JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 6, November–December 1995

Terrain-Aided Navigation Using the Viterbi Algorithm

Russell Enns*
McDonnell Douglas Helicopter Company, Mesa, Arizona 85215
and
Darryl Morrell†
Arizona State University, Tempe, Arizona 85287-5706

Without some form of correction, an inertial navigation system's (INS's) positional error grows without bound over time. Several techniques have been used to reduce the INS's positional error. One of these techniques, terrain-aided navigation (TAN), has been widely researched over the last several decades with the research focusing on two methods: Sandia inertial terrain-aided navigation (SITAN), which is based on an extended Kalman filter (EKF), and terrain contour matching, which is based on correlation techniques. These methods are applied most successfully to fixed-wing aircraft flying over specific types of terrain; they perform poorly for low-velocity, highly maneuverable aircraft such as helicopters, especially when flying over flat or very rough terrain. In this paper, we introduce VATAN, a new TAN method based on the Viterbi algorithm (VA), and compare its performance to an implementation of SITAN based on a single EKF. Simulation results show that the VA in VATAN overcomes divergence problems associated with the EKF in SITAN and provides position estimates with smaller average-squared error. These results show that VATAN has the potential to provide good performance for low-velocity aircraft flying over all types of terrain.

Nomenclature

h_{alt}	= true altitude
h_{gnd}	= true ground clearance
$\hat{h}_{\text{gnd}}$	= measured ground clearance
$\hat{h}_{\text{INS}}$	= INS altitude
h_{terr}	= nominal terrain elevation
$\hat{h}_{\text{terr}}$	= measured terrain elevation
$L_k(\boldsymbol{x}_k)$	= VATAN metric
$p(\cdot)$	= probability density function
$S(\cdot)$	= VATAN survivor function
$\boldsymbol{x}$	= state (position) sequence
$\hat{\boldsymbol{x}}$	= estimated state (position) sequence
$\boldsymbol{x}_k$	= state (position) at time k
$\hat{\boldsymbol{x}}_k$	= estimated state (position) at time k
$\boldsymbol{x}^{\text{INS}}$	= INS measured state
z	= observation sequence
z_k	= observation at time k
σ_r^2	= variance of random variable r

I. Introduction

AIRCRAFT navigation is often performed with an inertial navigation system (INS). The INS calculates the aircraft's velocity and position by integrating the acceleration measured by the INS and adding the results to the initial velocity and position. However, INSs are susceptible to various errors, most notably slow drifts in the acceleration measurements due to Shuler oscillations.[1] These errors cause the aircraft's estimated INS position to significantly differ from the aircraft's true position, especially long after the INS has been initialized.

Several techniques for reducing INS's errors have been developed over the years. These include Doppler-aided, stellar-aided, radio-aided, and terrain-aided navigation techniques. Terrain-aided navigation (TAN) combines terrain profile information with the traditional INS information to provide more accurate estimates of the aircraft's position. Several TAN methods exist: terrain contour matching (TERCOM) and Sandia inertial terrain-aided navigation (SITAN) are the most researched.[2–6] These methods have been successfully applied to high-velocity aircraft that fly over terrain with certain advantageous characteristics. However, these methods do not generally perform well for low-velocity, highly maneuverable aircraft such as helicopters, especially when flying through canyons, over fairly flat terrain, or over very rough terrain.

This paper focuses on the development of a TAN method that performs well for all types of aircraft over wide variations in terrain characteristics. Success for such a TAN method requires that the method appropriately combine a model of the aircraft's navigation system error dynamics, the measurements from the aircraft's navigation equipment, and information from a terrain database. SITAN uses one or more extended Kalman filters (EKFs) based on terrain linearization to accomplish this, whereas TERCOM uses a unique terrain correlation method. This paper proposes a third TAN method, VATAN, based on the Viterbi algorithm (VA). The VA is a maximum a posteriori state sequence estimator that incorporates both system dynamics and observations into a simple algorithm. Unlike the EKF, the VA does not require linearization, and it is robust with respect to partially observable system models; thus, VATAN promises good performance for all aircraft over a wide range of terrain without being susceptible to the divergence problems encountered using the EKF.

In this paper we develop error dynamics and observation models, implement a VA state estimator for these models, and evaluate the resulting VATAN using simulated data. Its operational characteristics are compared to an implementation of SITAN using a single EKF; we show that the VATAN algorithm provides more accurate position estimates for both relatively flat and extremely rough terrain.

This paper is structured as follows. We first present the theoretical development of VATAN. We then discuss certain aspects of implementing VATAN. Next we present the simulation models used to produce the results presented. The findings are then summarized.

II. VA

The VA is a maximum a posteriori (MAP) estimator that estimates a sequence of system states from a sequence of observation values. The state sequence is denoted $\boldsymbol{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n)$; in the TAN application considered here, the vector $\boldsymbol{x}_k$ is the aircraft's

---

Received Aug. 29, 1994; revision received Feb. 17, 1995; accepted for publication Feb. 20, 1995.

