Track Splitting Filtering Implementation for Terrain Aided Navigation

Dr. Vedat EKÜTEKİN and Prof. Dr. M. Kemal ÖZGÖREN
TÜBİTAK-SAGE, Navigation Division /
Middle East Technical University, Mechanical Engineering Department
TURKEY

veku@sage.tubitak.gov.tr / ozgoren@metu.edu.tr

ABSTRACT

Terrain Aided Navigation (TAN) is a technique which estimates the real position of a moving vehicle by comparing the measured terrain profile under the vehicle with a stored elevation map. TAN has been operational especially for unmanned air vehicles since 1960's and is used in order to aid an Inertial Navigation System (INS) by providing position fixes. Especially, if other sources for position aids, like Global Positioning System (GPS), are not available, TAN can provide reliable position information, especially in low level flights over significant terrain and increase the accuracy of the INS. Until now, several TAN techniques have been developed and tested. These fall into two algorithmic categories:

1. *Batch algorithms where post navigation solutions are performed. For example, TERCOM.*

2. *Recursive algorithms where real-time navigation solutions are performed. For example, SITAN, VATAN, etc.*

Moreover, TAN algorithms are applied for both acquisition and tracking purposes. In acquisition mode of operation, large initial position errors of the INS are fixed using the TAN algorithm, generally TERCOM. On the other hand, in tracking mode of operation, relatively small position errors of INS which are determined by the grid size of the elevation data used are reduced in real-time by using recursive algorithms, like SITAN.

In this work, Track Splitting Filtering (TSF) which was previously developed for radar tracking algorithms is implemented as a new TAN algorithm for real-time navigation solutions. The implemented recursive algorithm can be used for both acquisition and tracking modes of operation. Implementation of the TSF algorithm for acquisition and tracking modes of operation is shown with simulations. In addition, it is shown that results of the acquisition mode are improved and probability of false position fixes decrease compared with TERCOM using Monte Carlo simulations.

1.0 INTRODUCTION

Terrain Aided Navigation (TAN) is a technique to estimate the position of a moving vehicle by comparing the measured terrain profile under the vehicle to a stored elevation map and it has been operational especially for unmanned vehicles since 1960's. TAN provides position fixes, which can be used to aid an integrated navigation system. Especially, if other sources for position aids, like the Global Positioning System (GPS), are not available, TAN can provide reliable position information in low level flights over significant terrain.

TAN consists of sensing a terrain elevation profile beneath an air vehicle and correlating the profile with stored digital terrain elevation data (DTED) to produce an estimate of vehicle position. An INS, usually

Ekütekin, V.; Özgören, M.K. (2007) Track Splitting Filtering Implementation for Terrain Aided Navigation. In *Military Capabilities Enabled by Advances in Navigation Sensors* (pp. 8-1 – 8-12). Meeting Proceedings RTO-MP-SET-104, Paper 8. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int.

with barometric altimeter aiding, provides the approximate trajectory. TAN systems provide three dimensional position updates to the navigation system by estimating INS trajectory errors. Radar or laser altimeter measures ground clearance and the DTED gives terrain elevation above mean sea level (MSL). Implementation requires an INS, an altimeter, DTED, and a flight computer for executing the TAN algorithm. In Figure 1, an illustration is given for TAN measurement process [1].

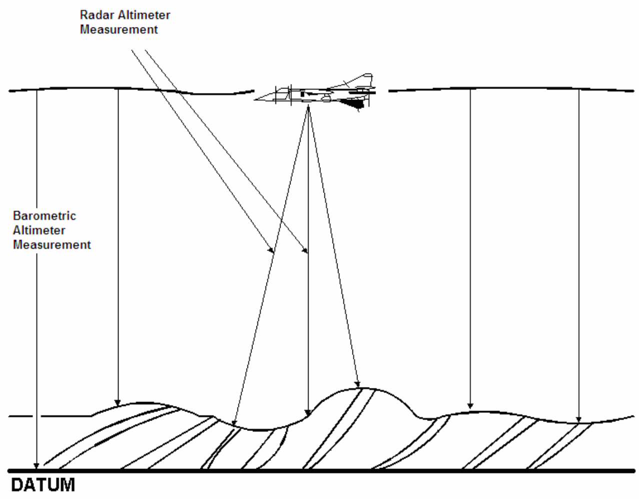

Figure 1: TAN Measurements [1]

TAN techniques fall into two algorithmic categories; batch algorithms where post navigation solutions are performed and recursive algorithms where real-time navigation solutions are performed. Moreover, TAN algorithms are applied for both acquisition and tracking purposes. In acquisition mode of operation, large initial position errors of the INS are fixed using the TAN algorithm. On the other hand, in tracking mode of operation, relatively small position errors of INS which are determined by the grid size of the elevation data used are reduced in real-time by using recursive algorithms [2].

