A Neuro PD Control Applied for Free Gait on a Six Legged Robot

EFREN GORROSTIETA, EMILIO VARGAS, ALBERTO AGUADO.
Department of Electric and Electronic, Postgraduate Unit in Science and Technology
Technological Institute of Queretaro, Centro de Ingeniería y Desarrollo Industrial, ICIMAF
Av. Tecnologico Esq. Escobedo s/n, Av PlayaPie de la Cuesta 702
MEXICO,CUBA
efren.hurtado@usa.net, emilio@mecatronica.net , http://www.mecamex.net

Abstract: - In this work the design of a neuro PD control with gravity compensation is shown for the free gait generation on a six-legged robot. This kind of control does auto tuning on PD parameters, with the aid of neural networks, which use as input variables past and present error. A neural network is proposed, which permits a soft computing, so that it is possible to be accomplished in real time.

Key-Words: - Mechatronics, walking machine, control trajectory, hexapod robot, neuro PD.

1 Introduction

The control system design in a walking robot plays an important role in it's ability to walk efficiently. Several problems have to be solved to obtain an automatic locomotion behavior. Some of these problems are the stability control during the walking process, the working space restriction to avoid crashing impacts between legs or body, the force distribution in the robot and the adaptable way to walk on different terrains. At this point in time, several ways exist to facilitate the adaptability of walking locomotion but more research is needed.

The dynamic control design of the robot is responsible for assuring stable gait generation, which in the case of a walking robot, each of the legs will need to generate gait.

El PID is the most known and applied control in the different branches of technology, in this case one uses this type of control so that each of the legs on a walking robot follows a defined trajectory.

The configuration of a walking hexapod robot is presented in Fig 1, as it was developed in [1], this robot has the morphology of an ant, as one can see in the figure there are different relationships between each of the pairs of legs distributed around the body of the robot.

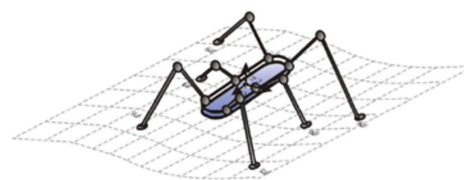

Fig. 1. Configuration of the walking robot

The development of this type of system requires the joint application of three areas of engineering, mechanical, electronic and computer systems, which together make a mechatronic system. This control design applies this kind of system.

2 Problem Formulation

Various types of parameter PID tuning exist, which are presented in classic literature such as in some of the works published [2][3].

The determination of these parameters depends mainly on the robot characteristics as well as some external elements, like surface irregularities.

For mechanical system application like the other systems, one needs the auto tuning in the control parameters; in this case it is realized in the moving leg of the walking machine, which permits the successful gait trajectories.

On the other hand one is looking for the neural systems to help with the control performance considering that the control system can present non-linear characteristics of the mechanical systems.

3 Trajectory for the Leg

The primary objective for the control proposed in this paper is to follow a trajectory for a leg step. This trajectory is parametrically designed, and it can change according to the type of land or application of the robot. Fig 2 shows a parabolic type trajectory motion for the leg, which is typical of the movement of some animals [4][5]. In the simulation study, however, the parabolic trajectory was used due to its simplicity. In this case the equations for the three degrees of freedom in the leg are described by the next equations:

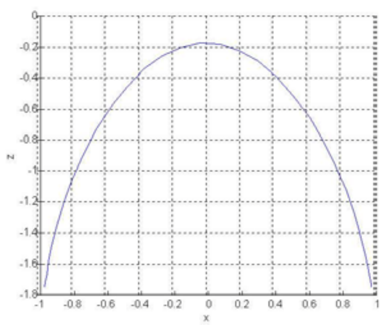

Fig. 2. Parabolic trajectory motion.

$$\begin{aligned} \theta_1 &= d\gamma - A\gamma(\cos\xi - 1) \\ \theta_2 &= d\beta - A\beta(\cos\xi - 1) \\ \theta_3 &= d\chi - A\chi(\cos\xi - 1) \end{aligned} \qquad (1)$$

where $d\gamma$, $d\beta$, $d\chi$ are the initial values of angles $\theta 1, \theta 2, \theta 3$, respectively. Those values fix the initial position of the leg in the space. Variable $A\gamma$ defines the step length and $A\beta$, $A\chi$, define the robot leg height; ξ is an angle which takes values between 0 and π.