*Member of Technical Staff, Building 530, MSB345, 5000 E. McDowell Road.

†Associate Professor, Telecommunication Research Center and Department of Electrical Engineering.

two-dimensional position at time t_k. The observation sequence is denoted $z = (z_1, \ldots, z_n)$ where, in our application, z_k is the measured terrain elevation at time t_k.

The VA is based on the assumption that the system dynamics are Markov; that is, the state at $k+1$ is conditionally independent, given the state at time k, of the state at any previous time. The dynamics are thus characterized by $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$, the conditional probability density of $\boldsymbol{x}_{k+1}$ given $\boldsymbol{x}_k$, which represents the probabilistic knowledge of the states' evolution in time.

The VA also requires that the observation at time k depend only on the state at time k and not on the state at any other time; this is equivalent to requiring that observation errors at different times be independent. The probabilistic knowledge of the observation given the state's value is represented by $p(z_k \mid \boldsymbol{x}_k)$.

The VA consists of the computation of a metric function L_k that is a measure of the likelihood of each state value being the true state at time k; L_k can be computed recursively using $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$ and $p(z_k \mid \boldsymbol{x}_k)$ as follows[7–9]:

$$
\begin{aligned}
L_k(\boldsymbol{x}_k) &= \max_{\boldsymbol{x}_{k-1},\ldots,\boldsymbol{x}_0} \left[\sum_{i=1}^{k} \ln p(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1}) + \ln p(\boldsymbol{x}_0) + \sum_{i=1}^{k} \ln p(z_i \mid \boldsymbol{x}_i) \right] \\
&= \ln p(z_k \mid \boldsymbol{x}_k) + \max_{\boldsymbol{x}_{k-1}} [\ln p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) + L_{k-1}(\boldsymbol{x}_{k-1})] \qquad (1)
\end{aligned}
$$

where the recursion in Eq. (1) is initialized with $L_0(\boldsymbol{x}_0) = \ln p(\boldsymbol{x}_0)$.

The optimal estimate $\hat{x}_k$ is that $\boldsymbol{x}_k$ for which L_k is maximum. The value $\boldsymbol{x}_{k-1}$ that maximizes Eq. (1) for each $\boldsymbol{x}_k$ is termed the survivor. Denoted $S_k(\boldsymbol{x}_k)$, it is used to generate MAP state sequence estimates. The MAP state sequence estimate $\hat{x} = (\hat{x}_k, \ldots, \hat{x}_0)$ can be generated via the following recursive procedure:

$$\hat{x}_k = \arg\max_{\boldsymbol{x}_k} L_k(\boldsymbol{x}_k)$$

$$\hat{x}_{k-1} = S_k(\hat{x}_k)$$

$$\hat{x}_{k-2} = S_{k-1}(\hat{x}_{k-1}) = S_{k-1}(S_k(\hat{x}_k))$$

and by induction,

$$\hat{x}_{k-j} = S_{k-j+1}(\hat{x}_{k-j+1}) = S_{k-j+1}(S_{k-j}(\cdots(S_k(\hat{x}_k)))) \qquad (2)$$

The recursion in Eq. (1) is a filter, providing state estimates based on the system dynamics and observations. For an observable linear system model with Gaussian noises, Eq. (1) is functionally equivalent to a Kalman filter and Eq. (2) is equivalent to a fixed interval smoother (e.g., a Rauch–Tung–Striebel smoother[10]). These equivalences suggest that the VA is a suitable replacement in applications that use Kalman filtering.

For the TAN problem, the VA has two significant advantages over the EKF:

1) The fact that the metric function is computed for all possible state values makes the VA much more robust than the EKF in situations where the observations do not strongly support a single estimate of the state value. Such is the case in TAN, for example, when the aircraft is flying over terrain that is nearly flat or that is very rough. In such cases, divergence of the EKF becomes a significant problem, necessitating the use of multiple filters and complex voting logic; the VA does not suffer from these divergence problems.

2) The system model is represented by the conditional probability densities $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$ and $p(z_k \mid \boldsymbol{x}_k)$, which can embody nonlinear relationships in the state evolution and in the relationship between states and observations; these nonlinear relationships are also represented directly in the VA. Thus, in the TAN problem, the nonlinear relationship between aircraft position and measured terrain elevation can be represented exactly with the VA but must be approximated for the EKF.

These advantages of the VA with respect to the EKF come at the price of a larger computational burden.

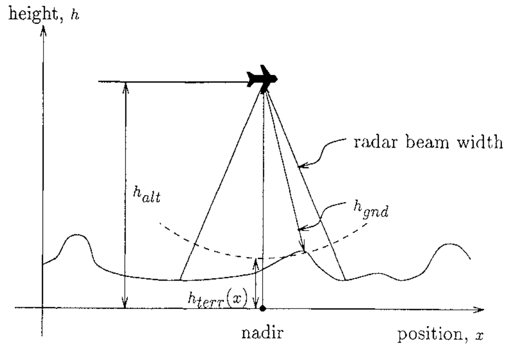

Fig. 1 Radar altimeter measurement.