The most widely known form of TAN which uses batch algorithm is TERCOM. TERCOM is a form of correlation guidance based on a comparison between the measured and the pre-stored features of the profile of the ground (i.e., terrain) over which a missile or aircraft is flying. Generally, terrain height forms the basis of this comparison [3]. Actually, TERCOM is a maximum likelihood estimator which uses only terrain height information for determining the vehicle's actual position. On the other hand, the major recursive TAN algorithm found in literature is SITAN proposed by Hostetler and Andreas [4]. SITAN uses an extended Kalman Filter (EKF) and a local terrain linearization technique to implement a recursive algorithm. This algorithm operates on individual terrain elevation measurements as they become available and for the entire duration of the mission [2]. In order to perform real-time navigation solutions using TAN, kinematical behavior of the system should be modeled. However, batch algorithms use only height measurements and related DTED unlike recursive TAN algorithms where kinematical system models are used for real-time navigation solutions.

The modern need for tracking algorithms began with the development of radar during World War II. A near-optimal method for addressing a large class of tracking problems was developed in 1960 by R.E. Kalman. His approach, referred to as Kalman filtering, involves the recursive fusion of noisy measurements to produce an accurate estimate of the state of a system of interest. Kalman's work had a dramatic impact on the field of target tracking in particular and data fusion since mid-1960's [5]. Modern radar tracking algorithms generally deal with complex problems of tracking like clutter or multiple targets. In tracking targets with less-than-unity probability of detection in the presence of false alarms (clutter), data association, deciding which of the received multiple measurements to use to update each track is crucial. Data association becomes more difficult with multiple targets where the tracks compete for measurements. Here, in addition to a track validating multiple measurements as in the single target case, a measurement itself can be validated by multiple tracks. Several algorithms are developed to handle this contention like track splitting, multiple hypothesis tracking (MHT), probabilistic data association (PDA) and joint probabilistic data association (JPDA) [5].

TAN is a nonlinear estimation problem; since, terrain height information is used for navigation solution. Actually, TAN can be considered as a data association problem, especially for the acquisition operation mode where INS position errors are considerably large. From the literature survey of Quintang, et al [6] and Dezert [7] where Probabilistic Data Association (PDA) filter is used for navigation applications, it has been thought that modern data association algorithms can be implemented for real-time TAN algorithms. In this paper, Track Splitting Filtering (TSF) is implemented as a new TAN algorithm for real-time navigation solutions. The implemented recursive algorithm can be used for both acquisition and tracking modes of operation. Implementation of the TSF algorithm for acquisition and tracking modes of operation is shown with simulations. In addition, it is shown that results of the acquisition mode are improved and probability of false position fixes decrease compared with TERCOM using Monte Carlo simulations.

2.0 THEORY

2.1 Multiple Hypothesis Tracking (MHT) and Track Splitting Filtering (TSF)

In classical multiple-target tracking, the problem is divided into two steps, association and estimation. Step 1 associates contacts with targets. Step 2 uses the contacts associated with each target to produce an estimate of that target’s state. Complications arise when there is more than one reasonable way to associate contacts with targets. The classical approach to this problem is to form association hypotheses and to use MHT. In this approach, alternative hypotheses are formed to explain the source of the observations. Each hypothesis assigns observations to targets or false alarms. For each hypothesis, MHT computes the probability that it is correct. This is also the probability that the target state estimates that result from this hypothesis are correct. Most MHT algorithms display only the estimates of target state associated with the highest probability hypothesis [8].

The original MHT method, denoted Reid’s algorithm, was first presented by Reid [9]. There are two basic approaches to MHT implementation. The first (hypothesis-oriented) approach follows the original work of Reid [9]. It maintains the hypothesis structure from scan to scan and continually expands and cuts back (prunes) the hypotheses as new data are received. At each scan, a set of hypotheses will be carried over from the previous scan and composed of one or more tracks that are compatible with all other tracks in the hypothesis. Compatible tracks are defined to be tracks that do not share any common observations. Then, on the receipt of new data, each hypothesis is expanded into a set of new hypotheses by considering all observation-to-track assignments for the tracks within the hypothesis. Again, as new hypotheses are formed, the compatibility constraint for tracks within a hypothesis is maintained [10].

An alternative (track-oriented) approach does not maintain hypotheses from scan to scan. The tracks formed on each scan are reformed into hypotheses and the tracks that survive pruning are predicted to the next scan where the process continues [10].

In Figure 2, the operations of MHT that are required by both implementation methods are summarized.

