HELI/SITAN: A TERRAIN REFERENCED NAVIGATION ALGORITHM FOR HELICOPTERS

Jeff Hollowell
Sandia National Laboratories[1]

ABSTRACT

Heli/SITAN is a Terrain Referenced Navigation (TRN) algorithm that utilizes radar altimeter ground clearance measurements in combination with a conventional navigation system and a stored digital terrain elevation map to accurately estimate a helicopter's position. Multiple Model Adaptive Estimation (MMAE) techniques are employed using a bank of single state Kalman filters to ensure that reliable position estimates are obtained even in the face of large initial position errors. A real-time implementation of the algorithm was tested aboard a U.S. Army UH-1 helicopter equipped with a Singer-Kearfott Doppler Velocity Sensor (DVS) and a Litton LR-80 strapdown Attitude and Heading Reference System (AHRS). The median radial error of the position fixes provided in real-time by this implementation was less than 50 m for a variety of mission profiles.

INTRODUCTION

The Heli/SITAN algorithm estimates the errors in a reference navigation system (INS, AHRS, etc.) by processing radar altimeter measurements, barometric altimeter measurements, and stored digital terrain elevation values with a bank of one state Kalman filters. Since Kalman filters are recursive, the altimeter measurements are incorporated into the current position estimate as they become available. This is in contrast to batch oriented processing techniques that have two distinct phases: a collection phase during which many measurements are gathered followed by a processing phase during which the collected measurements are processed to yield a position estimate. One major disadvantage of a batch oriented processing technique is that the time required to perform the processing phase is likely to introduce significant latency into the position estimate. Heli/SITAN position fixes are provided at discrete points in time when a fix decision rule is satisfied. Since no assumptions are made regarding the higher order structure of the reference navigation system's error processes, this algorithm is easily integrated with a wide variety of existing navigation systems.

[1]This work was funded by the U.S. Army AVionics Research And Development Activity (AVRADA) under project order 5E20112DA.

ALGORITHM DESIGN

Requirements

The Heli/SITAN algorithm is designed to operate with initial horizontal position errors of 926 m (0.5 nm) CEP and navigation systems with drift rates of less than two percent of distance travelled. It must function with aircraft speeds of up to 100 m/s (194 kts). Since helicopters tend to fly short distance missions at slow speeds, it is important that the algorithm provide position fixes as often as possible while retaining high immunity to false fixes. Ultimately, the algorithm must utilize Defense Mapping Agency (DMA) Digital LandMass System (DLMS) Level I Digital Terrain Elevation Data (DTED).

General Philosophy

Two techniques that have been used to provide position fixes in the face of large initial position errors are batch oriented techniques[1,2] and Multiple Model Adaptive Estimation (MMAE) techniques[3]. The MMAE technique is used in this algorithm because it processes measurements recursively as the measurements become available. Hence, the system's CPU is utilized more uniformly and position estimates are available without the latency resulting from the period of intense processing required in a batch oriented technique. The MMAE technique consists of a bank of Kalman filters and some method for producing a position estimate based on information from these Kalman filters. The next topic of discussion is the structure of the Kalman filters.

Previous SITAN algorithms used a bank of three state Kalman filters[3]. The three states consisted

CH2811-8/90/0000/0616 $1.00 © 1990 IEEE

of the reference navigation system's easting and northing errors, and a vertical channel bias. Conceptually, one can think of initializing the bank of filters such that they form a regular grid about the reference navigation system's position (figure 1a). As the aircraft flies, the bank processes radar altimeter and barometric altimeter measurements causing one or more of the filters to migrate to the true position (figure 1b). If a set of fix decision rules are satisfied, the position error estimates from the best candidate filter in the bank are added to the reference navigation system's position to form an estimate of the aircraft's current position, a position fix. If the filters move such that there are no filters near the true position (not very likely, but a possibility all the same) the likelihood of a filter migrating to the true position is small. This is because the migration of the three state filters depends upon linearizing the nonlinear terrain function within the 'neighborhood' of each filter. If the true position does not lie within any filter's 'neighborhood' then none of the filters migrate to it. In order to counteract this effect, the filters are reset to the regular grid configuration if the fix decision rules are not satisfied after processing 128 measurements.

The disadvantage of this approach is that these intervals of 128 measurements are arbitrarily aligned with respect to the terrain beneath the aircraft. This may result in missed position fixes. Consider the scenario illustrated in figure 2a. During the first interval of 128 measurements the aircraft flies over flat terrain followed by moderately rough terrain. The fix decision rules are not satisfied so the filters are reset resulting in the loss of all previously processed terrain information. During the second interval the aircraft flies over moderately rough terrain followed by flat terrain. Again the fix decision rules are not satisfied. No position fixes result from this scenario. Now consider the scenario depicted in figure 2b. The aircraft flies over flat terrain in the first interval. The fix decision rules are not satisfied so the filters are reset. Next, the aircraft flies over the moderately rough terrain which is entirely contained within the second interval. The terrain information is sufficient, the fix decision rules are satisfied, and a position fix results. The point is, the algorithm may behave differently over exactly the same trajectory and terrain if the measurement intervals are aligned differently. It is interesting to note that batch oriented techniques, applied randomly along a trajectory, suffer from the same problem since they typically gather measurements over some fixed length interval.

