

Neural network aided adaptive Kalman filter for GPS/INS navigation system design

Dah-Jing Jwo¹, Jyh-Jeng Chen²

¹Department of Communications and Guidance Engineering, National Taiwan Ocean University

2 Pei-Ning Rd., Keelung 20224, Taiwan

Fax: +886-224633492; E-mail: djjwo@mail.ntou.edu.tw

²Quanta computer Inc.

Abstract

A mechanism called PSO-RBFN, which is composed of radial basis function (RBF) network and particle swarm optimization (PSO), for predicting the errors and to filtering the high frequency noise is proposed. As a model nonlinearity identification mechanism, the PSO-RBFN will implement the on-line identification of nonlinear dynamics errors such that the modeling error can be compensated. The PSO-RBFN will be applied to the loosely-coupled Global Positioning System (GPS)/inertial navigation systems (INS) navigation filter design and has demonstrated substantial performance improvement in comparison with the standard Kalman filtering method.

Keywords: Global Positioning System (GPS), inertial navigation system (INS), radial basis function (RBF), neural network, particle swarm optimization (PSO)

1. Introduction

GPS/INS integration is typically carried out through Kalman filter (KF) (Brown and Hwang, 1997, Gelb, 1974). If the Kalman filter is provided with information that the process behaves a certain way, whereas, in fact, it behaves a different way, the filter will continually intend to fit an incorrect process signal. When the measurement situation does not provide sufficient information to estimate all the state variables of the system, in other words, the estimation error covariance matrix becomes unrealistically small and the filter disregards the measurement. In various circumstances where there are uncertainties in the system model and noise description, and the assumptions on the statistics of disturbances are violated since in a number of practical situations, the availability of a precisely known model is unrealistic due to the fact that in the modelling step, some phenomena are disregarded and a way to take them into account is to consider a nominal model affected by uncertainty. To overcome the deficiency of Kalma filter, the so-called adaptive Kalman filter (AKF) has been proposed. Many efforts have been made to

improve the estimation of the covariance matrices (Mehra, 1970, 1972) based on the innovation-based estimation (IAE) approach.

In actual navigation filter designs, there exist the model uncertainties which cannot be expressed by the linear state-space model. The linear model includes modeling errors since the actual vehicle motions are non-linear process. The system model, system initial conditions, and noise characteristics have to be specified *a priori*. It is very often the case that little *a priori* knowledge is available concerning the maneuver. The implementation of IAE based AKF to navigation designs has been widely explored (Hide *et al*, 2003, Mohamed and Schwarz 1999). The application of artificial intelligence to adaptive Kalman filter has been explored. A relatively large amount of research has been essentially based on the use of fuzzy logic.

A new approach is proposed for improving GPS/INS navigation system designs. The method makes use of the radial basis function network (RBFN) (Haykin, 1994) and the particle swarm optimization (PSO) techniques (Kennedy and Eberhart, 1995, Eberhart and Shi, 1998), resulting in an aiding mechanism called PSO-RBFN, which is employed into the navigation systems for real-time identification of noise covariance matrices to prevent divergence of the Kalman filter. The PSO is employed to obtain suitable RBFN parameters for filtering out the high frequency noise; RBFN is employed to filter out the high frequency noise for estimating the noise covariance matrices of the process noise and measurement noise for the navigation Kalman filter.

2. Inertial error modeling in state space

To avoid the complex coupling as in the 'perturbation error' equations, the so-called 'psi-angle' equations (Kong, Nebot and Durrant-Whyte, 1999, GPSof LLC, 2005) approach is taken to model the inertial errors. In the approach, the nomenclature used is summarized for convenience.

Body frame (b-frame): frame fixed to the vehicle.

Computer frame (c-frame): local level frame at the computed position

Platform frame (p-frame): frame which the transformed accelerations and angular rates from the

accelerometers and gyros are resolved.

Earth frame (e-frame): located at the earth center.

True frame (n-frame): true local level frame at the true position.

psi-angle (ψ): the angle between c-frame and p-frame.

phi-angle (Φ): the angle between n-frame and p-frame.

theta-angle (θ): the angle between n-frame and c-frame.