Equation 2 shows the values considering the dimensions that are required in the generation of steps for the walking robot of six legs.

$$\begin{aligned} d\gamma &= 70 \quad A\gamma = 20 \\ d\beta &= 4 \quad A\beta = 15 \\ d\chi &= 310 \quad A\chi = 10 \end{aligned} \qquad (2)$$

4 Dynamic Model of the Leg

The design of the control law is based in the dynamic model of the robot's leg. Each leg of the robot consists of a basic configuration of three degrees of freedom as is shown in Fig. 3. The variables and parameters that conform the mathematical model of the robot are the following: θ_1, θ_2, θ_3 are the relative angles between the links, which are independent; l_2 and l_3 are the effective longitude for the link 2 and link 3; m1, m_2, m_3 and J1, J2 and J3 are the mass and the inertia for link 1, link2 and link3, respectively.

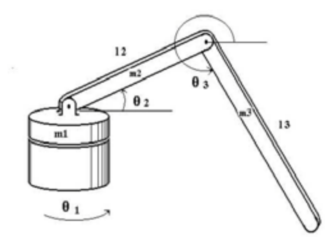

Fig. 3. Parameter and variables used in the leg modelling.

Each link is considered as a rigid body. From an energy point of view, the Lagrange technique is used to obtain the dynamic model for the legs. Equation 1 is a fundamental relationship between internal and external energy. K represents the kinetic energy of the mechanical system and U represents the potential energy.

$$L = K - U \qquad (3)$$

Equation 3 is the fundamental relationship to determine the external torque for each generalized coordinate.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_n}\right) - \frac{\partial L}{\partial \theta_n} = \tau \qquad (4)$$

The mathematical dynamic model for the legs is expressed by Equation 4, the system can be reasonably represented as a second order differential equation.

$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ b_{21} & 0 & b_{23} \\ b_{31} & b_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix} + ..$$

$$+\begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{21} & 0 & c_{23} \\ 0 & 0 & c_{33} \end{bmatrix} \begin{bmatrix} \dot\theta_1\dot\theta_2 \\ \dot\theta_1\dot\theta_3 \\ \dot\theta_2\dot\theta_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & f_{22} & f_{23} \\ 0 & 0 & f_{33} \end{bmatrix} \begin{bmatrix} m_1 g \\ m_2 g \\ m_3 g \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (5)$$

The general equation for this kind of systems can be seen in Equation 6, which represents another form as seen in Equation 5.

$$D(q)\ddot{q} + c(q,\dot{q}) + g(q) + b(\dot{q}) = \tau \quad (6)$$

where D(q) is the matrix of inertia, $c(q,\dot{q})$ the Coriolis and centrifugal terms, g(q) the gravity effect and $b(\dot{q})$ the effect of the friction.

Equation 7 is the representation of the variable of state of Equation 6.

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= D^{-1}(q)\left[\tau - g(q) - c(q,v) - b(v)\right] \\ y &= w(q) \end{aligned} \quad (7)$$

5 Dynamic Control of the Leg

Once the dynamic model of the leg is constructed, a simulation study can be accomplished that will illustrate the behavior of the system. Attention is focused to make the study considering the system torques of each joint as the inputs and the joint angles and velocities as the outputs of the model. In order to describe the position of each leg in a cartesian coordinate frame, a cinematic model to transform the robot positions to cartesian positions (X,Y,Z) was used. On the other hand, a control law was designed, which uses the angle joints errors to calculate the torques which must be applied to each joint actuator, and then evaluate the dynamic model to get new angles for the joints. Fig. 3 shows the block diagram of the control law, this scheme is well known as control with gravity compensation which is represented by C(g) [6][7].

$$\tau = Kpe + Kv\dot{e} + g(q) \quad (8)$$

Based on Equation 7, in order to consider the control systems in close loop, a change of variable (the z instead of q) is made, where z is defined $z = q - \theta$, which shows the difference between the real value and the desired value [2][8]. In Equation 9 the states are rewritten with the variable changes.

