TERRAIN REFERENCED UAV NAVIGATION WITH LIDAR – A COMPARISON OF SEQUENTAIL PROCESSING AND BATCH PROCESSING ALGORITHMS

Yeonha Hwang, Min-Jea Tahk
KAIST (Korea Advanced Institute of Science and Technology)
yhhwang@fdcl.kaist.ac.kr; mjtahk@fdcl.kaist.ac.kr

Keywords: *SITAN, TERCOM, DSMAC, Extended Kalman filter, Cross-Correlation Matching*

Abstract

This paper discusses terrain referenced navigation (TRN) system using light detection and ranging (LiDAR) and pressure altimeter in order to compensate for an Inertial Navigation system (INS) errors. The paper addresses sequential processing such as Sandia Inertial Terrain Aided Navigation (SITAN) and batch processing algorithms such as Digital Scene Matching Area Correlator (DSMAC) and carries out simulations with scenarios in order to compare the performance of these two processing algorithms and verify which one is more useful in each specific environment.

1 Introduction

An Inertial Navigation System (INS) has been used for estimating positions, velocities, and attitude angles of UAVs but it has disadvantages in that errors are accumulated as time goes by because of integrating measurements of acceleration and angular velocity obtained from an inertial measurement unit (IMU). For this reason, the INS generally employs Global Positioning System (GPS) in order to correct and calibrate itself through a Kalman filtering algorithm. However, GPS is easily affected by outside disturbance signals. As a result, TRN (Terrain Referenced Navigation), which is a technique that corrects the INS using the position measurement obtained through comparing the measured altitude data from a sensor with the stored digital elevation map (DEM)[5] regardless of outside disturbance signals, has been recently studied as an alternative method for INS correction. Thus, this technique can be used when GPS is not available or jammed.

There are two processing methods for a TRN system[2]. One is a sequential processing method such as SITAN (Sandia Inertial Terrain-Aided Navigation), which compares one measurement data with the stored DEM at the time when the sensor measures altitudes. The other is a batch processing such as Digital Scene Matching Area Correlator (DSMAC), which compares between sensed images and stored reference images to determine position measurements through the best match location of the image, and Terrain Contour Matching (TERCOM), which obtains the measurements through correlating a sensed terrain profile to a stored map terrain profile.

To measure the altitude data for TRN, a pressure altimeter, a radar altimeter and a LiDAR (Light Detection and Ranging) are needed. Comparing with the radar altimeter, the LiDAR has several advantages in that it can measure large areas that are difficult to approach with higher resolutions in a shorter period of time and construct a digital elevation map. The constructed DEM can be used for hazard avoidance as well as TRN.

Therefore, this paper proposes the TRN system using the LiDAR in order to compensate for INS errors. Section 2 describes the sequential processing method based on the LiDAR measurements and INS system and measurement modeling. Section 3 discusses the batch processing method with cross correlation matching. Section 4 explains EKF algorithm. Section 5 gives results of the simulations consisted of two different environments and comparison. Finally, section 6 gives conclusion.

2 Sequential processing method

This section explains SITAN as a sequential processing method and describes the system model.

2.1 SITAN

SITAN[2] is a sequential processing method which means when the sensor measures an altitude, the position of UAV is derived by correlating the altitude to the stored DEM at the time. An inclination of terrain is input into an Extended Kalman filter (EKF) which updates the UAV's INS.

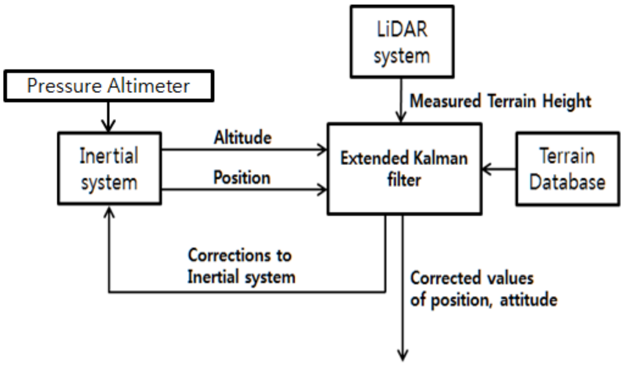


Fig. 1. SITAN system

2.2 INS system model

The states of system model are defined as 6 error states expressed the position and the velocity of UAV.

$$\mathrm{X} = [\delta x \quad \delta y \quad \delta h \quad \delta v_x \quad \delta v_y \quad \delta v_h]^T \quad (1)$$

where, $\delta x, \delta y, \delta h$ are the errors of x, y and z coordinates. δv_x , δv_y , δv_z are the errors of the velocities of UAV.

