = \frac{1}{g(x)}(-f(x) + \ddot{y}_d + K_d \dot{e} + K_p e). \quad (4)$$

Notice that using the linearized functions $f(x)$ and $g(x)$, this approach would result in a regular PD direct controller.

3. Neural network adaptive control

For the three cases presented in the previous section the second controller is essentially the same, based on a neural network adaptive control approach. For this controller, a configuration of two neuron layers is used, with a linear layer at the output and a nonlinear as the input layer. The ANN output is found according to

$$u_2(t) = W_2 s_1 [W_1 f(t)], \quad (5)$$

where W_1 and W_2 are the weights of the input and output layers, $f(t)$ is the plant output error vector, and s_1 is the activation nonlinear function of the input neuron layer. Training of the network is accomplished on-line, based on a sliding-mode surface (Yu and Lloyd, 1997) and adopting the error defined as the difference between the desired plant response and measured one. The weight adaptation is according to the following expressions:

$$W_1(t) = W_1(t-1) + h(W_1(t-1) - W_1(t-2)) + \frac{\alpha \text{sign}(s_1) |e_1| X_s}{e + X_s^T X_s}, \quad (6)$$

and

$$W_2(t) = W_2(t-1) + h(W_2(t-1) - W_2(t-2)) + \frac{\alpha \text{sign}(s_2) |e_2| X_i}{e + X_i^T X_i}, \quad (7)$$

where X_s and X_i are respectively the neuron output vectors of the output and input layers, the parameters α and η are respectively the learning rate and the momentum, and ε is a small valued parameter to avoid singularity problem. The sliding surface is imposed by the variables s_1 and s_2 , defined below, and $\text{sign}(s)$ is the signum function. This formulation guarantees convergence if the system is persistently excited (Khanmohammadi 2000), a condition fulfilled due to the on-line continuous training of the ANN.

The sliding surface is presented in Equations 8.

$$\begin{aligned} s_1 &= \dot{e}_1 + l e_1, \\ s_2 &= \dot{e}_2 + l e_2, \end{aligned} \quad (8)$$

where e_1 is the performance error $y_d - y$ and e_2 is related with e_1 as $e_2 = \frac{e_1^2}{2}$, and $l > 0$ is a scalar parameter. A

complete survey of sliding-mode control can be found in Hung et. Al. (1993).

Residue generator

Two different approaches are used to implement the residue generator, a model-based observer and a neural network observer. For the first case, the adopted formulation is used for model-based fault detection.

$$\begin{aligned} r &= \hat{N}u - \hat{M}y \\ \hat{N} &= D + C(sI - A + LC)^{-1}(B - LD), \\ \hat{M} &= I - (sI - A + LC)^{-1}L, \end{aligned}$$

where $\hat{N}$ and $\hat{M}$ are rational function matrices resulting by left-factorization of the robot transfer matrix obtained through the linearized state space model $\begin{bmatrix} A & B & C & D \end{bmatrix}$, and L is the designed observer gain. Details of this residual generator formulation may be found in Ding (1994). Using this approach, the residuals are due to the behavior differences between this nominal model and the real plant. In the results presented in the numerical simulations section, they represent the error between the linear and nonlinear model of the plant.

The neural residue generator is based on an ANN trained to identify the plant. In other words, the output of the network aims to estimate the output of the plant, and may be trained with experimental data. The main advantage of this approach is that it is not necessary to know the plant model, at least to design the residual generator. The ANN output reflects the condition of the plant and the circumstances when the data was acquired. An exhaustive training using all the robot workspace is indicated. Any variation of the behavior of the plant, due e.g. to parameter drift, different payload, or even an incipient fault, will influence the residual signals.

4. Robot modelling

The dynamic model for a robotic manipulator with n links can be formulated as:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = u, \quad (9)$$

where q , $\dot{q}$, $\ddot{q}$ are the joint position, velocity and acceleration vectors, M is the inertia matrix, C represents the Coriolis and centripetal forces, G is the gravitational terms and u is the vector of joint torques. Considering that the inertia matrix is always symmetrical and positive definite Equation 9 may be solved for the joint acceleration vector as

$$\ddot{q} = M^{-1}(q)[-C(q, \dot{q}) - G(q)] + M^{-1}(q)u. \quad (10)$$

Converting to state space, the model may be written as:

$$\begin{aligned}\dot{X}_1 &= X_2 \\ \dot{X}_2 &= f(X_1, X_2) + g(X_1, X_2)u\end{aligned}\quad (11)$$

where $X_1 = [x_1 \cdots x_n]^T$, is the joint angular displacement vector, $X_2 = [\dot{x}_1 \cdots \dot{x}_n]^T$, is the velocity vector, and $f(X_1, X_2) = M^{-1}G$ and $g(X_1, X_2) = M^{-1}$. For a two link planar robotic arm, the matrices M and G are given in Equations 12 and 13 (Lewis, 1998).

