This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier’s archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright
GPS orbit approximation using radial basis function networks

Polona Pavlovič Prešeren *, Bojan Stopar

University of Ljubljana, Faculty of Civil and Geodetic Engineering, Jamova cesta 2, Ljubljana 1000, Slovenia

A R T I C L E   I N F O

Article history:
Received 5 October 2006
Received in revised form 31 July 2007
Accepted 2 February 2008

Keywords:
GPS orbit computation
Artificial neural network
Radial basis function networks (RBFN)
Broadcast orbit correction
Polynomial and trigonometric interpolation
Extrapolation

A B S T R A C T

We present solutions for GPS orbit computation from broadcast and precise ephemerides using a group of artificial neural networks (ANNs), i.e. radial basis function networks (RBFNs). The problem of broadcast orbit correction, resulting from precise ephemerides, has already been solved using traditional polynomial and trigonometric interpolation. As an alternative approach RBFN broadcast orbit correction produces results within the accuracy range of the traditional methods. Our study shows RBFN broadcast orbit correction performs well also near the end of data intervals and for short data spans (~20 min). Regarding limitations of polynomial and trigonometric extrapolation, the most significant advantage of using RBFNs over the traditional methods for GPS broadcast orbit approximation arises from its short time prediction capability.

1. Introduction

The problem of orbit computation has been an important astrodynamical topic for several centuries. In recent decades, the need for precision satellite geodesy has given rise to even more intense investigation of the problem because the satellites represent reference points for satellite-based observations. The final precision and accuracy derived from satellite geodesy depends strongly on how well satellite positions can be determined.

GPS orbit computation methods are based on ephemerides, acquisition mode, time delay of access, and data form. The GPS operational control segment (OCS) generates and uploads broadcast ephemerides. To meet higher precision and accuracy needs, GNSS (Global Navigation Satellite Service) services, like IGS (International GNSS Service), prepare and distribute precise GNSS ephemerides over the web. The greater the accuracy, the greater the time of availability.

Broadcast and precise ephemerides use different data. Therefore, each uses different orbit construction methods. Broadcast orbits are generated using a method, proposed in ICD-GPS-200C, Table 20-JV 1997. Several methods are used for determining precise orbits. For complete precise ephemerides (position and velocity), precise but time consuming numerical integration is used (Montebruck and Gill, 2000). Numerical integration with collocation method works for positions alone and performs faster (Hugentobler et al., 2005). Interpolation polynomial construction performs well also for positions only. Precise GPS ephemerides interpolation experiments show that interpolation polynomials, constructed from n evenly spaced points, work best for central subinterval interpolation (Sagovac et al., 1995). Errors between the interpolating polynomial and estimated values grow worse near the end-points. A solution to this problem is several successive interpolations. This is time consuming, and the problem is not resolved at the beginning and at the end of a 24 h precise ephemerides interval unless it is linked to that from the previous and following days. In the successive polynomial method, Lagrange interpolation is used with polynomial order varying from the 8th to 11th degree (Remondi, 1991). Neville interpolation avoids some of the limitations of the Lagrange interpolation, but at least 10 nodes are needed (Schüler, 2001). In computation optimization, use of differences tables would be more efficient than the direct construction of polynomials (Neta et al., 1996). A recent study by Feng and Zheng (2005) introduced the results from analyses that used several different interpolation methods (Lagrange, Chebyshev, trigonometric). The study showed that a 19- or 20-degree trigonometric function was the most efficient interpolator for an orbital arc, and achieved 1 cm of 3D accuracy. In addition, a 9-term trigonometric function produced optimal interpolation results for a 2 h orbit arc even for near the end of the arc.

The broadcast orbit corrections treatment, where GPS broadcast orbits are used as a reference for the precise broadcast
orbit correction interpolation, is followed in the present paper. The precise broadcast orbit corrections treatment takes the advantages of broadcast and precise ephemerides and improves the weakness of each one. The advantage of using broadcast ephemerides is that they tie computationally to a simple continuous function. The disadvantage is that accuracy degrades over time. Precise ephemerides have better accuracy that is constant for the whole interval, but these data are also available in smaller discrete time intervals (usually 15 min of the SP3 files) (Spofford and Remondi, 1999). Precise broadcast orbit corrections are first restricted to a discrete 15 min time interval and actually result from position differences from precise and broadcast ephemerides. The RBFN method produces continuous functions also for the precise broadcast orbit corrections. Based on such a background, we use broadcast ephemerides for continuous orbit reconstruction. In addition RBFN continuous function results, acquired from discrete precise broadcast orbit corrections training, are used to diminish the broadcast orbit accuracy degradation in any time.