This error states are used in indirect kalman filter for INS correction.

Equation (2) and (3) are error state equations.

$$\dot{\mathrm{X}} = \boldsymbol{F}\mathrm{X} + \boldsymbol{G}\boldsymbol{W} \qquad (2)$$

$$\boldsymbol{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad (3)$$

where, $\boldsymbol{W}$ means random white noise of INS.

2.3 Measurement model[4]

The terrain-clearance measurement equation is a nonlinear function of x, y, z coordinates. To apply the EKF, the linearization of the measurement equation is needed.

$$\mathbf{z}_k = \hat{\mathbf{y}}_k - \mathbf{y}_k \qquad (4)$$
$$= \begin{bmatrix} \hat{h}_k - h_{DB}(\hat{x}, \hat{y}) \\ \hat{h}_k \end{bmatrix} - \begin{bmatrix} y_{LiDAR} \\ y_{pressure} \end{bmatrix}$$

$$\mathrm{H}_k = \begin{bmatrix} -\dfrac{\partial h_{DB}}{\partial x} & -\dfrac{\partial h_{DB}}{\partial y} & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad (5)$$

where, $\hat{\mathbf{y}}_k$ and $\mathbf{y}_k$ describe the estimated measurements and true measurements of z coordinate. $\hat{h}_k$ is estimated altitude and $h_{DB}(\hat{x}, \hat{y})$ is an elevation obtained in DEM when x and y coordinates are $(\hat{x}, \hat{y})$. y_{LiDAR} is a measurement by LiDAR and $y_{pressure}$ is a measurement by pressure altimeter. Finally, H_k is a measurement sensitivity matrix and the first and second rows mean the terrain slops of x and y coordinates.

3 Batch processing method

This section discusses DSMAC method and cross correlation.

3.1 DSMAC

DSMAC[3] is originally used for Tomahawk cruise missile in order to provide a reliable and precise measurement of location. During the missile flying, the current location of missile where the most well matched is determined by acquired images of the ground comparing with an existing image that is stored.

In this paper, as previously mentioned, altitude measurements obtained from the sensors such as LiDAR and pressure altimeter are used for generating terrain elevation map instead of images and compares with DEM in order to estimate the current position.

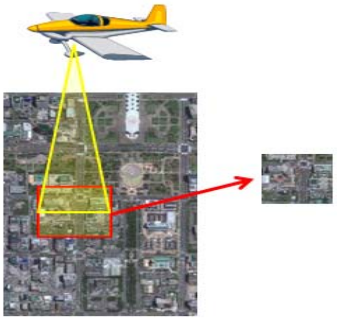


Fig 2. DSMAC operation

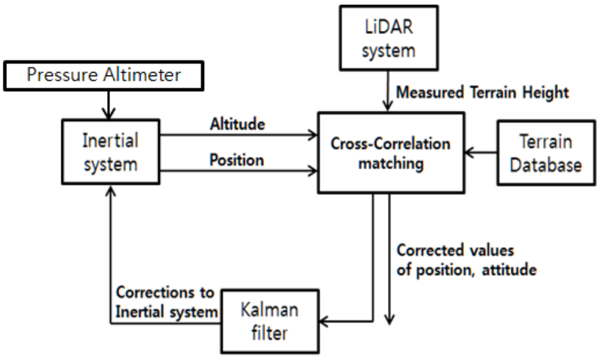
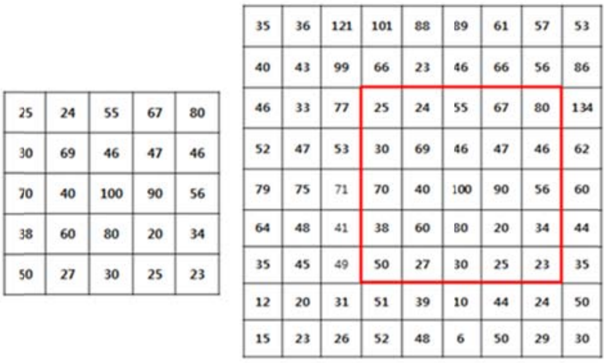