$$M = \begin{bmatrix} (b_1 + b_2)a_1^2 + b_2a_2^2 + 2b_2a_1a_2\cos(x_2) & b_2a_2^2 + b_2a_1\cos(x_2) \\ b_2a_2^2 + b_2a_1\cos(x_2) & b_2a_2^2 \end{bmatrix}, \quad (12)$$

$$G = \begin{bmatrix} -b_2a_1a_2(2x_3x_4 + x_4^2)\sin(x_2) + 9.8(b_1 + b_2)a_1\cos(x_1) + 9.8b_2a_2\cos(x_1 + x_2) \\ b_2a_1a_2x_1^2\sin(x_2) + 9.8b_2a_2\cos(x_1 + x_2) \end{bmatrix}, \quad (13)$$

where the parameters are here selected as $a_1=a_2=1$, $b_1=b_2=1$.

5. Numerical simulations

In this section, the two direct controllers and two residue generators are used to explore the configuration through numerical simulations.

Case 1

Figure 2 shows both the desired and controlled positions of the two joints, using only the fed-back linearized model-based direct controller.

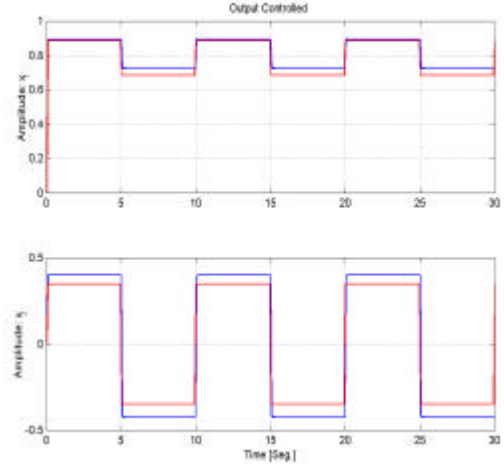


Figure 2 - Direct control

It can be easily noticed in Figure 2 the stationary error between the plant output and the desired output. This error is caused by the feedback linearization method and is proportional to the amplitude of the desired position. Figure 3 shows the respective control signal.

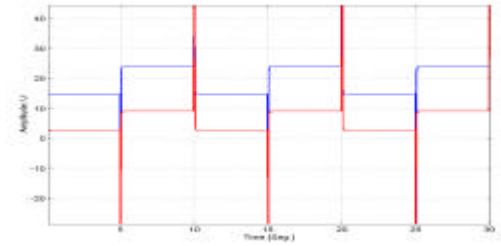


Figure 3 – Direct control signal

It is evident from Figure 3 the big overshoots necessary to maintain the square response of the joints positions as seen in Figure 4.

In Figure 4, the same results of Figure 4 are presented, but now including the AANN controller besides the direct controller, and using the model-based linear observer to generate the residuals.

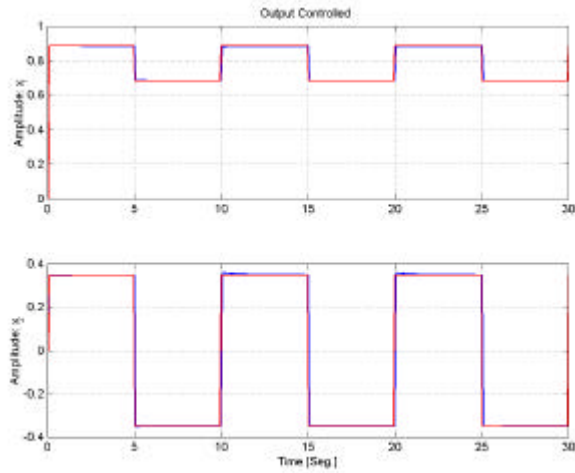


Figure 4 - Output signals for case 1

The first result seen from Figure 4 is that there is a quasi-perfect match between the two signals, for both joints. Figure 5 shows the control signals u_{11} , u_{12} , u_{21} and u_{22} . The two first signal are related to controller C_1 and the others two to controller C_2 .

