Satellite Attitude Acquisition Using a Neural Network Controller

Valdemir Carrara[+] & Atair Rios Neto[++]
[+]*Instituto Nacional de Pesquisas Espaciais – INPE/MCT*
CEP 12201-970 CP 515 São José dos Campos, SP. E-mail: val@dem.inpe.br
[++]*Instituto de Pesquisa e Desenvolvimento, Universidade do Vale do Paraíba - UNIVAP*
CEP 12244-000 São José dos Campos, SP. E-mail: atair@univap.br

ABSTRACT

Neural networks are non-linear functional approximators, and natural candidates to be used in non-linear systems control. Satellite nominal attitude control dynamics is an example of non-linear system where conventional control has been applied sucessfully since the early space age. However, conventional control like the PD (proportional and derivative of the error) can only guarantee robustness to work in a small region around the reference or nominal attitude. This certaninly is not the case of an attitude acquisition, where large maneuvers are commanded autonomously in order to point the satellite close to a nominal attitude. Searching for a solution that overcomes this limitation, in this work it is investigated the possibility of performing satellite attitude acquisition control by means of an artificial neural net. It is employed the feedback error learning method in order to train the network with the inverse dynamics. A simulated example of an attitude acquisition of a satellite with deploying solar arrays is presented.

INTRODUCTION

Artificial Neural Networks (ANN) present several characteristics that make them promising to be used in non-linear systems controllers. Since ANN can approximate with a given degree any continuous and limited function (Hornik, Stinchcombe and White, 1989), they can also perform system identification and control even in presence of non-linearities. Neural nets have found several uses in many areas such robotics, stock market, character recognition, image analysis and feature extraction, voice identification and speech recognition, among others (Demuth and Beale, 1992). In spite of its potential capacity, neural nets are being employed with care in space vehicles control, mainly in non-critical jobs (Vadali, Krishnan and Singh, 1993; Carrara, Varotto and Rios Neto, 1998).

Traditionally, attitude control is based on linearization of the dynamical equations of motion in order to obtain the gains of a linear controller. The theory guarantees that the system is stable and controllable as long as the linearization error remains small and the disturbances are limited. These requirements are no longer needed for an ANN based attitude controller and therefore the control can accomplish large attitude maneuvers, where the linearization is not valid. This is certainly the case of attitude acquisition, when angular rate reduction and rough spacecraft pointing shall be done in reduced time. Although pointing accuracy is not a constraint in this phase, it shall be noted that during attitude acquisition the solar arrays are being opened and appendages may be deployed, which causes changes in attitude and variations in the satellite inertia and center of mass.

To validate an ANN attitude controller, the dynamics and kinematics of a given satellite was simulated in computer. The dynamics of the solar array during deployment was added to the problem in order to prove the net effectiveness in the presence of highly non-linear perturbations. Although the solution proposed here was applied to a small satellite, the concept is valid also for large space systems with dynamic variation, due to, for instance, module accretion, rotation of the solar arrays to follow the Sun and boom deployment.

NEURAL NETWORKS

A neural network is a computational structure composed of several units called artificial neurons (Figure 1). Each neuron can be understood as an operator that process its inputs and transfer the output to the next neuron layer. The signal processing performed by the neuron establishes its functionality. The connections between the artificial neurons, on the other hand, define the behavior of the net, identify its applicability and training methods. In a perceptron network the neurons are grouped in one or more layers, with the output of each layer being the input of the next one. The neurons combine their inputs after applying a weight on each one, and then apply a non-linear activation function *f* on the result. In some ANN architectures, the neuron connectivity can be extended to allow feedback between consecutive layers (Hunt et alli, 1992).

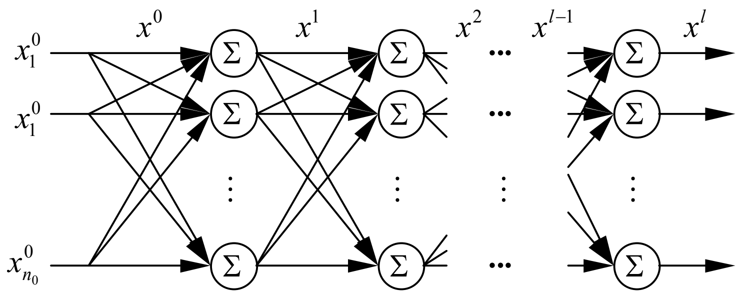

Fig. 1 – A feedforward neural network.

