Fuzzy state noise-driven Kalman filter for sensor fusion

S Chauhan[1], **C Patil**[2], **M Sinha**[2*], and **A Halder**[2]
[1]Department of Electronic and Electrical Engineering, IIT Kharagpur, Kharagpur, West Bengal, India
[2]Department of Aerospace Engineering, IIT Kharagpur, Kharagpur, West Bengal, India

The manuscript was received on 19 February 2009 and was accepted after revision for publication on 21 July 2009.

DOI: 10.1243/09544100JAERO536

Abstract: This article proposes a fuzzy state noise-driven Kalman filter for sensor fusion to estimate the instantaneous position and attitude of an unmanned air vehicle for navigation purpose. The formulation of the state noise covariance matrix has been carried out using the fuzzy regression method applied to the state residuals. This algorithm has been embedded in the real-time hardware and tested for performance on ground and not in real flight. A comparative study between the proposed and conventional algorithm illustrates its efficacy.

Keywords: sensor fusion, Kalman filter, unmanned air vehicles, hardware-in-loop-simulation

1 INTRODUCTION

This is a fact that while solving an engineering problem, hardware implementation can throw light on areas where theory and simulation lack [**1**]. Many engineering problems involve constraints such as synchronization, computational complexity, size, weight, power, cost, saturation, and so on, which are either knowingly neglected in theoretical description for the sake of mathematical simplicity or are too resource-demanding to run a meaningful simulation. In addition, to assess the credibility of the theory itself, hardware experiments also provide an avenue to testify the assumptions that lay the foundation of the theory.

At the heart of sensor fusion lies Kalman filter. A number of implementations to estimate the states of a system, either in simulation or on hardware, have been reported in various literatures. A gimballed inertial navigation system (INS) with nine state estimation using Kalman filter has been reported in reference [**2**]. Large number of states estimation using Kalman filter has been described in references [**3**] to [**6**]. An implementation for an air-data-based dead reckoning system for the unmanned air vehicle (UAV) Nishant has been described in reference [**7**]. Large-scale state estimation using Kalman filter has been described in references [**8**] to [**12**]. In reference [**13**], a fuzzy-based sensor fusion technique has been reported, while in reference [**14**], INS and global positioning system (GPS) integration using Kalman filter have been worked out. Unscented Kalman filter and its variants have been worked out in references [**15**] to [**18**]. The past research in aircraft sensor fusion algorithm has primarily strived for better accuracy, either by incorporating sophisticated hardware to improve arithmetic precision, or by implementing a filtering scheme with higher algorithmic complexity. Neither of these approaches led to a cost-effective solution from system integration point of view.

The main objective of this article is to overcome this shortcoming by designing a novel sensor fusion algorithm taking sensor uncertainty and ambiguity into account in a way that is realizable in low-cost hardware platform, in real time, for a UAV. A state noise-driven extended Kalman filter (EKF) has been implemented for the real-time state estimation of a UAV using sensor fusion. The state estimates obtained from the embedded sensor fusion algorithm proposed in this article can also be used to construct artificial sensor feedbacks for pan-tilt stabilization of the camera and sonar altimeter platform affixed to the UAV. The aim of the work reported here is to design and implement a sensor fusion algorithm on a hardware platform, in real time, for a UAV. However, the current work shows the performance of the algorithm implemented on hardware and tested on ground, which is a precursor to the final flight testing.

*Corresponding author: Department of Aerospace Engineering, IIT Kharagpur, Kharagpur, West Bengal 721302, India.
email: ms.aissq@gmail.com

2 ATTITUDE MEASUREMENT

The complete implementation can be bifurcated into three steps as listed below.

1. Sensor fusion for inertial measurement unit (IMU) and GPS data computes the states of the UAV such as position (x, y, z) and attitude (ϕ, θ, ψ).
2. Elimination of divergence in the filter.
3. Implementation on hardware that involves various sensors interfaced with the on-board microcontroller of the UAV to acquire flight data.

