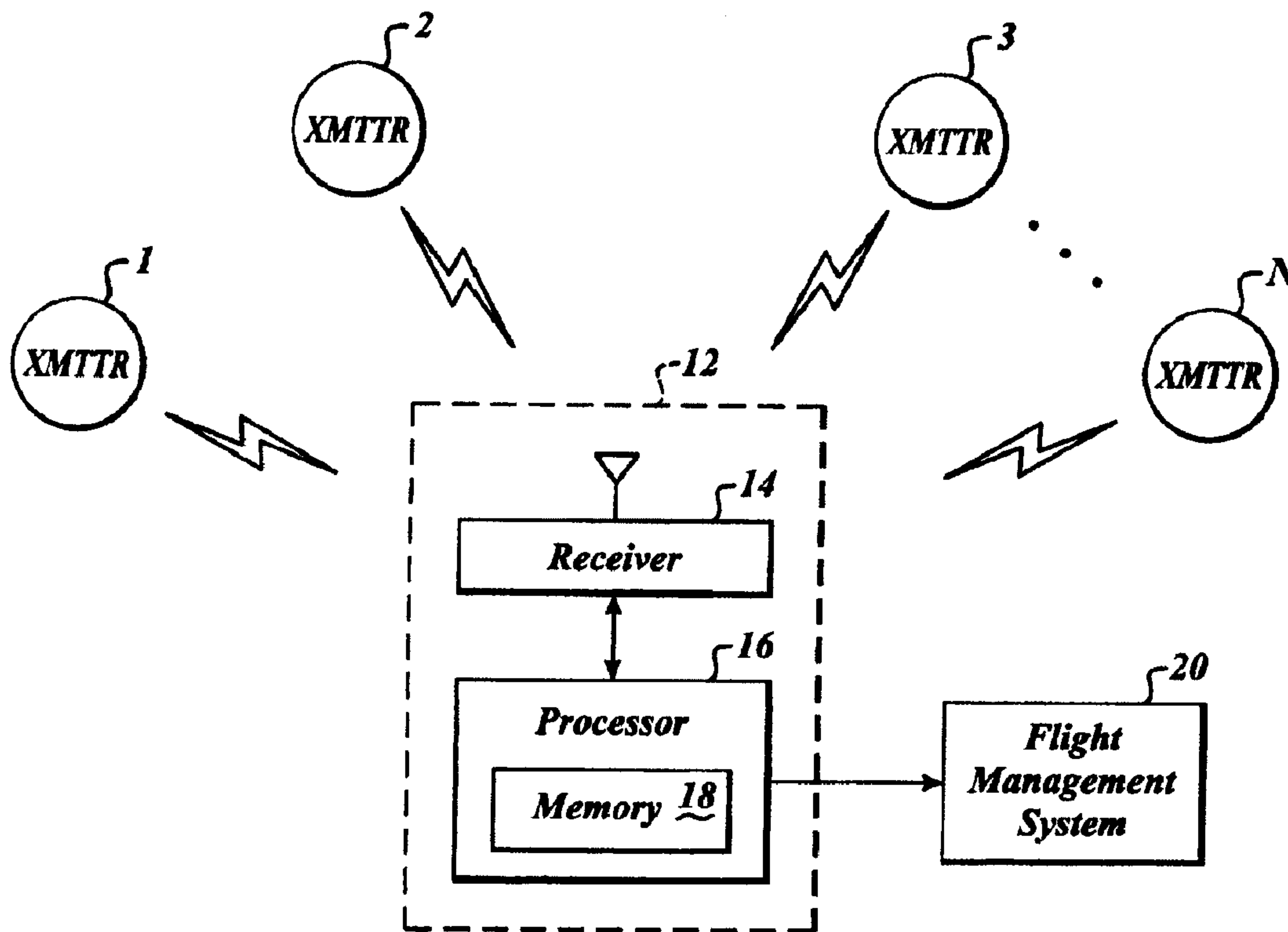




(22) **Date de dépôt/Filing Date:** 2008/12/03  
 (41) **Mise à la disp. pub./Open to Public Insp.:** 2009/06/07  
 (45) **Date de délivrance/Issue Date:** 2017/05/09  
 (30) **Priorités/Priorities:** 2007/12/07 (US60/012,303);  
 2008/04/14 (US12/102,527)

(51) **Cl.Int./Int.Cl. G01S 19/38** (2010.01),  
**G01C 21/00** (2006.01), **G01S 5/10** (2006.01)  
 (72) **Inventeur/Inventor:**  
 VANDERWERF, KEVIN, US  
 (73) **Propriétaire/Owner:**  
 HONEYWELL INTERNATIONAL INC., US  
 (74) **Agent:** GOWLING WLG (CANADA) LLP

(54) **Titre : SYSTEME DE NAVIGATION AVEC DETECTEUR D'IMPRECISIONS**  
 (54) **Title: NAVIGATION SYSTEM WITH APPARATUS FOR DETECTING ACCURACY FAILURES**



(57) **Abrégé/Abstract:**

A navigation system (12) for a vehicle having a receiver (14) operable to receive a plurality of signals from a plurality of transmitters includes a processor (16) and a memory device (18). The memory device (18) has stored thereon machine-readable instructions that, when executed by the processor (16), enable the processor (16) to determine a set of error estimates corresponding to pseudo-range measurements derived from the plurality of signals, determine an error covariance matrix for a main navigation solution using ionospheric-delay data, and, using a parity space technique, determine at least one protection level value based on the error covariance matrix.

**NAVIGATION SYSTEM WITH APPARATUS FOR DETECTING  
ACCURACY FAILURES****ABSTRACT OF THE DISCLOSURE**

A navigation system (12) for a vehicle having a receiver (14) operable to receive a plurality of signals from a plurality of transmitters includes a processor (16) and a memory device (18). The memory device (18) has stored thereon machine-readable instructions that, when executed by the processor (16), enable the processor (16) to determine a set of error estimates corresponding to pseudo-range measurements derived from the plurality of signals, determine an error covariance matrix for a main navigation solution using ionospheric-delay data, and, using a parity space technique, determine at least one protection level value based on the error covariance matrix.

**NAVIGATION SYSTEM WITH APPARATUS FOR DETECTING  
ACCURACY FAILURES**

**BACKGROUND OF THE INVENTION**

**[0002]** Conventional RAIM algorithms may be based on either a weighted or un-weighted least squares solution where the errors in each satellite's pseudo-range measurement are uncorrelated with the errors in the other satellites' pseudo-range measurements.

**[0003]** However, the ionospheric error (which can be the dominant error source) in each satellite's pseudo-range is, in fact, highly correlated with that of each of the other satellites. By ignoring this correlation, the computed Horizontal Protection Limit (HPL) which bounds the horizontal position error is much larger than necessary. As a result the availability of GPS to do a low Required Navigation Performance (RNP) approach suffers.

#### **SUMMARY OF THE INVENTION**

**[0004]** In an embodiment of the invention, a navigation system for a vehicle having a receiver operable to receive a plurality of signals from a plurality of transmitters includes a processor and a memory device. The memory device has stored thereon machine-readable instructions that, when executed by the processor, enable the processor to determine a set of error estimates corresponding to pseudo-range measurements derived from the plurality of signals, determine an error covariance matrix for a main navigation solution using ionospheric-delay data, and, using a parity space technique, determine at least one protection level value based on the error covariance matrix.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0005]** Preferred and alternative embodiments of the present invention are described in detail below with reference to the following drawings.

**[0006]** FIG. 1 shows a navigation system incorporating embodiments of the present invention; and

[0007] FIG. 2 shows a process according to an embodiment of the invention.

FIG. 3 depicts a noise scatter that would occur if there was a bias on the most difficult to detect satellite in the presence of the expected noise; and

FIG. 4 shows an embodiment to rotate the parity space such that the parity error due to a bias failure on satellite  $k$  is entirely along axis 1 of the parity space.

#### **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

[0008] FIG. 1 shows a radio navigation system incorporating the teachings of an embodiment of the present invention. The system includes several transmitters 1-N and user set 12. Transmitters 1-N, in the preferred embodiment, are a subset of the NAVSTAR GPS constellation of satellite transmitters, with each transmitter visible from the antenna of user set 12. Transmitters 1-N broadcast  $N$  respective signals indicating respective transmitter positions and signal transmission times to user set 12.

[0009] User set 12, mounted to an aircraft (not shown), includes receiver 14, processor 16, and a memory device, such as processor memory 18. Receiver 14, preferably NAVSTAR GPS compatible, receives the signals, extracts the position and time data, and provides pseudorange measurements to processor 16. From the pseudorange measurements, processor 16 derives a position solution for the user set 12. Although the satellites transmit their positions in World Geodetic System of 1984 (WGS-84) coordinates, a Cartesian earth-centered earth-fixed system, the preferred embodiment determines the position solution in a local reference frame  $L$ , which is level with the north-east coordinate plane and tangential to the Earth. This frame choice, however, is not critical, since it is well-understood how to transform coordinates from one frame to another.

