

**Military Technical
College
Kobry El-Kobbah,
Cairo, Egypt**



**8th International
Conference on Electrical
Engineering**

ICEENG 2012

Fusion of Low-Cost MEMS IMU/GPS Integrated Navigation System

A. D. Abosekeen¹, A. E. Abdalla²

Abstract

In this paper, the multi-sensor data fusion technique based on fuzzy clustering is used to fuse the data from low cost MEMS IMUs to build an INS model. Using this model, the inertial navigation data PVA is extracted. The navigation data PVA is integrated with the GPS data using Kalman filters to build an accurate navigation system of an UAV. Simulation results show that the method can achieve higher accuracy solutions with low cost IMU sensors and improve the performance of integrated navigation system.

Key words

Data Fusion, fuzzy, Data Clustering, Inertial Navigation system (INS), Global Navigation system (GPS), Kalman Filter (KF), and INS/GPS Integration.

1. Introduction

In recent years, obvious attention has been focused on multi-sensor data fusion for military applications like automated target recognition (smart weapons),

¹Master Student, M.T.C, Egyptian Air Forces

²Assoc. Prof, Dean of M.T.C, M.T.C, Egyptian Air Forces

Navigation, guidance for autonomous vehicles, remote sensing, battlefield surveillance, and automated threat recognition systems, such as identification-friend-foe-neutral(IFFN)systems, also forcivilian applicationsas monitoring of manufacturing processes, condition- based maintenance of complex machinery, robotics, and medical applications.Data fusion techniques combine data from multiple sensors and related information to achieve more specific inferences than could be achieved by using a single, independent sensor[1].Fusion processes can be categorized into three levels of modes (low, Intermediate, and Highlevel fusion), as follows[2]:

1. Lowlevel fusion where several sources of essentially the same type of raw data are combinedto produce a new data set that is expected to be more informative and useful than the inputs.
2. Intermediatelevel fusion or featurelevel fusion which combinesvarious features into a feature map.
3. Highlevel fusion or decision fusion which combines decisions fromseveral experts. Methods of decision fusion are voting, fuzzylogic, and statistical methods.

The integration between GPS and INS exploitstheir synergy in various approaches, based on the useof KF, with the goal to mitigate the short time errorof GPS and longtime error of INS[3]. Theresulting plant is a combined navigation system thathas better performance than GPS or INS, consideredas stand-alone navigation systems. In order to calculate the estimate of INS error,KF constantly updated by the information from INS and GPS.Using multi-sensor data fusion algorithm to fuse the raw data from several IMUin INS will improve the integration process between the INS and the GPS systems. In this paper the effect of using multi-sensor data fusion algorithm on the INS/GPS process is studied using aerosonde UAV model see Figure (1)[4].The reference navigation data (PVA), and gyroscopes (gyros) rates and accelerations are extracted from the aerosonde UAV model.

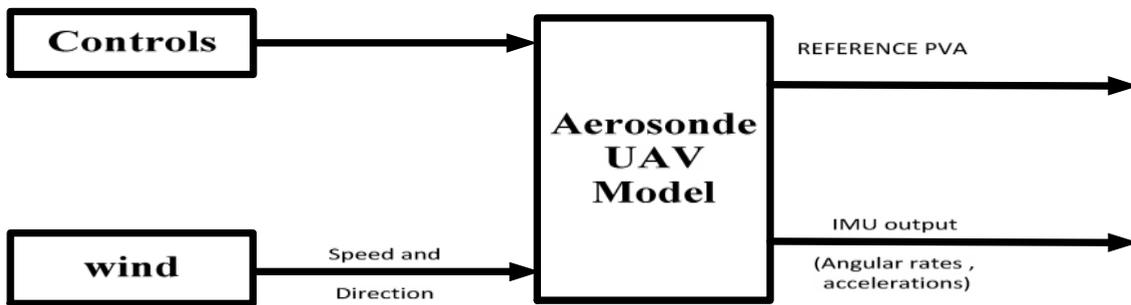


Figure 1 Aerosonde UAV Model

2. IMU Error Model

There are several sources of INS errors that can be grouped into deterministic and stochastic errors. The deterministic errors include bias offset, scale factor and axes misalignment errors. The bias offset is a constant error within the sensor measurement. The scale factor is the relationship between the output of the signal of the sensor and the physical quantity being measured. The axes misalignment is the error resulting from the imperfection of mounting the sensors. This misalignment has a deterministic and a stochastic error. The deterministic errors mentioned above can be corrected once they have been properly determined by standard calibration procedures. The stochastic errors include bias drift, bias stability, scale factor stability, noise and axes misalignment errors. Bias drift can be seen when the measurement drifts from the observable with a relationship that may be linear, quadratic, or of a higher order. The bias stability is the maximum expected drift in a sensor output. This can be measured as a function of time or as a function of temperature. Scale factor stability is the capability of the inertial sensor to accurately sense linear accelerations and angular velocities. Deviations from the theoretical scale are due to system imperfections. The noise results from the sensor itself that interferes with the output signals. Several stochastic models are available for modeling these errors[5]. For simulating IMU measurement, the IMU noise model consists of (gyroscope noise model + accelerometer noise model).

- The gyro measurement model is[6, 7]:

$$\tilde{\omega}_{ib}^b = (I_{3 \times 3} + s_g) \omega_{ib}^b + b_g + \eta_{gv} \quad (1)$$

$$\dot{\tilde{b}}_g^x = \eta_{gu} \quad (2)$$

Where: b_g is the gyro bias, s_g is a diagonal matrix of gyro scale factors, η_{gv} and η_{gu} are zero-mean Gaussian white-noise processes with spectral densities given by $\sigma_{gv}^2 I_{3 \times 3}$ and $\sigma_{gu}^2 I_{3 \times 3}$, respectively.

- The accelerometer measurement model is [6, 7]:

$$\tilde{a}^b = (I_{3 \times 3} + s_a) a^b + b_a + \eta_{av} \quad (3)$$

$$\dot{\tilde{b}}_a^x = \eta_{au} \quad (4)$$

Where: b_a is the accelerometer bias, s_a is a diagonal matrix of accelerometer scale factors η_{av} , and η_{au} are zero-mean Gaussian white-noise processes with spectral densities given by $\sigma_{av}^2 I_{3 \times 3}$ and $\sigma_{au}^2 I_{3 \times 3}$ respectively.

A discrete-time simulation of gyro model using the spectral densities is shown in **Error! Reference source not found.**

Where: $N = (0, \sigma^2)$ denotes a zero-mean normal distribution with σ^2 . The same model can be used for the accelerometer[6, 7].

