NEURAL PREDICTIVE FLIGHT TRAJECTORY CONTROL BASED ON KALMAN FILTERING ALGORITHMS

Jaime Augusto DA SILVA
VPI/DTE/GET – Gerência de Sistemas Eletro-Eletrônicos
Empresa Brasileira de Aeronáutica S.A. – EMBRAER
Av. Brigadeiro Faria Lima, 2170
12227-901 - São José dos Campos, SP, Brasil
e-mail: jaimes@directnet.com.br

Atair RIOS NETO
Instituto Nacional de Pesquisas Espaciais – INPE
Divisão de Mecânica Espacial e Controle – DMC
Avenida dos Astronautas, 1758
12227-010 – São José dos Campos, SP, Brasil
e-mail: atairrn@uol.com.br

ABSTRACT *An artificial neural network predictive control scheme is considered for satellite launch vehicle real time control. Kalman filtering algorithms are used (i) to solve the spacecraft ascent optimization problem and define a reference trajectory, (ii) to train the associated feedforward neural network modeling the dynamics of the plant and (iii) to estimate the control actions. It is shown that the optimization of a predictive quadratic performance functional, used to determine the discrete control actions, can be viewed and treated, in a typical iteration, as a stochastic optimal linear parameter estimation problem. The algorithms obtained are shown to be the result of application of Newton's method to appropriate control optimization functionals that provide solutions that converge to smooth and reference tracking controls. The proposed scheme is then applied to a three-degrees of freedom nonlinear flight trajectory control of a satellite launch vehicle. A vehicle reference ascent trajectory is initially obtained by parameterization of the control history and by using a nonlinear programming technique similar to the one used to solve the control problem Results of simulations and tests for the situation of trajectory control show an excellent performance of the proposed scheme.*

Key Words: nonlinear systems, predictive control, neural networks, satellite launcher, trajectory optimization

1. Introduction

Although practical processes involve nonlinear behavior, due to implementation difficulties, specially the real time control determination, most predictive control algorithms are based on a linear model of the process. As a result, they do not give satisfactory control performance when the controlled process is highly nonlinear. Recently, it has been proved that multilayer feedforward neural networks can model and approximate nonlinear functions arbitrarily well (Cybenko, 1989, Hornik *et al*., 1989, Funahashi, 1989). Based on this fact, a large number of identification and control structures that use neural networks have been proposed (Narendra and Parthasarathy, 1990, Sanner and Slotine, 1992, Chen and Billings, 1992, Soloway and Harley, 1997, Liu *et al*., 1998). For neural network training, many authors have explored recursive least squares (Chen and Billings, 1992) and Kalman filtering theory (Singhal and Wu, 1989, Watanabe *et al*., 1991, Iiguni *et al*., 1992, Chen and Ögmen, 1994, Lange, 1995, Rios Neto, 1997) to develop both off line and on line neural network supervised training algorithms. It has been observed that, generally, these algorithms furnish better performance than the usual backpropagation algorithm (e.g., Da Silva and Rios Neto, 1999).

When a predictive control scheme is considered, besides training the neural network that will model the plant dynamics, one needs to solve an optimization problem to get the control actions (Su and McAvoy, 1993, Mills *et al*., 1994, Liu *et al*., 1998, Soloway and Harley, 1997, Zhu *et al*., 1999). In this case, control performance indexes are generally minimized using nonlinear programming techniques.

A different approach is used here in the sense that stochastic optimal parameter estimation theory is used to design a neural predictive control Kalman filtering algorithm. As a result, the problems of neural network training and predictive control are viewed and treated in an integrated way as stochastic optimal linear parameter estimation problems.

In the second part of this paper, the feasibility of the proposed algorithm in the design of a satellite launcher's guidance control scheme is evaluated. A tri-dimensional guidance and control of a satellite solid-fuel launcher is investigated. Initially a near optimal reference trajectory is obtained by application of a Stochastic Gradient Projection Method. Neural training patterns are then obtained and feedforward neural networks are used to identify and model the nonlinear behavior of the plant. Then, the predictive control algorithm is used to obtain the real time control actions necessary for reference trajectory tracking and implementation of the guidance scheme. Results of simulations are very satisfactory with small tracking errors and orbital elements very close to nominal values at injection point.

