AN ORTHOGONAL TRANSFORMATION ALGORITHM
FOR GPS POSITIONING*

XIAO-WEN CHANG† AND CHRISTOPHER C. PAIGE†

Abstract. The Global Positioning System (GPS) is a satellite based navigation system. GPS satellites transmit signals that allow one to determine the location of GPS receivers. In GPS, a typical technique for kinematic position estimation is differential positioning where two receivers are used: one receiver is stationary and its exact position is known, and the other is roving and its position is to be estimated. We describe the physical situation and derive the mathematical model based on the difference of the so-called carrier phase measurements at the stationary and roving receivers. We then present a recursive least squares approach for position estimation. We take full account of the structure of the problem to make our algorithm efficient, and use orthogonal transformations to ensure numerical reliability of the algorithm. Simulation results are presented to demonstrate the performance of the algorithm. A comparison with the van Graas and Lee positioning algorithm [*Navigation, Journal of the Institute of Navigation*, 42 (1995), pp. 605–618] is given. Our algorithm is seen to be both efficient and accurate, but an additional contribution of this approach is that some of the drawbacks of double differencing are avoided, and yet the vector of double differenced integer ambiguities is still available and can be used to fix the integer ambiguities and handle satellite rising and setting.

Key words. Global Positioning System, recursive least squares, covariance matrices, integer ambiguities, orthogonal transformations, updating

AMS subject classifications. 62L12, 65F20, 65F25

PII. S1064827501397937

1. Introduction. The Global Positioning System (GPS) is a continuous coverage, all-weather, worldwide, satellite based navigation system. GPS consists of three segments: (1) the space segment consisting of satellites which broadcast signals; (2) the control segment steering the whole system; (3) the user segment including many types of receivers. GPS signals allow one to determine, with great accuracy, the location of GPS receivers (see, for example, [9], [12], [15], [17], [18], [19], and [20]). We have found [15] to be a particularly readable and useful text.

Each GPS satellite continuously transmits two carrier radio waves L1 and L2 with precise frequencies 1575.42 MHz and 1227.60 MHZ, respectively. Superimposed on the L1 carrier are the C/A code (coarse acquisition) and the P-code (precision), while superimposed on the L2 carrier is only the P-code. The C/A code is for civilian users, while the P-code is for U.S. military or authorized users. The C/A and P-codes allow a receiver to measure the signal travel time from satellite to receiver instantaneously. Many receivers are designed to receive only the L1 carrier, mainly for economical reasons. Basically, there are two kinds of measurements which can be used for positioning: code pseudorange (we will consider the C/A code) and carrier phase (we will consider the L1 carrier). Both measurements are subject to ionospheric refraction errors, tropospheric refraction errors, the receiver clock error, the satellite clock error, multipath errors, and random measurement noise (referred to simply as

"noise" below); see, for example, [15] for more details. But the multipath error and noise are typically 100 times larger for the code pseudorange measurement than for the carrier phase measurement, as can be seen from Table 1.1. Here the noise is random and the multipath is regarded as pseudorandom.

TABLE 1.1
Typical noise standard deviations; see, for example, [15, p. 153].

	Noise	Noise and multipath
Code	0.25–0.5m	0.5–1.0m
Carrier	1–2mm	0.5–1.0cm

The code pseudorange measurement gives an approximation to the true range between the receiver and the satellite and can easily be used for positioning. Using the carrier phase measurement for position estimation is not nearly so easy, but it is needed to get highly accurate position estimates (see Table 1.1).

In this paper we are interested in *kinematic* (also called *dynamic*) position estimation, i.e., for (often rapidly) moving objects. A typical technique for obtaining highly accurate position estimates is differential positioning where two receivers are used: one is stationary and is set up at a surveyed site whose exact position is known, and the other is roving and its position is to be estimated (see, for example, [15, sect. 4.8]). One example would be that of an aircraft (containing the roving receiver) approaching an airport (with a nearby stationary receiver). In real-time applications there is a radio link between the two receivers so that the information about the stationary receiver and the signal measurements it receives can be sent to the roving receiver. In this paper we assume the baseline (the position of the roving receiver relative to the stationary receiver) is short, say, within 30 kilometers, so that a satellite signal has almost the same ionospheric refraction error, and the same tropospheric refraction error, at the two receivers. Sometimes these errors are modeled, but even then there are modeling errors. Whether we model these errors or not, we almost always difference the two signal measurements from the same satellite at the two receivers to essentially eliminate these errors. This also eliminates the *satellite* clock error, which can also be modeled. This technique is called single differencing (see [5]). One can choose a particular satellite to be the "reference satellite" and then difference the single differenced measurements from the reference satellite with those from the other satellites. This is called the double differencing technique (see [2]). Using double differencing can eliminate the two *receivers'* clock errors. Double differencing is widely used in GPS computations, but it has some drawbacks. For example, it is numerically slightly dubious, it makes the measurements correlated, and it gives an unnecessary prominence to the reference satellite. Here we give a different approach which avoids these first two drawbacks, while maintaining all the advantages of double differencing.

Many people have studied kinematic position estimation based on carrier phase measurements alone (see, for example, [1], [4], [6], [10], [13], [21]). If the dynamics of the roving receiver can be modeled sufficiently accurately for cases where the movements of the roving receiver are predicable, then these can be included as the state equations of a Kalman Filter model (see, for example, [17, pp. 417–420] and [7, sect. 7.1]). But these are not always available, and even when they are, the GPS measurements are so accurate that the uncertainties in the state equations can sometimes *degrade* the results, compared with using the measurements alone. For example, partly because of air turbulence, this is usually the case for aircraft circling to land at an airport. To overcome this difficulty, van Graas and Lee [21] used what they called the

"complementary Kalman filter" approach for obtaining estimates from carrier phase measurements. In their approach, double differenced carrier phase measurements are used to construct measurement equations, and triple differenced measurements (the differences of double differenced measurements at two consecutive epochs) are used to propagate the user position, which, according to [21] is the key to achieving submeter accuracies with an initialization time of less than one minute. Essentially, the triple differences implicitly provide the state equations which are needed in Kalman filtering. However, these "state equations" do not provide any new information, since they are constructed from the double differenced measurement equations. In our view, this complementary Kalman filter is not necessary in this case. Artificially constructing a Kalman filter may delay the convergence, and the additional computation in each epoch is unnecessarily expensive. In this paper, which is based on single difference carrier phase measurement equations, we use a more straightforward recursive least squares (LS) approach to estimate the position when state equations are either not available or not certain enough to improve the results. In designing the algorithm we made full use of the structure to make the algorithm efficient, and used orthogonal transformation techniques to ensure that the algorithm is numerically reliable. Our simulations suggest our approach can usually converge slightly faster than the van Graas and Lee approach [21]. Also, in each epoch the computational cost of the former is less than half that of the latter. In cases where the state equations improve the estimates, they could be added to the equations here, and the general approach and good numerical techniques we describe can easily be extended to include them.

It is now common to use both carrier phase and code pseudorange data to obtain better results, but here we consider carrier phase data only. The purpose is to develop the mathematics, statistics, and reliable numerical algorithms for this case as cleanly as possible. Once this has been done and understood, it will be easier to develop an effective combination with code pseudorange measurements (see, for example, [3]). Also, we consider only the L1 carrier here, since many receivers can receive only the L1 signal. But it is easy to extend our approach to the dual frequency case.

This paper is organized as follows. In section 2 we derive the mathematical model we use for position estimation. In section 3 we present an efficient and numerically reliable algorithm for computing the position estimates. In section 4 some simulation results are given. Finally, a summary is given in section 5. To our knowledge, the new material is as follows. In section 3.2 we give a new way of handling the rank deficient problem of the single differenced measurement equations. Because of this, double differencing is avoided, the noise covariance matrices remain diagonal, and the integer nature of the ambiguities is not lost. Then we show how to effectively use orthogonal transformations to make full use of the structure of the problem to compute the LS estimates of the positions and the corresponding error covariance matrices. In section 3.5 we indicate that the LAMBDA method for fixing ambiguities can be used in the context of our computations, and in section 3.6 we show how satellite rising and setting can be handled seamlessly.

Notation used. We work with reals only and use i, j, k, l, m, n, r, s to denote indices and dimensions (superscript i will refer to the ith satellite, and subscript k to the kth time step, which is called the kth "epoch" in the GPS literature), while lowercase Greek letters will denote scalars. Other lowercase letters of the Roman alphabet (a, b, etc.) will denote vectors, while uppercase A, B, etc. will denote matrices. $\mathcal{R}(A)$ will denote the range space of A, and $A^\dagger$ is the Moore–Penrose generalized inverse of A. The unit matrix will be denoted by I and its ith column by e_i, while $e \equiv$

$(1, 1, \ldots, 1)^T$ (we use $\equiv$ to mean "is defined to be"). I_n will denote the $n \times n$ unit matrix. Throughout this paper we use the norm $\|x\| = \|x\|_2 = \sqrt{x^T x}$ for vectors. We will use $\mathcal{E}\{\cdot\}$ to denote the expected value and $\mathrm{cov}\{\cdot\}$ to denote the covariance; that is, $\mathrm{cov}\{x\} = \mathcal{E}\{(x - \mathcal{E}\{x\})(x - \mathcal{E}\{x\})^T\}$. $v \sim \mathcal{N}(\bar{v}, V)$ will indicate that v is a normally distributed random vector with expected value $\bar{v}$ and covariance V. We will use notation used in the GPS literature, unless it conflicts with the above notation. For example, we will use α to denote an individual integer ambiguity, and a to denote the integer ambiguity vector, whereas N is often used in the GPS literature.
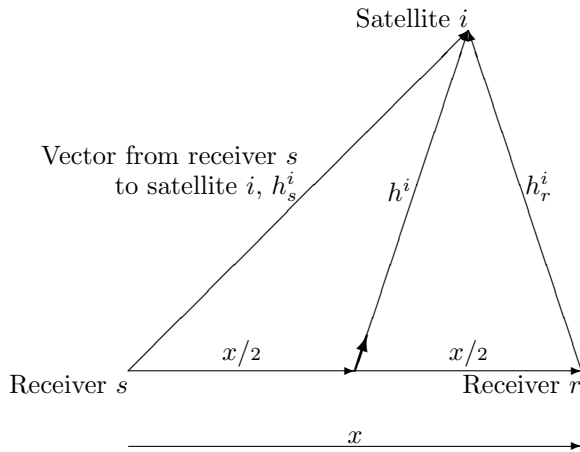

FIG. 2.1. *Geometry for two receivers and one satellite.*

2. The mathematical model. Here we derive the mathematical model for position estimation; however, we omit some minor subtleties in order to simplify the presentation. Suppose we have a stationary receiver s of known position, a roving receiver r whose position we wish to determine, and signals from m satellites in known positions. The positions of the GPS receivers and satellites can be expressed in the so-called World Geodetic System 1984 which defines an earth-centered, earth-fixed Cartesian coordinate system (see, for example, [15, Chap. 3]). Figure 2.1 gives the picture for satellite i. We want to find the baseline vector x, i.e., the vector pointing from receiver s to receiver r. By definition

h_s^i is the vector from receiver s to satellite i (h^i from midbaseline to satellite i),
e^i is the unit vector from the midpoint of the baseline to satellite i ($e^i = h^i/\|h^i\|$),
ρ_s^i is the range in wavelengths from receiver s to satellite i,
λ is the wavelength (for the L1 carrier signal used here, $\lambda \approx 19$ cm).