2. Artificial neural networks

ANN provides an alternative to the traditional orbit approximation methods. ANN have universal approximation properties for continuous real-valued functions (Cybenko, 1989; Funahashi, 1989; Ito, 1991, 1992; Castro et al., 2000). Their characteristics enable solving of problems without prior mathematical knowledge by introducing input to known output data in a process called learning (or training). ANN (in greater detail RBFN) has previously been used to solve nonlinear GPS pseudo-range equations (Jwo, 2004) and determine orbit using observation data (Sinha et al., 2000). We use only RBFN ANN because they perform better during short training times and are better at approximation when trained using small data sets.

2.1. Radial basis function networks

RBFN (Xu et al., 1994; Billings and Zheng, 1995; Fung et al., 1996) run feed-forward from input to output over a hidden layer (Fig. 1). Their basic characteristic is that they are restricted to a single hidden layer. Each hidden unit (neuron) implements a radial basis activation function. Contrary to the multi-layer perceptron where activation functions of a neuron are defined per iteration of the supervised learning: $w_{jk}^{new} = w_{jk}^{old} + \Delta w_{jk}$,

where $w_{jk}$ is the weights initialized using small random values, and then they are tuned at each iteration of the supervised learning:

$$E = \frac{1}{2} \sum_{k=1}^{O} (y_k - o_k)^2,$$

where $k = 1, ... , O$ and $y_k$ are the elements of the output feature vector $y = [y_1, y_2, ... , y_O]^T$. At the beginning, weights are initialized using small random values, and then they are tuned at each iteration of the supervised learning:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = \eta (y_k - o_k) h_j(x) = \eta \delta_j h_j(x)$$

with a non-fixed learning rate $\eta$. These operations are repeated until the maximum number of iterations is reached or until the prediction error is less than some given threshold.

2.2. RBFN for orbit approximation

Using the RBFN architecture shown in Fig. 1, our RBFN used for orbit approximation consisted of one neuron in the input layer (time) and three neurons in the output layer, representing the three components of the precise broadcast orbit corrections ($\Delta x$, $\Delta y$, $\Delta z$). This architecture refers to a single set of broadcast ephemerides since ANNs can approximate continuous function from discrete data. The 24 h discrete precise broadcast orbit corrections we use, acquired from different sets of broadcast ephemerides, do not satisfy the continuous function requirement.
Differences between control and RBFN estimated values are evaluated using a test data set (all 5 min position data from the SP3 files). RBFN training success was measured by how closely the RBFN output matched the control 5 min data orbits. The measure of RBFN computation success was taken as the difference of testing data (2nd, 3rd, 5th, 6th, 10th (0 h 45 min), 16th (1 h 15 min), 19th (1 h 30 min), 22nd (1 h 45 min), 25th (2 h 00 min) training epoch and none of the differences of testing data (2nd, 3rd, 5th, 6th,...,24th) exceed 2 mm in magnitude.

Further analysis was aimed at proper training and testing data selection from 15 min segment of the 2 h intervals. We constructed tests using 12 different RBFNs for satellite PRN1 for one 24 h interval. After training and testing such a RBFN architecture was used to determine orbits for all of the satellites. Finally, we use RBFN extrapolation experiments to show the RBFN learning effects.

3. Numerical examples

In this work all tests were done using precise broadcast orbit corrections. For RBFN orbit computation validations, we compared our results with 5 min control data. The tests show the effectiveness of RBFN approximation near the end-points of the data arcs. Further analysis was aimed at proper training and testing data selection from 15 min segment of the 2 h intervals. We constructed tests using 12 different RBFNs for satellite PRN1 for one 24 h interval. After training and testing such a RBFN architecture was used to determine orbits for all of the satellites. Finally, we use RBFN extrapolation experiments to show the RBFN learning effects.

3.1. Preprocessing of data

Precise ephemerides were imported from the IGS web site. In SP3 files position data for all active GPS-satellites are given in the IGS reference system for a single 15 min interval. For Block II/IIA satellites we incorporate a phase center offset correction in \( \times \)-component of 1.023 m, while for Block IIR satellites the center of mass and phase center are considered equal. IGS precise ephemeride accuracy range is approximately 5 cm. This level of orbits accuracy is about the level of coordinate differences between the IGS reference system to the WGS-84 (G873) system, so we omit any IGS to WGS-84 reference system transformations.