The training process consists in adjusting the neuron weights based on the expected output and some optimization rules. Normally, the weights are adjusted interactively, by comparing the output of the network with the desired value at each step. Consequently, the ANN learns progressively how to approximate the functional mapping.

A neural net with linear activation function in the output layer and the sigmoid activation function in the hidden layers better represents discretized dynamical systems and limited continuous functions (e. g., Chen and Billings, 1992). The greater the number of hidden layers the better the representation of dynamical system and the smaller the output error (Nguyen and Widrow, 1990; Chen and Billings, 1992), even if one considers the same number of neurons. Nevertheless, the capacity of generalization,

i. e., the ability to interpolate between points where the neural net was not trained is more accentuated on nets with few or even only one hidden layer (Baffes et. alli, 1991). On the other hand, nets with high number of neurons or layers have small output errors at the trained points. Thus if the system dynamics is not complex, a neural net with one hidden sigmoid and linear output layers is sufficient for a large number of applications. The number of neurons in the hidden layers is important for the approximation degree: few neurons tend to decrease the stability and result in bad approximation, too many neurons cause oscillation on the output between the trained points (Billings et alli, 1992).

Backpropagation algorithm

Neural nets present some advantages when compared with traditional function approximation methods. Firstly, ANN has parallel processing structures that can be used in conjunction with parallel processors. They can also easily handle large number of input and sensorial elements, and there is no need of mathematical modeling of the sensors. The main characteristic is, however, the capacity of gradually learning how to respond to a given input-output relationship. Training a neural net generally consists in applying methods in order to obtain the neuron weights. The training process normally minimizes the output error through the application of an optimized identification method as, for example, batch estimators, stochastic estimators (Rios Neto, 1994) or recursive algorithms (Chen and Billings, 1992; Carrara, Varotto and Rios Neto, 1998). Independently of the training procedure, these methods need to know how the net output varies with respect to a given neuron weight, in order to adjust the weights based on the output error. This can be achieved with the back propagation algorithm (Hunt et alli, 1992), which obtains the partial derivative of the output elements in a recursive way. The most commonly used training procedure today is the steepest descent or gradient method (Nguyen and Widrow, 1990). It is easy to implement in computers, very fast to compute a single point, but converges in a excessively slow rate. This certainly is the main reason of the extremely long training times in most neural net applications. There are several other training methods, showing each one some advantages and disadvantages when compared with the steepest descent.

Steepest descent method

The steepest descent method (sometimes misunderstood with the backpropagation algorithm) shows, as the backpropagation, a high degree of parallelism and simplicity. The weights are corrected based on the minimization of the neural net output error. Weight updating starts at the net output layer and then the error is backpropagated to the preceding layer in order to compute its weight corrections. The minimization criterion uses the network output quadratic error as the performance index:

$$J(t) = \frac{1}{2} \varepsilon(t)^{\mathrm{T}} \ \varepsilon(t) \tag{1}$$

where $\varepsilon(t)$ is the network output error at time t, defined by:

$$\varepsilon(t) = y^d(t) - y(t) \ , \tag{2}$$

where $y^d(t)$ and $y(t)$ are expected and actual network output. The weight matrix on layer k is updated using:

$$W^k(t+1) = W^k(t) - \lambda \nabla J^k \tag{3}$$

where ∇J^k is the gradient of the square error. Convergence of the weights depends on the adjusting of the learning rate coefficient λ, ranging from 0 to 1. Small λ slows down the learning process, while high gain turns the weight updating unstable.

ATTITUDE DYNAMICS

When external torques are applied to the satellite, the variation rate of the angular momentum, expressed in body coordinates x^o, y^o and x^o, is equal to the sum of the applied torques (Crandall et alli, 1968; Wertz, 1978):

$$\dot{L}^o = I_o \dot{\omega}_o^o + \Omega(\omega_o^o) I_o \omega_o^o = N_{cont} + N_{pert} \tag{4}$$

where I_o is the satellite inertia matrix, ω_o^o is its angular velocity, and $\Omega(\cdot)$ is the vector product matrix, defined by:

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{5}$$

External torque can be separated in environmental or disturbance torque, N_{pert} and attitude control torque, N_{cont}. If the satellite is composed of articulated appendages, or if some appendages like the solar arrays are flexible, Equation 4 shall be modified in order to reflect the effects caused by the non-rigidity. Some examples of non-rigid satellites include: a spacecraft pointed to Earth with solar arrays tracking the Sun, or during the process of unfolding the solar arrays after orbit injection; a robot space arm; or even the docking of a new module to a space station. In all these examples, both the inertia and center of mass position vary with time.

