

Research Article

Application of Computational Intelligence in Order to Develop Hybrid Orbit Propagation Methods

Iván Pérez, Juan Félix San-Juan, Montserrat San-Martín, and Luis María López-Ochoa

University of La Rioja, La Rioja, 26004 Logroño, Spain

Correspondence should be addressed to Juan Félix San-Juan; juanfelix.sanjuan@unirioja.es

Received 4 August 2013; Revised 5 October 2013; Accepted 6 October 2013

Academic Editor: Ren-Jieh Kuo

Copyright © 2013 Iván Pérez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new approach in astrodynamics and celestial mechanics fields, called *hybrid perturbation theory*. A hybrid perturbation theory combines an *integrating technique*, general perturbation theory or special perturbation theory or semianalytical method, with a *forecasting technique*, statistical time series model or computational intelligence method. This combination permits an increase in the accuracy of the integrating technique, through the modeling of higher-order terms and other external forces not considered in the integrating technique. In this paper, neural networks have been used as time series forecasters in order to help two economic general perturbation theories describe the motion of an orbiter only perturbed by the Earth's oblateness.

1. Introduction

The goal of an orbit propagator is to determine accurately the position and velocity of an object orbiting the Earth, mainly an artificial satellite or a space debris object. In order to achieve this, the position and velocity at an initial instant t_0 must be known, as well as the set of forces acting on the orbiter. The main force is the gravitational attraction from an ideally spherical Earth, but some perturbations exist, such as the nonsphericity of the Earth, atmospheric drag, the effect of other celestial bodies, and solar radiation pressure, among others. All these perturbations are not always taken into account but only the most determinant ones for the specific orbiter, depending on its type of orbit and the scientific requirements for the mission.

In order to solve the problem, the nonlinear equations of motion of this complex dynamical system must be integrated. Depending on both the integrating techniques and applied analytical transformations, three different methods have been described: general perturbation theory, special perturbation theory, and semianalytical techniques.

General perturbation methods, also analytical theories, refer to the analytical integration of the equations of motion, using perturbation theories based on series expansions [1–8]. General perturbation methods provide analytical solutions valid for any set of initial conditions. These solutions are

explicit functions of time, constants of the problem, and integration constants, which are mainly characterized by retaining the essential behavior of motion. Provided that the position and velocity of the orbiter at an initial instant t_0 are known, their determination for any other instant t_f requires only one evaluation of the analytical solution.

Nevertheless, it must be noticed that most analytical theories currently in use do not lead to accurate solutions but to approximations, mainly due to two reasons. The first is that only very basic models of external forces are considered, because in some cases their corresponding analytical expressions are very complicated. The second reason for such inaccuracy is that only low-order approximations are usually developed, since the analytical expansions for the higher-order solutions may become unmanageably long.

On the other hand, special perturbation methods [9] refer to the accurate numerical integration of the equations of motion, including any external forces, even those in which analytical manipulations are complicated. Nonetheless, in order to achieve the intended accuracy, it is necessary to use small integrating steps, which means that all the intermediate positions of the orbiter must be computed in a dense grid between t_0 and t_f .

Finally, semianalytical techniques [10, 11] are characterized by combining special and general perturbation methods. This approach allows for including any external forces in

the equations of motion, which are simplified using analytical techniques, whereas the transformed equations of motion are integrated numerically. The determination of the position of the orbiter in t_f also requires the integration of a grid starting at t_0 , but in this case the grid can be significantly less dense than in special perturbation methods. Consequently, depending on both the integrating step and the difference between t_0 and t_f , the efficiency of this method can be very close to that of the general perturbation theory.

We present a fourth approach, called *hybrid perturbation theory*, which can combine one of the aforementioned methods with time series forecasting techniques, based on either statistics [12–14] or computational intelligence [15]. In this paper, we focus on the combination of economic general perturbation theories and neural networks [16–20], which is one of the most widespread techniques for time series forecasting in computational intelligence.

The goal of the forecasting part of the method is to model the dynamics not present in the approximate analytical expressions, due to the fact that not all the forces acting on the orbiter may have been considered and also because the higher-order terms may have been excluded to avoid extremely long expressions. To allow for the subsequent increase in precision, the forecaster needs additional data from which to model absent dynamics. Therefore, besides knowing the position and velocity of the orbiter at an initial instant t_0 , the same magnitudes must also be accurately known for a certain period of time.

To express this concept in a formal manner, the aim is to determine an accurate estimation \hat{x}_{t_f} of the real position and velocity x_{t_f} of an orbiter, expressed in any set of variables, at an instant t_f . That estimation will be calculated with the addition of these two components:

$$\hat{x}_{t_f} = x_{t_f}^A + \hat{\varepsilon}_{t_f}, \quad (1)$$

where $x_{t_f}^A$ represents the approximate position and velocity determined by means of the analytical expressions, whilst $\hat{\varepsilon}_{t_f}$ does so for the neural network forecast of the error of those expressions at epoch t_f .

The first component, $x_{t_f}^A$, can easily be obtained from the analytical propagator expressions, which are explicit functions \mathcal{P} of both time t and a known value x_{t_0} at an initial instant t_0 :

$$x_{t_f}^A = \mathcal{P}(t_f, x_{t_0}). \quad (2)$$

The second component, $\hat{\varepsilon}_{t_f}$, is the output of the neural network for the instant t_f , after having been trained to forecast the error of the analytical expressions. We define the error ε_{t_i} , at any instant t_i , as the difference between the real value of position and velocity, x_{t_i} , and the value obtained from the approximate analytical expressions, $x_{t_i}^A$,

$$\varepsilon_{t_i} = x_{t_i} - x_{t_i}^A, \quad (3)$$

which represents the dynamics missing from the analytical expressions, due to the simplifications assumed both in the

considered set of forces and in the number of terms of the expressions.

The process in order to provide the neural network with the capability to forecast errors is the training, which requires knowledge of the real errors ε_{t_i} between t_0 and another instant t_n with a regular cadence. According to (3), that implies knowing the real positions and velocities x_{t_i} , as well as approximate positions and velocities $x_{t_i}^A$, for $i = 0, \dots, n$. The former can be obtained by means of either any observational method or through the precise propagation of the equations of motion. Meanwhile, the latter can be easily determined evaluating the analytical expressions for $t = t_0, \dots, t_n$.

Once the neural network has been trained, it is capable of forecasting future errors $\hat{\varepsilon}_{t_i}$ from previous ones, thus complementing the approximate analytical expressions so as to reach more accurate results, as stated in (1).

In this paper, we illustrate and analyze the utility of the hybrid perturbation theory, considering simple models with only conservative forces, in order to draw valuable conclusions regarding its viability.

This paper is organized as follows. The considered analytical theories are summarized in Section 2. Section 3 briefly introduces neural network techniques. Section 4 illustrates the creation of a hybrid orbit propagator in the case of a LEO orbiter. Finally, Section 5 analyzes the results of the developed hybrid orbit propagator.

2. General Perturbation Methods

An analytical orbit propagator program (AOPP) is an application that collects and arranges all mathematical expressions involved in an approximate analytical solution of the satellite equations of motion. The two AOPPs here considered have been derived from the Kepler problem, which can be solved analytically (see [21] for more details) and from a first-order closed-form analytical theory of the *main problem* of the artificial satellite theory.

