ABSTRACT
Recent flight tests of the Integrity Beacon Landing System (IBLS) have demonstrated the feasibility of using GPS for Category III precision landing. To achieve these results, an airborne architecture that provided position solutions in real-time was developed. Centimeter-level positioning accuracy was achieved using a single-frequency receiver without using integer search techniques. This capability distinguishes the Integrity Beacon Landing System from other proposed kinematic GPS landing systems.

At the heart of the real-time architecture is a cycle ambiguity estimator. This estimator makes use of all available information to arrive at floating estimates of the integer biases associated with the GPS carrier phase measurements. The uncertainty in these estimates is stored in a covariance matrix. The estimates are updated in several ways:
1) After each carrier phase measurement epoch, the satellite phase measurements are transformed to a reduced measurement set that is only a function of the integers. The position and clock error terms are eliminated from the measurement, thereby partitioning the estimation of the constant integers from the estimation of the changing position.
2) Code DGPS measurements are incorporated into the estimates, achieving an effect similar to carrier-smoothed-code.
3) Phase measurements from the Integrity Beacons (low-power pseudolite transmitters placed under the approach path) provide a high-accuracy, high integrity update to the estimator.

New satellites are added to the estimate and lost satellites are removed from the estimate with ease. Given redundant satellites, the estimator will converge toward the cycle ambiguities using satellite motion. With 7 satellites, the integer estimates typically converge to the cycle level in 15 minutes. During the pseudolite overpass, the estimates converge to the centimeter level in a matter of seconds. Receiver Autonomous Integrity Monitoring (RAIM) is performed during the pseudolite overpass to verify the consistency of the satellite and pseudolite measurements. Additionally, in all phases of flight RAIM is performed before each integer update to verify that the update is consistent with the existing integer estimates. Despite the flexibility of this architecture, it is straightforward to implement. The details of this implementation are presented.

1.0 INTRODUCTION

Stanford University has been developing IBLS as a means of augmenting GPS to provide the performance required to achieve Category III specifications [1-4]. Figure 1 shows the IBLS concept. The system consists of a differential reference station located at or near the airport tower and two pseudolites located on the ground beneath the approach path. These pseudolites are called Integrity Beacons. The low-power Integrity Beacon signal is detectable only within the low power broadcast "bubble" shown in the figure. Initially, the aircraft navigates using traditional code-based differential GPS. As the aircraft passes through the bubble, the rapid geometry change allows the cycle ambiguities associated with the carrier phase to be estimated with high accuracy and integrity. These estimates provide high accuracy position solution for the remainder of the approach, even after the aircraft exits the bubble. A patent application has been submitted for the Integrity Beacon Landing System.

Presented at ION 51st Annual Meeting, Colorado Springs, Colorado, June 5-7, 1995

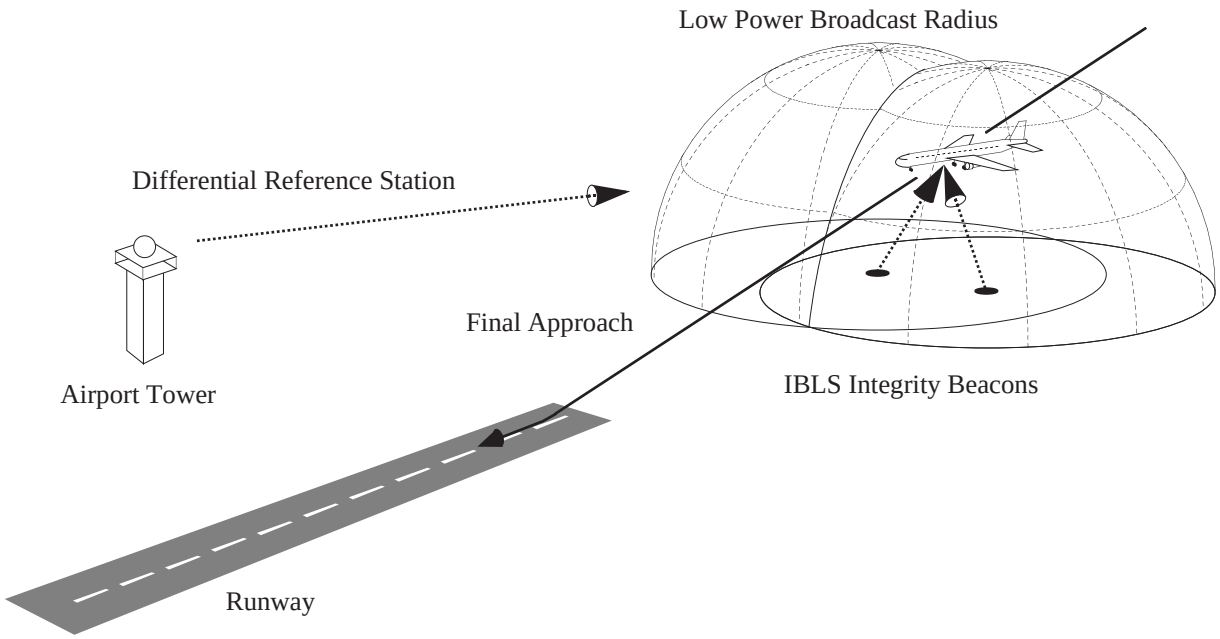

Figure 1: Integrity Beacon Landing System Diagram

For early flight tests of IBLS [1], system evaluation was performed entirely in post-processing. The next step in the development of the IBLS test platform was to add a data link and allow position solutions to be calculated in flight [2]. However, the system still had several limitations:

- Positions were only available at the data uplink frequency and were delayed by the data link latency.
- Positions were only calculated when the integers were known; no positions were available before the bubble.