2. Control problem and neural network predictors

It is proposed control a dynamic system represented by

$$\dot{\mathbf{x}} \;=\; \mathbf{f}(\mathbf{x},\mathbf{u}) \tag{1}$$

for which discrete time nonlinear input-output models can be taken to predict approximate responses

$$\mathbf{y}(t_j) \quad \mathbf{f}_n\Big(\mathbf{y}(t_{j-1}),\ \ldots, \mathbf{y}(t_{j-n_y});\ \mathbf{u}(t_{j-1}),\ \ldots, \mathbf{u}(t_{j-n_u})\Big) \tag{2}$$

where $t_j = t + j\Delta t$.

The adopted neural predictive control scheme uses a feedforward neural network which can uniformly and with the desired accuracy learn a mapping as that of Eq. (2) (Chen and Billings, 1992) to model the dynamic system of Eq. (1). The fundamental idea in predictive control is to predict the vector of future tracking errors and minimize its norm over a given number of future control moves. To accomplish this, the internal model neural network will provide the response model that will be used to determine a smooth and reference trajectory tracking control actions. These actions are obtained by minimizing a predictive quadratic index of performance of the type usually adopted in predictive control schemes

$$J \;=\; \frac{1}{2}\left\{\sum_{j=1}^{n}\left(\mathbf{y}_r(t_j) - \hat{\mathbf{y}}(t_j)\right)^T \mathbf{R}_y^{-1}(t_j)\left(\mathbf{y}_r(t_j) - \hat{\mathbf{y}}(t_j)\right) \;+\; \sum_{j=0}^{n-1}\left(\mathbf{u}(t_j) - \mathbf{u}(t_{j-1})\right)^T \mathbf{R}_u^{-1}(t_j)\left(\mathbf{u}(t_j) - \mathbf{u}(t_{j-1})\right)\right\} \tag{3}$$

where as before $t_j = t + j\Delta t$; $\mathbf{y}_r(t_j)$ is the reference response, n defines the horizon over which the tracking errors and control increments are considered; $\mathbf{R}_y(t_j)$ and $\mathbf{R}_u(t_j)$ are positive definite weight matrices, and $\hat{\mathbf{y}}(t_j)$ is the output of the feedforward neural network, trained to approximately model the dynamic system of Eq. (1) and which can be formally represented as

$$\hat{\mathbf{y}}(t_j) \;=\; \hat{\mathbf{f}}\Big(\hat{\mathbf{y}}(t_{j-1}),\ \cdots,\ \hat{\mathbf{y}}(t_{j-n_y});\ \mathbf{u}(t_{j-1}),\ \cdots,\ \mathbf{u}(t_{j-n_u});\ \hat{\mathbf{w}}\Big) \tag{4}$$

Here $\hat{\mathbf{w}}$ represents the neural network parameter vector, adjusted or estimated along training. Thus, in summary, for the solution of the resulting neural predictive control problem it is needed:

(i) to choose a feedforward neural network with appropriate architecture and size. Then, in a process usually involving both off line and on line supervised training, learn from dynamic system input-output data sets, how to represent the mapping of the considered nonlinear discrete model (Eq. 2);

(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 as that in Eq. (3) subjected to the constraint of Eq. (4).

3. Kalman filtering integrated solution

The problem of supervised training of the feedforward neural network used in the predictive control scheme can be treated using Kalman filtering algorithms. Versions of these algorithms, with different levels of approximation, can be found in the literature. These 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 use. Here, a method is proposed where the problem of determining the predictive control actions is also treated as one of stochastic optimal linear parameter estimation. This allows the derivation and use, in a given iteration, of the same Kalman filtering type of algorithms as in the neural network training.

The method starts by assuming that the problem of control determination can be viewed, in a more general stochastic framework, as a stochastic parameter estimation problem such as

$$\mathbf{y}_r(t_j) \;=\; \hat{\mathbf{f}}\Big(\hat{\mathbf{y}}(t_{j-1}),\ \cdots,\ \hat{\mathbf{y}}(t_{j-n_y}),\ \mathbf{u}(t_{j-1}),\ \cdots,\ \mathbf{u}(t_{j-n_u}),\ \hat{\mathbf{w}}\Big) \;+\; \mathbf{n}_y(t_j) \tag{5}$$

$$0 \;=\; \mathbf{u}(t_{j-1}) \;-\; \mathbf{u}(t_{j-2}) \;+\; \mathbf{n}_u(t_{j-1}) \tag{6}$$

$$E\left[\mathbf{n}_y(t_j)\right] \;=\; 0\,; \qquad E\left[\mathbf{n}_y(t_j)\mathbf{n}_y^T(t_j)\right] \;=\; \mathbf{R}_y(t_j)\,; \tag{7}$$

$$E\left[\mathbf{n}_u(t_j)\right] \;=\; 0\,; \qquad E\left[\mathbf{n}_u(t_j)\mathbf{n}_u^T(t_j)\right] \;=\; \mathbf{R}_u(t_j)\,; \tag{8}$$

where $j = 1,2,..,n$. The errors $\mathbf{v}_y(t_j)$ and $\mathbf{v}_u(t_j)$ are considered to be constituted of uncorrelated components as well as uncorrelated for different values of t_j. A first consequence of this more general stochastic framework is that the weight matrices in the objective function, Eq. (3), have now the meaning of covariance matrices. This certainly facilitates their definition.

