Outlier Resistance Estimator for GPS Positioning – the Neural Network Approach

Dah-Jing Jwo
(National Taiwan Ocean University)
(Email: djjwo@mail.ntou.edu.tw)

One of the popular techniques for solving GPS pseudorange nonlinear quadratic equations to obtain the receiver’s position and clock bias involves linearizing the equations and solving them by the least-squares (LS) scheme, which is based on the $L_2$ minimization criterion. When outliers are a problem, LS estimation scheme may not be the best approach because the LS estimator minimizes the mean squared error of the observations. This paper discusses the implementation aspects of $L_1$ and $L_{\infty}$ criteria, and their outlier resistance performance for GPS positioning. Three ordinary differential equation formulation schemes and corresponding circuits of neuron-like analogue $L_1$ (least-absolute), $L_{\infty}$ (minimax), and $L_2$ (least-squares) processors will be employed for GPS navigation processing. The circuits of simple neuron-like analogue processors are employed essentially for solving systems of linear equations. Experiments on single epoch and thereafter kinematic positioning will be conducted by computer simulation for investigating the outlier resistance performance for the least-absolute and minimax schemes as compared to the one provided by the least-squares scheme.

KEY WORDS

1. INTRODUCTION. The most popular techniques for solving the linearized GPS pseudorange equations for the receiver’s position and clock bias is the $L_2$ (least-squares, LS) scheme (Axelrad and Brown, 1996; Bancroft, 1985; Krause, 1987). When outliers occur, LS estimation scheme may not be the good choice because the LS estimator minimizes the mean squared error of the observations. In order to perform robust positioning estimation, other criteria rather than LS may be employed. The common alternatives might be the $L_1$ (least-absolute, LA) or the $L_{\infty}$ (minimax, MM) estimators.

The issues of $L_1$ and $L_{\infty}$ criteria in GPS positioning are far from practical realization if compared with the least-squares method based on the $L_2$ criterion. The traditional LS estimator is rather sensitive to the large noises, resulting in large estimation errors. The reason for that is its Gaussian error assumption. The LA and MM estimators are known as two of the robust estimators. The LA estimator is also known to be able to produce approximately the maximum-likelihood
estimation. In this case the maximum-likelihood estimator is obtained by minimizing the mean absolute deviation, rather than the mean square deviation; accordingly it can perform robust and effective estimation. Even if the desired signals are corrupted by the unknown errors, it tends to be impervious to the unexpected large errors.

For real-time applications, the solution is required within a time of the order of a hundred nanoseconds. Although there are very efficient numerical algorithms to solve least absolute value and minimax optimization problems, they are not able to solve these problems in real time. In such cases, a digital computer often cannot comply with the desired computation time, or its use is too expensive. One possible and very promising very realistic alternative to solve such optimization problems in real time is to apply neural networks (NN). The application of NN approach for navigation solution processing has not yet been widely explored in the GPS community. 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 need 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. Jwo (2005) demonstrated the feasibility of an analogue neural network least-squares processor for GPS navigation solution computation.

In this paper, further extension of the NN-based GPS navigation applications will be conducted. The circuits of simple analogue neural network least-absolute and minimax processors will be employed and compared to the least-squares processors, in solving the systems of linear equations. The related ordinary differential equation formulation schemes and corresponding circuit architectures will be discussed. Experiments via computer simulation will be conducted for single epoch solutions and then for kinematic positioning solutions. The properties and performance of the navigation solutions based on the least-absolute and minimax methods will be assessed and compared to those provided by the least-squares method.

This paper is organized as follows. In Section 2, preliminary background on linearization of GPS pseudorange equations is briefly reviewed. The neural network-like architectures for solving systems of linear equations based on the least-absolute, minimax, and least-squares criteria are introduced in Section 3. In Section 4, simulation examples are presented; performance assessment by the three neural network-like processors for GPS navigation will be provided. Section 5 gives the conclusions.