III. Developing VATAN

Applying the VA to TAN results in VATAN. In VATAN, the VA generates optimal MAP aircraft position estimates using the terrain elevation beneath the aircraft as its observation. In order to implement VATAN, we need the conditional observation and state transition densities in Eq. (1) as well as an initial value of the metric L_0. In this paper, very simple models are used to obtain the required densities: in particular, we assume a flat, nonrotating Earth and simple aircraft dynamics; we do not model bias error terms; and we do not model mapmaking errors. Conceptually, extension of VATAN to more complicated models is straightforward; however, some care would be needed to keep the algorithm computationally manageable.

A. Generating Observation Conditional Probability Density

The measurement used by VATAN, the observed terrain elevation, is an indirect measurement obtained as the difference between the INS altitude measurement and the radar altimeter reading. To obtain the observation conditional probability density $p(z_k \mid \boldsymbol{x}_k)$, we must model the radar altimeter operation as well as characterize the errors in the radar altimeter and INS altitude measurements.

Ideally, the radar altimeter measures the distance between the aircraft and its nadir, the ground point directly beneath the aircraft. The measurement is made by reflecting electromagnetic energy off the terrain beneath the aircraft, as depicted in Fig. 1. All radar altimeters have a nonzero radar beam width, typically 14–30 deg. This may result in the altimeter measuring the distance to some point in the beam other than the nadir point. Occurrences of the radar altimeter returning the distance to a point other than the nadir are prevalent in rough terrain. Depending on the type of radar altimeter, the measurement may be the distance to the point in the beam that either is closest to the aircraft (i.e., distance h_{gnd} in Fig. 1) or maximizes the power in the reflected signal. For our simulations we model the first type of radar altimeter.

In the following, we denote as $h_{\text{terr}}(\boldsymbol{x}_k)$ the nominal terrain height that would be computed assuming zero measurement errors. It is computed, using the true INS altitude h_{INS} and error-free radar altimeter reading h_{gnd}, as

$$h_{\text{terr}}(\boldsymbol{x}_k) = h_{\text{INS}} - h_{\text{gnd}} \qquad (3)$$

Note that since h_{gnd} may not be the distance from the aircraft to the terrain point directly below the aircraft, $h_{\text{terr}}(\boldsymbol{x}_k)$ may not be the true terrain height at the point $\boldsymbol{x}_k$. However, using the terrain database and the known characteristics of the radar altimeter, the nominal value of h_{gnd} that would be measured by an aircraft at an elevation h_{INS} and position $\boldsymbol{x}_k$ can be computed, from which $h_{\text{terr}}(\boldsymbol{x}_k)$ can be obtained.

The INS provides a measurement of the aircraft's altitude $h_{\text{INS}}(k)$; we denote this measurement as $\hat{h}_{\text{INS}}(k)$. The radar altimeter provides a measurement of $h_{\text{gnd}}(k)$; we denote this measurement as $\hat{h}_{\text{gnd}}(k)$. We assume that the measurement errors are independent and Gaussian with variances $\sigma^2_{\hat{h}_{\text{INS}}}(k)$ and $\sigma^2_{\hat{h}_{\text{gnd}}}(k)$. The observation used in VATAN is the measured terrain elevation

$$z_k \doteq \hat{h}_{\text{terr}}(k) = \hat{h}_{\text{INS}}(k) - \hat{h}_{\text{gnd}}(k)$$

This indirect observation is used because it, along with a terrain elevation database, is sufficient to generate the observation conditional density $p(z_k \mid \boldsymbol{x}_k)$ used in the VA.

If the measurements $\hat{h}_{\text{INS}}(k)$ and $\hat{h}_{\text{gnd}}(k)$ are unbiased (i.e., $E[\hat{h}_{\text{INS}}(k)] = h_{\text{INS}}(k)$ and $E[\hat{h}_{\text{gnd}}(k)] = h_{\text{gnd}}(k)$), then

$$E[z_k \mid \boldsymbol{x}_k] = E[\hat{h}_{\text{INS}}(k)] - E[\hat{h}_{\text{gnd}}(k)] = h_{\text{terr}}(\boldsymbol{x}_k)$$

The conditional variance is

$$\sigma_z^2(k) \doteq \text{Var}(z_k \mid \boldsymbol{x}_k) = \sigma_{\hat{h}_{\text{INS}}}^2(k) + \sigma_{\hat{h}_{\text{gnd}}}^2(k)$$

Thus the conditional observation probability density function is

$$p(z_k \mid \boldsymbol{x}_k) = \frac{1}{\sqrt{2\pi\sigma_z^2(k)}} \exp\left(-\frac{[z_k - h_{\text{terr}}(\boldsymbol{x}_k)]^2}{2\sigma_z^2(k)}\right) \tag{4}$$