Note that superscripts indicate satellites and that subscripts indicate receivers. The next results are obvious:

$\|h_s^i\| = \lambda \rho_s^i$ is the distance from receiver s to satellite i,

(2.1) $x = h_s^i - h_r^i,$

(2.2) $e^i = \dfrac{h_s^i + h_r^i}{\|h_s^i + h_r^i\|},$

$$(2.3) \quad (e^i)^T x = \frac{(h_s^i + h_r^i)^T (h_s^i - h_r^i)}{\|h_s^i + h_r^i\|} = \frac{(\|h_s^i\| + \|h_r^i\|)(\|h_s^i\| - \|h_r^i\|)}{\|h_s^i + h_r^i\|} = \frac{1}{\mu^i} \lambda (\rho_s^i - \rho_r^i),$$

$$(2.4) \qquad \mu^i \equiv \frac{\|h_s^i + h_r^i\|}{\|h_s^i\| + \|h_r^i\|}.$$

It is straightforward to show, using $2h^i = h_s^i + h_r^i$, that μ^i satisfies

$$1 \leq \frac{1}{\mu^i} = \frac{\|h_s^i\| + \|h_r^i\|}{\|h_s^i + h_r^i\|} = \frac{\|h^i + x/2\| + \|h^i - x/2\|}{\|h_s^i + h_r^i\|} \leq \frac{2\|h^i\| + \|x\|}{\|h_s^i + h_r^i\|} = 1 + \frac{\|x\|}{\|h_s^i + h_r^i\|}.$$

Since the altitude of a satellite is about 20,200 kilometers above sea level, we will have $\|h_s^i + h_r^i\| > 40,000$ kilometers; thus if $\|x\| \leq 30$ kilometers, $1/\mu^i$ certainly lies in $[1, 1.00075]$ and approaches 1 as $\|x\| \to 0$. In fact, it can be shown that the largest $1/\mu^i$ occurs when x is perpendicular to h^i in the figure, and $\|h^i\|$ is as small as possible, giving $1/\mu^i = \sqrt{1 + [\|x/2\|/\|h^i\|]^2}$, which is about $1 + 0.28 \times 10^{-6}$ for $\|x\| \approx 30$ kilometers (when the stationary receiver is at or near sea level).

Probably because of the above, μ^i is replaced by 1 in all the literature we have seen. A relative error of 0.28×10^{-6} in 30 kilometers is about 8 millimeters. For most applications this error is negligible, but there are practical situations such as plate motion and crustal (earth surface) deformation where we actually want millimeter accuracy, so we will use the more correct equations here. We will see the computational cost of doing so is minimal, and the theory is essentially the same.

We now explain how the components of the model are evaluated. The position of the stationary receiver s is known, as is the position of each satellite i, so that h_s^i is known and used in the following. From (2.1), (2.2), and (2.4) we have

$$(2.5) \qquad e^i = \frac{2h_s^i - x}{\|2h_s^i - x\|}, \quad \mu^i = \frac{\|2h_s^i - x\|}{\|h_s^i\| + \|h_s^i - x\|},$$

$$\mu^i e^i = \frac{2h_s^i - x}{\|h_s^i\| + \|h_s^i - x\|}.$$

The true vector x will not be known, but we will see that at each step we will have an estimate of it, and this will be used to evaluate the above. We see that computing $e^i \mu^i$ is about twice the cost of computing e^i alone, and since this extra cost is minimal compared to later costs, we will use the more correct version (see (2.3)):

$$(2.6) \qquad (\mu^i e^i)^T x = \lambda (\rho_s^i - \rho_r^i).$$

We do not know the ρ_s^i, etc. exactly, but we will have available the carrier phase measurements. To progress further we must introduce time dependency. Suppose the signal from satellite i arrives at receiver s at epoch k (that is, at time point t_k) and its travel time is t_s^i. Briefly, the receiver generates a carrier signal to compare with the one it receives. Ideally all clocks are synchronized and the generated signal is synchronized in time with the satellite signal as it is generated. At t_1, when tracking begins, the receiver measures the *fractional* phase difference (i.e., part of a wavelength) between its signal and the received signal, and from then on tracks how this phase difference changes. With the definitions below, it follows that at t_k the carrier phase measurement at receiver s for satellite i ($\eta_s^i(t_k)$, in wavelengths) ideally satisfies $\rho_s^i(t_k) = \eta_s^i(t_k) + \alpha_s^i$, where α_s^i, called the integer ambiguity, is an unknown number of full wavelengths. The errors alter this to give (see, for example, [7, p. 165])

$$(2.7) \qquad \eta_s^i(t_k) + \alpha_s^i = \rho_s^i(t_k) - \iota_s^i(t_k) + \tau_s^i(t_k) + \beta^i(t_k - t_s^i) + \beta_s(t_k) + \nu_s^i(t_k),$$

where the "units" of each of the following components is "number of wavelengths":

- $\eta_s^i(t_k)$ denotes the *measured* carrier phase at time t_k. It is the fractional phase at t_k augmented by the change in integer cycles since the initial time t_1 (see, for example, the explanation in [15, p. 127]).
- α_s^i denotes the integer ambiguity. It is the initial number of full cycles between satellite i and receiver s when receiver s starts tracking the satellite signal at t_1. It is an unknown integer but remains the same while tracking is continued without loss of lock.
- $\rho_s^i(t_k)$ denotes the range between receiver s at arrival time t_k and satellite i at departure time $t_k - t_s^i$.
- $\iota_s^i(t_k)$ denote the ionospheric range error at t_k.
- $\tau_s^i(t_k)$ denote the tropospheric range error at t_k.
- $\beta^i(t_k - t_s^i)$ denotes the satellite clock range error at the transmission time $t_k - t_s^i$.
- $\beta_s(t_k)$ denotes the receiver clock range error at t_k.
- $\nu_s^i(t_k)$ denotes the noise, including multipath error, at t_k.

The ionospheric range error ι_s^i, the tropospheric range error τ_s^i, and the satellite clock range error β^i can be modeled (see, for example, [15, sect. 4.4 and 5.3]). If they have been, we simply assume that (2.7) is the model after the corrections have been applied, and the error terms in (2.7) are now the corresponding modeling errors. A more complete carrier phase measurement equation which includes phase offsets of the satellite-generated signal and the receiver-generated signal, as well as satellite and receiver hardware delays, can be found in [20, sect. 5.2]. We can assume that these errors are included in our two clock error terms, and note that these will be eliminated in our later computation. Subtracting the equation corresponding to receiver r from (2.7) and noticing that $-[\iota_s^i(t_k) - \iota_r^i(t_k)] + [\tau_s^i(t_k) - \tau_r^i(t_k)] + [\beta^i(t_k - t_s^i) - \beta^i(t_k - t_r^i)]$ will be negligible since the baseline is short, we obtain the single difference equation

$$(2.8) \quad \eta_s^i(t_k) - \eta_r^i(t_k) = \rho_s^i(t_k) - \rho_r^i(t_k) - (\alpha_s^i - \alpha_r^i) + \beta_s(t_k) - \beta_r(t_k) + \nu_s^i(t_k) - \nu_r^i(t_k).$$

Receivers s and r occur in every equation, so we can drop these indices and then indicate the time epoch k by subscript k if we define

$$\eta_k^i \equiv \eta_s^i(t_k) - \eta_r^i(t_k), \quad \alpha^i \equiv \alpha_s^i - \alpha_r^i, \quad \beta_k \equiv \beta_s(t_k) - \beta_r(t_k), \quad \nu_k^i \equiv \nu_s^i(t_k) - \nu_r^i(t_k).$$

This with (2.6) gives for (2.8)

$$\eta_k^i = \lambda^{-1}(\mu_k^i e_k^i)^T x_k - \alpha^i + \beta_k + \nu_k^i.$$

In the usual model, the ν_k^i for different satellites and different epochs are assumed to be unbiased independently distributed noises with the same normal distribution. Thus writing

$$(2.9) \quad y_k \equiv \begin{bmatrix} \eta_k^1 \\ \cdot \\ \eta_k^m \end{bmatrix}, \quad E_k \equiv \lambda^{-1} \begin{bmatrix} (\mu_k^1 e_k^1)^T \\ \cdot \\ (\mu_k^m e_k^m)^T \end{bmatrix}, \quad a \equiv \begin{bmatrix} \alpha^1 \\ \cdot \\ \alpha^m \end{bmatrix}, \quad v_k \equiv \begin{bmatrix} \nu_k^1 \\ \cdot \\ \nu_k^m \end{bmatrix},$$

we have for m satellites

$$(2.10) \qquad y_k = E_k x_k - a + e\beta_k + v_k, \qquad v_k \sim \mathcal{N}(0, \sigma^2 I_m).$$

This is the desired single differences of measurements equation for the carrier phase problem given the physical situation of m satellites with a known fixed receiver and

a roving receiver whose position is to be estimated. The almost identical equation appears in van Graas and Lee [21, eq. (1)].

We can rewrite (2.10) as

$$(2.11) \qquad y_k = \begin{bmatrix} e & E_k & -I_m \end{bmatrix} \begin{bmatrix} \beta_k \\ x_k \\ a \end{bmatrix} + v_k$$

and combine these for the epochs $k = 1, 2, \ldots$ to get

$$(2.12) \qquad \begin{bmatrix} y_1 \\ y_2 \\ . \\ y_k \end{bmatrix} = \begin{bmatrix} e & E_1 & & & & -I_m \\ & & e & E_2 & & -I_m \\ & & & & . & . \\ & & & & e & E_k & -I_m \end{bmatrix} \begin{bmatrix} \beta_1 \\ x_1 \\ \beta_2 \\ x_2 \\ . \\ . \\ \beta_k \\ x_k \\ a \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ . \\ . \\ v_k \end{bmatrix},$$

where the noise vector follows the distribution $\mathcal{N}(0, \sigma^2 I_{km})$. This is the mathematical model for which our positioning algorithm will be developed. Here for simplicity we have assumed that the number of visible satellites does not change from one epoch to the next. But during a long observation span, GPS satellites may rise and set. We will deal with this case in section 3.6. Notice that in E_k, $\mu_k^i e_k^i$ depends on the baseline x_k (see (2.5) and (2.9)). So we may write

$$E_k \equiv E(x_k).$$

This E_k is known once x_k is known. Given an approximation to x_k (because $E(x)$ is not very sensitive to changes in x, our estimate of x_{k-1} is usually sufficient), we can compute our approximation to E_k. Then given the measurements y_k, we can obtain a better estimate of x_k, and of E_k if necessary.

