Improving GNSS Positioning of Satellites using Artificial Neural Networks

Sergiu-Stefan Mihai[1,*], Dan Selaru[2], Mihnea Ion[1], Alexandru Pandele[1], and Mugurel Balan[1]

[1]Romanian InSpace Engineering S.R.L., Bucharest, Romania
[2]Institute of Space Science, Magurele, Romania

Abstract. Small and nanosatellites have a very limited energy budget and their precise positioning capacity is limited in time as positioning sensors are, in general, power demanding. In this context, the article aims to present a comparison between the positioning results of a power demanding Precise Point Positioning (PPP) algorithm and the ones provided as the output of a low-power consumption Artificial Neural Network (ANN). The data that is to be processed with PPP and ANN consists of GNSS measurements. In order to validate the results, we compare the outputs with a valid set of data containing the real positions and velocities, which will be referred to as control data. The novelty of this article consists of the ANN architecture, which is designed to better exploit information coming from both an Orbit Propagator (OP) and GNSS measurements. The idea behind the OP is to estimate the Earth-orbiting satellite's position and velocity at any moment in time. Its main disadvantage comes from the finite precision of the machine that performs the computations. Thus, numerical errors accumulate in time and the estimation becomes less and less accurate. Simultaneously, the GNSS measurements alone are not sufficiently precise to allow complex orbital maneuvers such as inspection and controlled flight formation, giving only approximations of the actual satellite's position and velocity. The ANN is trained to compensate for the errors that the OP and the GNSS receiver intrinsically have. Two main approaches are to be tested, considering that the OP gives estimated positions and velocities with a sufficiently large frequency, as follows: (i) the GNSS data is queried at the same frequency with the OP; this approach is expected to give satisfactory results, because the estimation can be improved faster by the ANN, having constant available GNSS data. (ii) the GNSS data is acquired at a much slower frequency; the challenge for the ANN is now to better improve the predictions with limited GNSS readings. Theoretical aspects regarding the development of the ANN and its training phase are, too, described in the current paper, followed by the comparison with the control data which reveals the performance of PPP algorithms and ANN.

1 Introduction

This paper aims to highlight the benefits of using Artificial Neural Networks for the improvement of positioning estimation. We have divided our paper into four main parts, the first being the paper's introduction. Following, Section 2 focuses on the design of the neural network. The problem formulation is highlighted and the inputs and outputs of the network

---

e-mail: stefan.mihai@roinspace.com

are clearly defined. Section 3 presents the results we have obtained after training the network, aiming to emphasize its potential. The satellite's estimated positioning error is compared with the positioning error that is computed using the trained neural network. Additionally, information regarding the validation input data is discussed in this section. Lastly, Section 4 summarizes the presented technique, the results and future research objectives.

2 Technical approach

2.1 Problem formulation

Before designing an Artificial Neural Network that improves the positioning, we must first define the mathematical formulation following a clear objective. Figure 1 reflects the geometrical aspects of the positioning problem and hints towards a few possible solutions.

We begin with the assumption that the satellite has a known position defined by a vector $\vec{r}_i$ in the Earth-Centered Eath-Fixed reference frame. Additionally, we consider this position to be accurate and obtained from a GNSS receiver. Using an Orbital Propagator we can estimate the position and velocity of an object revolving around the Earth, after an arbitrary period of time Δt. The estimated position, symbolized with the positioning vector $\vec{r}_c$, is, undoubtedly, different from the precise position of the same object after the same Δt seconds. Considering the precise position defined by the vector $\vec{r}_p$, we define an error vector that gives the positioning error between the Orbital Propagator's estimation and the precise position as follows:

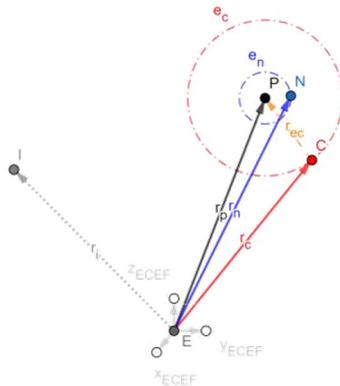$$\vec{r}_{ec} = \vec{r}_p - \vec{r}_c \tag{1}$$

Figure 1. Problem formulation: geometric aspects

The magnitude of $\vec{r}_{ec}$ quantifies the radius e_c of a sphere centered in the precise position, as follows:

$$e_c = \left\| \vec{r}_p - \vec{r}_c \right\|_2 \tag{2}$$

It now becomes clear that the Artificial Neural Network should be designed to estimate a new position which is closer to the precise one. Using the same operations with the network's estimated position vector $\vec{r}_n$ we define the radius e_n of a sphere centered in the

precise position that gives the error between the new estimation and the precise location, as follows:

$$e_n = \left\| \vec{r}_p - \vec{r}_n \right\|_2 \tag{3}$$

Alternatively, this error can be quantified using the vector addition rule:

$$e_n = \left\| \vec{r}_p - (\vec{r}_c + \vec{r}_{ec}) \right\|_2 \tag{4}$$

where $\vec{r}_{ec}$ is the estimated propagation error which is computed with the network.