C_m^o : DCM from m-frame to o-frame, where DCM is the direction cosine matrix.

ω_{kl}^j : angular rate between k-frame and l-frame resolved in j-frame.

The psi-angle position, velocity and attitude errors are given as follows:

Position Error:

$$\delta\dot{\mathbf{R}} = -\omega_{en}^n \times \delta\mathbf{R}^n + \delta\mathbf{V}^c \quad (1)$$

Velocity Error:

$$\delta\dot{\mathbf{V}} = C_b^p \delta\mathbf{X}^b + \begin{bmatrix} \frac{-g}{r_e} & 0 & 0 \\ 0 & \frac{-g}{r_e} & 0 \\ 0 & 0 & \frac{2g}{r_e+h} \end{bmatrix} \delta\mathbf{R} - \Psi \times (\omega_{ec}^c + 2\omega_{ie}^c) \times \delta\mathbf{V}^c \quad (2)$$

Psi-angle:

$$\dot{\Psi} = -(\omega_{ec}^c + \omega_{ie}^c) \times \Psi + C_b^p \delta\omega_{ib}^b \quad (3)$$

where $\delta\omega_{ib}^b$ refers to the gyro errors in the body-frame expressed in continuous-time.

The dynamic process model in state space form is

$$\begin{bmatrix} \delta\dot{\mathbf{R}} \\ \delta\dot{\mathbf{V}} \\ \dot{\Psi} \\ \dot{\mathbf{e}}_a \\ \dot{\mathbf{e}}_g \\ \dot{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & \mathbf{F}_{23} & \mathbf{C}_b^p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{33} & \mathbf{0} & \mathbf{C}_b^p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{1}{\tau_g} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{1}{\tau_g} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\mathbf{R} \\ \delta\mathbf{V} \\ \Psi \\ \mathbf{e}_a \\ \mathbf{e}_g \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{u}_a \\ \mathbf{u}_g \\ \mathbf{u}_b \end{bmatrix} \quad (4)$$

In Equation (4), the ' $\mathbf{0}$'s are 3×3 matrices of zeros, and the ' \mathbf{C} 's stand for the direction cosine matrices.

$$\text{and } \mathbf{F}_{11} = [-\omega_{en}^n \times]; \quad \mathbf{F}_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$\mathbf{F}_{21} = \begin{bmatrix} \frac{-g}{r_e} & 0 & 0 \\ 0 & \frac{-g}{r_e} & 0 \\ 0 & 0 & \frac{2g}{r_e+h} \end{bmatrix}; \quad \mathbf{F}_{22} = [-(\omega_{en}^n + 2\omega_{ie}^n) \times];$$

$$\mathbf{F}_{23} = [\mathbf{f}^b \times]; \quad \mathbf{F}_{33} = [-(\omega_{en}^n + \omega_{ie}^n) \times]$$

where ' \mathbf{f} ' is the specific force vector; ' g ' is gravity; r_e is the radius of the earth; h is the vehicle height above the reference ellipsoid; ω_{ec}^n refers to transport-rate; ω_{ie}^n refers to earth-rate vector. The

term $[\mathbf{A} \times]$ refers to the skew-symmetric matrix form of the vector \mathbf{A} .

The error model for INS is augmented by some sensor error states such as accelerometer biases and gyroscope drifts. Actually, there are several random errors associated with each inertial sensor. Noise contributions in typical optical gyroscope systems include white noise, correlated random noise, bias instability and angle random walk. A state vector with 18 states is employed: nine so-called inertial error states (position, velocity and psi-angle), three accelerometer bias states, three gyro bias states, and three GPS estimated position bias states. The position and velocity errors are expressed in the local-level frame ENU (East-North-Up) frame.

3. Particle Swarm Optimization and Radial Basis Function (RBF) network

3.1 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic searching technique developed by Kennedy and Eberhart (1995). It is a relatively recent heuristic search method whose mechanics are inspired by swarming or collaborative behavior of biological populations. Among various evolutionary optimizer techniques, Genetic Algorithms (GA) and PSO have attracted considerable attention. The PSO is a robust stochastic evolutionary computation technique based on the movement and intelligence of swarms looking for the most fertile feeding location. Unlike the drawback of expensive computational cost of GA, PSO has better convergence speed.