**[0010]** Processor 16 also uses the pseudorange measurements to detect satellite transmitter failures and to determine a worst-case error, or protection limit, both of which it outputs with the position solution to flight management system 20. Flight management system 20 compares the protection limit to an alarm limit corresponding to a particular aircraft flight phase. For example, during a pre-landing flight phase, such as nonprecision approach, the alarm limit (or allowable radial error) is 0.3 nautical miles, but during a less-demanding oceanic flight phase, the alarm limit is 2-10 nautical miles. (For more details on these limits, see RTCA publication DO-208, which is incorporated herein by reference.) If the protection limit exceeds the alarm limit, the flight management system, or its equivalent, announces or signals an integrity failure to a navigational display in the cockpit of the aircraft. The processor also signals whether it has detected any satellite transmitter failures.

**[0011]** An embodiment of the invention models the correlation of the ionospheric errors between each pair of satellites as a function of the distance between their ionospheric pierce points. The closer the pierce points, the higher the correlation. The root-mean-square (RMS) uncertainty (or sigma) of each satellite's pseudo-range measurement is computed using the ionospheric variance model defined in DO-229D, Appendix J. Using the computed correlation coefficients and the sigma for each satellite, the ionospheric measurement error covariance matrix is formed. The remaining errors (satellite clock and ephemeris, tropospheric, multi-path and receiver noise) are assumed to be uncorrelated. Thus, the combined measurement error covariance matrix for these error sources is diagonal. These two matrices are

added to form the total measurement error covariance matrix. This matrix is then inverted to form the weighting matrix for the least squares solution. Fault detection and exclusion can then be performed and the various protection levels such as the horizontal protection level (HPL), vertical protection level (VPL), horizontal exclusion level (HEL), and vertical exclusion level (VEL) computed based on the methods of solution separation previously described in U. S. Patent Nos. 5,760,737 and 6,639,549.

**[0012]** FIG. 2 illustrates a process 200, according to an embodiment of the invention, that can be implemented in the radio navigation system illustrated in FIG. 1. The process 200 is illustrated as a set of operations or steps shown as discrete blocks. The process 200 may be implemented in any suitable hardware, software, firmware, or combination thereof. As such the process 200 may be implemented in computer-executable instructions that can be transferred from one electronic device to a second electronic device via a communications medium. The order in which the operations are described is not to be necessarily construed as a limitation.

**[0013]** Referring to FIG. 2, at a step 210, the processor 16 computes the sigma (error) values on pseudo-range and measurements.

**[0014]** At a step 220, the processor 16 determines the measurement matrix. The true vector of pseudo-range residuals  $\Delta\rho$  is related to the incremental position/time solution vector  $\Delta x$  (distance from the position linearization point) as follows:

$$\Delta \mathbf{p} = \mathbf{p} - \hat{\mathbf{p}} = \mathbf{H} \Delta \mathbf{x} \quad (1)$$

where  $\mathbf{H}$  is the measurement matrix and is given by:

$$\mathbf{H} = \begin{bmatrix} -LOS_{1x} & -LOS_{1y} & -LOS_{1z} & 1 \\ -LOS_{2x} & -LOS_{2y} & -LOS_{2z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -LOS_{Nx} & -LOS_{Ny} & -LOS_{Nz} & 1 \end{bmatrix}$$

$$\begin{bmatrix} LOS_{ix} \\ LOS_{iy} \\ LOS_{iz} \end{bmatrix} = \text{Line-of-sight unit vector pointing from the user to satellite } i$$

$$\Delta \mathbf{x} = \mathbf{x} - \hat{\mathbf{x}} = \text{true position/clock bias - linearization point}$$

**[0015]** At a step 230, the processor 16 computes the Error Covariance Matrix. The vector of measured pseudo-range residuals  $\Delta \tilde{\mathbf{p}}$  is the true pseudo-range residual vector plus the vector of residual errors  $\delta \mathbf{p}$  and is thus:

$$\begin{aligned} \Delta \tilde{\mathbf{p}} &= \mathbf{p} + \delta \mathbf{p} - \hat{\mathbf{p}} \\ &= \tilde{\mathbf{p}} - \hat{\mathbf{p}} \\ &= \mathbf{H} \Delta \mathbf{x} + \delta \mathbf{p} \end{aligned} \quad (2)$$

**[0016]** The processor 16 designates the post-update estimate of  $\Delta \mathbf{x}$  as  $\Delta \hat{\mathbf{x}}$ . Then, the processor 16 can define the vector of post-update measurement residuals as:

$$\boldsymbol{\xi} = \Delta \tilde{\mathbf{p}} - \mathbf{H} \Delta \hat{\mathbf{x}} \quad (3)$$

**[0017]** Each post-update measurement residual is the difference between the measured pseudo-range residual and the predicted pseudo-range residual based on the post-update estimate  $\Delta \hat{\mathbf{x}}$ .

[0018] A “weighted least-squares solution” can be determined by the processor 16 by finding the value of  $\Delta\hat{x}$  which minimizes the weighted sum of squared residuals. Thus, the processor 16 may minimize:

$$\xi^T \mathbf{W} \xi = (\Delta\tilde{\rho} - \mathbf{H}\Delta\hat{x})^T \mathbf{W} (\Delta\tilde{\rho} - \mathbf{H}\Delta\hat{x}) \quad (4)$$

where  $\mathbf{W}$  is an appropriate weighting matrix. The weighting matrix generally chosen is one which normalizes the residuals based on the uncertainty of each pseudo-range measurement. Thus, the processor 16 yields:

$$\mathbf{W} = \begin{bmatrix} 1/\sigma_1^2 & 0 & \cdots & 0 \\ 0 & 1/\sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\sigma_N^2 \end{bmatrix}, \text{ assuming uncorrelated measurements} \quad (5)$$

which represents the inverse of the pseudo-range measurement error covariance matrix assuming each pseudo-range error is uncorrelated with the others.

[0019] However, the vertical ionospheric delay component of each pseudo-range error is highly correlated with the others. If this correlation is known, then the processor 16 can take advantage of that knowledge by using the true pseudo-range measurement error covariance matrix  $\mathbf{R}$ . The weighting matrix then becomes

$$\mathbf{W} = \mathbf{R}^{-1} = \begin{bmatrix} \sigma_1^2 & E[\delta\rho_1\delta\rho_2] & \cdots & E[\delta\rho_1\delta\rho_N] \\ E[\delta\rho_1\delta\rho_2] & \sigma_2^2 & \cdots & E[\delta\rho_2\delta\rho_N] \\ \vdots & \vdots & \ddots & \vdots \\ E[\delta\rho_1\delta\rho_N] & E[\delta\rho_2\delta\rho_N] & \cdots & \sigma_N^2 \end{bmatrix}^{-1} \quad (6)$$

**[0020]** The value of  $\Delta\hat{\mathbf{x}}$  that minimizes (4) is determined by taking the derivative, setting it equal to zero, and solving for  $\Delta\hat{\mathbf{x}}$ . This yields:

$$\begin{aligned}\Delta\hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \Delta\tilde{\boldsymbol{\rho}} \\ &= \mathbf{S} \Delta\tilde{\boldsymbol{\rho}}\end{aligned}\tag{7}$$

where the processor 16 has defined the weighted least-squares solution matrix **S** as:

$$\mathbf{S} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W}\tag{8}$$

### Altitude Aiding

**[0021]** Barometric altitude can be used by the processor 16 to augment the GPS pseudo-range measurements. If it is used, the measurement matrix is then augmented as follows

$$\mathbf{H} = \begin{bmatrix} -LOS_{1x} & -LOS_{1y} & -LOS_{1z} & 1 \\ -LOS_{2x} & -LOS_{2y} & -LOS_{2z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -LOS_{Nx} & -LOS_{Ny} & -LOS_{Nz} & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}\tag{9}$$

**[0022]** This measurement matrix assumes that the incremental position vector (the first 3 elements) within  $\Delta\mathbf{x}$  are given in local-level coordinates (with the z axis down). The line-of-sight (LOS) elements then must also be

expressed in the local-level coordinates. The weighting matrix is also augmented as follows

$$\mathbf{W} = \mathbf{R}^{-1} = \begin{bmatrix} \sigma_1^2 & E[\delta\rho_1\delta\rho_2] & \cdots & E[\delta\rho_1\delta\rho_N] & 0 \\ E[\delta\rho_1\delta\rho_2] & \sigma_2^2 & \cdots & E[\delta\rho_2\delta\rho_N] & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ E[\delta\rho_1\delta\rho_N] & E[\delta\rho_2\delta\rho_N] & \cdots & \sigma_N^2 & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{baro}^2 \end{bmatrix}^{-1} \quad (10)$$