3. A recursive LS method for position estimation. In this section we use orthogonal transformation techniques to develop an efficient and numerically reliable method to recursively estimate the receiver positions and the error covariance matrices based on the model (2.12). We discuss the requirements for the number of satellites needed to give meaningful position estimates. We also give some remarks on the integer ambiguity issue and provide a method to handle the fact that satellites rise and set.

3.1. Some background for the LS solution. Suppose we have a linear model

$$(3.1) \qquad y = Gz + v, \qquad v \sim \mathcal{N}(0, \sigma^2 I),$$

where the matrix G is of full column rank. Then the best linear unbiased estimate (BLUE) of z is the solution of the LS problem

$$\min_z \|Gz - y\|^2.$$

In order to find the LS solution we can compute the QR factorization of G,

$$Q^T G \equiv \begin{bmatrix} U \\ W \end{bmatrix} G = \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where $Q = [U^T, W^T]$ is orthogonal and usually is the product of Householder transformations or Givens rotation matrices, R is nonsingular upper triangular, and U has the same number of rows as R (see, for example, [8, Chap. 5]). Since the 2-norm is unaffected by orthogonal transformations, the LS solution $\hat{z}$ satisfies

$$\hat{z} = \arg\min_z \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} z - \begin{bmatrix} Uy \\ Wy \end{bmatrix} \right\|^2 = \arg\min_z (\|Rz - Uy\|^2 + \|Wy\|^2)$$

so that

$$R\hat{z} = Uy.$$

This upper triangular system can be solved by back substitution. Thus

$$\hat{z} = G^\dagger y = R^{-1}Uy = R^{-1}U(Gz + v) = z + R^{-1}Uv,$$
(3.2) $$\qquad \mathcal{E}\{\hat{z} - z\} = 0, \qquad \text{cov}\{\hat{z} - z\} = \sigma^2 (R^T R)^{-1}.$$

The error in the estimate $\hat{z}$ and its covariance matrix have the following relationship:

(3.3) $$\qquad \mathcal{E}\{\|\hat{z} - z\|^2\} = \mathcal{E}\{\text{trace}((\hat{z} - z)(\hat{z} - z)^T)\} = \text{trace}(\text{cov}\{\hat{z} - z\}).$$

3.2. The orthogonal transformation approach. For the time being we assume that the E_j $(j = 1, \ldots, k)$ in (2.12) are known. Later we will discuss how to compute them. The condition for (2.12) to have a unique LS solution is that the coefficient matrix has full column rank. Unfortunately it does not—the dependency comes from the first column and last m columns of the matrix in (2.11) being linearly dependent. The most common approach to getting around this difficulty is to use the double differencing technique (see, for example, [2] and [14])—choose one satellite as the reference satellite and then subtract the single difference measurement equations corresponding to other satellites from the single difference measurement equation corresponding to the reference satellite. But as we mentioned in section 1, this has some drawbacks. Here we use a numerically reliable approach which avoids some of these drawbacks—orthogonal transformations of single differences.

Let $P \in \mathcal{R}^{m \times m}$ be an orthogonal transformation $(P^T P = PP^T = I)$ such that $P^T e = e_1 \sqrt{m}$. We could use a product of rotations to form P, but to be precise here we will use a Householder transformation (see, for example, [8, p. 209]), which here has the form

$$P \equiv I - u\left(\frac{2}{u^T u}\right) u^T, \qquad u \equiv e_1 - e/\sqrt{m}.$$

By simple algebraic operations, we obtain for this matrix

(3.4) $$P = \left[\begin{array}{c|c} \frac{1}{\sqrt{m}} & \frac{e^T}{\sqrt{m}} \\ \hline \frac{e}{\sqrt{m}} & I_{m-1} - \frac{ee^T}{m-\sqrt{m}} \end{array} \right] = \left[\begin{array}{c|ccc} \frac{1}{\sqrt{m}} & \frac{1}{\sqrt{m}} & \frac{1}{\sqrt{m}} & \cdot \\ \hline \frac{1}{\sqrt{m}} & 1 - \frac{1}{m-\sqrt{m}} & -\frac{1}{m-\sqrt{m}} & \cdot \\ \frac{1}{\sqrt{m}} & -\frac{1}{m-\sqrt{m}} & 1 - \frac{1}{m-\sqrt{m}} & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right],$$

where the symmetric matrix P is made of only three distinct values.

Writing

(3.5) $$\qquad [p_1, p_2, \ldots, p_m] \equiv [p_1, P_2] \equiv P$$

and applying P^T to (2.10), we obtain the initial orthogonal transformation of (2.10):

$$(3.6) \qquad \begin{bmatrix} p_1^T y_k \\ P_2^T y_k \end{bmatrix} = \begin{bmatrix} p_1^T E_k \\ P_2^T E_k \end{bmatrix} x_k - \begin{bmatrix} p_1^T a \\ P_2^T a \end{bmatrix} + e_1 \sqrt{m} \beta_k + \begin{bmatrix} p_1^T v_k \\ P_2^T v_k \end{bmatrix}.$$

Define the δ^i, d, and γ_k (where d will depend on our choice of P) by

$$(3.7) \qquad \begin{bmatrix} \delta^1 \\ \hline d \end{bmatrix} \equiv \begin{bmatrix} \delta^1 \\ \hline \delta^2 \\ . \\ \delta^m \end{bmatrix} \equiv \begin{bmatrix} -p_1^T a \\ -P_2^T a \end{bmatrix}, \qquad \gamma_k \equiv \delta^1 + \sqrt{m} \beta_k.$$

This eliminates one variable in (3.6) to give

$$(3.8) \qquad \begin{bmatrix} p_1^T y_k \\ P_2^T y_k \end{bmatrix} = \begin{bmatrix} 1 & p_1^T E_k \\ & P_2^T E_k & I_{m-1} \end{bmatrix} \begin{bmatrix} \gamma_k \\ x_k \\ d \end{bmatrix} + \begin{bmatrix} p_1^T v_k \\ P_2^T v_k \end{bmatrix}, \qquad \begin{bmatrix} p_1^T v_k \\ P_2^T v_k \end{bmatrix} \sim \mathcal{N}(0, \sigma^2 I_m).$$

Combining these for $k = 1, 2, \ldots$ and reordering gives

$$(3.9) \quad \begin{bmatrix} p_1^T y_1 \\ p_1^T y_2 \\ . \\ p_1^T y_k \\ P_2^T y_1 \\ P_2^T y_2 \\ . \\ P_2^T y_k \end{bmatrix} = \begin{bmatrix} 1 & & & & p_1^T E_1 & & & \\ & 1 & & & & p_1^T E_2 & & \\ & & . & & & & . & \\ & & & 1 & & & & p_1^T E_k \\ \hline & & & & P_2^T E_1 & & & & I_{m-1} \\ & & & & & P_2^T E_2 & & & & I_{m-1} \\ & & & & & & . & & & & . \\ & & & & & & & P_2^T E_k & & & & I_{m-1} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ . \\ \gamma_k \\ x_1 \\ x_2 \\ . \\ x_k \\ d \end{bmatrix} + \begin{bmatrix} p_1^T v_1 \\ p_1^T v_2 \\ . \\ p_1^T v_k \\ P_2^T v_1 \\ P_2^T v_2 \\ . \\ P_2^T v_k \end{bmatrix}.$$

Notice that the transformed noise vector, the second term on the right side of (3.9), still follows the same distribution $\mathcal{N}(0, \sigma^2 I_{km})$ after orthogonal transformations. From the structure of the coefficient matrix in (3.9) we observe that in order to obtain LS estimates of the position, we need only find LS estimates for the following submodel formed by the last $k(m-1)$ equations of (3.9):

$$(3.10) \quad \begin{bmatrix} P_2^T y_1 \\ P_2^T y_2 \\ . \\ P_2^T y_k \end{bmatrix} = \begin{bmatrix} P_2^T E_1 & & & & I_{m-1} \\ & P_2^T E_2 & & & I_{m-1} \\ & & . & & . \\ & & & P_2^T E_k & I_{m-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_k \\ d \end{bmatrix} + \begin{bmatrix} P_2^T v_1 \\ P_2^T v_2 \\ . \\ P_2^T v_k \end{bmatrix}.$$

Estimates of $\gamma_1, \ldots, \gamma_k$ in (3.7) are not usually wanted but could easily be found from the first k equations of (3.9) once the estimates of $x_1, \ldots, x_k$ are available.

Now look at the reduced model (3.10), and let B_k denote the coefficient matrix. Notice that B_k is $k(m-1) \times (3k+m-1)$, so that we can obtain a unique LS solution if B_k has full column rank, which certainly requires that $k(m-1) \geq 3k+m-1$, i.e.,

$$(3.11) \qquad\qquad m \geq 4 + \frac{3}{k-1}.$$

This relationship between the number of epochs and the required minimum number of satellites is displayed in Table 3.1. We see for position estimation that at least

TABLE 3.1
Satellite requirements.

# of epochs (k)	2	3	4	5	≥ 6
Minimum # of satellites	7	6	5	5	5

five satellites are required, so from now on we will assume that $m \geq 5$, a condition that is met for the vast majority of the time in practice. Let us now examine when B_k has full column rank. Suppose B_k does not have full column, so that there exist $x_1, \ldots, x_k$ and d which are not all zero such that

$$P_2^T E_j x_j + d = 0, \qquad j = 1, \ldots, k.$$

Notice that $P_2^T E_j$ is $(m-1) \times 3$. The geometry of the satellite positions is such that it is reasonable to assume that when $m \geq 4$, $P_2^T E_j$ has full column rank. In this case $d = 0$ implies $x_1 = \cdots = x_k = 0$, so $d \neq 0$ and

$$d \in \mathcal{R}(P_2^T E_j), \qquad j = 1, \ldots, k.$$

This shows that if the column spaces of $P_2^T E_1, \ldots, P_2^T E_k$ do not intersect, then B_k has full column rank. For $m \geq 5$ this is extremely likely for small k, and even more likely as k increases. In this paper we assume B_k has full column rank as long as k and m satisfy (3.11).