The data coming from these sensors are, in general, noise corrupted. Hence, it is imperative to have an algorithm that can yield realistic, unique, and optimal values of the desired quantities describing the state of the system by fusing the data coming out from various sensors while meeting the time constraint. This article proposes such an algorithm and demonstrates the efficacy of it through real-time hardware-in-loop-simulation (HILS). Generally, attitude is estimated from directly measurable variables. Using carrier-phase GPS signal from three GPS antenna, with a known geometry, attitude estimation can be obtained [**19**, **20**]. After the resolution of phase ambiguity, phase differences between the antennas can be worked out and attitude can be estimated. However, the estimation accuracy increases with the increase in the baseline length between the antennas, which poses constraint for the small UAVs. Gebre-Egziabher *et al.* [**21**] proposed an ultra short baseline solution (∼36 cm baseline), but even this is too large and heavy for small UAVs [**22**]. Another promising method is to use vector measurements of the magnetic and gravitational fields and then solve a set of non-linear equations using optimization methods to come up with an attitude measurement [**23**, **24**]. Akella *et al.* [**25**] formulated a feedback law that directly regulates the attitude with only gyro and inclinometer measurements. The attitude estimation follows the following steps:

(a) obtain three consecutive GPS position measurements;
(b) difference the three GPS measurements to obtain two velocity measurements;
(c) average the two velocity measurements to give average velocity over 2 s;
(d) calculate the track angle ψ from velocity as $\psi = \tan^{-1}(\dot{y}/\dot{x})$;
(e) difference the GPS calculated velocities to obtain a GPS acceleration measurement a_{GPS};
(f) average the accelerometers over the same 2 s as the GPS velocity is calculated to obtain a;
(g) calculate θ and ϕ using the accelerometers and the GPS acceleration rotated by ψ from corresponding equations [**26**].

3 SENSOR FUSION USING EXTENDED KALMAN FILTER

This section gives a summary of attitude filtering and position filtering, respectively, using EKF. Estimation using EKF involves time updates and measurement updates as described below. The notation of the symbols used is standard and can be referred to reference [**3**].

3.1 Attitude filtering using EKF

Initialize with

$$\hat{\vec{\boldsymbol{x}}}_0 = \mathrm{E}[\vec{\boldsymbol{x}}_0] \tag{1}$$

$$P_0 = \mathrm{E}[(\vec{\boldsymbol{x}}_0 - \hat{\vec{\boldsymbol{x}}}_0)(\vec{\boldsymbol{x}}_0 - \hat{\vec{\boldsymbol{x}}}_0)^{\mathrm{T}} \tag{2}$$

Time update

$$\hat{\vec{\boldsymbol{q}}}_{k+1} = \hat{\vec{\boldsymbol{q}}}_k + T_{\text{sampling}}\boldsymbol{\Omega}(\vec{\boldsymbol{\omega}}_k - \hat{\vec{\boldsymbol{b}}}_k)\hat{\vec{\boldsymbol{q}}}_k \tag{3}$$

$$\hat{\vec{\boldsymbol{q}}}_{k+1} = \frac{\hat{\vec{\boldsymbol{q}}}_{k+1}}{\|\hat{\vec{\boldsymbol{q}}}_{k+1}\|} \tag{4}$$

$$\hat{\vec{\boldsymbol{b}}}_{k+1} = \hat{\vec{\boldsymbol{b}}}_k \tag{5}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k + T_{\text{sampling}}(\mathbf{A}_k\mathbf{P}_k + \mathbf{P}_k\mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k) \tag{6}$$

Measurement update

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{C}_k^{\mathrm{T}}(\mathbf{C}_k\mathbf{P}_k\mathbf{C}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1} \tag{7}$$

$$\hat{\vec{\boldsymbol{x}}}_{k+1} = \hat{\vec{\boldsymbol{x}}}_k + \mathbf{K}_k[z_k - \mathrm{Euler}(\hat{\vec{\boldsymbol{q}}}_k)] \tag{8}$$

$$\hat{\vec{\boldsymbol{q}}}_{k+1} = \frac{\hat{\vec{\boldsymbol{q}}}_{k+1}}{\|\hat{\vec{\boldsymbol{q}}}_{k+1}\|} \tag{9}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_k \tag{10}$$

3.2 Position filtering using EKF

State

$$\vec{\boldsymbol{x}} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}} \tag{11}$$

Time update

$$\hat{\vec{\boldsymbol{x}}}_{k+1} = \hat{\vec{\boldsymbol{x}}}_k + T_{\text{sampling}}\left(\mathrm{DCM}^{\mathrm{T}}\begin{bmatrix} V_{\mathrm{P}} \\ 0 \\ 0 \end{bmatrix}\right) \tag{12}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \mathbf{T}_{\text{sampling}}(\mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^{\mathrm{T}}) \tag{13}$$

