THE JOURNAL OF NAVIGATION (2005), **58**, 105–118. © The Royal Institute of Navigation
DOI: 10.1017/S0373463304003017 Printed in the United Kingdom

GPS Navigation Solutions by Analogue Neural Network Least-Squares Processors

Dah-Jing Jwo

(*Department of Communications and Guidance Engineering*,
National Taiwan Ocean University)
(Email: djjwo@mail.ntou.edu.tw)

The solution for the receiver's position and clock bias using four or more GPS pseudo-range measurements involve nonlinear quadratic equations. One of the popular techniques attempts to linearise the equations and solve them by the least-squares (LS) scheme based on an iterative gradient approach. For real-time applications when the solution is to be obtained within a time of the order of a hundred nanoseconds, a digital computer often cannot comply with the desired computation time, or its use is too expensive. In this paper, two ordinary differential equation formulation schemes and corresponding circuits of neuron-like analog processors will be employed for GPS navigation processing. The circuits of simple neuron-like analog processors are employed essentially for solving systems of linear equations based on the criterion of mean square error minimization, which is commonly utilized for determining positioning solutions in the GPS receivers. Experiments on single epoch and thereafter dynamic positioning will be conducted by computer simulation to validate the usefulness of the proposed scheme. The solutions will be assessed and compared to those provided by the conventional method in which pseudo-inverse matrix calculation by digital computer is involved.

KEY WORDS

1. GPS. 2. Real-time Neurocomputing. 3. Neural Networks. 4. Analogue Processors.

1. INTRODUCTION. The solution for the receiver's position and clock bias from four or more GPS pseudorange measurements involves nonlinear quadratic equations (Bancroft, 1985; Krause, 1987; Axelrad and Brown, 1996; Lundberg, 2001). One of the popular techniques attempts to linearise the equations and solve them by the least squares (LS) scheme in which pseudo-inverse matrix calculation by digital computer is involved.

Although numerical methods are quite useful and commonly used, they suffer from the intrinsic intensive computation complexity and require a considerable amount of computer time. For real-time applications, the solution is required within a time of the order of a hundred nanoseconds. As an example, the fast convergence speed becomes particularly important in deep space where the initial estimate may contain large errors. In such cases, a digital computer often cannot comply with the desired

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

computation time, or its use is too expensive, and, the neural network (NN) approach provides a promising and very realistic computational alternative.

The application of NN approach for navigation solution processing has not been widely explored yet in the GPS community. Simon and El-Sherief initially proposed the NN approach to approximate and classify the Geometric Dilution of Precision (GDOP) factors (1995). In their work, the NNs were employed to perform the function approximation and group classification, in which the NNs were used to learn the functional relationships between the entries of a measurement matrix and the eigenvalues of its inverses, and thus generated GDOP. Chansarkar (2000) solved the GPS pseudorange equations using a three-layer radial basis function (RBF) neural network. However, there are some difficulties in his approach. For example, the NN needs to be trained for specific numbers of satellite measurements. Also, large errors might occur once the geographic positions for training are different from positions for testing.

In the present paper, further extension of the NN application to GPS navigation problem will be conducted. The circuits of simple analog neural network least-squares processors are employed to solve the systems of linear equations. Two ordinary differential equation formulation schemes and corresponding circuit architectures are discussed. One of the electronic implementations of a neural network-like processor employed in the present work would be able to determine the navigation solution virtually instantaneously. This would enable the use of high-integrity navigation solutions without the delay required for many pseudo-inverse matrix calculations and is ideally suited for GPS navigation application. The advantages for the proposed method also include that the training stage is not required. Experiments via computer simulation will be conducted for single epoch solutions and then for dynamic positioning solutions. The properties and performance of the navigation solutions based on the method in which the analogue NN least-squares processor is employed will be assessed and compared to those provided by the conventional method where pseudo-inverse matrix calculation by digital computer is involved.

This paper is organized as follows. In Section 2, preliminary background on GPS LS solutions is briefly reviewed. The neural network-like architectures for solving systems of linear equations are introduced in Section 3. In Section 4, simulation examples are presented; performance assessment by incorporating neural network-like processors for GPS navigation will be provided. Section 5 gives the conclusions.

2. GPS SOLUTIONS USING LEAST-SQUARES APPROACH.

The GPS measurements and the errors are briefly reviewed. Consider the vectors relating the Earth's centre, satellites and user position. The vector $\mathbf{s}$ represents the vector from the Earth's centre to a satellite, $\mathbf{u}$ represents the vector from the Earth's centre to the user's position, the pseudorange ρ_i is defined for the ith satellite by

$$\rho_i = \|\mathbf{s}_i - \mathbf{u}\| + ct_b + v_{\rho_i} \tag{1}$$

where c is the speed of light and t_b is the receiver clock offset from system time, and v_{ρ_i} is the pseudorange measurement noise. Consider the user position in three dimensions, denoted by (x_u, y_u, z_u), the GPS pseudorange measurements made to

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

the n satellites can then be written as

$$\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + ct_b + v_{\rho_i}, \quad i = 1, \ldots, n \tag{2}$$

where (x_i, y_i, z_i) denotes the i-th satellite's position in three dimensions.