The *main problem* is defined as a Kepler problem perturbed by the Earth's oblateness. The Hamiltonian of this dynamical system can be written in a Cartesian coordinate system (x, X) as

$$\mathcal{H} = \frac{1}{2} (X \cdot X) - \frac{\mu}{r} \left[1 - J_2 \left(\frac{\alpha}{r} \right)^2 P_2 \left(\frac{z}{r} \right) \right], \quad (4)$$

where $r = \|x\| = \sqrt{x^2 + y^2 + z^2}$, μ is the gravitational constant, α is the equatorial radius of the Earth, J_2 is the oblateness coefficient, and P_2 is the Legendre polynomial of degree 2.

The first step in carrying out the analytical theory consists in expressing the Hamiltonian (4) in terms of Delaunay variables (l, g, h, L, G, H) . This set of canonical action-angle variables can be defined in terms of orbital elements such as $l = M$, $g = \omega$, $h = \Omega$, $L = \sqrt{\mu a}$, $G = \sqrt{\mu a(1 - e^2)}$, and $H = \sqrt{\mu a(1 - e^2)} \cos i$, where M , ω , Ω , a , e , and i are the mean anomaly, argument of the perigee, longitude of the ascending node, semimajor axis, eccentricity, and inclination,

respectively. Then, the transformed Hamiltonian is expressed in Delaunay variables as

$$\mathcal{H} = -\frac{\mu^2}{2L^2} - \frac{\epsilon}{2} \frac{\mu}{r} \left(\frac{\alpha}{r} \right)^2 (1 - 3s^2 \sin^2(f + g)), \quad (5)$$

where $\epsilon = J_2$ is a small parameter, $s = \sin i$, and f is the true anomaly.

Next, the Hamiltonian (5) is normalized [22] by applying the Lie transform $\varphi: (l, g, h, L, G, H) \rightarrow (l', g', h', L', G', H')$, which up to first order reads

$$\mathcal{H}_0 = \mathcal{H}_0, \quad (6)$$

$$\mathcal{H}_1 = \mathcal{H}_1 - \frac{\mu^2}{L'^3} - \frac{\partial \mathcal{W}_1}{\partial l'}. \quad (7)$$

Equation (7) is solved by taking \mathcal{H}_1 as the average over the mean anomaly:

$$\mathcal{H}_1 = \frac{3\alpha^2 \mu^4 s'^2}{4L'^6 \eta'^3} - \frac{\alpha^2 \mu^4}{2L'^6 \eta'^3}, \quad (8)$$

and then \mathcal{W}_1 is computed as

$$\begin{aligned} \mathcal{W}_1 &= \frac{L'^3}{\mu^2} \int (\mathcal{H}_1 - \mathcal{H}_1) dl \\ &= \frac{\mu^2 \alpha^2 (3s'^2 - 2) \phi'}{4L'^3 \eta'^3} + \frac{\mu^2 \alpha^2 e' (3s'^2 - 2)}{4L'^3 \eta'^3} \sin f' \\ &\quad - \frac{3\mu^2 \alpha^2 e' s'^2}{8L'^3 \eta'^3} \sin(f' + 2g') \\ &\quad - \frac{3\mu^2 \alpha^2 s'^2}{8L'^3 \eta'^3} \sin(2f' + 2g') \\ &\quad - \frac{\mu^2 \alpha^2 e' s'^2}{8L'^3 \eta'^3} \sin(3f' + 2g'), \end{aligned} \quad (9)$$

where $\eta' = \sqrt{1 - e'^2}$ and $\phi' = f' - l'$.

Hence, up to the first order the transformed Hamiltonian is given by

$$\mathcal{H} = -\frac{\mu^2}{2L'^2} + \epsilon \left(\frac{3\alpha^2 \mu^4 s'^2}{4L'^6 \eta'^3} - \frac{\alpha^2 \mu^4}{2L'^6 \eta'^3} \right). \quad (10)$$

We must point out that the Hamiltonian \mathcal{H} is integrable because it only depends on the momenta L' , G' , and H' , and thus the equations of motion are obtained as

$$\begin{aligned} \frac{dl'}{dt} &= \frac{\partial \mathcal{H}}{\partial L'} = \frac{\mu^2}{L'^3} + \epsilon \left(\frac{3\alpha^2 \mu^4}{2L'^7 \eta'^3} - \frac{9\alpha^2 \mu^4 s'^2}{4L'^7 \eta'^3} \right), \\ \frac{dg'}{dt} &= \frac{\partial \mathcal{H}}{\partial G'} = \epsilon \left(\frac{3\alpha^2 \mu^4}{L'^7 \eta'^4} - \frac{15\alpha^2 \mu^4 s'^2}{4L'^7 \eta'^4} \right), \\ \frac{dh'}{dt} &= \frac{\partial \mathcal{H}}{\partial H'} = -\epsilon \frac{3\alpha^2 \mu^4 c'}{2L'^7 \eta'^4}, \\ \frac{dL'}{dt} &= \frac{dG'}{dt} = \frac{dH'}{dt} = 0, \end{aligned} \quad (11)$$

where $c' = \sqrt{1 - s'^2}$.

By integrating (11), we directly obtain the values of the momenta L' , G' , and H' as constants, whereas the variables l' , g' , and h' yield

$$\begin{aligned} l' &= \left[\frac{\mu^2}{L'^3} + \epsilon \left(\frac{3\alpha^2 \mu^4}{2L'^7 \eta'^3} - \frac{9\alpha^2 \mu^4 s'^2}{4L'^7 \eta'^3} \right) \right] (t - t_0) + l'_0, \\ g' &= \left[\epsilon \left(\frac{3\alpha^2 \mu^4}{L'^7 \eta'^4} - \frac{15\alpha^2 \mu^4 s'^2}{4L'^7 \eta'^4} \right) \right] (t - t_0) + g'_0, \\ h' &= \left[-\epsilon \frac{3\alpha^2 \mu^4 c'}{2L'^7 \eta'^4} \right] (t - t_0) + h'_0, \end{aligned} \quad (12)$$

where l'_0 , g'_0 , h'_0 , L'_0 , G'_0 , and H'_0 are the transformed initial conditions l_0 , g_0 , h_0 , L_0 , G_0 , and H_0 at the epoch t_0 .

Finally, from (9) we can calculate the first-order explicit equations of the direct and inverse transformations [23].

An AOPP is derived from the above analytical theory, which has to evaluate 93 terms. This AOPP is called Z2DN1. The algebraic manipulations required to carry out this analytical theory and its corresponding AOPP are built using a set of *Mathematica* packages called MathATESAT [24].

Z2DN1 begins initializing the physical parameters and initial conditions at epoch t_0 . Next, it transforms the initial conditions into Delaunay variables (l_0 , g_0 , h_0 , L_0 , G_0 , and H_0) and transports them across the inverse transformation of the Delaunay normalization (l'_0 , g'_0 , h'_0 , L'_0 , G'_0 , and H'_0). Then, the program provides Delaunay variables at epoch t_f from integrated Hamilton equations (l' , g' , h' , L' , G' , and H'). Finally, the direct transformation of the Delaunay normalization is applied, and then the osculating Keplerian elements (a , e , g , h , i , and l) and the state vector (x , y , z , \dot{x} , \dot{y} , and \dot{z}) can be calculated.