### Computing the Measurement Covariance Matrix

[0023] There are multiple methods that can be employed to determine the measurement error covariance matrix. In the case of a Kalman filter application, the temporal behavior (time-correlation) of the ionospheric delays may be modeled. The spatially correlated ionospheric error for satellite  $i$  can be modeled as a weighted sum of three independent normalized (sigma = 1.0) Gaussian random errors scaled by the nominal iono sigma value for that satellite as follows:

$$\delta\rho_{iono\_i} = \sigma_{iono\_i} \mathbf{k}_{iono\_i}^T \mathbf{x}_{ref} \quad (11)$$

where  $\mathbf{x}_{ref}$  is a 3x1 vector of independent Gaussian random errors with zero mean and a variance of 1. The weighting vector  $\mathbf{k}_{iono\_i}^T$  is determined by the processor 16 by first defining three grid points on the thin shell model of the ionosphere (at a height of 350 km) equally spaced in azimuth at a great circle distance of 1500 km from the user. The processor 16 may then define a 3x1 vector  $\mathbf{x}_{grid}$  of normalized

delays at these points. The delays at gridpoints  $i$  and  $j$  may be spatially correlated with each other based on the great circle distances between them according to:

$$E[x_{grid\_i} x_{grid\_j}^T] = 1 - \left(1 - e^{-d_{grid\_i,grid\_j} / d_{iono}}\right)^2 \quad (12)$$

where

$$\begin{aligned} d_{grid\_i,grid\_j} &= \text{great circle distance between grid point } i \text{ and grid point } j \\ d_{iono} &= \text{correlation distance of the ionospheric delay} = 4000 \text{ km} \end{aligned} \quad (13)$$

**[0024]** Using that relationship the processor 16 can form a 3x3 covariance matrix  $\mathbf{P}_{grid}$  which describes the correlations between each of the grid points:

$$\mathbf{P}_{grid} = E[\mathbf{x}_{grid} \mathbf{x}_{grid}^T]$$

**[0025]** If the delay processes that exist at these grid points are a certain linear combination of the reference independent Gaussian random errors, then they will have the desired spatial and temporal correlation. The processor 16 may assume that the desired linear combination is obtained by using a 3x3 upper-triangular mapping matrix  $\mathbf{U}_{grid}$  as follows:

$$\mathbf{x}_{grid} = \mathbf{U}_{grid} \mathbf{x}_{ref} \quad (14)$$

The grid covariance matrix is then:

$$\begin{aligned} \mathbf{P}_{grid} &= E[\mathbf{x}_{grid} \mathbf{x}_{grid}^T] = \mathbf{U}_{grid} E[\mathbf{x}_{ref} \mathbf{x}_{ref}^T] \mathbf{U}_{grid}^T \\ &= \mathbf{U}_{grid} \mathbf{U}_{grid}^T \end{aligned} \quad (15)$$

**[0026]** Therefore, the mapping matrix  $\mathbf{U}_{grid}$  can be formed by the processor 16 simply by factoring the covariance matrix  $\mathbf{P}_{grid}$ . Since the geometry of the three gridpoints is fixed, the covariance matrix  $\mathbf{P}_{grid}$  is constant and can thus be pre-computed by the processor 16. Now the processor 16 can choose a linear combination of the three grid-point delays that yields a normalized delay at the

pierce-point of the satellite  $i$  such that the proper spatial correlation with the three grid points (and thus, presumably, each of the other satellites) is achieved as follows:

$$\delta\rho_{norm\_iono\_i} = \mathbf{k}_{sat\_i\_grid}^T \mathbf{x}_{grid} \quad (16)$$

where

$\mathbf{k}_{sat\_i\_grid}$  = 3-vector of weighting factors

$\delta\rho_{norm\_iono\_i}$  = normalized delay at the satellite pierce-point

[0027] The satellite pseudo-range delay may be correlated to the delay at the  $k^{th}$  grid point according to:

$$E[\delta\rho_{norm\_iono\_i} \mathbf{x}_{grid\_k}^T] = 1 - \left(1 - e^{-d_{sat\_i\_grid\_k} / d_{iono}}\right)^2 \quad (17)$$

where

$$\begin{aligned} d_{sat\_i\_grid\_k} &= \text{great circle distance between the satellite pierce point and the grid point} \\ d_{iono} &= \text{correlation distance of the nominal ionospheric delay} \end{aligned} \quad (18)$$

[0028] The 1x3 covariance matrix  $\mathbf{P}_{sat\_i\_grid}$ , which defines the correlations between satellite  $i$  and each of the grid points, is

$$\begin{aligned} \mathbf{P}_{sat\_i\_grid} &= E[\delta\rho_{norm\_i} \mathbf{x}_{grid}^T] = E[\mathbf{k}_{sat\_i\_grid}^T \mathbf{x}_{grid} \mathbf{x}_{grid}^T] \\ &= \mathbf{k}_{sat\_i\_grid}^T \mathbf{P}_{grid} \end{aligned} \quad (19)$$

[0029] Therefore the weighting vector  $\mathbf{k}_{sat\_i\_grid}$  can be found by the processor 16 as follows:

$$\mathbf{k}_{sat\_i\_grid}^T = \mathbf{P}_{sat\_i\_grid} \mathbf{P}_{grid}^{-1} \quad (20)$$

[0030] Combining (14) and (16), the processor 16 can obtain the normalized vertical delays directly from the three independent reference delays as follows:

$$\begin{aligned}\delta\rho_{norm\_iono\_i} &= \mathbf{k}_{sat\_i\_grid}^T \mathbf{U}_{grid} \mathbf{x}_{ref} \\ &= \mathbf{k}_{iono\_i}^T \mathbf{x}_{ref}\end{aligned}\quad (21)$$

[0031] Thus, the weighting vector is:

$$\mathbf{k}_{iono\_i}^T = \mathbf{k}_{sat\_i\_grid}^T \mathbf{U}_{grid}\quad (22)$$

[0032] The processor 16 can form a vector of  $N$  normalized pseudo-range iono delays from (21) as follows:

$$\begin{aligned}\delta\mathbf{p}_{norm\_iono} &= \begin{bmatrix} \mathbf{k}_{sat\_1\_grid}^T \\ \mathbf{k}_{sat\_2\_grid}^T \\ \vdots \\ \mathbf{k}_{sat\_N\_grid}^T \end{bmatrix} \mathbf{U}_{grid} \mathbf{x}_{ref} \\ &= \mathbf{K}_{sat\_grid} \mathbf{U}_{grid} \mathbf{x}_{ref}\end{aligned}\quad (23)$$

[0033] The actual (non-normalized) delay along the line of sight can be obtained by the processor 16 by scaling the normalized delay by the sigma value for that satellite based on the geomagnetic latitude of the pierce-point and obliquity factor as defined in DO-229. In vector form, the processor 16 yields:

$$\begin{aligned}
\delta \mathbf{p}_{iono} &= \begin{bmatrix} \sigma_{iono\_1} & 0 & \cdots & 0 \\ 0 & \sigma_{iono\_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{iono\_N} \end{bmatrix} \delta \mathbf{p}_{norm\_iono} \\
&= \begin{bmatrix} \sigma_{iono\_1} & 0 & \cdots & 0 \\ 0 & \sigma_{iono\_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{iono\_N} \end{bmatrix} \mathbf{K}_{sat\_grid} \mathbf{U}_{grid} \mathbf{x}_{ref} \\
&= \Gamma \mathbf{K}_{sat\_grid} \mathbf{U}_{grid} \mathbf{x}_{ref}
\end{aligned} \tag{24}$$

[0034] The ionospheric delay error covariance matrix may be defined as:

$$\begin{aligned}
\mathbf{R}_{iono} &= E[\delta \mathbf{p}_{iono} \delta \mathbf{p}_{iono}^T] \\
&= \Gamma \mathbf{K}_{sat\_grid} \mathbf{U}_{grid} E[\mathbf{x}_{ref} \mathbf{x}_{ref}^T] \mathbf{U}_{grid}^T \mathbf{K}_{sat\_grid}^T \Gamma^T \\
&= \Gamma \mathbf{K}_{sat\_grid} \mathbf{P}_{grid} \mathbf{K}_{sat\_grid}^T \Gamma^T \\
\mathbf{R}_{iono} &= \begin{bmatrix} \sigma_{iono\_1}^2 & E[\delta \rho_{iono\_1} \delta \rho_{iono\_2}] & \cdots & E[\delta \rho_{iono\_1} \delta \rho_{iono\_N}] \\ E[\delta \rho_{iono\_1} \delta \rho_{iono\_2}] & \sigma_{iono\_2}^2 & \cdots & E[\delta \rho_{iono\_2} \delta \rho_{iono\_N}] \\ \vdots & \vdots & \ddots & \vdots \\ E[\delta \rho_{iono\_1} \delta \rho_{iono\_N}] & E[\delta \rho_{iono\_2} \delta \rho_{iono\_N}] & \cdots & \sigma_{iono\_N}^2 \end{bmatrix}
\end{aligned} \tag{25}$$

[0035] The rest of the pseudo-range measurement errors are assumed to be uncorrelated with a composite one-sigma value denoted by  $\sigma_{other\_i}$  for satellite  $i$ . For simplicity, the processor 16 can assume that the one-sigma value for each satellite is a constant six meters. The total measurement error covariance matrix is then:

$$\mathbf{R} = \mathbf{W}^{-1} = \mathbf{R}_{iono} + \begin{bmatrix} \sigma_{other\_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{other\_2}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{other\_N}^2 \end{bmatrix} \tag{26}$$

[0036] In a snapshot RAIM approach, the correlations between satellites are computed directly without the use of a grid. Computing the correlations between satellites directly may be both simpler and slightly more accurate.