Equation 2 can be linearised by expanding Taylor's series around the approximate (or nominal) user position $(\hat{x}_n, \hat{y}_n, \hat{z}_n)$ and neglecting the higher terms. Defining $\hat{\rho}_i$ as ρ_i at $(\hat{x}_n, \hat{y}_n, \hat{z}_n)$ gives

$$\Delta\rho_i = \rho_i - \hat{\rho}_i = e_{i1}\Delta x_u + e_{i2}\Delta y_u + e_{i3}\Delta z_u + ct_b + v_{\rho_i} \tag{3}$$

where

$$\begin{gathered} e_{i1} = \frac{\hat{x}_n - x_i}{\hat{r}_i}; \quad e_{i2} = \frac{\hat{y}_n - y_i}{\hat{r}_i}; \quad e_{i3} = \frac{\hat{z}_n - z_i}{\hat{r}_i} \\ \hat{r}_i = \sqrt{(\hat{x}_n - x_i)^2 + (\hat{y}_n - y_i)^2 + (\hat{z}_n - z_i)^2} \end{gathered} \tag{4}$$

The vector $(e_{i1}, e_{i2}, e_{i3}) = \mathbf{E}_i$, $i = 1, \ldots, n$, denotes the line-of-sight vector from the user to the satellites. Equation 3 can be written in a matrix formulation

$$\begin{aligned} \Delta\rho_i &= [\Delta\rho_1 \quad \Delta\rho_2 \quad \Delta\rho_3 \quad \cdots \quad \Delta\rho_n]^T \\ &= \begin{bmatrix} e_{11} & e_{12} & e_{13} & 1 \\ e_{21} & e_{22} & e_{23} & 1 \\ e_{31} & e_{32} & e_{33} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ e_{n1} & e_{n2} & e_{n3} & 1 \end{bmatrix} \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ ct_b \end{bmatrix} + v_{\rho_i} \end{aligned} \tag{5}$$

which can be represented as

$$\Delta\mathbf{z} = \mathbf{H}\Delta\mathbf{x} + \mathbf{v} \tag{6}$$

The dimension of matrix $\mathbf{H}$ is $n \times 4$ with $n \geqslant 4$, and $\mathbf{H}$ is usually referred to as the 'geometry matrix' or 'visibility matrix'. The $\mathbf{v}$ are assumed to be zero-mean, so that the least-squares solution to Equation 6 is given by the follows:

$$\Delta\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\Delta\mathbf{z} \tag{7}$$

3. NEURON-LIKE ARCHITECTURES FOR SOLVING SYSTEMS OF LINEAR EQUATIONS.

The solution of a set of linear equations (or equivalently, an online calculation of pseudo-inverse matrices) is desired. The analog circuit solution is particular attractive in real-time applications. For real-time applications when the solution is to be obtained within a time of the order of a hundred nanoseconds, a digital computer often cannot comply with the desired computation time, or its use is too expensive. There are many different ways to connect neuron-like computing units (cells) into a large network. These different patterns of connections between the cells are called architectures or circuit structures.

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

Assume that a set of linear algebraic equations is to be solved

$$\mathbf{Ax} \cong \mathbf{b} \tag{8}$$

which can be written in scalar form as

$$\sum_{j=1}^{n} a_{ij}x_j \cong b_i \quad (i=1,2,\cdots,m)$$

Here, $\mathbf{A}=[a_{ij}]$ is the $m\times n$ real coefficient matrix with known elements, $\mathbf{x}$ is the n-dimensional unknown vector, and $\mathbf{b}$ is the m-dimensional observation vector (it should be noted that m can be less than, equal to or greater than n). The ordinary LS problem is one in which we minimize the energy function:

$$E_2(\mathbf{x})=\frac{1}{2}\sum_{i=1}^{m} r_i^2(\mathbf{x})=\frac{1}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2 \tag{9}$$

Using a general gradient approach for minimization of a function the problem formulated by Equation 9 can be mapped to a set of differential equations:

$$\begin{aligned}&\frac{d\mathbf{x}}{dt}=-\boldsymbol{\mu}(t)\nabla E_2(\mathbf{x})\\&\nabla E_2(\mathbf{x})=\mathbf{A}^T(\mathbf{Ax}-\mathbf{b}),\ \mathbf{x}(0)=\mathbf{x}^{(0)}\end{aligned} \tag{10}$$

where $\mathbf{x}=[x_1\,x_2\,\ldots\,x_n]^T$ and $\boldsymbol{\mu}(t)=[\mu_{ij}(t)]$ is an $n\times n$ positive-definite matrix that is often diagonal, and $\nabla E_2(\mathbf{x})$ is the gradient of the energy function, $E_2(\mathbf{x})$. Generally, the entries of the matrix $\boldsymbol{\mu}(t)$ depend on the time and the vector $\mathbf{x}$. Expressing Equation 13 in scalar form, we have

$$\frac{dx_j}{dt}=-\sum_{p=1}^{n}\mu_{jp}\left[\sum_{i=1}^{m}a_{ip}\left(\sum_{k=1}^{n}a_{ik}x_k-b_i\right)\right] \tag{11}$$

with $x_j(0)=x_j^{(0)}$, $m\geqslant n$, for $j=1,2,\ldots,n$. This is an initial value problem in which we can compute a trajectory $\mathbf{x}(t)$ starting at the initial point and has the solution when $t\to\infty$. The specific choice of the coefficients $\mu_{jp}(t)$ must ensure the stability of the differential equations and an appropriate convergence speed to the stability solution state. In order to reduce the influence of the outliers, the iteratively reweighted LS criterion can be employed