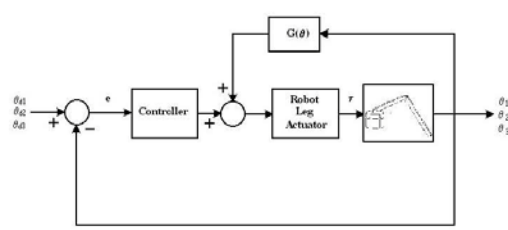

Fig. 2. Block diagram of the control law.

$$\begin{aligned} \dot{z} &= v \\ \dot{v} &= D^{-1}(z+\theta)\left[Kp + Kv - c(z+\theta, v) - b(v)\right] \end{aligned} \quad (9)$$

In order to know about the stability of the mechanical system, the Lyapunov criterion stability is applied [9][10]. In order to apply the criterion, the function V (z,v) in Equation 10 is needed, as described in [8].

$$V(z,v) = \frac{z^T K z + v^T D(z+\theta) v}{2} \quad (10)$$

The control algorithm used to move a leg in the simulations was a PD control law, which requires, as it is known, to tune two constants by each degree of freedom, proportional gain K and the derivative gain Kd. In attempt to control the position and velocity of each joint, tuning a total of 12 variables is a must.

	K1	K2	K3	Kd1	Kd2	Kd3
Value	0.06	0.09	0.2	0.009	0.6	0.01

Table 1 PD Control Constants for Position Control

	K1	K2	K3	Kd1	Kd2	Kd3
Value	0.04	0.9	0.17	0.001	0.02	0.02

Table 2 PD Control Constants for Velocity Control

Table 1 and table 2 shows the values obtained by trial and error using simulation. These PD Control constants are used to follow trajectories generated for the Equation 1, and its derivates.

The final angle results arrive at the desired trajectories even when some oscillations are observed, especially in the third joint. However, the tracking of velocity trajectories is not very successfully achieved; in fact, big deviations of desired trajectories can be observed.

To improve the performance of the PD controller, adding an adaptive component which can cope with the external disturbances, the multivariable essence and the strong nonlinearities of the robot, a neural component is proposed which can correct in real time the values of the proportional and derivative gain coefficients. In the next section, the derivation of the neural correction algorithm will be presented.

6 Neuro-PD Algorithm

In Fig. 4, a diagram of the neuro-PD controller is presented. For the sake of clarity, only two nets are depicted.

The neural nets are of the perceptron type, with a hidden layer, which has only a neuron with a sigmoid activation function. The output layer is linear and has also a single neuron. The outputs of the nets are respectively ΔK_p and ΔK_v, that is the increments of proportional and derivative gains.

The inputs of all the neuronal networks are the output errors and their first differences, that is:

$$x(t) = \lfloor e_p(t) \quad \Delta e_p(t) \rfloor \tag{11}$$

where

$$e_p(t) = \theta_d(t) - \theta(t) \tag{12}$$

represent the errors in the joints angles.

The neural networks are trained by means of special backpropagation algorithm which will be developed in detail for the case of proportional gain changes. The derivative constants modifications can be obtained in a similar way.

The objective function that must be minimized by means of the neural nets is the quadratic function:

$$E(t) = \frac{1}{2}\sum e_p^2(t) \tag{10}$$

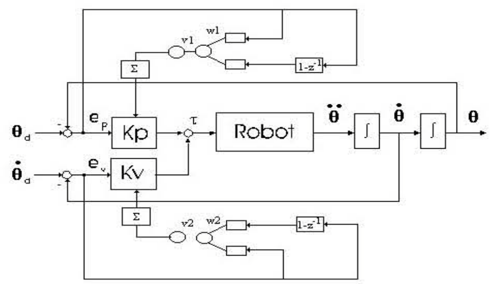

Fig. 3 Neuro PD control structure.

The joint angles $\theta(t)$ are related with the input torques by means of a supposed unknown non-linear dynamic model that we denote by:

$$\theta(t) = R(\tau(t)) \tag{13}$$

The torqueses calculated by the controller are:

$$\tau(t) = k_p e_p(t) + k_v e_v(t) \tag{14}$$

The proportional gain will change as follows:

$$k_p = k_p + \Delta k_p \tag{15}$$

$$\Delta k_p = vh \tag{16}$$

where v is the weight coefficient which connects the hidden with the output neuron and h is the output of the hidden neuron.