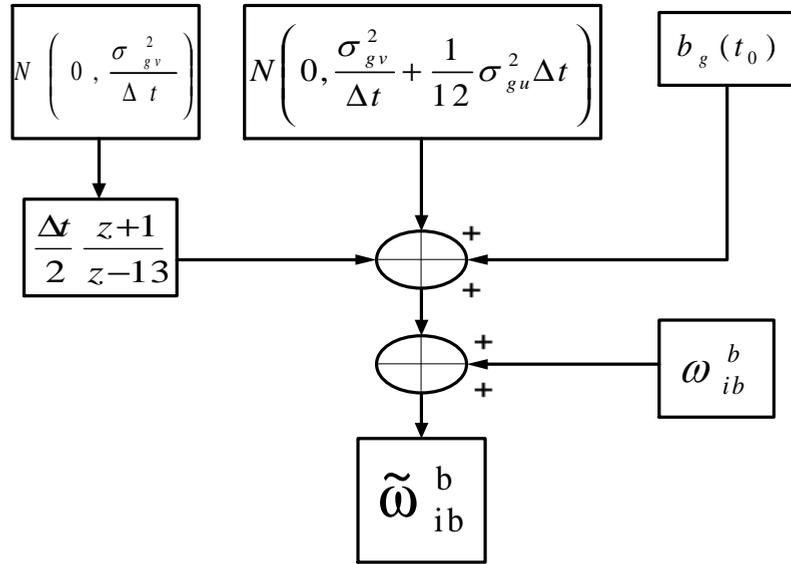


Figure (2)Gyro Error Model

3. Multi-sensor data fusion

data fusion techniques combine data from multiple sensors and related information to achieve more specific inferences than could be achieved by using a single, independent sensor[1].Using multi-sensor data fusion in aircraft navigation applications has appeared in recent years with the advent of [Low cost, small size and low mass navigation sensors (e.g. optical gyros, MEMS inertial sensors and GNSS sensors)][8].In this paper The multi-sensor data fusion technique is based on fuzzyc-means (FCM).

4. FCM Algorithm

The FCM clustering algorithm is based on the minimization of an objective function called *C-means* functional. This function can be defined as in Eq.(6)[9-16].

$$J(X; U, V) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|x_k - v_i\|^2 \tag{6}$$

The FCM algorithm shown in **Error! Reference source not found.**) computes with the standard Euclidean distance norm, which induces hyperspherical clusters. Hence it can only

detect clusters with the same shape and orientation, because the common choice of norm inducing matrix is $A = I$ [13].

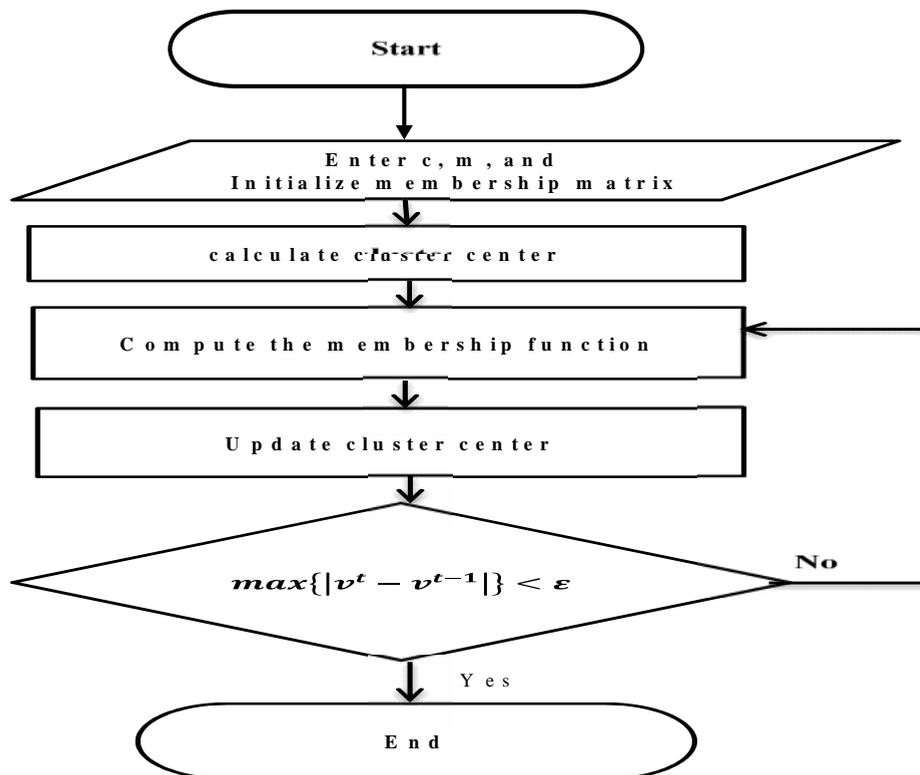


Figure (3) FCM Algorithm

Three parameters in this algorithm have to be determined in the beginning which they are $[c, m]$, they represent the number of clusters, the weighting parameter that represent the fuzziness of the system, and the ending threshold respectively.

The Algorithm can be explained as follows:

For a given dataset

Step1 Enter

- Number of clusters $1 < c < N$,
- The weighting exponent $m > 1$,
- The termination tolerance > 0 .
- Initialize the membership matrix randomly

Repeat for $l = 1, 2, \dots$

Step 2 Compute Cluster Centers

$$v_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, 1 \leq i \leq c \quad (7)$$

Step 3 Compute Distances (Euclidian distance)

$$D_{ikA}^2 = (x_k - v_i)^T A (x_k - v_i) \quad (8)$$

Step 4 Compute the membership matrix

Repeat until

$$\max |v^{(l)} - v^{(l-1)}| \leq \varepsilon \quad (9)$$

5. Fuzzy Clustering as Data Fusion Technique

The data fusion technique using the fuzzy clustering algorithm is shown in **Error! Reference source not found.** This technique consists of three main parts:

1. Collecting data from the environment by using multiple sensors.
2. Separating the collected data by using the FCM algorithm.
3. The output from FCM applied to the fusion mechanism to produce a single output.