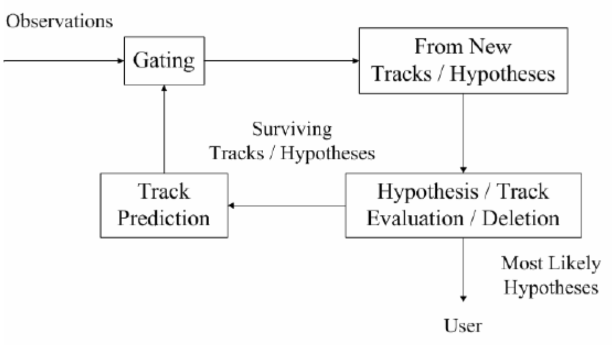

Figure 2: MHT Logic Overview [10]

TSF is proposed by Smith and Buechler [11] and older than the original MHT method presented by Reid [9]. In TSF, a tree of hypotheses is kept for each target individually, and a maximum likelihood criterion is used to prune the tree. On the other hand, Reid's MHT constructs a tree of all possible hypotheses, including all possible new track initiations at every time step. Reid discusses a number of strategies to prune the tree in order to achieve reasonable computation times. In the paper, TSF is implemented for TAN due to INS error model characteristics. Since, horizontal INS error bound is estimated for the air vehicle and errors do not change rapidly, implementation of TSF for TAN became sufficient for navigation solutions.

2.2 Implementation of TSF to TAN

TSF is a recursive branching algorithm for multiple-object discrimination and tracking consists of a bank of parallel filters of the Kalman form, each of which estimates a trajectory associated with a certain selected measurement sequence. The measurement sequences processed by the algorithm are restricted to a tractable number by combining similar trajectory estimates, by excluding unlikely measurement/ state associations, and by deleting unlikely trajectory estimates. The measurement sequence selection is accomplished by threshold tests based on the innovations sequence and state estimates of each filter [11].

Implementation of TSF to TAN is done using standard TSF procedure [12] considering INS error characteristics of the air vehicle as follows:

1. A predicted observation and validation gate are computed.
   - Measurement gate is taken as the 3σ horizontal error bound of the INS and the invalid possibilities for all height differences $\delta h_i(k)$ used in measurement gate, where i denotes the index of the position in the gate, are discarded such that all height differences satisfy:

     $$\delta h_i(k) \leq \left| \left[\gamma \cdot \left(\sigma_{h_{INS}}^2 + \sigma_{radar}^2 \right) \right]^{1/2} \right| \tag{1}$$

     where, γ is the gate threshold taken as 16 (4σ vertical error bound) considering 99.9989% of the measurements to be in the gate and σ^2 are the variances of INS height (i.e. barometric altimeter) and radar altimeter. In other words, impossible height difference measurements are discarded from the navigation solutions.

     Height differences $\delta h_i(k)$ are defined as the difference between the measured and estimated height clearances for all grid positions in the defined region. Considering TAN measurements given in Figure 3, single height difference is defined as:

$$\delta h(k) = C_{meas}(k) - C_{est}(k) \tag{2}$$

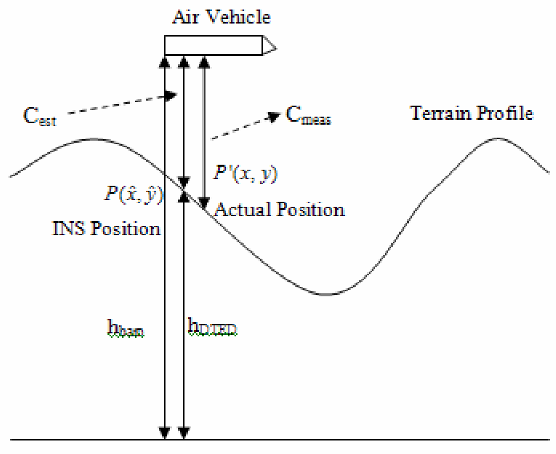

Figure 3: TAN Measurement Process

2. All validated observations are associated.
   - Every grid position (i.e. index) that satisfy validated height differences in the 3σ horizontal error bound of the INS is considered to be one of the possible navigation solutions.
3. The track is updated separately with each validated hypothesis and the likelihood of the each entire track sequence is computed.
   - Navigation solution is assumed to be one of the grid index followed by some of the tracks in the 3σ horizontal error bound of the INS. According to the index of the grid position, there exist "n x n" possible tracks (i.e. hypothesis) for each time step where "n x n" denotes the batch size of the DTED considered. At the initial time step, there exist "n x n" possible tracks from INS position grid to all possible grid positions as shown in Figure 4. The modified log-likelihood of the each possible track sequence is computed as follows:

$$\lambda_i(k) \equiv -2 \cdot \log\left[\frac{\Lambda_i(\Theta^{k,l})}{c_k}\right] = \sum_{j=1}^{k} v_{M_i}^T(j) \cdot S^{-1}(j) \cdot v_{M_i}(j) \qquad \text{for} \quad i = 1 \ldots "n \times n" \tag{3}$$

   where, Λ_i is the likelihood function, c_k is constant, v_{M_i} is the innovation between tracked and measured height differences and S is the innovation covariance matrix.
4. Some pruning of the hypothesis tree takes place.
   - Number of possible tracks is limited with the positions in the 3σ horizontal error bound of the INS considering INS error characteristics. In order to reduce the number of hypotheses, definite number of tracks with minimum likelihoods is selected for the navigation solution. In other words, best "M" tracks with minimum likelihoods of the existing "n x n" possible tracks are selected for the navigation solution. Hence, hypotheses are pruned.
5. Each track hypothesis is now independently predicted forward to the next time-step.
   - Navigation solution is found for each possible track using standard Kalman filter equations as given in the reference papers for MHT/ TSF procedure. Using a definite number of minimum likelihood values of the entire track sequences, navigation solution is achieved. This is done by computing modified log-likelihood function recursively for each formed new track and

selecting the track with minimum likelihood value as follows:

$$\lambda_{ij}(k+1) = \lambda_{ij}(k) + v_{M_{ij}}^{T}(k+1) \cdot S^{-1}(k+1) \cdot v_{M_{ij}}(k+1) \qquad \text{for } i = 1...M,\ j = 1..."n \times n" \tag{4}$$

where, k is the time step.

- In order to decrease the effects of the old measurements, modified log-likelihood function defined in equation (3) can be used by a weighting factor K_{WF} as follows:

$$\lambda_{ij}(k+1) = K_{WF} \cdot \lambda_{ij}(k) + v_{M_{ij}}^{T}(k+1) \cdot S^{-1}(k+1) \cdot v_{M_{ij}}(k+1) \qquad \text{for } i = 1...M,\ j = 1..."n \times n" \tag{5}$$

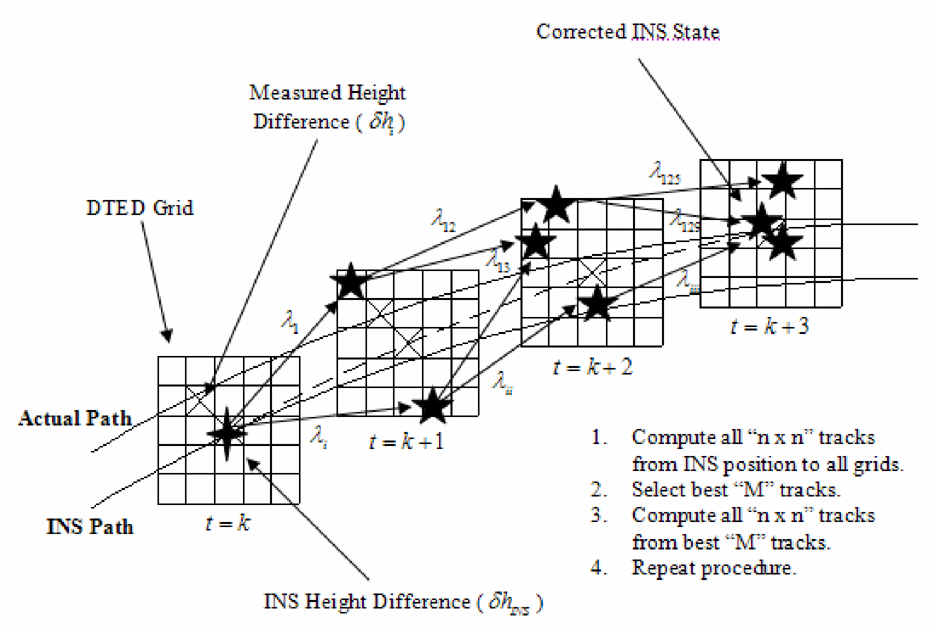

Figure 4: TSF Track Formation and Pruning

6. Procedure defined at the previous steps is done recursively in order to obtain navigation solution in real-time.

3.0 SIMULATIONS

In order to perform simulations, first trajectory and INS error models are formed. Then, DTED height model is prepared. Next, TAN models are formed which include SITAN, TERCOM and TSF in order to compare the results of the implemented TSF algorithm for both tracking and acquisition modes of operation. Finally, the overall architecture is formed in order to perform simulations for position errors along east and north directions of the vehicle motion.

Simulation architecture is shown in Figure 5. Loosely coupled integration structure is used where INS is not updated at each TAN correction step but updated at a greater period. This is done in order not to influence INS results from possible fault corrected TAN solutions.