The inputs to the neurons of the hidden layer are

$$s = w_1 e_p(t) + w_2 \Delta e_p(t) \tag{17}$$

The activation function of the hidden neuron is:

$$h = \frac{1}{1+e^{-s}} \tag{18}$$

To apply the back propagation algorithm, it is necessary to calculate the gradient of the function E_p with respect to the coefficient v and w, that is:

$$\nabla E(t) = \begin{bmatrix} \frac{\partial E(t)}{\partial v_1} \\ \frac{\partial E(t)}{\partial w_j} \end{bmatrix} \tag{19}$$

The partial derivatives which appear in equation 19 can be calculated by means of the chain rule of derivations, that is:

$$\frac{\partial E(t)}{\partial v_1} = \frac{\partial E(t)}{\partial e_p}\frac{\partial e_p}{\partial \theta}\frac{\partial \theta}{\partial \tau}\frac{\partial \tau}{\partial K_p}\frac{\partial K_p}{\partial \Delta K_p}\frac{\partial \Delta K_p}{\partial v_1} \qquad (20)$$

$$\frac{\partial E(t)}{\partial w_j} = \frac{\partial E(t)}{\partial e_p}\frac{\partial e_p}{\partial \theta}\frac{\partial \theta}{\partial \tau}\frac{\partial \tau}{\partial K_p}\frac{\partial K_p}{\partial \Delta K_p}\frac{\partial \Delta K_p}{\partial h}\frac{\partial \Delta K_p}{\partial h}\frac{\partial h}{\partial s}\frac{\partial s}{\partial w_j} \qquad (21)$$

After the substitution of the partial derivatives which appear in equations 20 and 21, we arrive to:

$$\frac{\partial E_p}{\partial v_1} = -e_p(t)^2 h \frac{\partial \theta}{\partial \tau} \qquad (22)$$

$$\frac{\partial E_p}{\partial w_1} = -e_p(t)^2 \frac{\partial \theta}{\partial \tau} v_1 h(1-h) x_j \qquad (23)$$

As can be seen, in 22 and 23 it appears the partial derivative $\frac{\partial \theta}{\partial \tau}$, which can not be evaluated under the assumption that the robot model is not precisely known. Even in the case that a precise model was built, the evaluation would be very time consuming for real time realization of this algorithm. As it was shown in [11] [12], under some not very restrictive conditions, that partial derivative, also known as the Jacobian or equivalent gain of the process under control, can be substituted by its sign. Then, equations 22 and 23 can be simplified to:

$$\frac{\partial E_p}{\partial v_1} = -e_p(t)^2 h \, \mathrm{sign}(R) \qquad (24)$$

$$\frac{\partial E_p}{\partial w_1} = -e_p(t)^2 \, sign(R) \, v_1 h(1-h) x_j \qquad (25)$$

The function sign (R) is evaluated as +1 or –1 depending on the sign of the relationship between the angular position and the applied torque which can be considered, for the case of the robot, as always positive.

Equations 24 and 25 can be used to derive the adaptation equations for the weight coefficients of the neural net, w and v, using the steepest descent method as follows:

$$v_1(t+1) = v_1(t) + \eta e_p(t)^2 h \, \mathrm{sign}(R) \qquad (24)$$

$$w_j(t+1) = w_j(t) + \eta e_p(t)^2 \, \mathrm{sign}(R) \, v_1 h(1-h) x_j \qquad (25)$$

The adaptive equations 24 and 25 can be calculated easily in real time and serve to adapt the values of proportional coefficients Kp, using equations 15 and 16. A similar derivation can be used to find the adaptation equations for derivative coefficients K_d .

The neural adaptive model can be used in a permanent way, which achieves successful improvements of the control behavior.