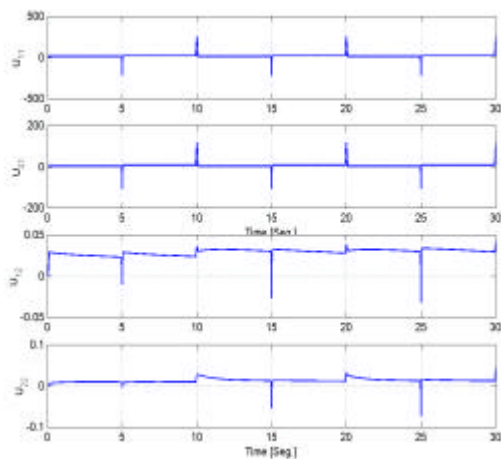


Figure 5 - Control signals for case 1

It can be seen from Figure 5 that the amplitude of u_1 is of the same order of the previous case, and that u_2 is very small compared to u_1 . But increasing tendency in the level of the control signal u_2 is clearly visible. The next figure clarifies this aspect.

Figure 6 shows the weight adaptation process. The higher panel presents the time evolution of the parameters for the nonlinear neuron layer. The lower panel present the parameters for the two output neurons.

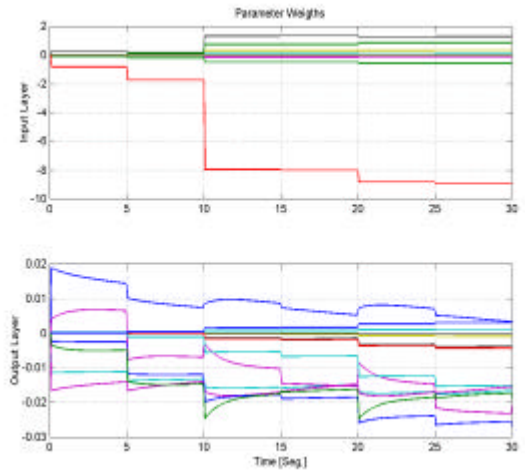


Figure 6 - Weights adaptation process

It is seen from Figure 6 also a growing tendency along the time. This weight increasing is causing the variation in the u_2 signal seen in Figure 5. This results from the continuous adaptation process, and is very similar to what happens with the least square methods used to estimate online plant parameters. The same known solution adopted there, was slightly modified to be used here, i.e. to reinitiate the neural network weights periodically.

Figure 7 shows the respective model-based generated residue. At the same time the network weights are being adapted, the AANN processes the residues to produce the control signals.

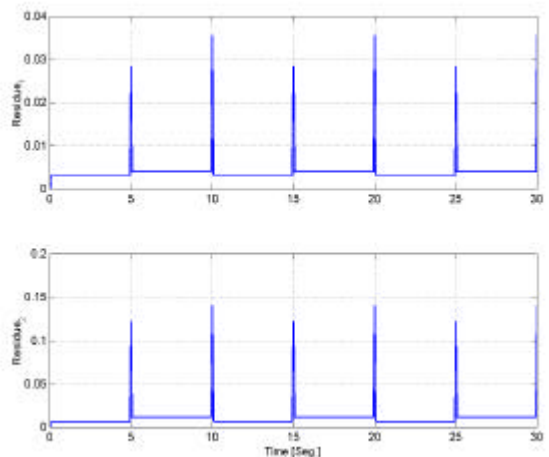


Figure 7 - Residue for case 1

Case 2

The output signals for the second case are presented in Figure 8, the residues in Figure 9 and the weights variation in Figure 10.

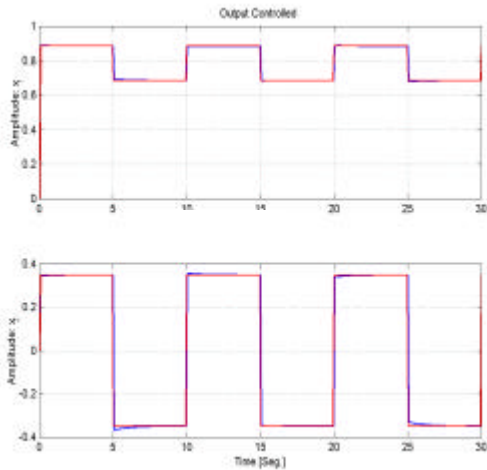


Figure 8 - Output signals for case 2

Figure 8 shows good response of the controlled output, similar to the previous case with the two controllers. Again the overshoots and the stationary errors, present small amplitudes.