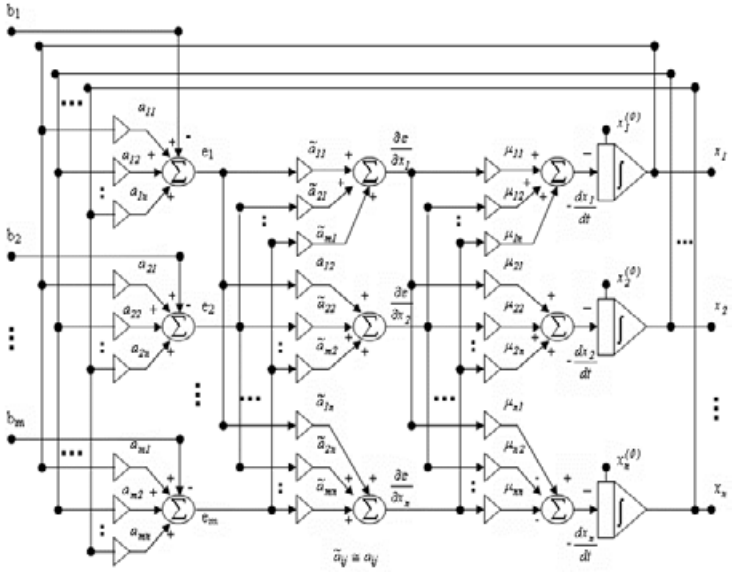
$$\frac{dx_j}{dt}=-\sum_{p=1}^{n}\mu_{jp}\left(\sum_{i=1}^{m}\left(a_{ip}g_i\left[\sum_{k=1}^{n}a_{ik}x_k-b_i\right]\right)\right) \tag{12}$$

where $g_i[\cdot]$ is the nonlinear sigmoid activation function defined as $g_i[r_i]=\beta tanh(\alpha r_i)$. The above system of differential equations can be rewritten as the set of nonlinear equations:

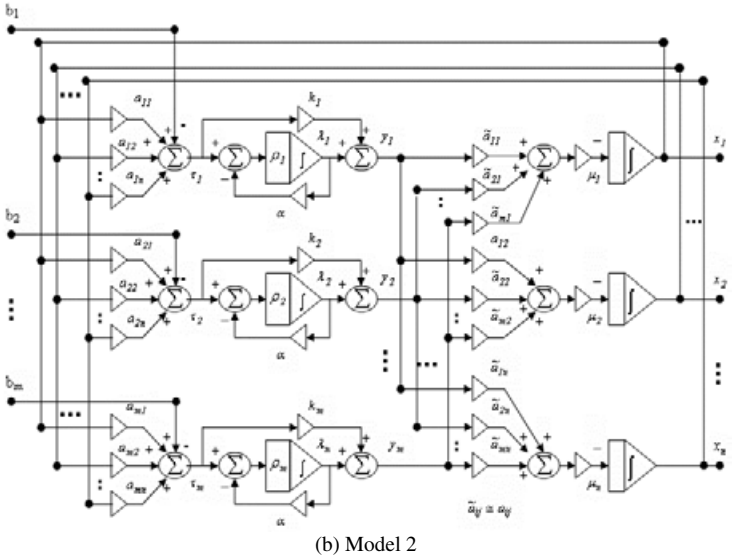
$$e_i(\mathbf{x})=g_i\left[\sum_{k=1}^{n}a_{ik}x_k-b_i\right],\quad i=1,2,\cdots,m \tag{13a}$$

$$\frac{\partial\varepsilon(\mathbf{x})}{\partial x_p}=\sum_{i=1}^{m}a_{ip}e_i(\mathbf{x}),\quad p=1,2,\cdots,n \tag{13b}$$

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press



(a) Model 1



(b) Model 2

Figure 1. Schematic architectures of the NN for solving a system of linear equations (Cichocki and Unbehauen, 1992).

$$\frac{dx_j}{dt} = -\sum_{p=1}^{n} \mu_{jp} \frac{\partial \varepsilon(\mathbf{x})}{\partial \mathbf{x}_p}, \quad x_j(0) = x_j^{(0)}, \quad j = 1, 2, \cdots, n \tag{13c}$$

which can be directly implemented by a neural network depicted in Figure 1 (a). The architecture of the network resembles the general Tank-Hopfield model.

The circuit structures (or, NN architectures) shown in Figure 1 (a) can be considerably simplified for some important special cases to get the advantage of relatively

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

few interconnections. Cichocki and Unbehauen (1992) proposed a NN architecture to improve convergence properties and accuracy. The architecture was based on minimization of the energy of augmented Lagrangian function, which is obtained from the ordinary Lagrangian by adding penalty terms. The resulting circuits can be transferred to the set of differential equations

$$\frac{dx_j}{dt} = -\mu_j \sum_{i=1}^{m} [k_i r_i(\mathbf{x}) + \lambda_i] a_{ij} \tag{14a}$$

$$\frac{d\lambda_i}{dt} = \rho_i [r_i(\mathbf{x}) - \alpha \lambda_i] \tag{14b}$$

for $j = 1, 2, \ldots, n$; $i = 1, 2, \ldots, m$ and with $\mu_j > 0$ and $\rho_i > 0$, or, in compact matrix form

$$\frac{d\mathbf{x}}{dt} = -\boldsymbol{\mu} \mathbf{A}^{\mathrm{T}} \mathbf{y} \tag{15a}$$

$$\frac{d\boldsymbol{\lambda}}{dt} = \boldsymbol{\rho} [(\mathbf{A}\mathbf{x} - \mathbf{b}) - \alpha \boldsymbol{\lambda}] \tag{15b}$$

which can be directly implemented by a neural network depicted in Figure 1 (b). In the following discussion, the circuit as in Equation (13) is referred to as the Model 1 structure, and the circuit as in Equation (14), or equivalently (15), is referred to as the Model 2 structure. More detailed discussion on the topic is in Cichocki and Unbehauen (1992).