Fig 3. DSMAC system

3.2 Cross-Correlation matching[1]

Area-based matching technique is used to extract the most similar parts by analyzing the similarity between elevations measured by the LiDAR sensor and DEM. In other words, terrain elevation map generated by LiDAR is defined as a template window and then, the size of a search window is set as much as including the template window. While the template window is moving within the search window, the most well matched areas are found. This paper uses cross-correlation matching technique for area-based matching technique.



25	24	55	67	80
30	69	46	47	46
70	40	100	90	56
38	60	80	20	34
50	27	30	25	23

35	36	121	101	88	89	61	57	53
40	43	99	66	23	46	66	56	86
46	33	77	25	24	55	67	80	134
52	47	53	30	69	46	47	46	62
79	75	71	70	40	100	90	56	60
64	48	41	38	60	80	20	34	44
35	45	49	50	27	30	25	23	35
12	20	31	51	39	10	44	24	50
15	23	26	52	48	6	50	29	30

(a) template window (b) search window and matching window

Fig 4. Concept of Cross-Correlation matching

The first picture (a) is a template window through LiDAR. The second picture (b) is a search window in DEM and red areas mean a matching window.

Cross-correlation coefficient ($\rho, -1 \le \rho \le 1$) is defined as equation (8).

$$\rho = \frac{\sigma_{TM} = \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left[(h_T(x_i, y_j) - \widehat{h_T})(h_M(x_i, y_j) - \widehat{h_M})\right]}{n \bullet m - 1}}{\sigma_T = \sqrt{\dfrac{\sum_{i=1}^{n}\sum_{j=1}^{m}(h_T(x_i, y_j) - \widehat{h_T})^2}{n \bullet m - 1}} \sigma_M = \sqrt{\dfrac{\sum_{i=1}^{n}\sum_{j=1}^{m}(h_M(x_i, y_j) - \widehat{h_M})^2}{n \bullet m - 1}}} \tag{8}$$

where, σ_{TM} is covariance between a template window and a matching window and σ_T and σ_M are the standard deviations of the template window and the matching window.

Cross-correlation coefficient has a statistical characteristic so that the maximum cross correlation coefficient does not mean the best matched. Thus, when the cross correlation coefficient is denoted as a three-dimensional curved surface, the position where the partial derivative is zero in the three-dimensional curved surface is the best well-matched. This position can be the location of UAV estimated.

$$\begin{aligned} \rho &= c_1 x^3 + c_2 x^3 y + c_3 x^3 y^2 + c_4 x^3 y^3 \\ &+ c_5 x^2 + c_6 x^2 y + c_7 x^2 y^2 + c_8 x^2 y^3 \\ &+ c_9 x + c_{10} xy + c_{11} xy^2 + c_{12} xy^3 \\ &+ c_{13} + c_{14} y + c_{15} y^2 + c_{16} y^3 \end{aligned} \tag{9}$$

Using the three-dimensional surface approximation equation (9), the pixel position and any polynomial coefficients can express the cross correlation coefficient (10).

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \vdots \\ \rho_{25} \end{bmatrix} = \begin{bmatrix} x^3 & x^3y & x^3y^2 & \cdots & y^3 \\ x^3 & x^3y & x^3y^2 & \cdots & y^3 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x^3 & x^3y & x^3y^2 & \cdots & y^3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_{16} \end{bmatrix} \tag{10}$$

Through pseudo inverse, the polynomial coefficients can be obtained. The three-dimensional surface approximation equation can be calculated by substituting the polynomial coefficients for $c_1, c_2 \cdots c_{16}$.