   Notice that the unknown vector d in (3.10) and the single difference integer ambiguity vector a satisfy $d = -P_2^T a$, where $P_2 \in \mathcal{R}^{m \times (m-1)}$ is part of an orthogonal matrix (see (3.4), (3.5), and (3.7)). Thus even though a is a vector of integers, d is usually not. The integer nature is lost when d is used. For some applications, highly accurate position estimates are demanded. Then fixing the integer ambiguities as integers is crucial (see, for example, [20, Chap. 8]) for details. Even though in this paper we will not discuss how to fix the integer ambiguities, we would still like to keep the integer nature of the ambiguities so that they can be explored when we want. We can do this by introducing a new unknown vector of integers with the same dimension as d to replace d. It also appears to be easiest to use a vector of integers to handle the situation that satellites rise or set, and in section 3.6 it is shown how to do this using our new unknown vector of integers. Define the double difference integer ambiguity vector a^{DD} by

$$(3.12) \qquad a^{DD} \equiv [\alpha^1 - \alpha^2, \alpha^1 - \alpha^3, \ldots, \alpha^1 - \alpha^m]^T,$$

where without loss of generality we have chosen satellite 1 as the "reference" satellite. Notice a^{DD} is still a vector of integers. Define

$$(3.13) \qquad F \equiv I_{m-1} - \frac{ee^T}{m - \sqrt{m}}, \qquad J \equiv [e, -I_{m-1}],$$

where F is nonsingular. It is easy to verify from (3.4) and (3.5) that

$$(3.14) \qquad P_2^T = -FJ, \qquad d = -P_2^T a = FJa = Fa^{DD}.$$

Replacing d in (3.10) by Fa^{DD}, we obtain the reduced model we will now work with:

$$(3.15) \quad \begin{bmatrix} P_2^T y_1 \\ P_2^T y_2 \\ \cdot \\ P_2^T y_k \end{bmatrix} = \begin{bmatrix} P_2^T E_1 & & & F \\ & P_2^T E_2 & & F \\ & & \cdot & \cdot \\ & & P_2^T E_k & F \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_k \\ a^{DD} \end{bmatrix} + \begin{bmatrix} P_2^T v_1 \\ P_2^T v_2 \\ \cdot \\ P_2^T v_k \end{bmatrix}.$$

Notice that we have not used double differencing, but the double differenced integer ambiguity vector is an unknown in this model.

To approach the LS estimate for this, let the QR factorization of $P_2^T E_j$ be

$$(3.16) \quad Q_j^T(P_2^T E_j) \equiv \begin{bmatrix} U_j \\ W_j \end{bmatrix} P_2^T E_j = \begin{bmatrix} R_j \\ 0 \end{bmatrix}, \ 3 \times (m-1) \ U_j, \ (m-4) \times (m-1) \ W_j,$$

where $Q_j = [U_j^T, W_j^T]$ is an orthogonal $(m-1) \times (m-1)$ matrix and R_j is a 3×3 nonsingular upper triangular matrix. Let

$$(3.17) \qquad\qquad Q_j^T(P_2^T y_j) = \begin{bmatrix} U_j \\ W_j \end{bmatrix} P_2^T y_j \equiv \begin{bmatrix} u_j \\ w_j \end{bmatrix}.$$

Transforming (3.10) by $\mathrm{diag}(Q_1^T, \ldots, Q_k^T)$ and reordering gives

$$(3.18) \quad \begin{bmatrix} u_1 \\ u_2 \\ . \\ u_k \\ \hline w_1 \\ w_2 \\ . \\ w_k \end{bmatrix} = \left[\begin{array}{cccc|c} R_1 & & & & U_1 F \\ & R_2 & & & U_2 F \\ & & . & & . \\ & & & R_k & U_k F \\ \hline & & & & W_1 F \\ & & & & W_2 F \\ & & & & . \\ & & & & W_k F \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_k \\ \hline a^{DD} \end{bmatrix} + \begin{bmatrix} U_1 P_2^T v_1 \\ U_2 P_2^T v_2 \\ . \\ U_k P_2^T v_k \\ \hline W_1 P_2^T v_1 \\ W_2 P_2^T v_2 \\ . \\ W_k P_2^T v_k \end{bmatrix},$$

where forming $U_j F$ and $W_j F$ can be done efficiently by using the special form of F (see (3.13)). The transformed noise vector still follows the distribution $\mathcal{N}(0, \sigma^2 I_{k(m-1)})$.

If a_k^{DD} denotes the LS estimate for a^{DD} at epoch k, and $x_{1|k}, x_{2|k}, \ldots, x_{k|k}$ are the LS estimates of $x_1, x_2, \ldots, x_k$ at epoch k, we see that a_k^{DD} is the LS estimate for the even smaller submodel:

$$(3.19) \qquad \begin{bmatrix} w_1 \\ w_2 \\ . \\ w_k \end{bmatrix} = \begin{bmatrix} W_1 F \\ W_2 F \\ . \\ W_k F \end{bmatrix} a^{DD} + \begin{bmatrix} W_1 P_2^T v_1 \\ W_2 P_2^T v_2 \\ . \\ W_k P_2^T v_k \end{bmatrix},$$

and once a_k^{DD} has been computed, we see from (3.18) that $x_{1|k}, x_{2|k}, \ldots, x_{k|k}$ can be computed by solving the upper triangular systems

$$(3.20) \qquad\qquad R_j x_{j|k} = u_j - U_j F a_k^{DD}, \qquad j = 1, \ldots, k.$$

Notice that the $x_{1|k}, x_{2|k}, \ldots, x_{k|k}$ can be computed in any order once a_k^{DD} is available. So if a_k^{DD} is updated, we could, for example, update $x_{k|k}$ without updating any of the earlier position estimates. Since the coefficient matrix in (3.19) is $k(m-4) \times (m-1)$, in order to get a unique LS solution for a^{DD} we require $k(m-4) \geq m-1$, which is equivalent to (3.11).

Now our problem is to obtain the estimate a_k^{DD} of a^{DD} from (3.19). We use a recursive approach. Suppose at epoch $k-1$ that the corresponding coefficient matrix in (3.19) has full column rank (in the initial stage the coefficient matrix does not have full column rank and we will discuss this case later in section 3.3). We also assume we have computed the following orthogonal transformations:

$$(3.21) \qquad T_{k-1}^T \begin{bmatrix} W_1 F \\ W_2 F \\ . \\ W_{k-1} F \end{bmatrix} = \begin{bmatrix} S_{k-1} \\ 0 \end{bmatrix}, \qquad T_{k-1}^T \begin{bmatrix} w_1 \\ w_2 \\ . \\ w_{k-1} \end{bmatrix} = \begin{bmatrix} \hat{w}_{k-1} \\ \bar{w}_{k-1} \end{bmatrix},$$

where T_{k-1} is orthogonal, and S_{k-1} is nonsingular upper triangular with the same number of rows $m-1$ as $\hat{w}_{k-1}$. Then at epoch k after obtaining W_k and w_k (cf. (3.16) and (3.17)), we perform the following orthogonal transformations:

$$(3.22) \quad \tilde{T}_k^T \begin{bmatrix} S_{k-1} \\ W_k F \end{bmatrix} = \begin{bmatrix} S_k \\ 0 \end{bmatrix}, \quad \tilde{T}_k^T \begin{bmatrix} \hat{w}_{k-1} \\ w_k \end{bmatrix} = \begin{bmatrix} \hat{w}_k \\ \bar{\bar{w}}_k \end{bmatrix}, \quad \bar{w}_k \equiv \begin{bmatrix} \bar{w}_{k-1} \\ \bar{\bar{w}}_k \end{bmatrix},$$

where $\tilde{T}_k$ is orthogonal, S_k is nonsingular upper triangular, and S_k and $\hat{w}_k$ each have $m - 1$ rows. The orthogonal transformations can be implemented by a sequence of Householder transformations, which make use of the upper triangular structure of S_{k-1}. But the matrices $\tilde{T}_k$ and T_{k-1} are neither formed nor stored. By using similar notation for the transformed noise vector, we get the transformed form of (3.19):

$$(3.23) \qquad \begin{bmatrix} \hat{w}_k \\ \bar{w}_k \end{bmatrix} = \begin{bmatrix} S_k \\ 0 \end{bmatrix} a^{DD} + \begin{bmatrix} \hat{v}_k \\ \bar{v}_k \end{bmatrix}, \qquad \begin{bmatrix} \hat{v}_k \\ \bar{v}_k \end{bmatrix} \sim \mathcal{N}(0, \sigma^2 I_{k(m-4)}).$$

Thus by solving the upper triangular system

$$(3.24) \qquad\qquad\qquad\qquad S_k a_k^{DD} = \hat{w}_k,$$

we obtain a_k^{DD}, the LS estimate of a^{DD} at epoch k. After this, we can solve (3.20) to obtain any $x_{j|k}$, the estimate of x_j at epoch k.

Remark 3.1. Since $E_k = E(x_k)$ $(k = 1, 2, \ldots)$, our problem of estimating the positions is actually nonlinear. We have to use approximations to E_k during the processing. Suppose we have obtained an estimate $x_{k-1|k-1}$ of x_{k-1}. Then we use $E(x_{k-1|k-1})$ as an approximation to E_k. This approximation is usually acceptable, since usually x_{k-1} and x_k are not far away from each other, and E_{k-1} and E_k are very close. Also, if necessary, after obtaining the estimate $x_{k|k}$ of x_k, we can use $E(x_{k|k})$ to approximate E_k, and we could even do some further iterations to get an improved estimate of x_k. However, neither of these last two produced any significant improvement for the situation in section 4. In fact, E varies so slowly as a function of x that it can save computations to update E_k only every five or so steps, depending on the application.

Remark 3.2. From (3.23) we observe $\|\bar{w}_k\|$ is (the norm of) the LS residual of (2.12). This information is useful for fault detection (see, for example, [11] and [20, Chap. 7]).

3.3. Computing the initial points. This section deals with the initial stage, which extends from epoch $k = 1$ until that epoch where the coefficient matrix in (3.19) reaches full column rank. The standard estimate from the LS model is not unique until the end of the initial stage, but because E_k depends on x_k, we need to provide some other estimates of these initial x_k.