4. STUDY EXAMPLES. Computer simulation is usually employed for evaluating the neural network models since analysis of these models via analytical methods is a difficult task due to the complexity of the neural networks, the large number of components and the nonlinearities associated with these components. To illustrate application of the neural network-like processors for GPS navigation, numerical experiment will be performed for the two circuit structures discussed in Section 3 (Equations 13 and 14).

Simulation was conducted on a notebook computer powered by 1·7 GHz Intel Pentium 4 mobile CPU. The computer code was written using the Matlab® 6·5 version software. Due to the limit of space, only some illustrated examples will be presented. Single epoch solutions involving four and eight measurements will be examined, followed by the dynamic positioning solutions involving five to eight satellites. Navigation solutions based on the proposed approach will be compared to those based conventional method where the pseudo-inverse matrix calculation by digital computer is involved.

4.1. *Evaluation on single epoch solutions.* Experiments on single epoch positioning are performed to assure the applicability of the present method on GPS navigation processing. Comparison of the results to those based on conventional method involving pseudo-inverse calculation is provided. The circuit parameters used in the following examples are:

$$\mu_o = \rho_o = \frac{1}{\tau} = 10^8, \quad \alpha = 0{\cdot}2, \ \beta = 5 \text{ and } k_j = 0{\cdot}1$$

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

Table 1. Simulated data for single epoch solution – 4 GPS satellites visible (in metres).

ECEF coordinates of 4 GPS satellites and the pseudorange observables				
GPS-1	7766188·44	−21960535·34	12522838·56	22228206·42
GPS-2	−25922679·66	−6629461·28	31864·37	24096139·11
GPS-3	−5743774·02	−25828319·92	1692757·72	21729070·63
GPS-4	−2786005·69	−15900725·80	21302003·49	21259581·09

Computed ECEF user's position and clock bias as a function of iteration number (Standard least-squares solutions with the initial nominal state vector at the centre of the Earth)

Iteration No.	*dx*	*dy*	*dz*	*cdt*
0	0000000·000	0000000·000	0000000·000	0000000·000
1	−2977571·476	−5635278·159	4304234·506	1625239·802
2	−2451728·534	−4730878·461	3573997·520	314070·733
3	−2430772·219	−4702375·802	3546603·872	264691·707
4	−2430745·096	−4702345·114	3546568·706	264691·130
5	−2430745·096	−4702345·114	3546568·706	264691·129

Table 2. Simulated data for single epoch solution – 8 GPS satellites visible (in meters).

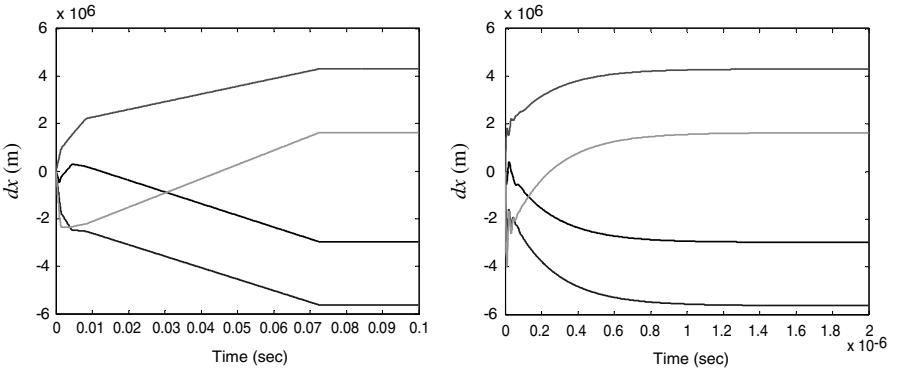
Geocentric coordinates of 8 GPS satellites and the pseudorange observables				
GPS-1	16414028·668	660383·618	20932036·907	24658975·317
GPS-2	16896800·648	−18784061·365	−7418318·856	22964286·412
GPS-3	9339639·616	−14514964·658	20305107·161	21338550·645
GPS-4	−18335582·591	−11640868·305	15028599·071	23606547·294
GPS-5	−2077142·705	−20987755·987	−15879741·196	24263298·504
GPS-6	−4957166·885	−23306741·039	12039027·096	20758264·108
GPS-7	17977519·820	−13089823·312	14331151·065	21847468·817
GPS-8	9682727·508	−24060519·485	3985404·530	20352077·193

Computed ECEF user's position and clock bias as a function of iteration number (Standard least-squares solutions with the initial nominal state vector at the centre of the Earth)