To avoid resetting the grid to a regular configuration, the Heli/SITAN algorithm uses a bank of one

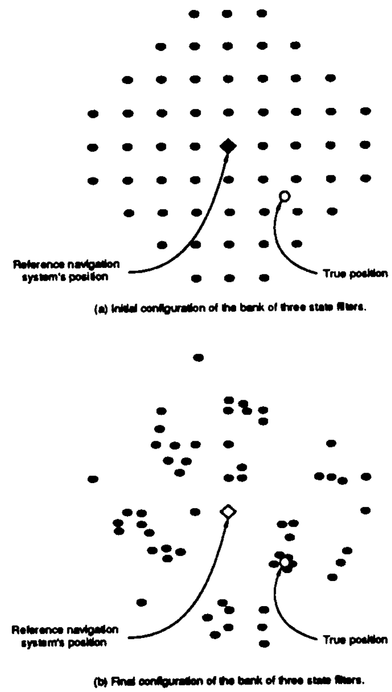

Figure 1: As the bank of three state filters processes measurements one or more filters migrate to the true position.

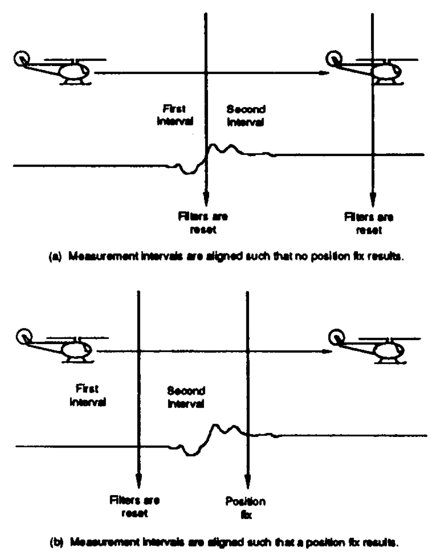

Figure 2: Position fixes depend upon alignment of measurement intervals with respect to terrain.

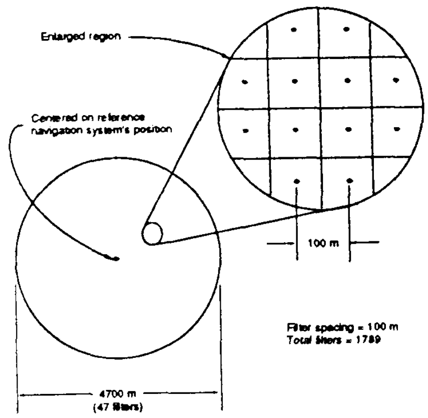

Figure 3: Filter bank configuration.

state Kalman filters. The single state modelled within each filter is a vertical channel bias. The easting and northing offsets for each filter relative to the reference navigation system's position are constant. Thus, each filter is pinned horizontally such that the bank of filters is always arranged in a regular grid. Since the filters are pinned in a regular configuration and they do not migrate horizontally there is no need to reset them.

Filter Bank Configuration

The size of the bank of one state Kalman filters is governed by the 926 m (0.5 nm) CEP requirement. Assuming that the initial position errors are Rayleigh distributed with a 50^{th} percentile value of 926 m the 99^{th} percentile value is 2387 m. The bank's area of coverage is roughly circular in shape with a radius of 2350 m (figure 3). This value is near the 99^{th} percentile value given above. Each filter's nearest neighbor lies 100 m directly to the east, west, north, and south. The 100 m filter spacing is approximately equal to the post spacing of DMA DLMS Level 1 DTED. Thus, each filter processes terrain elevation values that differ from those processed by neighboring filters. The total number of filters in the bank is 1789. The diameter of the bank is 47 filters in length.

One State Filter Equations

The single state modelled within each of the Kalman filters in the bank is a slowly varying vertical channel bias. This bias results from several sources: barometric altimeter bias, radar altimeter bias, and DTED bias. It is important to understand that none of these biases are individually observable, however, the collective bias that results from these individual biases is observable.