The equation (11) is for finding the sub-pixel position when a partial differential of three-dimensional surface approximate equation is zero.

$$d\rho = \frac{\partial \rho}{\partial x}dx + \frac{\partial \rho}{\partial x}dy = 0 \tag{11}$$

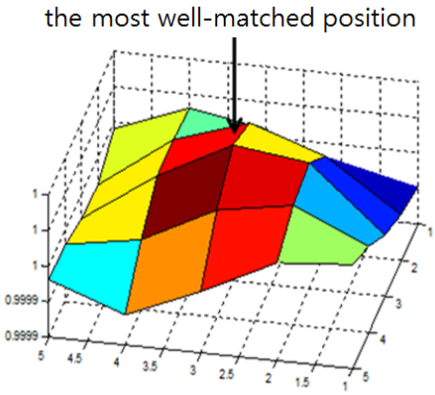


Fig 5. Position of maximum coefficient of correlation

3.3 Measurement model

True measurement of the x and y coordinates is calculated by Cross-Correlation matching using LiDAR.

$$\mathbf{z}_k = \hat{\mathbf{y}}_k - \mathbf{y}_k$$
$$= \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{h}_k \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \\ h_{pressure} \end{bmatrix} \tag{12}$$

where, $\hat{x}_k$, $\hat{y}_k$, $\hat{h}_k$ are x, y and z coordinates by INS. x_k , y_k are x and y coordinates by using cross-correlation matching. $h_{pressure}$ is the altitude measured by pressure altimeter. The true measurement is used in kalman filter.

4 Extended Kalman Filter

This section describes the EKF[4] used for this research simulation. Kalman filter is originally used in linear system. However, when kalman filter is applied to nonlinear system, extended kalman filter (EKF) is needed in instead of Kalman fitler. The filter method has two stages: one is a state propagation step, the other is a state update step. The equation (6) represents the propagation step for error states and error covariance.

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f(\hat{\mathbf{x}}_{k-1}^+) = \Phi_{k-1}\hat{\mathbf{x}}_{k-1}^+ \\ \hat{P}_k^- &= \Phi_{k-1}\hat{P}_{k-1}^+\Phi_{k-1}^T + \Phi_{k-1}GQ_{k-1}G^T\Phi_{k-1}^T\Delta t^2 \\ \Phi_{k-1} &= I + F\Delta t \end{aligned} \tag{6}$$

where, $\hat{\mathbf{x}}_k^-$ is an priori estimated error state and $\hat{\mathbf{x}}_k^+$ is an posterior estimated error state. $\hat{P}_k$ represents an error state covariance. Q_{k-1} represents a noise covariance matrix of INS. Φ_{k-1} is a state transition matrix. . The equation (7) represents the update step for error states and error covariance.

$$\begin{gathered} \hat{\mathbf{z}}_k = \mathrm{H}_k\hat{x}_k^- \\ K = \hat{P}_k^-\mathrm{H}_k^T(\mathrm{H}_k\hat{P}_k^-\mathrm{H}_k^T + R_k)^{-1} \\ \hat{x}_k^+ = \hat{x}_k^- + K(\mathbf{z}_k - \hat{\mathbf{z}}_k) \\ \hat{P}_k^+ = (I - K\mathrm{H}_k)\hat{P}_k^-(I - K\mathrm{H}_k)^T + KR_kK^T \end{gathered} \tag{7}$$

where, $\mathbf{z}_k$ is errors of true measurements which are introduced earlier in section 2.3 and $\hat{\mathbf{z}}_k$ is the estimated measurements. H_k is measurement sensitivity matrix. After obtaining kalman gain, $\hat{x}_k^+$ and $\hat{P}_k^+$ are calculated by substituting the kalman gain in equation (7).

Indirect kalman filter is used along with EKF so that $\hat{\mathbf{x}}_{k-1}^+$ is zero vector every steps.

5 Simulation results

5.1 Simulation environment

This section analyzes the simulation results of TRN system of UAV. The speed of UAV is set 90m/s. HG 1700 is used as the sensor of IMU and the noise of LiDAR and pressure altimeter are assumed each 5m.

5.2 Example 1

In this study, simulations perform by considering two different situations that initial position errors have small or large values. DSMAC and SITAN methods are compared to each other with the two different situations. If the initial position error is bigger than 90m, the position could have divergence because the grid size of DEM is 90m and the terrain inclination varies discretely. In this section, x, y and z coordinates errors are assumed 10m, 5m, and 2m.