2. LINEARIZATION OF GPS PSEUDORANGE EQUATIONS.

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 $r_i$ is defined for the $i$th satellite by

$$r_i = \|\mathbf{s}_i - \mathbf{u}\| + ct_b + v_{\rho_i}$$  \hspace{1cm} (1)

where $c$ is the speed of light and $t_b$ is the receiver clock offset from system time, and $v_{\rho_i}$ is the pseudorange measurement noise. Consider the user position in three
dimensions, denoted by \((x_i, y_i, z_i)\), the GPS pseudorange measurements made to the \(n\) satellites can then be written as
\[
\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2 + c t_b + v_{\rho_i}}, \quad i = 1, \ldots, n
\]  
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 \(\rho_i\) at \((\hat{x}_n, \hat{y}_n, \hat{z}_n)\) gives
\[
\Delta \rho_i = \rho_i - \hat{\rho}_i = e_{1i} \Delta x_u + e_{2i} \Delta y_u + e_{3i} \Delta z_u + c t_b + v_{\rho_i}, \quad i = 1, \ldots, n
\]  
where
\[
e_{1i} = \frac{\hat{x}_n - x_i}{\hat{r}_i}, \quad e_{2i} = \frac{\hat{y}_n - y_i}{\hat{r}_i}, \quad e_{3i} = \frac{\hat{z}_n - z_i}{\hat{r}_i}
\]  
\(\hat{r}_i = \sqrt{(\hat{x}_n - x_i)^2 + (\hat{y}_n - y_i)^2 + (\hat{z}_n - z_i)^2}\)

The vector \((e_{1i}, e_{2i}, e_{3i}) \equiv \mathbf{e}_i, \quad i = 1, \ldots, n\), denotes the line-of-sight vector from the user to the satellites. Equation 3 can be written in a matrix formulation
\[
\Delta \mathbf{\rho} = \mathbf{H} \Delta \mathbf{x} + \mathbf{v}
\]  
which can be represented as
\[
\Delta \mathbf{z} = \mathbf{H} \Delta \mathbf{x} + \mathbf{v}
\]  
where \(\mathbf{v}\) are assumed to be zero-mean. The dimension of matrix \(\mathbf{H}\) is \(n \times 4\) with \(n \geq 4\), \(\mathbf{H}\) is usually referred to as the ‘geometry matrix’ or ‘visibility matrix’.

3. NEURAL NETWORKS FOR SOLVING SYSTEMS OF LINEAR EQUATIONS. Solution of a set of linear equations is desired in many engineering applications. The analogue circuit solution is particularly 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.

Assume that a set of linear algebraic equations is to be solved
\[
\mathbf{A} \mathbf{x} \approx \mathbf{b}
\]  
\[\text{NO. 1 OUTLIER RESISTANCE ESTIMATOR FOR GPS POSITIONING 131} \]
which can be written in scalar form as

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

Here, \( A = [a_{ij}] \) is the \( m \times n \) real coefficient matrix with known elements, \( x \) is the \( n \)-dimensional unknown vector, and \( b \) is the \( m \)-dimensional observation vector (it should be noted that \( m \) can be less than, equal to or greater than \( n \)).

### 3.1. Least-absolute and minimax solutions

The \( L_p \)-normed minimization can be described as the minimization of the following norm:

$$E_p(x) = \frac{1}{p} \sum_{i=1}^{m} |r_i|^p$$

with \( 1 \leq p < \infty \) where the residuals

$$r_i(x) = \sum_{j=1}^{n} a_{ij} x_j - b_i \quad (i = 1, 2, \ldots, m)$$

$$\frac{dx_j}{dt} = -\mu_j \frac{\partial E_p(x)}{\partial x_j}$$

where \( \mu_j = 1/\tau_j > 0 \) are referred to as learning rates, and \( \tau_j \) are the time constants of the integrators. By applying the chain rule to the error function \( E_p(x) \), Equation 9
can be mapped into the system of differential equations
\[
\frac{dx_j}{dt} = -\mu_j \sum_{i=1}^{m} a_{ij}g[r_i(x)], \quad (j = 1, 2, \ldots, n)
\] (10)

which can be directly implemented by a neural network depicted in Figure 1. Two special cases are \(p = 1\) (least-absolute values or \(L_1\)-norm problem) and \(p = \infty\) (minimax or \(L_\infty\)-Chebyshev norm problem).