Since the ultimate goal of the IBLS flight tests was to demonstrate automatic approaches and landings, these limitations had to be overcome. The current version of the IBLS architecture does overcome these limitations and Category III feasibility demonstrations were presented in [3,4].

2.0 AIRBORNE ARCHITECTURE

The current version of the IBLS architecture is shown in Figure 2. The inputs to the airborne software come from the aircraft GPS receiver, the data link, and an attitude source (attitude is required to account for the lever arm from the top GPS antenna to the pseudolite receive antenna). The time-tagged data from each of these inputs is stored in ring buffers. The newest data on the aircraft receiver ring buffer is used immediately to obtain a position solution. Because the reference phase is delayed and available only at the uplink frequency, its value must be predicted from the reference phase history in the data link ring buffer. Also, an estimate of the integers is required to perform a carrier phase position solution. This estimate is continuously updated using aligned data from the ring buffers. That is, when a timetag is found that is common to all of the ring buffers, the synchronous data is used to improve the estimate of the integers. This process is described in more detail in the "Integer Estimation" section.

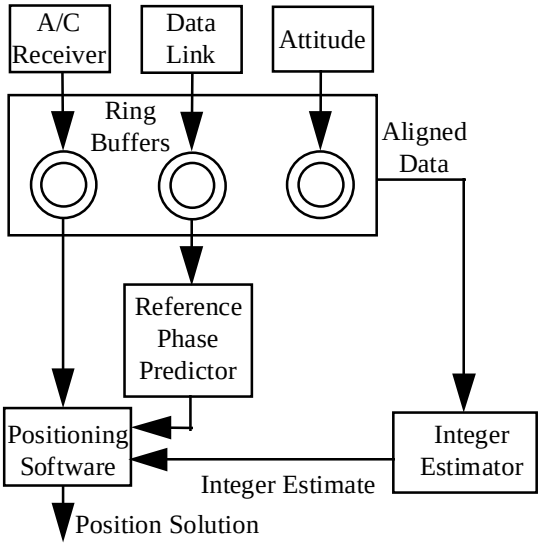

Figure 2: IBLS Airborne Architecture

2.1 REFERENCE PHASE PREDICTOR

Instead of waiting for the data link to send up the carrier phase measured at the reference station, the reference phase can be predicted based on the history of the

reference phase. Because the reference phase is relatively slowly changing (satellite motion and SA dominate the dynamics of the reference signal), it can be accurately predicted several seconds into the future. The goal of the reference phase predictor was to perform this prediction in a simple yet robust manner. Although it is possible to predict more than several seconds in advance, a data latency or data dropout of that magnitude would most likely be unacceptable for the precision landing application. Therefore, the requirements for the predictor were derived to be:

- Predict the reference phase several seconds with error on the order of centimeters.
- Be robustness to data dropouts.
- Be easy to implement.
- Predict phase at arbitrary timetags (i.e. - not just at reference sample times).

Although several types of predictors were evaluated, a least-squares quadratic predictor was chosen because it was simple, yet it performed quite well. A quadratic function of time is fit to the previous *m* reference phases (*m*>=3). Given the timetag of an aircraft receiver measurement, the corresponding reference phase can be predicted. The performance of the predictor is given in Table 1. The table was generated by finding the difference between the predicted phase and the actual phase for different values of *m* and different data latencies. The data uplink frequency was one hertz. The predictor errors are quite acceptable for data latencies less than 5 seconds and *m* between five and seven. As expected, the prediction error starts increasing for larger values of *m* because older data is weighted the same as more recent data. A weighted least-squares predictor which decreased the weights with the age of the data was also evaluated. The performance improved slightly, but the improvement was not worth the additional complexity.

Table 1: Reference Phase Predictor Error Versus Data Latency and *m*.

1 σ Error (cm)	1 sec	2 sec	3 sec	5 sec	10 sec
m=3	1.09	2.58	4.68	10.74	36.70
m=4	0.75	1.53	2.55	5.57	18.50
m=5	0.63	1.12	1.82	3.86	13.08
m=6	0.56	0.95	1.52	3.24	11.32
m=7	0.55	0.93	1.47	3.15	11.14
m=8	0.56	0.95	1.52	3.24	11.43

2.2 INTEGER ESTIMATION

To use the differential carrier phase to perform position solutions, an estimate of the integer cycle ambiguities is required. The carrier phase measurement equation can be written:

$$\phi = \begin{bmatrix} G & I \end{bmatrix} \begin{pmatrix} x \\ \tau \\ N \end{pmatrix} + \delta\phi \qquad (1)$$

where ϕ (ns×1) is the single-difference carrier phase measurement (expressed in L1 wavelengths), G (ns×4) is the traditional GPS geometry matrix, I (ns×ns) is the identity matrix, x (3×1) is the position, τ (scalar) is the differential receiver clock bias, N (ns×1) are the integers, $\delta\phi$ (ns×1) are the measurement errors (including reference phase prediction errors), and ns is the number of satellites.