[0037] Specifically, ionospheric error covariance may be modeled as a function of the great circle distance between the pierce points along the ionospheric shell (350 km above the earth's surface):

$$E[\delta\rho_{iono\_i}\delta\rho_{iono\_j}] = \sigma_{iono\_i}\sigma_{iono\_j} \left[ 1 - \left( 1 - e^{-d_{ij}/d_{iono}} \right)^2 \right]$$

where:

$d_{ij}$  = great circle distance between pierce points for sats  $i$  and  $j$

$d_{iono}$  = de-correlation distance = 4000 km

[0038] Ionospheric errors are highly correlated. As such:

$$\mathbf{W} = \mathbf{R}^{-1} = (\mathbf{R}_{iono} + \mathbf{R}_{other})^{-1}$$

where:

$$\mathbf{R}_{iono} = \begin{bmatrix} \sigma_{iono\_1}^2 & E[\delta\rho_{iono\_1}\delta\rho_{iono\_2}] & \cdots & E[\delta\rho_{iono\_1}\delta\rho_{iono\_N}] \\ E[\delta\rho_{iono\_1}\delta\rho_{iono\_2}] & \sigma_{iono\_2}^2 & \cdots & E[\delta\rho_{iono\_2}\delta\rho_{iono\_N}] \\ \vdots & \vdots & \ddots & \vdots \\ E[\delta\rho_{iono\_1}\delta\rho_{iono\_N}] & E[\delta\rho_{iono\_2}\delta\rho_{iono\_N}] & \cdots & \sigma_{iono\_N}^2 \end{bmatrix}$$

$$\mathbf{R}_{other} = \begin{bmatrix} \sigma_{other\_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{other\_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{other\_N}^2 \end{bmatrix}$$

### Error Covariance for the Weighted Least Squares Solution

[0039] At a step 240, the processor 16 computes a weighted least-squares solution. The error in the post-updated solution is:

$$\begin{aligned}\delta\hat{\mathbf{x}} &= \Delta\hat{\mathbf{x}} - \Delta\mathbf{x} \\ &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \Delta\tilde{\mathbf{p}} - \Delta\mathbf{x}\end{aligned}\tag{27}$$

[0040] Substituting (2) into (27) yields:

$$\begin{aligned}\delta\hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} (\mathbf{H} \Delta\mathbf{x} + \delta\boldsymbol{\rho}) - \Delta\mathbf{x} \\ &= \Delta\mathbf{x} - (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \delta\boldsymbol{\rho} - \Delta\mathbf{x} \\ &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \delta\boldsymbol{\rho} \\ &= \mathbf{S} \delta\boldsymbol{\rho}\end{aligned}\tag{28}$$

[0041] Thus, the solution matrix  $\mathbf{S}$  maps the pseudo-range errors into the post-updated solution error vector. The solution error covariance matrix may be defined as:

$$\begin{aligned}\mathbf{P} &= E[\delta\hat{\mathbf{x}}\delta\hat{\mathbf{x}}^T] = \mathbf{S} E[\delta\boldsymbol{\rho}\delta\boldsymbol{\rho}^T] \mathbf{S}^T \\ &= \mathbf{S} \mathbf{W}^{-1} \mathbf{S}^T \\ &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{W}^{-1} \mathbf{W} \mathbf{H} (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \\ &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1}\end{aligned}\tag{29}$$

**[0042]** The x and y horizontal position errors are statistically described by the upper 2x2 portion of  $\mathbf{P}$ . The major and minor axes of the horizontal position error ellipse are equal to the square roots of the maximum and minimum eigenvalues of this 2x2 matrix and represent the one-sigma errors in the corresponding directions. Thus, the one-sigma error in the worst-case direction is given by:

$$\begin{aligned}\sigma_{horz\_max} &= \sqrt{\lambda_{max}^{P(1:2,1:2)}} \\ &= \sqrt{\frac{P_{11} + P_{22}}{2} + \sqrt{\left(\frac{P_{11} - P_{22}}{2}\right)^2 + (P_{12})^2}}\end{aligned}\quad (30)$$

**[0043]** The one-sigma error in the vertical position is given by:

$$\sigma_{vert} = \sqrt{P_{33}}\quad (31)$$

**[0044]** Horizontal Figure of Merit is a conservative 95% fault-free error bound and may be computed by the processor 16 as the 2D RMS error from the error covariance matrix

$$HFOM = 2\sqrt{P(1,1) + P(2,2)}$$

**[0045]** Similarly, the Vertical Figure of Merit may be computed by the processor 16 as the 2-sigma vertical error from the error covariance matrix

$$VFOM = 2\sqrt{P(3,3)}$$

### Parity Space RAIM

[0046] At a step 250, the processor 16 computes at least one protection level value. In doing so, the processor 16 can employ a parity space technique. Recall that the weighted least squares solution may be expressed as:

$$\begin{aligned}\Delta\hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \Delta\tilde{\boldsymbol{\rho}} \\ &= \mathbf{S} \Delta\tilde{\boldsymbol{\rho}}\end{aligned}$$

$$\mathbf{W} = \mathbf{R}^{-1} = \left( E[\delta\boldsymbol{\rho} \delta\boldsymbol{\rho}^T] \right)^{-1}$$

[0047] The correlated measurement set can be transformed by the processor 16 into a set that is uncorrelated by factoring, using a matrix factorization method known in the art, the weighting matrix  $\mathbf{W}$ :

$$\mathbf{W} = \mathbf{L}^T \mathbf{L}$$

[0048] Where  $\mathbf{L}$  is the lower triangular square root of  $\mathbf{W}$ . This results in:

$$\begin{aligned}\Delta\hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{L}^T \mathbf{L} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{L}^T \mathbf{L} \Delta\tilde{\boldsymbol{\rho}} = [(\mathbf{L} \mathbf{H})^T \mathbf{L} \mathbf{H}]^{-1} (\mathbf{L} \mathbf{H})^T \mathbf{L} \Delta\tilde{\boldsymbol{\rho}} \\ &= (\bar{\mathbf{H}}^T \bar{\mathbf{H}})^{-1} \bar{\mathbf{H}}^T \Delta\bar{\boldsymbol{\rho}} = \bar{\mathbf{S}} \Delta\bar{\boldsymbol{\rho}} \\ \text{where } \bar{\mathbf{H}} &= \mathbf{L} \mathbf{H}, \Delta\bar{\boldsymbol{\rho}} = \mathbf{L} \Delta\tilde{\boldsymbol{\rho}}, \bar{\mathbf{S}} = (\bar{\mathbf{H}}^T \bar{\mathbf{H}})^{-1} \bar{\mathbf{H}}\end{aligned}$$

[0049] The covariance of the transformed measurement errors is:

$$\begin{aligned}
E[\delta\bar{\rho}\delta\bar{\rho}^T] &= E[\mathbf{L}\delta\rho\delta\rho^T\mathbf{L}^T] \\
&= \mathbf{L}\mathbf{W}^{-1}\mathbf{L}^T = \mathbf{L}(\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T \\
&= \mathbf{L}(\mathbf{L}^{-1}\mathbf{L}^{-T})\mathbf{L}^T \\
&= \mathbf{I}_N
\end{aligned}$$

[0050] Thus, it is seen that the transformed measurements are uncorrelated, each with unity variance. Multiplying our measurement equation by  $\mathbf{L}$  the processor 16 gets:

$$\begin{aligned}
\mathbf{L}\Delta\tilde{\rho} &= \mathbf{L}\mathbf{H}\Delta\mathbf{x} + \mathbf{L}\delta\rho \\
\Delta\bar{\rho} &= \bar{\mathbf{H}}\Delta\mathbf{x} + \delta\bar{\rho}
\end{aligned}$$

[0051] An  $N \times N$  orthogonal matrix  $\mathbf{Q}$  can be found by the processor 16, such that:

$$\mathbf{Q}\bar{\mathbf{H}} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ q_{31} & q_{32} & \cdots & q_{3N} \\ q_{41} & q_{42} & \cdots & q_{4N} \\ \hline q_{51} & q_{52} & \cdots & q_{5N} \\ \vdots & \vdots & \vdots & \vdots \\ q_{N1} & q_{N2} & q_{N3} & q_{NN} \end{bmatrix} \bar{\mathbf{H}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \bar{\mathbf{H}} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \\ \hline 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix}$$

[0052] Thus, if the processor 16 multiplies the measurement equation by  $\mathbf{Q}$ , the result is:

$$\mathbf{Q}\Delta\bar{\mathbf{p}} = \mathbf{Q}\bar{\mathbf{H}}\Delta\mathbf{x} + \mathbf{Q}\delta\bar{\mathbf{p}}$$

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \Delta\bar{\mathbf{p}} = \begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix} \Delta\mathbf{x} + \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \delta\bar{\mathbf{p}}$$

[0053] Where  $\mathbf{A}$  is the upper  $4 \times N$  portion of  $\mathbf{Q}$  and  $\mathbf{B}$  is the lower  $(N-4) \times N$  portion.

The following two equations result:

$$\mathbf{A}\Delta\bar{\mathbf{p}} = \mathbf{U}\Delta\mathbf{x} + \mathbf{A}\delta\bar{\mathbf{p}}$$

$$\mathbf{B}\Delta\bar{\mathbf{p}} = \mathbf{B}\delta\bar{\mathbf{p}} = \mathbf{p}$$

[0054] The first equation can be used to solve for the estimated least squares solution by setting the pseudo-range error to zero:

$$\mathbf{A}\Delta\bar{\mathbf{p}} = \mathbf{U}\Delta\hat{\mathbf{x}} \Rightarrow \Delta\hat{\mathbf{x}} = \mathbf{U}^{-1}\mathbf{A}\Delta\bar{\mathbf{p}}$$

[0055] This equation is a more efficient than the previously given one since it only requires inverting an upper-triangular matrix.

