10th Brazilian Congress on Computational Intelligence (CBIC'2011), November 8 to 11, 2011, Fortaleza, Ceará Brazil

THREE-AXIS STABILIZED GEOSTATIONARY SATELLITE ATTITUDE CONTROL USING NEURAL PREDICTIVE ALGORITHMS

[1]Jaime A. da Silva, [2]Atair Rios Neto

[1]jaimeausilva@uol.com.br
[2]atairrn@uol.com.br

Abstract – A study of a three-axis stabilized satellite attitude control in geostationary orbit is presented. The analysis is based on a neural predictive approach using Kalman filtering algorithms. Optimization of a quadratic performance functional is used to determine the discrete predictive control actions. The control determination on a typical iteration is viewed as a stochastic optimal linear parameter estimation problem. Direct analogy with Kalman filtering algorithms allows the derivation of full non-parallel as well as approximated parallel processing algorithms. The satellite control system is based on a gyro device, which furnishes control torques on all three vehicle's axes. Therefore, wheel speed control plus two-degree-of-freedom gyrotorquing supply the required moments to counterbalance attitude perturbations due to the solar pressure torques, limited orbit control thruster misalignment, or small initial satellite attitude misalignment. The results demonstrate that the proposed neural predictive scheme, at only one step-ahead prediction furnishes smooth control actions required to point and to maintain the satellite stabilized in the desired direction. Simulations are presented for a one-day satellite attitude control and for small initial attitude angles misalignment.

Keywords – Neural Predictive Control, Neural Networks, Kalman Filtering, Satellite Attitude, Geostationary Satellite.

1 Introduction

A neural predictive control scheme, using Kalman filter algorithm (Silva and Rios Neto, 1999, 2000, 2001, Tasinaffo and Rios Neto, 2007) is applied to a geosynchronous satellite attitude control problem. One case of great interest is related with the design of a double gimbaled momentum wheel attitude control system employed in the three-axis stabilization of a satellite attitude in geostationary orbit. This type of gyro device offers control torques about all three vehicle's axes through wheel speed control and two-degree-of-freedom gyrotorquing. With an adequate wheel size and selected speed, momentum exchange permits cancellation of cyclic torques. This kind of system substitutes with advantages mass expulsion control systems in which continual thrusting and roll, pitch, and yaw sensors are required for accurate attitude maintenance. Specific satellite properties are assumed and appropriate control laws are calculated to counteract disturbance torques due to solar pressure and limited orbit control thruster misalignment. Applicability of the control scheme is tested and responses obtained considering various conditions for initial attitude misalignment and the presence of the cyclic disturbance torques.

The proposed solution considers a one step-ahead neural predictive control approach. On this scheme a multilayer perceptron neural network is trained to learn and emulate the satellite attitude nonlinear dynamic behavior. Subsequently, a neural predictive control algorithm with Kalman filter structure is employed to obtain the control actions required to three-axis stabilize the attitude of a geosynchronous satellite in the presence of solar periodical perturbations and gravity gradient torques. For small initial satellite attitude misalignment, the applied methodology demonstrates that it furnishes precise control actions to smoothly position the satellite within the allowed orbit attitude accuracy requirements, and that it is also capable of maintaining the correct satellite attitude during a one-day attitude simulation.

For the derivation of the predictive control algorithm, previous knowledge in using stochastic optimal parameter estimation to solve optimization problems (Rios Neto and Pinto, 1989, Rios Neto and Cruz, 1990, Freitas Pinto and Rios Neto, 1990, Prado and Rios Neto, 1994) is explored to derive a control scheme completely based on Kalman filtering algorithms. The feedforward predictive control actions determination is viewed and treated as a stochastic linear parameter estimation problem. This allows the derivation of full non-parallel as well as, approximated parallel processing versions of the predictive control algorithm. Those algorithms are formally equivalent to versions of Kalman filtering algorithms used for feedforward neural network training (Rios Neto, 1997). Analysis of these control algorithms shows that they converge to the optimized solution of performance indexes formulated to guarantee smooth and reference trajectory-tracking controls.

2 Satellite Equations of Motion

A satellite configuration typical of body-stabilized vehicle technology was selected and is illustrated in Figure 1. The three body axes x, y, and z are represented as well as the nominal wheel orientation. Satellite mass properties and inertia properties are listed on Table 1. Also included on Table 1 are the adopted model for the expected disturbance torque due to solar pressure and the attitude pointing accuracy requirements. Development of the equations of motion follows the usual treatment employed for a rigid body with internal momentum. The adopted body system of reference, with origin fixed on the center of mass of the satellite, rotates with respect to an inertial frame at the orbit rate, ω_0 (7.28×10^{-5} rad/s). Nominal orientation of the double-gimbaled wheel is schematically illustrated in Figure 2. Euler's equations for the rigid satellite and wheel are given by

$$\boldsymbol{T} + \boldsymbol{G} = \frac{d\,\boldsymbol{h}}{dt} = \left[\frac{d\,\boldsymbol{h}}{dt}\right]_b + \boldsymbol{\omega}\, x\, \boldsymbol{h} \tag{2.1}$$