In order to iteratively solve the problem of Eqs. (5) and (6) as one of linear parameter estimation, one takes in a given ith iteration the linearized approximation of Eq. (5)

$$\alpha(i)\left[\mathbf{y}_r(t_j) - \bar{\mathbf{y}}(t_j, i)\right] = \sum_{k=k_0}^{j-1} \left[\frac{\partial \hat{\mathbf{y}}(t_j)}{\partial \mathbf{u}(t_k)}\right]_{\{\bar{\mathbf{u}}(t_k, i)\}} \left[\mathbf{u}(t_k, i) - \bar{\mathbf{u}}(t_k, i)\right] + \mathbf{n}_y(t_j) \tag{9}$$

where $k_0 = \max\left[0, (j - n_y - n_u)\right]$. The parameter α, $0 < \alpha(i) \le 1$, is to be adjusted to guarantee the linear perturbation approximation hypothesis. The partial derivatives indicated above, are to be calculated recursively using the backpropagation rule and the trained feedforward neural network (Chandran, 1994, Soloway and Haley, 1997). This observation type of condition is then processed, taking as a priori information, based on Eqs. (6) and (9), the following equation (Rios Neto and Da Silva, 2000)

$$\alpha(i)\left[\hat{u}(t_l) - \bar{u}(t_l, i)\right] = \left[u(t_l, i) - \bar{u}(t_l, i)\right] + \sum_{k=0}^{l} n_u(t_k) \tag{10}$$

where *l = 0,1, ..., n-1* and *i=1,2, ..., I*. The control variable $\hat{\mathbf{u}}(t_{-1})$ is the estimated solution from last control step and for a new iteration it is assumed that: $\alpha(i) \leftarrow \alpha(i+1)$ and $\bar{\mathbf{u}}(t_l, i+1) = \hat{\mathbf{u}}(t_l, i)$. For *i*=1 estimates or extrapolations of the control variables are used.

For *j=1,2, …, n* and *l=0,1, ..., n-1*, the problem represented by Eqs. (9) and (10) is one of stochastic linear parameter estimation. In a more compact notation,

$$\mathbf{U}(t,i) \equiv \left[\mathbf{u}^T(t_0,i) : \mathbf{u}^T(t_1,i) : \cdots : \mathbf{u}^T(t_{n-1},i)\right]^T; \qquad \hat{\mathbf{U}}_l(t_{-1}) \equiv \hat{\mathbf{u}}(t_{-1})$$

$$\alpha(i)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right] = \mathbf{U}(t,i) - \bar{\mathbf{U}}(t,i) + \mathbf{V}_u(t) \tag{11}$$

$$\alpha(i)\bar{\mathbf{Z}}^u(t,i) = \mathbf{H}^u(t,i)\left[\mathbf{U}(t,i) - \bar{\mathbf{U}}(t,i)\right] + \mathbf{V}_y(t) \tag{12}$$

The meanings of the compact notation variables becomes obvious if Eqs. (11) and (12) are identified with Eqs. (10) and (9), respectively. Using a Kalman filtering estimator, results

$$\hat{\mathbf{U}}(t,i) = \bar{\mathbf{U}}(t,i) + \alpha(i)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right] + \mathbf{K}(t,i)\alpha(i)\left\{\bar{\mathbf{Z}}^u(t,i) - \mathbf{H}^u(t,i)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right]\right\} \tag{13}$$

$$\mathbf{K}(t,i) = \mathbf{R}_u(t)\mathbf{H}^{uT}(t,i)\left[\mathbf{H}^u(t,i)\mathbf{R}_u(t)\mathbf{H}^{uT}(t,i) + \mathbf{R}_y(t)\right]^{-1} \tag{14}$$

$$\hat{\mathbf{R}}_u(t,I) = \left[\mathbf{I}_u - \mathbf{K}(t,I)\mathbf{H}^u(t,I)\right]\mathbf{R}_u(t) \tag{15}$$

$$\bar{\mathbf{U}}(t, i+1) = \hat{\mathbf{U}}(t,i); \quad \hat{\mathbf{U}}(t) = \hat{\mathbf{U}}(t,I); \quad \alpha(i) \leftarrow \alpha(i+1) \tag{16}$$