B. Generating State Transition Probability Density

The state transition density $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$ describes the states' evolution with time. Given a known velocity vector $\dot{\boldsymbol{x}}_k$ that is constant over the Δt second interval from t_k to t_{k+1}, the state's evolution is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \dot{\boldsymbol{x}}_k \Delta t \tag{5}$$

Since the aircraft's velocity is provided by the INS, it is not known precisely. This uncertainty is dealt with by modeling the INS velocity as a Gaussian random variable with mean $\bar{\dot{\boldsymbol{x}}}_k^{\text{INS}}$ and variance $\sigma_{\dot{x}_k^{\text{INS}}}^2$, indicative of the INS's precision. Thus the random variable $\boldsymbol{x}_{k+1}^k$ conditioned on $\boldsymbol{x}_k$ is a random variable with mean

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k + \bar{\dot{\boldsymbol{x}}}_k^{\text{INS}} \Delta t$$

and variance

$$\sigma_{\boldsymbol{x}_{k+1}}^2 = \sigma_{\dot{x}_k^{\text{INS}}}^2 \Delta t^2$$

Since $\boldsymbol{x}_0$ is assumed to be Gaussian, by Eq. (5), $\boldsymbol{x}_k$ is Gaussian for $k > 0$.

C. Initializing Metric

Recalling the VA (1), the metric function is initialized by

$$L_0(\boldsymbol{x}_0) = \ln p(\boldsymbol{x}_0)$$

where $\boldsymbol{x}_0$ is assumed to be a Gaussian random variable whose mean is the INS's position reading

$$\bar{\boldsymbol{x}}_0 = \boldsymbol{x}_0^{\text{INS}}$$

and whose variance is specified by a known circular error probability (CEP) of the INS. Since $p(\boldsymbol{x}_0)$ is Gaussian, the initial metric is a paraboloid, so the best initial estimate, that $\boldsymbol{x}_0$ that maximizes $L_0(\boldsymbol{x}_0)$, is

$$\hat{x}_0 = \boldsymbol{x}_0^{\text{INS}}$$

which makes intuitive sense.

IV. Implementing VATAN

VATAN consists of implementing Eq. (1), where $\boldsymbol{x}_k$ is the aircraft's position and z_k is the measured elevation of the terrain beneath the aircraft. Performing the smoothing in Eq. (2) is not necessary for TAN because TAN requires an optimal estimate of the current state, not past states. At present implementing Eq. (1) over a continuous state space is still an open problem; thus we approximate Eq. (1) using a discrete grid of points.

Unfortunately, implementing Eq. (1) over a discrete state space leads to suboptimal estimates for two reasons. First, the optimal state estimate is given by that $\hat{x}_k$ that maximizes $L(\hat{x}_k)$, an arbitrary function; however, when computing $L(\cdot)$ on a discrete state space, it is probable that the maximum value of $L(\cdot)$ occurs for a value of $\boldsymbol{x}_k$ not among the discrete state values. As a result, the use of a discrete state space leads to suboptimal estimates. The problem is further compounded by the implementations of $\ln p(z_k \mid \boldsymbol{x}_k)$ and $\ln p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$ in a discrete state space. Obviously, these functions' maxima play an important role in determining the maxima of $L(\cdot)$. However, in a discrete state space these peaks may be lost due to discretization errors that occur when implementing $\ln p(z_k \mid \boldsymbol{x}_k)$ and $\ln p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k)$, again leading to suboptimal estimates.

Fortunately, several implementation techniques improve the performance of the discrete state VA. These include techniques to adaptively choose the grid spacing and grid extent at each instant in time, techniques for representing each term in the VA, and techniques for performing the algorithm's computations.[7]

V. Simulation Model

We evaluate the performance of VATAN and compare it to a particular implementation of SITAN through the use of Monte Carlo simulations. All of the simulations are performed using a computer-generated simulated terrain database. In each simulation run, an aircraft trajectory is created; then simulated INS velocity, INS altitude, and radar altimeter ground clearance measurements are generated. These measurements are used by VATAN and SITAN to estimate the aircraft trajectory. This estimated trajectory is then compared with the true trajectory to obtain an error. The error is averaged over many simulation runs to determine sample error bias and sample error variance.

The implementation of these simulations requires that several models be developed: the aircraft dynamics model, the INS model, and the radar altimeter model. In addition, a method for synthetically generating a terrain database must be developed.