Here are definitions for many of the variables referred to throughout this paper:

$n \equiv$ Time indexing variable ($n = 0, 1, 2, \ldots$)
$j \equiv$ Filter indexing variable ($1 \leq j \leq 1789$)
$t_n \equiv$ Time (sec)
$T_n \equiv$ Sampling interval (sec) ($T_n = t_n - t_{n-1}$)
$x_n \equiv$ True vertical channel bias (m)
$\hat{x}_{jn} \equiv$ Estimated vertical channel bias (m)
$p_{jn} \equiv$ Estimation error variance (m^2)
$z_{jn} \equiv$ Measured vertical channel bias (m)
$z_{jn} = h_{jn} - s_n = h_{jn} - (baro_n - rad_n)$
where:
$h_{jn} \equiv$ Computed terrain elevation from DTED (m)
$s_n \equiv$ Sensed terrain elevation (m)
$s_n = baro_n - rad_n$
$baro_n \equiv$ Barometric altimeter measurement (m)
$rad_n \equiv$ Radar altimeter measurement (m)
$\mathrm{WRS}_{jn} \equiv$ Weighted Residual Squared
$\mathrm{SWRS}_{jn} \equiv$ Smoothed Weighted Residual Squared
$N \equiv$ Same SWRS_{min} filter update counter
$\alpha \equiv$ SWRS smoothing parameter
$\tau \equiv$ SWRS time constant

The discrete time process and measurement models for the vertical channel bias are assumed to be as follows:

Process Model:
$x_n = x_{n-1} + w_n$
where $w_n \equiv$ White process noise
$E\{w_n\} = 0;\ E\{w_n^2\} = qT_n;\ q = 4.0\ \mathrm{m}^2/\mathrm{sec}$

Measurement Model:
$z_n = x_n + v_n$
where $v_n \equiv$ White measurement noise
$E\{v_n\} = 0;\ E\{v_n^2\} = r;\ r = 20.0\ \mathrm{m}^2$

The process model shown above indicates that the quantity being estimated, the vertical channel bias, is modelled as integrated white noise or Brownian motion. The measurement model indicates that the measurements are simply the vertical channel bias itself contaminated with additive white noise. The value of q, which is an *a priori* assumption regarding the variability of the vertical channel bias, accounts for slow variations within all of the bias sources mentioned previously and for mismodelling that results from the true position of the aircraft not being coincident with the position of the nearest filter within the bank. This introduces a slowly varying bias whose magnitude depends upon the slope of the terrain between the nearest filter's position and the true position of the aircraft. The value of r, which is an *a priori* assumption regarding the noisiness of the measurements, accounts for unexpected objects that cause uncorrelated errors in the sensed terrain elevation such as buildings and trees, uncorrelated sensor errors, and uncorrelated map errors.

Although a slowly varying vertical channel bias is an appropriate model for this implementation, different models are required for other systems. For instance, if integrated vertical velocity is used as a vertical reference instead of barometric altitude, then the structure of each Kalman filter is modified to accommodate the different vertical channel error process. A state is added to each filter to model a vertical velocity error.

The algorithm for processing barometric altimeter measurements and radar altimeter measurements with a bank of one state Kalman filters is as follows:

1. Initialization:
   $p_{j0} = 3600.0\ \mathrm{m}^2,\ \hat{x}_{j0} = 0.0\mathrm{m},$
   $\mathrm{SWRS}_{j0} = 1.0,\ n = 1$

2. If the aircraft has travelled 100 m or more since the last propagate-update cycle:

   Propagate the variance and bias estimates:
   $$p_{jn} = p_{j(n-1)} + qT_n$$
   $$\hat{x}_{jn} = \hat{x}_{j(n-1)}$$

   Compute the Kalman gains:
   $$k_{jn} = \frac{p_{jn}}{p_{jn} + r}$$

   Compute the WRS and SWRS values:
   $$\mathrm{WRS}_{jn} = \frac{(z_{jn} - \hat{x}_{jn})^2}{p_{jn} + r}$$
   $$\mathrm{SWRS}_{jn} = \alpha\mathrm{WRS}_{jn} + (1 - \alpha)\mathrm{SWRS}_{j(n-1)}$$

   Update the variance and bias estimates:
   $$p_{jn} = (1.0 - k_{jn})p_{jn}$$
   $$\hat{x}_{jn} = \hat{x}_{jn} + k_{jn}(z_{jn} - \hat{x}_{jn})$$

   Check for a position fix:
   $$\frac{\mathrm{SWRS}_{mine} - \mathrm{SWRS}_{min}}{\mathrm{SWRS}_{min}} > \frac{18.0}{N}$$

   Increment the time index:
   $$n = n + 1$$

3. Go to step 2.

Keep in mind that the above algorithm is iterated for each filter within the bank, i.e for $1 \leq j \leq 1789$.

The computed terrain elevation values (h_{jn}) for a given filter are calculated as follows. First, the particular filter's position is calculated by adding its constant easting and northing offsets to the reference navigation system's easting and northing values. Next, bilinear interpolation is used to compute a terrain elevation from the four nearest posts of terrain elevation data.