The matrices $\mathbf{R}_u(t)$, $\mathbf{R}_y(t)$ and $\hat{\mathbf{R}}_u(t,I)$ are error covariance matrices of $\mathbf{V}_u(t)$, $\mathbf{V}_y(t)$ and $\left(\hat{\mathbf{U}}(t,I) - \mathbf{U}(t)\right)$, respectively and $\mathbf{I}_u$ is an identity matrix. A way of showing that convergence is guaranteed is by considering the algorithm in the equivalent form (Eq. (13a))

$$\hat{\mathbf{U}}(t,i) = \bar{\mathbf{U}}(t,i) + \left[\mathbf{R}_u^{-1}(t) + \mathbf{H}^{uT}(t,i)\mathbf{R}_y^{-1}(t)\mathbf{H}^u(t,i)\right]^{-1}\alpha(i)\left\{\mathbf{R}_u^{-1}(t)\left[\hat{\mathbf{U}}(t_{-1}) - \bar{\mathbf{U}}(t,i)\right] + \mathbf{H}^{uT}(t,i)\mathbf{R}_y^{-1}(t)\bar{\mathbf{Z}}^u(t,i)\right\} \tag{13a}$$

and noticing that this is the result of applying Newton's Method to the functional (Luenberger, 1984)

$$J_P = \frac{1}{2}\left\{\mathbf{Z}^{uT}(t)\mathbf{R}_y^{-1}(t)\mathbf{Z}^u(t) + \left[\mathbf{U}(t) - \hat{\mathbf{U}}(t_{-1})\right]^T\mathbf{R}_u^{-1}(t)\left[\mathbf{U}(t) - \hat{\mathbf{U}}(t_{-1})\right]\right\} \tag{17}$$

In a way completely analogous to that adopted for the problem of neural network training (Rios Neto, 1997), one can obtain approximated versions of Eqs. (11) and (12) which can be processed in parallel for each value of *l=0,1, ..., n-1*. To get this simplified version one can approximate the values of $\mathbf{U}_k(t,i),\ k \neq l,$ in Eq. (12) by $\overline{\mathbf{U}}_k(t,i)$. These approximations lead to a problem, which can be locally processed, and which also converges to a smooth control that tracks the reference trajectory.

4. Flight trajectory control and guidance

As an application example, a control scheme of a three degrees of freedom guidance of a satellite launch vehicle is investigated. This scheme comprises trajectory optimization and definition of a reference trajectory, identification of the plant by a multilayer perceptron neural network and plant control by using the proposed predictive control method in a closed loop guidance scheme. The guidance scheme is supposed to act during the complete phase of vehicle ascent flight. The principle applied here is to observe vehicle's center of mass deviations in position and velocity from a suboptimal reference trajectory and, calculate control actions in order to ensure nominal position and circular velocity at the injection point. Neural network training patterns are created by appropriately perturbing the reference trajectory represented by the history of the angles of yaw and pitch as functions of time. The projected gradient method (Madeira and Rios Neto, 2000) is initially used to obtain a suboptimal reference trajectory. Then, an algorithm similar to the one presented here (Da Silva and Rios Neto, 1999 and 2000) is employed to train a multilayer perceptron neural network that models and is used as the plant emulator in the predictive control algorithm scheme. Four neural networks are used to model the plant with each net modeling one of the four flight phases represented by the flight of the four stages.

4.1 Vehicle and mission

The launcher is supposed to be a four stages solid fuel LEO rocket capable of placing a payload of approximately 190 kilos at a 750 kilometers circular orbit with an orbital inclination of 25 degrees. At take off the vehicle has an estimated mass of approximately 50 tons and each solid fuel rocket has a burning time of approximately 60 seconds. The mission considered here consists of launching a payload from the launch site of Alcântara, Brazil, which is located at an altitude of 43,01 meters above sea level, -2,3173 degrees of latitude and –44.3677 degrees of longitude.