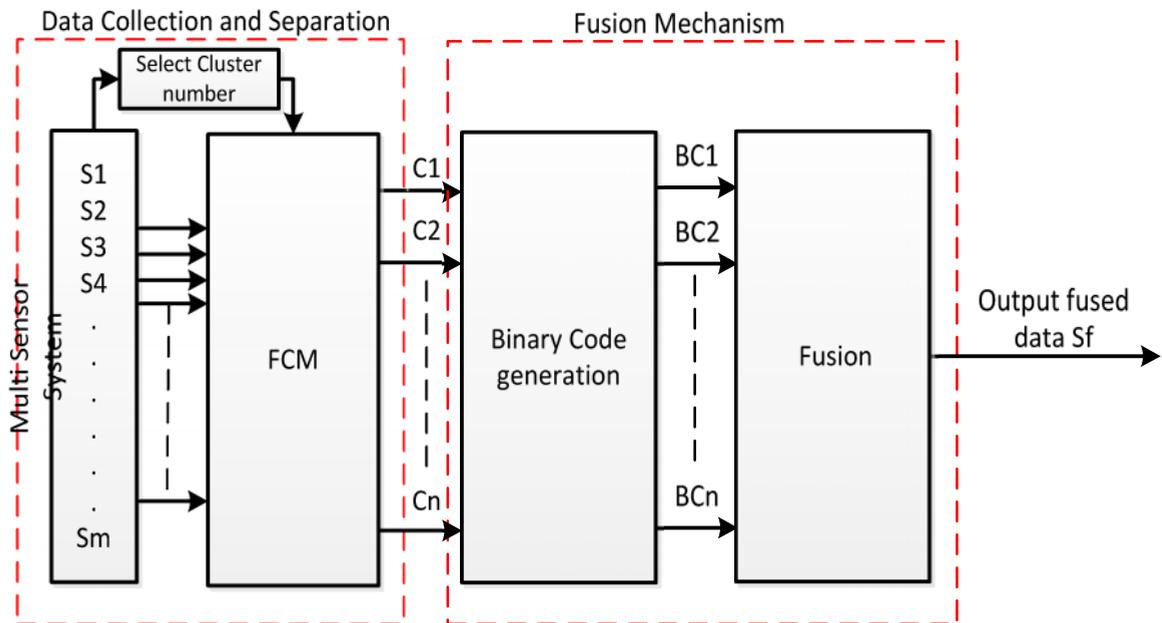


Figure 4 Data Fusion Mechanization

The first part is concerned with setting the sensors for measuring a particular phenomenon from the environment ,the second part uses the FCM to separate the data collected by the sensors to a certain number of cluster with membership matrix and cluster centers ,the third part deals with the output clusters and membership functions by generating a binary code this binary code is representing the membership function of the clusters and generated by comparing the membership function with a certain threshold value ,and the minimum binary code denote to the cluster with the center that represent the fusion process output.

6. Selection of Cluster Number

According to the FCM algorithm it is required to predefine the number of clusters (c), This number can be calculated with many methods such as the validation parameters but only in offline mode, or by The smooth kernel density estimator (SKDE) which is powerful in the real time and can be obtained as follows[17, 18]:

$$p(s) = \frac{1}{mh} \sum_{i=1}^m k \frac{(s - s_i)}{h}$$

And (10)

$$k(s) = \frac{1}{(2\pi)^{1/2}} e^{(-\frac{1}{2}s^T s)}$$

Where : h is the length of the estimation window.

The main idea is that the measurements values drifted in two directions around the acceptable region of measurements as shown in Figure (5), so the number of clusters have to be determined every instance of measurement. From the figure below the partitions may be three ($c=3$) if the drift was in two directions from the accepted region or may be two partitions ($c=2$) if the drift at any instance were to the left or to the right direction (one direction drift)[19].

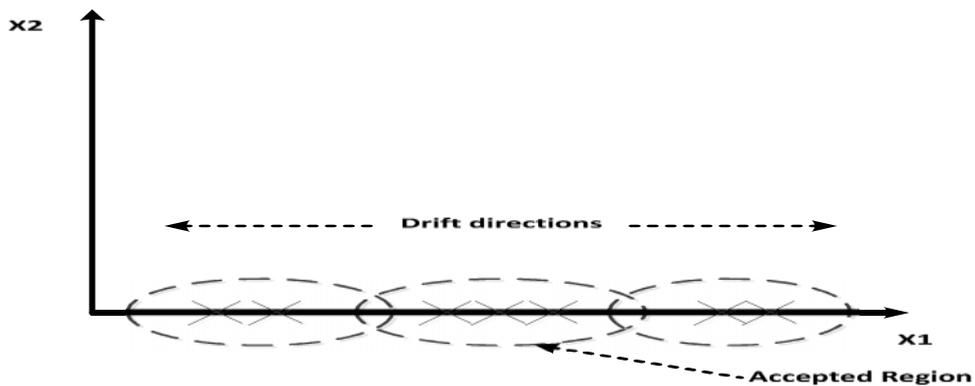


Figure (5) partitioning for the measurements

Smooth Kernel density estimator (SKDE) for a given inputs of measurements is shown in Figure (6), where the maximum peak in the figure is denoted as *Max* and located by the dotted lines.

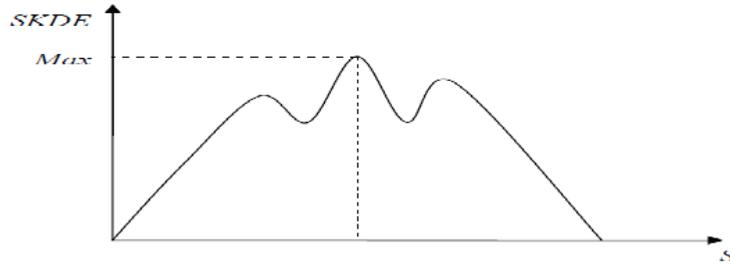


Figure (6) SKDE Output

Subsequently, the number of clusters is determined according to the following two rules, based on the maximum peak location of the kernel estimator:

1. If the maximum peak of the SKDE is left or right skewed then $C = 2$.
2. If the maximum peak of the SKDE is centered then $C = 3$.

7. Fusion Mechanism

After the clustering process as in **Error! Reference source not found.**4) each cluster membership function is represented as a binary code B_{C_i} . The creation of this code depends upon the membership functions for the clusters and a variable threshold level such that [20].

$$B_{C_i}(S) = \begin{cases} 1, & \text{if } \mu(s) < \alpha \\ 0, & \text{if } \mu(s) > \alpha \end{cases} \quad (11)$$

Where () is given as follows:

$$\alpha = 1 - \frac{1}{m} \sum_{j=1}^c \sum_{i=1}^m \mu_j(s_i) \log \mu_j(s_i) \quad (12)$$

The first term in the equation above is used for mapping the values, while the second term is an entropy measure for the resulted clustering process from previous section. The fused output will be the cluster center that achieves the following minimization argument for the generated binary codes:

$$s_f = v(\mu_{i^*}) \quad ,$$

$$i^* = \operatorname{argmin}(B_{c1}, B_{c2}, \dots, B_{ci}) \quad (13)$$

8. Kalman filter

The Kalman filter can be divided into two stages, the time update (prediction), and the measurement update (Correction). In the former as shown in Figure (), the Kalman gain K_k is computed first, and then the state and the error covariance are updated using the prior estimate $\hat{\chi}_k^-$ and its error covariance P_k^- . In the prediction the state is predicted with the dynamic model, and in the correction(update) predicted value is corrected with the observation(measurement) model, so that the error covariance of the estimator is minimized[21, 22].

The different INS error states can be described using the following 1st order state equations:

$$\dot{\chi} = F \chi + G u \quad (14)$$

Where: F is the dynamics matrix (state matrix), χ is the state vector, u is the forcing vector function (input vector), and G is a design matrix (input matrix). These terms are described in details as:

$$\chi = \begin{pmatrix} \delta r^n \\ \delta v^n \\ \delta q_b^n \end{pmatrix} = [\delta\varphi \quad \delta\lambda \quad \delta h \quad \delta v_N \quad \delta v_E \quad \delta v_D \quad \delta q_0 \quad \delta q_1 \quad \delta q_2 \quad \delta q_3]^T \quad (15)$$

$$u = \begin{pmatrix} \delta f^b \\ \delta \omega_{ib}^b \end{pmatrix} = [\mathcal{F}_x \quad \mathcal{F}_y \quad \mathcal{F}_z \quad \delta\omega_x \quad \delta\omega_y \quad \delta\omega_z]^T \quad (16)$$

$$F = \begin{pmatrix} F_{RR} & F_{RV} & 0_{3 \times 4} \\ F_{VR} & F_{VV} & F_{VAQ} \\ 0_{4 \times 4} & 0_{4 \times 4} & F_{AAQ} \end{pmatrix}, G = \begin{pmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ G_{AVQ} & 0_{3 \times 3} \\ 0_{4 \times 3} & G_{AAQ} \end{pmatrix} \quad (17)$$