where $\boldsymbol{T}$ is the disturbance torque due to solar pressure and thrust misalignment, $\boldsymbol{G}$ is the gravity gradient torque, and $\boldsymbol{h}$ is the total angular momentum, including the wheel. The total angular momentum $\boldsymbol{h}$ is the sum of the angular momentum of the vehicle, $\boldsymbol{h}_v$, and the angular momentum of the wheel, $\boldsymbol{h}_w$, i.e.,

$$\boldsymbol{h} = \boldsymbol{h}_v + \boldsymbol{h}_w \tag{2.2}$$

Considering the unit vectors $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ along the body principal axes x, y, z the angular momentum components of the vehicle are given by

$$\boldsymbol{h}_v = I_x\omega_x\,\boldsymbol{i} + I_y\omega_y\,\boldsymbol{j} + I_z\omega_z\,\boldsymbol{k} \tag{2.3}$$

Referring to Figure 3, the wheel momentum components are

$$h_{wx} = \cos\delta \sin\gamma\, h_w \tag{2.4a}$$

$$h_{wy} = -\cos\delta \cos\gamma\, h_w \tag{2.4b}$$

$$h_{wz} = -\sin\delta\, h_w \tag{2.4c}$$

where δ, γ are the roll and yaw gimbal angles, respectively. Combining these expressions leads to

$$\begin{aligned} I_x\dot{\omega}_x = {} & T_x + G_x + \dot{\delta}\sin\delta\sin\gamma\, h_w - \dot{\gamma}\cos\delta\cos\gamma\, h_w - \cos\delta\sin\gamma\,\dot{h}_w \\ & - \omega_y\left(I_z\omega_z - \sin\delta\, h_w\right) + \omega_z\left(I_y\omega_y - \cos\delta\cos\gamma\, h_w\right) \end{aligned} \tag{2.5a}$$

$$\begin{aligned} I_y\dot{\omega}_y = {} & T_y + G_y - \dot{\delta}\sin\delta\cos\gamma\, h_w - \dot{\gamma}\cos\delta\sin\gamma\, h_w + \cos\delta\cos\gamma\,\dot{h}_w \\ & - \omega_z\left(I_x\omega_x + \cos\delta\sin\gamma\, h_w\right) + \omega_x\left(I_z\omega_z - \sin\delta\, h_w\right) \end{aligned} \tag{2.5b}$$

$$\begin{aligned} I_z\dot{\omega}_z = {} & T_z + G_z + \dot{\delta}\cos\delta\, h_w + \sin\delta\,\dot{h}_w \\ & - \omega_x\left(I_y\omega_y - \cos\delta\cos\gamma\, h_w\right) + \omega_y\left(I_x\omega_x + \cos\delta\sin\gamma\, h_w\right) \end{aligned} \tag{2.5c}$$

The angular velocities are related according to the equation

$$\boldsymbol{\omega} = \boldsymbol{\omega}_0 + \boldsymbol{\omega}_{\psi\theta\phi} \tag{2.6}$$

where $\boldsymbol{\omega}$, $\boldsymbol{\omega}_0$, and $\boldsymbol{\omega}_{\psi\theta\phi}$ represent the angular velocities: absolute, orbital and relative. Adopting the sequential Euler angles rotations of yaw ψ, pitch θ, and roll ϕ to move from the local vertical axes system to the body axes system, Eq. 2.6 can be expressed as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c\theta}\begin{bmatrix} c\theta & s\phi\, s\theta & s\theta\, c\phi \\ 0 & c\phi\, c\theta & -s\phi\, c\theta \\ 0 & s\phi & c\phi \end{bmatrix}\begin{bmatrix} \omega_x + \omega_0\, s\psi c\theta \\ \omega_y + \omega_0\left(c\psi c\phi + s\psi s\theta\, s\phi\right) \\ \omega_z + \omega_0\left(s\psi s\theta c\phi - c\psi s\phi\right) \end{bmatrix} \tag{2.7}$$

In the above equation, the letters "s" and "c" stand for the trigonometric functions sine and cosine, respectively. The set of equations 2.5a, 2.5b, 2.5c, and equations 2.7 are the non-linear ordinary differential equations that describe and governs the satellite attitude dynamic behavior. For the application of the proposed neural predictive control method, an artificial feedforward neural network must identify the plant, which is represented by this set of differential equations.

The gravity gradient torque components are given by (Kaplan, 1976)

$$G_x = \frac{3}{2}\omega_0^2 \sin 2\phi \cos^2\theta \left(I_z - I_y\right) \tag{2.8a}$$

$$G_y = \frac{3}{2}\omega_0^2 \cos\phi \sin 2\theta \left(I_z - I_x\right) \tag{2.8b}$$

$$G_z = \frac{3}{2}\omega_0^2 \sin\phi \sin 2\theta \left(I_x - I_y\right) \tag{2.8c}$$

3 Neural Predictive Control Algorithms

On this paragraph, the neural predictive control methodology is derived. Initially, consider the problem of controlling a dynamic system expressed by a set of n non-linear differential equations of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3.1}$$

For such a system, discrete time nonlinear input-output models can be obtained to predict the approximate system responses

$$\mathbf{y}(t_j) = \mathbf{f}\left(\mathbf{y}(t_{j-1}), \ldots, \mathbf{y}(t_{j-n_y}); \quad \mathbf{u}(t_{j-1}), \ldots, \mathbf{u}(t_{j-n_u})\right) \tag{3.2}$$

where the time at the instant t_j is given by $t_j = t + j\Delta t$.