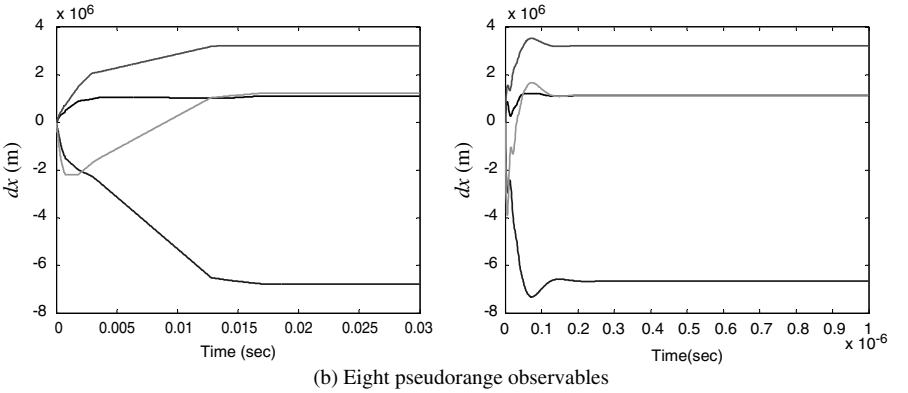
Iteration No.	*dx*	*dy*	*dz*	*cdt*
0	0000000·000	0000000·000	0000000·000	0000000·000
1	1117874·114	−6665513·833	3225078·117	1156492·923
2	963952·130	−5696556·680	2751432·902	−5750·594
3	961334·836	−5674088·368	2740543·515	−35984·242
4	961333·830	−5674076·370	2740537·661	−36000·000
5	961333·830	−5674076·370	2740537·661	−36000·000

The ECEF coordinates of GPS satellites and the pseudorange observables, and the computed user's position and clock bias by conventional least-squares method at each iteration step are listed in Table 1 for the four-observable case, and in Table 2 for the eight-observable case. Computer simulated state trajectories for the cases of four and eight observables using the two NN circuit structures are provided in

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press



(a) Four pseudorange observables



(b) Eight pseudorange observables

Figure 2. Convergence of the state trajectories at the 1st iteration step for the cases of (a) four, and (b) eight pseudorange observables, when the deviation between initial nominal solution and final solution is very large: Model 1 – left; Model 2 – right.)

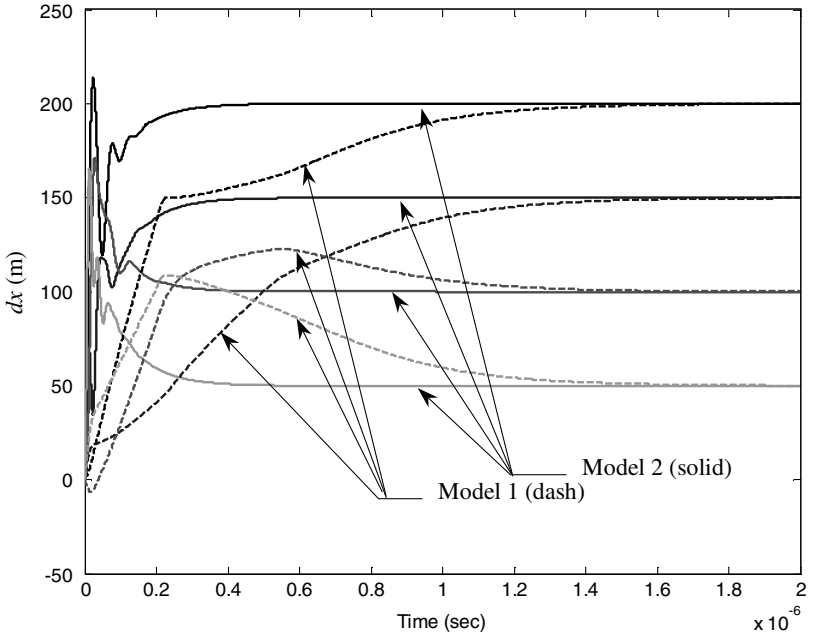
Figures 2 and 3. Experience shows that it normally takes five iterations to obtain the navigation solution, which is also true in the present work. The deviation between initial nominal solution and final solution (i.e., $\Delta\mathbf{x}$) has significant influence on the time to convergence. Ideally, when the positioning solution converges, the vector $\Delta\mathbf{x}$ at the last iteration step should approach to a zero vector (with size 4×1).

For comparison of the performance based on the two NN models, convergence histories of the state trajectories at the 1st iteration step are presented. The reason that result at the 1st iteration was selected is due to the fact that it takes the longest time ($\because$ largest $\Delta\mathbf{x}$) in all iterations at a certain epoch for convergence, so that the two NN models would reflect their distinguishing characteristics. For the example of four observables, when the deviation is very large, e.g., the initial nominal state vector is taken to be at the centre of the Earth $\mathbf{x} = [0\,0\,0\,0]^T$ m (the origin of the ECEF coordinate system), this results in large initial errors (which are also the steady-state solution to be solved for):

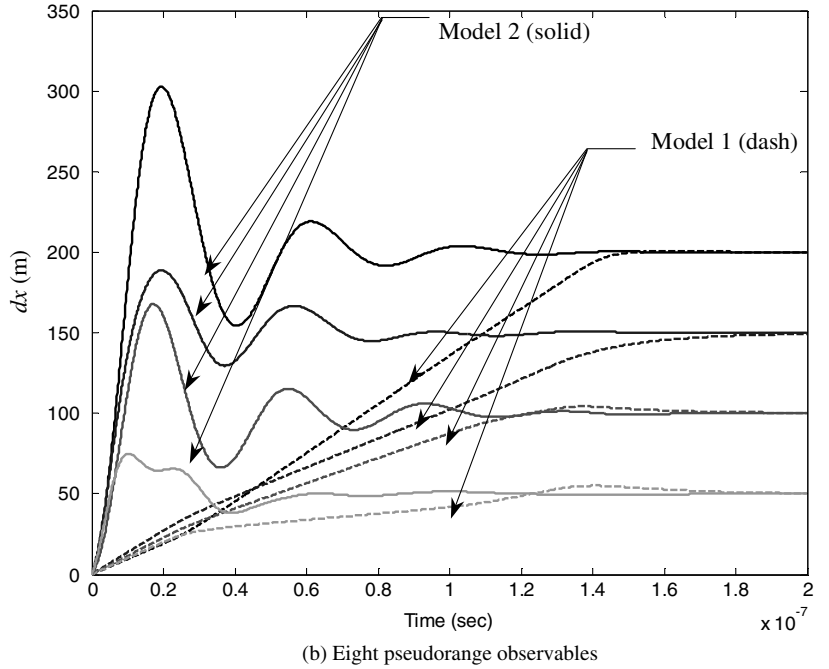
$$\Delta\mathbf{x}_{LS} = [\,-2977571{\cdot}476 \quad -5635278{\cdot}159 \quad 4304234{\cdot}506 \quad 1625239{\cdot}802\,]^{\mathrm{T}}\ \mathrm{m}$$