If an integer estimate, $\hat{N}$, and the corresponding covariance, P_N, are available, this equation may be rewritten:

$$\phi - \hat{N} = G\begin{pmatrix} x \\ \tau \end{pmatrix} + \left(\tilde{N} + \delta\phi\right)$$

where $\tilde{N}$ is the error in the integer estimate. Assuming the measurement errors are uncorrelated with variance σ^2, a weighted least-squares position/clock estimate may be calculated:

$$\begin{pmatrix} \hat{x} \\ \hat{\tau} \end{pmatrix} = \left[G^T R_e^{-1} G\right]^{-1} G^T R_e^{-1}\left(\phi - \hat{N}\right)$$

with covariance: $P_{x\tau} = \left[G^T R_e^{-1} G\right]^{-1}$

where $R_e \equiv \sigma^2 I + P_N$.

It is therefore convenient to keep a running estimate of the integers. An integer estimator was developed with the following goals in mind:

- Eliminate the need for mode switching from code to carrier differential GPS after the bubble pass.
- Maintain an estimate and covariance of the integers.
- Update estimates using all available information, including integrity beacon results.
- Bring satellites on and off line gracefully as they are acquired and lost.
- Allow for continuous RAIM.
- Be easy to implement.

Implementing the estimator as described here, these primary goals are met, and the following advantages are also achieved:

- Integer estimates converge from satellite motion.
- Flexible architecture allows simple extensions described later.

2.2.1 ESTIMATE INITIALIZATION

When the program is first started, the integer estimates are initialized using the differential code phase measurements:

$$\phi_{code} = [G]\begin{pmatrix} x \\ \tau \end{pmatrix} + \delta\phi_{code}$$

$$\phi - \phi_{code} = N + \delta\phi - \delta\phi_{code}$$

$$\hat{N} = \phi - \phi_{code}$$

$$P_N = \left(\sigma^2 + \sigma_{code}{}^2\right) I \cong \sigma_{code}{}^2 I$$

where ϕ_{code} is the code phase *expressed in L1 cycles.*

2.2.2 INTEGER MEASUREMENT UPDATES

After the integer estimates are initialized in this manner, they are refined from a variety of sources. It is important to note that only the aligned measurements from Figure 2 are used to update the integer estimates (error in the reference phase predictor does not corrupt the estimates). In all cases, the estimate update is performed by casting the new information into the following form:

$$z = HN + \nu \qquad (2)$$
$$E\left[\nu\nu^T\right] = R$$

In this form, the measurement is used in a minimum variance measurement update:

$$K = P_N{}^- H^T \left[HP_N{}^- H^T + R\right]^{-1}$$

$$\hat{N}^+ = \hat{N}^- + K\left(z - H\hat{N}^-\right) \qquad (3)$$

$$P_N{}^+ = [I - KH]P_N{}^-$$

where the '-' and '+' in the superscripts indicate before and after the measurement update. No process update is necessary, because the states being estimated are constants. The measurement update process is shown in Figure 3.

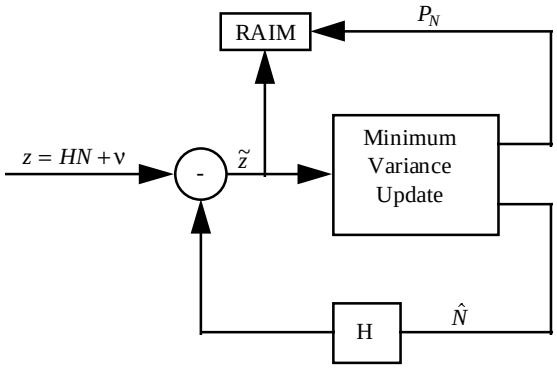

Figure 3: Measurement Update Process

CARRIER PHASE UPDATE

If there are more than four satellites, each differential carrier phase measurement can be cast into the form of equation 2. Pre-multiplying equation 1 by L, an orthonormal basis for the left null space of G:

$$L\phi = LN + L\delta\phi \qquad (4)$$
$$z = L\phi$$
$$H = L$$
$$R = \sigma^2 LL^T = \sigma^2 I$$

CODE PHASE UPDATE

The code phase measurements can be used to update the integers by subtracting them from the carrier phase measurements in equation 1. This equation is already in the desired form.

$$\phi - \phi_{code} = N + \delta\phi - \delta\phi_{code}$$
$$z = \phi - \phi_{code}$$
$$H = I$$
$$R = \left(\sigma^2 + \sigma_{code}{}^2\right) I$$

Updating the carrier phase cycle ambiguity estimates using code phase measurements is similar to carrier smoothed code. Both techniques make use of the advantages of each measurement. The carrier phase has low noise but an integer bias; the code phase has high noise but no bias. One technique uses code phase measurements smoothed by the carrier, while the other uses the carrier phase gravitated toward the code.