Figure 9 shows the residues for the two outputs, now generated by another neural network. Note that the behavior is completely different of the residue of the first configuration.

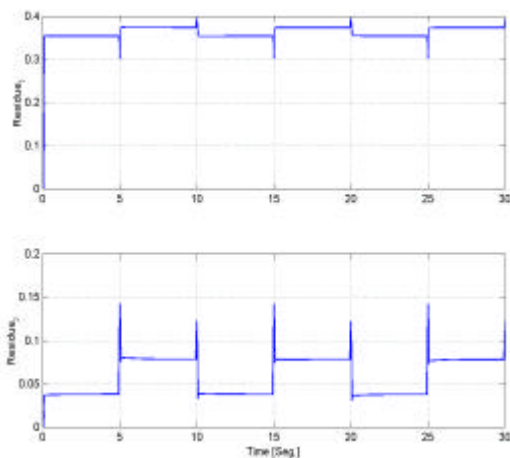


Figure 9 - Residues due to the neural observer

In Figure 10 the weight adjustment process is presented.

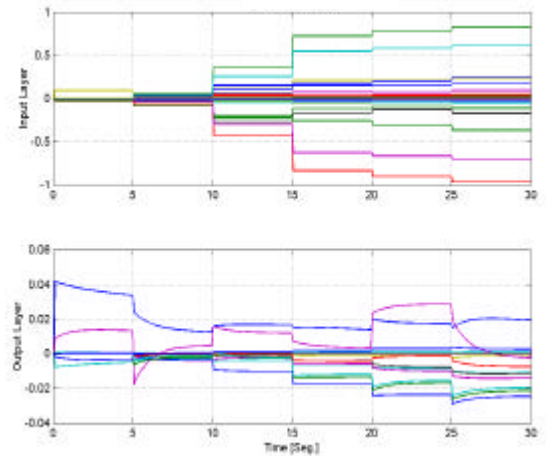


Figure 10 - Weights adjustment for case 2

Notice that here the weight variation is slightly different from Figure 6, but again the tendency of growth is seen.

Figure 11 shows the control signal related to second case.

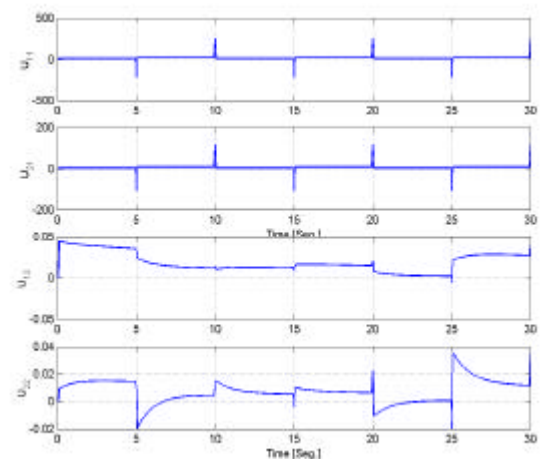


Figure 11 - Weights adjustment for case 2

Case 3

Figure 12 shows the output signals for a greater time history, 100 seconds instead of 30 seconds for the previous cases. In this case, also the first controller is neural, and it takes some time to stabilize.

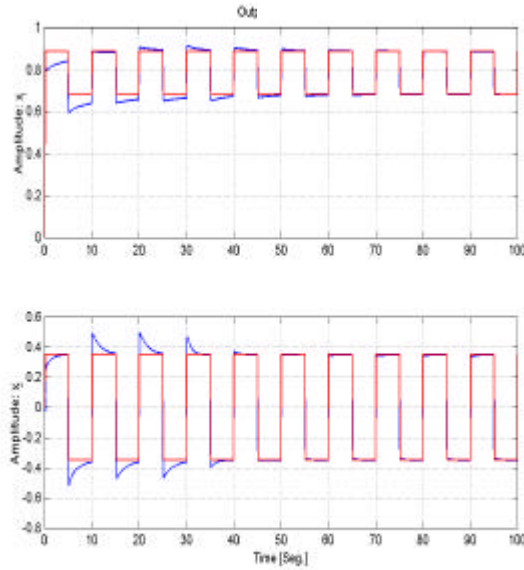


Figure 12 - Output signals for case 3

It can be seen from Figure 12 that the output signals for the first 40 seconds, while the C_1 controller is adjusting its weights, present some overshoots, but they are not so bad. After this adjustment, the signals are compatible with the previous results.