A. Aircraft Models

The model used to simulate the aircraft motion is the standard kinematic equation implemented in discrete time that is described in Sec. III.B. This is a simplification of an aircraft's state dynamics, but it is sufficient for testing the TAN methods. The INS model simulates measurements of the aircraft's position and velocity by adding measurement errors to the true aircraft states. These measurement errors are initialized at $k = 0$ and are propagated using standard INS modeling techniques (i.e., Shuler oscillations).[11,12] The INS altitude measurement is modeled as a Gaussian random variable whose mean is the true altitude and is subject to a specified calibration bias and a specified measurement error variance. The model of the radar altimeter and the computation of the terrain height from the radar altimeter and INS altitude is described in Sec. III.A.

B. Terrain Model

Both simulated terrain and its corresponding topographical map have to be generated to test the performance of SITAN and VATAN over a variety of terrain types. There are two approaches to terrain and map simulation. The first approach is to use an existing map to provide the true terrain data, bilinearly interpolating where necessary. However, generating terrain data from a map results in terrain with constant gradients between map post points, and SITAN's performance relies explicitly on locally constant terrain gradients.[3,5] Thus, this first approach masks potential problems due to nonconstant gradients of actual terrain.

The second approach is to first generate simulated terrain on the computer and then create map data from this simulated terrain. This approach is taken since it does not result in unnaturally constant gradients and thus gives more realistic results. Several methods based on fractals generate realistic simulated terrain.[13–15] However, any method for generating terrain suffices, as long as it generates the terrain types of interest; flat terrain, mountainous terrain, rough terrain, etc.

We found it simplest to develop our own algorithm to generate the specific types of terrain of interest. Our algorithm begins with a large square region and generates random elevations at each corner. The elevations are modeled as Gaussian random variables of specified mean and variance. For rougher terrain, higher variances are used. The square is then subdivided into four smaller squares, and the elevation at each new corner is generated by bilinearly interpolating between the original four corners and adding to each new corner an additional realization of a zero-mean Gaussian random variable. The variance of this new random variable determines the magnitude of the local terrain variations. Each square is then further subdivided and new elevations are generated repeatedly, using the above technique, until a complete set of elevations at the desired resolution is generated.

For testing the TAN algorithm, the simulated terrain is generated at a 20-m post spacing over a 10×10 km region. Most topographical databases have a post spacing of 100 m, so the simulation's resolution of 20 m captures the small-scale details that are not provided by interpolation of an existing map. The simulated region is limited to 10×10 km so that we can assume a flat-Earth model, eliminating modeling difficulties associated with the effects of Earth's curvature.

A topographical map was simulated by adding a mapmaking measurement error to the true elevation at each data point on the topographical map model. The measurement error was modeled as a Gaussian random variable of specified bias and variance; measurement errors were independent from point to point.

VI. Simulation Results

To evaluate the performance of VATAN, we performed Monte Carlo simulations (consisting of 100 runs) for low-velocity helicopter flight over four different terrain profiles. As a reference, we also performed the same simulations on a five-state, single EKF implementation of SITAN operating in the tracking mode. The SITAN implementation parameters are set to those used in previous SITAN work.[3] This SITAN implementation was chosen to compare the relative performances of the VA and the EKF in TAN.

The terrain database used for the simulations was generated by the method outlined in Sec. V.B. The four terrain profiles chosen for evaluation represent typical rough terrain, flat terrain, mountainous terrain, and a mixture of sloped and flat terrain. The first three profiles are chosen because they represent terrain in which TAN methods typically experience difficulty. The fourth profile is chosen because it is a type of terrain in which TAN methods perform well in general. Figure 2 shows a graph and Fig. 3 shows a contour plot of the simulated terrain in which the TAN methods are evaluated. The contour plot also shows the flight paths of the four scenarios.

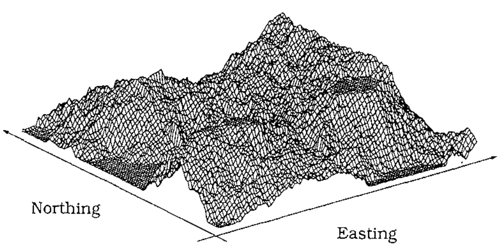

Fig. 2 Simulated terrain.

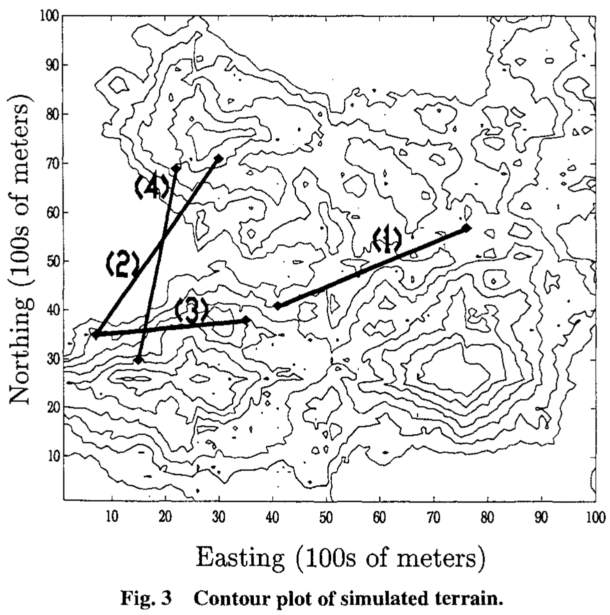

Fig. 3 Contour plot of simulated terrain.