The objective is to design an Artificial Neural Network that, for an estimated position $\vec{r}_c$ after Δt seconds, determines a new position $\vec{r}_n$ that satisfies the inequality $e_n \le e_c$ so that it is closer to the precise position.

2.2 The design of the orbit propagator

The construction of the orbit propagator begins by choosing a reference system in which we define a position vector $\vec{r}$. A suitable candidate is the Earth-Centered Inertial coordinate system; the final estimated position, however, must be converted to the ECEF reference system. Generally, the acceleration induced by the force of gravity, for a body with mass M, or a gravitational mass , at a position $\vec{r}$, is defined by the formula:

$$\vec{g} = -G \frac{M}{r^3} \vec{r} = -\frac{\mu}{r^3} \vec{r} \tag{5}$$

In order to simplify the computations, we have considered the gravitational influence of only three celestial bodies, namely, the Earth, the Sun and the Moon.

Additionally, the spin of the Earth leads to a redistribution of its mass which renders its shape an oblate spheroid; that makes the Earth slightly flatter around its poles and induces a gravitational perturbation,

$$\vec{a}_{J2} = -\frac{3}{2} \frac{J_2}{r^4} \mu_E \, R_E^2 \left[\left(1 - \frac{5}{r^2} r_z^2 \right) + \frac{2}{r} r_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right] \tag{6}$$

Furthermore, the solar radiation pressure (SRP) is an external force acting on the satellite, created by sunlight hitting the body's surface. The acceleration resulting from SRP is given by

$$\vec{a}_{SRP} = -\mu p_S C_r \frac{A}{m} \frac{\vec{r}_{Sat,S}}{r_{Sat,S}} \tag{7}$$

where C_r is a reflectivity coefficient, A is the cross-sectional area of the satellite, m represents its mass and p_S is the solar constant E divided by the speed of light.

The last acceleration that is computed in the orbit propagator is a consequence of the atmospheric drag; for this we have used the Jacchia-Roberts atmospheric model to get an approximation of the air density which is required for computations.

Positions and velocities result from integrating the acceleration with the Runge-Kutta-Fehlberg numerical method.

2.3 Artificial Neural Network's inputs and outputs

Section 2.1 revealed the geometric foundation upon which the Artificial Neural Network's objective is built. To achieve a better position $\vec{r}_n$, we chose to create a network with 12 inputs and a hidden layer of 34 neurons and a total of 816 connections.

The inputs consists of the:

  1. Initial position $\vec{r}_i$, defined by the three cartesian projections (r_{ix}, r_{iy}, r_{iz}),

2. Estimated position $\vec{r}_c$, defined by the three cartesian projections (r_{cx}, r_{cy}, r_{cz}),,

3. The error position $\vec{r}_{ec}$, which is considered to be zero, i.e., $(0, 0, 0)$, which means that we tell the network that we have the best estimation for the position,

4. The time duration of the estimation Δt,

5. The angle between the initial position vector and Moon's position vector, and

6. The angle between the initial position vector and Sun's position vector.

We have to consider the angles that $\vec{r}_i$ makes with the Sun's position vector and Moon's position vector to give additional information about the initial positions with respect to different moments in time during the year.

The output of the neural network consists of both the position of the body in orbit, $\vec{r}_n$, and the error position $\vec{r}_{ec}$.

3 Simulation results

The current section highlights insightful simulation results that ascertain the potential benefits of using Artificial Neural Networks to estimate a more accurate position of a satellite that has a Low-Earth Orbit. Details regarding the network's architecture and the structure of its input data were discussed in Section 2.

For the training phase, 31960 random inputs were selected from the GRACE-FO space mission's position measurements, spanning an entire month (June 2018). As highlighted in Section 2, once a random position is chosen, the trajectory of the satellite is computed with an orbit propagator for two hours and the final estimated position is fed into the Network.

To validate the success of the training, 347000 new input data was selected from the same month. From the new given inputs, 333215 (96%) cases gave a smaller error magnitude, as it's defined in Section 2. Figure 2 illustrates the magnitude of the errors.
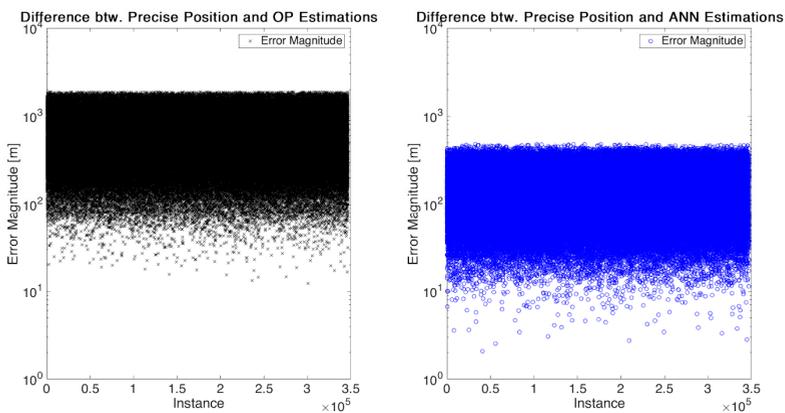

Figure 2. Error magnitude comparison (input data from June 2018).