At the beginning $(k = 1)$, we do not know x_1 (since this is what we want). But in many GPS applications we may know an approximate location of the roving receiver (in fact, we can often use code pseudorange measurements to estimate the initial position). Then we can use this to construct an approximation to E_1. If we do not have any information about the position of the roving receiver and do not bother to use code pseudorange measurements, we may take each e_1^i in E_1 to be the direction cosine from the *stationary* receiver to satellite i and take $\mu_1^i = 1$, for $i = 1, \ldots, m$ (see (2.9) and (2.5)); in other words, we may take $x_{1|1} = 0$.

Suppose we have already obtained $x_{j-1|j-1}$, the estimate of x_{j-1} at epoch $j-1$. We would like to get an estimate of x_j when the measurements at epoch j are available.

Here we use an idea of [21]. From (3.10) we obtain

$$P_2^T E_j x_j = P_2^T E_{j-1} x_{j-1} + P_2^T (y_j - y_{j-1}) - P_2^T (v_j - v_{j-1}).$$

Notice that $P_2^T (v_j - v_{j-1}) \sim \mathcal{N}(0, 2\sigma^2 I_{m-1})$. Finding the LS solution $x_{j|j}$ of the nearby problem

$$P_2^T E_j x_{j|j} \approx P_2^T E_{j-1} x_{j-1|j-1} + P_2^T (y_j - y_{j-1})$$

would give the following estimate for x_j:

(3.25) $$x_{j|j} = (P_2^T E_j)^\dagger [P_2^T E_{j-1} x_{j-1|j-1} + P_2^T (y_j - y_{j-1})].$$

But E_j is unknown, so we can approximate E_j by E_{j-1}, and with the QR factorization of $P_2^T E_{j-1}$ in (3.16) (with j replaced by $j-1$), we get the preliminary estimate

$$\bar{x}_{j|j} = x_{j-1|j-1} + R_{j-1}^{-1} U_{j-1} P_2^T (y_j - y_{j-1}).$$

Then we can use this $\bar{x}_{j|j}$ to construct (our new approximation to) E_j and compute the QR factorization of $P_2^T E_j$. From (3.25), we have the new estimate

$$x_{j|j} = R_j^{-1} U_j [P_2^T E_{j-1} x_{j-1|j-1} + P_2^T (y_j - y_{j-1})].$$

Of course we do not compute R_{j-1}^{-1} and R_j^{-1} in the above computations. Instead we solve upper triangular systems. We could use this new estimate to update E_j again and compute the QR factorization of $P_2^T E_j$, which could be used in the next epoch. But our tests show that there is no significant advantage in doing so.

Remember that we are assuming we have $m \geq 5$ satellites, and that W_j is $(m - 4) \times (m - 1)$ (see (3.16)), so that W_j might have only one row. The initial stages of (3.21)–(3.22) will differ from the general stages as follows. The submatrix 0 and subvector $\bar{w}_{k-1}$ will usually not appear in (3.21), while the submatrix 0 and subvector $\bar{\bar{w}}_k$ will usually not appear in (3.22). The computations will now be designed to ensure S_{k-1} and S_k have full row rank and be in upper trapezoidal form. We continue this initial process until S_k becomes *nonsingular upper triangular* in (3.22).

The number of epochs needed at the initial stage is usually the minimal k satisfying (3.11) for a fixed number of satellites m. For example, if there are six satellites available, then after three epochs the coefficient matrix usually has full column rank. With full column rank, the initial stage is complete.

3.4. Approximating the covariance matrices. In order to have some idea of the errors in the estimates of positions and the estimates of double difference integer ambiguities, we would like to know the corresponding covariance matrices (see section 3.5 for yet another reason why the covariances for double difference integer ambiguities are needed). Now if (3.1) was an exact *linear* model, then we could obtain the exact covariance matrices (see section 3.1). But since E_k depends on x_k, the matrices R_j in (3.16), (3.18), etc. depend on the unknowns, so here we will only *approximate* the true covariance matrices. To do so we will assume the R_j do not depend on the unknowns, since this appears to give acceptable results (see section 4). We can rewrite the top part of (3.23) as

(3.26) $$\hat{w}_k = S_k a^{DD} + \hat{v}_k, \qquad \hat{v}_k \sim \mathcal{N}(0, \sigma^2 I_{m-1}),$$

which with (3.24) gives

$$(3.27) \quad S_k(a_k^{DD} - a^{DD}) = \hat{v}_k, \qquad \mathrm{cov}\{a_k^{DD} - a^{DD}\} = \sigma^2 S_k^{-1} S_k^{-T} = \sigma^2 (S_k^T S_k)^{-1}.$$

Thus S_k/σ is the Cholesky factor of $[\mathrm{cov}\{a_k^{DD} - a^{DD}\}]^{-1}$. This is very nice, since the inverse of a covariance matrix is more useful than the covariance matrix itself in many cases, and for numerical reliability it is better to work with the Cholesky factor of a positive definite matrix rather than the matrix itself. For another advantage of having the Cholesky factor, see section 3.5.

For $j = 1, \ldots, k$ the covariance matrices $\mathrm{cov}\{x_{j|k} - x_j\}$ can be obtained as follows. From (3.18) and (3.26)

$$\begin{bmatrix} u_j \\ \hat{w}_k \end{bmatrix} = \begin{bmatrix} R_j & U_j F \\ 0 & S_k \end{bmatrix} \begin{bmatrix} x_j \\ a^{DD} \end{bmatrix} + \begin{bmatrix} U_j P_2^T v_j \\ \hat{v}_k \end{bmatrix}, \qquad \begin{bmatrix} U_j P_2^T v_j \\ \hat{v}_k \end{bmatrix} \sim \mathcal{N}(0, \sigma^2 I_{m+2}).$$

If we follow [16] and apply an orthogonal $Z_{j|k}^T$ from the left to zero the $U_j F$ block we obtain, with obvious notation,
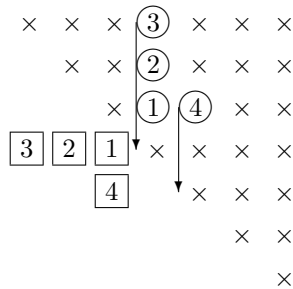
$$\begin{bmatrix} u_{j|k} \\ \hat{w}_{j|k} \end{bmatrix} \equiv Z_{j|k}^T \begin{bmatrix} u_j \\ \hat{w}_k \end{bmatrix} = \begin{bmatrix} R_{j|k} & 0 \\ \tilde{R}_{j|k} & S_{j|k} \end{bmatrix} \begin{bmatrix} x_j \\ a^{DD} \end{bmatrix} + \begin{bmatrix} v_{j|k} \\ \hat{v}_{j|k} \end{bmatrix},$$

$$\begin{bmatrix} v_{j|k} \\ \hat{v}_{j|k} \end{bmatrix} \equiv Z_{j|k}^T \begin{bmatrix} U_j P_2^T v_j \\ \hat{v}_k \end{bmatrix} \sim \mathcal{N}(0, \sigma^2 I_{m+2}).$$

We see $u_{j|k} = R_{j|k} x_{j|k}$, where $x_{j|k}$ also satisfies (3.20), and so

$$R_{j|k}(x_{j|k} - x_j) = v_{j|k}, \qquad \mathrm{cov}\{x_{j|k} - x_j\} = \sigma^2 R_{j|k}^{-1} R_{j|k}^{-T} = \sigma^2 (R_{j|k}^T R_{j|k})^{-1}.$$

Thus $R_{j|k}/\sigma$ is the Cholesky factor of $[\mathrm{cov}\{x_{j|k} - x_j\}]^{-1}$ if $R_{j|k}$ is upper triangular.

The computation for $R_{j|k}$ above uses Givens rotations to take advantage of the triangular structure of R_j and S_k and produce upper triangular $R_{j|k}$. Since $U_j F$ is $3 \times (m-1)$, $3(m-1)$ rotations are needed. We zero $U_j F$ column by column, and for each column we go from the bottom to the top. Only one element of $U_j F$ and one corresponding diagonal element of S_k are used to construct one rotation. This process can be described schematically in the case $m = 5$ as follows, where the symbol $\textcircled{i}$ indicates the element is eliminated in the ith rotation, while the symbol $\boxed{i}$ indicates this element is generated by the ith rotation:

$$\begin{array}{cccccccc} \times & \times & \times & \textcircled{3} & \times & \times & \times \\ & \times & \times & \textcircled{2} & \times & \times & \times \\ & & \times & \textcircled{1} & \textcircled{4} & \times & \times \\ \boxed{3} & \boxed{2} & \boxed{1} & \times & \times & \times & \times \\ & & \boxed{4} & & \times & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{array}$$

Similarly, but more simply, we can compute $\mathrm{var}\{\gamma_{j|k} - \gamma_j\}$ for $j = 1, \ldots, k$, if these are wanted.

3.5. Fixing integer ambiguities. It is known in the GPS community that while the use of code pseudorange measurements alone will only rarely give position accuracies approaching one meter (and this also depends on the application), if we use carrier phase measurements and treat the integer ambiguities as floating point numbers we can regularly obtain one meter accuracy, and often eventually obtain near centimeter accuracy. However, to obtain high accuracy for position estimates in as few epochs as possible, the integer nature of the ambiguities should be used. Notice that in our algorithm in section 3.2 we just regarded a^{DD} as a general real vector and did not fix it as a vector of integers. For some applications this will be sufficient, and it is not worth spending extra computation time to fix the integer ambiguities. But if one wants, the results obtained here can be used to fix the integer ambiguities. This can sometimes lead to accurate estimates in far fewer epochs, especially when there are only a small number of satellites available.

There are several approaches to fixing double difference integer ambiguities. One of the well-known approaches is called the LAMBDA method (see [20, Chap. 8]). This is based on $\text{cov}\{a_k^{DD} - a^{DD}\}$, and one type of input is the Cholesky factor of $[\text{cov}\{a_k^{DD} - a^{DD}\}]^{-1}$, which is provided by (3.27). If at epoch k, a_k^{DD} is fixed as a vector of integers, then we can get the corresponding position estimates from (3.20). Starting from epoch $k + 1$, we will not need to estimate a^{DD} anymore. We simply move Fa^{DD} to the left-hand side of (3.15). So for any epoch l after k, we need only solve those upper triangular systems that we are interested in among

$$R_j x_{j|l} = u_j - U_j(Fa_k^{DD}), \qquad j = 1, \ldots, l, \quad l > k.$$

3.6. Handling satellite rising and setting. In the model (2.12) we assumed that we had the same satellites during the whole observation period. But if the observation span is long, there will often be satellite rising and/or setting, and we will handle this now. It is reasonable to assume that this does not occur at the initial stage (see section 3.3) because that stage is so short. Even if it does, we can use the measurements from common satellites for each pair of consecutive epochs to estimate the initial points (see (3.25)). So we just consider the regular stage.