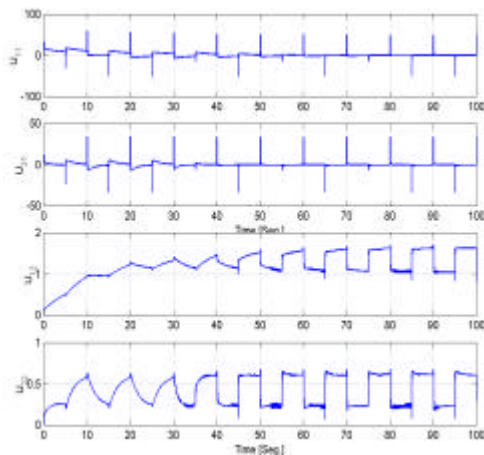


Figure 13 - Control signals for case 3

Figure 13 shows the control signals for case 3. It can be seen the initial behavior till adjustment of the C_1 weights and after that a stabilization of the signals. The levels are lower than case 2 for the signal u_1 but of the same order for signals u_2 .

Figure 14 shows the weights variation process for the C_2 controller. Here, the weight adjustment also follows the C_1 controller weights adjustment, presenting a more stable behavior after the initial phase than the other two cases.

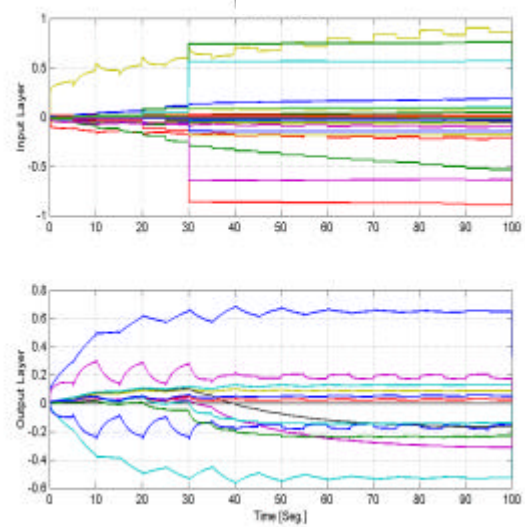


Figure 14 – Weights adjustment for case 3

Figure 15 shows the residue generation for case 3.

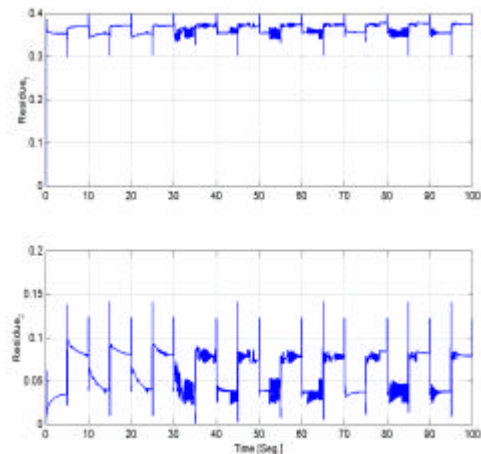


Figure 15 – Neural residues for case 3

It can be seen in Figure 15 that after the stabilization of the C_1 controller the residues present some high frequency noise, which reflects also on the control signals u_2 . This problem is probably due to the chattering provoked by the switching, but it demands further investigation because it is not present at the other cases. Any way, the output signals are not apparently affected by this noise.

6. Conclusion

As an overall conclusion, the configuration presents good results for the three studied cases. An adequate compensation for modeling errors when only the C_1 controller is used is provided by the second controller,

conducting to good reference tracking at the two arms. Further studies are necessary to investigate the effect of the switching on the residues and control signals, and also to confirm the stability and robustness of the configuration. Two different approaches may be considered for the application of the configuration, based on the availability of a reliable mathematical model of the plant. If such a model is available, a simple controller may be designed and the second controller easily implemented without previous training of the neural networks. The designer may choose between a linear observer or a neural observer, exercising his or hers judgement. If the model is not available or it is too complex to be used, a neural network only configuration may be adopted.