Table 1 Equipment parameters for Monte Carlo simulations

Parameter	Value
Initial INS position deviation (one axis)	100 m
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
Accelerometer bias	0 μG
Accelerometer deviation	10 μG
Gyroscope bias stability	0.01 deg/h
INS altitude bias	0 m
INS altitude deviation	2.5 m
Radar altimeter bias	0.5 m
Radar altimeter deviation	2.5 m
Radar altimeter beamwidth	30 deg

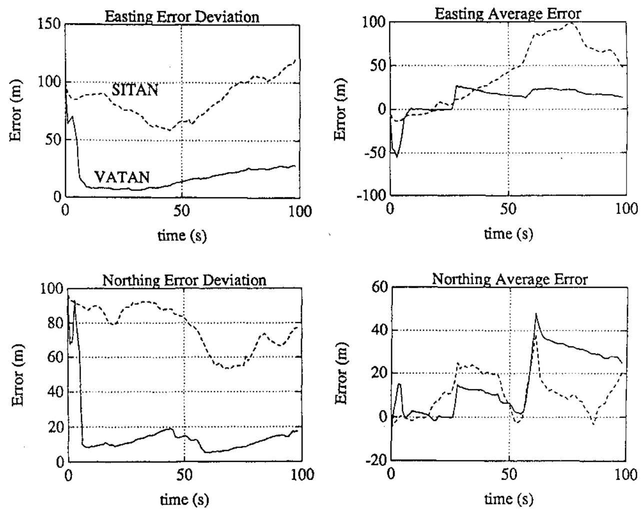

Fig. 4 Monte Carlo statistics for scenario 1.

All scenarios use the aircraft parameters given in Table 1. The simulations are randomized with respect to both the initial INS errors and the measurement variances of each piece of aircraft equipment. All scenarios are 100 s in duration with observations made at 1-s intervals.

A. Scenario 1: Typical Rough Terrain

Scenario 1 compares VATAN's performance to that of our implementation of SITAN when flying over typical rough terrain. The flight begins at position (4000 m, 4000 m) (coordinates are easting, northing). Flying with a constant velocity of (35 m/s, 16 m/s) at an altitude of 1300 m, the aircraft arrives at (7500 m, 5600 m) after 100 s.

Figure 4 shows the Monte Carlo sample mean and sample variance over 100 runs of the estimation error in easting and northing for both VATAN and SITAN; the values for VATAN are shown with the solid line, and the values for SITAN are shown with the dashed line. In terms of average position error, VATAN performs much better than SITAN; SITAN performs poorer because the terrain is rough and the EKF used in SITAN is based on a model that essentially assumes terrain with a constant gradient.

B. Scenario 2: Flat Terrain

Scenario 2 contrasts VATAN's performance to our implementation of SITAN when flying over regions of flat terrain. The flight begins over rough terrain at position (600 m, 3400 m). However, flying with a constant velocity of (23 m/s, 36 m/s) at an altitude of 1500 m, the aircraft quickly enters a large flat region. The flight is terminated over rough terrain at (2900 m, 7000 m).

The Monte Carlo simulation results for VATAN and SITAN are shown in Fig. 5. VATAN performs very well compared to SITAN, which is unable to recover after flying over the flat terrain. SITAN is unable to recover after flying over the flat terrain because the position estimate becomes so inaccurate that it is no longer in the

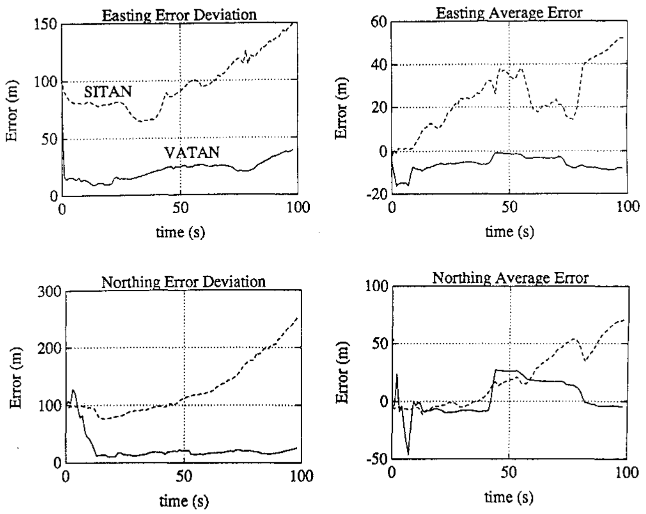

Fig. 5 Monte Carlo statistics for scenario 2.