where the subscript '*LS*' stands for the least-squares approach. The steady-state solutions are reached in approximately 0·08 sec for the Model 1 circuit, and

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press



(a) Four pseudorange observables



(b) Eight pseudorange observables

Figure 3. Convergence of the state trajectories at the 1st iteration for the cases of (a) four, and (b) eight pseudorange observables, when the deviation between initial nominal solution and final solution is small.

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

approximately 1·5 μs for Model 2 circuit. The result is given in Figure 2 (a). For the example of eight observables, if the initial nominal state vector is taken to be at the centre of the Earth, the initial errors by the least-squares is found to be:

$$\Delta \mathbf{x}_{LS} = [1117874{\cdot}114 \quad -6665513{\cdot}833 \quad 3225078{\cdot}117 \quad 1156492{\cdot}923]^{\mathrm{T}}$$

The steady-state solutions are reached in approximately 0·02 sec based on Model 1 circuit, and approximately 0·3 μs based on the Model 2 circuit. As for the steady-state solutions, Model 2 has very good agreement to the least-squares approach, while the steady-state solutions by Model 1 circuit are found to be:

$$\Delta \mathbf{x}_{NN1} = [1110395{\cdot}845 \quad -6675284{\cdot}681 \quad 3222725{\cdot}666 \quad 1223598{\cdot}499]^{\mathrm{T}}$$

It is seen that the results by Model 1 and least-squares approach have a significantly large difference (about $[-7478 \quad -9771 \quad -2352 \quad 67106]^{\mathrm{T}}$ m). The result is shown in Figure 2 (b). Next, consider the case that the difference between initial nominal solution and final solution is small, e.g., $\Delta \mathbf{x} = [200 \quad 150 \quad 100 \quad 50]^{\mathrm{T}}$. For both NN circuit structures, the steady-state solutions are reached in approximately 2 μs and 0·2 μs for the case of four and eight observables, respectively. The convergence histories are shown in Figure 3.

From the above results, it is seen that Model 2 circuit structure provides a very good steady-state accuracy and convergence speed even when the deviation between the initial nominal solution and the final solution is very large, while Model 1 circuit structure requires a substantially longer time to converge and results in worse solution accuracy. The experiment indicates that the Model 1 circuit is not practical on GPS navigation application (unsatisfactory steady-state accuracy and slower convergence speed), while Model 2 presents very good potential in terms of both convergence speed and solution accuracy.

4.2. *Dynamic positioning experiments*. Two types of NN circuit structures will be utilized for the dynamic positioning experiments to evaluate the steady-state error performances and to compare to the conventional least-squares method. The commercial software Satellite Navigation toolbox by GPSoft LLC was employed. A 24-satellite constellation was simulated and the error sources corrupting GPS measurements include ionospheric delay, tropospheric delay, receiver noise and multipath.

The scenario for simulation is as follows. A circle trajectory with a radius of 5 km, as shown in Figure 4, is designed for simulation. The origin of the circle was located at the position of 25·1492°N 121·7775°E at an altitude of 100 m. This is equivalent to $[-3042329{\cdot}2 \quad 4911080{\cdot}2 \quad 2694074{\cdot}3]^{T}$ m in WGS-84 ECEF coordinates. The location of the origin is defined as the (0, 0, 0) m location in the local tangent ENU frame. The experiment was conducted on a simulated vehicle trajectory originating from the (5000, 0, 0) m location. The user was simulated to move in the counterclockwise direction, at 26·2 km/hr constant speed. The vehicle completed 10 circle movements within the 12-hour simulation period. Navigation solutions were computed every one minute, consequently, 721 sets of solutions were recorded. The numbers of GPS satellites visible during the simulation period were five to eight to check the feasibility and applicability on GPS solutions.