All the filters within the bank process identical sensed terrain elevation values (s_n) but each filter processes different computed terrain elevation values (h_{jn}). This is because each filter occupies a different position within the bank and, therefore, each filter uses a different portion of the stored DTED to compute h_{jn}.

Notice that new measurements are not processed until the aircraft has travelled 100.0 m. This distance is called the update distance because each Kalman filter within the bank is updated after this distance is exceeded. The update distance is approximately equal to the post spacing of DMA DLMS Level I DTED.

Consider the scenario where filter k is nearest to the true position of the aircraft. Then the difference between the terrain elevation values as computed from the DTED at this filter's position (h_{kn}) and the sensed terrain elevation values (s_n) is nearly a bias. Thus, this filter is the best match for the *a priori* bias process model. The other filters within the bank will have differences between their computed terrain elevation values (h_{jn}) and the sensed elevation values (s_n) that do not fit the bias model. This is because they are positioned over DTED that differs from the terrain that the aircraft is flying over. The next section addresses the issue of how to decide which filter within the bank is best matching the *a priori* bias model.

Computation of the SWRS values

A quantity known as the Smoothed Weighted Residual Squared (SWRS) is computed for each filter in the bank. The SWRS provides some measure of how well each filter is matching the *a priori* vertical channel bias model: the lower the SWRS value, the better the match. To understand how the SWRS value is computed, it is necessary to define some other quantities. Each Kalman filter within the bank has a quantity known as the residual for each measurement that it processes. The residual is defined as follows:

$$residual_{jn} = z_{jn} - \hat{x}_{jn}$$

Consider a Kalman filter, call it filter k, within the bank. Kalman filter k is processing data that is well modelled by the *a priori* process and measurement models. Such a filter is said to be well tuned. The residual sequence from Kalman filter k is a white sequence with the following properties:

$$E\{residual_{kn}\} = 0.0$$

$$E\{residual_{kn}^2\} = p_{kn} + r$$

Next, a quantity known as the Weighted Residual Squared (WRS) is defined.

$$\mathrm{WRS}_{jn} = \frac{residual_{jn}^2}{p_{jn} + r}$$

Once again, consider Kalman filter k. If $p_{kn} + r$ is indeed the variance of the residuals then the WRS_{kn} values are the squares of normally distributed unit variance zero mean random variables. The sum of the WRS_{kn} values yields a quantity which is χ^2 distributed. The expected value of WRS_{kn} is 1.0. This property was used to select the fix decision structure of previous SITAN implementations.

Previous SITAN implementations used a bank of three state Kalman filters. The easting and northing errors that were modelled within each filter allowed the filter that was nearest the true position to track drift that might occur within the reference navigation system during the time interval over which radar altimeter and barometric altimeter measurements were processed. Since the single state filters in the Heli/SITAN algorithm do not model easting and northing errors, there must be some other provision for dealing with reference navigation system drift.

The Heli/SITAN algorithm indirectly accounts for drift of the reference navigation system by weighting recent residuals heavier than past residuals. In this way, the algorithm is better able to determine the aircraft's current true position, rather than the aircraft's position at some time in the past. A recursive filtering algorithm is used to apply exponentially decaying weights to the sequence of WRS values. Here is the way the Smoothed Weighted Residual Squared (SWRS) values are computed:

$$\mathrm{SWRS}_{jn} = \alpha \mathrm{WRS}_{jn} + (1 - \alpha)\mathrm{SWRS}_{j(n-1)}$$

$$\mathrm{SWRS}_{j0} = 1.0, \; 0.0 < \alpha < 1.0$$

This technique is attractive because it does not require that a large number of WRS values be stored. This is especially important when one considers that this value must be computed for each of the 1789 filters within the bank. Notice that the SWRS value is initialized to 1.0, the expected value of the WRS for a well tuned Kalman filter. The most recent WRS value is weighted by the constant α, the next most recent is weighted by the constant $\alpha(1-\alpha)$, the one prior to that is weighted by the constant $\alpha(1-\alpha)^2$, and so forth. The parameter α is chosen to maintain a time constant that is based on worst case assumptions about the reference navigation system's drift performance.