Broadcast ephemeris data for all satellites for 24 h periods were imported from the IGS ftp server. An alternative would be to use broadcast ephemerides stored in receivers, but usually these files contain only data for satellites in view. All of the broadcast ephemeris data were checked for outliers and health (e.g. problems with satellite maneuvers). Occasionally extreme position errors were found in the broadcast ephemerides (50 m or more). In accordance with the precise ephemerides, 15 min time intervals were used for broadcast orbit intervals and further comparison. When a new set of broadcast ephemerides was introduced, broadcast orbit positions were calculated using a new set of data. Broadcast orbits were calculated in earth center earth fixed reference frame (ECEF).

3.2. Implementation of RBFN

For RBFN learning, the Matlab® Neural Network Toolbox function newrb was used. In this function ANN training is based on iterative augmentation of the RBF (in our case Gaussian) in the hidden layer until the sum-squared error minimization of a training data set was satisfied. At each computational step a new RBF is created from the input vector that minimizes error. If this method failed to satisfy initial conditions, the new neuron in the hidden layer was added, and a new iteration was run and a new error was computed.

3.3. The RBFN orbit computation validation

Our RBFN orbit computation output was next compared to positions computed from other orbit computation methods. Testing data of 5 min were adopted from the ECF_5MIN.200 file2 from the program ATEST (Schenewerk, 2003). Our RBFN solutions were compared to 5 min positions from the IGS rapid orbits. The measure of RBFN computation success was taken as how closely the RBFN output matched the control 5 min data from the ECF_5MIN.200 files. RBFN training success was evaluated using a test data set (all 5 min position data from the IGS rapid orbits) and comparing actual differences between the broadcast and precise orbit position and differences, estimated by RBFN.

Fig. 3 depicts the result from a single RBFN training session. Differences between control and RBFN estimated values are shown for each vector component for individual 5 min intervals. For the 2 h interval shown there were 25 comparisons made. We used only data with 15 min interval step for training (actual data from the SP3 files). From these, we excluded two data (0 h 30 min and 1 h 00 min) for additional tests. Such treating was based on the presumption that, in general, only 15 min data are available in SP3 orbit files and from those we have to select the training and testing data set.

Fig. 3 shows that there are small differences (\(<0.5\) mm) in the data used for 1st (epoch 0 h 00 min), 4th (epoch 0 h 15 min), 10th (0 h 45 min), 16th (1 h 15 min), 19th (1 h 30 min), 22nd (1 h 45 min), 25th (2 h 00 min) training epoch and none of the differences of testing data (2nd, 3rd, 5th, 6th,...,24th) exceed 2 mm in magnitude.

Further analysis was aimed at proper training and testing data selection from 15 min segment of the 2 h intervals. We constructed tests using 12 different RBFNs for satellite PRN1 for one 24 h interval. First we have used five training and four testing

---

discrete 15 min data in each 2 h interval, next seven training and two testing data, finally we have used all nine available 15 min discrete data in the 2 h interval for training. For the final estimation of RBFN training success the following statistics were used:

- minimum value of differences $\delta_{\text{min}}$
- maximum value of differences $\delta_{\text{max}}$
- mean value of differences $\bar{\delta} = \sum_i \delta_{\text{i}} / N_{\text{i-o}}$
- standard deviation of differences:

$$s = \frac{1}{N_{\text{i-o}} - 1} \sqrt{\sum_i (\delta_i - \bar{\delta})^2},$$

where $N_{\text{i-o}}$ is the number of the input–output training data.

If we use only five discrete 15 min data from nine available 2 h data for each RBFN training (i.e. epochs: hh 00 min, hh 15 min, hh 45 min, (hh+1) 00 min, (hh+1) 15 min, (hh+1) 30 min, (hh+1) 45 min, (hh+2) 00 min, where hh means hours), there are always four discrete 15 min data left for each RBFN testing. In one 24 h period 12 sets of broadcast ephemerides are introduced, consequently we train 12 different RBFNs. If we compare all available RBFN generated broadcast orbit corrections for 5 min interval of 24 h to 5 min precise broadcast orbit corrections from ECF_5MIN.200 file, we get the statistics shown in Table 1.

As shown in Fig. 4, RBFN training proved successful. The maximum difference in each position vector component ($x, y, z$) is less than 4 cm and is at most 2.7 cm. In addition, the distribution of deviations is that of a normal distribution.

For further analysis we chose seven training data with a 15 min time interval (i.e. epochs: hh 00 min, hh 15 min, hh 45 min, (hh+1) 15 min, (hh+1) 30 min, (hh+1) 45 min, (hh+2) 00 min, where hh means hours) to try achieving better results. Using 12 RBFNs we acquire the following statistics for one 24 h time interval (Table 2, see also Fig. 5).