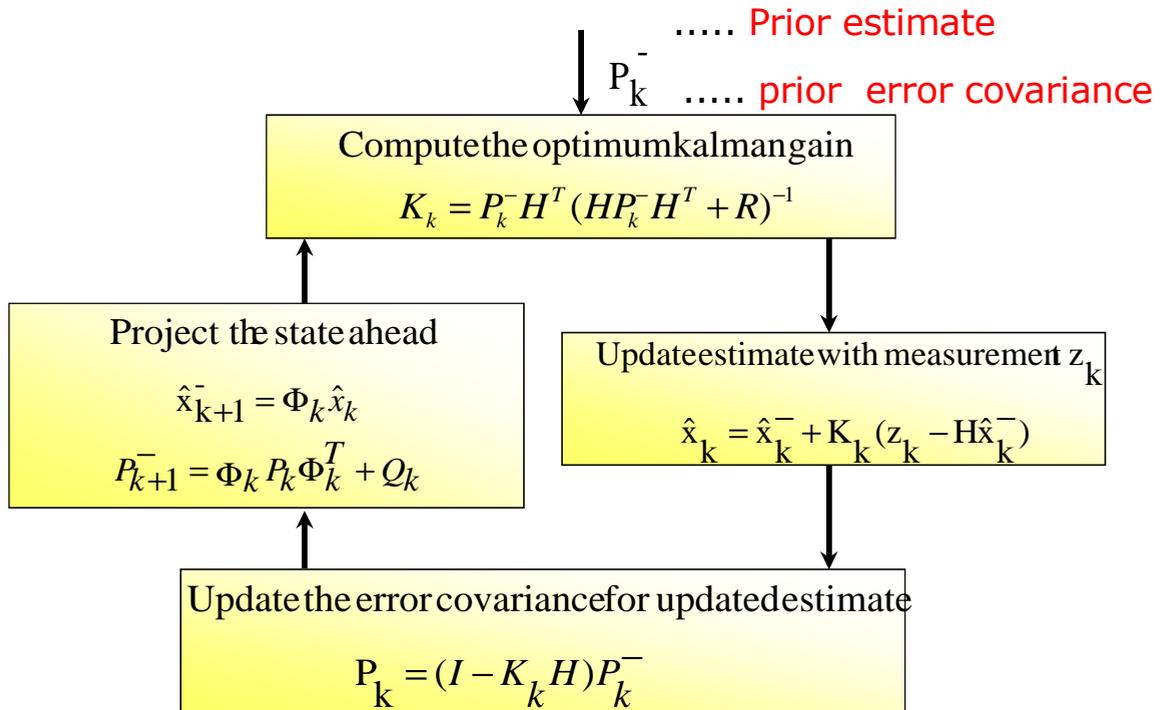


Figure (7) Kalman Filter Algorithm

8.1 Lever-arm Correction

Since the sensors cannot be installed at the same place in the host. The position and velocity of the IMU are different from those of the GPS. This is called the lever-arm effect. The lever-arm correction for the position and velocity can be written as [21]

$$\underline{r}_{IMU}^n = \underline{r}_{GPS}^n - \begin{bmatrix} 1 & 0 & 0 \\ M+h & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} C_b^n \Delta \underline{r}^b \quad (18)$$

And,

$$\underline{v}_{IMU}^n = \underline{v}_{GPS}^n - C_b^n \Omega_{nb}^n \Delta \underline{r}^b$$

Where: h is the height of moving object, R is the meridian radius in meters, r is the prime vertical radius in meters, \mathbf{d} is the offset vector of the GPS antenna or any other IMU from the center of the IMU placed in the CG of the moving object in the body frame, \mathbf{C}_b^n is the transformation matrix from body to navigation frame, and \mathbf{S} is the skew-symmetric matrix of $\boldsymbol{\omega}_b^n$ that represents the projection of the rotating rate vector of the B-frame with respect to the N-frame on the N-frame.

9. The Proposed Scenario and Simulation Results

In this section the steps of the complete process is clarified in the block diagram in Figure (8).

9.1 The Simulation Parameters and Results

1. The GPS
 - Simulation time =300 seconds.
 - Sampling time = 1 second.
 - Masking angle = 5 degrees.
2. The simulation parameters that make the first IMU measurements be as MEMS grade IMU are as follows:
 - Simulation time $T=300$ seconds and the sampling time $T_s=0.01$ seconds.
 - Initial biases for the gyros are $b_g = 10$ deg/hr .
 - Initial biases for accelerometers are $b_a = 0.003$ m/s² .
 - The gyro noise parameters $[\sigma_{g_v} = 8.7266 \times 10^{-7}$ rad/sec^{1/2}, $\sigma_{g_u} = 9.1989 \times 10^{-7}$ rad/sec^{3/2}].
 - The accelerometer parameters $[\sigma_{a_v} = 1.5 \times 10^{-5}$ m/sec^{3/2}, $\sigma_{a_u} = 6 \times 10^{-5}$ m/sec^{5/2}].

3. The simulation parameters that make the first IMU measurements be as MEMS grade IMU are as follows:
 - Simulation time $T=300$ seconds and the sampling time $T_s=0.01$ seconds.
 - Initial biases for the gyros are $b_g = 15$ deg/hr .
 - Initial biases for accelerometers are $b_a = 0.006$ m/s².
 - The gyro noise parameters $[\sigma_{gv} = 7.7266 \times 10^{-6}$ rad/sec^{1/2}, $\sigma_{gu} = 7.1989 \times 10^{-6}$ rad/sec^{3/2}].
 - The accelerometer parameters $[\sigma_{av} = 2.6 \times 10^{-5}$ m/sec^{3/2}, $\sigma_{au} = 7 \times 10^{-5}$ m/sec^{5/2}].

4. The simulation parameters that make the first IMU measurements be as MEMS grade IMU are as follows:
 - Simulation time $T=300$ seconds and the sampling time $T_s=0.01$ seconds.
 - Initial biases for the gyros are $b_g = 20$ deg/hr .
 - Initial biases for accelerometers are $b_a = 0.009$ m/s².
 - The gyro noise parameters $[\sigma_{gv} = 9.7266 \times 10^{-8}$ rad/sec^{1/2}, $\sigma_{gu} = 8.1989 \times 10^{-8}$ rad/sec^{3/2}].
 - The accelerometer parameters $[\sigma_{av} = 1.9 \times 10^{-5}$ m/sec^{3/2}, $\sigma_{au} = 8 \times 10^{-5}$ m/sec^{5/2}].

5. Assuming that the Lever-arm between the used sensors is as follows:
 - Lever-arm from the first IMU to GPS in body frame: X=1.72m, Y=0.30m, Z=-1.49m
 - Lever-arm from the second IMU to GPS in body frame: X=1.59m, Y=0.30m, Z=-1.49m
 - Lever-arm from the third IMU to GPS in body frame: X=1.99m, Y=0.30m, Z=-1.49m

And the first IMU is assumed is the one placed at the CG of the moving vehicle.