Examine figure 4. As the aircraft flies, the reference navigation system drifts off the true position. Since the bank of one state Kalman filters is centered on the reference navigation system's position, the true position of the aircraft drifts through the bank of filters. For reasons that will become apparent later, the value of α is related to how far the aircraft can fly while the true position drifts across a three filter by three filter square area. This three filter by three filter square area is called the exclusion region. Assume the true position drifts along a linear path and that this path can enter the exclusion region at any point with any angle. Figure 5 shows the exclusion region and several examples of paths that the true position might follow as it drifts across the exclusion region. The segment of each path that lies within the exclusion region has length l_i. The average distance that the true position will remain within the exclusion region is given by:

$$E\{l_i\} = 258.0 \text{ m}$$

Assuming a worst case short term reference navigation system performance of three percent of distance travelled, the aircraft will travel

$$\frac{258.0 \text{ m}}{0.03} = 8600.0 \text{ m}$$

while the true position is within the exclusion region. This corresponds to

$$\frac{8600.0 \text{ m}}{100.0 \text{ m / update}} = 86 \text{ updates}$$

The time constant of the SWRS smoothing equation in updates is given by

$$\tau \simeq \frac{1}{\alpha} \text{ updates}$$

So let five time constants equal the average number of updates that the true position remains within the exclusion region.

$$5\tau = 86 \text{ updates} \rightarrow \alpha = 0.058, \; \tau = 17.2 \text{ updates}$$

A valid question is "Why was the smoothing time constant computed based on the time it takes the

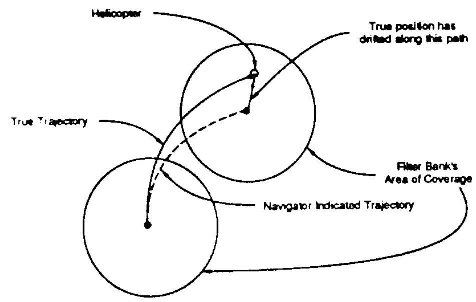

Figure 4: As the reference navigation system drifts, the aircraft's true position moves through the bank of filters.

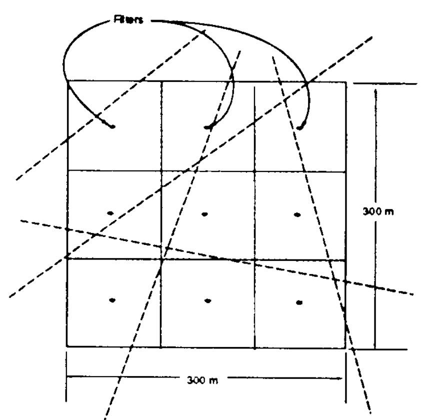

Figure 5: Sample paths that the true position might follow as it drifts across the exclusion region.

true position to drift across a three filter by three filter exclusion region instead of a single filter exclusion region ?" The answer is that there is a lower bound on the time constant. If the time constant falls below a certain value, then the filters are computing SWRS values that are based on too few measurements. However, a single filter exclusion region is appropriate for reference navigation systems that have worst case drift rates on the order of one percent of distance travelled.

Some simulations were performed using flight data from a 1 nm/hr INS. The time constant (τ) was computed based on a single filter exclusion region instead of a three filter by three filter exclusion region. The smaller drift rate in combination with the smaller size of the exclusion region resulted in a time constant that was very close to the value shown above.

Computing a Position Estimate

A position estimate is computed after each new measurement is processed by using the SWRS values from some of the filters in the bank. As mentioned before, the lower the SWRS value, the better the filter is matching the *a priori* vertical channel bias and measurement models. Thus, one might assume that the filter with the lowest SWRS value, the SWRS_{min} filter, is the best candidate for estimating the aircraft's current position. However, when the reference navigation system's drift rate is high, the filter that is nearest to the true position is often adjacent to the SWRS_{min} filter. This is because the time constant for computing the SWRS value was chosen based on a three filter by three filter exclusion region. Thus, it is expected that the position of the SWRS_{min} filter will lag the true position of the aircraft when the reference navigation system is drifting at a high rate. In order to combat this effect, the position estimate is computed based on the SWRS values of nine filters arranged in a three filter by three filter square centered on the SWRS_{min} filter. First, a Normalized Weighted Residual Squared (NWRS) value is computed for each of the nine filters in the exclusion region.

$$\text{NWRS}_i = \frac{\text{SWRS}_i}{\text{SWRS}_{min}}, \ 1 \le i \le 9$$

This is done because oftentimes the SWRS_{min} value is something other than its expected value of 1.0 due to mismodelling of the measurement noise variance (r). Next, a probability value is computed for each of the nine filters.

$$psum = \sum_{i=1}^{9} exp\{-\frac{1}{2}\text{NWRS}_i\}$$

$$p_i = \frac{exp\{-\frac{1}{2}\text{NWRS}_i\}}{psum}, \; 1 \leq i \leq 9$$

Finally, a position estimate is computed by weighting the positions of each of the nine filters by their associated probability values (p_i).

$$easting = \sum_{i=1}^{9} easting_i * p_i$$

$$northing = \sum_{i=1}^{9} northing_i * p_i$$

This method of formulating the position estimate is essentially that outlined in D.T. Magill's classic paper on MMAE[4] with the assumption that each of the nine filters have equal *a priori* probabilities and that they all have the same residual variances.