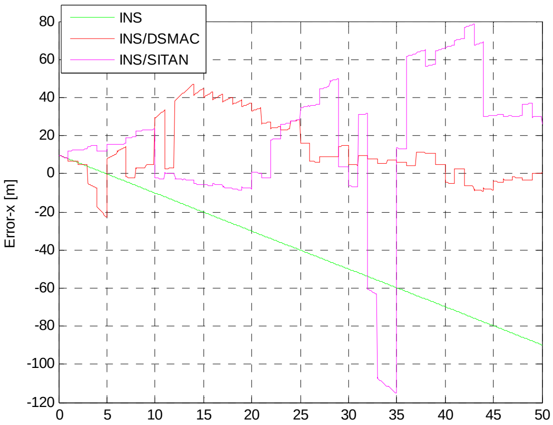


Fig 6. X coordinate error history

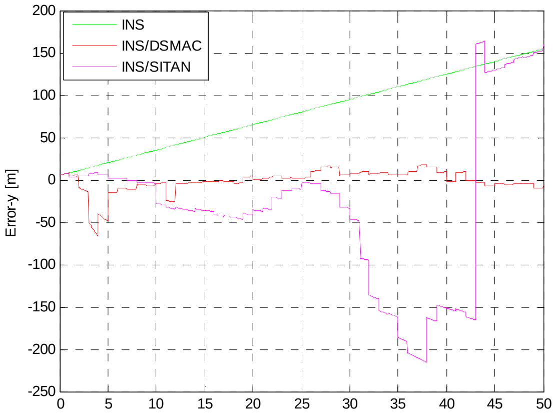


Fig 7. Y coordinate error history

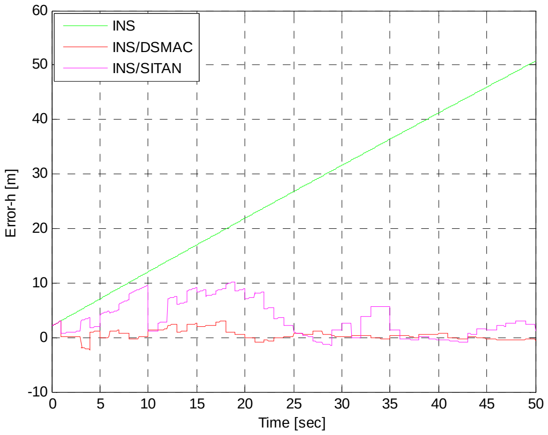


Fig 8. Z coordinate error history

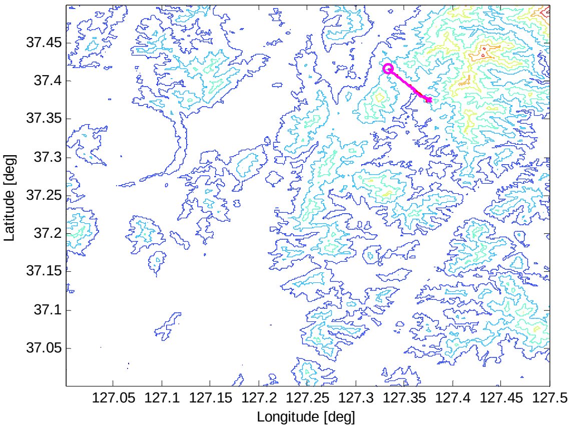


Fig 9. Start and End location of simulation

Figure 6-8 are SITAN and DSMAC performance. The positions through SITAN and DSMAC follow reference positions with small or large errors. The results represent a disadvantage of SITAN[2] which is high probability of divergence due to the highly nonlinear characteristic of terrain and use of only one measurement. To make sure to compare to each methods, mean values of errors are represented in table 1.

Table 1. Mean for TRN with initial small errors

	x(m)	y(m)	z(m)
SITAN	16.40	-30.36	3.3823
DSMAC	12.02	-1.54	0.4673

Averagely, the errors of SITAN are seen to be larger than the errors of DSMAC according to table 1.

5.3 Example 2

. In this section, x, y and z coordinates errors are assumed 150m, 100m, and 2m.