The adopted neural predictive control scheme uses a feedforward neural network, which can uniformly and with the desired accuracy learn a mapping such as that expressed by Eq. (3.2) (Chen and Billings, 1992) to model the dynamic system represented by Eq. (3.1). This internal model neural network then provides the response model that can be used to determine smooth and reference trajectory tracking control actions by minimizing a predictive quadratic index of performance of the type usually adopted in predictive control schemes (Hunt, Sbarbaro, Zbikowski and Gawthrop, 1992, and Su and McAvoy, 1993). Consider then the performance index defined as

$$J = \frac{1}{2}\sum_{j=1}^{n}\left[\mathbf{y}_r(t_j) - \hat{\mathbf{y}}(t_j)\right]^T \mathbf{R}_y^{-1}(t_j)\left[\mathbf{y}_r(t_j) - \hat{\mathbf{y}}(t_j)\right] + \frac{1}{2}\sum_{j=0}^{n-1}\left[\mathbf{u}(t_j) - \mathbf{u}(t_{j-1})\right]^T \mathbf{R}_u^{-1}(t_j)\left[\mathbf{u}(t_j) - \mathbf{u}(t_{j-1})\right] \tag{3.3}$$

$\mathbf{y}_r(t_j)$ is the desired reference response; n defines the horizon over which the tracking errors and control increments are considered; $\boldsymbol{R}_y(t_j)$ and $\boldsymbol{R}_u(t_j)$ are positive definite weight matrices; $\hat{\mathbf{y}}(t_j)$ is the output of the feedforward neural network trained to approximately model the dynamic system of Eq. (3.1). Therefore, the dynamic system modeled by Eq. (3.1) can be formally represented by

$$\hat{\mathbf{y}}(t_j) = \hat{\mathbf{f}}\left(\hat{\mathbf{y}}(t_{j-1}), \ldots, \hat{\mathbf{y}}(t_{j-n_y}); \quad \mathbf{u}(t_{j-1}), \ldots, \mathbf{u}(t_{j-n_u}); \quad \hat{\mathbf{w}}\right) \tag{3.4}$$

where $\hat{\mathbf{w}}$ are the neural network weight parameters adjusted or estimated along training.

Thus, in summary, for solution of the resulting neural network predictive control problem it is needed:

(i) to choose a feedforward neural network with appropriate architecture and size, which in a process usually involving both off line and on line supervised training can learn from the dynamic system input output data sets how to represent the mapping of the nonlinear discrete model of the dynamic system;

(ii) to solve with respect to the control actions, on line and in a small fraction of Δt the nonlinear programming problem of minimizing an objective function constraining smooth and reference trajectory tracking control actions, as that given by Eq. (3.3), and subjected to the constrains expressed by Eq. (3.4).

3.1 Kalman Filtering Integrated Solution

Kalman filtering algorithms can be used to solve the problem of supervised training of the feedforward neural network used in the predictive control scheme. Versions of this algorithm with different levels of approximation can be found in the literature. Those versions may vary from full non-parallel algorithms, mostly suitable for off line use, to simplified parallel processing algorithms (Rios Neto, 1997) for on line applications.

Exploring previously developed and related results (Rios Neto and Pinto, 1989, Rios Neto and Cruz, 1990, Pinto and Rios Neto, 1990, Prado and Rios Neto, 1994), a method is proposed where the problem of determining the predictive control actions is also treated as one of stochastic optimal linear parameter estimation.

The method starts by assuming that the problem of control determination by minimization of Eq. (3.3) can be viewed in a more general stochastic framework as a stochastic parameter estimation problem expressed as

$$\hat{\mathbf{y}}(t_j) = \hat{\mathbf{f}}\left(\hat{\mathbf{y}}(t_{j-1}), \ldots, \hat{\mathbf{y}}(t_{j-n_y}); \quad \mathbf{u}(t_{j-1}), \ldots, \mathbf{u}(t_{j-n_u}); \quad \hat{\mathbf{w}}\right) \tag{3.5}$$

$$0 = \mathbf{u}(t_{j-1}) - \mathbf{u}(t_{j-2}) + \mathbf{v}_u(t_{j-1}) \tag{3.6}$$

$$E\left[\mathbf{v}_y(t_j)\right] = 0; \quad E\left[\mathbf{v}_y(t_j)\mathbf{v}_y^T(t_j)\right] = \mathbf{R}_y(t_j) \tag{3.7}$$

$$E\left[\mathbf{v}_u(t_j)\right] = 0; \quad E\left[\mathbf{v}_u(t_j)\mathbf{v}_u^T(t_j)\right] = \mathbf{R}_u(t_j) \tag{3.8}$$

where $j = 1,2,\ldots,n$; the output of the neural network $\hat{\mathbf{y}}(t_j)$ is represented by Eq. 3.4; $\hat{\mathbf{y}}(t_{j-1}),\ldots,\hat{\mathbf{y}}(t_{j-n_y})$ and $\hat{\mathbf{u}}(t_{j-1}),\ldots,\hat{\mathbf{u}}(t_{t-n_u})$ are past values of the system response and control action; the errors $\mathbf{v}_y(t_j)$ and $\mathbf{v}_u(t_j)$ are of uncorrelated components as well as uncorrelated for different values of t_j. A first consequence of this more general stochastic framework in the treatment of the problem is that the weight matrices, $\boldsymbol{R}_y$ and $\boldsymbol{R}_u$, in the objective function (Eq. (3.3)) have now the meaning of covariance matrices. This certainly facilitates their definition.