From (3.22), (3.24), and (3.20), we see that the major task is to update the estimate of the double difference integer ambiguity vector from epoch $k-1$ to epoch k. When this has been done, the position estimates at epoch k can easily be obtained. So the key to handling satellite rising and setting is to find the equivalents of (3.22), (3.24), and (3.20). Due to satellite rising and/or setting, the double difference integer ambiguity (DDIA) vectors may be different for different epochs, so we will use notation such as $a^{DD}(k)$ instead of a^{DD}. Because the number of visible satellites changes, we will write $P(k)$, $F(k)$, and $J(k)$ instead of P in (3.5) and F and J in (3.13), but for simplicity we will still use e to denote a vector of ones, no matter what its dimension.

Remember from (3.12) that a DDIA element has the form $\alpha^i - \alpha^n$, and we will say this element "corresponds to" the nonreference satellite n, and i is the reference satellite. It is important to be aware that we will use the *same* reference satellite for *every* element in *every* DDIA vector at a given epoch j. This reference satellite must be visible at epoch j, so if it sets between epochs $k-1$ and k, for some $k > j$, then (see Case 2 later) we will arrange to use a *different* reference satellite at epoch k. For $k = 1, 2, \ldots$, let $\tilde{a}^{DD}(k)$ be the DDIA vector for some reference satellite which is visible at epoch k, where the elements of $\tilde{a}^{DD}(k)$ correspond to *all* the other satellites we have encountered from epoch 1 to epoch k. If a satellite sets (perhaps behind a mountain) and rises again, it will be considered a new satellite. Thus when a satellite

rises, the dimension of $\tilde{a}^{DD}$ will increase by one, but a setting satellite leaves the dimension unchanged. In our constant satellite case $\tilde{a}^{DD}(k)$ was just a^{DD} in (3.12).

Assume at the end of epoch $k-1$ that we have obtained the equivalent of the top part of (3.23) for epoch $k-1$:

$$(3.28) \qquad \tilde{w}_{k-1} = \tilde{S}_{k-1}\tilde{a}^{DD}(k-1) + \tilde{v}_{k-1}, \qquad \tilde{v}_{k-1} \sim \mathcal{N}(0, \sigma^2 I),$$

where $\tilde{S}_{k-1}$ is nonsingular upper triangular. We can partition $\tilde{a}^{DD}(k-1)$ as

$$(3.29) \qquad \tilde{a}^{DD}(k-1) = \begin{bmatrix} \tilde{a}_d^{DD}(k-1) \\ a^{DD}(k-1) \end{bmatrix},$$

where for $k = 1, 2, \ldots$, the elements of $\tilde{a}_d^{DD}(k)$ correspond to the nonreference satellites which have gone *down* between epochs 1 and k, and the elements of $a^{DD}(k)$ correspond to the nonreference satellites which are visible at epoch k. Then with compatible partitioning, we can rewrite (3.28) as

$$(3.30) \qquad \begin{bmatrix} w_{k-1}^{(1)} \\ \hat{w}_{k-1} \end{bmatrix} = \begin{bmatrix} \tilde{S}_{k-1}^{(1)} & \tilde{S}_{k-1}^{(2)} \\ 0 & S_{k-1} \end{bmatrix} \begin{bmatrix} \tilde{a}_d^{DD}(k-1) \\ a^{DD}(k-1) \end{bmatrix} + \begin{bmatrix} v_{k-1}^{(1)} \\ \hat{v}_{k-1} \end{bmatrix},$$

where both $\tilde{S}_{k-1}^{(1)}$ and S_{k-1} are nonsingular upper triangular. Notice that if no satellites rise or set from epoch 1 to epoch $k-1$, then the top part of (3.30) will disappear and the bottom part is just the top part of (3.23) with k replaced by $k-1$. In the following we will combine (3.30) with relevant equations at epoch k in order to obtain the position estimates. It is helpful to consider two cases separately.

Case 1. The reference satellite at epoch $k-1$ remains at epoch k. We still use it as the reference satellite at epoch k. When a satellite sets between epochs $k-1$ and k, we simply drop the corresponding measurement equation from (2.10). When a satellite rises between epochs $k-1$ and k, we just append one equation to (2.10). We rewrite the single difference measurement equation (2.10) at epoch k for the new situation as

$$y_k = E_k x_k - a(k) + e\beta_k + v_k, \qquad v_k \sim \mathcal{N}(0, \sigma^2 I),$$

where the first scalar equation corresponds to the reference satellite, and $a(k)$ is the single difference integer ambiguity vector at epoch k. Applying $P_2(k)^T$ (cf. (3.5)) to the above equation and using the relationship $d(k) = -P_2(k)^T a(k) = F(k)J(k)a(k) = F(k)a^{DD}(k)$ (cf. (3.14)) gives the new kth block equation of (3.15):

$$P_2(k)^T y_k = P_2(k)^T E_k x_k + F(k)a^{DD}(k) + P_2(k)^T v_k, \qquad P_2(k)^T v_k \sim \mathcal{N}(0, \sigma^2 I).$$

Then using the QR factorization of $P_2(k)^T E_k$ (cf. (3.16) and (3.17)) gives the new kth and $2k$th block equations of (3.18), respectively:

$$(3.31) \quad u_k = R_k x_k + U_k F(k)a^{DD}(k) + U_k P_2(k)^T v_k, \quad U_k P_2(k)^T v_k \sim \mathcal{N}(0, \sigma^2 I),$$
$$(3.32) \quad w_k = \qquad W_k F(k)a^{DD}(k) + W_k P_2(k)^T v_k, \quad W_k P_2(k)^T v_k \sim \mathcal{N}(0, \sigma^2 I).$$

The main task is to combine (3.32) with (3.28) (i.e., (3.30)), which involves $\tilde{a}^{DD}(k-1)$ rather than $a^{DD}(k)$, to obtain the equivalent of (3.28) for $\tilde{a}^{DD}(k)$ at epoch k. The resulting LS estimate of $\tilde{a}^{DD}(k)$ can then be used to give the position estimates based on (3.31) and corresponding equations for epochs $j = 1, 2, \ldots, k-1$,

$$(3.33) \quad u_j = R_j x_j + U_j F(j)a^{DD}(j) + U_j P_2(j)^T v_j, \quad U_j P_2(j)^T v_j \sim \mathcal{N}(0, \sigma^2 I).$$

The DDIA vector $a^{DD}(k)$ can be partitioned as follows:

$$(3.34) \qquad a^{DD}(k) = \begin{bmatrix} a_r^{DD}(k) \\ a_u^{DD}(k) \end{bmatrix},$$

where the elements of $a_r^{DD}(k)$ correspond to the nonreference satellites which are visible at epoch $k-1$ and *remain* at epoch k, and those of $a_u^{DD}(k)$ correspond to the nonreference satellites which come *up* between epochs $k-1$ and k. Let $a_d^{DD}(k)$ denote the DDIA vector whose elements correspond to the nonreference satellites which go *down* between epochs $k-1$ and k. Then for $k = 1, 2, \ldots$, we can summarize the DDIA vectors at epoch k as follows, where every element of every DDIA vector at epoch k involves the same *reference* satellite, which must be visible at epoch k:

$\tilde{a}^{DD}(k)$: (whose elements correspond to) all the nonreference satellites that were visible for at least one epoch from epoch 1 to epoch k;

$\tilde{a}_d^{DD}(k)$: all the nonreference satellites that went *down* between epochs 1 and k;

$a^{DD}(k)$: all the nonreference satellites that are visible at epoch k;

$a_r^{DD}(k)$: all the nonreference satellites that were visible at epoch $k-1$ and *remain* at epoch k;

$a_u^{DD}(k)$: all the nonreference satellites that come *up* between epochs $k-1$ and k;

$a_d^{DD}(k)$: all the nonreference satellites that go *down* between epochs $k-1$ and k.

Obviously $a^{DD}(k-1)$ is a rearrangement of the combined elements of $a_r^{DD}(k)$ and $a_d^{DD}(k)$, so we can find a permutation $\Pi = [\Pi_1, \Pi_2]$ such that

$$(3.35) \qquad \Pi^T a^{DD}(k-1) = \begin{bmatrix} \Pi_1^T a^{DD}(k-1) \\ \Pi_2^T a^{DD}(k-1) \end{bmatrix} = \begin{bmatrix} a_d^{DD}(k) \\ a_r^{DD}(k) \end{bmatrix}.$$

The following are obvious relationships between these DDIA vectors that we will use:

$$(3.36) \qquad \tilde{a}_d^{DD}(k) = \begin{bmatrix} \tilde{a}_d^{DD}(k-1) \\ a_d^{DD}(k) \end{bmatrix}, \quad \tilde{a}^{DD}(k) = \begin{bmatrix} \tilde{a}_d^{DD}(k) \\ a^{DD}(k) \end{bmatrix} = \begin{bmatrix} \tilde{a}_d^{DD}(k-1) \\ a_d^{DD}(k) \\ a_r^{DD}(k) \\ a_u^{DD}(k) \end{bmatrix}.$$

We are now ready to combine (3.30) and (3.32). Partition $F(k) = \begin{bmatrix} F_1(k) \mid F_2(k) \end{bmatrix}$ compatibly with (3.34) so that in (3.32)

$$F(k)a^{DD}(k) = F_1(k)a_r^{DD}(k) + F_2(k)a_u^{DD}(k).$$

In (3.30), use (3.35) to write

$$S_{k-1}a^{DD}(k-1) = S_{k-1}\Pi\,\Pi^T a^{DD}(k-1) = \begin{bmatrix} S_{k-1}\Pi_1 \mid S_{k-1}\Pi_2 \end{bmatrix} \begin{bmatrix} a_d^{DD}(k) \\ a_r^{DD}(k) \end{bmatrix},$$

and similarly for $\tilde{S}_{k-1}^{(2)}a^{DD}(k-1)$. Then stacking (3.30) on top of (3.32) gives (see (3.36) and (3.34))

$$(3.37) \quad \begin{bmatrix} w_{k-1}^{(1)} \\ \hat{w}_{k-1} \\ w_k \end{bmatrix} = \begin{bmatrix} \tilde{S}_{k-1}^{(1)} & \tilde{S}_{k-1}^{(2)}\Pi_1 & \tilde{S}_{k-1}^{(2)}\Pi_2 & 0 \\ 0 & S_{k-1}\Pi_1 & S_{k-1}\Pi_2 & 0 \\ 0 & 0 & W_kF_1(k) & W_kF_2(k) \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \tilde{a}_d^{DD}(k-1) \\ a_d^{DD}(k) \end{bmatrix} = \tilde{a}_d^{DD}(k) \\ \begin{bmatrix} a_r^{DD}(k) \\ a_u^{DD}(k) \end{bmatrix} = a^{DD}(k) \end{bmatrix}$$
$$+ \begin{bmatrix} v_{k-1}^{(1)} \\ \hat{v}_{k-1} \\ W_k^T P_2(k)^T v_k \end{bmatrix}.$$