Z2DN1 AOPP has been compared with the numerical integration of the original problem (4) using a high-order Runge-Kutta method [25] for the case of a LEO orbiter ($a = 7221$ km, $e = 0.0622$, and $i = 48.98^\circ$).

Figure 1 shows the distance error in a time span interval of 3 days, which is about 42 orbiter periods. As can be observed, the distance error of the first-order J_2 analytical theory when compared with a more complex perturbation model is at a level of 1.2 km. Figure 2 shows the orbital element errors.

3. Neural Networks (NN) Overview

Artificial neural networks constitute one of the most widespread techniques in the field of computational intelligence. They are based on parallel processing of information by simple units, called neurons, as happens in the brain. Knowledge resides in the strength of the connections between neurons, which are called synapses. Such strength can be reinforced or weakened through a learning process based on known data, which allows the neural network to acquire and emulate underlying dynamics in the data. Some of the fields in which artificial neural networks have been successfully used are pattern recognition, data classification, data mining, and time series forecasting.

Figure 3 shows an example of neural network architecture with three layers: the input layer, which comprises the neural

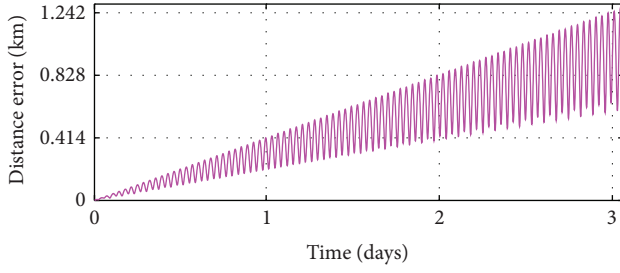


FIGURE 1: Distance error of the first-order J_2 analytical theory for a LEO orbiter.

network inputs, the hidden layer, with a certain number of hidden neurons, and the output layer, which includes one neuron per neural network output. The number of inputs and outputs is determined by the nature of the problem to be solved, whereas, on the other hand, the number of hidden layers and hidden neurons may vary. The more hidden layers are and the more hidden neurons within them are, the higher processing power the neural network will reach.

The typical connection pattern implies that each input or each neuron is connected to all the neurons in the following layer through a set of weights w , which determine the strength of the synapses. Additionally, neurons have a special input with a constant value of 1, weighted by a bias b , which represents the neuron threshold. A neuron collects the weighted sum of all its inputs and applies it to an activation function, which can be linear or nonlinear, depending on the problem to be solved. One of the most frequent nonlinear activation functions is the hyperbolic tangent.

For example, the value of the output y for the neural network in Figure 3 is given as

$$\begin{aligned}
 y = & g(f(x_1 w_{1,1} + x_2 w_{2,1} + x_3 w_{3,1} + x_4 w_{4,1} + b_1) w_{1,\text{out}} \\
 & + f(x_1 w_{1,2} + x_2 w_{2,2} + x_3 w_{3,2} + x_4 w_{4,2} + b_2) w_{2,\text{out}} \\
 & + f(x_1 w_{1,3} + x_2 w_{2,3} + x_3 w_{3,3} + x_4 w_{4,3} + b_3) w_{3,\text{out}} \\
 & + f(x_1 w_{1,4} + x_2 w_{2,4} + x_3 w_{3,4} + x_4 w_{4,4} + b_4) w_{4,\text{out}} \\
 & + b_{\text{out}}).
 \end{aligned} \tag{13}$$

As can be seen, all the involved operations are products and additions, together with the selected activation functions, which can be either calculated or determined by means of a look-up table, depending on the available hardware.

An initial training process is necessary to find the set of weights w and biases b that allows the neural network to behave as desired. In time series forecasting, supervised training methods are used, which need a training data set of inputs and target outputs. The difference between the calculated outputs and the targets regulates the adjustment of weight and bias values in an iterative process, until such difference becomes small enough.

4. Hybrid Analytical-NN AOPP

4.1. Data Preprocessing. Delaunay variables (l , g , h , L , G , and H) have been chosen to characterize the orbit. Two sets of values have been used in this process. The first consists of accurate simulated values, obtained through the numerical integration of the original problem (4) using a high-order Runge-Kutta method, which are considered as actual values from precise observations. The second is obtained by applying the analytical methods described in Section 2; these values are approximate because they assume the simplifications used in Section 2. In both cases, values have been generated with a cadence of 1 minute.

Then, ε_{t_i} errors have been calculated for each variable, subtracting both data sets, as stated by (3). After this operation, the error sequences of the angular variables, that is, l , g , and h , may include some outliers that differ from the rest of the values in an approximate quantity of 2π . Such differences correspond to complete spins and, although they have no effect on trigonometrical calculations, for a neural network they represent abrupt discontinuities in values that are actually very close. Therefore, complete spin differences have to be removed by adding or subtracting 2π from outliers, as can be seen in Figure 4.

The resulting errors for variable H are always 0 for the problem considered here, which means that the analytical theory is able to determine H values accurately. Therefore, forecasting of H error is not necessary in this case. For each of the remaining variables, a different neural network has been used to forecast future error values, which allows for using parallel training processes.

Data have to be arranged in a suitable format for the neural network. Forecasting will be done through a sliding window; a set of consecutive error values constitutes the input vector from which the neural network has to calculate the error value for the next instant. In the following step, the sliding window moves, and the new input vector loses the initial value, whilst simultaneously appending the recently forecasted value. This process extends until the final error value to be forecasted is reached.

Therefore, during this step, each sequence of error values, ε_{t_i} , has to be converted into a set of vectors that will be used during the training phase of the corresponding neural network. A sliding window length of 2 orbiter periods has been taken for this first study, which means that error values corresponding to the 2 previous periods will be available for the neural network in order to forecast the next error value. Since the orbiter period is 102 minutes, and a cadence of 1 minute has been considered, each neural network must have $2 \cdot 102 = 204$ inputs and 1 output. Consequently, the training data sets to be prepared must be sets of vectors specifying the following error value for each 204-element set of consecutive error values in each variable.

The number of vectors to be prepared for each neural network training process depends on the period of time desired for the neural network to learn the underlying dynamics. We have chosen for this purpose the duration of the number of complete orbiter periods which is closest to 1 day, that is, 14 periods of 102 minutes: 1428 minutes.