Comparison with our earlier result (Tables 2 and 1) shows that this RBFN training run also proved successful. The deviations in each component are $\pm 1$ cm at most and the distribution of deviations approaches normal distribution. Fig. 6 illustrates that the maximal deviations are about four times smaller in the second run. The only change we made from the first run was in the use of number of training data.

![Fig. 3. Single RBFN training and testing results (2 h) with seven training 15 min data selection.](image)

<table>
<thead>
<tr>
<th>$\delta_{\text{min}}$ (m)</th>
<th>$\delta_{\text{max}}$ (m)</th>
<th>$\bar{\delta}$ (m)</th>
<th>$s$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta x$</td>
<td>$-0.0278$</td>
<td>$0.0340$</td>
<td>$0.0000$</td>
</tr>
<tr>
<td>$\Delta y$</td>
<td>$-0.0264$</td>
<td>$0.0310$</td>
<td>$0.0013$</td>
</tr>
<tr>
<td>$\Delta z$</td>
<td>$-0.0168$</td>
<td>$0.0287$</td>
<td>$-0.0001$</td>
</tr>
<tr>
<td>$\Delta r$</td>
<td>$0$</td>
<td>$0.0440$</td>
<td>$0.0099$</td>
</tr>
</tbody>
</table>

Each RBFN has 2 h validation and it is constructed from five training data with the 15 min time interval.
Using all available training data from standard SP3 files (epochs: hh 00 min, hh 15 min, hh 30 min, hh 45 min, (hh+1) 00 min, (hh+1) 15 min, hh 30 min, (hh+1) 45 min, (hh+2) 00 min, where hh means hours), and the same RBFN characteristics, we were next able to achieve even better results. Maximum deviations in each component were less than 7 mm (Table 3, Fig. 6). During such training data selection we did not have additional 15 min testing data available when we used only data from standard SP3 files; however, in our study we have 5 min testing data from ECF_5MIN.200 file. Generally if we have only SP3 files available testing data had to be determined using other orbit interpolation techniques.

Fig. 7 shows our resulting approximation error for a selected day using data from 27 active satellites. Using 12 different RBFNs for each set of broadcast ephemerides and seven training 15 min data gives an average approximation error of about 3 mm. For validation we have used 5 min data from ECF_5MIN.200 file.

### Table 2

Results of 12 RBFN training for one 24 h period.

<table>
<thead>
<tr>
<th></th>
<th>$\delta_{\text{min}}$ (m)</th>
<th>$\delta_{\text{max}}$ (m)</th>
<th>$\bar{\delta}$ (m)</th>
<th>$\sigma$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta X$</td>
<td>-0.0079</td>
<td>0.0095</td>
<td>-0.00006</td>
<td>0.0020</td>
</tr>
<tr>
<td>$\Delta Y$</td>
<td>-0.0060</td>
<td>0.0054</td>
<td>-0.00007</td>
<td>0.0016</td>
</tr>
<tr>
<td>$\Delta Z$</td>
<td>-0.0035</td>
<td>0.0027</td>
<td>-0.00006</td>
<td>0.0010</td>
</tr>
<tr>
<td>$\Delta r$</td>
<td>0</td>
<td>0.0095</td>
<td>0.00201</td>
<td>0.0018</td>
</tr>
</tbody>
</table>

Each RBFN is valid for 2 h and it is constructed upon seven data with the 15 min time interval.

### 4. Extrapolation

Computing satellite positions before introducing new SP3 files or broadcast ephemerides set can be advantageous. To show the benefits of the RBFN orbit performance for prediction we have used broadcast and precise data to compute satellite positions over the last known precise broadcast orbit correction data. The 3D accuracy of IGS ultra-rapid (predicted) products is about 10–25 cm, so we use this value for a measure of the learning success. The summary in Table 4 shows that RBFN extrapolation method performs well over the 30 min time interval for the satellite PRN 7. Let us emphasize we used precise broadcast orbital corrections, introduced from 2 h before, for training. Fig. 8 emphasizes this by plotting differences in all three components.

We conclude that RBFN orbit computation using several different approaches is as effective as the more traditional interpolation techniques. Unlike traditional
Fig. 5. Histogram for one 24 h period: seven training data (15 min interval) for each RBFN with 2 h validation, testing data refer to a 5 min interval, in 24 h period 12 RBFNs with 2 h validation were used.

Fig. 6. Histogram for one 24 h period: nine training data (all available 15 min data in SP3 files) for one RBFN. In analysis we have used 5 min data for testing phase.
interpolation techniques RBFN also performs well during extrapolation over short time spans (ca 20 min).