In order to re-iteratively treat the problem represented by Eqs. (3.5) and (3.6) as one of linear parameter estimation, one takes in an i[th] iteration the linearized approximation of Eq. (3.5), that is

$$\alpha(i)\left[\mathbf{y}_r(t_j) - \bar{\mathbf{y}}(t_j,i)\right] = \sum_{k=k_0}^{j-1}\left[\frac{\partial\hat{\mathbf{y}}(t_j)}{\partial\mathbf{u}(t_k)}\right]_{\bar{\mathbf{u}}(t_k,\,i)}\left[\mathbf{u}(t_k,i) - \bar{\mathbf{u}}(t_k,i)\right] + \mathbf{v}_y(t_j) \tag{3.9}$$

where $k_0 = max\,[\,0,\ (j-n_y-n_u)]$. The parameter $0<\alpha\le 1$ must be adjusted to guarantee the linear perturbation approximation hypothesis. The indicated partial derivatives are to be calculated using the *back-propagation rule* in the feedforward neural network that approximates the dynamic system response model (Chandran, 1994). Those observation type of conditions are then processed taking as a priori information, based on conditions of Eq. (3.6), and consistent with the linearized approximation in Eq. (3.9), the following:

$$\alpha(i)\left[\hat{\mathbf{u}}(t_{-1}) - \bar{\mathbf{u}}(t_l,i)\right] = \left[\hat{\mathbf{u}}(t_l,i) - \bar{\mathbf{u}}(t_l,i)\right] + \sum_{k=0}^{l}\mathbf{v}_u(t_k) \tag{3.10}$$

where *l=0,1,...,n-1 and i=1,2,...,I*; $\hat{\mathbf{u}}(t_{-1})$ is the estimated solution from last control step; $\alpha(i)\leftarrow\alpha(i+1)$; $\bar{\mathbf{u}}(t_l,i+1) = \hat{\mathbf{u}}(t_l,i)$, the approximated estimated value of $\mathbf{u}_l$ in the i[th] iteration; for i=1 an estimate or extrapolation of the last control step is to be used.

For *j=1,2,...,n* and *l=0,1,...,n-1,*the problem given by Eqs. (3.9) and (3.10) is one of stochastic linear parameter estimation. In a more compact notation, defining

$$\mathbf{U}(t,i) \equiv \left[\mathbf{u}^T(t_1,i):\ldots:\mathbf{u}^T(t_{n-1},i)\right]; \quad \hat{\mathbf{U}}_l(t_{-1}) \equiv \hat{\mathbf{u}}(t_{-1}) \tag{3.11}$$

it can equivalently be expressed as

$$\alpha(i)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right] = \mathbf{U}(t,i) - \bar{\mathbf{U}}(t,i) + \mathbf{V}_{\mathbf{u}}(t) \tag{3.12}$$

$$\alpha(i)\bar{\mathbf{Z}}^u(t,i) = \mathbf{H}^u(t,i)\left[\mathbf{U}(t,i) - \bar{\mathbf{U}}(t,i)\right] + \mathbf{V}_{\mathbf{y}}(t) \tag{3.13}$$

where the meanings of the compact notation variables are obvious by comparing Eqs. (3.12) and (3.13) with Eqs. (3.10) and (3.9), respectively. Using the Kalman filtering estimator, an algorithm for estimating the control actions is obtained which, for an i[th] typical iteration is given by

$$\hat{\mathbf{U}}(t,i) = \bar{\mathbf{U}}(t,i) + \alpha(i)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right] + \mathbf{K}(t,i)\,\alpha(i)\left\{\bar{\mathbf{Z}}^u(t,i) - \mathbf{H}^u(t,i)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right]\right\} \tag{3.14a}$$

$$\mathbf{K}(t,i) = \mathbf{R}_{\mathbf{u}}(t)\left(\mathbf{H}^{\mathbf{u}}(t,i)\right)^T\left[\mathbf{H}^{\mathbf{u}}(t,i)\mathbf{R}_{\mathbf{u}}(t)\left(\mathbf{H}^{\mathbf{u}}(t,i)\right)^T + \mathbf{R}_{\mathbf{y}}(t)\right]^{-1} \tag{3.14b}$$

$$\hat{\mathbf{R}}_{\mathbf{u}}(t,I) = \left[\mathbf{I}_{\mathbf{u}} - \mathbf{K}(t,I)\mathbf{H}^{\mathbf{u}}(t,I)\right]\mathbf{R}_{\mathbf{u}}(t) \tag{3.14c}$$

$$\bar{\mathbf{U}}(t,i+1) = \hat{\mathbf{U}}(t,i) \qquad \hat{\mathbf{U}}(t) = \hat{\mathbf{U}}(t,I) \qquad \alpha(i)\longleftarrow\alpha(i+1) \tag{3.14d}$$

where $\boldsymbol{R}_u(t)$, $\boldsymbol{R}_y(t)$, and $\hat{\mathbf{R}}_{\mathbf{u}}(t,I)$ are the error covariance matrices of $\boldsymbol{V}_u(t)$, $\boldsymbol{V}_y(t)$, and $[\,\hat{\mathbf{U}}(t,I)-\mathbf{U}(t)]$, respectively; $\boldsymbol{I}_u$ is an identity matrix. Algorithm convergence is guaranteed (Silva and Rios Neto, 2000, 2001).