In order to have a unique LS estimate for $\tilde{a}^{DD}(k) = \begin{bmatrix} \tilde{a}_d^{DD}(k) \\ a^{DD}(k) \end{bmatrix}$ at epoch k (see (3.36)), the number of equations should be greater than or equal to the number of (nonnoise) unknowns in (3.37). Suppose m_{k-1} satellites (including the reference satellite) are visible at epoch $k-1$, m_d satellites go down between epochs $k-1$ and k, and m_u satellites come up between epochs $k-1$ and k. Then E_k has $m_{k-1} - m_d + m_u$ rows, so W_k is $(m_{k-1} - m_d + m_u - 4) \times (m_{k-1} - m_d + m_u - 1)$ (cf. (3.16)). Notice that in (3.30) (or the first two parts of (3.37)) there are the same number of equations as unknowns, and (3.32) (or the bottom part of (3.37)) adds $m_{k-1} - m_d + m_u - 4$ more equations and m_u more unknowns ($a_u^{DD}(k)$). So we require

$$m_{k-1} - m_d + m_u - 4 \geq m_u, \qquad \text{which is just } m_{k-1} \geq m_d + 4.$$

This means that if more than $m_{k-1} - 4$ satellites go down between epochs $k-1$ and k we will have a rank deficient system, and we will have to wait until it becomes a full rank system at some later epoch before we can obtain unique estimates again. But we nearly always have either $m_d = 0$ or $m_d = 1$, giving the requirements $m_{k-1} \geq 4$ or $m_{k-1} \geq 5$, and these are both met by our assumption that at least 5 satellites are visible at any epoch. Under the condition $m_{k-1} \geq m_d + 4$, we can reasonably assume that the coefficient matrix in (3.37) has full column rank. To solve the LS problem, we compute the following orthogonal transformations, the new versions of (3.22):

$$\tilde{T}_k^T \begin{bmatrix} \tilde{S}_{k-1}^{(1)} & \tilde{S}_{k-1}^{(2)}\Pi_1 & \tilde{S}_{k-1}^{(2)}\Pi_2 & 0 \\ 0 & S_{k-1}\Pi_1 & S_{k-1}\Pi_2 & 0 \\ 0 & 0 & W_k F_1(k) & W_k F_2(k) \end{bmatrix} = \begin{bmatrix} \tilde{S}_k^{(1)} & \tilde{S}_k^{(2)} \\ 0 & S_k \\ 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} \tilde{S}_k \\ 0 \end{bmatrix},$$

$$\tilde{T}_k^T \begin{bmatrix} w_{k-1}^{(1)} \\ \hat{w}_{k-1} \\ w_k \end{bmatrix} = \begin{bmatrix} w_k^{(1)} \\ \hat{w}_k \\ \bar{\bar{w}}_k \end{bmatrix} \equiv \begin{bmatrix} \tilde{w}_k \\ \bar{\bar{w}}_k \end{bmatrix},$$

where both $\tilde{S}_k^{(1)}$ and S_k are nonsingular upper triangular. This has completed the update and provided the equivalents of (3.28) and (3.30) for epoch k. Since $\tilde{S}_{k-1}^{(1)}$ is already nonsingular upper triangular, we see that the above computation did not touch the top part of (3.37), which corresponds to the satellites going down between epochs 1 and $k-1$. We can now compute the LS estimates $a_k^{DD}(k)$ of $a^{DD}(k)$ and $\tilde{a}_{dk}^{DD}(k)$ of $\tilde{a}_d^{DD}(k)$ by solving

$$S_k a_k^{DD}(k) = \hat{w}_k, \qquad \tilde{S}_k^{(1)} \tilde{a}_{dk}^{DD}(k) = w_k^{(1)} - \tilde{S}_k^{(2)} a_k^{DD}(k).$$

Notice that if no satellites rise or set between epochs $k-1$ and k, $a_d^{DD}(k)$ and $a_u^{DD}(k)$ will have no elements, so in (3.35) and (3.34) we take $\Pi = \Pi_2 = I$, giving $a^{DD}(k-1) = a_r^{DD}(k) = a^{DD}(k)$; thus, in (3.37) the second and fourth blocks of columns of the coefficient matrix disappear and $F_1(k) = F(k)$, and the transformations revert to (3.22).

After this, we can use (3.31) and (3.33) to get $x_{k|k}$ and $x_{j|k}$ for $j = 1, \ldots, k-1$ by solving triangular systems

$$R_k x_{k|k} = u_k - U_k F(k) a_k^{DD}(k), \qquad R_j x_{j|k} = u_j - U_j F(j) a_k^{DD}(j),$$

where $a_k^{DD}(j)$ is the LS estimate of $a^{DD}(j)$ at epoch k. This will be found by transforming $\tilde{a}_k^{DD}(k)$ to obtain the corresponding vector whose reference satellite is the

one used at epoch j (see (3.39) to see how this is done) and extracting the relevant elements. Keeping track of this is part of the bookkeeping we must do.

Case 2. The reference satellite (satellite 1, say) at epoch $k-1$ goes down between epochs $k-1$ and k. Without loss of generality we assume that satellite 2 is visible at epoch $k-1$, remains at epoch k, and is used as the reference satellite at epoch k. Suppose at epoch k we obtain (cf. (3.32))

$$(3.38) \quad w_k = W_k F(k) a^{DD}(k) + W_k P_2(k)^T v_k, \qquad W_k P_2(k)^T v_k \sim \mathcal{N}(0, \sigma^2 I),$$

where $a^{DD}(k)$ is the DDIA vector of visible satellites with satellite 2 as the reference satellite. At the end of epoch $k-1$ we have (3.28), where $\tilde{a}^{DD}(k-1) \in \mathcal{R}^{m-1}$, say, uses satellite 1 as the reference satellite, and without loss of generality we assume

$$\tilde{a}^{DD}(k-1) = [\alpha^1 - \alpha^2, \alpha^1 - \alpha^3, \dots, \alpha^1 - \alpha^m]^T.$$

Define the corresponding vector $\bar{a}^{DD}(k-1)$ with satellite 2 as the reference satellite, along with the matrix K,

$$\bar{a}^{DD}(k-1) \equiv [\alpha^2 - \alpha^1, \alpha^2 - \alpha^3, \dots, \alpha^2 - \alpha^m]^T, \qquad K \equiv \begin{bmatrix} -1 & 0 \\ -e & I_{m-2} \end{bmatrix}.$$

Then it is easy to verify that

$$(3.39) \qquad\qquad K^2 = I_{m-1}, \qquad \bar{a}^{DD}(k-1) = K\tilde{a}^{DD}(k-1).$$

This indicates that we can easily transform a DDIA vector with one satellite as the reference satellite to a DDIA vector with another satellite as the reference satellite. Define $\bar{S}_{k-1} \equiv \tilde{S}_{k-1} K$. Then from (3.28) we have

$$\tilde{w}_{k-1} = \tilde{S}_{k-1}\tilde{a}^{DD}(k-1) + \tilde{v}_{k-1} = \tilde{S}_{k-1}KK\tilde{a}^{DD}(k-1) + \tilde{v}_{k-1} = \bar{S}_{k-1}\bar{a}^{DD}(k-1) + \tilde{v}_{k-1}.$$

We could apply an orthogonal transformation to the left of this to triangularize $\bar{S}_{k-1}$, giving essentially the same situation as in Case 1, so we can now use that approach.

Finally we would like to point out the computation of covariance matrices in section 3.4 can also be extended to the case where there are satellites rising and/or setting. But for brevity, we prefer not to do this here.

4. Simulations. To demonstrate the performance of our algorithm, we give some computer simulation results. All our computations were performed in MATLAB 5.2 on a Pentium III running Windows 2000. The 24 GPS satellite constellation data in YUMA ephemeris format for the week of June 30–July 6, 1998, is used in the simulations. The roving receiver is assumed to be on board an aircraft circling horizontally with center directly above the reference station (stationary receiver) at the constant speed of $100\,\mathrm{m/s}$. The baseline is $1\,\mathrm{km}$. Each single differenced carrier phase measurement is corrupted by a random normally distributed noise with zero mean and standard deviation $\sigma = 0.2\,\mathrm{cm}$, except for Figure 4.4, where $\sigma = \sqrt{2}\,\mathrm{cm}$. The $\sigma = 0.2\,\mathrm{cm}$ noises were chosen to roughly match what appear to be the noise levels used in [21], while the higher noise level shows the effect of the maximum carrier phase noise level suggested by Table 1.1. The receiver clock offset relative to GPS time is modeled by white noise input to a second order Markov process based on [17, p. 417]. The time interval between two consecutive epochs is 1 second.

The typical results for 7 satellites with initial errors of 5 meters, 100 meters, 1 kilometer, and 1 kilometer with more noise, are shown in Figures 4.1–4.4, respectively.

Notice how much longer convergence takes in Figure 4.4 than in the much less noisy but otherwise equivalent simulation in Figure 4.3. The typical results for 6 and 5 satellites with an initial error of 100 meters are shown in Figures 4.5 and 4.6. From these figures we see the new algorithm performed better than the van Graas and Lee algorithm in terms of position accuracy at the beginning. But as time progressed the two algorithms did not always have such significant differences, particularly when 7 satellites were simulated. However, in the 5 satellite example, at 2,000 seconds the new algorithm still gave results that were about four times as accurate as those obtained by the van Graas and Lee algorithm. The new algorithm is also more efficient than the van Graas and Lee algorithm. Our MATLAB simulations show that for each epoch the number of floating point operations of the former is less than half that of the latter. Figure 4.5 shows that using 6 satellites takes more time to get submeter accuracy than using 7 satellites. Figure 4.6 shows that with 5 satellites it takes much longer still.