7 Results

In tables IV and V the proportional and derivative gains are shown that were obtained after several simulations of the neuro PD control algorithm, parting from the values, which appear in Tables 1 and 2

	K1	K2	K3	Kd1	Kd2	Kd3
Value	0.071	0.301	0.181	0.01	0.011	0.007

Table 3 PD Control Constants for Position Control Obtained with Neuro Adaptation

	K1	K2	K3	Kd1	Kd2	Kd3
Value	0.053	1.702	0.183	0.014	0.022	0.023

Table 4 PD Control Constants for Velocity Control Obtained with Neuro Adaptation

The results of the gait generation, obtained by PD and neuro PD control can be observed in the graphic of Fig. 5.

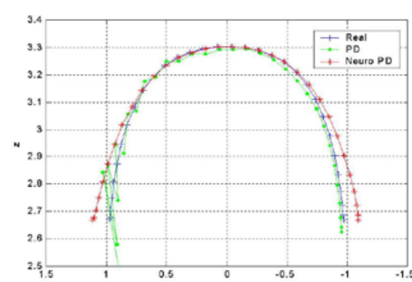

Fig 5 Result of the gait generation in the different controls

In Fig 6 the result is shown of the control implementation on simulation program designed in C++. El prototype is shown in Fig 7.

Fig. 6 Simulation of gait generation

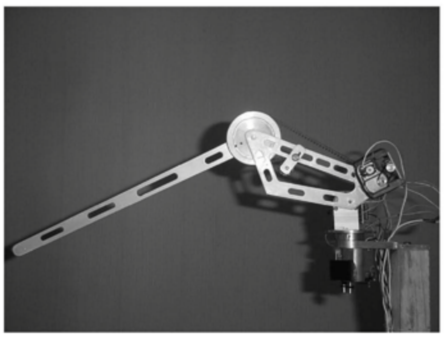

Fig.7 Prototype for gait generation.

8 Conclusion

The dynamic model described for a leg of a six-legged robot is a highly nonlinear system, obviously this complicates the design of the control system; however a PD control with gravity compensation was designed with satisfactory results. A series of preliminary trajectories were evaluated by simulation, considering a parametric mathematical model to facilitate the walking adaptation for the leg. the coefficients of the tables 1 and 2 can be used to help design a walking generator algorithm. this research is continued by considering an intelligent algorithm which includes six simple neural nets for each leg which permit the adaptation of PD coefficients and improve the performance of the robot. In the future, it is necessary also to research about the flexibility of the working space area for the legs which cause the mobility of the robot to be increased substantially. The determination of the mobility for the robot and the stability evaluation is possible by using a 3D graphic simulator before being proven in the prototype .

Appendix

Coefficients a_{ij} are the terms for the inertia

$$a_{11} = J_1 + m_3 l_2^2 \cos^2\theta_2 + \frac{1}{2} m_3 l_2 l_3 \cos\theta_2 \cos\theta_3 + \frac{1}{4} m_3 l_3^2 \cos^2\theta_3$$

$$a_{22} = J_2 + \frac{1}{4} l_2 m_2 + l_3 m_3$$

$$a_{23} = \frac{1}{2} m_3 l_2 l_3 (\mathrm{sen}\theta_2 \,\mathrm{sen}\theta_3 + \cos\theta_2 \cos\theta_3)$$

$$a_{32} = \frac{1}{2} m_3 l_2 l_3 (\mathrm{sen}\theta_2 \,\mathrm{sen}\theta_3 + \cos\theta_2 \cos\theta_3)$$

$$a_{32} = \frac{1}{4} l_3 m_3 + J_3$$

The Coriolis and centrifugal terms are defined by b_{ij} and c_{ij}.

$$b_{21} = -\frac{1}{2} l_2^2 (m_2 \,\mathrm{sen}2\theta_2 + m_3 \,\mathrm{sen}2\theta_2) - \frac{1}{4} m_3 l_2 l_3 \cos\theta_3 \,\mathrm{sen}\theta_2$$

$$b_{23} = \frac{1}{2} m_3 l_2 l_3 (\mathrm{sen}\theta_2 \cos\theta_3 - \cos\theta_2 \,\mathrm{sen}\theta_3)$$

$$b_{31} = \frac{1}{2} m_3 l_3 (l_2 \cos\theta_2 \,\mathrm{sen}\theta_3 - \frac{1}{2} \mathrm{sen}2\theta_3$$