The standard least absolute value problem based on the minimization of the energy function
\[
E_1(x) = \sum_{i=1}^{m} |r_i(x)|
\] (11)

can be formulated as
\[
\frac{dx_j}{dt} = -\mu_j \sum_{i=1}^{m} a_{ij} \text{sign}[r_i(x)], \quad (j = 1, 2, \ldots, n)
\] (12)

where \(\text{sign}[\bullet]\) is the signum (hard limiter) function defined as
\[
\text{sign}[r_i(x)] = \begin{cases} 
1 & r_i(x) > 0 \\
-1 & r_i(x) < 0
\end{cases}
\]

and \(\text{sign}[0]\) is to be interpreted as zero.

The minimax (\(L_\infty\)-norm) problem is described as the minimization of the following norm:
\[
\min_{x \in \mathbb{R}^n} E_\infty(x)
\] (13a)

where
\[
E_\infty(x) = \max_{1 \leq i \leq m} \{|r_i(x)|\}
\]

Equation 13a can be transformed into an equivalent one:
\[
\min \varepsilon
\] (13b)

subject to the constraints
\[
|r_i(x)| \leq \varepsilon, \varepsilon \geq 0.
\]

Thus the problem can be viewed as finding the smallest non-negative value of
\[
\varepsilon^* \geq E_\infty(x^*) \geq 0
\]
such that all absolute values of the residuals are not greater than \(\varepsilon^*\) (where \(x^*\) is a vector of the optimal values of the parameters).

The differential equations for \(L_\infty\)-normed minimization can be described as:
\[
\frac{de}{dt} = -\mu_0 \left( \frac{v}{k} - \sum_{i=1}^{m} S_i \right)
\] (14a)
\[
\frac{dx_j}{dt} = -\mu_j \sum_{i=1}^{m} a_{ij} S_i r_i(x)
\] (14b)
where
\[ \mu_j > 0 \quad \text{and} \quad S_i = \begin{cases} 0 & r_i^2(x) \leq \varepsilon \\ 1 & \text{otherwise} \end{cases} \]

More detailed discussion on the NN for solving a system of linear equations using the least-absolute value and minimax approach can be referred to Cichocki and Unbehauen (1992b).

3.2. Least-squares solution. The least-squares solution is based on the principle of \( L_2 \)-normed minimization. The ordinary LS problem is one in which we minimize the energy function:
\[
E_2(x) = \frac{1}{2} \sum_{i=1}^{m} r_i^2(x) = \frac{1}{2} \|Ax - b\|_2^2
\]

Using a general gradient approach for minimization of a function the problem formulated by Equation 15 can be mapped to a set of differential equations:
\[
\frac{dx}{dt} = -\mu(t) \nabla E_2(x) 
\]

\[
\nabla E_2(x) = A^T (Ax - b) \quad x(0) = x^0
\]

where \( x = [x_1 \ x_2 \ldots \ x_n]^T \) and \( \mu(t) = [\mu_{ij}(t)] \) is an \( n \times n \) positive-definite matrix that is often diagonal, and \( \nabla E_2(x) \) is the gradient of the energy function, \( E_2(x) \). Generally, the entries of the matrix \( \mu(t) \) depend on the time and the vector \( x \). Expressing Equation 16 in scalar form, we have
\[
\frac{dx_j}{dt} = - \sum_{p=1}^{n} \mu_{jp} \left[ \sum_{i=1}^{m} a_{ip} \left( \sum_{k=1}^{n} a_{jk} x_k - b_i \right) \right] 
\]