4.2 Equations of motion

A set of seven ordinary nonlinear differential equations are used to describe translational motion of vehicle's center of mass. Considering that r represents the modulus of the position vector; u, the radial velocity; v, the tangential velocity; T, d_x, d_y, d_z, l_x, l_y, l_z, the propulsion, drag and lift forces components; φ, θ and ψ, Euler angles; m, vehicle's current mass; μ, the gravitational constant; and, α and β as the control angles of yaw and pitch, the set of equations that defines the vehicle's motion are given by (Madeira and Rios Neto, 2000):

$$\dot{r} = u \tag{18}$$

$$\dot{u} = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T}{m} sen\beta\cos\alpha + \frac{d_x + l_x}{m} \tag{19}$$

$$\dot{v} = -\frac{uv}{r} + \frac{T}{m}\cos\beta\cos\alpha + \frac{d_y + l_y}{m} \tag{20}$$

$$\dot{\varphi} = \frac{v}{r} - \frac{Tsen\alpha + d_z}{mv} sen\varphi\cot\theta \tag{21}$$

$$\dot{\theta} = \frac{Tsen\alpha + d_z}{mv}\cos\varphi \tag{22}$$

$$\dot{\psi} = \frac{Tsen\alpha + d_z}{mv}\frac{sen\varphi}{sen\theta} \tag{23}$$

$$\dot{m}(t) = -\gamma_i \tag{24}$$

Vehicle's aerodynamic, propulsion and mass characteristics used for trajectory simulations are the same as those found in Madeira (1996).

4.3 Trajectory optimization

Application of the predictive control algorithm requires that a reference trajectory be specified. This reference trajectory was taken as the suboptimal trajectory obtained by application of a Stochastic Gradient Projection Method, using the same approach and methodology as done by Madeira and Rios Neto (2000). The control variables, chosen as the angles of pitch and yaw (angles α and β, respectively), are defined by line segments whose nodes are parameters to be optimized. For each control variable, four line segments are used to define flight of the first stage, two segments for the second stage, one segment for the third stage and four segments for the fourth stage. Considering that the satellite mass and phase coast time are also parameters to be optimized, a total of twenty-eight parameters are subjected to the optimization process. One inequality constraint related to aerodynamic loads is added to the system under optimization in order to assure vehicle's structural integrity. This constraint can be expressed as

$$q\alpha_l - (q\alpha_l)_{\max} \leq 0 \qquad (25)$$

where q represents dynamic pressure and α_l is the angle of attack. A limit of 3000 N.rad/m^2 for this constraint should be observed during atmospheric flight.

Results of optimization showed that for the proposed vehicle's configuration, it is possible to place a payload of approximately 194.01 kilos at a circular orbit of 750 kilometers of altitude and inclination of 25 degrees. Suboptimal histories of yaw and pitch control angles as functions of current time are plotted in Figures 1 and 2 (continuous curves).

4.4 System identification

Four feedforward neural networks of same structure identified the plant dynamics, represented by the set of seven nonlinear differential equations, Equations 18-24. Each net was chosen with 8 neurons at the input layer, 30 at the hidden layer and 6 at the output layer, all layers having a bias of +1. Patterns used for training were created by numerical integration of the set of differential equations, Equations 18-24. Control variables, α and β, were obtained from the suboptimal reference trajectory with values conveniently perturbed, in order to obtain trajectories covering a wide state phase. In order to provide a surplus of energy to compensate for tracking errors, due to unmodeled effects and approximation errors a satellite mass of 170 kilos, which is less than the suboptimal value, was used for trajectory simulations. A set of 250 trajectories was simulated with neural patterns obtained with a 1-second interval frequency from each trajectory simulated. This interval also defines the frequency of control actualization in the guidance scheme.

Neural patterns were obtained for each trajectory simulated by collecting a set of values for variables $\{h, u, v, \varphi, \theta, \psi, \alpha, \beta\}$ and collection of variables $\{h, u, v, \varphi, \theta, \psi\}$ obtained after 1 second interval. Here, variable h represents the geometrical altitude. With a set of conveniently chosen dimensional variables, those two groups of variables were further converted into values ranging from –1 and +1 and represented the inputs and desired outputs of the neural networks. From the patterns obtained of the 250 trajectories simulated, patterns of 200 trajectories were used as neural network training patterns and patterns of 50 trajectories were used for neural training verifications. Neural networks training were performed until mean square errors of less than 1.0×10^{-6} were obtained. Each of the four neural networks was chosen to model flight trajectory of a given stage.