Suppose that a rigid main body with n articulated appendages (also individually rigid) composes the satellite. In order to avoid extending the system degrees of freedom, the angular velocities and accelerations of each articulation is supposed known. This is true for a large number of satellites, such as an Earth pointing satellite with rotating solar arrays. The angular momentum rate of the satellite can now be expressed as a sum of the individual momentum:

$$\dot{L}^o = \int_{V_o} \Omega(r_o^o)\, \ddot{r}_o^o \, dm_o + \sum_{k=1}^{n} \int_{V_k} \Omega(r_k^o)\, \ddot{r}_k^o \, dm_k \, . \tag{6}$$

where r_o and r_k are, respectively, the position of the mass elements dm_o e dm_k, belonging to the main body and to appendage k (k = 1, …, n). The momentum rate with respect to the satellite center of mass and the position vectors are expressed in main body coordinates. V_o and V_k are the volumes of the main body and appendage k. The above integral yields:

$$\dot{L}^o = \left(I_o + J_n\right)\dot{\omega}_o^o + \Omega(\omega_o^o)\, I_o\, \omega_o^o + H_n\,. \tag{7}$$

Except for J_n and H_n, the angular momentum rate is similar to the Equation 4. So J_n and H_n represent the appendage and center of mass motion effects. They are defined by:

$$\begin{aligned} J_n = {} & \sum_{k=1}^{n}\left(A_{k,o}\, I_k\, A_{k,o}^T - m_k\, \Omega(a_{ok}^o - a_{ko}^o)\,\Omega(a_{ok}^o - a_{ko}^o)\right) + \\ & + \sum_{k=1}^{n}\left(m_k\, \Omega(a_{ok}^o - a_{ko}^o)\right)\sum_{k=1}^{n}\left(\mu_k\, \Omega(a_{ok}^o - a_{ko}^o)\right) \end{aligned} \tag{8}$$

and

$$\begin{aligned} H_n = {} & \sum_{k=1}^{n}\left[\Omega(\omega_o^o + \omega_k^o) A_{k,o} I_k A_{k,o}^T (\omega_o^o + \omega_k^o) + A_{k,o} I_k A_{k,o}^T (\dot{\omega}_k^o + \Omega(\omega_o^o)\omega_k^o)\right] + \\ & + \sum_{k=1}^{n} m_k\, \Omega(a_{ok}^o - a_{ko}^o)\beta_k - \left(\sum_{k=1}^{n} m_k\, \Omega(a_{ok}^o - a_{ko}^o)\beta_k\right)\sum_{k=1}^{n}\mu_k\, \beta_k\,, \end{aligned} \tag{9}$$

where I_k is the inertia matrix of appendage k expressed in the appendage coordinate system. $A_{k,o}$ is the rotation matrix between the appendage k and the main body coordinate systems and m_k is the appendage mass. Any fixed point in the articulation axis k will define the vector a_{ok}, with respect to the origin of the main body and a_{ko}, with respect to the origin of the appendage frame. The mass proportion μ_k is defined by:

$$\mu_k = \frac{m_k}{m_o + \sum_{k=1}^{n} m_k} \tag{10}$$

where m_o is the main body mass. The angular acceleration β_k is given by:

$$\begin{aligned} \beta_k = {} & \Omega(\omega_o^o)\,\Omega(\omega_o^o)\,(a_{ok}^o - a_{ko}^o) - \Omega(\omega_o^o)\,\Omega(\omega_k^o)\, a_{ko}^o - \\ & - \Omega(\omega_k^o)\,\Omega(\omega_o^o + \omega_k^o)\, a_{ko}^o + \Omega(a_{ko}^o)\left(\Omega(\omega_o^o)\,\omega_k^o + \dot{\omega}_k^o\right) \end{aligned} \tag{11}$$

Note that the appendage angular velocity ω_k^o and acceleration $\dot{\omega}_k^o$ vectors define both the momentum and the direction of the articulation joint. Equation 7, together with the kinematic equations of motion in quaternions $q = [q_1, q_2, q_3, q_4]^T$ (Wertz, 1978):

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & \omega_{oz}^o & -\omega_{oy}^o & \omega_{ox}^o \\ -\omega_{oz}^o & 0 & \omega_{ox}^o & \omega_{oy}^o \\ \omega_{oy}^o & -\omega_{ox}^o & 0 & \omega_{oz}^o \\ -\omega_{ox}^o & -\omega_{oy}^o & -\omega_{oz}^o & 0 \end{bmatrix} q \tag{12}$$

can now be integrated in order to simulate the attitude of a satellite with variable geometry.