[0056] The second equation gives us the parity vector  $\mathbf{p}$  which is non-zero only when there are pseudo-range errors. In the absence of a failure, the parity covariance is:

$$E[\mathbf{p}\mathbf{p}^T] = \mathbf{B}E[\delta\bar{\mathbf{p}}\delta\bar{\mathbf{p}}^T]\mathbf{B}^T = \mathbf{B}\mathbf{B}^T$$

$$= \mathbf{I}_{N-4}$$

[0057] Thus the parity elements are also uncorrelated zero mean Gaussian random variables with unity variance.

[0058] In an embodiment, the processor 16 employs a chi-square method using the concept of pbias. In such an embodiment, the processor 16 uses the square of the parity magnitude as the discriminator (test statistic)  $d$  as follows:

$$d = \mathbf{p}^T \mathbf{p} = p_1^2 + p_2^2 + \dots + p_{N-4}^2$$

[0059] The discriminator will then have a central chi-square distribution with  $N-4$  degrees of freedom. The processor 16 places a threshold on this discriminator above which a failure is declared. This threshold  $D$  is computed by the processor 16 from the chi-square probability density function to yield an allowable false alarm probability.

[0060] Once the threshold has been set, the question becomes, with a failure on a single satellite that results in the discriminator just reaching the threshold, how large the resulting position error can be while meeting the probability of missed detection. A bias failure  $\varepsilon$  on the  $k$ th satellite will result in a solution error and a parity magnitude error of:

$$\delta \hat{\mathbf{x}} = \mathbf{S} \delta \boldsymbol{\rho} = \mathbf{S} \begin{bmatrix} 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} s_{1k} \\ s_{2k} \\ s_{3k} \\ s_{4k} \end{bmatrix} \varepsilon$$

$$|\mathbf{p}| = \sqrt{\mathbf{p}^T \mathbf{p}} \\ = \sqrt{\delta \boldsymbol{\rho}^T \mathbf{L}^T \mathbf{B}^T \mathbf{B} \mathbf{L} \delta \boldsymbol{\rho}} \\ = \varepsilon \sqrt{\Lambda_{kk}}$$

where  $\Lambda = \bar{\mathbf{B}}^T \bar{\mathbf{B}}$  and  $\bar{\mathbf{B}} = \mathbf{B} \mathbf{L}$

[0061] Thus the parity bias is related to the horizontal position error through the following slope:

$$\text{slope}(k) = \frac{\sqrt{s_{1k}^2 + s_{2k}^2}}{\sqrt{\Lambda_{kk}}}$$

**[0062]** The satellite with the largest slope is the most difficult to detect. This slope is referred to as  $\text{Slope}_{\max}$ . Figure 3 depicts a noise scatter that would occur if there was a bias on the most difficult to detect satellite in the presence of the expected noise. The specific bias that results in the percentage of data to the left of the detection threshold  $D$  equal to the missed detection probability is of particular interest. The parity magnitude associated with this bias is called “pbias.”

**[0063]** With the bias present, the discriminator (square of the parity magnitude) has a non-central chi-square distribution with  $N-4$  degrees of freedom. It can be shown that the non-centrality parameter  $\lambda$  of the chi-square distribution is:

$$\lambda = \text{pbias}^2$$

**[0064]** Thus, using the non-central chi-square probability density function, the processor 16 can determine the value for pbias which meets the required probability of missed detection.

**[0065]** The Horizontal Protection Level (HPL) is then:

$$\text{HPL} = \text{Slope}_{\max} \cdot \text{pbias}$$

### **Gaussian Method by Rotation of the Parity Space**

[0066] With reference to FIG. 4, through additional orthogonal transformations, an embodiment can rotate the parity space such that the parity error due to a bias failure on satellite  $k$  is entirely along axis 1 of the parity space.

$$\mathbf{p}^k = \mathbf{Q}^k \bar{\mathbf{B}} \delta \boldsymbol{\rho} = \bar{\mathbf{B}}^k \delta \boldsymbol{\rho} = \begin{bmatrix} \bar{b}_1^k & \cdots & \bar{b}_k^k & \cdots & \bar{b}_N^k \\ X & \vdots & 0 & \vdots & X \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X & \vdots & 0 & \vdots & X \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} (\bar{\mathbf{b}}^k)^T \\ X \\ X \\ X \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{bmatrix} = \bar{b}_k^k \varepsilon$$

[0067] Since the bias only shows up on axis 1, the result is a scalar, and the discriminator, in general, can be determined by the processor 16 using the following:

$$d^k = p_1^k = (\bar{\mathbf{b}}^k)^T \delta \boldsymbol{\rho}$$

[0068] With no failure, only correlated random errors  $\mathbf{w}$  are present:

$$d^k = (\bar{\mathbf{b}}^k)^T \mathbf{w} = (\mathbf{b}^k)^T \mathbf{L} \mathbf{w} = (\mathbf{b}^k)^T \bar{\mathbf{w}}$$

$$\text{where } \mathbf{B}^k = \mathbf{Q}^k \mathbf{B} = \bar{\mathbf{B}}^k \mathbf{L}^{-1}, \quad \bar{\mathbf{w}} = \mathbf{L} \mathbf{w}, \quad E[\bar{\mathbf{w}} \bar{\mathbf{w}}^T] = \mathbf{I}_N$$

$$\text{Note: } \sigma_{d^k}^2 = E[(d^k)^2] = E[(\mathbf{b}^k)^T \bar{\mathbf{w}} \bar{\mathbf{w}}^T \mathbf{b}^k] = E[(\mathbf{b}^k)^T \mathbf{b}^k] = 1$$

[0069] With a bias failure on satellite  $k$  plus the correlated random errors  $\mathbf{w}$  on each of the satellites, the discriminator can be determined by the processor 16 using the following:

$$d^k = \bar{b}_k^k \varepsilon + (\mathbf{b}^k)^T \bar{\mathbf{w}}$$

[0070] The impact of the noise in addition to failure on the horizontal position is:

$$\begin{aligned}\delta \mathbf{r}^h &= \mathbf{S}^h \delta \boldsymbol{\rho} = \mathbf{s}_k^h \boldsymbol{\varepsilon} + \mathbf{S}^h \mathbf{w} = \mathbf{s}_k^h \boldsymbol{\varepsilon} + \bar{\mathbf{S}}^h \mathbf{L} \mathbf{w} \\ &= \mathbf{s}_k^h \boldsymbol{\varepsilon} + \bar{\mathbf{S}}^h \bar{\mathbf{w}}\end{aligned}$$

where  $\mathbf{S}^h, \bar{\mathbf{S}}^h = 1^{\text{st}}$  two rows of  $\mathbf{S}$  and  $\bar{\mathbf{S}}$ , and  $\mathbf{s}_k^h =$  column  $k$  of  $\mathbf{S}^h$

[0071] Using the Gaussian probability density function, the threshold  $D$ , which meets the probability of false alarm, can be determined by the processor 16. At detection, the discriminator magnitude is equal to the threshold:

$$|d^k| = \left| \bar{b}_k^k \boldsymbol{\varepsilon} + (\mathbf{b}^k)^T \bar{\mathbf{w}} \right| = D$$

[0072] Assuming the failure is positive and much larger than the noise:

$$\left| \bar{b}_k^k \right| \boldsymbol{\varepsilon} + (\mathbf{b}^k)^T \bar{\mathbf{w}} = D \Rightarrow \boldsymbol{\varepsilon} = \frac{D}{\left| \bar{b}_k^k \right|} - \frac{(\mathbf{b}^k)^T \bar{\mathbf{w}}}{\left| \bar{b}_k^k \right|}$$

[0073] The resulting horizontal position error can be determined by the processor 16 using the following:

$$\begin{aligned}\delta \mathbf{r}^h &= \mathbf{s}_k^h \boldsymbol{\varepsilon} + \bar{\mathbf{S}}^h \bar{\mathbf{w}} = \mathbf{s}_k^h \left( \frac{D}{\left| \bar{b}_k^k \right|} - \frac{(\mathbf{b}^k)^T \bar{\mathbf{w}}}{\left| \bar{b}_k^k \right|} \right) + \bar{\mathbf{S}}^h \bar{\mathbf{w}} \\ &= \mathbf{s}_k^h \left( \frac{D}{\left| \bar{b}_k^k \right|} - \frac{1}{\left| \bar{b}_k^k \right|} \sum_{i=1}^N b_i^k \bar{w}_i \right) + \sum_{i=1}^N \bar{s}_i^h \bar{w}_i\end{aligned}$$