Fix Decision Rule

Although a position estimate is computed after each new measurement is processed, this position estimate is only valid when a fix decision rule is satisfied. The main purpose of the fix decision rule is to check for filters in one and only one particular part of the bank that are matching the *a priori* vertical channel bias model much better than all of the other filters.

In order to gain a better understanding of which fix decision rules performed best, many different rules were tested using Monte Carlo simulations with recorded flight data and DMA DLMS Level 1 DTED. A curve from decision theory known as the Receiver Operating Characteristic (ROC) curve was used to evaluate the performance of each of the different decision rules in terms of probability of fixing versus probability of false fixing.

Of the decision rules that were tested, the best decision rule in terms of maximizing the probability of fixing with no false fixes over the Monte Carlo simulations was:

$$\frac{\text{SWRS}_{min*} - \text{SWRS}_{min}}{\text{SWRS}_{min}} > T$$

The SWRS_{min} value is the lowest SWRS value within the entire bank of filters. The SWRS_{min*} value is computed by finding the minimum SWRS value which belongs to a filter outside of a three filter by three filter exclusion region centered on the SWRS_{min} filter. T is a constant threshold. The numerator on the left side gives some measure of how much better the SWRS_{min} filter is matching the vertical channel bias model than the next best filter that is a significant distance away. The denominator is used for normalization of mismodelled measurement noise. Since current methods do not provide exact knowledge of the location of trees and buildings, the *a priori* value chosen for the measurement noise variance (r) is, at best, a compromise. Thus, the SWRS_{min} value often deviates from its expected value of 1.0 to values as low as 0.5 and as high as 2.0. When the SWRS_{min} value is higher than 1.0 the assumed measurement noise variance (r) is too low for both the SWRS_{min} filter and the SWRS_{min*} filter. Thus, the normalization resulting from the division by SWRS_{min} requires the difference between the two to be proportionally higher. When the SWRS_{min} value is lower than 1.0 the assumed measurement noise variance (r) is too high for both the SWRS_{min} filter and the SWRS_{min*} filter. Thus, the normalization resulting from the division by SWRS_{min} requires the difference between the two to be proportionally lower.

The next best decision rule was:

$$N > T$$

where N is the number of updates that the same filter or an adjacent filter is chosen as the SWRS_{min} filter and T is a constant threshold. The reason that N is not reset when an adjacent filter assumes the SWRS_{min} value is that one expects this to happen as the reference navigation system drifts and the true position wanders through the bank of filters. N is reset to one when the SWRS_{min} filter jumps to a filter that is not adjacent to the last SWRS_{min} filter N.

The fix decision rule used here is actually a hybrid that was formed by combining the two fix decision rules listed above.

$$\frac{\text{SWRS}_{min*} - \text{SWRS}_{min}}{\text{SWRS}_{min}} > \frac{18.0}{N}$$

At the outset, N is set to one and the rule behaves like the first rule. As the same filter or an adjacent filter begins to have the SWRS_{min} value during consecutive updates, the threshold on the right hand side is decreased until either the difference on the left side is sufficient to satisfy it or the SWRS_{min} value is assumed by a filter that is not adjacent resulting in N being reset to a value of one.

This decision rule gave no false fixes during 100 Monte Carlo runs. Each of the runs was 250 updates in duration and used recorded flight data and DMA DLMS Level 1 DTED. A false fix was defined as any position fix with a radial error greater than 212.0 m.

Failure Detection

As mentioned before, the expected value for SWRS_{min} is 1.0. If SWRS_{min} is inordinately large, this indicates that no filters in the bank are matching the vertical channel bias model. Several different conditions could cause this: initial reference navigation system position errors that are larger than the radius of the filter bank (2350 m), radar altimeter failures, and barometric altimeter failures. If the $SWRS_{min}$ value exceeds 9.0 for 10 consecutive updates the pilot is informed that a Heli/SITAN failure occurred and instructed to check the radar altimeter, the barometric altimeter, and update the reference navigation system.

Recentering the Bank

Initially, the bank of one state filters is centered on the reference navigation system's position estimate. A Heli/SITAN failure results if the true position drifts outside the bank. In order to avoid this, the filter bank is recentered when position fixes result in the outer annulus of the bank. If the distance between the current position fix and the center of the bank is greater than three fourths of the bank's radius for two consecutive position fixes then the bank is recentered. This is accomplished by adding fixed offsets to the reference navigation system's position such that the center of the bank is near the true position and reinitializing the algorithm.