SIMULATION RESULTS

The neural network control (NNC) was implemented and simulated using the MECB (Brazilian Complete Space Missions) satellite characteristics. They are small satellites designed to test low Earth orbit communications and to perform Earth observation. Immediately after orbit injection, the spacecraft shall perform a rate reduction, in order to stop the tumbling and rotation motion imposed by the launcher's last stage and separation torque. The satellite then opens 3 solar panels and enters in attitude acquisition in order to point the panels to Sun. During the deployment, the mass motion of the solar arrays changes the satellite inertia and center of mass position. It was supposed that a neural net controls the attitude of the satellite in this phase. For attitude data acquisition, the satellite uses a magnetometer and an analog sun sensor. Attitude is controlled with hydrazine thrusters, on 3 axes, with a torque generation of 0.2 Nm maximum.

The network training process uses the simulated attitude response to the torque control in order to update the neural weights. A feedback learning control (FLC) algorithm (Kawato et alli, 1988) was employed to train the network controller, as shown in Figure 2. Before training the ANN controller, its weights are adjusted to provide a minimum output in order to avoid applying an erratic torque on the satellite. In this situation, the attitude is controlled mainly by a conventional PD (proportional and derivative) acting as a feedback controller. During learning the weights are updated in such a way as to minimize the training error, or the PD signal. Once the training process is completed, the network controls the system based on a feedforward reference trajectory, y^d. The PD still controls the attitude, although in a smaller extent, just to correct small attitude deviations. This arrangement allows to perform large attitude maneuvers by means of the ANN, where non-linearities are important.

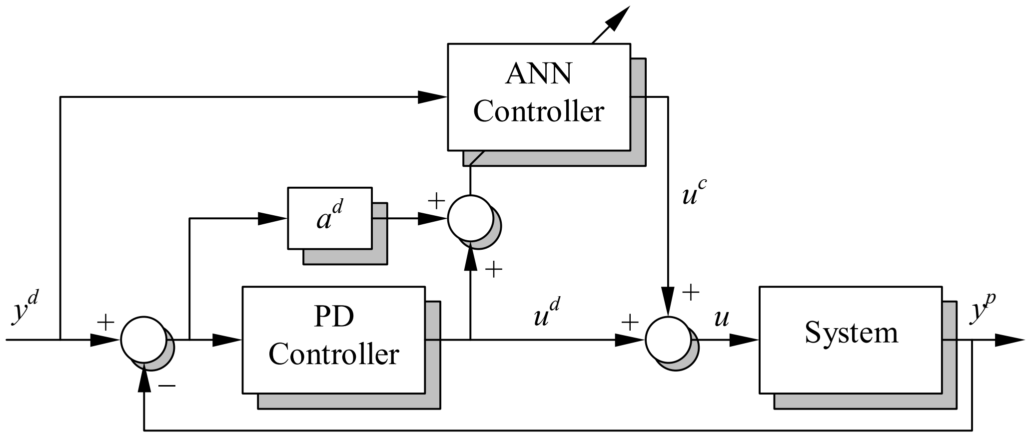

Fig. 2 – Feedback error learning control with acceleration threshold.

Depending on the input values used in the control net, however, a balance situation can occur, with both, PD and ANN controllers trying to control the attitude with the same signal intensity, but in opposite direction. To overcome this problem, Kyung (Kyung et alli., 1994) suggests the inclusion of a threshold to the torque applied to the system, as shown in Figure 2. The torque can be estimated by measuring (or numerically deriving) the angular acceleration at system output, resulting:

$$a^d = \nu\,\mathrm{sgn}(\ddot{y}^d - \ddot{y}^p)\left(\frac{1-e^{-\kappa\gamma}}{1+e^{-\kappa\gamma}}\right), \tag{13}$$

where ν is the angular acceleration gain, and sgn represents the sign function. The term in brackets is the sigmoid function of argument κγ, with κ being a constant and γ the absolute value of the deviation in acceleration:

$$\gamma = \left\|\ddot{y}^d - \ddot{y}^p\right\|. \tag{14}$$

The threshold a^d is limited to ± ν, with two values to be adjusted empirically: κ and ν. In the simulation, κ was assumed equals to 500 s^2/rd, and ν = 0.002. The learning signal can now be expressed as the sum of the PD control u^d and the acceleration threshold a^d. Since the training data is not related to the ANN output error, care must be taken to avoid non-convergence of the learning process by choosing a small value for the learning parameter λ, adopted equals to 0.001 in the simulation. The reference trajectory $y^d(t)$ was generated based on the Euler theorem that states that the rotation matrix between two distinct (initial and final) coordinate systems can be obtained by a single rotation, ϕ, on a particular fixed axis, e_v, given by:

$$\phi = 2\,\arccos(q_4) \text{ and } e_v = \frac{q}{\sin(\phi/2)}, \tag{15}$$

where e_v uses only the first 3 components of q. To generate the trajectory, it was assumed that the satellite is accelerated in the first half of the maneuver and then slowed down in the second half, according to the following rotation angle:

$$\varphi(t) = \begin{cases} 0{,}5\,\alpha\,t^2, \text{for } 0 \le t < \sqrt{\phi/\alpha} \\ 2\sqrt{\phi\,\alpha}\,t - \phi - 0{,}5\,\alpha\,t^2, \text{for } \sqrt{\phi/\alpha} \le t < 2\sqrt{\phi/\alpha} \\ 0, \text{for } t \ge 2\sqrt{\phi/\alpha} \end{cases}. \tag{16}$$

The maximum trajectory acceleration was α = 0.003 rd/s^2 so as to avoid thruster saturation.

During attitude acquisition, 2 analog sun sensors and a 3 axes magnetometer provide attitude information. The reference trajectory is converted so as to emulate the sensor measurements (3 for the magnetometer and 3 for the sun sensors) and to input both the PD and the ANN controllers. The network learns the system dynamics based on some past information concerning the thruster activation and the reference trajectory, as well as the solar array angle during deployment. This information is summarized in Table 1, where the past values are referred as t-Δt, t-2Δt, etc. Note also that the ANN needs to know the reference attitude at time t+Δt, in order to represent the system inverse dynamic (Narendra and Parthasarathy, 1990).

The ANN controller was composed of 30 neurons in the hidden layer and 3 neurons in the output. The activation functions were sigmoid and linear, respectively. Training was carried out with 5000 different initial conditions, starting from an initial random attitude, and reaching a fixed target, given by a 3-1-3 rotation angles: φ = -120°, η = 45° and ψ = -90°. Each simulated maneuver lasts 100 seconds (step size of 1s), with solar array deployment at t = 50s. Initial angular velocity was limited to 0.5 rd/s maximum. The PD controller gains were adjusted by trial and error so as to stabilize the training convergence and the satellite attitude, resulting K_p = 0.3 and K_d = 20. After training, a complete attitude acquisition process was simulated with a new initial condition. The results, presented in Figure 3, show that the control reaches its objective to perform an attitude acquisition. During solar array deployment (t = 50s), the thrusters saturate and the attitude errors becomes larger. In this figure, the superscript d indicates the reference trajectory and p means the actual trajectory. Nevertheless, the satellite attitude is still controllable. Torque history can be seen in Figures 4, 5 and 6, for both ANN and PD controllers. It can be noted that the neural net provides a smooth control, proper to a feedforward control, while the PD takes account of the attitude deviation around the reference trajectory. A later study indicated that the ANN plus PD arrangement presents only a marginal difference on the attitude error when compared with the PD controller only. The conclusion was that the ANN did not acquire enough information from the system dynamics, due perhaps to the large amount of training points and/or few neurons, which made the training incomplete. To better represent the attitude problem, probably it would be necessary to increase the number of neurons up to a number that would make the training process almost impracticable.