INTEGRITY BEACON UPDATE

The rapid geometry change the occurs during an integrity beacon overpass provides another update to the integer estimator. The output of the integrity beacon processing software is a high accuracy estimate of the individual satellite integer differences along with the corresponding covariances. The integers themselves are unobservable, but only the integer differences affect the position solutions. Any common bias affects only the clock solution. The details of calculating the integer differences from the integrity beacon measurements are given in the "Integrity Beacon Processing" section. As with the other integer updates, the new information can be written in the form of equation 2:

$$z = HN + \nu$$
$$R = E\left[\nu\nu^T\right]$$
$$H = \begin{bmatrix} -1 & \\ \vdots & I \\ -1 & \end{bmatrix}$$

H (ns-1×ns) reflects the fact only integer differences are output; the integer of an arbitrary satellite is subtracted

from all of the rest. The *R* used in the update equations is simply the integer difference covariance output by the integrity beacon processing code. *R* is a function of the overpass geometry.

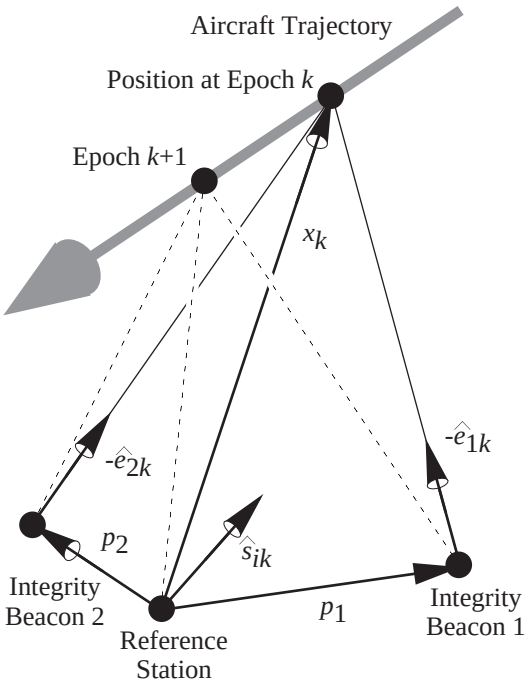

Figure 4: Integrity Beacon Geometry

2.2.3 INTEGRITY BEACON PROCESSING

The geometry change that occurs as an aircraft flies over a pair of integrity beacons provides information similar to several hours of satellite motion in a matter of seconds. However, due to the nonlinear nature of this geometry change (*G* is a function of *x* for the integrity beacons), this information must be processed separately. Expanding the single difference carrier phase measurement equation for SV *i* at epoch *k*, we obtain

$$\phi_{ik} = -\hat{s}_{ik}^T x_k + \tau_k + N_i^s + \varepsilon_{ik}^s ,$$

where ϕ_{ik} is the single-differenced phase, $\hat{s}_{ik}$ is the line-of-sight vector, x_k is the displacement vector from the differential station to the aircraft, τ_k is the difference in the aircraft and reference receiver clock biases, N_i^s is the integer cycle ambiguity, and ε_{ik}^s is the measurement error due to multipath and receiver noise. Similarly for pseudolite *j* at epoch *k*

$$\varphi_{jk} = \left|p_j - x_k\right| - \left|p_j\right| + \tau_k + N_j^p + \varepsilon_{jk}^p$$

where φ_{jk} is the single-differenced phase and p_j is the vector from the differential station to pseudolite *j* as shown in Figure 4.

Given an approximate trajectory $\bar{x}_k$ obtained from code-based differential GPS, the equations above can be expressed in terms of the deviation from the approximate trajectory: $\delta x_k \equiv x_k - \bar{x}_k$. Keeping first order terms only, the result is

$$\delta\phi_{ik} \equiv \phi_{ik} + \hat{s}_{ik}^T \bar{x}_k = -\hat{s}_{ik}^T \delta x_k + \tau_k + N_i^s + \varepsilon_{ik}^s$$

$$\begin{aligned}\delta\varphi_{jk} &\equiv \varphi_{jk} - \left|p_j - \bar{x}_k\right| + \left|p_j\right| \\ &= -\hat{e}_{jk}^T \delta x_k + \tau_k + N_j^p + \varepsilon_{jk}^p\end{aligned}$$

where $\hat{e}_{jk}^T \equiv \left(p_j - \bar{x}_k\right) \Big/ \left|p_j - \bar{x}_k\right|$. We now note that the value of one integer must be specified to due to the existence of the clock bias τ_k which is common to all measurements at epoch *k*. For simplicity, we choose $N_1^s = 0$. Defining $\delta\Phi_k$ to be the vector of *m* SV and two pseudolite measurements at epoch *k*

$$\delta\Phi_k \equiv \begin{bmatrix} \delta\phi_{1k} \\ \vdots \\ \delta\phi_{mk} \\ \delta\varphi_{1k} \\ \delta\varphi_{2k} \end{bmatrix}$$

and $\hat{S}_k$ as

$$\hat{S}_k \equiv \begin{bmatrix} \hat{s}_{1k}^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ \hat{s}_{mk}^{\mathrm{T}} & 1 \\ \hat{e}_{1k}^{\mathrm{T}} & 1 \\ \hat{e}_{2k}^{\mathrm{T}} & 1 \end{bmatrix}$$

we stack all *n* measurements collected during pseudolite overpass to obtain

$$\begin{bmatrix} \delta\Phi_1 \\ \vdots \\ \delta\Phi_k \\ \vdots \\ \delta\Phi_n \end{bmatrix} = \begin{bmatrix} \hat{S}_1 & 0 & \cdots & 0 & 0 & \bar{I} \\ 0 & \ddots & 0 & \ddots & 0 & \vdots \\ \vdots & \ddots & \hat{S}_k & \ddots & \vdots & \bar{I} \\ 0 & \cdots & 0 & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 0 & \hat{S}_n & \bar{I} \end{bmatrix} \begin{bmatrix} \delta x_1^* \\ \vdots \\ \delta x_k^* \\ \vdots \\ N^* \end{bmatrix} + \varepsilon$$

where

$$\delta x_k^* \equiv \begin{bmatrix} \delta x_k^T & \tau_k \end{bmatrix}^T,$$

$$N^* \equiv \begin{bmatrix} N_2^s & \cdots & N_m^s & N_1^p & N_2^p \end{bmatrix}^T,$$

and ε is the measurement error.