The process is also explained at the block diagram in Figure (8).

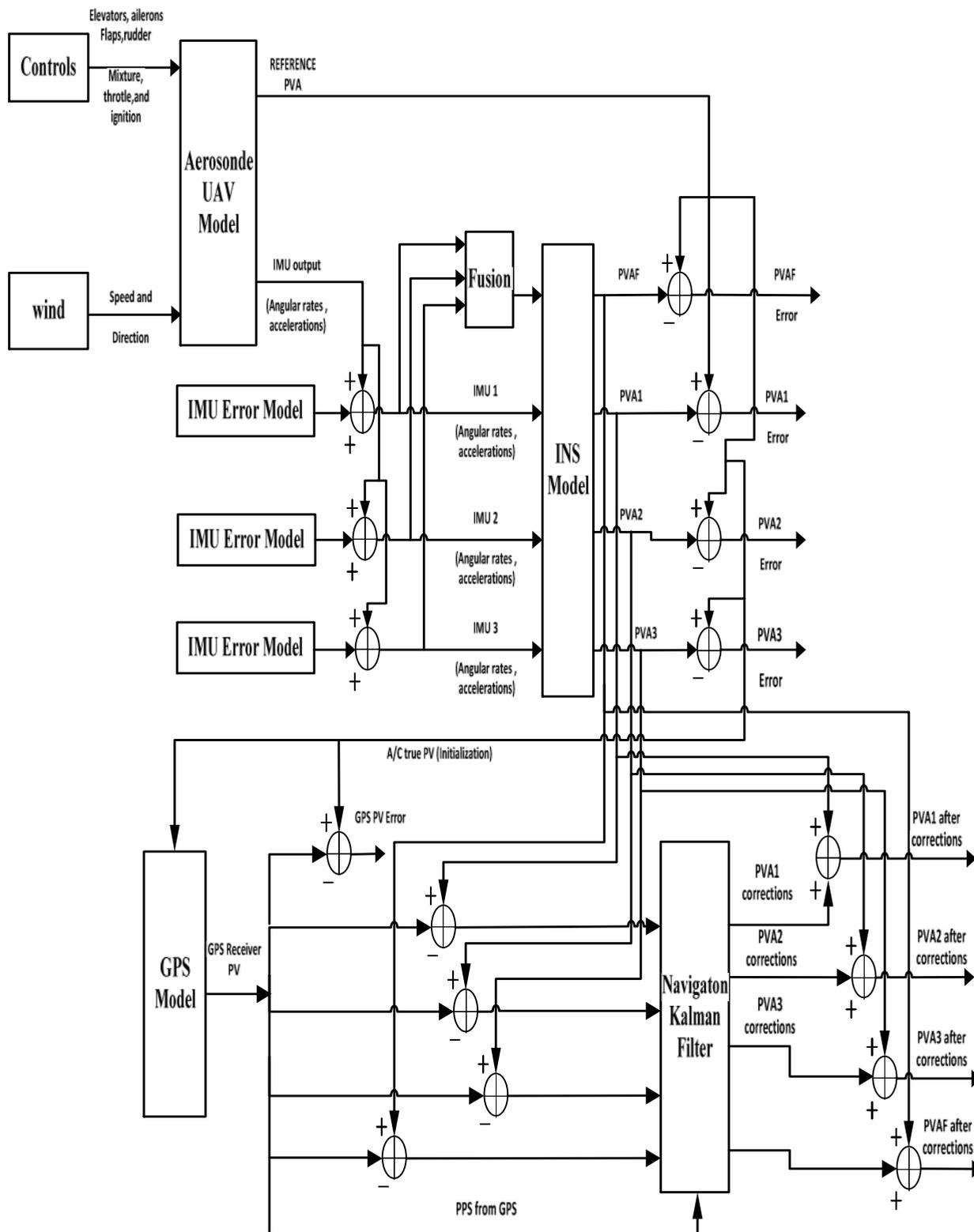


Figure (8)Block Diagram of Applying the Fusion and the Integration Algorithms

on the Navigation Process of Aerosonde UAV

9.1.1 Reference (P, V, A)

The reference position components in N-frame from the UAV model is shown in Figure (9), the reference velocity components in N-frame is shown in Figure (10), and the reference attitude components is shown in Figure (11).

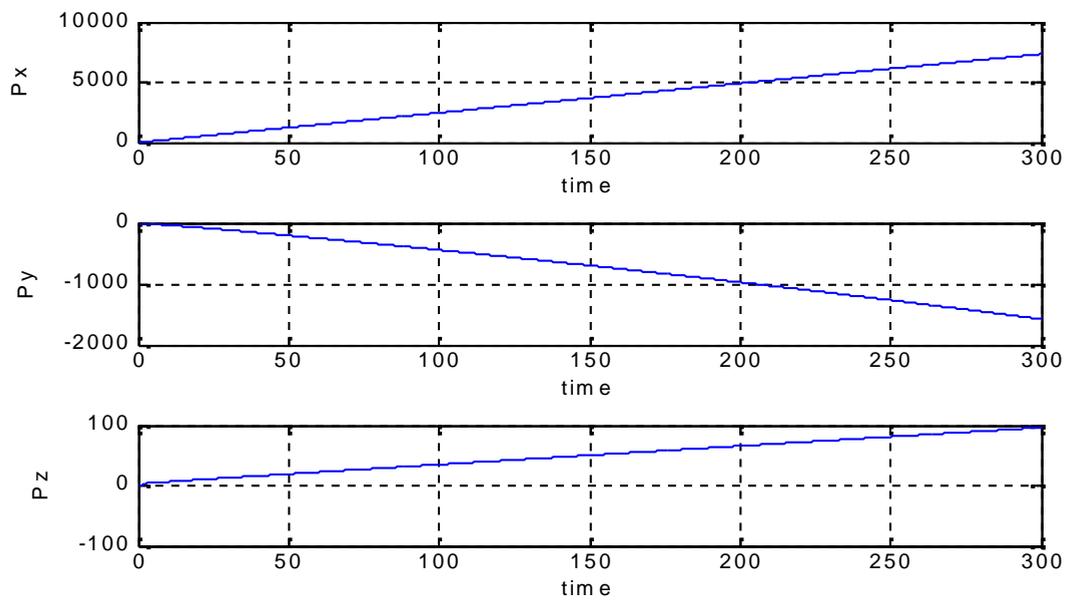


Figure (9) Reference Position Components in N-frame (Px, Py, Pz)