3.1. Simulation Model Development

For the simulations, the motion of the vehicle is modeled considering the mid-course flight of an air vehicle with constant heading and velocity motion at constant altitude. Since the height terms will be taken from the DTED database according to vehicle's latitude and longitude (i.e. horizontal positions), height is not considered in the vehicle's state.

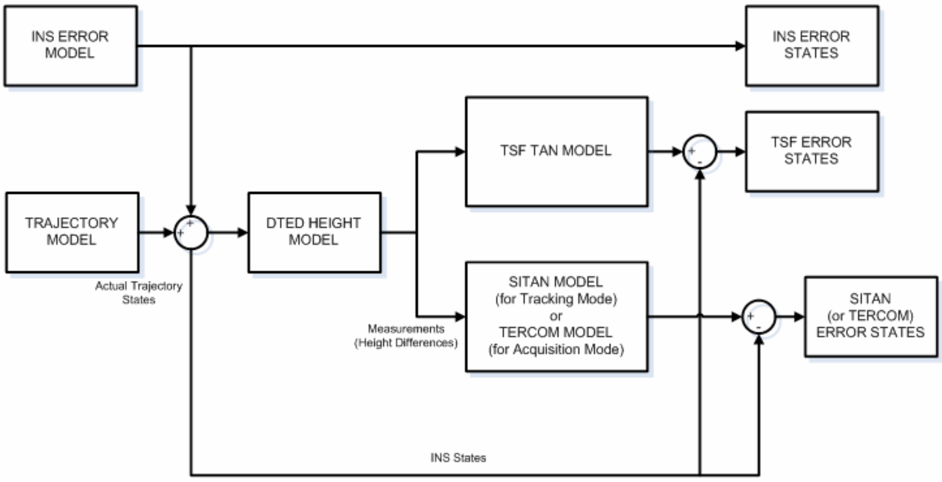

Figure 5: Simulation Architecture

Trajectory model of the vehicle considering continuous states can be modeled as follows:

$$\dot{\bar{x}}_{traj}(t) = F(t)\cdot \bar{x}_{traj}(t) \tag{6}$$

$$\bar{x}_{traj}(t) = \left[rN_{traj};rE_{traj};h_{traj};vN_{traj};vE_{traj};vD_{traj}\right]^T \tag{7}$$

$$F(t) = \begin{bmatrix} 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{bmatrix} \tag{8}$$

where, rN_{traj}, rE_{traj} are northward and eastward positions, h_{traj} is the altitude and vN_{traj}, vE_{traj}, vD_{traj} are the north, east and down velocities of the vehicle. Here, north and east velocities are assumed to be constant considering constant velocity and heading flight. Down velocity is also assumed to be zero considering level flight.

INS error model used in the simulations is taken from Bar-Itzhack and Berman [13] where ψ-angle approach is used and error equations are derived for Cartesian coordinates. Hence, INS error model can be written in discrete state-space form as follows:

$$\delta\bar{x}(k+1) = \Phi(k)\cdot\delta\bar{x}(k) + \bar{w}(k) \tag{9}$$

$$\delta\bar{x}(k) = \left[\delta rN;\delta rE;\delta h;\delta vN;\delta vE\right]^T \tag{10}$$

where, δ denotes the error states of positions, altitude and velocities. Here $\overline{w}(k)$ is the INS error state white noises derived from INS error model including INS sensor and mechanization errors.

For DTED height model, height of the vehicle is determined from the related DTED maps according to the related latitude and longitude of the vehicle. Measurement height differences are taken for the TERCOM, SITAN, and TSF models by adding INS white noises as system noise and radar white noises as measurement noise considering Figure 3. Then, height difference given in equation (2) used in DTED model becomes:

$$\delta h(k) = \underbrace{\left[h_{DTED}(\mu_{INS}(k), \lambda_{INS}(k)) + w_{INS}(k)\right]}_{\text{INS Height Model}} - \underbrace{\left[h_{DTED}(\mu_{traj}(k), \lambda_{traj}(k)) + w_{radar}(k)\right]}_{\text{Radar Height Measurement Model}} \tag{11}$$

where, μ is the longitude and λ is the latitude of the related INS and trajectory positions and w is the white noise term of the INS and radar height measurements.

SITAN and TERCOM models are taken considering the original models found in the literature. For SITAN, terrain slopes are derived considering the gradients of the height values of the related DTED files. Other equations are taken from Hostetler and Andreas [4] in order to form SITAN model for tracking mode. For TERCOM, maximum absolute difference (MAD) process [3] is used for correlation in acquisition mode.