Measurement update

$$\mathbf{K}_k = \mathbf{P}_k(\mathbf{P}_k + \mathbf{R}_k)^{-1} \tag{14}$$

$$\hat{\vec{\boldsymbol{x}}}_{k+1} = \hat{\vec{\boldsymbol{x}}}_k + \mathbf{K}_k(\vec{\boldsymbol{z}}_k - \hat{\vec{\boldsymbol{x}}}_k) \tag{15}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_k)\mathbf{P}_k \tag{16}$$

Here, $\mathbf{G}_k$ is the partial derivative of the dynamics with respect to each of the inputs. Formulation of $\mathbf{G}_k$ can be found in reference [**26**]. Ω is written as

$$\Omega(\vec{\omega}) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{17}$$

4 FUZZY STATE NOISE-DRIVEN FILTER

At times the EKF may diverge because of system modelling, finite arithmetic, and round off errors. Various divergence control strategies can be found in reference [**27**] along with the outline of the algorithm. The objective of this algorithm is to guarantee the positive definiteness of the state covariance matrix (**P**) by suitably formulating the process noise covariance matrix (**Q**).

4.1 Formulation of process noise covariance matrix

To facilitate computation, it is a common practice to diagonalize **Q**, which has been proposed to be formulated as

$$Q_{ij} = |\Delta\vec{\boldsymbol{x}}\Delta\dot{\vec{\boldsymbol{x}}}|\Delta t\delta_{ij} \tag{18}$$

In order to calculate the derivative of the state residuals, the following method has been proposed by the authors.

4.2 Proposed filter algorithm

A linear fuzzy regression model with suitable size of the moving window was adopted to plot the state residue curve. On a sample dataset, this provided better result than the least square plot with a moving window [**27**]. Hence, the linear fuzzy regression with a moving window was applied for the case of non-fuzzy data. In this algorithm, the state residual vector is formulated using linear fuzzy regression as

$$\Delta\widehat{\boldsymbol{x}} = f(t, \widehat{\mathbf{A}}) = \widehat{\mathbf{A}}_0 + \widehat{\mathbf{A}}_1 t_1 + \widehat{\mathbf{A}}_2 t_2 + \cdots + \widehat{\mathbf{A}}_n t_n \tag{19}$$

where $\widehat{\mathbf{A}}_i$ is the ith fuzzy coefficient, a fuzzy number (contrary to conventional crisp constant). Since each fuzzy number can be characterized by its membership function, each $\widehat{\mathbf{A}}_i$ was expressed as an isosceles triangular membership function such that c_i is the spread (half-width of the base) and p_i is the mid point of the base. Thus the aim of the fuzzy regression problem considered here is to determine a family of such symmetric triangles representing all the coefficients in the linear fuzzy regression formula. It can be shown that the corresponding symmetric triangular fuzzy membership function for the fuzzified state residual vector has mid-point and spread of $\sum_{i=1}^{n} p_i t_i$ and $\sum_{i=1}^{n} c_i|t_i|$, respectively [**27**]. Now one needs to find the fuzzy coefficients such that the spread of the fuzzy output is minimized. Tanaka *et al.* [**28**] formulated the objective function for this optimization problem as

$$O_{\mathrm{f}} = \min\left\{ mc_0 + \sum_{j=1}^{m}\sum_{i=1}^{n} c_i t_{ij} \right\} \tag{20}$$

where $t_{0j} = 1 \forall j = 1, 2, \ldots, m$. This optimization problem can be viewed as the minimization of total fuzziness (hence ambiguity) of the fuzzy linear model. This objective function O_{f} needs to be minimized subjected to two inequality constraints (derived by Tanaka *et al.* [**28**])

$$x_j \geqslant \sum_{i=0}^{n} p_i t_{ij} - (1-h)\sum_{i=0}^{n} c_i t_{ij} \tag{21}$$

$$x_j \leqslant \sum_{i=0}^{n} p_i t_{ij} + (1-h)\sum_{i=0}^{n} c_i t_{ij} \tag{22}$$

Equations (20) and (21) represent total $2m$ constraints. Now this becomes a linear programming problem, which must be solved to find the mid points and spreads of the fuzzy coefficients. The authors used the simplex method to solve this constrained optimization problem.