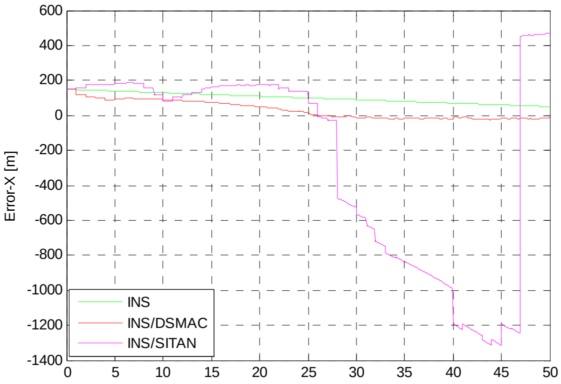


Fig 10 X coordinate error history

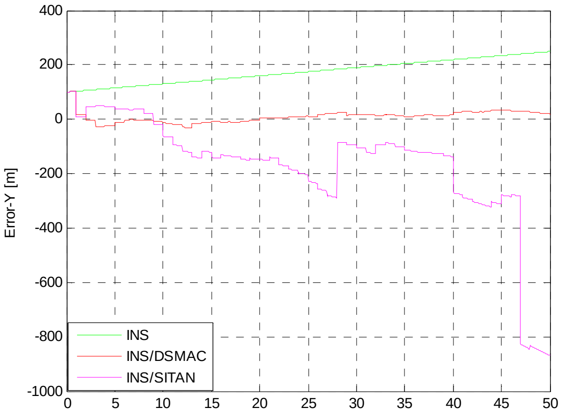


Fig 11. . Y coordinate error history

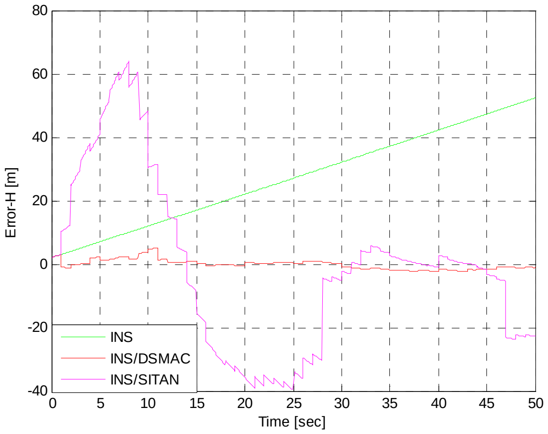


Fig 12. Z coordinate error history

Figure 10-12 are SITAN and DSMAC performance with large error. As shown in figures, the SITAN performance with large errors is showed a tendency of divergence and it is more distinct than the SITAN performance with small errors. Table 2 represents the errors means for TRN with initial large errors.

Table 2. Mean for TRN with initial large errors

	x(m)	y(m)	z(m)
SITAN	-249.29	-168.39	0.77
DSMAC	31.45	9.05	-0.10

6 Conclusion

This research carries out to verify the performance between the two methods (sequential processing *and* batch processing) for initial error conditions to analyze each method is more useful in each specific environment. Through the simulations, the batch processing method has a better performance than the sequential processing method in the whole simulation when the initial errors of the INS are small or large.

References

[1] Bo-mi lee *Terrain Referenced Navigation using Area-based Matching Algorithm*, Engineering Master's Thesis, University of Seoul, 2009

[2] Maarten Uijt de Haag *Application of Laser Range Scanner Based Terrain Referenced Navigation Systems for Aircraft Guidance*, Proceedings of the

Third IEEE International Workshop on Electronic Design, Test and Applications(DELTA'06) 2005

[3] Frederick W. *Guidance and Navigation in the Global Engagement Department*, JOHNS HOPKINS APL TECHNICAL DIGEST, VOLUME 29, NUMBER 2, 2010

[4] Sunghoon Mok, H.C.Bang *A Performance Analysis of Unscented Kalman Filter Based Terrain Referenced Navigation*, KSAS, P551-P556

[5] The CGIAR Consortium for Spatial Information (CGIAR-CSI) http://srtm.csi.cgiar.org/SELECTION/inputCoord.asp

Copyright Statement