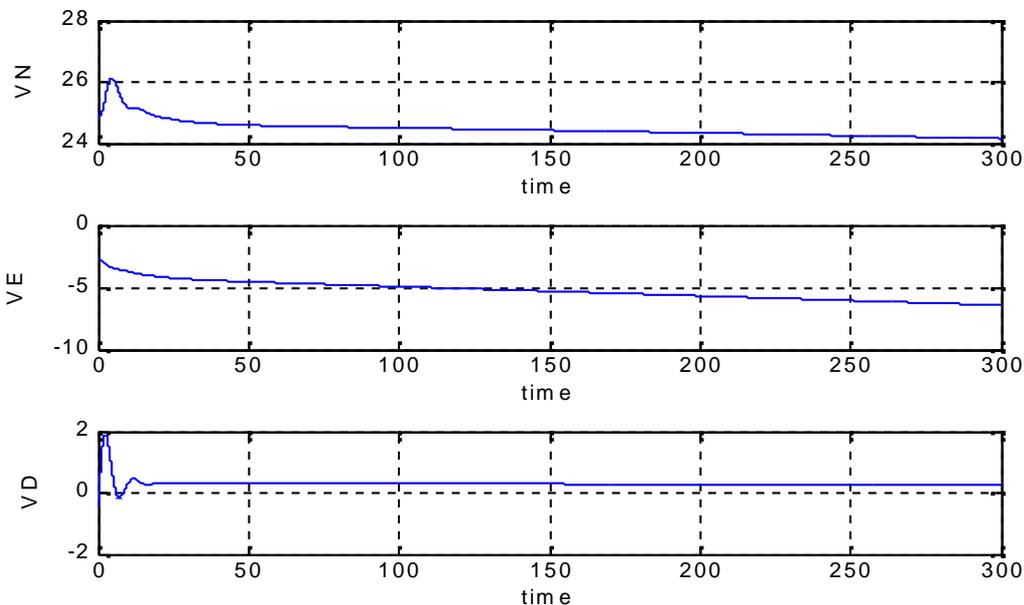


Figure (10) Reference velocity Components in N-frame (VN, VE, VD)

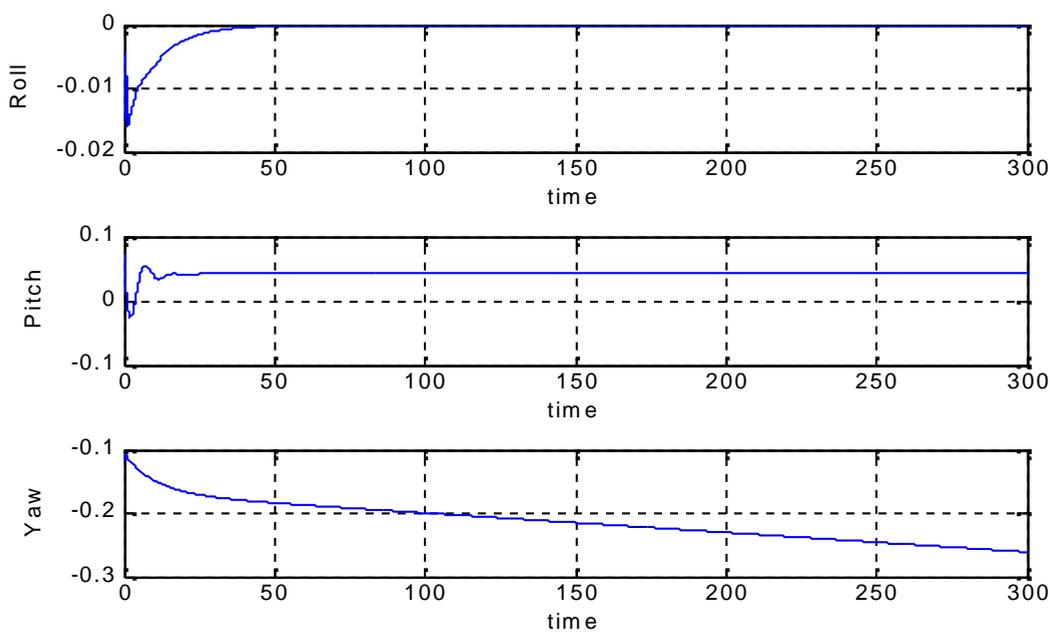


Figure (11) Reference Attitude Components (Roll, Pitch, Yaw)