$$b_{32} = \frac{1}{2} m_3 l_2 l_3 (\mathrm{sen}\theta_2 \,\mathrm{sen}\theta_3 + \cos\theta_2 \cos\theta_3)$$

$$c_{11} = -(m_3 l_2^2 \,\mathrm{sen}2\theta_2 + \frac{1}{2} m_3 l_2 l_3 \,\mathrm{sen}\theta_2 \cos\theta_3 + \frac{1}{4} m_2 l_2^2 \,\mathrm{sen}2\theta_2)$$

$$c_{12} = -(\frac{1}{2} m_3 l_3 l_2 \cos\theta_2 \,\mathrm{sen}\theta_3 + \frac{1}{2} m_3 l_3^2 \,\mathrm{sen}2\theta_3)$$

$$c_{21} = \frac{1}{2} m_3 l_2 l_3 (\cos\theta_2 \,\mathrm{sen}\theta_3 - \mathrm{sen}\theta_3 \cos\theta_3)$$

$$c_{23} = \frac{1}{2} m_3 l_2 l_3 (\cos\theta_2 \,\mathrm{sen}\theta_3 - \mathrm{sen}\theta_3 \cos\theta_3)$$

$$c_{33} = \frac{1}{2} m_3 l_2 l_3 (\mathrm{sen}\theta_2 \cos\theta_3 - \cos\theta_2 \,\mathrm{sen}\theta_3)$$

Coefficients f_{ij} are the terms for the gravity effect of the masses

$$f_{22} = \frac{1}{2} l_2 \cos\theta_2$$

$$f_{23} = l_2 \cos\theta_2$$

$$f_{33} = \frac{1}{2} l_3 \cos\theta_3$$

References:

[1] Solano. J, Vargas E, Gorrostieta E, Morales. C. Designing a Walking Robot of Six Legs , *In Proceedings of International Symposium on Robotics and Automation ISRA'2000* ,(Monterrey N. L., Mexico, Nov 10 – 12)2000,pp115-119

[2] Kelly Rafael and Moreno Javier, Learning PID Structures in an Introductory Course of Automatic Control, *IEEE Transaction on Education*, Vol 44, No 4 November 2001, pp 373-376.

[3] Basilio J.C. and Matos S.R. "Design of PI Controllers with Transient Specification", *IEEE*

Transaction on Education, Vol 45, No 4 Novemmber 2002, pp 364-370.

[4] Shaoping Bai, H Low and Weimiao Guo, Kinematographic Experiments on The Leg Movements and Body Trayectories of Cockroach Walking on Different Terrain , *In Proceedings of International Conferences on Robotics and Automation*, San Francisco Cal, April 2001, pp 2605-2610.

[5] Ilg. W., Mühlfriedel T., K. Berns, and R. Dillmanm, Hybrid Learning Concepts for a Biollogically Inspired Control of Periodic Movements for Walking Machines, *In Soft Computing in Mechatronics*, Germany 1999. pp 19-40.

[6] Gorrostieta Efrén, Vargas Emilio, Designig a PD control with Gravity Compensation for a six Legged Robot, *3rd International Symposium on Robotics and Automation, ISRA'2002*, IEEE, Toluca, Mexico 2002. pp. 70-74

[7] Tomás Francisco Calyeca Sánchez, Sergio Javier Torres Méndez, Germán Ardul Muñoz Hernández, PD control system whit gravity compensation, *In Proceedings of International Symposium on Robotics and Automation ISRA'2000* (Monterrey N. L., Mexico, Nov 10 – 12) 2000,pp 193-195.

[8] Robert J. Schilling, *Fundaments of Robotics Analysis and Control*, Prentice Hall. USA. 1990.

[9] K. Khalil., *Nonlinear System,* Prentice Hall, USA.1998

[10] E. Slotine and Li Weipin., *Applied Nonlinear Control* , Prentice Hall. USA.1991

[11] Cui, X., Chin, K.G., Direct Control and Coordination using Neural Networks, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3, 1992, pp 686-697,

[12] Aguado Behar Alberto, *Temas de Identificación y Control Adaptable*, Instituto de Cibernética, Matemática y Física. Habana, Cuba 2000.