The least-squares solution to the above can be obtained efficiently by sparse matrix batch algorithms or equivalently by sequential forward-backward smoothing. Due to nonlinear nature of the problem, the approximate trajectory and observation matrix is improved by the computed estimate of δx_k, and the process above is repeated through convergence. The residuals of this solution provide a high level of integrity as described in [7]. The result of this process is a high accuracy, high integrity estimate of the integer differences. This estimate is used to update the overall integer estimate, as described in the "Integrity Beacon Update" section.

2.2.4 CONSTELLATION SWITCHES

As an aircraft banks, it may lose the signal from some satellites and acquire the signal from others. Satellites may also be acquired or lost as they pass the receiver elevation mask angle. It is therefore desirable to remove satellites and add satellites to the integer estimate. The covariance form of the estimator was chosen primarily because it makes this task quite simple. To remove a satellite, the corresponding state is removed. The element of the estimate and the row and column of the covariance are simply discarded. To bring a new satellite on-line, the integer is initialized using a the code phase measurement for that satellite:

$$\hat{N}^{new} = \phi^{new} - \phi_{code}^{new}$$

The variance for the new integer is set consistent with the code phase measurement noise; the cross covariance for the other integer states is set to zero:

$$E\left[\tilde{N}^{new}\tilde{N}^{new}\right] = \left(\sigma^2 + \sigma_{code}{}^2\right)$$

$$E\left[\tilde{N}^{new}\tilde{N}^{old}\right] = \bar{0}$$

After one carrier phase measurement update, the state estimate and covariance for the new satellite are consistent with the other integer estimates. This technique has proven to be an efficient method for handing-off satellite integer estimates.

2.2.5 RAIM

RAIM is performed before each integer update to verify that the new measurement is consistent with the existing integer estimates. If the measurement does not pass this check, the approach can be aborted. In some cases, the failure may be isolated. Isolation has not yet been implemented in the real-time software.

In preparation for an integer update, the new measurement is already in the form of equation 2. The difference between the expected measurement and the actual measurement is calculated:

$$r \equiv E[z] - z = E[HN + \nu] - HN - \nu$$

$$r = H\left(\hat{N} - N\right) - \nu = H\tilde{N} - \nu$$

This residual quantity, r, is a random vector with zero mean and covariance given by:

$$P_r = HP_N H^T + R$$

A measure of the consistency of the new measurement is the weighted residual:

$$w = r^T P_r^{-1} r$$

If this weighted residual is greater than some predetermined threshold, a RAIM alert is issued. The threshold is a function of the number of the dimension of *r* and the desired continuity.

2.2.6 OTHER FEATURES

SATELLITE MOTION

Loomis [5] and Hwang [6] pointed out that the differences between the integers are observable in a dynamic environment because of satellite motion. A welcome side-effect of this airborne architecture is that the integer differences will tend to converge using satellite motion. This convergence results from the carrier phase measurement updates described earlier. Satellite motion is automatically taken into account each time there is a carrier phase measurement update, because the *L* matrix in equation 4 changes with time. Although the implementation of the carrier phase updates are performed sequentially, the observability analysis is shown below for a batch solution.

As the satellite geometry changes, the L matrix in equation 4 also changes. Grouping an arbitrary satellite integer with the clock bias term, equation 1 may be rewritten:

$$\phi = \begin{bmatrix} G & \bar{I} \end{bmatrix} \begin{pmatrix} x \\ \tau + N_1 \\ N_2 - N_1 \\ \vdots \\ N_{ns} - N_1 \end{pmatrix} + \delta\phi$$

$$\bar{I} \equiv \begin{bmatrix} 0 & \cdots & 0 \\ & I & \end{bmatrix}$$

Pre-multiplying by the left null space of *G*:

$$z = L\phi = L\bar{I}N' + L\delta\phi$$

$$N' \equiv \begin{pmatrix} N_2 - N_1 \\ \vdots \\ N_{ns} - N_1 \end{pmatrix}$$

Several of these measurements can be stacked together:

$$z_* \equiv \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} = L_*\bar{I}N' + \begin{pmatrix} L_1\delta\phi_1 \\ \vdots \\ L_k\delta\phi_k \end{pmatrix} \qquad (5)$$

where: $L_* \equiv \begin{bmatrix} L_1 \\ \vdots \\ L_k \end{bmatrix}$

The reason that only integer differences are observable while the integers themselves are unobservable is that L_* never has rank greater than (ns-1). The columns of *L* always sum to zero, as can be seen by manipulating the definition of the left null space of *G*:

$$LG \equiv [0]$$

$$\begin{bmatrix} \vec{L}_{c_1} & \cdots & \vec{L}_{c_{ns}} \end{bmatrix} \begin{bmatrix} & 1 \\ G_x & \vdots \\ & 1 \end{bmatrix} = [0]$$

$$\Rightarrow \sum_{j=1}^{ns} \vec{L}_{c_j} = \vec{0}$$

Therefore, the sum of the columns of L_* are also constrained to sum to zero, and the maximum rank is (ns-1). However, only integer differences are required to solve for position. Given sufficient geometry change and enough redundant satellites, the matrix $L_*\bar{I}$ will have rank (ns-1) and equation 5 may be used to estimate N':

$$\hat{N}' = \left[\bar{I}^T {L_*}^T L_* \bar{I}\right]^{-1} \bar{I}^T {L_*}^T z_*$$

If the measurement samples are widely spaced enough in time, the noise will be uncorrelated and the estimate error covariance reduces to:

$$E\left[\tilde{N}'\tilde{N}'^T\right] = D\sigma^2$$

$$D \equiv \left[\bar{I}^T {L_*}^T L_* \bar{I}\right]^{-1}$$

The matrix *D* is similar to dilution of precision (DOP). The square root of the trace of *D* is analogous to PDOP and is referred to as NDOP. The quantity (NDOP×σ) approximates the one-sigma integer estimate error. A typical value of NDOP using seven or more satellites and two measurements separated by 15 minutes is 20. Assuming a carrier phase measurement error of 0.5 cm, the one-sigma integer estimate is 10 cm after 15 minutes of satellite motion. In contrast, NDOP for a 10 second integrity beacon overpass is about 2. The information provided by the integrity beacon clearly dwarfs that provided by satellite motion. However, satellite motion is a welcome complement to the integrity beacon overpass.

In the sequential implementation of the carrier phase measurement update, it is not necessary to group one satellite with the clock bias. Although only integer differences are observable from satellite motion, the integers themselves are initialized from code phase measurements. The implication is that one direction of the integer covariance will remain at its initial value (neglecting code phase measurement updates). Scaling problems could arise as the minimum eigenvalue decreases while the maximum eigenvalue remains the same. This issue does not present a practical limitation because adding a small amount of process noise to the covariance prevents the minimum eigenvalue from decreasing without bound.

STATIC SURVEY

For experimental purposes, it is often convenient to know the integers before the aircraft takes off. For this reason, a static survey mode was added to the estimator. When the user changes to this mode, the estimator assumes the aircraft is not moving. This static constraint allows the estimator to converge faster and with fewer satellites. When the static constraint is imposed, the integer state is augmented with the position. Breaking the geometry matrix into the satellite line-of-sight matrix, G_x, and the column of ones that multiply the clock bias, equation 1 can be rewritten:

$$\phi = \begin{bmatrix} & 1 & \\ G_x & \vdots & I \\ & 1 & \end{bmatrix} \begin{pmatrix} x \\ \tau \\ N \end{pmatrix} + \delta\phi$$

The clock bias is removed from this measurement by pre-multiplying by L_τ,

where $L_\tau \equiv \left(null\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\right)^T$

$$L_\tau \phi = \begin{bmatrix} L_\tau G_x & L_\tau \end{bmatrix} \begin{pmatrix} x \\ N \end{pmatrix} + L_\tau \delta\phi$$

This is of the form of equation 2, but now the state contains both position and integers. As before, the state is updated using equations 3 with:

$$z = L_\tau \phi$$

$$H = \begin{bmatrix} L_\tau G_x & L_\tau \end{bmatrix}$$

$$R = \sigma^2 L_\tau L_\tau^T$$

Convergence in static mode is typically two or three times faster than dynamic mode. Integer estimates are often within a cycle of their true value in less than five minutes. Before the aircraft starts moving, position is simply discarded from the state estimate.

UPDATES AT A KNOWN POSITION

If the user knows the aircraft's position, this position can be incorporated into the integer estimate. In static mode, this update is straightforward. The position "measurement" can be written in the form of equation 2:

$$z = \hat{x} = H \begin{pmatrix} x \\ N \end{pmatrix} + \tilde{x}$$

where $H = \begin{bmatrix} I & 0 \end{bmatrix}$

The "measurement noise" is the uncertainty in the position. The user enters the position and a covariance matrix representing this uncertainty. An example of when this feature is useful is when the aircraft parked at the tie-down location. Each time the aircraft is there, its vertical position is the same to within a few centimeters; the horizontal position may be different by a meter. This uncertainty can be accurately entered into the estimate. After leaving the tie-down, the vertical position error and covariance will remain small, while the horizontal error will slowly converge. This feature was used during a flight test discussed later.

3.0 SYSTEM PERFORMANCE

This real-time system has been tested extensively. In July of 1994, it was used to perform 49 autocoupled approaches of an FAA King Air [3]. The most impressive results came in October of 1994 when this system was used to perform 110 automatic landings of a United Airlines Boeing 737-300 [4]. The following sections describe experiments designed to exercise several aspects of the system.