5 HARDWARE ARCHITECTURE

The hardware platform consists of two printed circuit boards (PCBs), namely sensor interface card and processor card arranged in a dual stack configuration connected by 88 pins (Fig. 1). The upper PCB is the sensor interface card which takes the sensor outputs and sends them to the lower PCB (i.e. the processor card). In addition to interfacing the sensors, the sensor interface card also synchronizes the bit rate of all sensors to the processing speed of the microcontroller (9600 bps).

The 32-bit microcontroller used in this custom-designed board is Motorola MC68332. Its features include a 32 bit CPU (CPU32), a system integration module, a time processor unit (TPU), a queued serial module, 2 kB static RAM with TPU emulation capability, a maximum system clock speed of 20.97 MHz, and a high density complementary metal-oxide semiconductor architecture to make low power consumption of the micro controller unit. Among various sensors interfaced with the processor card (via sensor interface card), IMU and GPS are of particular interest as the proposed algorithm fuses data coming from the two. The IMU consists of three gyros (ADXRS150) and three accelerometers (ADXL105). The gyros measure body angular velocities and the accelerometers measure

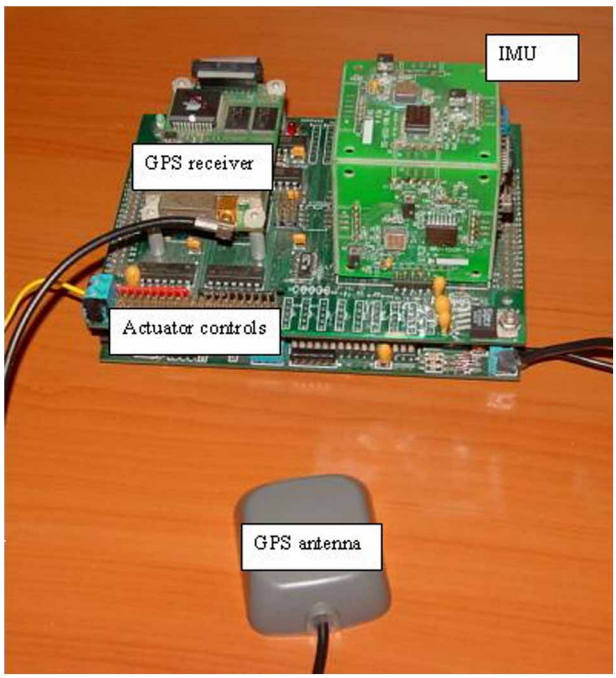

Fig. 1 Sensor interface card and processor card with sensors

translational accelerations. The GPS gives spatial position of the board.

6 HARDWARE IN LOOP SIMULATION

In actual hardware implementation, driving noise covariance matrix **Q** has been formulated with and without fuzzy state noise covariance (using random noise in the same range). The comparative results for real-time runs bring forth the superiority of the proposed algorithm.

6.1 Hardware specific consideration

Since measurement update occurs once in a second (GPS frequency is 1 Hz), in order to make a better estimate, the conventional – once time update, then only measurement update – philosophy was modified. In fact, time update (predictor step) is carried out for three time steps, and then one measurement update is done (i.e. at the fourth time step). Proper algorithm for matrix inversion and computation of trignometric functions were implemented to reduce the computational burden. The number of nested loops in the code was also minimized.

6.2 The effect of the magnitude of attitude on tracking performance

For a large set of pitch angles, tracking performance was compared and the performance was found to be satisfactory (<1 per cent error). In evaluating tracking performance, reference measurement of attitude was assumed to be the constructed measurement as explained in section 2. Figures 2 to 4 show the tracking performance for three different pitch angles, namely 4.4°, 17.4°, and 28.2°.