An approximated version of the algorithm, which can be paralleled processed for each value of l=0,1,...,n-1, can now be obtained. To obtain the simplified version, the values of $\mathbf{U}_k(t,i)$, $k \neq l$, in Eq. (3.13) are approximated by $\overline{\mathbf{U}}_k(t,i)$. From these approximations results a problem which can be locally processed, and which can be expressed as

$$\alpha(i)\left[\hat{\mathbf{u}}(t_{-1}) - \overline{\mathbf{u}}(t_l,i)\right] = \mathbf{u}(t_l,i) - \overline{\mathbf{u}}(t_l,i) + \mathbf{V}_{ul}(t) \tag{3.15}$$

$$\alpha(i)\overline{\mathbf{Z}}^{\mathbf{u}}(t,i) = \mathbf{H}_l^{\mathbf{u}}(t,i)\left[\mathbf{u}(t_l,i) - \overline{\mathbf{u}}(t_l,i)\right] + \mathbf{V}_{\mathbf{y}}(t) \tag{3.16}$$

Use of Kalman filtering to solve this problem leads to

$$\hat{\mathbf{u}}(t_l,i) = \overline{\mathbf{u}}(t_l,i) + \alpha(i)\left[\hat{\mathbf{u}}(t_{-1}) - \overline{\mathbf{u}}(t_l,i)\right] + \mathbf{K}(t,l,i)\alpha(i)\left\{\overline{\mathbf{Z}}^{\mathbf{u}}(t,i) - \mathbf{H}_l^{\mathbf{u}}(t,i)\left[\hat{\mathbf{u}}(t_{-1}) - \overline{\mathbf{u}}(t_l,i)\right]\right\} \tag{3.17a}$$

$$\mathbf{K}(t,l,i) = \left\{\mathbf{R}_{ul}^{-1}(t) + \left(\mathbf{H}_l^{u}(t,i)\right)^T\mathbf{R}_y^{-1}(t)\mathbf{H}_l^{u}(t,i)\right\}^{-1}\left(\mathbf{H}_l^{u}(t,i)\right)^T\mathbf{R}_y^{-1}(t) \tag{3.17b}$$

$$\hat{\mathbf{R}}_{ul}(t,I) = \left[\mathbf{I}_u - \mathbf{K}(t,l,I)\mathbf{H}_l^{u}(t,I)\right]\mathbf{R}_{ul}(t) \tag{3.17c}$$

$$\hat{\mathbf{u}}(t_l) = \hat{\mathbf{u}}(t_l,I) \qquad \alpha(i) \longleftarrow \alpha(i+1) \tag{3.17d}$$

Convergence to a smooth control, which tracks the reference trajectory, is also guaranteed (Silva and Rios Neto, 2000, 2001).

4 Satellite Attitude Control Simulation Results

To simulate the geosynchronous satellite attitude control, considering the attitude perturbations represented by periodical solar pressure torques and gravity gradient terms, using the proposed methodology of neural predictive Kalman filtering, it is initially required to identify the plant by an artificial feedforward neural network. The first order differential equations that describe the plant's attitude are given by Eqs. 2.5 and 2.7. The gimbal angle rates, $\dot{\delta}$ (roll), and $\dot{\gamma}$ (yaw), and the wheel angular momentum rate, $\dot{h}_w$, were chosen as control parameters. The control vector is then defined as

$$\mathbf{u}(t) = \left[\dot{\delta}(t) \quad \dot{\gamma}(t) \quad \dot{h}_w(t)\right]^T \tag{4.1}$$

The neural predictive control algorithm was then employed to obtain the control laws to position and maintain the satellite attitude stabilized in the desired direction considering a simulation time interval of one day.

A feedforward multilayer perceptron neural network with an input layer of 12 neurons, a layer of 20 hidden neurons, and a layer of 9 output neurons was chosen to emulate the plant's dynamics behavior. The neural network is fully connected with a bias of +1 added to each layer. The neural network has the following parameters in the input layer

$$\mathbf{x}(t) = \left[\dot{\phi}(t_1) \quad \dot{\theta}(t_1) \quad \dot{\psi}(t_1) \quad \phi(t_1) \quad \theta(t_1) \quad \psi(t_1) \quad \delta(t_1) \quad \gamma(t_1) \quad h_w(t_1)\right]^T \tag{4.2}$$

The parameters update frequency Δt was chosen as one second (1.0 s). This update frequency is compatible with the system time of response and demonstrated to be adequate for satisfactory solution of the problem. Neural network training and test patterns were then created by numerical integration of Eqs. 2.5 and 2.7, considering the one-second integration time interval. Initial conditions for the state and control parameters were conveniently chosen according to the state space where the plant were supposed to operate. The neural network was then trained until mean output quadratic errors of the order $5\text{x}10^{-4}$ were obtained.