9.1.2 Calculation of (P, V, A) RMSE using the INS

a) The Position RMSE

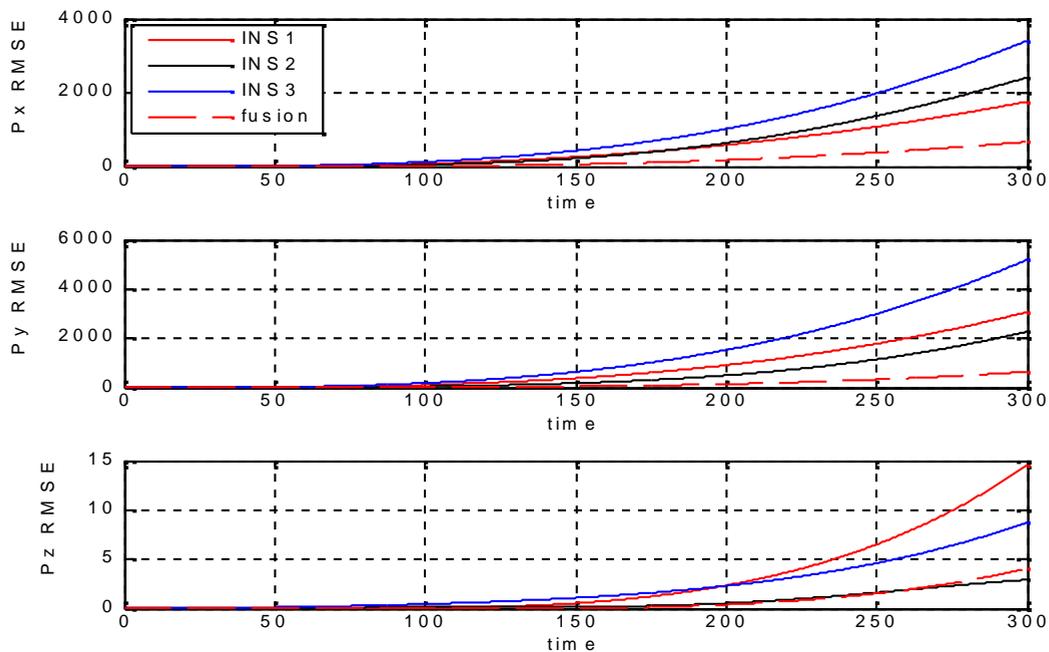


Figure (12) Position RMSE

b) The Velocity RMSE

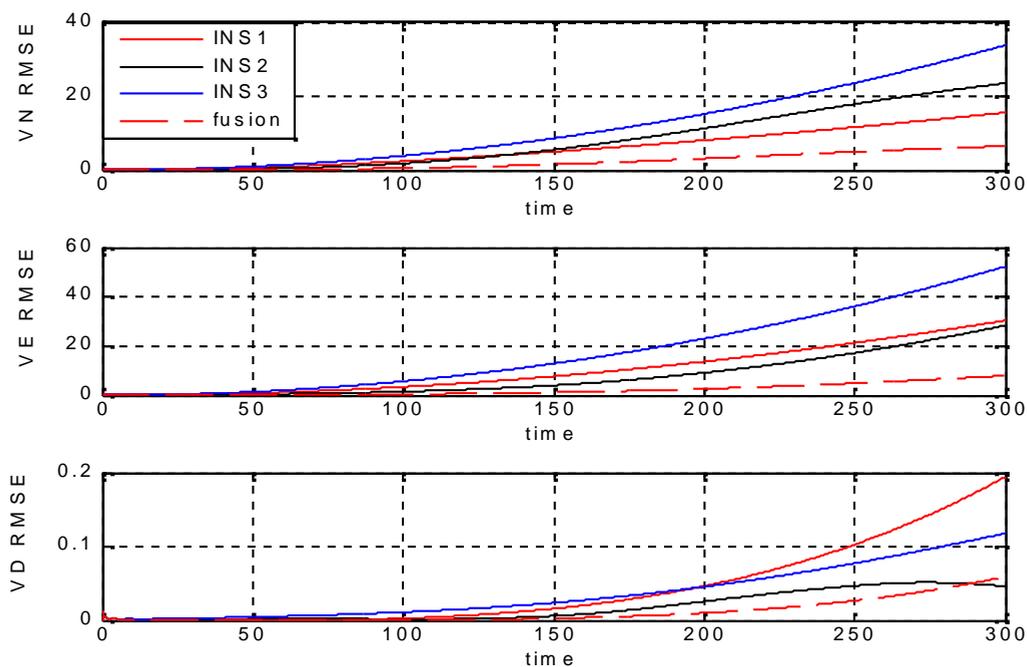


Figure (5) Velocity RMSE

c) The Attitude RMSE

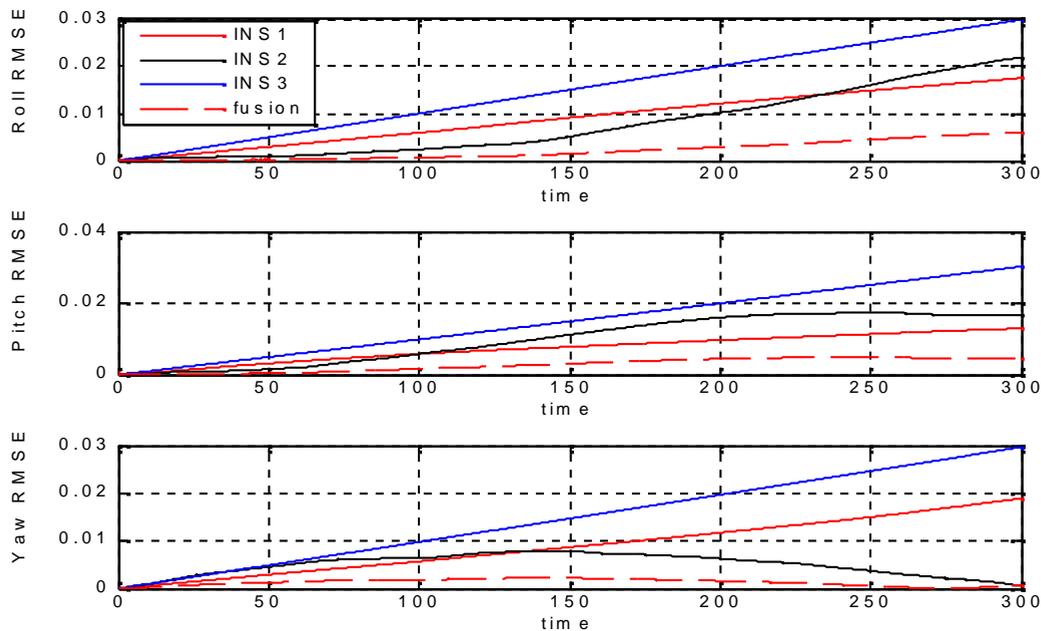


Figure (14) Attitude RMSE

9.1.3 Calculation of (P, V, A) RMSE using the INS/GPS Integration

a) Position RMSE Using Integration without Fusion and with Fusion

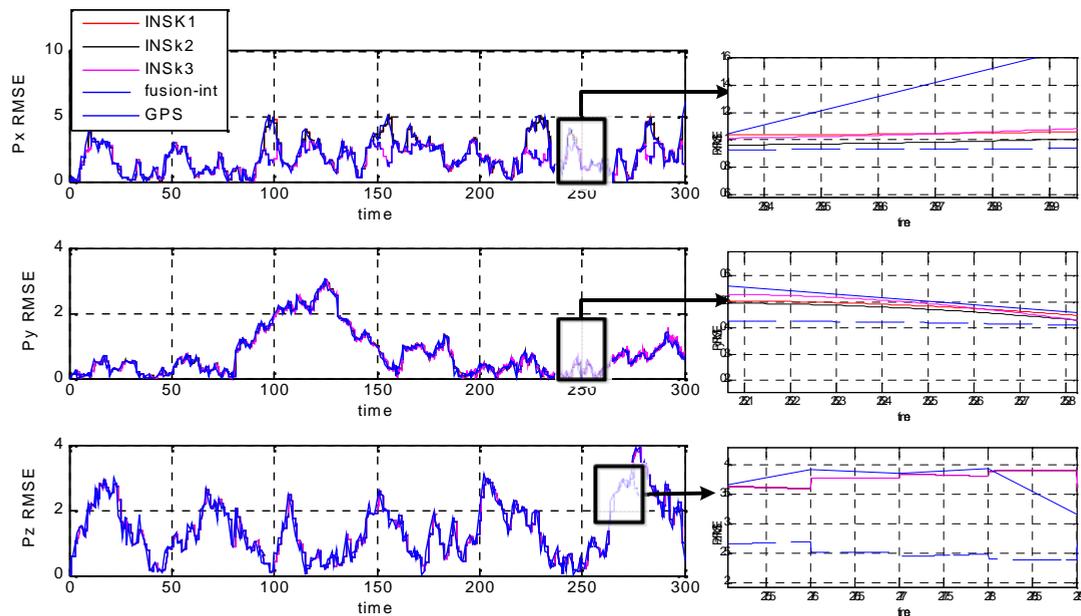


Figure (15) Position RMSE of using kalman filter and the GPS and the Fusion with kalman filter