[0074] The position error magnitude in the direction of the failure is:

$$\begin{aligned}
 \delta r_k^h &= \frac{\mathbf{s}_k^h}{|\mathbf{s}_k^h|} \cdot \delta \mathbf{r}^h \\
 &= |\mathbf{s}_k^h| \left( \frac{D}{|\bar{b}_k^k|} - \frac{1}{|\bar{b}_k^k|} \sum_{i=1}^N b_i^k \bar{w}_i \right) + \sum_{i=1}^N \frac{\mathbf{s}_k^h \cdot \bar{\mathbf{s}}_i^h}{|\mathbf{s}_k^h|} \bar{w}_i \\
 &= \underbrace{\frac{|\mathbf{s}_k^h|}{|\bar{b}_k^k|} D}_{\text{Bias}} + \underbrace{\frac{1}{|\bar{b}_k^k|} \sum_{i=1}^N \left( \frac{\mathbf{s}_k^h \cdot \bar{\mathbf{s}}_i^h}{|\mathbf{s}_k^h|} |\bar{b}_k^k| - |\mathbf{s}_k^h| b_i^k \right)}_{\text{Noise}} \bar{w}_i
 \end{aligned}$$

[0075] Since the random pseudo-range error is uncorrelated with unity variance, the variance of the noise term about the mean can be determined by the processor 16 using the following:

$$\sigma_{\delta r_k^h} = \sqrt{\frac{1}{|\bar{b}_k^k|} \sum_{i=1}^N \left( \frac{\mathbf{s}_k^h \cdot \bar{\mathbf{s}}_i^h}{|\mathbf{s}_k^h|} |\bar{b}_k^k| - |\mathbf{s}_k^h| b_i^k \right)^2}$$

[0076] The Horizontal Protection Level for satellite  $k$  can be determined by the processor 16 using the following:

$$\begin{aligned}
 \text{HPL}_k &= \frac{\mathbf{s}_k^h}{|\mathbf{s}_k^h|} D + K_{md} \sigma_{\delta r_k^h} \\
 &= \frac{\mathbf{s}_k^h}{|\mathbf{s}_k^h|} K_{fa} + K_{md} \sigma_{\delta r_k^h}
 \end{aligned}$$

where  $K_{md}$  and  $K_{fa}$  are sigma multipliers set to meet the probabilities of false alarm and missed detection

[0077] This process is repeated for all  $N$  satellites and the total HPL can be determined by the processor 16 using the following:

$$\text{HPL} = \max(\text{HPL}_k), \quad k = 1, N$$

### **Ionospheric Error Model Calculations**

#### **Determination of Ionospheric Grid Points and Pierce Point Coordinates**

[0078] For a Kalman filter approach, and in order to utilize (17), the processor 16 may first determine the coordinates of each gridpoint and the coordinates of the satellite's ionospheric pierce point. Then, using those two sets of coordinates, the great circle distance between the pierce point and the grid point can be calculated by the processor 16. For either a Kalman filter or snapshot RAIM approach, knowing the coordinates of a point  $i$  (e.g., the system illustrated in FIG. 1, or "user") and the distance and azimuth from point  $i$  to a point  $j$  (e.g. the gridpoint), the coordinates of point  $j$  can be determined by the processor 16 as follows:

$$\lambda_j = \tan^{-1} \left( \frac{\sin \lambda_i \cos \psi_{ij} + \cos \lambda_i \sin \psi_{ij} \cos A_{ij}}{\sqrt{(\cos \lambda_i \cos \psi_{ij} - \sin \lambda_i \sin \psi_{ij} \cos A_{ij})^2 + \sin^2 \psi_{ij} \sin^2 A_{ij}}} \right) \quad (\text{A.1})$$

$$\Lambda_j = \Lambda_i + \tan^{-1} \left( \frac{\sin \psi_{ij} \sin A_{ij}}{\cos \lambda_i \cos \psi_{ij} - \sin \lambda_i \sin \psi_{ij} \cos A_{ij}} \right) \quad (\text{A.2})$$

$\lambda_i$  = Geodetic latitude of point  $i$

$\lambda_j$  = Geodetic latitude of point  $j$

$\Lambda_i$  = Geodetic longitude of point  $i$

$\Lambda_j$  = Geodetic longitude of point  $j$

$A_{ij}$  = Azimuth angle (bearing) from point  $i$  to point  $j$

$\psi_{ij}$  = Angular distance (earth's central angle) from point  $i$  to point  $j$

$$= \frac{d_{ij}}{R_e + h_i}$$

$d_{ij}$  = Great circle distance from point  $i$  to point  $j$

$R_e$  = Radius of the earth = 6378 km

$h_i$  = Height of the ionosphere thin shell model = 350 km

**[0079]** The coordinates of the ionospheric pierce point of the satellite can also be calculated using (A.1) and (A.2). In this case,  $\psi_{ij}$  represents the central angle from the user location to the pierce point and may be calculated by the processor 16 as follows:

$$\psi_{ij} = \frac{\pi}{2} - E - \sin^{-1} \left( \frac{R_e}{R_e + h_i} \cos E \right) \quad (\text{A.3})$$

where  $E$  is the elevation angle of the satellite from the user location with respect to the local tangent plane.

### Computing Elevation and Azimuth Angles of Satellite

[0080] The elevation angle  $E$  of a satellite is defined as the angle the line-of-sight vector makes with the user's local tangent (horizontal) plane. The azimuth angle  $A$  of the satellite is the angle of the line-of-sight vector with respect to true north as measured in the horizontal plane. Thus, we have the following

$$E = \text{ATAN2}\left(-u_{\text{LOS}_z}, \sqrt{u_{\text{LOS}_x}^2 + u_{\text{LOS}_y}^2}\right) \quad (\text{A.4})$$

$$A_0 = \text{ATAN2}(u_{\text{LOS}_y}, u_{\text{LOS}_x}) + \alpha \quad (\text{A.5})$$

$$A = \begin{cases} A_0, & -\pi \leq A_0 < \pi \\ A_0 - 2\pi, & A_0 \geq \pi \\ A_0 + 2\pi, & A_0 < -\pi \end{cases} \quad (\text{A.6})$$

where

$u_{\text{LOS}_x}, u_{\text{LOS}_y}, u_{\text{LOS}_z} = x, y,$  and  $z$  components of the line-of-sight vector  $\mathbf{u}_{\text{LOS}}^t$   
 $\alpha =$  wander angle (angle in azimuth from north to the  $x$  local-level frame axis)

[0081] Note that the azimuth angle is adjusted by  $\pm 2\pi$  so that the result is between  $-\pi$  and  $+\pi$ .

### Determination of Great Circle Distance

[0082] The great circle distance along the ionospheric thin shell model from a point  $i$  (e.g. satellite pierce point) to another point  $j$  (e.g. grid point) may be calculated by the processor 16 as follows:

$$d_{ij} = (R_e + h_r) \tan^{-1} \left( \frac{\sqrt{\cos^2 \lambda_j \sin^2 \Delta\Lambda_{ij} + (\cos \lambda_i \sin \lambda_j - \sin \lambda_i \cos \lambda_j \cos \Delta\Lambda_{ij})^2}}{\cos \lambda_i \cos \lambda_j \cos \Delta\Lambda_{ij} + \sin \lambda_i \sin \lambda_j} \right) \quad (\text{A.7})$$

where

$$\Delta\Lambda_{ij} = \Lambda_j - \Lambda_i$$

### **Ionospheric Variance Model**

**[0083]** The algorithm that may be executed by the processor 16 for calculation of the ionospheric model error variance may be from ICD-GPS-200C and DO-229D J.2.3. Note that the symbols in this section are unique to this section.

**[0084]** Using the satellite's elevation angle  $E$ , form the earth's central angle between the user position and the earth projections of ionospheric pierce point  $\psi_{pp}$  using equation (A.3).

**[0085]** Next, using the satellite's elevation angle  $E$ , the azimuth angle  $A$ , the earth's central angle  $\psi_{pp}$  and the user geodetic latitude  $\lambda_u$  and longitude  $\Lambda_u$ , determine the pierce point geodetic latitude  $\phi_{pp}$  and longitude  $\lambda_{pp}$  using equations (A.1) and (A.2).

**[0086]** Form the absolute value of the geomagnetic latitude of the ionospheric pierce point.

$$|\lambda_m| = \left| \lambda_{pp} + 0.064\pi \cos(\Lambda_{pp} - 1.617\pi) \right| \quad \text{radians} \quad (\text{A.8})$$

**[0087]** Form an estimate of the vertical delay error based on geomagnetic latitude

$$\tau_{vert} = \begin{cases} 9 \text{ meters,} & |\lambda_m| \leq 20 \text{ degrees} \\ 4.5 \text{ meters,} & 22.5^\circ < |\lambda_m| \leq 55^\circ \\ 6 \text{ meters,} & |\lambda_m| > 55^\circ \end{cases} \quad (\text{A.9})$$

**[0088]** Using the elevation angle  $E$ , calculate the square of the obliquity factor.

$$F_{pp}^2 = \frac{1}{1 - \left( \frac{R_e \cos(E)}{R_e + h_I} \right)^2} \quad (\text{A.10})$$

**[0089]** Form the modeled estimated variance of the ionospheric delay.

$$\sigma_{model}^2 = F_{pp}^2 \tau_{vert}^2 \quad (\text{A.11})$$

**[0090]** Form the estimated variance using the compensation that is applied if available from the receiver. (If not, assume zero).

$$\sigma_{comp}^2 = \left( \frac{CT_{IONO}}{5} \right)^2 \quad (\text{A.12})$$

**[0091]** Form the estimated variance of the ionospheric delay.

$$\sigma_{iono}^2 = \max(\sigma_{model}^2, \sigma_{comp}^2) \quad (\text{A.13})$$