4.5 Guidance scheme

The predictive control algorithm was then employed in a scheme of guidance and control of the satellite launch vehicle. The reference trajectory to be tracked was defined by variables: geometric altitude, h, radial velocity, u, tangential velocity, v, and, orbital inclination, θ and is the same suboptimal trajectory obtained previously. Equations 13-16 were then used to obtain the control necessary for trajectory tracking. Results of simulations are presented in Figures 1-6. Figures 1 and 2 present the control histories obtained for the optimized trajectory (continuous curve) and for the guidance scheme (doted curve). Note that since there is a surplus of energy left, it is compensated in the guidance scheme with maneuvers in yaw, most significant during fourth stage flight. Figures 3a and 3b show variation of altitude and altitude tracking error as functions of current time. A tracking error of only 751 meters were observed at burnout of the forth stage. On Figures 4a, 4b, 5a and 5b are shown the behavior of velocity components and tracking errors as functions of current time. It is observed a small residual error of approximately 1.45 m/s on the radial velocity and a 0.46 m/s on the tangential velocity. Figures 6a and 6b show variation of orbital inclination and tracking error. The final orbit has an inclination of 24.91º, which is very close to the nominal value of 25º.

Calculations of orbital elements at the injection point shows an orbit with a semi-major axis of 7127769 m, eccentricity of 0.000194, radius of perigee of 747996m and radius of apogee of 750765 m. Those values are very close to nominal values. In conclusion, the guidance scheme based on the predictive control algorithm, for the condition of a satellite mass whose value is inferior to the suboptimal value showed good results.

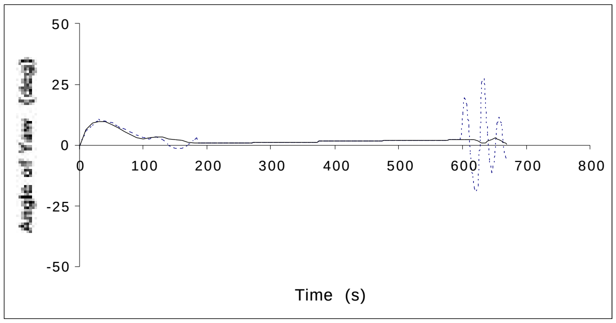

Fig. 1 - Angle of yaw α versus current time.

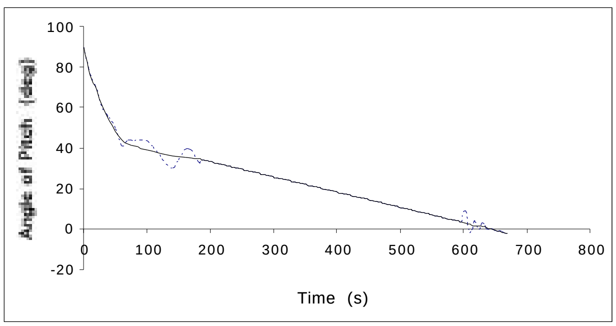

Fig. 2 - Angle of pitch β versus current time.

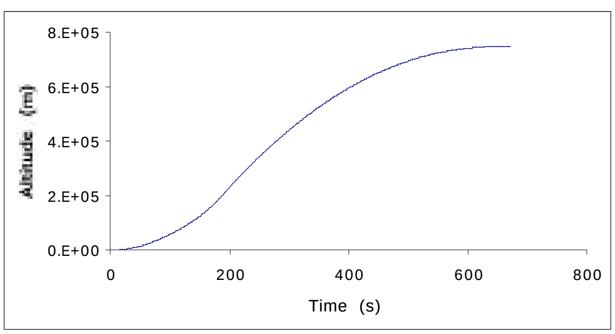

Fig. 3a - Altitude h versus current time.

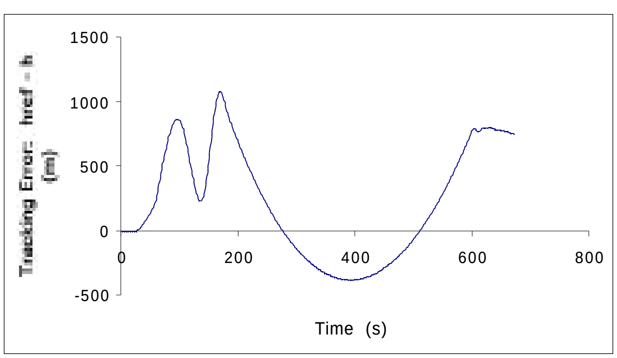

Fig. 3b - Tracking error in altitude.

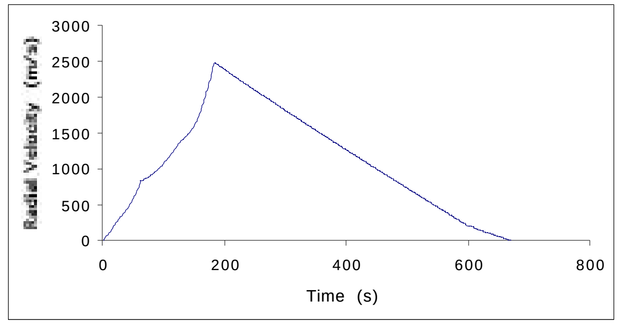

Fig. 4a - Radial velocity u versus current time.