b) Velocity RMSE Using Integration without Fusion and with Fusion

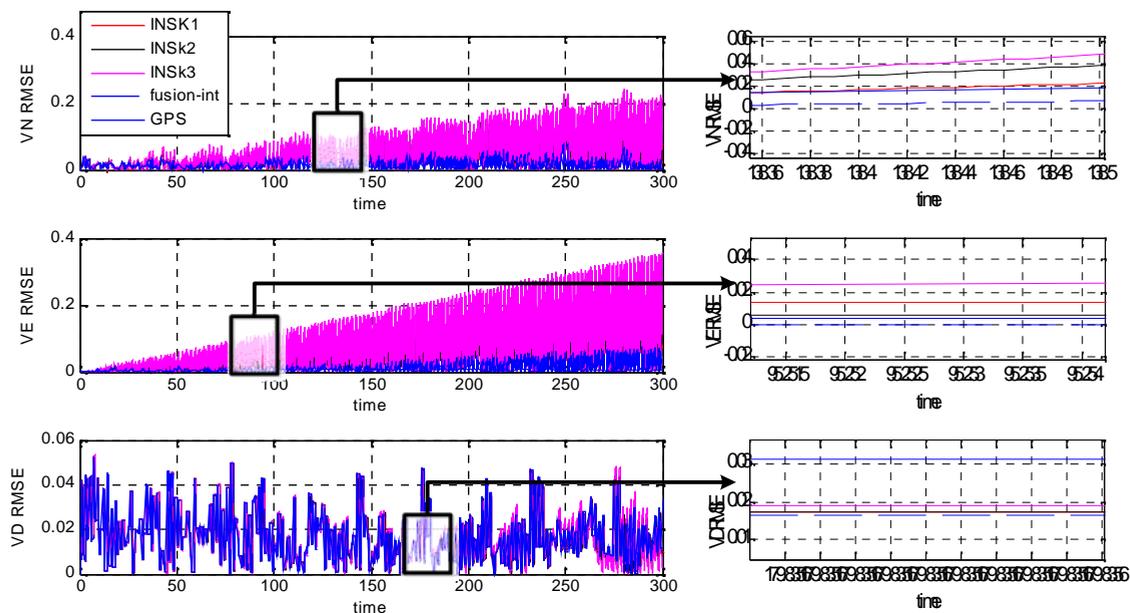


Figure (6) Velocity RMSE of using kalman filter and the GPS and the Fusion with kalman filter

c) Attitude RMSE Using Integration without Fusion and with Fusion

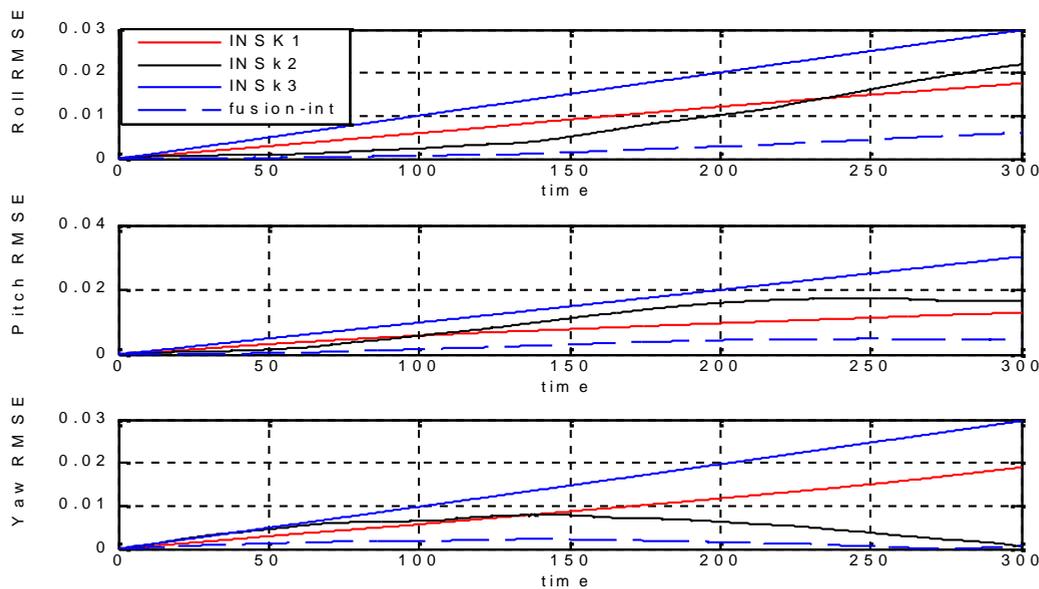


Figure (7) Attitude RMSE of using kalman filter and the GPS and the Fusion with kalman filter

10. Conclusion

From the results it is clear that the use of stand-alone INS system imply large error growth but when using the integration between the INS and the GPS the error has been decreased, but when the multi-sensor data fusion algorithm is used for fusing the three IMUs measurements, the INS system has been Improved and also the integration Results also improved by reducing the error of the navigation data (PVA), So from the Results using the multi-sensor data fusion algorithm is improving the navigation process by improving the IMUs measurements, even improve the integration between the INS and the GPS.

References

- [1] Marin E.Liggins, David L.Hall, James Llinas, *Handbook of Multisensor Data Fusion Theory and Practice*, Second ed.: CRC Press, 2009.
- [2] Jitendra R. Raol, *Multi-Sensor Data Fusion with MATLAB*. Boca Raton, London, New York: CRC Press, 2010.
- [3] ADRIAN HILIUTA, RENÉ JR. LANDRY, FRANCOIS GAGNON,, "Fuzzy Corrections in a GPS/INS Hybrid Navigation System," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, vol. 40, 2004.
- [4] Unmanned Dynamics. *AeroSim aeronautical simulation blockset Version 1.2*. Available: <http://www.u-dynamics.com>.
- [5] Julie M.A. Perreault, "Augmented Kalman Filter/Artificial Intelligence for Inertial Sensors/GPS Data Fusion," Royal Military College of Canada, 2008.
- [6] Jasbir Singh Vig, "Improved Navigation of a Vehicle Using INS/GPS with Line Of Sight Measurements" master Msc. Thesis, The State University of New York at Buffalo, 2005.
- [7] Ahmed Azouz, Ahmed Hamad, and Ahmed El-Said, "GPS/INS INTEGRATION," Msc., Aircraft Ground Electronic Support, Military Technical College, Cairo, 2006.
- [8] Huamin Jia, "Data Fusion Methodologies for Multisensor Aircraft Navigation Systems," PhD, College of Aeronautics, Cranfield, Cranfield, 2004.

- [9] J.C Bezdek , J.C Dunn, "Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions," *IEEE Transactions on Computers*, pp. 835-838, 1975.
- [10] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function*. New York: Plenum Press, 1981.
- [11] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis.*: Wiley ,Chichester, 1999.
- [12] Indrajit Saha, Anirban Mukhopadhyay, "Improved Crisp and Fuzzy Clustering Techniques," *IAENG international journal of computer science*, november 2008.
- [13] J.C Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*: Plenum Press, 1981.
- [14] J.C Bezdek, R. Ehrlich , W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, pp. 191-198, 1984.
- [15] J.C Bezdek, R. Hathaway, M. Sabin ,W. Tucker, "Convergence theory for fuzzy c-means: counterexamples and repairs," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 17, pp. 873-877, 1987.
- [16] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, 2nd ed. NewYork: John Wiley, 2001.
- [17] R. Duda, P. Hart, D. Stork, *Pattern Classification*. New York: Wiley, 2001.
- [18] C. Bishop, *Neural Networks for Pattern Recognition*. NewYork: Oxford University Press, 1995.
- [19] A. Vemuri, M. Polycarpou, "'On the use of on-line approximators for sensor fault diagnosis'," in *American Control Conference*, Philadelphia, Pennsylvania, USA, 1998, pp. 2857-2861.
- [20] Mohammad Abdel Kareem Jaradat , Reza Langaria,, "A hybrid intelligent system for fault detection and sensor fusion," *ELSEVIER*, pp. 415-422, 9/ 2009.
- [21] Eun-Hwan Shin, "Accuracy Improvement of Low Cost INS/GPS for Land Applications," Msc., Calgary, Alberta, 2001.
- [22] Mohinder S. Grewal, Lawrence R. Weill, Angus P. Anderows,, *Global Positioning Systems,Inertial Navigation,and Integration*, second ed.: WILEY, 2007.