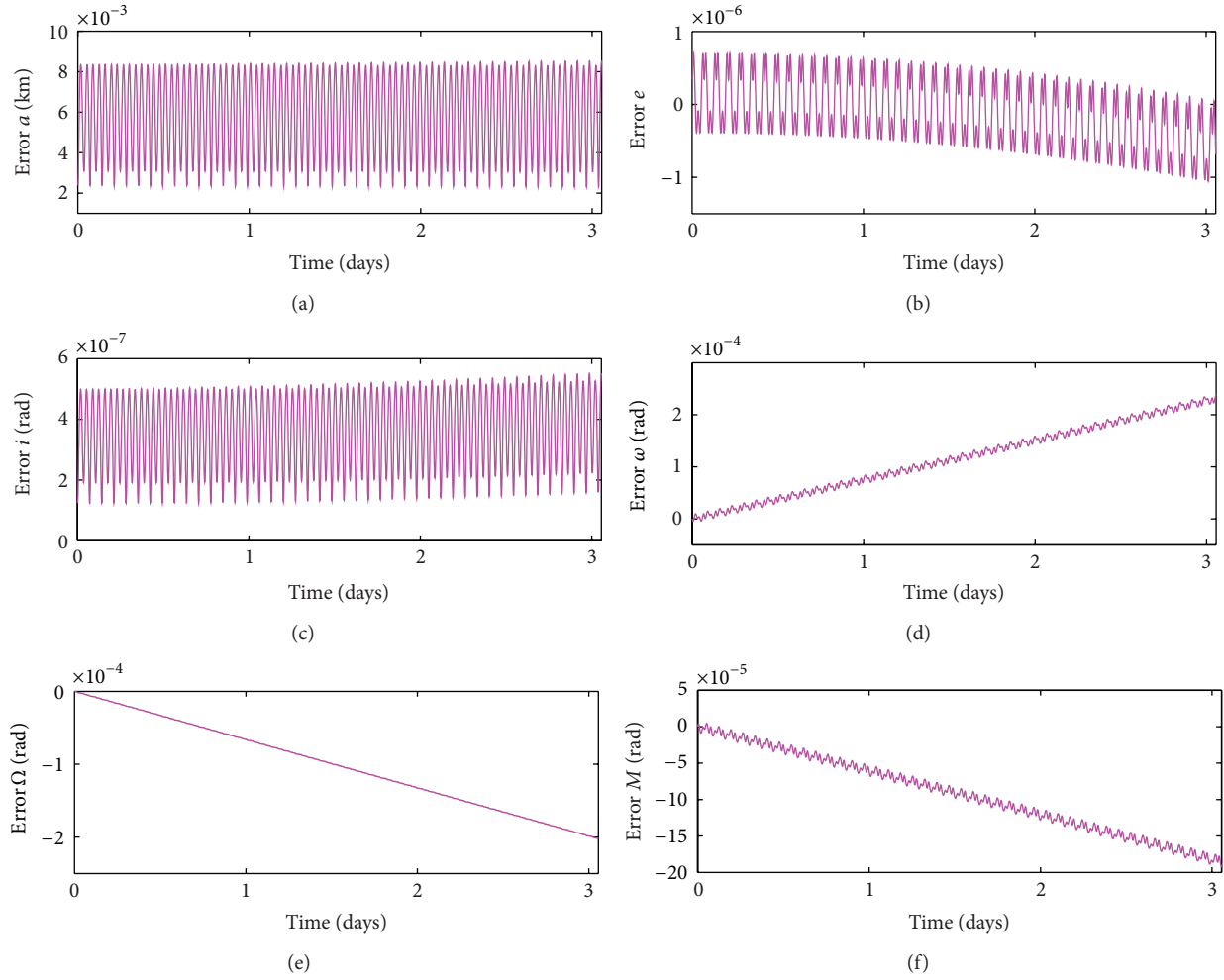


FIGURE 2: Orbital element errors (a , e , i , ω , Ω , and M) of the first-order J_2 analytical theory.

Therefore, error values from minute 1 to 1428 are used to train the neural network, and so forecasted values start at minute 1429. Distribution of all these data in 204-element vectors produces 1225 vectors for the training process of the neural network in each variable.

Finally, data to be used with neural networks must be normalized, so that values fall within the neural network operating range. A linear normalization to the interval $[-0.9, +0.9]$ has been carried out, maintaining a security margin up to the $[-1, +1]$ output range of the hyperbolic tangent activation function. Consequently, the inverse transformation must be applied to forecasted values so that they can be restored to their original ranges.

4.2. Design Parameters. The first step in designing the neural network is the selection of a network type suitable for the intended goal. Despite its simplicity, the multilayer perceptron constitutes a good time series forecaster [15]. We have considered a multilayer perceptron similar to Figure 3 but with a different number of inputs and hidden neurons. As mentioned in Section 4.1, the number of inputs to the neural network is 204, whereas the number of neurons in

the hidden layer, which determines the processing capability, constitutes a parameter for which two different values have been considered, namely, 30 and 60.

Neural networks can be provided with more than one hidden layer, but this is not recommended unless the problem to be solved is extremely complex, due to the increase in training computational cost. One hidden layer is enough for the neural network to approximate any function, provided that it has enough hidden neurons with a nonlinear activation function type [26–28]. The selected architecture for the work presented in this paper only includes one hidden layer, as in Figure 3.

Another design parameter is the type of activation function for each layer. According to [15], although there are other alternatives, the use of the hyperbolic tangent for the hidden layer, together with the linear activation function for the output layer, is a habitual configuration in multilayer perceptrons used for time series forecasting; thus this is our selected architecture.

The most common training algorithm for feedforward networks, such as the multilayer perceptron, was backpropagation [16]. It consisted in changing the weights in the direction of the negative gradient of a predetermined performance

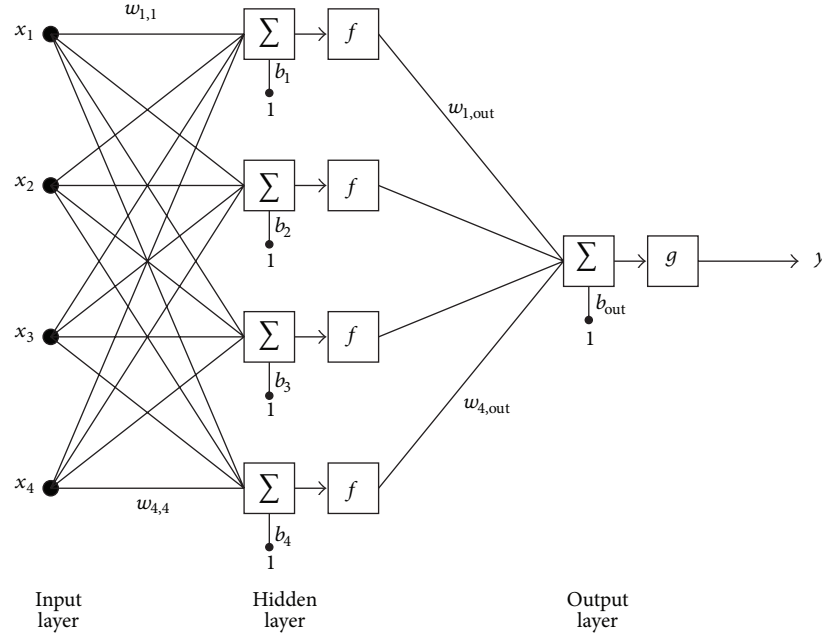


FIGURE 3: Neural network with four inputs, one hidden layer containing four neurons, and one output.

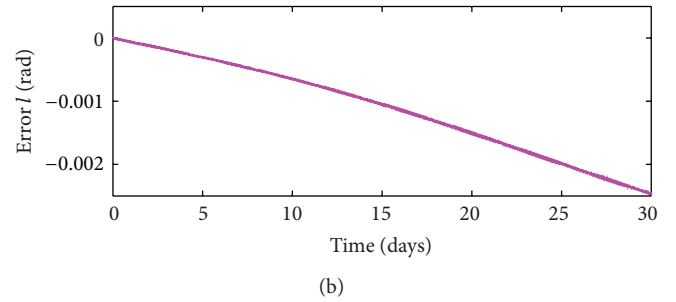
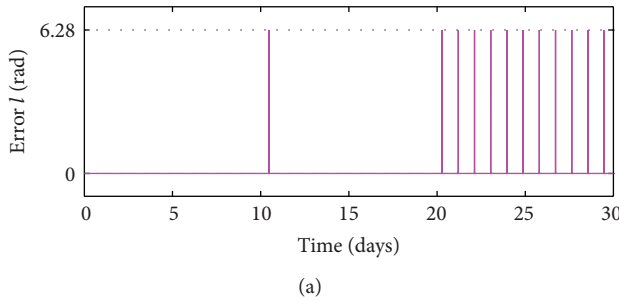


FIGURE 4: Error ε_{t_i} in variable l in the case of Z2DN1 AOPP (a) before subtracting complete spins and (b) after subtracting complete spins.