Therefore, it can be concluded that the magnitude of the angle does not have a significant effect as far

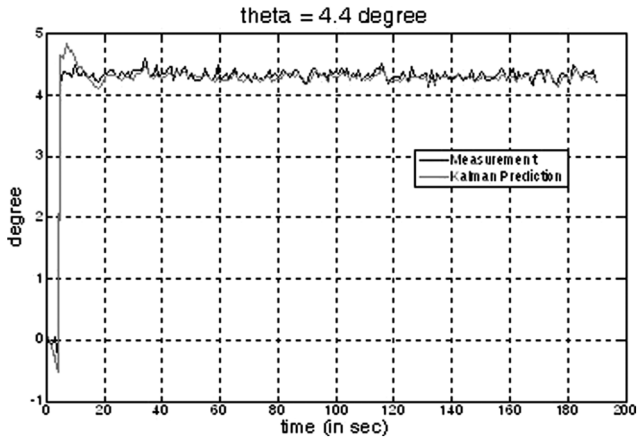

Fig. 2 Tracking performance for 4.4° pitch angle

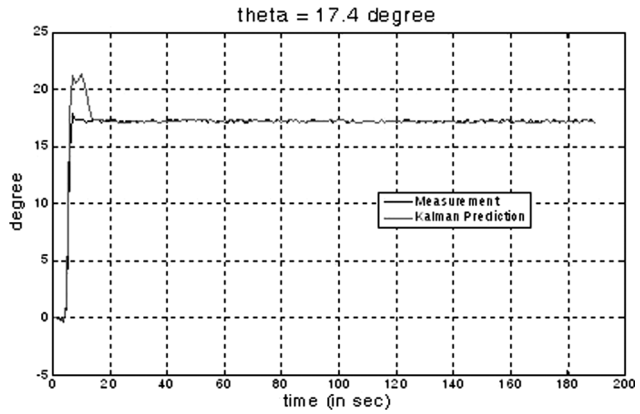

Fig. 3 Tracking performance for 17.4° pitch angle

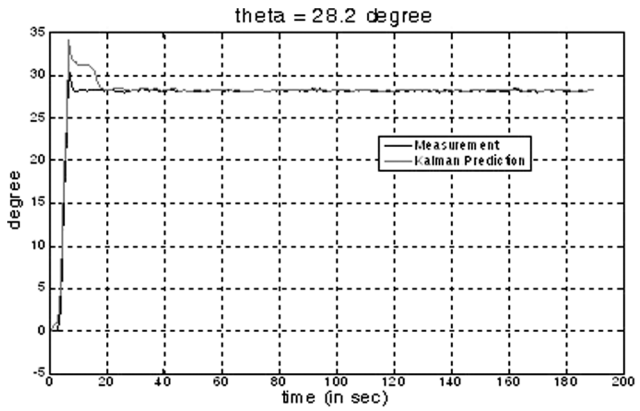

Fig. 4 Tracking performance for 28.2° pitch angle

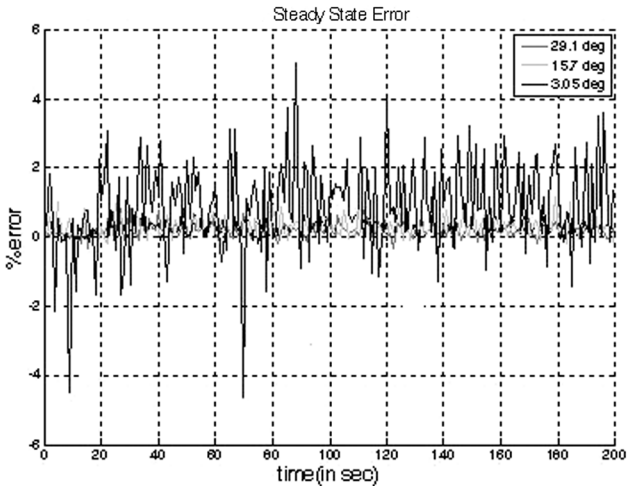

Fig. 5 Steady-state error comparisons for three different pitch angles

as tracking performance is concerned. However, when steady-state errors for three different pitch angles were plotted together, it turns out that smaller the magnitude of the angle, the larger will be the deviations from the steady-state value. Figure 5 illustrates this fact (three representative pitch angles were chosen to be 29.1°, 15.7°, and 3.05°).