Table 1 – Input data to the ANN controller

ANN input	**Number of elements**	$t+\Delta t$	t	$t-\Delta t$	$t-2\Delta t$
Analog Sun Sensor	3	✓	✓	✓	✓
Magnetometer	3	✓	✓	✓	✓
Thrusters	3			✓	✓
Solar array angles	1		✓	✓	✓

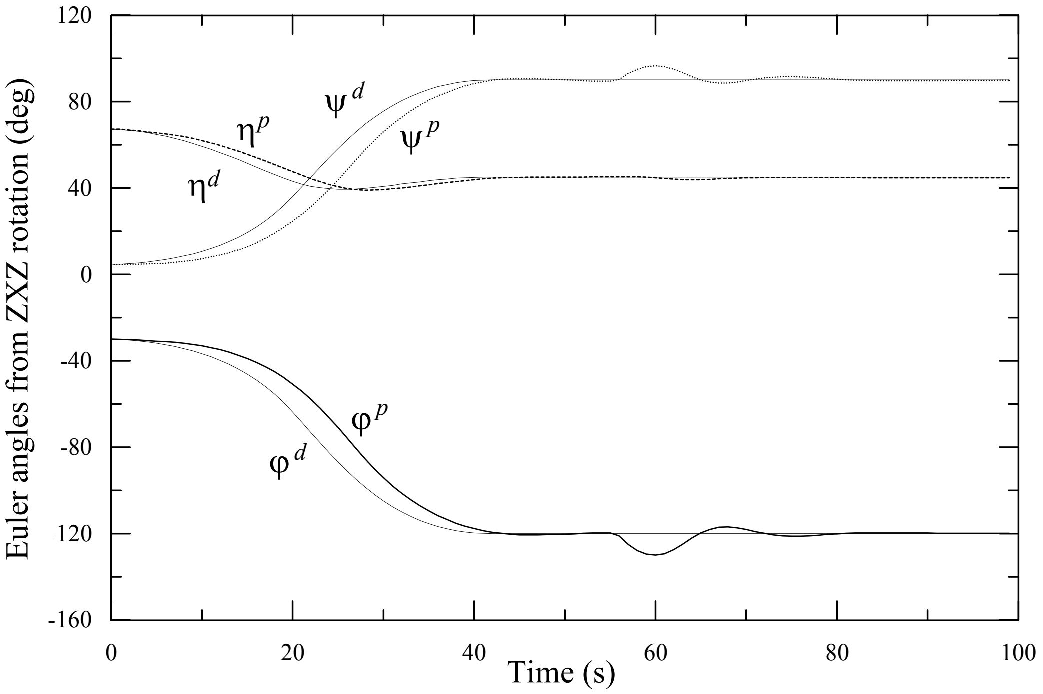

Fig. 3 – Attitude acquisition with ANN and PD controllers.

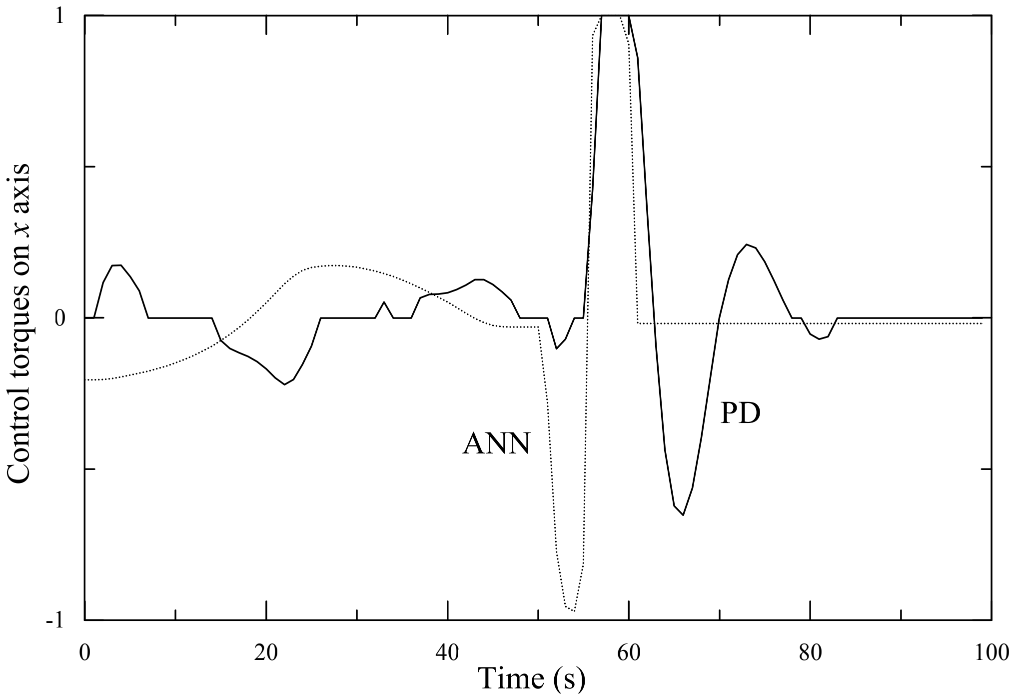

Fig. 4 – Control torque of the ANN and PD controllers, in the *x* axis.