with \( x_j(0) = x_{j0} ^0, m \geq n, \) for \( j = 1, 2, \ldots, n \). This is an initial value problem in which we can compute a trajectory \( x(t) \) 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
\[
\frac{dx_j}{dt} = - \sum_{p=1}^{n} \mu_{jp} \left( \sum_{i=1}^{m} a_{ip} g_i \left( \sum_{k=1}^{n} a_{jk} x_k - b_i \right) \right) 
\]

where \( g_i[\bullet] \) 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:
\[
\varepsilon_i(x) = g_i \left( \sum_{k=1}^{n} a_{jk} x_k - b_i \right), \quad i = 1, 2, \ldots, m
\]

\[
\frac{\partial \varepsilon(x)}{\partial x_p} = \sum_{i=1}^{m} a_{ip} \varepsilon_i(x), \quad p = 1, 2, \ldots, n
\]
\[
\frac{dx_j}{dt} = \sum_{p=1}^{n} \mu_{jp} \frac{\partial e(x)}{\partial x_p}, x_j(0) = x_j^{(0)}, j = 1, 2, \ldots, n
\] (19c)

which can be directly implemented by a neural network. The architecture of the network resembles the general Tank-Hopfield model.

The above mentioned circuit structures (or NN architectures) can be considerably simplified for some important special cases to get the advantage of relatively few interconnections. Cichocki and Unbehauen (1992a) 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

\[
\frac{d\lambda_i}{dt} = \rho_i \left[ r_i(x) - \alpha \lambda_i \right] \quad (20b)
\]

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

\[
\frac{d\lambda}{dt} = \rho \left[ (Ax - b) - \alpha \lambda \right] \quad (21b)
\]
which can be directly implemented by a neural network depicted in Figure 2. Further discussion on the NN for solving a system of linear equations using the least-squares approach can be referred to Cichocki and Unbehauen (1992a).

4. STUDY EXAMPLES. Computer simulation was employed for evaluating the outlier resistance performance for the three types of neural network circuit structures, i.e. LA, MM, and LS processors as represented by Equations 12, 14 and 20, respectively, for GPS navigation applications. Simulation was conducted on a notebook computer powered by 1.7GHz Intel Pentium 4 mobile CPU. The computer code was written using the Matlab® 6.5 version software. Due to space limits, only some illustrated examples will be presented. Single epoch solutions involving eight measurements will be examined, followed by the kinematic positioning solutions involving five to eight satellites. Navigation solutions based on the LA and MM approach will be compared to those based on the LS method.

4.1. Evaluation with single epoch solutions. Experiments on single epoch positioning have been performed to verify the feasibility of the present methods for GPS navigation processing. After that, comparison of the LA, MM results to the LS results is provided. The circuit parameters for the three circuit structures used in the following examples are:

1. Least-absolute:
\[
\mu_j = \frac{1}{\tau} = 10^9 \text{ and } dt = 5e - 12
\]

2. Minimax:
\[
\mu_o = \mu_j = \frac{1}{\tau} = 10^7, \quad \varepsilon = 0.1, \quad \frac{v}{k} = 0.5 \text{ and, } dt = 1e - 10
\]

3. Least-squares:
\[
\mu_o = \rho_o = \frac{1}{\tau} = 10^8, \quad a = 0.5, \quad k_j = 0.1 \text{ and } dt = 1e - 10
\]

The ECEF coordinates of GPS satellites and the pseudorange observables and the actual receiver position are listed in Table 1. Experience shows that it normally takes

| Geocentric coordinates of 8 GPS satellites and the pseudorange observables |
|-----------------|-----------------|-----------------|-----------------|
| GPS-1           | -14929765.949   | 19301747.549    | -1048863.006    | 22851390.227 |
| GPS-2           | 8169490.524     | 23684561.630    | 8818599.421     | 22708109.946 |
| GPS-3           | 948773.075      | 26483255.419    | -1789900.241    | 22391814.319 |
| GPS-4           | 9416737.486     | 11998085.003    | 21744832.540    | 23840845.394 |
| GPS-5           | -23189203.453   | 6593709.302     | -11146774.048   | 24500978.105 |
| GPS-6           | -14942489.076   | 8597756.911     | 20205626.125    | 21490906.431 |
| GPS-7           | -23389686.536   | -1338041.738    | 12513949.770    | 23441342.083 |
| GPS-8           | -10878980.552   | 16786546.988    | 17473590.163    | 20515185.115 |