REAL-TIME IMPLEMENTATION

A real-time implementation of the Heli/SITAN algorithm was coded in the 'C' programming language. A SANDAC V avionics computer[5] was selected as the hardware platform. The SANDAC V is a reconfigurable flight computer that consists of one or more processing modules and a variety of special purpose memory and I/O modules. Each processing module has a Motorola 68020 CPU, a Motorola 68881 floating point coprocessor, and 128k bytes of RAM. The CPU and coprocessor run at a clock speed of 16 MHz. The SANDAC V that was used for flight testing had two processor modules, an extended memory module, and a 1553 I/O module. One of the processor modules was used for I/O, data logging, data display, and user interface functions. The other module was dedicated to the Heli/SITAN algorithm. The Heli/SITAN algorithm took less than 64k bytes of RAM and had an iteration time of approximately 0.8 s. Thus, assuming an update distance of 100.0 m, this implementation accommodates maximum aircraft velocities of 100.0 m / 0.8 s = 125.0 m/s. For aircraft velocities in excess of 125.0 m/s, the algorithm continues to function with an update distance that is correspondingly larger than 100.0 m. All arithmetic for the algorithm was done in double precision. The 100 m post spaced DTED required for this series of tests was stored in 131 k bytes of memory on the extended memory board. The SANDAC V obtained data from the reference navigation system using the military's standard 1553 bus protocol.

The UTM easting and northing values provided by the Litton LR-80 AHRS had a resolution of 10 m. The barometric altimeter values had a resolution of 10 ft. These values were mixed with integrated velocity data using simple Kalman filters to provide position data with a finer resolution.

FLIGHT TESTING

Equipment

The real-time implementation of the Heli/SITAN algorithm was tested using the following equipment:

UH-1 Helicopter
Litton LR-80 strapdown AHRS (ASN-143)
Singer-Kearfott DVS (ASN-137)
Rosemount 542-AY Barometric Altimeter
Honeywell APN-209 Radar Altimeter
SANDAC V avionics computer
Motorola Eagle Mini-Ranger GPS receiver

Scoring Method

A Motorola Eagle Mini-Ranger GPS receiver provided scoring in real-time on board the aircraft. This four channel C/A code receiver provides positions accurate to less than 25 m Spherical Error Probable (SEP) when operating autonomously. Another Eagle GPS receiver recorded data at a surveyed site during the flight tests. This allowed post-flight differential correction of some of the aircraft's scoring data. Not all of the data could be differentially corrected because oftentimes the on board GPS receiver could not track the same set of four satellites being tracked by the receiver located on the ground. Differentially corrected GPS receiver data is believed to provide position data accurate to less than 5 m SEP.

Test Area

The test area is just south of Harrisburg, PA near Carlisle, PA. Personnel from the U.S. Army AVionics Research And Development Activity (AVRADA) have extensive experience conducting helicopter test missions over this area. The DTED for the test area covers a 16 km by 42 km rectangle and has a post spacing of 100 m. The DTED was produced from cartographic source material and is not DMA DLMS Level I data. The standard deviation of slopes (σ_s) values range from .01 to .27 as computed over 16 post by 16 post sub-blocks of DTED. The smoother portions of the terrain are occupied by farm fields, farm buildings, and occasional stands of trees that are used by the farmers as wind breaks. The rougher portions of the terrain are heavily wooded.

Mission Profiles

Four different mission profiles were flown over the test area: baseline, low level, contour, and Nap Of the Earth (NOE). The baseline missions were constant speed and constant altitude Above Ground Level (AGL). They were designed to uncover any altitude or speed dependencies in algorithm performance. The low level, contour, and NOE mission profiles were designed to emulate U.S. Army missions.

Results

Figure 6 shows the radial errors of the LR-80 AHRS and the position fixes provided by the Heli/SITAN algorithm. For this particular test, an 1800 m radial error was intentionally introduced into the LR-80 during alignment. The time scale on the graph has been adjusted so that time 0 represents the time when the helicopter's skids left the ground. The nominal aircraft velocity was 45 m/s (87 kts). The Heli/SITAN algorithm provided the first position fix after 140 s of flight. During this 140 s period, the aircraft travelled 5.1 km and the bank of one state filters was updated with altimeter measurements 51 times. The reference navigation system's radial error curve is set to 1500 m during time periods when scoring dropouts occurred. Position fixes that occurred during scoring dropouts are shown with no radial error. As expected, the algorithm provides fixes more often over rougher terrain than over smoother terrain. This is because of the higher terrain signal to measurement noise ratio. However, the first position fix occurred over terrain with a standard deviation of slopes value (σ_s) of about two percent. Thus, position fixes do occur over smooth terrain but less frequently. Figure 7 shows a histogram of the scored position fix radial errors from the same flight. The mean, median, and maximum radial errors in this histogram are typical of those obtained for all of the mission profiles listed above.