Position errors for 7 satellites with different initial errors (Figure 4.1–Figure 4.4)
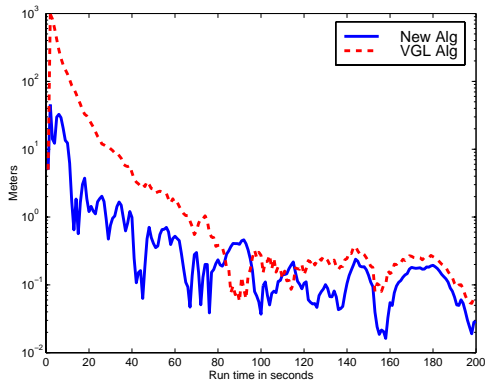
FIG. 4.1. *5 m initial error, noise* $\sigma = 0.002\,m$.   FIG. 4.2. *100 m initial error, noise* $\sigma = 0.002\,m$.
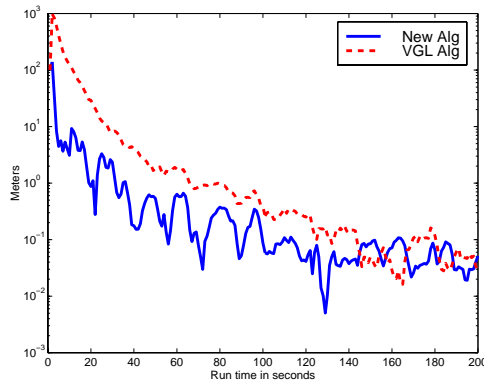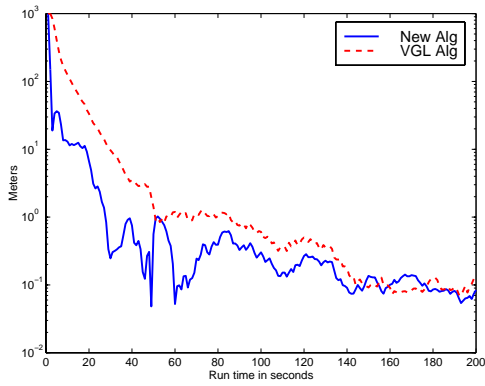

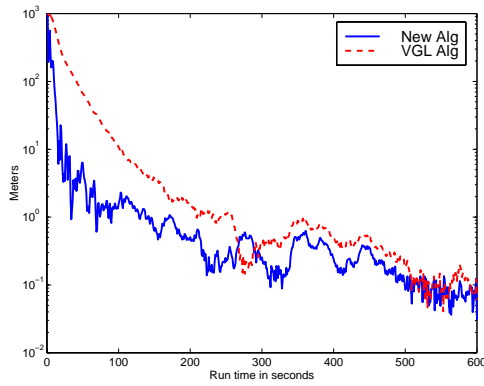FIG. 4.3. *1 km initial error, noise* $\sigma = 0.002\,m$.   FIG. 4.4. *1 km initial error, noise* $\sigma = 0.014\,m$.

Since the noise was randomly generated we obtained different results when we repeated each run, so the figures here were chosen to be typical examples. But in a few of our many runs the van Graas and Lee algorithm converged as well as and occasionally better than ours, and when we increased the single-signal noise standard

Position errors for 6 and 5 satellites with 100 m initial error (Figure 4.5–Figure 4.6)
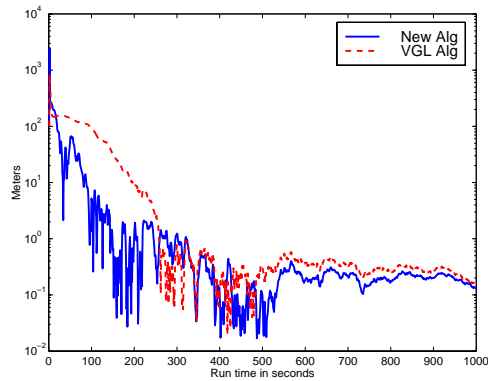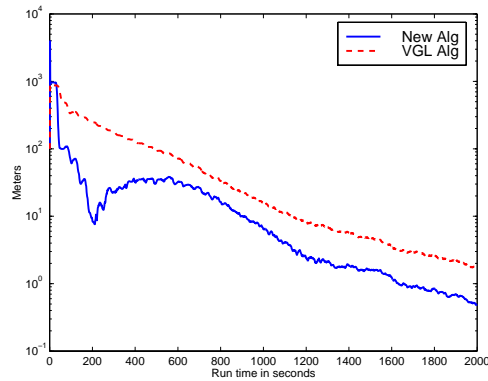

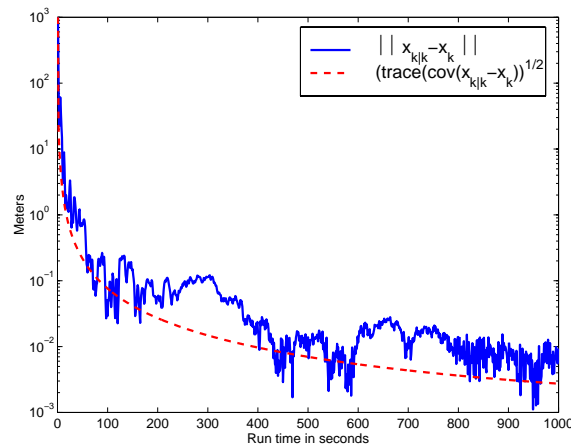FIG. 4.5. *6 satellites, noise* $\sigma = 0.002\,m$.   FIG. 4.6. *5 satellites, noise* $\sigma = 0.002\,m$.

FIG. 4.7. *Position errors and variances,* 7 *satellites,* 1 *km initial error, noise* $\sigma = 0.002\,m$.

deviation to higher than 1 cm (see Table 1.1), the general superior convergence of our algorithm was less marked. Also it can be seen that the van Graas and Lee algorithm provides a nice smoothing effect. However, it would seem preferable to have the general faster convergence and lower computation time of our algorithm for these problems, so the computational performance described in this section nicely complements the other advantages of the material presented here.

In order to test our covariance approximations from section 3.4, we use the fact that for a linear system $\mathcal{E}\{\|\hat{z} - z\|^2\} = \text{trace}(\text{cov}\{\hat{z} - z\})$ (see (3.3)). Thus we can test the trace of our approximate covariance matrix by plotting the position (or baseline) errors $\|x_{k|k} - x_k\|$ and $(\text{trace}(\text{cov}\{x_{k|k} - x_k\})^{1/2}$ for our approximation. This is done in Figure 4.7, where 7 satellites are used for positioning with 1 km initial error. From this figure, we see our approximate variances (we are considering only the diagonal of the covariance matrix) appear to give reasonable indicators of the position errors. Note that this is a different run with different noise of the same standard deviation as that in Figure 4.3.

5. Summary and remarks. We have presented a recursive LS approach for carrier phase based GPS positioning. From the simulation results we see that the recursive LS approach is effective and that for the examples treated here it is not necessary to use what van Graas and Lee [21] called the complementary Kalman filter. However, in cases where we have additional precise information about the dynamics of the roving receiver, the ideas here could also be used to handle the state equations.

Our algorithm is numerically reliable since we use numerically stable orthogonal transformations. It is also efficient, since it takes full advantage of the structure of the problem. A nice contribution of our approach is that some of the drawbacks of double differencing are avoided, and yet the double differenced integer ambiguities are still available.

One remark we would like to make is that when we apply our algorithm to real data, we need to incorporate an algorithm for cycle slip detection and correction. Cycle slip is the change in integer ambiguities mainly caused by obstructions of the satellite signal due to trees, buildings, bridges, mountains, etc. For details about cycle slip detection and correction, see, for example, [9].

REFERENCES

[1] J. Ashjaee, *GPS Doppler processing for precise positioning in dynamic applications*, Navigation, Journal of The Institute of Navigation, 32 (1985), pp. 370–385.

[2] J. D. Bossler, C. C. Goad, and P. L. Bender, *Using the Global Positioning System (GPS) for geodetic positioning*, Bull. Géodésique, 54 (1980), pp. 553–563.

[3] X.-W. Chang and C. C. Paige, *An algorithm for combined code and carrier phase based GPS positioning*, BIT, to appear.

[4] C. C. Counselman III and S. A. Courevitch, *Miniature interferometer terminals for earth surveying: Ambiguity and multipath with Global Positioning System*, IEEE Trans. on Geoscience and Remote Sensing, 19 (1981), pp. 244–252.

[5] C. C. Goad and B. W. Remondi, *Initial relative positioning results using the Global Positioning System*, Bull. Géodésique, 58 (1984), pp. 193–210.

[6] A. G. Evans, B. R. Hermann, and P. J. Fell, *Global Positioning System sensitivity experiment*, Navigation, Journal of The Institute of Navigation, 28 (1981), pp. 77–84.

[7] J. A. Farrell and M. Barth, *The Global Positioning System & Inertial Navigation*, McGraw-Hill, New York, 1998.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[9] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *GPS Theory and Practice*, 5th ed., Springer, New York, 2001.

[10] P. Y. C. Hwang, *Kinematic GPS for differencing positioning: Resolving integer ambiguities on the fly*, Navigation, Journal of The Institute of Navigation, 38 (1991), pp. 1–16.

[11] GPS, papers published in Navigation, Vol. V, Institute of Navigation, Alexandria, VA, 1998.

[12] E. D. Kaplan, *Understanding GPS, Principles and Applications*, Artech House Publisher, Boston, 1996.

[13] P. V. W. Loomis, *A kinematic GPS double differencing algorithm*, in Proceedings of the 5th International Geodetic Symposium on Satellite Positioning, New Mexico State University, Las Cruces, NM, 1989, pp. 611–620.

[14] W. Michakon, H. Hua, and D. Cox, *GPS carrier-phase RAIM*, in Proceedings of Institute of Navigation GPS-98, Nashville, TN, 1998, pp. 1975–1984.

[15] P. MISRA AND P. ENGE, *Global Positioning System, Signals, Measurements, and Performance*, Ganga-Jamuna Press, Lincoln, MA, 2001.

[16] C. C. PAIGE AND M. A. SAUNDERS, *Least squares estimation of discrete linear dynamic system using orthogonal transformations*, SIAM J. Numer. Anal., 14 (1977), pp. 180–193.

[17] B. W. PARKINSON, J. J. SPILKER, P. AXELRAD, AND P. ENGE, EDS., *Global Positioning System: Theory and Applications,* Vol. 1, American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1996.

[18] B. W. PARKINSON, J. J. SPILKER, P. AXELRAD, AND P. ENGE, EDS., *Global Positioning System: Theory and Applications,* Vol. 2, American Institute of Aeronautics and Astronautics, Washington, D.C., 1996.

[19] G. STRANG AND K. BORRE, *Linear Algebra, Geodesy, and GPS*, Wellesley-Cambridge Press, Wellesley, MA, 1997.

[20] P. J. TEUNISSEN AND A. KLEUSBERG, EDS., *GPS for Geodesy*, 2nd ed., Springer, Berlin, 1998.

[21] F. VAN GRAAS AND S.-W. LEE, *High-accuracy differential positioning for satellite-based systems without using code-phase measurements*, Navigation, Journal of The Institute of Navigation, 42 (1995), pp. 605–618.