Actual receiver position: \([-3042347.091, 4911109.205, 2694090.285]\)^T m
Figure 3. Convergence of the state trajectories at the 1\textsuperscript{st} iteration when the deviation between initial nominal solution and final solution is large.
Figure 4. Convergence of the state trajectories at the 1st iteration when the deviation between initial nominal solution and final solution is small.
no more than five iterations to obtain the navigation solution, which is also true in the present work.

For comparison of the performance based on the three NN models, convergence histories of the state trajectories at the 1st iteration step are presented. Computer simulated state trajectories using the three NN circuit structures are provided in Figures 3 and 4, respectively, showing the convergence behaviour of the state trajectories at the 1st iteration step when the initial error vector (defined as the deviation between initial nominal solution (i.e. initial guess of the solution) and final solution at each iteration step, denoted as $Dx$) is large and small, respectively. The reason for selecting the result at the 1st iteration step is due to the fact that it takes the longest time (due to the largest $Dx$) for convergence among all iteration steps at a certain epoch, so that the three NN models would reflect their distinguishing characteristics. The initial error vector has significant influence on the time to convergence, which is true especially for the least-absolute approach. Ideally, when the positioning solution converges, the vector $Dx$ at the last iteration step (i.e. the 5th iteration step for the study) should approach to a zero vector (with size $4 \times 1$).

### Table 2. Computed ECEF user’s position and clock bias as a function of iteration number (all cases are with the initial nominal state vector at the centre of the Earth).

#### (a) Least-squares solutions (position errors in ENU frame: [0.875 0.261 26.896]$^T$ m)

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>cdt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000000-000</td>
<td>0000000-000</td>
<td>0000000-000</td>
<td>0000000-000</td>
</tr>
<tr>
<td>1</td>
<td>-3614027-544</td>
<td>5782841-469</td>
<td>3189512-128</td>
<td>1238656-986</td>
</tr>
<tr>
<td>2</td>
<td>-3057562-977</td>
<td>4932036-888</td>
<td>2707031-528</td>
<td>34548-500</td>
</tr>
<tr>
<td>3</td>
<td>-3042371-251</td>
<td>4911142-818</td>
<td>2694110-995</td>
<td>53-313</td>
</tr>
<tr>
<td>4</td>
<td>-3042360-598</td>
<td>4911129-346</td>
<td>2694101-952</td>
<td>30-973</td>
</tr>
<tr>
<td>5</td>
<td>-3042360-598</td>
<td>4911129-346</td>
<td>2694101-952</td>
<td>30-973</td>
</tr>
</tbody>
</table>

#### (b) Least-absolute solutions (position errors in ENU frame: [2.831 -0.953 26.140]$^T$ m)

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>cdt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000000-000</td>
<td>0000000-000</td>
<td>0000000-000</td>
<td>0000000-000</td>
</tr>
<tr>
<td>1</td>
<td>-356309-453</td>
<td>5732298-121</td>
<td>3174809-945</td>
<td>120556-723</td>
</tr>
<tr>
<td>2</td>
<td>-305292-017</td>
<td>4926482-243</td>
<td>2704592-994</td>
<td>29237-347</td>
</tr>
<tr>
<td>3</td>
<td>-304236-620</td>
<td>4911134-390</td>
<td>2694105-464</td>
<td>41-197</td>
</tr>
<tr>
<td>4</td>
<td>-304236-1770</td>
<td>4911128-383</td>
<td>2694100-522</td>
<td>29-557</td>
</tr>
<tr>
<td>5</td>
<td>-304236-1774</td>
<td>4911128-379</td>
<td>2694100-521</td>
<td>29-557</td>
</tr>
</tbody>
</table>