The geosynchronous satellite attitude pointing problem and attitude maintenance was then solved. For orbit simulation and as initial condition, it was considered that the satellite was already inserted in the correct orbit altitude but with attitude pointing errors of the order of magnitude of one degree in all three axes. For the simulation results presented on this paper it was considered that the initial pointing errors were: roll ϕ and pitch θ with -1°, and yaw ψ with +1°. All three rate angles were supposed to have zero initial angular velocities. The control double-gimbaled system was then required to make the necessary attitude corrections to put the satellite within the allowed orbit attitude accuracy requirements (pitch and roll of ± 0.05°, and yaw of 0.40°, Table 1).

The satellite attitude simulation was then performed by numerical integration of Eqs. 2.5 and 2.7. The control actions required to point the satellite in the desired direction were obtained from the predictive neural algorithm, Eqs. 3.17. Notice that this algorithm requires that a reference trajectory be specified. The Euler angles and their rates were chosen as the reference

parameters to be specified. This choice had as purpose to restrict Euler angle rates to small values, which still allowed for small angle corrections but with very smooth maneuvers. The reference trajectory was set as

$$\dot{\phi}_r(t) = \dot{\theta}_r(t) = \dot{\psi}_r(t) \equiv 0 \tag{4.3}$$

$$\phi_r(t) = \theta_r(t) = \psi_r(t) \equiv 0 \tag{4.4}$$

This choice resulted in control actions that positioned the satellite with smooth trajectory maneuvers, without overshooting and attitude high frequency maneuvers that usually result from the use of typical neural control systems. Simulation results are presented on Figures 4 through 10. The neural predictive control algorithm was employed with the restriction of only one-step ahead prediction with the update frequency of 1 Hz.

On Figure 4 it is shown the evolution with current time of the three-axis satellite attitude Euler angles. The choice of the reference trajectory combined with setting of the predictive control algorithm parameters (α) allowed for very smooth maneuvers in all three-axis without overshooting. Notice from the figure that the initial satellite point maneuver takes approximately 300 seconds to be completed. After this period of time, all three attitude angles are inside the precision bandwidth (dotted lines).

Figures 5 and 6 present the control angles laws as functions of the current time that were obtained from the neural predictive control algorithm. The graphs detail the control actions required to perform the initial satellite-pointing maneuver. After the pointing maneuver completion, all three controls remain close to zero, but with enough magnitude to maintain the satellite stabilized on the desired attitude. Figures 7 and 8 present the double-gimbaled angles and wheel angular momentum variation with current time, for the initial pointing maneuver. The initial maneuver requires approximately 1° roll and yaw double-gimbaled angles displacements. The initial variation of the wheel angular momentum is small, of the order of 0.05 N.m.s.

Figures 9 and 10 show the results for a one-day satellite attitude simulation. The double-gimbaled angles (roll δ and yaw γ) and wheel angular momentum h_w are plotted as function of the current time t. Apart from the initial point maneuver, the three variables related to the satellite control parameters, have a periodical variation with a sine shape format. The two angles have maximum amplitude of 1.5 degrees, and the angular momentum a variation of ±1.5 N.m.s.

5 Conclusions

The problem of controlling and stabilizing a geosynchronous satellite in the presence of periodic solar pressure torques and gravity gradient components is presented. The satellite was supposed to be already inserted in the desired orbit but with initial small attitude angles. The solution for the satellite point problem and one-day orbit stabilization was obtained by a neural predictive control scheme.

The use of Kalman filtering as a tool to derive neural predictive control algorithms was explored. Viewing the solution of the optimization problem of control action determination as one of stochastic parameter estimation reduced this problem to one formally equivalent to that of estimating the weights in feedforward neural network supervised training. This allowed a problem solution by employing Kalman filtering algorithms.

Two versions of algorithms were developed for the control determination: one where the approximation is the iterative approach due to linearization of equations, and where local parallel processing is not attained. The other one was an approximated version, but which attains local parallel processing and is intended for real time control schemes.

The results for one-day satellite attitude simulation demonstrated that the proposed methodology is capable of dealing with control problems of this type. The results show that a smooth control was obtained from the control algorithms. Differently from usual neural schemes, the satellite attitude corrections were made with three-axis maneuvers, without inducing overshooting and satellite shacking.

Although the algorithm can be used with a multiple steps horizon prediction, the application of only one step-ahead scheme was used and demonstrated to be adequate for the problem solution. Also, although small initial attitude angle corrections, of the order of ±1 degree was considered in this analysis, it should be pointed out that if required, and with adequate neural network training, larger values of angle corrections are possible.

6 References

[1] J. A. Silva, A. Rios Neto, Preliminary testing and analysis of an adaptive neural network training Kalman filtering algorithm, **Proceedings of IV Brazilian Conference on Neural Networks**, July 20-22, 1999, ITA, São José dos Campos, SP, Brazil, 247-251.