3.1 24 HOUR POSITIONING TEST

The real-time system was set up in the lab; the reference station and aircraft receivers were connected to separate antennas on the roof. To accentuate multi-path errors, ground planes were not used. Due to the antenna gain pattern and cable loss, satellites were not acquired until they reached an elevation angle of about fifteen degrees. The system was initialized and data was collected for 24 hours. The first goal of this experiment was to demonstrate the integer estimator convergence using satellite motion. Although the baseline was static, the static survey mode of the estimator was not used. The second goal was to smoothly hand-off integer estimates for 24 hours.

Figure 5 plots the magnitude of the 3-D position error over the 24 hour period. The initial error was about three meters (off the vertical scale of the plot). Using satellite motion, the position error converged to the cycle level in about fifteen minutes. In less than an hour, it converged to less than ten centimeters where it stayed for the remainder of the test. After the first hour, the mean value of the magnitude of the position error was 2.2 cm.

Twenty-five different satellites were used during the 24 hours. Some satellites were brought on and off line several times while their signal strength was low. Satellite integer estimates were brought on-line a total of 192 times. These hand-offs were performed seamlessly as evidenced by the position error plot of Figure 5. After the estimator converged on the integers for one set of satellites, the position error remained small using entirely different sets of satellites. By the end of the test, the original satellites returned to their initial positions in the sky. The implication of this periodic geometry is that the satellites could be handed off indefinitely while retaining centimeter-level positioning accuracy.

3.2 AIRBORNE PERFORMANCE

To exercise the system in an airborne environment, a flight test was performed in a Piper Dakota. In contrast to the static test discussed earlier, the true position of the aircraft in flight is not known exactly. However, if a separate process knows the correct values of the rounded integers, a centimeter-level truth trajectory can be found. This trajectory can be compared with the trajectory estimate calculated using the integer estimator and reference phase predictor to evaluate the system performance.

To find the integer differences for the truth trajectory, the static survey mode of the estimator was used while the aircraft was at the tie-down. In about fifteen minutes, the estimator converged to within a half-cycle of the correct integer differences. The integer difference were rounded to the correct values. The measurement residual was monitored for another fifteen minutes to verify that they were correct. The flight test was performed when the satellite geometry was such that six satellites could be

continuously tracked, even when the aircraft banked. This ensured that the integer differences used for the truth trajectory were constant throughout the flight.

The real-time system was reset before the flight (the integer differences found from the static survey were only used to find the truth trajectory). The integers were initialized as usual, using differential code phase measurements. As mentioned earlier, the vertical position at the tie-down is well known. To demonstrate the "known position update" feature described earlier, this vertical information was incorporated into the estimate. After the position update, the vertical position was accurate to a few centimeters, while the horizontal position was wrong by several meters (as intended). During the flight, the horizontal position should converge toward the correct value, while the vertical position should remain accurate.

Six spacecraft were in view during the entire flight. However, to provide more of a challenge to the system, satellite outages were simulated in software. Every four minutes, one satellite was removed for a period of thirty seconds. During the 26 minute flight, each satellite was removed once. Therefore, each integer was taken off-line and brought back on-line.

The flight test consisted of three take-offs and landings. Figure 6 plots the horizontal and vertical position error and the 1σ horizontal position error bound. The times of the simulated satellite outages are marked with an "x" on the time axis. As expected, the vertical error remained small while the horizontal error converged.