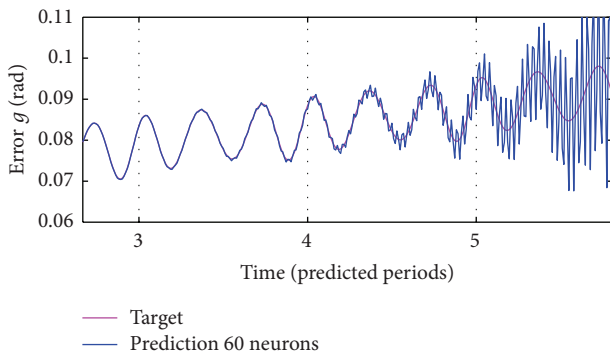


FIGURE 5: Prediction of error $\hat{\varepsilon}_{t_i}$ in variable g (Unperturbed Analytical-NN, 60 neurons in hidden layer).

function, usually the mean square error between the network outputs and target outputs. A more sophisticated learning method was developed [29] through the application of the Levenberg-Marquardt algorithm to backpropagation, which

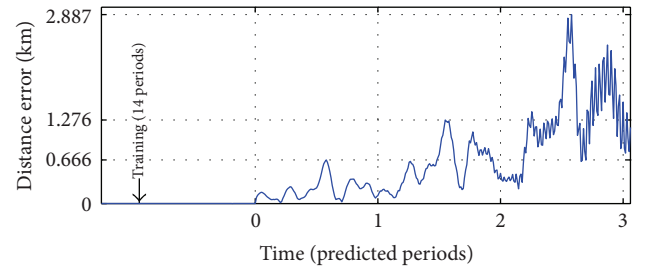
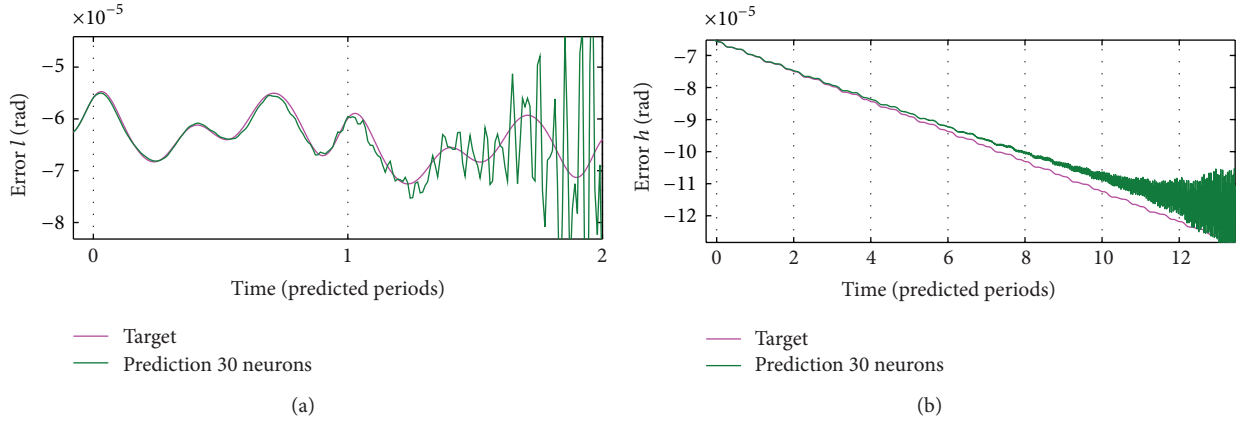
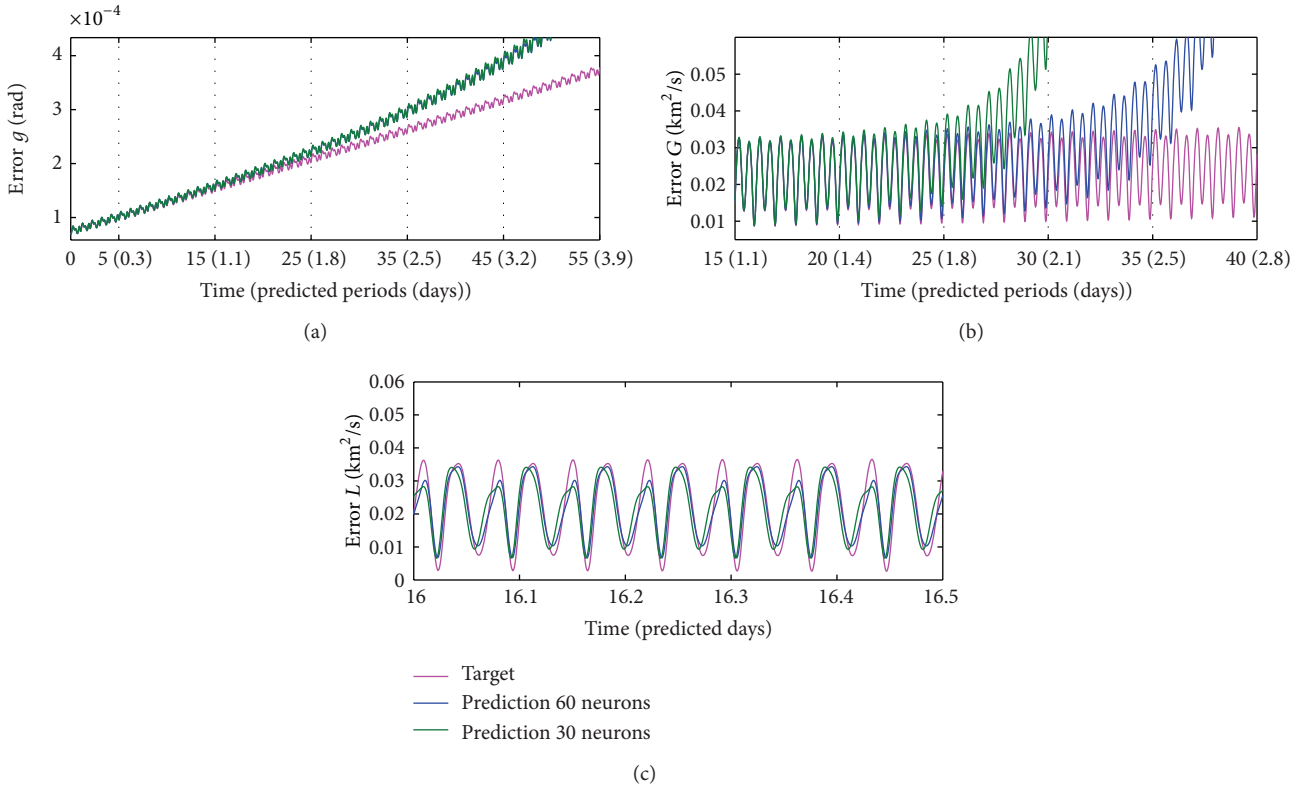


FIGURE 6: Distance error (Unperturbed Analytical-NN, 60 neurons in hidden layer, modeled variables: l , g , h , L , and G).

allowed for a remarkable acceleration of between 10 and 100 times faster in the training process.