On the left side we plotted the magnitude of the error between the real position of the satellite – two hours apart from the initial position – and it's estimated position, using only the orbit propagator. The magnitude is marked with an 'x' on a semi-logarithmic scale; y-axis renders these errors in meters for each instance of the input on the x-axis. The right side displays the error magnitude of the position obtained with the Artificial Neural Network,

marked with 'o'. For the 333215 favorable cases, the error improvement is illustrated in Figure 3, revealing that the error improvement in most cases revolves around 200 m.

As previously mentioned, the network was trained with input data collected from June 2018. Without additional training, 23666 random initial positions were selected from November 2018. Out of the specified inputs, 16983 (71.76%) cases had an improvement in the positioning estimation.

Figure 4 shows the error magnitude comparison for the new set of inputs, with the same interpretation as in Figure 2. Although the positioning refinement does not outweigh the performance of the orbital propagator, the results indicate that the network learned to compensate for the effects that were not modelled in the orbit propagator.
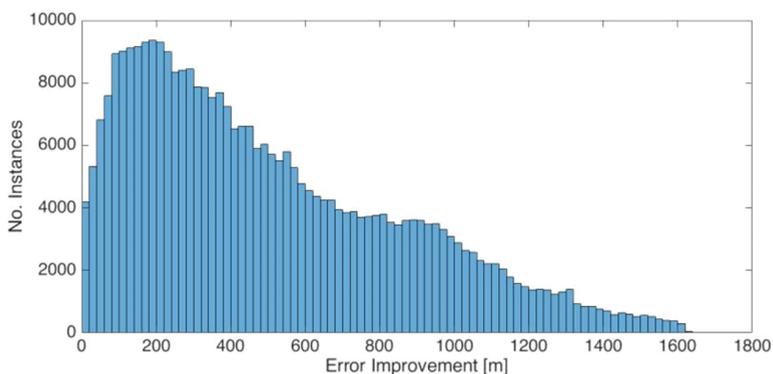

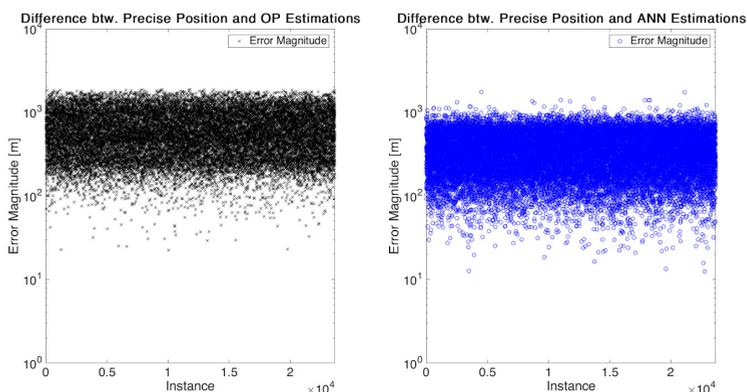
Figure 3. Error improvement (input data from June 2018).

Figure 4. Error magnitude comparison (input data from November 2018).

4 Conclusion

The paper emphasized the potential benefits of using Artificial Neural Networks for the improvement of positioning estimation. Section 2 discussed the design and implementation of the propagator and the network. Section 3 presented the results based on a scenario where GNSS data is given as an initialization; with the acquired data, an orbit propagator estimates the satellite's position after two hours and an Artificial Neural Networks uses the estimations to improve the positioning of the satellite. Our future research objective is to design an

Artificial Neural Network that improves the positioning with a flexible initialization frequency.

References

1. David A. Vallado, Fundamentals of Astrodynamics and Applications (Space Technology Series, 1997) 945

2. Hui Ying Wen et al., Gravity Recovery and Climate Experiment Follow-On (GRACE-FO) Level-1 Data Product User Handbook (NASA Jet Propulsion Laboratory, California Institute of Technology, 2019) 57

3. I. A. Basheer, M. Hajmeer, Journal of Microbiological Methods Artificial neural networks: fundamentals, computing, design, and application, 30 (2001)

4. Kleanthis Neokleous, Master's Thesis Modeling and Control of a Satellite's Geostationary Orbit, 101 (2006)

5. P. Rosa et al., 7th International Workshop on Numerical Modelling in Aerospace Sciences Functional Engineering Simulator for Close-Proximity Operations: TheProba-3 RendezVous eXperiment (P3RVX) FES, (2019)