CONCLUSION

A recursive Terrain Referenced Navigation (TRN) algorithm called Heli/SITAN was developed and demonstrated over a variety of helicopter mission profiles. This algorithm provided discrete position fixes in real-time. The median radial error of the position fixes was less than 50 m with no false fixes. Although Heli/SITAN was demonstrated on a helicopter

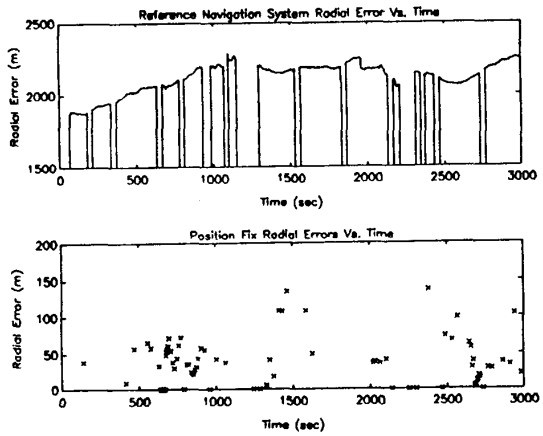

Figure 6: Radial errors for the reference navigation system and the Heli/SITAN position fixes.

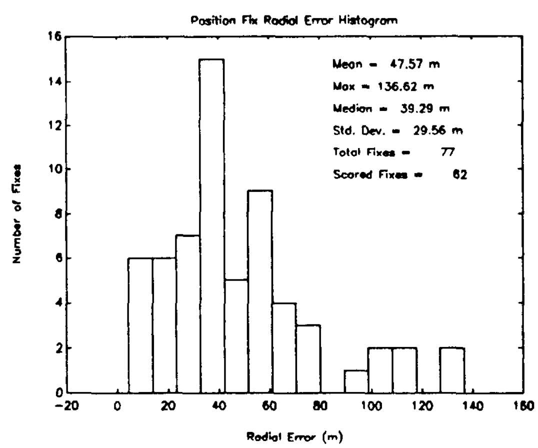

Figure 7: Histogram of the radial errors for Heli/SITAN's position fixes.

equipped with a Doppler aided AHRS, the extension to other types of aircraft and navigation systems is straightforward.

RECOMMENDATIONS

The current implementation of the Litton LR-80 AHRS uses a 14 state Kalman filter to mechanize the alignment and navigation functions[6]. This filter accepts velocity updates from the Singer-Kearfott DVS and position updates from the pilot or other navigation systems. Using Heli/SITAN position fixes as position updates for the LR-80 will remove large initial position errors and bound the LR-80's position errors during the remainder of the flight. Position updates will also allow the LR-80 to estimate its internal state vector more accurately resulting in reduced drift rates.

Although the Heli/SITAN algorithm was designed using recorded flight data and DMA DLMS Level I DTED, the real-time tests were not performed using DTED from this data base. Future tests should be conducted over Level I DTED. This will allow Heli/SITAN's real-time performance to be assessed using DTED that is currently available for large areas of the world.

ACKNOWLEDGEMENTS

The author wishes to thank Stan Sokolowski (AVRADA) for his help in planning and coordinating the flight tests, Jack Brady (AVRADA) for making all the hardware on board the aircraft work, and Ken Homb (Litton) for explaining the subtleties of the Litton LR-80 AHRS.

REFERENCES

1. W. Baker and R. Clem, "Terrain contour matching (TERCOM) primer," *Technical Report ASP-TR-77-61*, Aeronautical Systems Division, Wright-Patterson AFB, Ohio, Aug. 1977.

2. E.P. Bialecke and R.C. Lewis, "A digital terrain correlation system for tactical aircraft," *Proceedings of the IEEE/AIAA 5th Digital Avionics Systems Conference*, Oct. 31 - Nov. 3, 1983.

3. D.D. Boozer, M.K. Lau, and J.R. Fellerhoff, "The AFTI/F-16 terrain-aided navigation system," *Proceedings of the IEEE National Aerospace and Electronics Conference - NAECON*, pp. 351-357, May 20-24, 1985, Dayton, OH.

4. D.T. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Transactions on Automatic Control*, vol. AC-10, no. 4, pp. 434-439, Oct. 1965.

5. C.R. Borgman and L.R. Dalton, "An embedded multiprocessor computer for proof-of-principle testing of exploratory systems concepts," *AIAA Computers in Aerospace VI Conference*, Oct. 7-9, 1987, Wakefield, MA.

6. D. Lottman, K. Homb, and A. Brown, "Kalman filter implementation in the Litton LR-80 AHRS," *Proceedings of the IEEE National Aerospace and Electronics Conference - NAECON*, pp. 406-413, May 21-25, 1984, Dayton, OH.