When designing a neural network for a particular task, the possibility that overfitting might occur has to be considered. It happens when the neural network architecture is more complex than required for the problem to be solved, and this leads to a loss in the generalization capability, which causes

FIGURE 7: Prediction of error $\hat{\varepsilon}_{t_i}$ in variables l and h (Z2DNI-NN, 30 neurons in hidden layer).FIGURE 8: Prediction of error $\hat{\varepsilon}_{t_i}$ in variables g , G , and L (Z2DNI-NN, 30 versus 60 neurons in hidden layer).

the neural network output to adjust overexcessively to data being used in the training, as well as being unable to forecast new values properly.

Overfitting risk can be minimized through the regularization method, which implies the use of the regularized mean square error performance function, comprising a weighted sum of both the habitual mean square error plus a new mean square weight value. As a consequence of the introduction of this new parameter, lower weight and bias values are preferred during the training process, leading to a smoother neural network output, which reduces the possibility that overfitting

may occur. In order to automatically adjust the necessary parameters, the Bayesian regularization method [30, 31] can be applied.

The training method we have used is a Matlab implementation of the Bayesian regularization backpropagation, which includes the Levenberg-Marquardt algorithm. This process minimizes a combination of squared errors and weights in the right proportion to achieve a neural network with generalization capability.

Finally, a selection of the initial weights and biases in order to start the training process is necessary, which has been

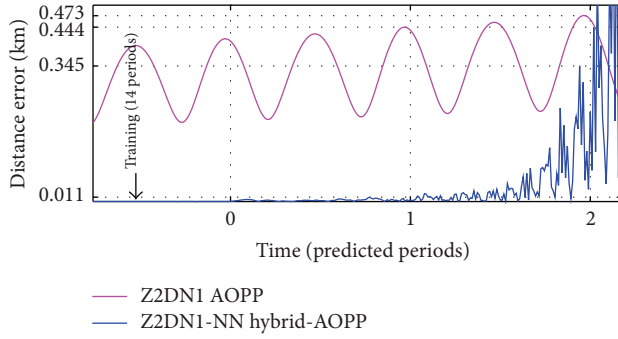


FIGURE 9: Distance error (Z2DN1-NN, 30 neurons in hidden layer, modeled variables: l , g , h , L , and G).

carried out according to the Nguyen-Widrow algorithm [32]. This algorithm distributes neuron active regions uniformly in each layer.

5. Results

After the training phase, the neural network is able to forecast future errors, $\hat{\varepsilon}_t$, in Delaunay variables l , g , h , L , and G . These errors can be finally added to the variable values, obtained from the analytical theory, and the result constitutes the hybrid method output. Two different hybrid methods based on two different analytical theories have been implemented and compared with the accurate integration of (4). The former, which we have called *Unperturbed Analytical-NN*, has been carried out from the Kepler analytical solution, whereas the latter, called *Z2DN1-NN*, is from a first-order closed-form analytical theory of the main problem of the artificial satellite theory. The neural network architecture of both AOPPs is the same, namely, the multilayer perceptron. We must note that in the case of the *Z2DN1-NN* model, two different neural networks have also been considered, one with 30 neurons in its hidden layer and the other with 60. In order to compare results, the possibility of restricting neural network forecasting to a representative subset of Delaunay variables has been explored.

5.1. Unperturbed Analytical-NN Hybrid-AOPP. The analytical part of this hybrid-AOPP only takes into account the Kepler problem, that is, to consider the Earth's oblateness, J_2 , equal to zero in (4). We must note that the Earth's oblateness has a strong influence on satellite orbits and this hybrid-AOPP allows us to analyze the capability of the neural network in modeling complex dynamics, which are not present in the analytical calculations. To accomplish this task, the neural network hidden layer has been provided with 60 neurons, which, taking into account the number of inputs, constitutes a neural network that might be sufficiently powerful.

The most significant Delaunay variables for determining the studied orbiter position are l , g , and h , as can be concluded from the analysis of the influence of different subsets of variables over the distance error of the predicted orbiter

position. However, forecasts of their errors $\hat{\varepsilon}_t$ remain accurate during a few periods, as can be seen in Figure 5 for variable g . A more complex neural network and a longer training data set would be necessary to extend the forecasting horizon.

The other two modeled variables, L and G , have been forecasted more precisely, but they do not have such a great influence on the orbiter dynamics as the other three. Therefore, the orbiter position can only be determined with relative precision during two periods. Figure 6 shows the maximum distance errors after periods 1, 2, and 3: 0.666 km, 1.276 km, and 2.887 km, respectively.

5.2. Z2DN1-NN Hybrid-AOPP. The second approach is based on the Z2DN1 AOPP described in Section 2, which includes part of the first-order effects of the Earth's oblateness. Therefore, the neural network only has to model the higher-order terms. As the dynamics to be modeled are simpler than in the previous case, the necessary neural network should also be less complex. First, we have analyzed the effect of having 30 neurons in the hidden layer, and then we have compared the results with those obtained from a network with 60 hidden neurons.

Analyzing with 30 neurons, error prediction of some of the variables quickly deteriorates, as shown in Figure 7 for l and h , while others maintain accurate forecasts for longer, as can be seen in Figure 8; the green plots for variables g and G remain close to their target values up to approximately 25 predicted orbiter periods, that is, 1.8 days. Variable L forecast performs well even after 16 days, as can also be seen in Figure 8.

The distance error compared with the real position can be plotted when all the forecasted errors are added to their respective analytical values, and the resulting hybrid output is converted into the coordinates of the orbiter position. Figure 9 shows acceptable behavior during 1 or 2 periods, with maximum distance errors of 11 m and 345 m, respectively, both below the maximum errors of the pure analytical method for such instants, which are 444 m and 473 m, respectively.

Since this result is clearly improvable, a more complex neural network with 60 hidden neurons has been applied to the same forecasting task. Then, some of the variable errors extend their prediction horizon significantly, as can be seen in Figure 10 for l and h , in comparison with Figure 7 with 30 hidden neurons. Figure 8 shows an improvement of approximately 5 orbiter periods in variable G , whereas g and L remain similar to results with 30 hidden neurons.