In determining the navigational solutions, five iteration steps were taken at each time epoch; while at each iteration step, the steady-state solutions (i.e., circuit outputs)

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

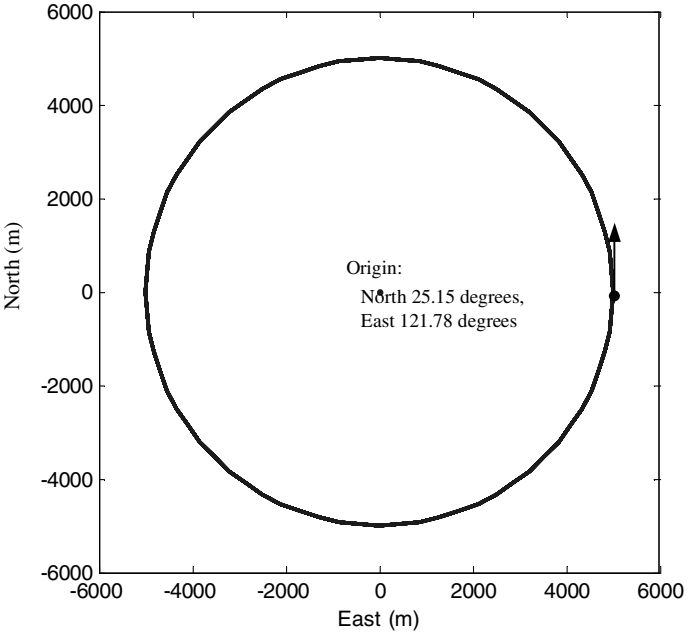


Figure 4. Reference East-North trajectory for the dynamic positioning experiments.

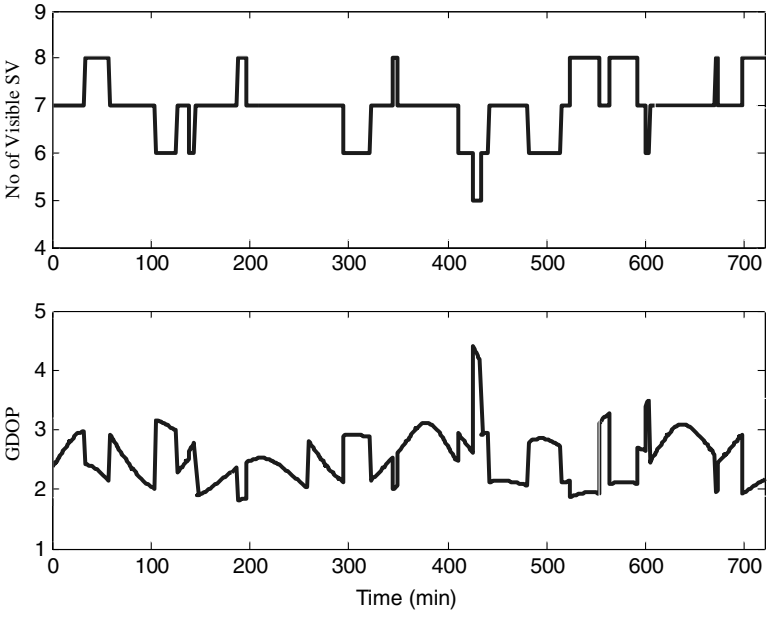
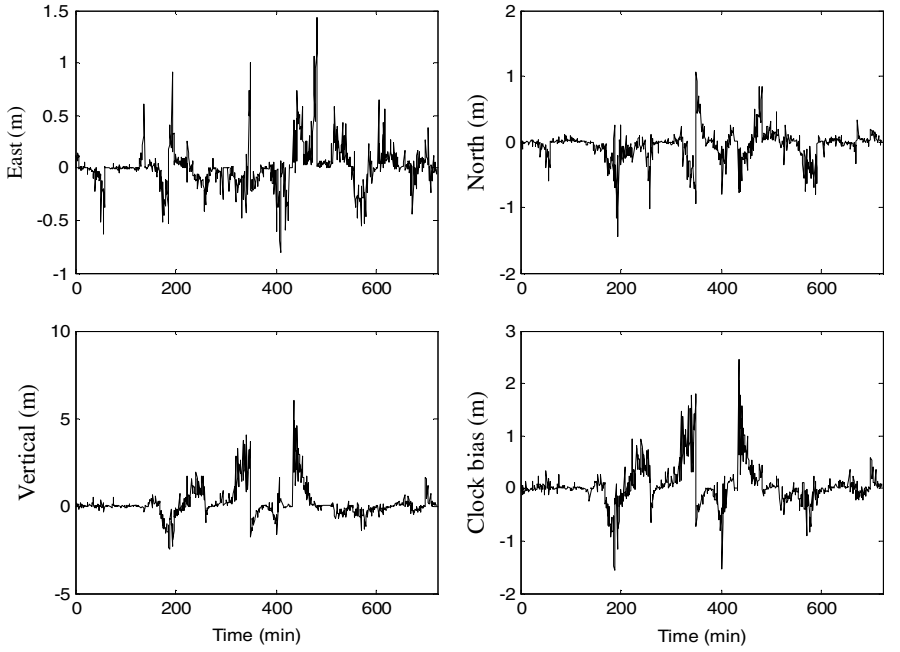