#### (c) Minimax solutions (position errors in ENU frame: [0.875 0.261 26.896]$^T$ m)

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>cdt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000000-000</td>
<td>0000000-000</td>
<td>0000000-000</td>
<td>0000000-000</td>
</tr>
<tr>
<td>1</td>
<td>-3607505-941</td>
<td>5773304-450</td>
<td>3185406-421</td>
<td>1230277-398</td>
</tr>
<tr>
<td>2</td>
<td>-3057257-858</td>
<td>4931616-037</td>
<td>2706796-341</td>
<td>33838-837</td>
</tr>
<tr>
<td>3</td>
<td>-304237-0879</td>
<td>4911142-352</td>
<td>2694110-679</td>
<td>52-484</td>
</tr>
<tr>
<td>4</td>
<td>-304236-598</td>
<td>4911129-346</td>
<td>2694101-952</td>
<td>30-973</td>
</tr>
<tr>
<td>5</td>
<td>-304236-598</td>
<td>4911129-346</td>
<td>2694101-952</td>
<td>30-973</td>
</tr>
</tbody>
</table>
For this eight observables example, when $\Delta x$ is very large (e.g. take the initial nominal state vector to be at the centre of the Earth (the origin of the ECEF coordinate system) with zero clock bias: $x = [0 \ 0 \ 0]^T$ m) results in the large initial errors (which are also the steady-state solution to be solved for) as follows:

$$
\Delta x_{\text{LS}} = \begin{bmatrix}
-3614027.544 \\
782841.469 \\
3189512.128 \\
1238656.986
\end{bmatrix}^T \text{m}
$$

$$
\Delta x_{\text{LA}} = \begin{bmatrix}
-3563094.537 \\
5732298.121 \\
3174809.945 \\
1205565.723
\end{bmatrix}^T \text{m}
$$

$$
\Delta x_{\text{MM}} = \begin{bmatrix}
-3607505.941 \\
5773304.450 \\
3185406.421 \\
1230277.398
\end{bmatrix}^T \text{m}
$$

where the subscript ‘LS’, ‘LA’, and ‘MM’ stand for the least-squares, least-absolute, and minimax approaches, respectively. The steady-state solutions are reached in
approximately (1) 0.3 μs for LS circuit; (2) 0.01 sec for LA circuit; (3) and 5 μs for minimax circuit, as shown in Figure 3. Next, consider the case that when Δx is small, e.g., Δx=[200 150 100 50]T. The steady-state solutions can be reached in approximately (1) 0.3 μs for LA and LS circuits, (2) 3 μs for MM circuit, respectively, as shown in Figure 4.

The computed user’s position and clock bias (at the 5th iteration steps) by the three methods at each iteration step are listed in Table 2, where the initial nominal state vectors for all cases are set at the centre of the Earth. The experiment result indicates that LS and the MM approach present very close positioning results, while LS approach has better convergence speed than the MM approach does. When there is no SV failure, the LS circuit structure presents very good potential in terms of both steady-state accuracy and convergence speed, even when the initial error vector is very large, while LA approach requires longer time to convergence and results in worse solution accuracy when the initial error vector is large. When comparing Figure 3(b) with Figure 4(a), it can be found that the convergence speed for the LA method is significantly influenced by the value of initial error vector.

Further experiments have been conducted when the bias error is injected onto one of the pseudorange observables. Using the same set of data given in Table 1, an extra bias error of 100 metres was added to one of the eight pseudorange observables (one by one each time). Figure 5 provides the position error norms for the three approaches and the corresponding numerical data is listed in Table 3. It is seen that the LA approach has the best outlier resistance performance among the three approaches, while the outlier resistance capability between the LS and MM approaches are very close to each other.
4.2. Kinematic positioning experiments. The three types of NN circuit structures were utilized for kinematic positioning experiments for evaluating the outlier resistance performances. The commercial software Satellite Navigation toolbox by GPSsoft LLC was employed. A 24-satellite constellation was simulated and the error sources

![Comparison of position error for the case of no failure.](image)

Figure 7. Comparison of position error for the case of no failure.
corrupting GPS measurements include ionospheric delay, tropospheric delay, receiver noise and multipath. For determining navigational solutions, five iteration steps were taken at each time epoch.