6.3 The effect of the driving noise

The objective of this study was to investigate how the formulation of process noise covariance matrix (**Q**) affects the performance of the filter. Hence for three different pitch angles (14.1°, 35.1°, and 44.6°), the fuzzy state noise-driven EKF and conventional EKF code (with square of the random numbers as the entries of the diagonal **Q** matrix) were run simultaneously. The results are plotted in Figs 6 to 8, where the percentage

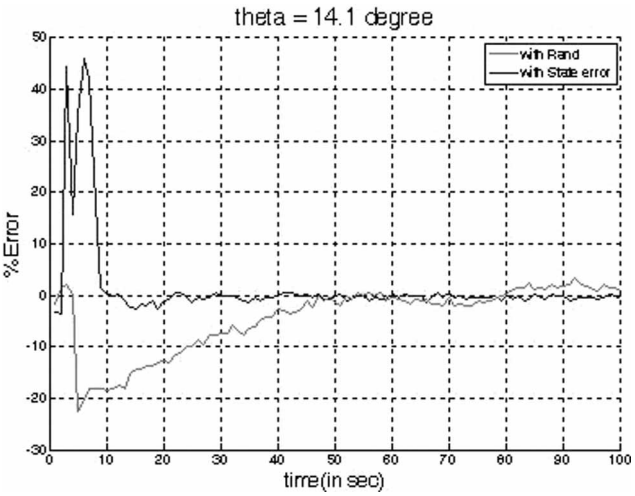

Fig. 6 Effect of driving noise for 14.1° pitch angle

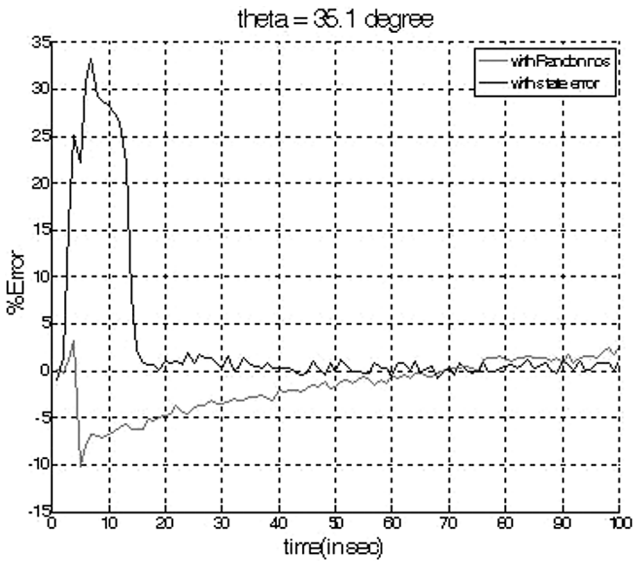

Fig. 7 Effect of driving noise for 35.1° pitch angle

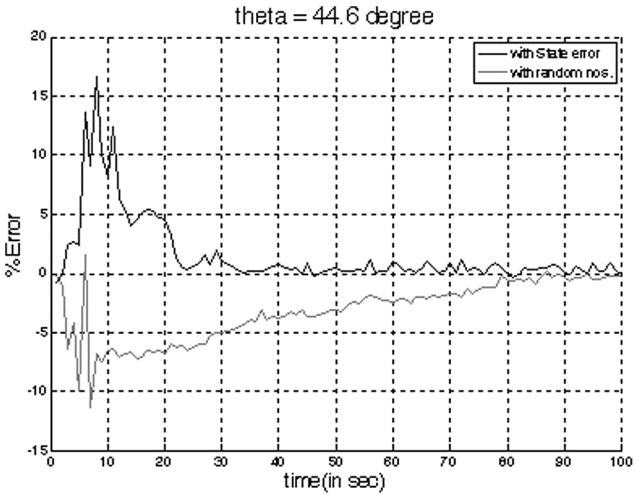

Fig. 8 Effect of driving noise for 44.6° pitch angle

estimation errors (with respect to the measurements) were plotted against time.

Some general observations can be made from Figs 6 to 8. It can be noted that the maximum overshoot in the transient state (corresponding to changing the angle) for state error formulation is much larger compared to that with random noise. However, the settling time is significantly larger for the case when random noise is used. Also, as it is evident from the last three plots, random noise formulation has a slight tendency of overshoot in steady state too (i.e. the steady-state error value for random noise is higher). It can be noted that, in all three plots above, the random noise curve slowly captures the state noise curve from bottom and has a tendency to cross it to a positive value.