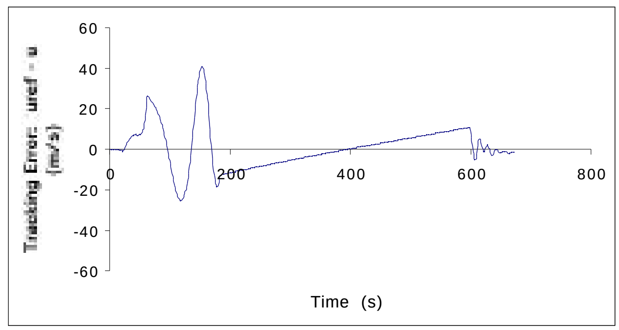

Fig. 4b - Tracking error in radial velocity.

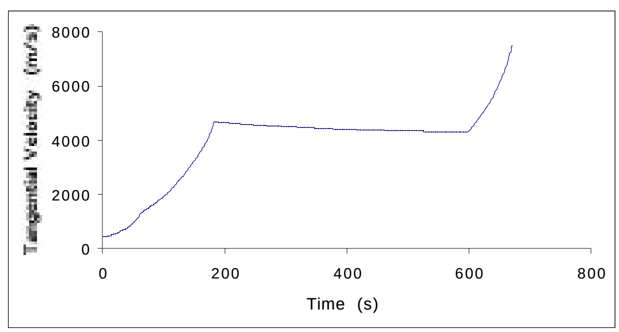

Fig. 5a - Tangential velocity v versus current time.

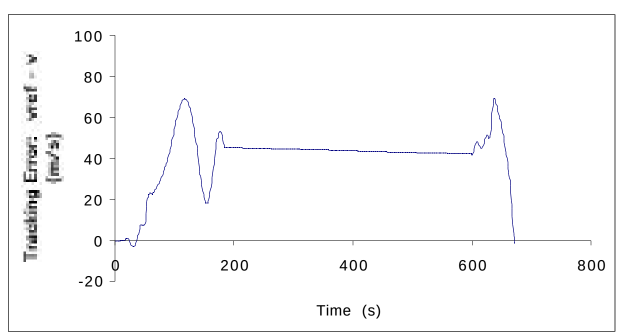

Fig. 5b - Tracking error in tangential velocity.

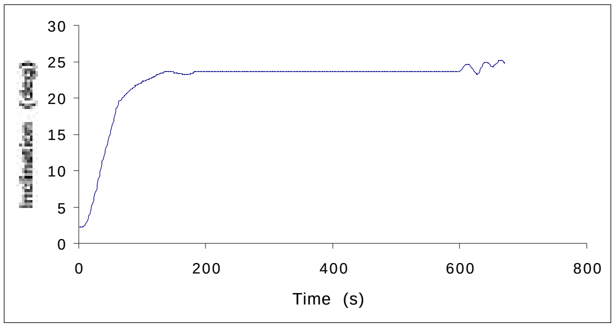

Fig. 6a - Inclination θ versus current time.

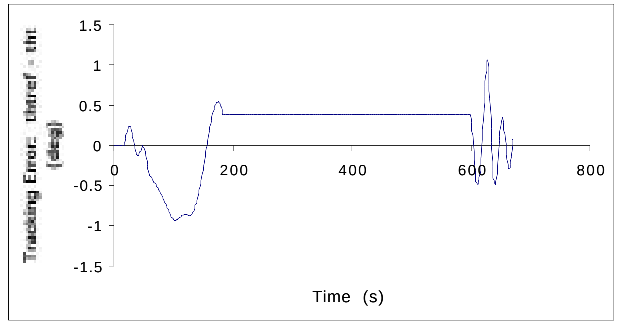

Fig. 6b - Tracking error in inclination.

5. Conclusion

The use of Kalman filtering as an optimal parameter estimation tool allows design of a method to solve neural predictive control problems. This method can be shown to converge to Newton's Method solution of minimizing functionals which constraint smooth and reference trajectory tracking controls.

A numerically simulated test was considered by applying the proposed methodology to the control of a satellite launch vehicle. Nonlinear equations of motion were used, neural network training patterns were created, feedforward neural networks were trained and then, the control actions were obtained for a guidance vehicle control scheme. At injection point it was observed a final orbit very close to the nominal orbit with small final tracking errors. The results are very satisfactory and show that a guidance scheme based on a predictive control method is a viable possibility.