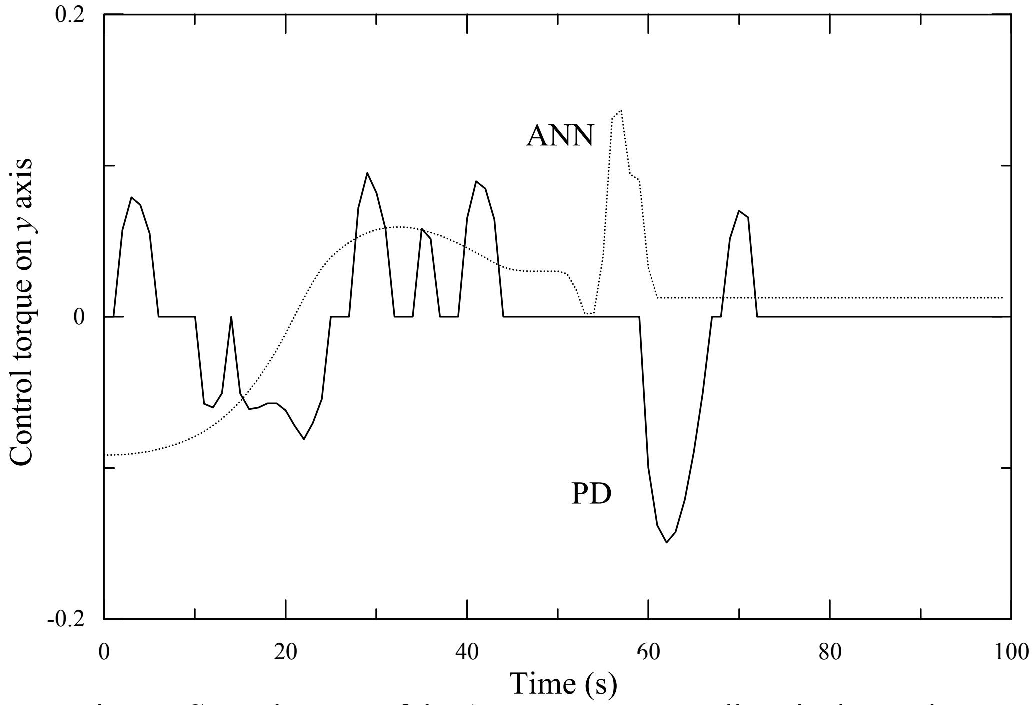

Fig. 5 – Control torque of the ANN and PD controllers, in the *y* axis.