To investigate it further, steady-state errors (after convergence) were magnified for both the

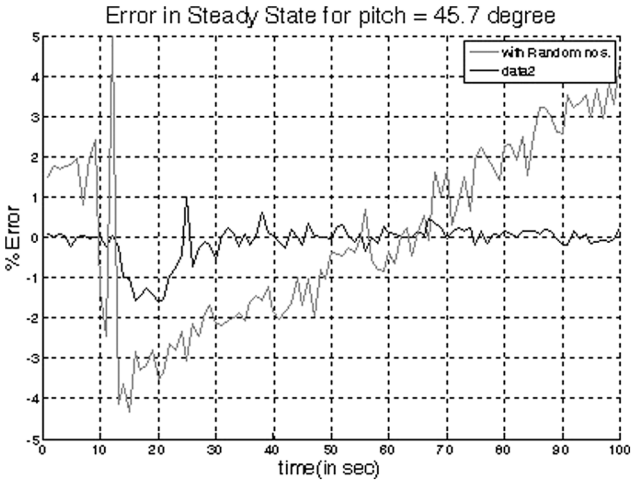

Fig. 9 Effect of driving noise after the filter has converged (magnified steady-state comparison)

formulations for the same magnitude of pitch angle (45.7°). This (Fig. 9) confirms the speculation made in the earlier paragraph. The steady-state oscillations are much larger in amplitude for random noise formulation. This probably sheds light on why fuzzy state noise is better than the random noise formulation.

7 CONCLUSIONS

Real-time hardware implementation of a fuzzy state noise-driven EKF for sensor fusion is presented in this article with detailed results of HILS. The results showed that the proposed algorithm yields faster convergence with better steady-state performance compared to conventional EKF.

ACKNOWLEDGEMENT

This work has been sponsored by ER&IPR, DRDO, India.

© Authors 2009

REFERENCES

1 **Bernstein, D. S.** Four and a half control experiments and what I learned from them: a personal journey. In Proceedings of the IEEE American Control Conference, Albuquerque, New Mexico, 1997, pp. 2718–2725.
2 **Schimdt, G. T.** Strapdown inertial systems – theory and applications. AGARD Lecture Series, 1978, No. 95.
3 **Grewal, M. S.** and **Andrews, A. P.** *Kalman filtering*: *theory and practice using MATLAB*, 2001 (John Wiley, New York).
4 **Grewal, M. S., Weill, L. R.,** and **Andrews, A. P.** *Global positioning systems, inertial navigation and integration*, 2001 (John Wiley, New York).
5 **Wolf, R., Eissfeller, B.,** and **Hein, G. W.** A Kalman filter for the integration of a low cost INS and an attitude GPS, Institute of Geodesy and Navigation, Munich, Germany. Available from http://www.ifen.com/content/publications/KIS1997_GPS_INS.pdf.
6 **Grejner-Brzezinska, D. A.** and **Wang, J.** Gravity modeling for high accuracy GPS/INS integration. *Navigation*, 1998, **45**(3), 209–220.
7 **Srikumar, P.** and **Deori, C. D.** Simulation of mission and navigation functions of Nishant. In Proceedings of the National Workshop on Aerospace Flight Simulation, VSSC, Trivandrum, India, 2000.
8 **Randle, S. J.** and **Horton, M. A.** Low cost navigation using micro-machined technology. In Proceedings of the IEEE Intelligent Transportation Systems Conference, Boston, Massachusetts, 9–12 November 1997.
9 **Brown, A.** and **Sullivan, D.** Precision kinematic alignment using a low-cost GPS/INS System. In Proceedings of the ION GPS 2002, Navsys Corporation, Oregon, 2002.
10 Navsys, the navigation systems innovators. Available from http://www.navsys.com.
11 **Brown, A.** Test results of a GPS/inertial navigation system using a low-cost MEMS IMU. In Proceedings of the 11th Annual Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg, Russia, May 2004.
12 **Moon, S. W., Kim, J. H., Hwang, D. H., Ra, S. W.,** and **Lee, S. J.** Implementation of loosely coupled GPS/INS integrated system. In Proceedings of the Fourth International Symposium on Satellite Navigation Technology and Applications, Brisbane, Queensland, Australia, 20–23 July 1999.
13 **Mayhew, D. M.** *Multi-sensor fusion for GPS navigation using Kalman filtering*. M Sc Thesis, Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, 1999.
14 **Moore, J. B.** and **Qi, H.** Direct Kalman filtering approach for GPS/INS integration. *IEEE Trans. Aerosp. Electron. Syst.*, 2002, **38**(2), 687–693.
15 **Julier, S.** and **Uhilmann, J.** A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ UK, 1996.
16 **Julier, S., Uhlmann, J.,** and **Durrant-Whyte, H. F.** A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control*, 2000, **45**, 477–481.
17 **Brunke, S.** *Nonlinear filtering and system identification algorithm for autonomous systems*. PhD Thesis, University of Washington, Seattle, Washington, 2001.
18 **Van der Merwe, R.** and **Wan, E.** The square root unscented Kalman filter for state and parameter estimation. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, Utah, 2001.
19 **Peng, H. M., Chiang, Y. T., Chang, F. R.,** and **Wang, L. S.** Maximum-likelihood-based filtering for attitude determination via GPS carrier phase. In Proceedings of the IEEE Position, Location, and Navigation Symposium, San Diego, California, March 2000, pp. 480–487.