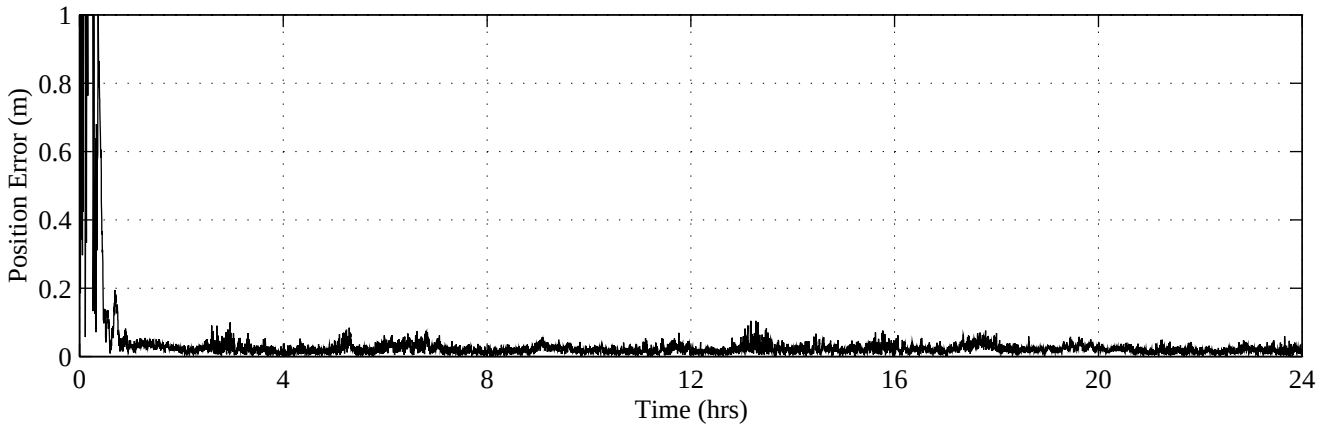

Figure 5: Position Error for 24 Positioning Test

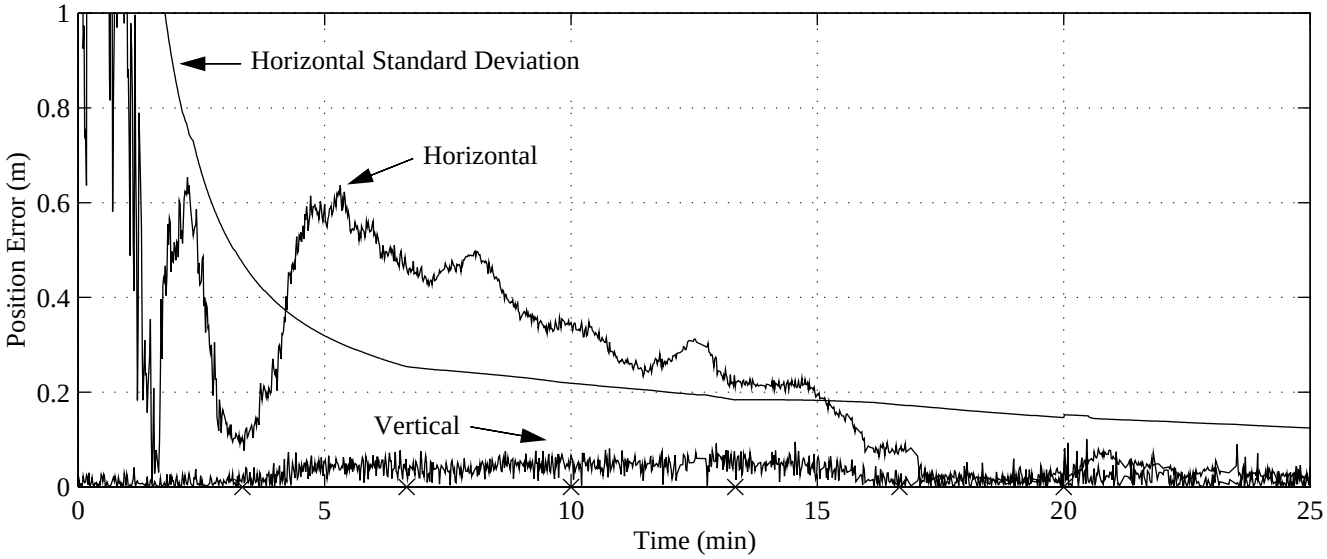

Figure 6: Aircraft Position Error

4.0 CONCLUSIONS

The real-time system developed to demonstrate the Integrity Beacon Landing System was designed to be easy to implement yet provide the flexibility required of a research tool. Distinguishing features of this architecture include:

- Only the L1 carrier and C/A code are used.
- Kinematic position solutions are provided with minimal delay in all phases of flight. Integer estimates are continuously refined.
- Several layers of RAIM provide high-integrity. The Integrity Beacon information is particularly powerful in this respect.
- Cycle ambiguities are found without using integer searches.
- Integer estimates are smoothly handed off as satellites are acquired and lost.
- Additional information such as static constraints is easily incorporated into the system.

5.0 ACKNOWLEDGMENTS

The authors wish to thank several individuals and organizations who made this research possible. At Stanford, Arnaud Masson, Konstantin Gromov, Ping-Ya Ko, Jock Christie, Mike O'Connor, Andy Barrows, and Gabe Elkaim provided generous assistance. Trimble Navigation provided the GPS equipment used to conduct the experiments. This research was sponsored by the FAA.

6.0 REFERENCES

1. C.E. Cohen, B. Pervan, H.S. Cobb, D. Lawrence, J.D. Powell, B.W. Parkinson, "Real-Time Cycle Ambiguity Resolution using a Pseudolite for Precision Landing of Aircraft with GPS", DSNS '93, Amsterdam, The Netherlands, March 30 - April 2, 1993.

2. C.E. Cohen, B. Pervan, H.S. Cobb, D. Lawrence, J.D. Powell, B.W. Parkinson, "Real-Time Flight Test Evaluation of the GPS Marker Beacon Concept for Category III Kinematic GPS Precision Landing", ION GPS-93, Salt Lake City, UT, September, 1993.

3. C.E. Cohen, D. Lawrence, B. Pervan, H.S. Cobb, A. Barrows, J.D. Powell, B.W. Parkinson, "Flight Test Results of Autocoupled Approaches Using GPS and Integrity Beacons", ION GPS-94, Salt Lake City, UT, September, 1994.

4. C.E. Cohen et. al., "Automatic Landing of a 737 using GNSS Integrity Beacons", ISPA '95, Braunschweig, Germany, February, 1995.

5. P. Loomis, "A Kinematic Double-Differencing Algorithm", Proceedings of the Fifth International Geodetic Symposium on Satellite Positioning, New Mexico State University, Las Cruces, NM, March, 1989.

6. P. Hwang, "Kinematic GPS for Differential Positioning: Resolving Integer Ambiguities on the Fly", Navigation, Vol. 38, No. 1, Spring 1991, pp. 1-15.

7. B. Pervan, C.E. Cohen, D. Lawrence, H.S. Cobb, J.D. Powell, B.W. Parkinson, "Autonomous Integrity Monitoring for GPS-Based Precision Landing Using Ground-Based Integrity Beacon Pseudolites", ION GPS-94, Salt Lake City, UT, September, 1994.