The simulation demonstrated that the predictive control algorithm convergence performance is equivalent to that of the correspondent neural network training Kalman filtering algorithms because they are completely similar algorithms used to solve numerically equivalent parameter estimation problems.

Acknowledgement

This research has been partially supported by CNPq and FAPESP grants.

6. References

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

Chen, S., Billings, S. A., 1992, "Neural networks for nonlinear dynamic system modeling and identification", *International Journal of Control*, Vol. 56, No. 2, pp. 319-346.

Chen, G. and Ögmen, H., 1993, "Modified Extended Kalman Filtering for Supervised Learning", *International Journal of Systems Sciences*, 24(6), pp. 319-346.

Cybenko, G., 1989, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, Vol. 2, No. 3, pp. 303-314.

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

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

Funahashi, K., 1989, "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, Vol. 2, No. 3, pp. 183-192.

Hornik, K., Stinchcombe, M. White, H., 1989, "Multilayer feedforward networks are universal approximators", *Neural Networks*, Vol. 2, No. 5, pp. 359-366.

Iiguni, Y., Sakai, H., Tokumaru, H., 1992, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter", *IEEE Transactions on Signal Processing,* 40(4), pp. 959-966.

Lange, F., 1995, "Fast and accurate training of multilayer perceptrons using an extended Kalman filter (EKFNet)", *Internal paper, DLR (German Institute Aerospace Research Establishment), Institute for Robotics and Systems Dynamics*.

Liu, G. P., Kadirkamanathan, V., Billings, S. A., 1998, "Predictive control for nonlinear systems using neural networks", *International Journal of Control*, Vol. 71, No. 6, pp. 1119-1132.

Luenberger, D. G., 1984, "*Linear and Nonlinear Programming*", Addison-Wesley, Reading, MA, pp. 261-262.

Madeira, F., Rios Neto, A., 2000, “Guidance and Control of a Launch Vehicle Using a Stochastic Gradient Projection Method”, *Automatica,* 36(2000), pp.427-438.

Madeira, F.,1996, “Guiagem e Controle Não-Linear Sub-Ótimo de Veículos Lançadores de Satélites em Malha Fechada e em Tempo Quase Real”. PhD Thesis in Mechanical and Aeronautical Engineering, ITA, São José dos Campos, Brazil.

Mills, P. M., Zomaya, A. Y., Tadé, M. O., 1994, “Adaptive model-based control using neural networks”, *International Journal of Control*, Vol. 60, No. 6, pp. 1163-1192.

Narendra, K. S, Parthasarathy, K., 1990, “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, March, pp. 4-27.

Rios Neto, A., 1997, “Stochastic optimal parameter estimation and neural nets training in systems modeling”, *RBCM – Journal of the Brazilian Society of Mechanical Sciences*, Vol. XIX, No. 2, pp. 138-146.

Rios Neto, Da Silva, J. S., 2000, “Neural Predictive Control Based on Kalman Filtering Algorithms”, Nonlinear Dynamics, Chaos and Their Applications to Engineering Sciences, Vol. 6: Applications of Nonlinear Phenomena, 2000, J.M Balthazar, P.B. Gonoçalves, R.M.F.L.R.F. Brasil, I.L. Caldas, F.B. Rizzatto, Editors, ISBN: 85-900351-6-6, pp.212-221.

Sanner, R. M., Slotine, J. J. E., 1992, “Gaussian networks for direct adaptive control”, *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, November, pp. 837-863.

Singhal, S. Wu, L., 1989, “Training multilayer perceptrons with the extended Kalman algorithm”, *Advances in neural Information Processing Systems*, Vol. 1, Morgan Kaufman Pub. Inc., pp. 133-140.

Soloway, D., Haley, P. J., 1997, “Neural generalized predictive control: a Newton-Raphson implementation”, *NASA Technical Memorandum* 110244, February.

Su, H., McAvoy, T. J., 1993, “Neural model predictive control of nonlinear chemical processes”, *Proceedings of the 1993 International Symposium on Intelligent Control*, Chicago, Illinois, USA, August, pp. 358-360.

Zhu, K. Y. Qin, X. F., Chai, T. Y., 1999, “A new decoupling design of self-tuning multivariable generalized predictive control”, *International Journal of Adaptive Control and Signal Processing* 13, pp. 183-196.

Watanabe, K. Fukuda, T. Tzafestas, S. G., 1991, “Learning algorithms of layered neural networks via extended Kalman filters”, *International Journal of System Science*, 22(4), pp. 753-768.