For TSF model, system is taken as INS error model and standard Kalman filter equations are used as discussed in the previous section. These equations are summarized as follows:

TSF Measurement Model:

$$z_i(k) = H_m(k) \cdot \delta\overline{x}_i(k) + \overline{w}_{meas}(k) \tag{12}$$

where, $H_m(k) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$: Height measurement matrix,

$w_{meas}(k) = N(0, \sigma^2_{radar})$: Measurement white noise,

$R(k) = Cov\{\overline{w}_{meas}(k)\overline{w}_{meas}(k)^T\}$: Measurement noise covariance matrix.

State Estimate Propagation (for each track):

$$\delta\hat{\overline{x}}(k \mid k-1) = \Phi(k-1) \cdot \delta\hat{\overline{x}}(k-1 \mid k-1) \tag{13}$$

Error Covariance Propagation (for each track):

$$P(k \mid k-1) = \Phi(k-1) \cdot P(k-1 \mid k-1) \cdot \Phi(k-1)^T + Q(k-1) \tag{14}$$

TSF Gain Matrix (for each track):

$$K(k) = P(k \mid k-1) \cdot H_P(k)^T \cdot S(k)^{-1} \tag{15}$$

where, $H_P(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$: TSF measurement matrix,

$S(k) = H_P(k) \cdot P(k \mid k-1) \cdot H_P(k)^T + R(k)$: Innovation covariance matrix.

State Estimate Update (for each track):

$$\delta\hat{x}(k\,|\,k) = \delta\hat{x}(k\,|\,k-1) + K(k)\cdot v_P(k) = \delta\hat{x}(k\,|\,k-1) + K(k)\cdot\left[z_p(k) - \hat{z}_p(k\,|\,k-1)\right] \quad (16)$$

Error Covariance Update (for each track):

$$P(k\,|\,k) = [I - K(k)\cdot H(k)]\cdot P(k\,|\,k-1) \quad (17)$$

Then, TSF procedure given in section 2.2 is applied for selecting best track with minimum likelihood.

3.2. Simulation Results

Simulations are performed for two modes of operation of the TAN algorithms:

- Tracking mode, where SITAN single filter and TSF are compared with Monte Carlo simulations along the trajectory;
- Acquisition mode, where TERCOM and TSF are compared with Monte Carlo simulations for the position update at a defined time.

First, simulation area is selected. For TAN algorithm applications, roughness and uniqueness of the selected terrain is very critical. It should be noted that the TAN algorithms will not work over all types of terrain. Good terrain must be more than just rough, it must be unique. In order to select terrain, two parameters are used; σ_T, the standard deviation of the terrain elevation samples which define terrain roughness and σ_Z, the standard deviation of the point-to-point changes in terrain elevation which define terrain uniqueness [3].

Terrain height profile and parameters used in the tracking mode simulations are given in Figure 6 and Table 1. Monte Carlo simulations of 100 runs are performed. Horizontal position errors and their RMS values are plotted for INS, SITAN and TSF for a single terrain. Simulation results are given in Figure 7, Figure 8 and Figure 9.

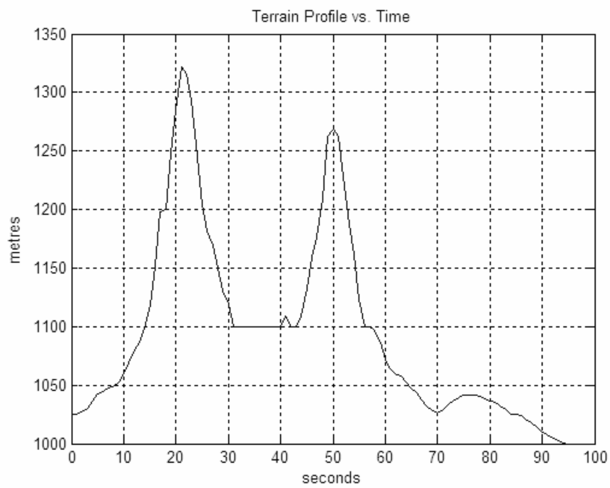

Figure 6: Terrain Height Profile for Tracking Mode

Table 1: Simulation Parameters for Tracking Mode

Initial INS position deviation (one axis)	60 m
Initial vehicle velocity	240 m/s
INS Quality (INS Class)	10 nmi/hr
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
INS horizontal position standard deviation	9 m
INS altitude position standard deviation	3 m
Radar altimeter standard deviation	3 m
INS velocity standard deviation	0.05 m/s
DTED Accuracy	DTED 1
DTED Grid Size (for PDAF and TSF)	3x3
Mean height of the terrain	1093 m
σ_T of the terrain	77.9 m
σ_Z of the terrain	16.2 m