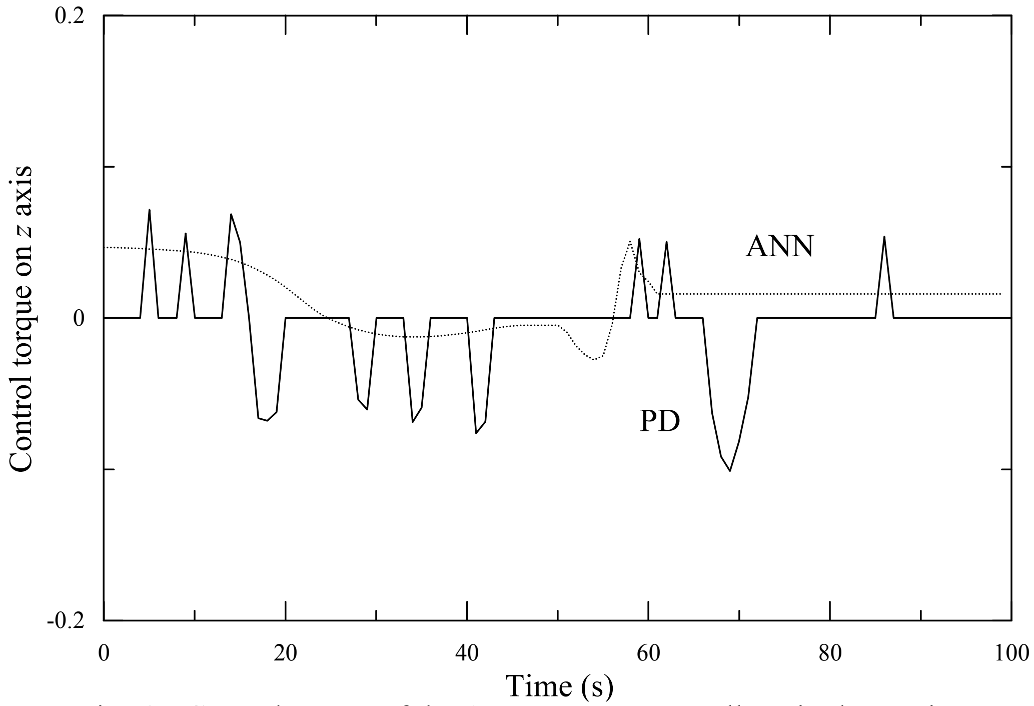

Fig. 6 – Control torque of the ANN and PD controllers, in the *z* axis.

CONCLUSIONS

A neural network based attitude controller of a satellite with deployable solar arrays was developed. The control architecture was derived from the feedback error learning algorithm with acceleration threshold, but since the training error in this approach depends on the system feedback error instead of the net output error, the convergence of the learning process can not be guaranteed. Also, the training process needs too many training points in order to acquire the system dynamics making it difficult to get a complete learning. Also, the large number of state variables in the attitude dynamic makes the neural net too big and the training process too long. In spite of these particularities, attitude control by means of ANN is possible, although the results with the particular control architecture used have shown that the performance of the ANN was only marginally better than that of a conventional PD controller. It should be noted, nevertheless, that neural network control applied to satellite attitude control is still under development and it is expected to have improved solutions as more computing power, better training algorithms and other control architectures are considered.

REFERENCES

Baffes, P. T.; Shelton, R. O.; Phillips, T. A. *NETS, a neural network development tool*. Lyndon B. Johnson Space Center, JSC-23366, 1991.

Billings, S. A.; Jamaluddin, H. B.; Chen, S. Properties of neural networks with applications to modelling non-linear dynamical systems. *International Journal of Control*. 55 (1):193-224, 1992.

Carrara, V.; Varotto, S. E. C.; Rios Neto, A. Satellite Attitude Control Using Multilayer Perceptron Neural Networks (98-345). *Advances in the Astronautical Sciences*. Vol. 100, Part 1, 1998. p. 565-579.

Chen, S.; Billings, S. A. Neural networks for nonlinear dynamic system modelling and identification. *International Journal of Control*, 56 (2):319-346, 1992.

Demuth, H; Beale, M. *Neural network toolbox user's guide.* Natick, MA. Math Works, 1992.

Crandall, S. H.; Kornopp, D. C.; Kurtz Jr, E. F.; Pridmore-Brown, D. C.; *Dynamics of mechanical and electromechanical systems*. Mc Graw-Hill, NY, 1968.

Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, p. 359-366, 1989.

Hunt, K. J.; Sbarbaro, D.; Zbikowski, R.; Gawthrop, P. J. Neural networks for control systems - a survey. *Automatica.* 28 (6):1083-1112, 1992.

Kawato, M; Uno, Y.; Isobe, M; Susuki, R. Hierarchical neural network model for voluntary movement with application to robotics. *IEEE Control Systems Magazine*, v. 8, n. 2, p. 8-15, Apr. 1988.

Kyung, K. H.; Lee, B. H.; Ko, M. S. Acceleration based learning control of robotic manipulators using a multi-layered neural network. *IEEE Transactions on Systems, Man and Cybernetics*. v. 24, n. 8, p. 1265-1272, Aug. 1994.

Narendra, K. S.; Parthasarathy, K. Identification and control for dynamic systems using neural networks. *IEEE Transactions on Neural Networks*, 1, 1990.

Nguyen, D. H.; Widrow, B. Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, 10 (3), Apr. 1990.

Rios Neto, A. *Stochastic parameter estimation neural nets supervised learning approach*. Proceedings of the 1st Brazilian Congress in Neural Nets, Itajubá, MG, 1994.

Vadali, S. R.; Krishnan, S.; Singh, T. Attitude control of spacecraft using neural networks. *Advances in the astronautical sciences.* Americam Astronautical Society. Vol. 82, Part 1, 1993.

Wertz, J. R. *Spacecraft attitude determination and control*, London, D. Reidel, 1978 (Astrophysics and Space Science Library).