The simulation scenario is designed as follows. A circle trajectory with a radius of 5 km is designed for simulation. The origin of the circle was located at the position of North 25.1492° and East 121.7775° at the sea level. This is equivalent to \([-304229942 \quad 491103225 \quad 269404779]\) 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, 100)m location. The user was simulated to move in the counter-clockwise direction, at 26.2 km/hr constant speed. The vehicle completes a ten circle movement within the 12-hour simulation period. Navigation solutions were computed every one minute, consequently there are 721 epochs recorded. The numbers of GPS satellites visible during the simulation period varied from five to eight so as to assure the feasibility and applicability of the GPS solutions. Figure 6 shows the number of GPS satellites visible and GDOP within the 12-hour simulation period.

Positioning and outlier resistance performance was evaluated for the case of normal condition (i.e. no failure occurred) followed by the case of one SV failure, where an extra bias error of 100 metres was injected onto one pseudorange every one hour. In our study, solutions based on the MM approach are very close to those based on the LS method, therefore, only LS results are provided. Figure 7 provides the comparison of position errors for the case of no failure; further information on the errors statistic is provided in Table 4.

For testing the outlier performance, there are a total of 12 epochs intentionally injected onto an extra 100 metre bias error. Figure 8 provides the comparison of position errors for the case of one SV failure. The result indicates that the LS (also true for MM) approach is very sensitive to the bias error at all the test epochs. Although the LA approach usually provided better outlier resistance capability, however, there were still 2 ~ 3 test epochs (e.g. at time 180, 480, and 600 min) that were virtually influenced in which one was influenced significantly (at time 600 min). In both Figures 7 and 8, three dimensional error plots are provided for LS and LA methods, followed by a two-dimensional E-N plot provided for comparison of these two methods. It is concluded that although under normal condition, LS and MM approaches provide better accuracy than the LA approach does, nevertheless, they are very sensitive to the error injection, and therefore, very weak in outlier resistance capability.

<table>
<thead>
<tr>
<th></th>
<th>East</th>
<th>North</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Least-squares</td>
<td>0.171</td>
<td>0.462</td>
<td>33.439</td>
</tr>
<tr>
<td>Least-absolute</td>
<td>0.196</td>
<td>0.284</td>
<td>33.440</td>
</tr>
<tr>
<td>Minimax</td>
<td>0.171</td>
<td>0.462</td>
<td>33.439</td>
</tr>
<tr>
<td>Std dev. Least-squares</td>
<td>1.915</td>
<td>2.528</td>
<td>6.167</td>
</tr>
<tr>
<td>Least-absolute</td>
<td>2.583</td>
<td>3.019</td>
<td>7.536</td>
</tr>
<tr>
<td>Minimax</td>
<td>1.915</td>
<td>2.528</td>
<td>6.167</td>
</tr>
</tbody>
</table>
5. CONCLUSIONS. The ordinary differential equation formulation schemes and corresponding circuits of analogue neural network least-absolute, minimax, and least-squares processors have been employed for GPS navigation processing.

Figure 8. Comparison of position errors for the case of one SV failure.
Numerical algorithms are not able to solve the least absolute value and minimax optimization problems in real time, therefore, the neural network approach has been used as a realistic alternative. The electronic implementation of a neural network-like processor employed in this paper shows the capability to determine the navigation solution virtually instantaneously. Examples on both single epoch and kinematic positioning have been presented. Of the three architectures, under normal conditions, LS and MM approaches provide better accuracy than does the LA approach. Unfortunately, they are very sensitive to the error injection, and therefore, very weak in outlier resistance capability. The LA approach demonstrates a much better outlier resistance capability compared with the LS and MM approaches.

ACKNOWLEDGEMENTS

This work has been supported in part by the National Science Council of the Republic of China under grant no. NSC 94-2212-E-019-003.

REFERENCES