When all these error forecasts are transformed into the orbiter position distance error, as shown in Figure 11, the hybrid method stands out against the purely analytical method, reducing the maximum distance error from 828 m to 22 m after 1 predicted day and from 1242 m to 120 m after 2 days. When only l , g , and h errors are forecasted by the neural network, maintaining the approximate analytical values for L and G , the distance error hardly increases (24 m instead of 22 m after 1 predicted day and 121 m instead of 120 m after 2 days), proving that the former are the most important variables to be forecasted for determining the position of

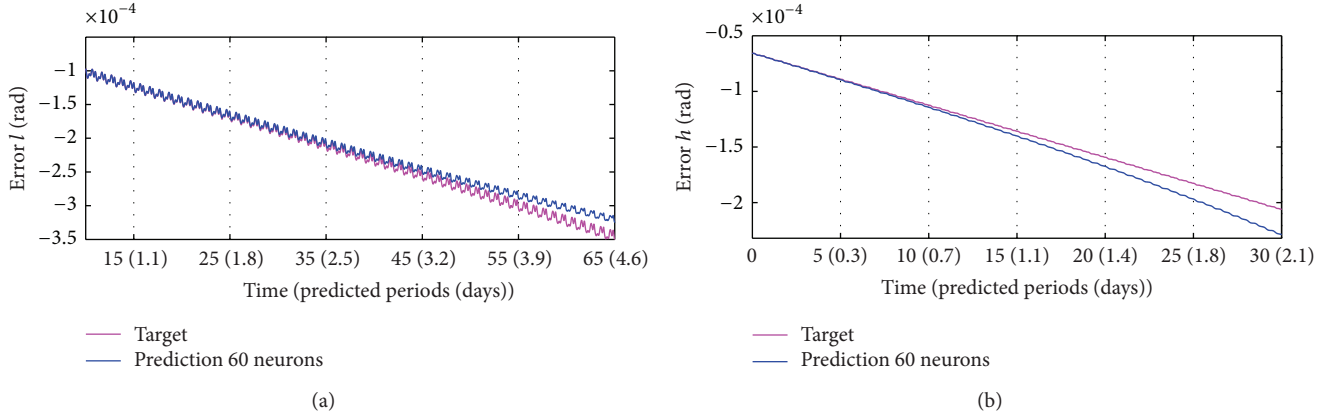


FIGURE 10: Prediction of error $\hat{\varepsilon}_{t_i}$ in variables l and h (Z2DN1-NN, 60 neurons in hidden layer).

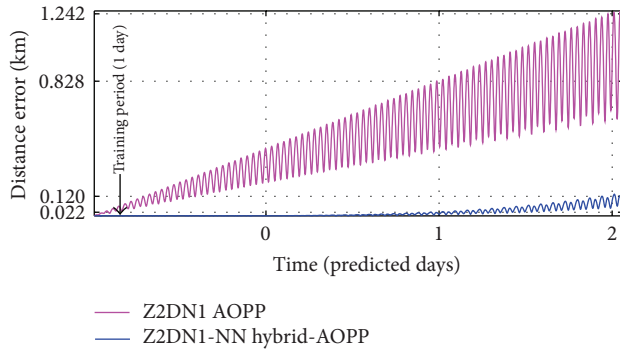


FIGURE 11: Distance error (Z2DN1-NN, 60 neurons in hidden layer, modeled variables: l , g , h , L , and G).

the orbiter under study, since the latter can be estimated analytically with sufficient accuracy.

Finally, Delaunay element errors have been transformed into the classical orbital element errors (semimajor axis a , eccentricity e , inclination i , argument of the perigee ω , longitude of the ascending node Ω , and mean anomaly M) in Figure 12, which shows a notable improvement to the hybrid Z2DN1-NN AOPP over the analytical Z2DN1 during 2 predicted days.

5.3. Analysis and Comparison of Results. The hybrid-AOPP considered in Section 5.1, namely, Unperturbed Analytical-NN, is characterized by the simplicity of the chosen analytical method, which only considers the Kepler solution with no perturbations. Despite the 60 neurons in the hidden layer, the analytical method is too simple for the neural network to model the complete error, leading to poor performance.

In Section 5.2, the analytical part of the Z2DN1-NN hybrid-AOPP includes J_2 perturbation up to the first order, which makes it closer to the precise values. Nevertheless, when the neural network is provided with only 30 neurons in its hidden layer, very little improvement is achieved during an extremely short horizon, which reveals the clearly insufficient power of the neural network for this forecasting task.

In contrast, the increase in the neural network power caused by the 60 hidden neurons used in the last part of Section 5.2 leads to a great improvement and a more satisfactory result.

As a general conclusion for all the analyzed cases herein, we can state that modeling all the Delaunay variables l , g , h , L , and G , instead of only l , g , and h , which were the most representative for this problem, does not extend the improvement horizon but does increase the improvement level during that time. H error for this problem was 0 and thus had no need to be forecasted.

Finally, as far as the validity horizon of predictions is concerned, we have reached the following conclusions. This horizon is of only 1 orbiter period for the Z2DN1-NN with 30 hidden neurons, due to the inferior power of the neural network for this prediction task. Meanwhile, the Unperturbed Analytical-NN AOPP with 60 hidden neurons extends the horizon but only up to 4 orbiter periods, since the analytical part of the method does not include perturbations; thus, the neural network has to model more complex dynamics, which exceed its prediction capacity. The validity horizon of predictions extends up to 2 days for the Z2DN1-NN with 60 hidden neurons, which shows a notable initial result that opens a new field in which to research for further improvement.

6. Conclusions and Future Work

A new type of orbit propagator program, the *hybrid-AOPP*, has been presented. This kind of AOPP consists of an integrating technique, which obtains an approximate solution, followed by a second phase in which the error is determined through either statistical time series models or computational intelligence methods. In this paper, we increase the propagation capabilities of general perturbation theories with one of the most widespread methods for time series forecasting in computational intelligence, namely, neural networks. A hybrid-AOPP, named Z2DN1-NN, has been created. It combines an economic first-order closed-form analytical orbit propagator and a neural network trained for the case of a LEO orbiter. It has been shown that the hybrid-AOPP outperforms