20 **Chiang, Y. T., Wang, L. S., Chang, F. R.,** and **Peng, H. M.** Constrained filtering method for attitude determination using GPS and gyro. In Proceedings of the IEEE Conference on Radar, Sonar and Navigation, October 2002, vol. 149, pp. 285–264.

21 **Gebre-Egziabher, D., Hayward, R. C.,** and **Powell, J. D.** A low-cost GPS/inertial attitude heading reference system (AHRS) for general aviation applications. In Proceedings of the IEEE Position, Location, and Navigation Symposium, Palm Springs, California, April 1998, pp. 518–525.

22 **Ettinger, S. M., Nechyba, M. C., Ifju, P. G.,** and **Waszak, M.** Vision-guided flight stability and control for micro air vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System, 2002, vol. 3, pp. 2134–2140.

23 **Gebre-Egziabher, D., Elkaim, G. H., Powell, J. D.,** and **Parkinson, B. W.** A gyro free quaternion-based attitude determination system suitable for implementation using low cost sensors. In Proceedings of the IEEE Position, Location, and Navigation Symposium, San Diego, California, March 2000, pp. 185–192.

24 **Marins, J. L., Yun, X., Bachmann, E. R., McGhee, R. B.,** and **Zyda, M. J.** An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, October 2001, vol. 4, pp. 2003–2011.

25 **Akella, M. R., Halbert, J. T.,** and **Kotamraju, G. R.** Rigid body attitude control with inclinometer and low-cost gyro measurements. In Proceedings of the Elsevier Systems and Control Letters 49, Santa Barbara, California, Feburary 2003, pp. 151–159.

26 **Kingston, D. B.** *Implementation issues of real-time trajectory generation on small UAVs*. MS Thesis, Brigham Young University, April 2004.

27 **Garhwal, R., Halder, A.,** and **Sinha, M.** An adaptive fuzzy state noise driven extended Kalman filter for real time orbit determination. In Proceedings of the 58th International Astronautical Congress, Hyderabad, India, September 2007.

28 **Tanaka, H., Uejima, S.,** and **Asai, K.** Linear regression analysis with fuzzy model. *IEEE Trans. Syst. Man Cybern.*, 1982, **12**, 903–907.

APPENDIX

Notation

$\mathbf{A}_k$	state transition matrix at time t_k
$\vec{\boldsymbol{b}}_k$	bias vector at time t_k
$\mathbf{C}_k$	observation matrix at time t_k
DCM	direction cosine matrix
$E[\cdot]$	expectation operator
$\mathbf{G}_k$	state noise gain matrix at time t_k
$\mathbf{I}$	identity matrix
$\mathbf{K}_k$	Kalman gain matrix at time t_k
$\vec{\mathbf{P}}_k$	state covariance matrix at time t_k
$\vec{\boldsymbol{q}}_k$	attitude quaternion at time t_k
$\mathbf{Q}_k$	state noise covariance matrix at time t_k
$\mathbf{R}_k$	observation noise covariance matrix at time t_k
T_{sampling}	sampling time
V_{p}	velocity
x, y, z	position coordinates
$\dot{x}, \dot{y}, \dot{z}$	rate of change of position coordinates
$\vec{\boldsymbol{x}}_0, \vec{\boldsymbol{x}}$	state vector at time t_0 and t, respectively
δ_{ij}	Kronecker delta function
$\Delta\vec{\boldsymbol{x}}$	state residual vector
$\Delta\dot{\vec{\boldsymbol{x}}}$	derivative of the state residual vector
ϕ, θ, ψ	bank, elevation, and azimuth angles, respectively
$\vec{\boldsymbol{\omega}}$	angular rate vector in body coordinates