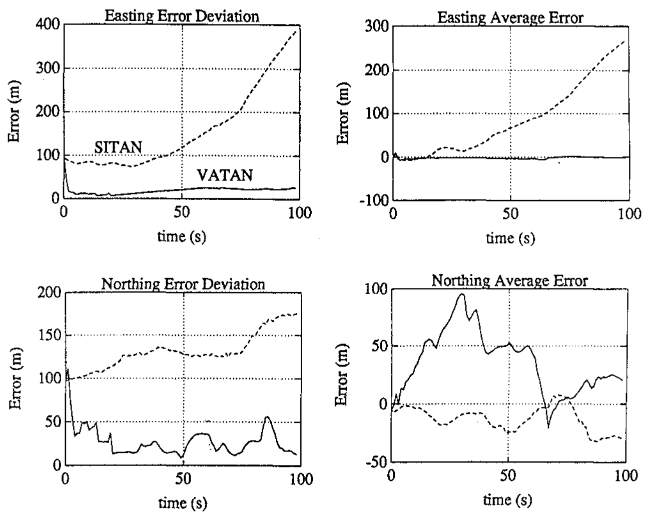

Fig. 6 Monte Carlo statistics for scenario 3.

linearized region of SITAN's EKF. On the other hand, VATAN is able to recover because the VA's metric function can be evaluated over as large an area as required. This scenario illustrates well how VATAN's algorithm is valid over a global area (and is limited only by processing capabilities) whereas SITAN's algorithm is only valid within a local area.

C. Scenario 3: Mountainous Terrain

Scenario 3 compares VATAN's performance to that of SITAN when flying across a mountain face. The flight begins at (600 m, 3400 m) and continues across the face at a constant velocity of (28 m/s, 3 m/s) and altitude of 1500 m for a duration of 100 s.

The Monte Carlo simulation results for VATAN and SITAN are shown in Fig. 6. VATAN performs very well whereas SITAN does not work well. In fact, SITAN's average easting error diverges badly. This occurs because the terrain is nearly flat with respect to movement eastward along the mountain face (and hence there is no slope information by which the SITAN estimate can be corrected).

D. Scenario 4: Sloped and Flat Terrain

Scenario 4 illustrates that for terrain with a fairly constant gradient, VATAN's performance is comparable to SITANs. The flight begins at (1400 m, 2900 m) and passes over linear terrain for the first third of the flight. The second third of the flight is over flat terrain and the final leg is over linear terrain. The flight is at a constant velocity of (28 m/s, 3 m/s) and altitude of 1500 m for a duration of 100 s.

The Monte Carlo simulation results for VATAN and SITAN are shown in Fig. 7. Both VATAN and SITAN perform well during the flight. Over terrain with a fairly constant gradient, both methods use the observation information to improve the state estimate. Over flat terrain, the state estimate error increases as the INS drifts because of the Shuler oscillation. When terrain with a constant gradient is again encountered, both methods use the observation information to improve the state estimate.

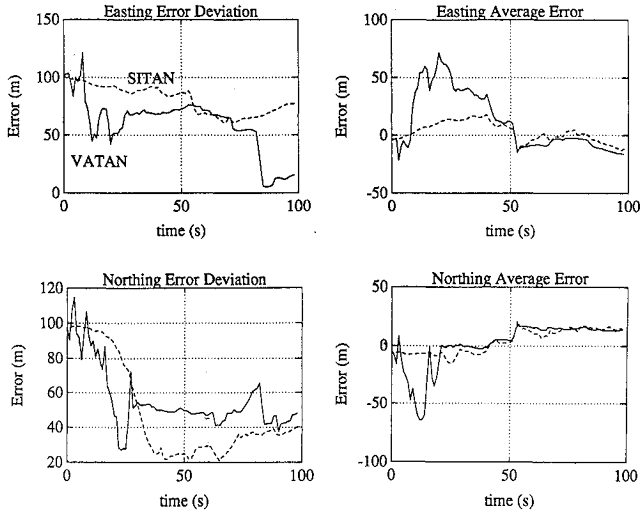

Fig. 7 Monte Carlo statistics for scenario 4.

E. Summary of Results

In the scenarios considered in this paper, VATAN consistently performs as well as or better than our implementation of SITAN. VATAN performs as well as SITAN in moderately rough, sloped terrain (scenario 4) and it exceeds SITAN's performance in very flat or very rough terrain (scenarios 1–3).

VII. Conclusions

This paper has introduced VATAN, a new TAN method based on the VA. Our simulation results demonstrate that the VA estimator in VATAN has several advantages over the EKF used in SITAN. As a result, VATAN is less susceptible to large errors in the position estimate that occur in common circumstances such as flying over very flat or very rough terrain. These results also show that VATAN performs as well as or better than our implementation of SITAN under all the circumstances that we evaluated, indicating that the use of the VA for a TAN system could overcome many of the problems associated with the use of the EKF.