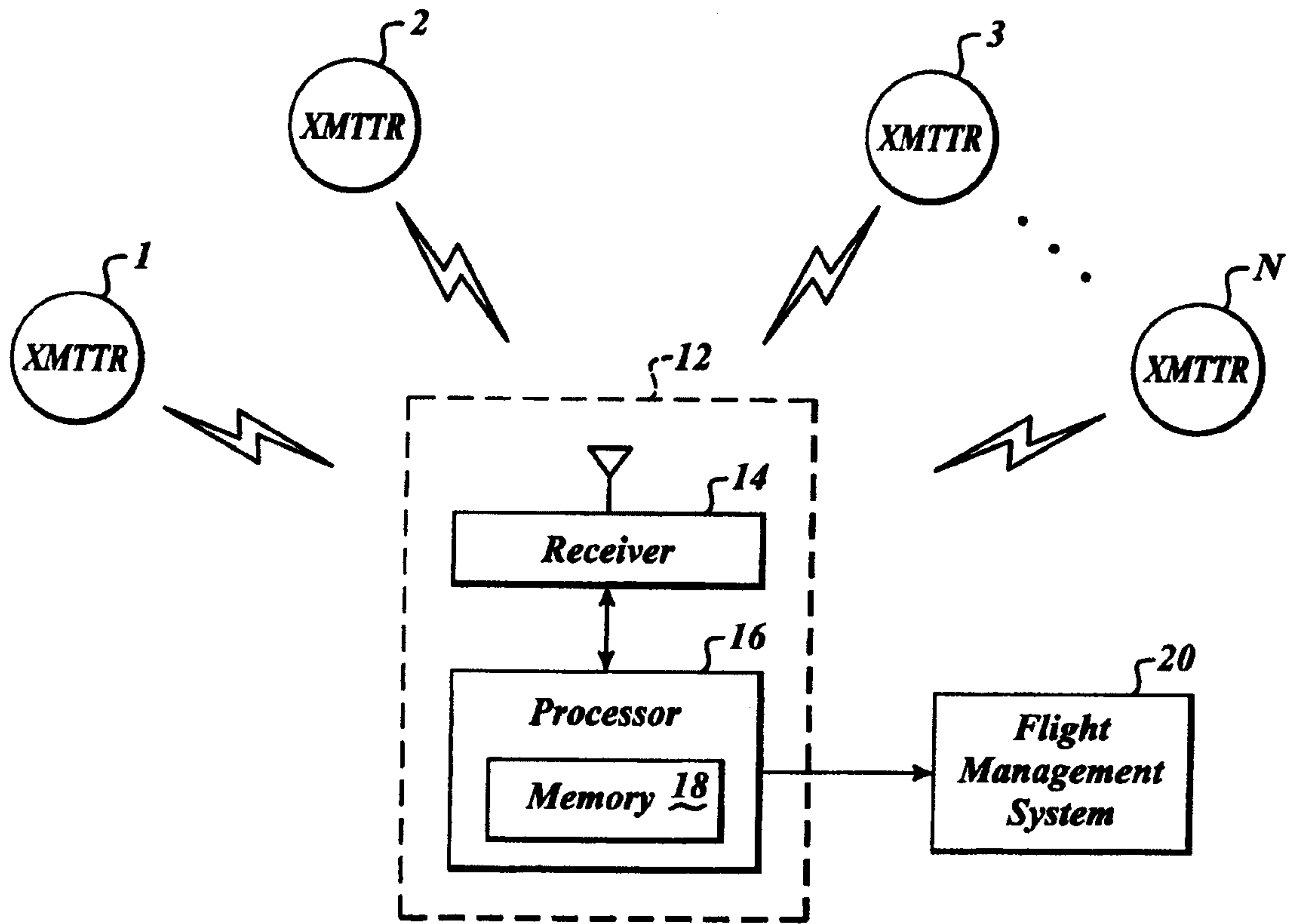
**[0092]** While the preferred embodiment of the invention has been illustrated and described as noted above, the scope of the claims should not be limited by any preferred embodiments or examples set forth, but should be given the broadest interpretation consistent with the description as a whole.

## CLAIMS

1. A navigation system for a vehicle having a receiver operable to receive a plurality of signals from a plurality of transmitters, the navigation system comprising:
  - 5 a processor; and
  - a memory device having stored thereon machine-readable instructions that, when executed by the processor, enable the processor to:
    - 10 determine a set of error estimates corresponding to pseudo-range measurements derived from the plurality of signals,
    - determine an error covariance matrix for a navigation solution based on the determined set of error estimates, the covariance matrix including off-diagonal elements representing ionospheric-delay spatial correlation data, and
    - 15 using a parity space technique, determine at least one protection level value based on the error covariance matrix.
2. The system of claim 1 wherein determining the error covariance matrix includes determining a spatially correlated ionospheric error associated with each of the transmitters.
3. The system of claim 2 wherein determining the error covariance matrix includes defining  
20 a plurality of grid points on a thin shell model of the ionosphere.
4. The system of claim 3 wherein the defined grid points are equally spaced in azimuth at a great circle distance from the system.
- 25 5. The system of claim 4 wherein the great circle distance is 1500 km.
6. The system of claim 3 wherein determining the error covariance matrix includes defining a vector of normalized ionospheric delays at the grid points.

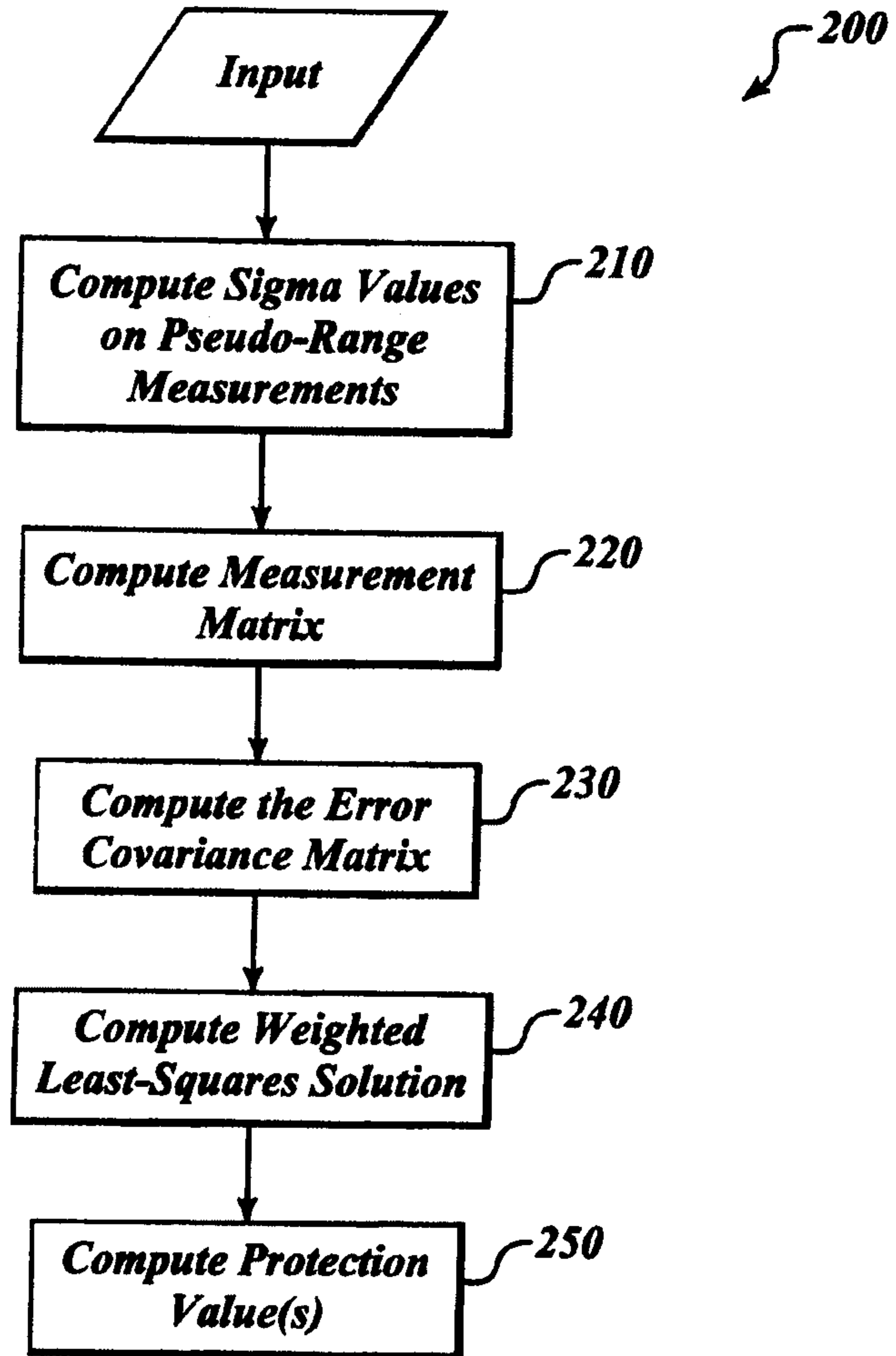
7. The system of claim 1 wherein the at least one protection level value comprises a horizontal protection level value.
8. The system of claim 1 wherein the at least one protection level value comprises a vertical protection level value.
9. The system of claim 1 wherein the at least one protection level value comprises a horizontal exclusion level value.
10. A non-transitory computer-readable medium having computer-executable instructions for performing steps comprising:  
determining a set of error estimates corresponding to pseudo-range measurements derived from the plurality of signals;  
determining an error covariance matrix for a navigation solution using ionospheric-delay data, wherein the error covariance matrix includes off-diagonal elements representing the spatially correlated ionospheric error for the plurality of signals;  
using a parity space technique, determining at least one protection level value based on the error covariance matrix; and  
displaying the at least one protection level value.
11. The medium of claim 10 wherein determining the error covariance matrix includes determining a spatially correlated ionospheric error associated with each of the transmitters.
12. The medium of claim 11 wherein determining the error covariance matrix includes defining a plurality of grid points on a thin shell model of the ionosphere.
13. The medium of claim 12 wherein the defined grid points are equally spaced in azimuth at a great circle distance from the system.
14. The medium of claim 13 wherein the great circle distance is 1500 km.