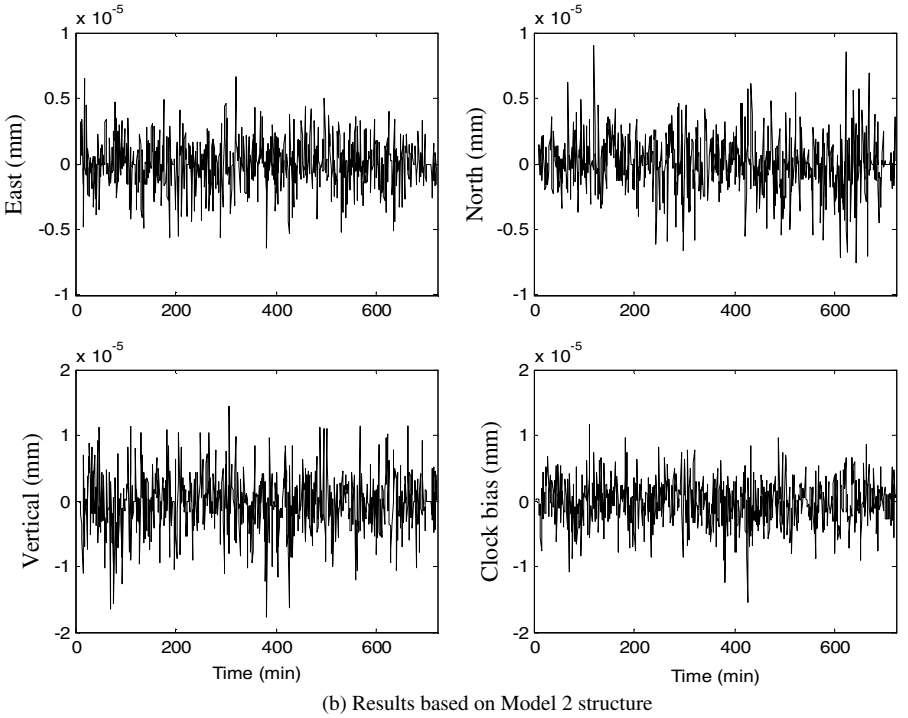
Figure 5. Number of GPS satellites visible and GDOP in the 12-hour simulation period.

were taken after 1 μs of processing time (although they can be obtained actually much less than 1 μs). The integration intervals have been tested using different values, in which both $dt = 1e^{-9}$ and $dt = 1e^{-10}$ provide very similar values. Figure 5 shows

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press



(a) Results based on Model 1 structure



(b) Results based on Model 2 structure

Figure 6. Differences on navigation solutions between the present approach and the conventional method.

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

Table 3. Statistic of the difference between the present approach and the conventional method (in metres).

		East	North	Vertical	Clock bias
Mean	**Model 1**	0·00962	−0·05603	0·15038	0·05803
	Model 2	$3{\cdot}4e^{-12}$	$2{\cdot}2e^{-11}$	$-3{\cdot}2e^{-10}$	$-2{\cdot}2e^{-10}$
Variance	**Model 1**	0·04446	0·05449	0·72165	0·14753
	Model 2	$3{\cdot}5e^{-18}$	$4{\cdot}5e^{-18}$	$2{\cdot}0e^{-17}$	$1{\cdot}2e^{-17}$

the number of GPS satellites visible and GDOP within the 12-hour simulation period. The difference in navigational solutions between the present approach and the conventional method is provided in Figure 6. When the Model 1 circuit structure is employed, the results show that there are large differences appearing at several epochs compared to the conventional approach (about 1 m in the horizontal components, 3–5 m in the vertical component, and 2–3 m for clock bias). On the contrary, solutions based on the Model 2 circuit structure are virtually exactly same as those based on the conventional method, which have differences that are all far less than mm level. Table 3 gives the statistic of the difference between the present approach and the conventional method. Although very small, the statistic for Model 2 is still provided in the table for comparison to Model 1. It is concluded that there is very good agreement for the results using the Model 2 NN circuit and the results using conventional method in which pseudo-inverse matrix calculation by the digital computer is involved.

5. CONCLUSIONS. The ordinary differential equation formulation schemes and corresponding circuits of analog neural network least-squares processors have been employed for GPS navigation processing. It is seen that the electronic implementation of a neural network-like processor employed in this paper has the capability to determine the navigation solution virtually instantaneously. This would enable the use of high-integrity navigation solution without the delay required for many pseudo-inverse matrix calculations by digital computer. Examples on both single epoch and dynamic positioning have been presented and very good agreement in navigation accuracy is obtained for one of the present approaches as compared to the conventional method. The results given in this paper clearly demonstrate the feasibility of analogue neural network least-squares processor for GPS navigation solution computation.

ACKNOWLEDGEMENTS

This work has been supported in part by the National Science Council of the Republic of China under grant no. NSC 92-2212-E-019-004.

REFERENCES

Axelrad P. and Brown R. G. (1996). GPS navigation algorithms, in Parkinson, B. W., Spilker, J. J., Axelrad, P., and Enga, P. (ed), Global Positioning System: Theory and Applications. Volume I, AIAA, Washington DC, Chap. 9.

Bancroft, S. (1985). An algebraic solution of the GPS equations. *IEEE Trans. Aero. Electro Systems*, AES-**21**(**7**), 56.

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press

Chansarkar, M. (2000). Neural networks in GPS navigation. *GPS Solutions*, **4**(**2**), 14.
Cichocki A. and Unbehauen R. (1992). Neural networks for solving systems of linear equations and related problems. *IEEE Trans. Circuits and Systems-1: Fundamental Theory and Applications* **39**(**2**), 124.
Farrell, J. and Barth, M. (1999). The Global Positioning System and Inertial Navigation. McGraw-Hill.
Krause, L. O. (1987) A direct solution to GPS-type navigation equations, *IEEE Trans. Aero. Electro Systems*. AES-**23**(**2**), 225.
Lundberg, J. B. (2001). Alternative algorithms for the GPS static positioning solution, *Applied mathematics and computation*, **119**, 21.
Simon, D. and El-Sherief, H. (1995). Navigation satellite selection using neural networks, *Neurocomputing*. **7**, 247.
Yalagadda, R., Ali, I., Al-Dhahir, N. and Hershey, J. (2000). GPS GDOP metric, *IEE Proc.-Radar, Sonar Navig*. **147**(**5**), 259.

https://doi.org/10.1017/S0373463304003017 Published online by Cambridge University Press