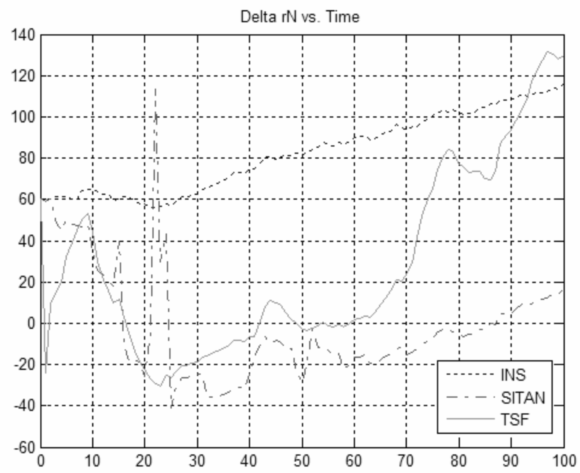

Figure 7: Northward Position Errors vs. Time

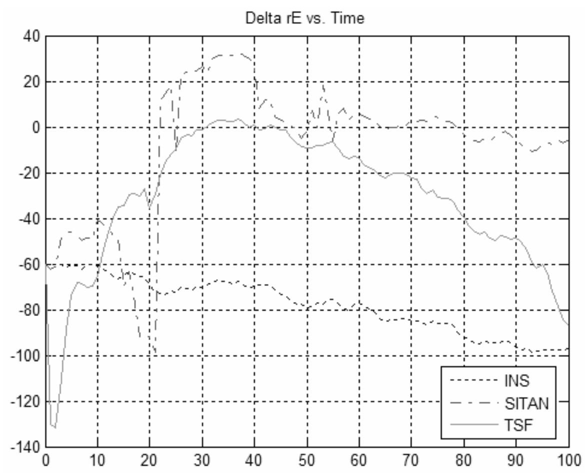

Figure 8: Eastward Position Errors vs. Time

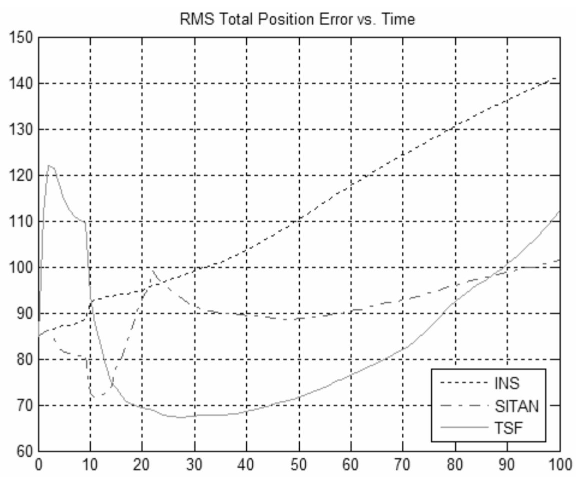

Figure 9: Total Position RMS Error vs. Time

As it can be seen from the results, TSF total RMS errors are better than SITAN results for tracking mode of operation. It should be noted that INS is not updated during the simulation period. However, for real-time operations, update will be performed in small time periods.

For acquisition mode of operation, simulations are performed with greater DTED batch sizes of 11 x 11 for TSF algorithm. In the acquisition mode simulations, same parameters given in Table 1 are used except initial INS position error and INS quality. Since, batch process is used for TERCOM, INS quality is taken as 1 nmi/hr class as in real TERCOM applications. Moreover, large initial position errors are used for simulations. Monte Carlo simulations of 100 runs are performed for each terrain. Percentage of false fixes and total horizontal errors for TERCOM and TSF are tabulated with initial INS horizontal errors. Standard deviations of total horizontal errors are also calculated considering Monte Carlo simulations. Six different terrains are used for simulations. Each terrain has different roughness and uniqueness parameters. Simulation results and terrain parameters for acquisition mode are given in Table 2.

Table 2: Terrain Parameters and Simulation Results for Acquisition Mode

Terrain No.	σ_T [m]	σ_Z [m]	Initial Total Horizontal INS Error and Standard Deviation [m]	Total Average Horizontal Error and Standard Deviation (after position fix) [m]		Percentage of False Fix [%]			
						Total Position Error (< 120 m)		Total Position Error (< 250 m)	
				TERCOM	TSF	TERCOM	TSF	TERCOM	TSF
1	32.4	1.7	456.3 / 57.9	62.1 / 27.0	63.3 / 27.2	30	22	12	6
2	25.7	1.6	453.7 / 59.8	66.0 / 27.9	64.8 / 29.0	43	44	22	9
3	157.4	19.5	444.9 / 50.2	41.0 / 19.4	49.2 / 27.9	5	3	5	0
4	176.2	6.5	446.9 / 53.7	42.2 / 19.1	44.8 / 20.8	6	2	6	2
5	736.4	21.6	449.6 / 50.3	58.8 / 32.0	55.8 / 27.8	16	4	11	0
6	308.7	21.1	451.9 / 54.0	39.4 / 18.4	42.5 / 21.4	11	5	11	2

As it can be seen from the results, percentage of false fix decreases considerably with the implemented TSF algorithm. Total average errors and standard deviations are calculated for TERCOM and TSF considering true position fixes. Position error results are similar compared with TERCOM. On the other hand, error values of most of the false fix positions are less than INS errors for TSF which is the main advantage of the proposed algorithm.