[2] J. A. Silva, A. Rios Neto, Neural predictive satellite attitude control based on Kalman filtering algorithms, **Proceedings 15th International Symposium of Spaceflight Dynamics**, June 26-30, 2000, Biarritz, France, 565-574.

[3] J. A. Silva, A. Rios Neto, Neural Predictive Flight Trajectory Control Based on Kalman Filtering Algorithms, **Proceedings of XVI Congresso Brasileiro de Engenharia Mecânica – COBEM**, November 26-30, 2001, Uberlândia, Brasil, ISBN: 85-85769-07-6, Vol. 6, 62-69.

[4] P. M. Tasinaffo, A. Rios Neto, Predictive Control With Mean Derivative Based Neural Euler Integrator Dynamic Model, **SBA Sociedade Brasileira de Automática**, v. 18 (2007), 94-105.

[5] S. Chen, S. A. Billings, Neural Networks for Nonlinear Dynamic System Modelling an Identification, **International Journal of Control**, 56(2) (1992), 319-346.

[6] P. S. Chandran, Comments on Comparative Analisys of Backpropagation and the Extended Kalman Filter for Training Multilayer Perceptron, **IEEE Transactions on Pattern Analysis and Machine Inteligence**, 16(8) (1994), 862-863.

[7] M. H. Kaplan, Modern Spacecraft Dynamics and Control, **John Wiley & Sons** (1976).

[8] A. Rios Neto, Stochastic Optimal Linear Parameter Estimation and Neural Nets Training in Systems Modeling, **RBCM- J. of the Braz. Soc. Mechanical Sciences**, XIX (2) (1997), 138-146.

[9] K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop, Neural Networks for Control Systems- A Survey, **Automatica**, 28(6) (1992), 1083-1112.

[10] H. T. Su, T. J. McAvoy, Neural Model Predictive Control of Nonlinear Chemical Procesesses, in **Proc. of the 1993 International Symposium on Inteligent Control**, Chicago, U.S.A. (1993), 358-363.

[11] A. Rios Neto, R. L. U. Freitas Pinto, A Stochastic Approach to Generate a Projection of the Gradient Type Method, in **Proc. VIII Congresso Latino Americano e Ibérico sobre Métodos Computacionais para Engenharia**, PUC/RJ, Rio de Janeiro, Brasil (1989), 331-345.

[12] R. L. U. Freitas Pinto, A. Rios Neto, An optimal linear estimation approach to solve systems of linear algebraic equations, **Journal of Computational and Applied Mathematics**, n. 33 (1990), 261-268.

[13] A. Rios Neto, J. J. Cruz, Design of a Stochastic Linear Discrete Regulator Based on an Optimal State Estimation Approach, **SBA: Controle e Automação**, 2(3) (1990), 151-157.

[14] A. F. B. A. Prado, A. Rios Neto, Suboptimal and Hybrid Numerical Solution Schemes for Orbit Transfer Maneuvers, **SBA: Controle e Automação**, 4(2) (1994), 82-88.

[15] D. G. Luenberger, Linear and Nonlinear Programming, **Addison-Wesley, Reading**, MA (1984), 261-262.

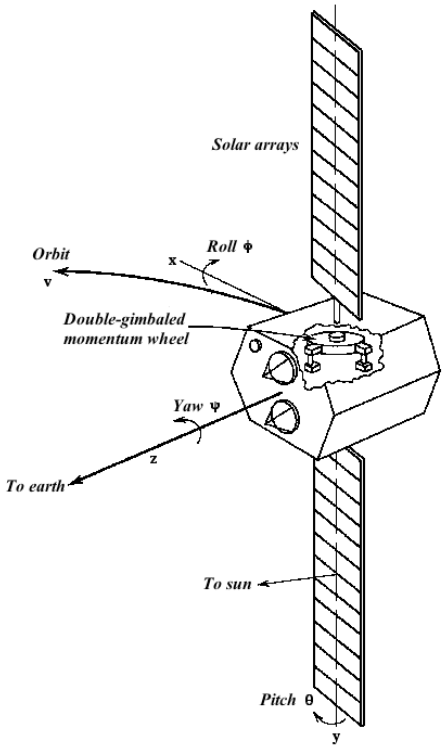

Figure 1 – Satellite Configuration.

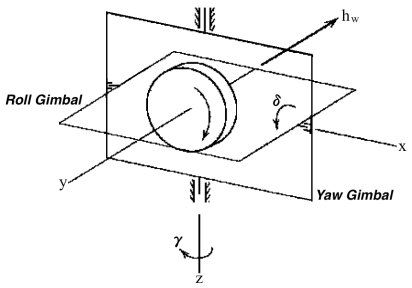

Figure 2 – Double-Gimbaled Control System.

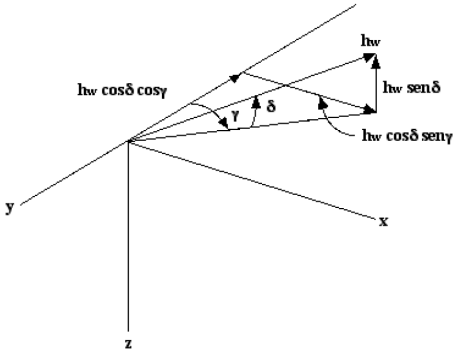

Figure 3 – Wheel Angular Momentum Components.