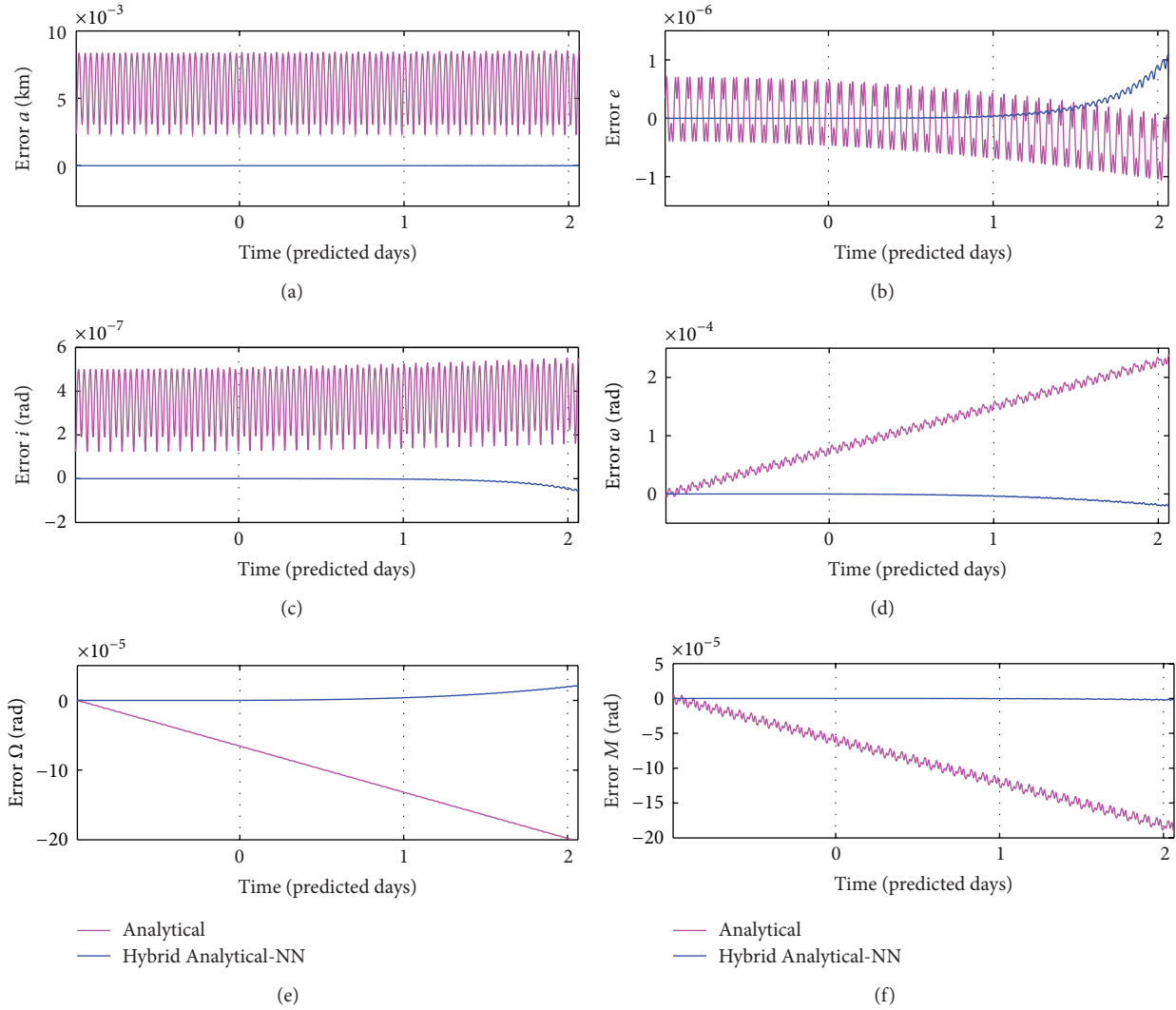


FIGURE 12: Orbital element errors (Z2DN1-NN, 60 hidden neurons, modeled variables: l , g , h , L , and G).

the analytical orbit propagator since it can drastically reduce the orbiter position error.

At present, we are carrying out research to determine the most appropriate neural network architectures and parameters for the hybrid-AOPPs. Simultaneously, statistical time series models are also being successfully applied to hybrid-AOPPs in a parallel research line.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been supported by the Government of La Rioja (Project Fomenta 2010/16) and is an extract from the dissertation that Mr. Iván Pérez will submit to the University of La Rioja in order to obtain his Doctoral degree. The authors

would like to thank an anonymous reviewer for his/her valuable suggestions.

References

- [1] D. Brouwer, "Solution of the problem of artificial satellite theory without drag," *The Astronomical Journal*, vol. 64, pp. 379–397, 1959.
- [2] Y. Kozai, "Second-order solution of artificial satellite theory without air drag," *The Astronomical Journal*, vol. 67, pp. 446–461, 1962.
- [3] R. N. Lyddane, "Small eccentricities or inclinations in Brouwer theory of the artificial satellite," *The Astronomical Journal*, vol. 68, pp. 555–558, 1963.
- [4] K. Aksnes, "A second-order artificial satellite theory based on an intermediate orbit," *The Astronomical Journal*, vol. 75, pp. 1066–1076, 1970.
- [5] H. Kinoshita, "Third-order solution of an artificial satellite theory," Special Report No. 379, Smithsonian Astrophysical Observatory, Cambridge, Mass, USA, 1977.

- [6] F. R. Hoots and R. L. Roehrich, "Models for propagation of the NORAD element sets," Spacetrack Report no. 3, U.S. Air Force Aerospace Defense Command, Colorado Springs, Colo, USA, 1980.
- [7] J. F. San-Juan, "ATESAT: automatization of theories and ephemeris in the artificial satellite problem," Tech. Rep. no. CT/TI/MS/MN/94-250, CNES, Toulouse, France, 1994.
- [8] J. F. San-Juan, "ATESAT: review and improvements. Study of a family of analytical models of the artificial satellite generated by ATESAT and their numerical validation versus PSIMU and MSLIB," Tech. Rep. no. DGA/T/TI/MS/MN/97-258, CNES, Toulouse, France, 1998.
- [9] A. C. Long, J. O. Cappellari, C. E. Velez, and A. J. Fuchs, "Godard trajectory determination system (GTDS)," Mathematical Theory Revision 1 CSC/TR-89/6001, 1989.
- [10] J. G. Neelon, P. J. Cefola, and R. J. Proulx, "Current development of the Draper semianalytical satellite theory standalone orbit propagator package," in *Proceedings of the AAS/AIAA Astrodynamics Conference*, AAS Pre-print 97-731, Sun Valley, Idaho, USA, August 1997.
- [11] P. J. Cefola, D. Phillion, and K. S. Kim, "Improving access to the semi-analytical satellite theory," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, AAS 09-341, Pittsburgh, Pa, USA, August 2009.
- [12] K. Chan, "TSA: Time Series Analysis, R package version 0.98," 2010, <http://CRAN.R-project.org/package=TSA>.
- [13] A. Trapletti and K. Hornik, "tseries: Time Series Analysis and Computational Finance, R package version 0.10-27," 2011, <http://CRAN.R-project.org/package=tseries>.
- [14] J. F. San-Juan, M. San-Martín, and I. Pérez, "An economic hybrid J_2 analytical orbit propagator program based on SARIMA models," *Mathematical Problems in Engineering*, vol. 2012, Article ID 207381, 15 pages, 2012.
- [15] A. K. Palit and D. Popovic, *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications*, Springer, London, UK, 2005.
- [16] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, PWS Publishing Company, Boston, Mass, USA, 1996.
- [17] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [18] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Computers & Operations Research*, vol. 28, no. 4, pp. 381–396, 2001.
- [19] G. P. Zhang, "An investigation of neural networks for linear time-series forecasting," *Computers & Operations Research*, vol. 28, no. 12, pp. 1183–1202, 2001.
- [20] G. P. Zhang, "Neural networks for time-series forecasting," in *Handbook of Natural Computing*, G. Rozenberg, J. N. Kok, and T. Bäck, Eds., pp. 461–477, Springer, Berlin, Germany, 2012.
- [21] J. M. Danby, *Fundamentals of Celestial Mechanics*, Willmann-Bell, Richmond, Va, USA, 1988.
- [22] A. Deprit, "Delaunay normalisations," *Celestial Mechanics and Dynamical Astronomy*, vol. 26, no. 1, pp. 9–21, 1982.
- [23] A. Aly Kamel, "Perturbation method in the theory of nonlinear oscillations," *Celestial Mechanics and Dynamical Astronomy*, vol. 3, no. 1, pp. 90–106, 1970.
- [24] J. F. San-Juan, L. M. López, and R. López, "MathATESAT: a symbolic-numeric environment in astrodynamics and celestial mechanics," *Lecture Notes in Computer Science*, vol. 6783, no. 2, pp. 436–449, 2011.
- [25] J. R. Dormand and P. J. Prince, "Practical Runge-Kutta processes," *SIAM Journal on Scientific and Statistical Computing*, vol. 10, no. 5, pp. 977–989, 1989.
- [26] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [27] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [28] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.
- [29] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [30] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 3, no. 4, pp. 415–447, 1992.
- [31] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proceedings of the International Conference on Neural Network*, pp. 1930–1935, June 1997.
- [32] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '90)*, vol. 3, pp. 21–26, June 1990.