4.0 DISCUSSION AND CONCLUSION

In this paper, a radar tracking algorithm is implemented for TAN. Several conclusions are achieved from the implemented TSF algorithm. The advantages of the new algorithm proposed can be summarized as follows:

- Real-time TAN solution can be obtained with a single TSF structure. However, TSF operations are more complex than SITAN. On the other hand, in TSF, more than one track is selected in order to determine navigation solution. Hence, probability of false fix decreases unlike TERCOM.
- Real-time TAN solution is obtained by considering horizontal position errors of DTED used in real-time TSF. Hence, horizontal position states are added to the Kalman filters used in TSF.
- Application of the TSF is simple and the filter is linear since INS error model is used.
- Batch size of the DTED area concerned can be changed independent of the model used. Both larger DTED areas for acquisition mode or smaller DTED areas for tracking mode of operation can be selected using the same TSF structure.
- Results of the filters are good for both modes of operation. For tracking mode, position RMS error is less than 50 meters for each axis.
- TSF can be considered as a real-time TERCOM process for large position errors, i.e. large DTED batch size. Possibility of false position fixes decrease with TSF when compared with TERCOM. Moreover, required DTED batch and correlation time for correlation process is also decreased. On the other hand, for small position errors, decreasing the weighting factor of the past measurements for TSF, better real-time solutions can be obtained.

The main disadvantage of the proposed algorithm is calculation time. Due to track formation process in

TSF, number of calculations increase excessively. Actually, number of tracks is kept constant in TSF with selecting constant number of tracks for each time step. However, number of calculations is still considerable. Actually, with the use of new processors in navigation computers, TSF algorithm can be used in real-time applications.

REFERENCES

[1] Boozer, D., D. (1995). *Terrain Referenced Navigation*, AGARD-AG-331, *Aerospace Navigation Systems*, pp. 152-157.

[2] Johnson, N., Tang, W., Howell, G. (1990). *Terrain Aided Navigation Using Maximum A Posteriori Estimation.* IEEE, Position Location and Navigation Symposium, 1990.

[3] Siouris, G., M. (2004). *Missile Guidance and Control Systems*. Springer-Verlag New York, Inc., pp. 551-576.

[4] Hostetler, L., D., Andreas, R., D. (1983). *Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation.* IEEE Transactions on Automatic Control, Vol. AC-28, No. 3, March 1983.

[5] Uhlmann, J. (2001). *Introduction to the Algorithmics of Data Association in Multiple-Target Tracking. Handbook of Multisensor Data Fusion*, CRC Press LLC.

[6] Qingtang, F., Lincheng, S., Wenseng, C. (2003). *Terrain Aided Navigation Using PDAF.* Proceedings of the 2003 IEEE, International Conference on Robotics, Intelligent Systems and Signal Processing, China, 2003.

[7] Dezert, J. (1999). *Improvement of Strapdown Inertial Navigation using PDAF.* IEEE Transactions on Aerospace and Electronic Systems, Vol. 35, No. 3, July 1999.

[8] Stone, L., D. (2001). *A Bayesian Approach to Multiple-Target Tracking. Handbook of Multisensor Data Fusion*, CRC Press LLC.

[9] Reid, D., B. (1979). *An Algorithm for Tracking Multiple Targets.* IEEE Transactions on Automatic Control, Vol. AC-24, No. 6, Dec. 1979.

[10] Blackman, S., Popoli, R. (1999). *Design and Analysis of Modern Tracking Systems.* Artech House, ISBN No: 1-58053-006-0, pp. 360-402.

[11] Smith, P., B., Buechler, G. (1975). *A Branching Algorithm for Discriminating and Tracking Multiple Objects.* IEEE Transactions on Automatic Control, pp. 101-104, Feb. 1975.

[12] Durrant-Whyte, H. (2003). *Introduction to Estimation and Data Fusion.* Lecture Notes, Australian Centre for Field Robotics, The University of Sydney.

[13] Bar-Itzhack, I., Y., Berman, N. (1988). *Control Theoretic Approach to Inertial Navigation Systems.* AIAA Journal of Guidance, Vol. 13, pp. 237-245, May-June 1988.