Table 1 – Satellite Specifications and Disturbance Torques

Satellite mass	716 kg
Moments of inertia	$I_x = I_z = 2000$ N.m.s^2 $\quad I_y = 400$ N.m.s^2
Attitude accuracy requirements	Pitch and Roll = 0.05° $\quad$ Yaw = 0.40°
Solar pressure torques (t=0 at 6 am or 6 pm orbital position)	$T_x = 2x10^{-5}(1 - 2.0\sin\omega_0 t)$ $N.m$ $T_y = 10^{-4}(\cos\omega_0 t)$ $N.m$ $T_x = -5x10^{-5}(\cos\omega_0 t)$ $N.m$

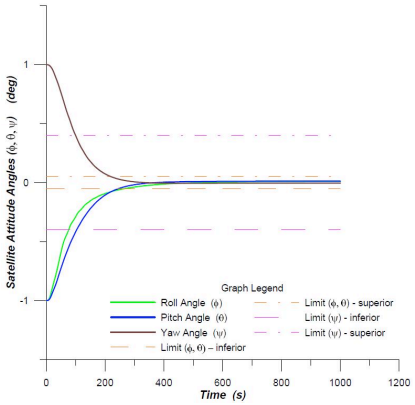

Figure 4 – Three-Axis Satellite Attitude Correction.

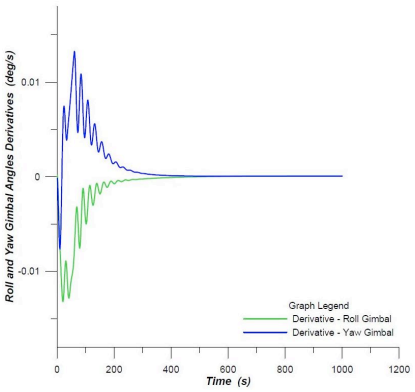

Figure 5 – Roll and Yaw Double-Gimbaled Controls.

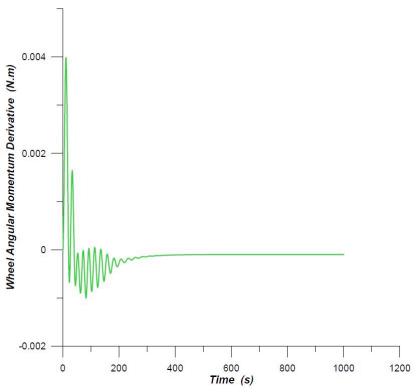

Figure 6 – Wheel Angular Momentum Control.

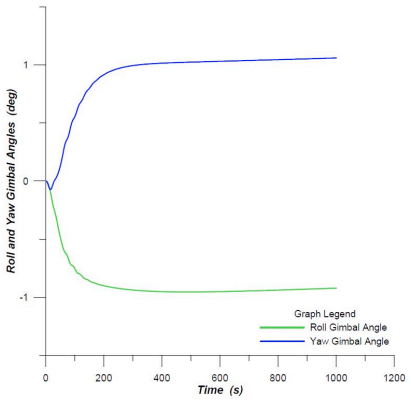

Figure 7 – Pointing Maneuver: Double-Gimbaled Roll and Yaw Angles.

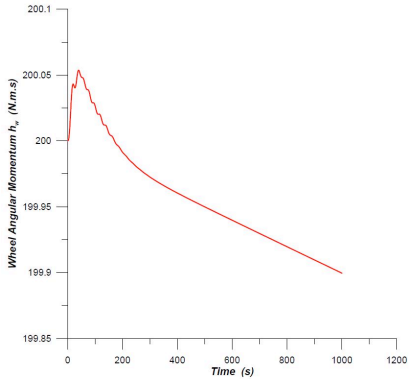

Figure 8 – Pointing Maneuver: Wheel Angular Momentum Variation.

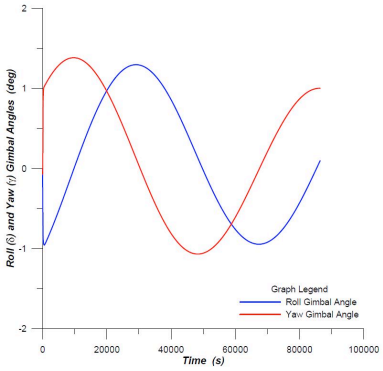

Figure 9 – Attitude Stabilization: Double-Gimbaled Roll and Yaw Angles Variation.

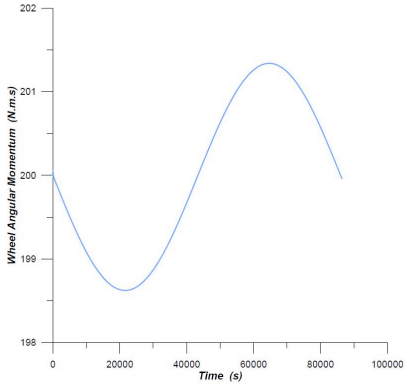

Figure 10 – Attitude Stabilization: Wheel Angular Momentum Variation.