In addition to more accurate position estimates, VATAN has several other advantages over SITAN and other methods such as TERCOM. First, VATAN incorporates a model of the radar altimeter into its algorithm. Using the terrain database, a model of the radar altimeter's operation can be directly incorporated into the conditional observation probability density function of the VA. Thus, even with misleading measurements due to a nonzero radar altimeter beamwidth, VATAN can still improve the state estimate. This contrasts to SITAN, which does not explicitly incorporate a realistic radar altimeter model; hence, poor observations will just as likely worsen the state estimate.

Another advantage of VATAN is that observations of physical structures, including buildings, bridges, trees, and other features not located in the terrain database, can be directly incorporated into the VA's observation probability density. Thus, with straightforward modifications, VATAN could be extended to use a separate database containing the locations of observed physical structures to improve the position estimate.

VATAN's major limitation is the increased computational capacity necessary to implement the VA when compared to an EKF. Currently, the two-dimensional VA has only been implemented in a discrete state space; a continuous state-space implementation of the two-dimensional VA would improve VATAN's accuracy and could result in a substantial reduction of the computational capacity necessary to implement VATAN. The implementation of a continuous state-space VA is presently an active topic of research.[16]

So far, VATAN has only been evaluated in a somewhat oversimplified simulated environment and has only been compared to a single EKF implementation of SITAN. These comparisons demonstrate that VATAN overcomes the problems associated with the EKF. However, more detailed modeling and simulation of the INS, radar altimeter, and errors associated with terrain maps, as well as comparison with a multiple filter implementation of SITAN, will be necessary to establish VATAN as a viable TAN method. This simulation, as well as eventual flight testing, will be required to optimize VATAN's implementation and to evaluate VATAN's performance relative to other existing TAN methods.

References

[1]Farrell, J. L., *Integrated Aircraft Navigation*, Academic, New York, 1976.

[2]Hollowell, J., "Heli/SITAN: A Terrain Referenced Navigation Algorithm for Helicopters," *IEEE PLANS'90 Proceedings* (Las Vegas, NV), Inst. of Electrical and Electronics Engineers, New York, 1990, pp. 616–625.

[3]Fellerhoff, J. R., Hochrein, G. J., Lau, M. K., Schindoff, R., and Wardlaw, R., "SITAN Design for Low-Level Attack Aircraft, Vol. 1: Flight Tests, Maps, Algorithm Design, and Performance," Sandia National Labs., TR AFWAL-TR-84-1148, Albuquerque, NM, 1985.

[4]Boozer, D. D., Lau, M. K., and Fellerhoff, J. R., "The AFTI/F16 Terrain-Aided Navigation System," *National Aerospace and Electronics Conference Proceedings*, Inst. of Electrical and Electronics Engineers, New York, pp. 351–357.

[5]Hostetler, L. D., and Andreas, R. D., "Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation," *IEEE Transactions on Automatic Control*, Vol. 28, No. 3, 1983, pp. 315–323.

[6]Baker, W. R., and Clem, R. W., "Terrain Contour Matching (TERCOM) Primer," Wright–Patterson AFB, TR ASD-TR-77-61, 1977.

[7]Enns, R. J., "A Terrain-Aided Navigation Method Based on the Continuous State Viterbi Algorithm," M.S. Thesis, Arizona State Univ., Tempe, AZ, Aug. 1993.

[8]Larson, R. E., and Peschon, J., "A Dynamic Programming Approach to Trajectory Estimation," *IEEE Transactions on Automatic Control*, Vol. AC-11, July 1966, pp. 537–540.

[9]Forney, G. D., "The Viterbi Algorithm," *Proceedings of the IEEE*, Vol. 61, No. 3, 1973, pp. 268–278.

[10]Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum Likelihood Estimates of Linear Dynamic Systems," *AIAA Journal*, Vol. 3, 1965, pp. 1445–1450.

[11]O'Donnell, C. F., *Inertial Navigation Analysis and Design*, McGraw–Hill, New York, 1964.

[12]Broxmeyer, C., *Inertial Navigation Systems*, McGraw–Hill, New York, 1964.

[13]Mandlebrot, B. B., "Stochastic Models for the Earth's Relief, the Shape and the Fractal Dimension of Coastlines, and the Number-Area Rule for Islands," *Proceedings of the National Academy of Science*, Vol. 72, No. 10, 1975, pp. 3825–3828.

[14]Mandlebrot, B. B., *The Fractal Geometry of Nature*, Freeman, San Francisco, 1982.

[15]Musgrave, F. K., Kolb, C. E., and Mace, R. S., "The Synthesis and Rendering of Eroded Fractal Terrains," *Computer Graphics*, Vol. 23, No. 3, 1989, pp. 3825–3828.

[16]Champlin, C. R., and Morrell, D. R., "Non-Linear MAP Estimation Using a Continuous-State Viterbi Algorithm," *Proceedings of the IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, 1992, pp. 3707–3708.