Using RBFN for prediction, it performed well for all the satellites 20 min after last known precise broadcast orbit correction used for training (time: 2 h and 20 min). This means RBFN was trained on precise broadcast orbit corrections in the interval from 0 to 2 h, but it has been used to compute broadcast orbit correction for time 2 h and 20 min. Extrapolated results were

### Table 3
Results of 12 RBFN training for one 24 h period.

<table>
<thead>
<tr>
<th>δ_{min} (m)</th>
<th>δ_{max} (m)</th>
<th>δ (m)</th>
<th>s (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ΔX</td>
<td>−0.0048</td>
<td>0.0064</td>
<td>0.00000</td>
</tr>
<tr>
<td>ΔY</td>
<td>−0.0033</td>
<td>0.0065</td>
<td>0.00007</td>
</tr>
<tr>
<td>ΔZ</td>
<td>−0.0029</td>
<td>0.0028</td>
<td>0.00000</td>
</tr>
<tr>
<td>Δr</td>
<td>0</td>
<td>0.0073</td>
<td>0.00148</td>
</tr>
</tbody>
</table>

Each RBFN is constructed upon nine 15 min data; each RBFN has a 2 h time validation.

### Table 4
Extrapolation over time limits in the SP3 files for the satellite PRN 7.

<table>
<thead>
<tr>
<th>Time (h:mm)</th>
<th>ΔX (m)</th>
<th>ΔY (m)</th>
<th>ΔZ (m)</th>
<th>Δr (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:00</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2:05</td>
<td>−0.0005</td>
<td>0.0021</td>
<td>−0.0008</td>
<td>0.0023</td>
</tr>
<tr>
<td>2:10</td>
<td>−0.0007</td>
<td>0.0067</td>
<td>0.0007</td>
<td>0.0068</td>
</tr>
<tr>
<td>2:15</td>
<td>−0.0004</td>
<td>0.0146</td>
<td>0.0049</td>
<td>0.0154</td>
</tr>
<tr>
<td>2:20</td>
<td>0.0012</td>
<td>0.0277</td>
<td>0.0114</td>
<td>0.0300</td>
</tr>
<tr>
<td>2:25</td>
<td>0.0053</td>
<td>0.0475</td>
<td>0.0231</td>
<td>0.0531</td>
</tr>
<tr>
<td>2:30</td>
<td>0.0113</td>
<td>0.0773</td>
<td>0.0425</td>
<td>0.0890</td>
</tr>
<tr>
<td>2:35</td>
<td>0.0219</td>
<td>0.1208</td>
<td>0.0696</td>
<td>0.1411</td>
</tr>
<tr>
<td>2:40</td>
<td>0.0401</td>
<td>0.1798</td>
<td>0.1088</td>
<td>0.2140</td>
</tr>
<tr>
<td>2:45</td>
<td>0.0674</td>
<td>0.2591</td>
<td>0.1634</td>
<td>0.3136</td>
</tr>
<tr>
<td>3:00</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>3:00</td>
<td>0.2534</td>
<td>0.6715</td>
<td>0.4601</td>
<td>0.8525</td>
</tr>
</tbody>
</table>

Fig. 7. Resulting approximation error for 1 January 2002. Each 2 h a new RBFN was used and trained using seven data. Testing data refer to ECF_5MIN.200 file.
does not permit a single ANN network to train over long observation times. When a new set of broadcast ephemerides are introduced, a new ANN must be started. Because new broadcast ephemerides generally appear every second hour, we needed to use 12 sets of ANN networks to train during a single 24 h period. Proper ANN selection is critical. As we demonstrated, RBFN selection is effective in achieving desired computation accuracy.

The effectiveness of RBFN learning depends on training data selection. The most effective learning uses the first two and the last two input–output data pairs for training. From the others we chose two pairs for testing. Such geometry performed well also for prediction over the 20 min time interval beforehand with the accuracy of the IGS ultra-rapid ephemerides.

5. Conclusions

A wide spectrum of computation methods have been traditionally used to obtain precise satellite positions from broadcast and precise ephemeris data. This paper presents an alternative to the traditional methods, using ANN. ANN computation needs discrete broadcast to precise ephemerides differences upon which training and testing are applied. The inconsistent nature of positions resulting from different sets of broadcast ephemerides compared to precise broadcast orbit corrections, obtained from new broadcast and precise ephemerides introduction. The maximum difference obtained was 23 cm. In principle a method could be applicable in a situation when new broadcast and/or precise ephemerides were not available (Fig. 9).

References