15. The medium of claim 12 wherein determining the error covariance matrix includes defining a vector of normalized ionospheric delays at the grid points.
- 5 16. The medium of claim 10 wherein the at least one protection level value comprises a horizontal protection level value.
17. The medium of claim 10 wherein the at least one protection level value comprises a vertical protection level value.
- 10 18. The medium of claim 10 wherein the at least one protection level value comprises a horizontal exclusion level value.
19. A method, comprising the steps of:
- 15 accessing from a first computer the computer-executable instructions of claim 10; and  
providing the instructions to a second computer over a non-transitory communication  
medium.



**FIG. 1**

2/4



**FIG. 2**

3/4

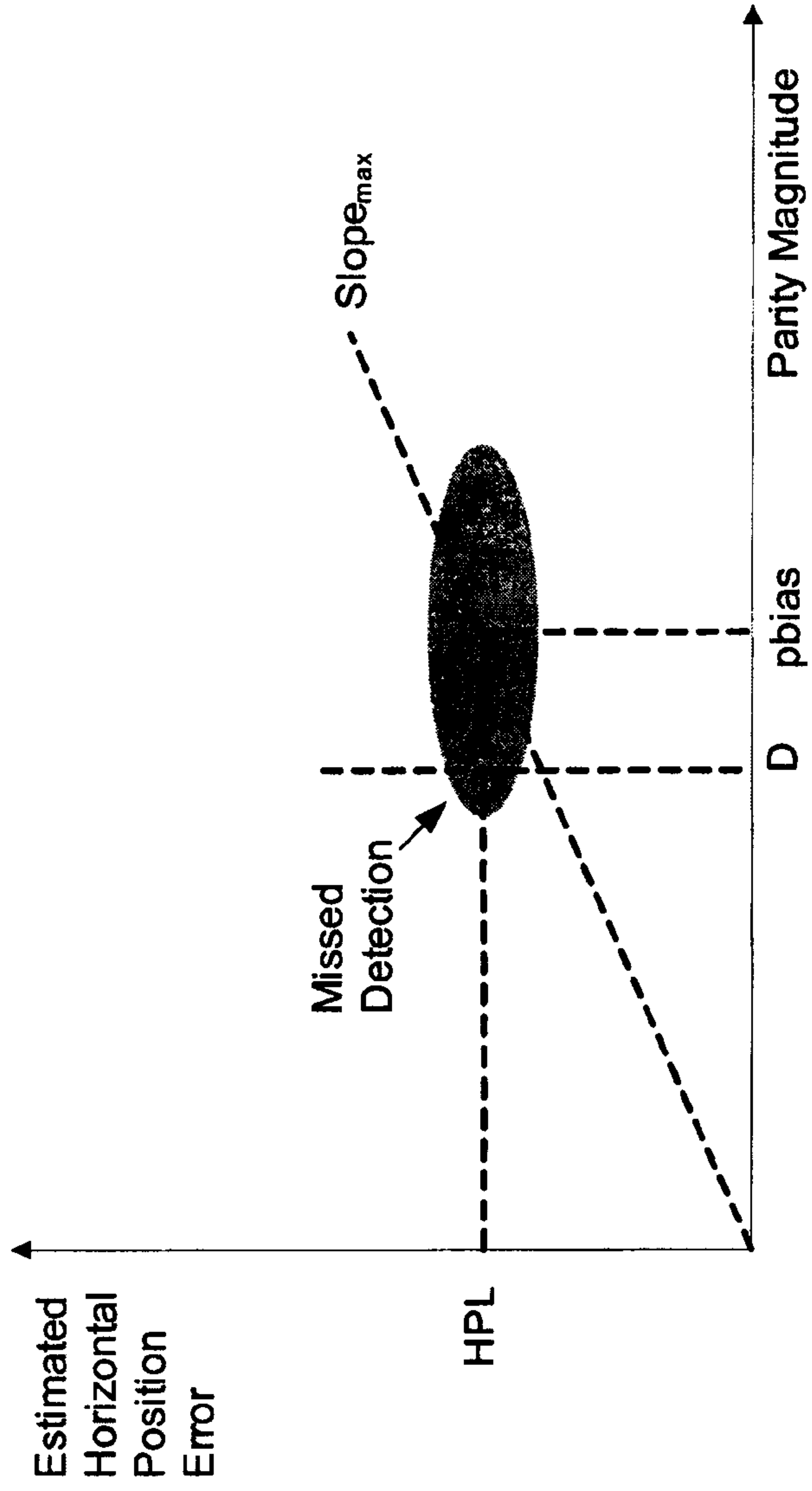


FIG. 3

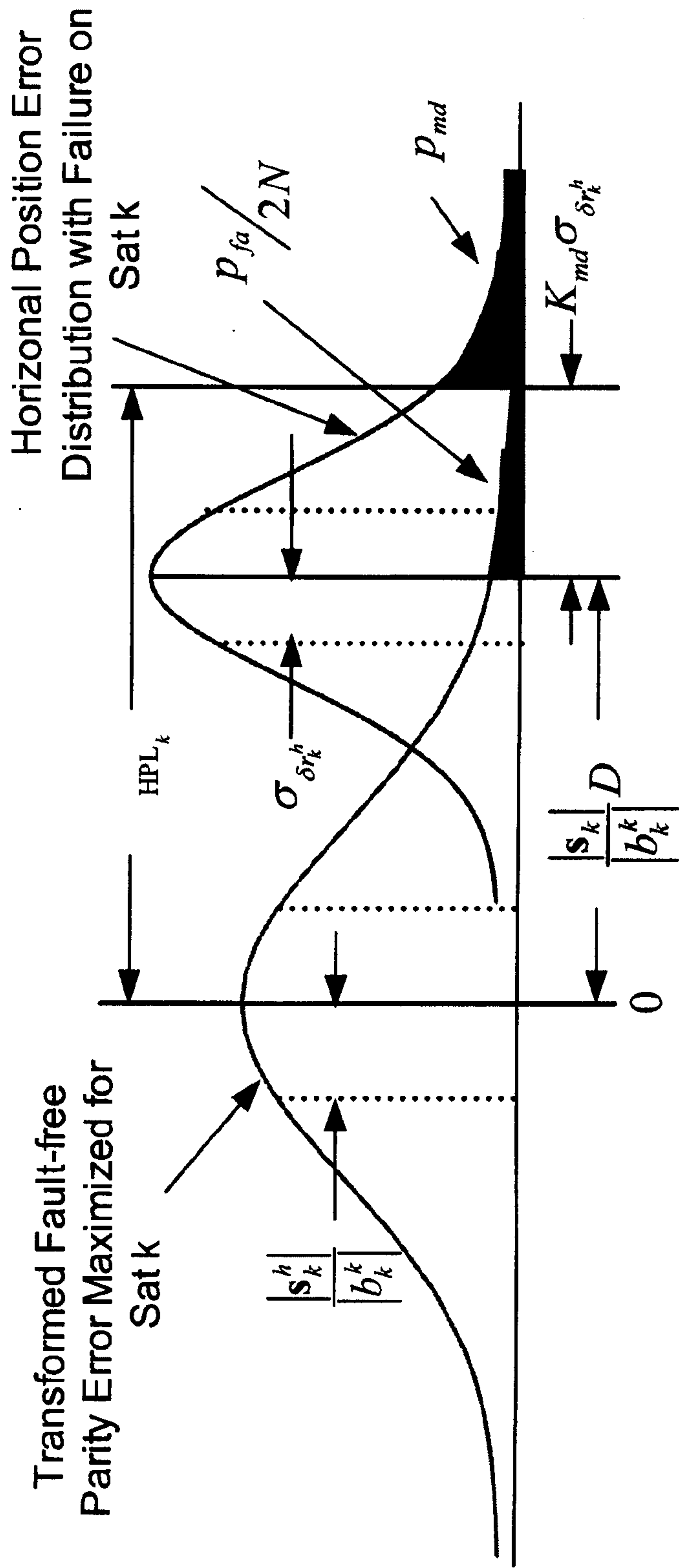


FIG. 4