References

- Barambones, O. and Etxebarria, V., Robust neural control for robotic manipulator, *Automatica*, V 38, 2002, pp. 235-242
- Barambones, O. and Etxebarria, V., An adaptive neural control for mechanical manipulators with guaranteed stability, 1999, pp. 357-362
- Chang, Y.-C., Neural network-based H-inf tracking control for robotic systems, *IEE Proc.-Control Theory Appl.* V. 147, no 3, 2000, p. 303-311
- Danchi Jiang and Jun Wang, On-Line Learning of Dynamical Systems in the Presence of Model Mismatch and Disturbances, *IEEE Transaction on Neural Networks*, Vol. 11, No. 6, November 2000.
- DeCarlo, R. A., Zak, S. H. and Matthews, G. P., Variable structure control of nonlinear multivariable systems: a tutorial, *Proceedings of the IEEE*, V. 76, no 3, 1988, pp. 212-232
- Dorf, Richard and Bishop, Robert; *Modern Control Systems*, Addison Wesley, 7th edition, 1995.
- Frank, P. M. and X. Ding, *Survey of Robust Residual Generation and Evaluation Methods in Observer-Based Fault Detection Systems*, Elsevier Science Ltd, 1997.
- Hung, Y. J. and Hung, J. C., Variable structure control: a survey, *IEEE Trans. on Industrial Electronics*, V. 40, no 1, 1993, pp. 2-22
- Hwang, C.-L. and Hsieh, C.-Y., A neuro-adaptive variable structure control for partially unknown nonlinear dynamic systems and its application, *IEEE Trans. on Control Systems Technology*, V. 10, no 2, 2002, pp. 263-271
- Jung, S. and Hsia, T. C., Robust neural force control scheme under uncertainties in robot dynamics and unknown environments, *IEEE Trans. On Industrial Electronics*, V. 47, no 2, 2000, pp. 403-412
- Kemin Zhou and Zhang Ren, A New Architecture for High Performance, Robust, and Fault-Tolerant Control, *IEEE Transaction on Automatic Control*, Vol 46, No. 10, October 2001.
- Khanmohammadi, S; et. al., Modified Adaptive Discrete Control System Containing Neural Estimator and Neural Controller, *Artificial Intelligence in Engineering*, 14, 2000, 31-38.
- Lewis, F. L.; Jagannathan and Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, 1999.
- Liu, G., Control of robot manipulators with consideration of actuator performance degradation and failures, *Proc. of the 2001 Intern. Conf. on Robotics and Automation*, Seoul, 2001, pp. 2566-2571
- Mogens Blanke et. al., *Concepts and Methods in Fault – Tolerant Control*, Preceedings of the American Control Conference, Arlington, VA June 25-27, 2001.
- Noriega, J. R., and Wang, H., A direct adaptive neural-network control for unknown nonlinear systems and its application, *IEEE Trans. on Neural Networks*, V. 9, no 1, 1998, pp. 27-34
- Patiño, H. D., Carelli, R. and Kuchen, B. R., Neural networks for advanced control of robot manipulators, *IEEE Trans. on Neural Networks*, V. 13, no 2, 2002, pp. 343-354
- Sun, F. C., Sun, Z. Q., Zhang, R. J. and Chen, Y. B., Neural adaptive tracking controller for robot manipulators with unknown dynamics, *IEE Proc.-Control Theory Appl.* V. 147, no 3, 2000, p. 366-370
- Topalov, A. V. and Kaynak, O., Online learning in adaptive neurocontrol schemes with sliding mode algorithm, *IEEE Trans. On Systems, Man and Cybernetics*, V. 31, no 3, 2001, pp. 445-450
- Yu, H. and Lloyd, S., Variable structure adaptive control of robot manipulators, *IEE Proc.-Control Theory Appl.* V. 144, no 2, 1997
- Patiño, H. D., Carelli, R. and Kuchen, B. R., Neural networks for advanced control of robot manipulators, *IEEE Trans. on Neural Networks*, V. 13, no 2, 2002, pp. 343-354
- Sun, F. C., Sun, Z. Q., Zhang, R. J. and Chen, Y. B., Neural adaptive tracking controller for robot manipulators with unknown dynamics, *IEE Proc.-Control Theory Appl.* V. 147, no 3, 2000, p. 366-370
- Topalov, A. V. and Kaynak, O., Online learning in adaptive neurocontrol schemes with sliding mode algorithm, *IEEE Trans. On Systems, Man and Cybernetics*, V. 31, no 3, 2001, pp. 445-450
- Yu, H. and Lloyd, S., Variable structure adaptive control of robot manipulators, *IEE Proc.-Control Theory Appl.* V. 144, no 2, 1997, p. 167-176