A *swarm* consists of a set of particles moving around the search space, each representing a potential solution (fitness). Each particle has a position vector (x_i), a velocity vector (v_i), the position at which the best fitness ($Pbest_i$) encountered by the particle, and the index of the best particle ($Gbest$) in the swarm. The position of each particle is updated every generation. This is done by adding the velocity to the position vector.

$$v_i = v_i + C_1 \times rand() \times (Pbest_i - x_i) + C_2 \times rand() \times (Gbest - x_i) \quad (5)$$

The positions are based on their movement over a discrete time interval (Δt) as follows, with Δt usually set to 1.

$$x_i = x_i + v_i \cdot \Delta t \quad (6)$$

The parameters C_1 and C_2 are set to positive constant values, which are normally taken as 2 whereas $rand()$ represent uniformly distributed random values, uniformly distributed in $[0, 1]$ and w is called as inertia weight, the inertia weight is employed to control the impact of the previous history of velocities on the current one. Fig. 1 shows the flowchart for the PSO algorithm.

3.2 The Radial Basis Function network

A radial basis function network (RBFN) shuns the biological paradigm in favor of a topology which is

simpler and more amenable to analysis and training. In an RBFN, only a single layer of nodes with radially symmetric basis activation functions is needed to achieve a smooth approximation to an arbitrary real nonlinear function. Fig. 2 shows a schematic of a typical RBF network. The overall response of the RBF network $F(\mathbf{x})$ of Fig. 2 using Gaussian function $\varphi_j(\mathbf{x})$ can be formulated as

$$F(\mathbf{x}) = \sum_{j=1}^J w_j \varphi_j(\mathbf{x}) + b = \sum_{j=0}^J w_j \varphi_j(\mathbf{x}) \quad (7)$$

and

$$\varphi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - m_j\|^2}{2\sigma_j^2}\right) \quad (8)$$

where \mathbf{x} is the input pattern; m_j represents the center of the RBFN, which have the same dimensionality as the input vector; $\|\bullet\|$ denotes the Euclidean norm; $\varphi_j(\cdot)$ is the Gaussian (basis) function of the j -th neuron; σ is a positive constant representing the width of the RBF; w_j is the weight of the RBFN; and b is the bias; p is the number of inputs; J is the number of neurons in the hidden layer (also the number of centers for the RBF).

$$\begin{bmatrix} 1 & \varphi_1(\mathbf{x}_1) & \cdots & \varphi_J(\mathbf{x}_1) \\ 1 & \varphi_1(\mathbf{x}_2) & \cdots & \varphi_J(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \varphi_1(\mathbf{x}_N) & \cdots & \varphi_J(\mathbf{x}_N) \end{bmatrix}_{N \times (J+1)} \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_J \end{bmatrix}_{(J+1) \times 1} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}_{N \times 1} \quad (9a)$$

where N is the number of training pattern. The above equation can be written in the simplified form

$$\mathbf{W}\Phi = \mathbf{d} \quad (9b)$$

The weigh vector \mathbf{W} is calculated by the pseudoinverse matrix method

$$\mathbf{W} = \Phi^+ \mathbf{d} \quad (10)$$

where $\Phi^+ = (\Phi^T \Phi)^{-1} \Phi^T$ represents the pseudo-inverse of Φ . For further discussion on the topics of RBFN, the readers are referred to, e.g., Haykin (1999).

4. The proposed PSO-RBFN design

It is usually difficult to set a certain stochastic model for each inertial sensor that works efficiently at all environments and reflects the long-term behavior of sensor errors. The difficulty of modeling the errors of INS raised the need for a model-less GPS/INS integration technique. The fact that KF highly depends on a predefined dynamics model forms a major drawback. If the estimated internal model does not reflect the real model, the KF estimates may not be reliable and divergence problem might occur. Many efforts have been made to improve the estimation of the covariance matrices.

In this work, the PSO is employed to obtain suitable RBF parameters. RBF is employed to filter out the high frequency noise for deriving the noise covariance matrices of the process noise and measurement noise in the navigation KF. Fig. 3 shows the PSO-RBFN functional block diagram.

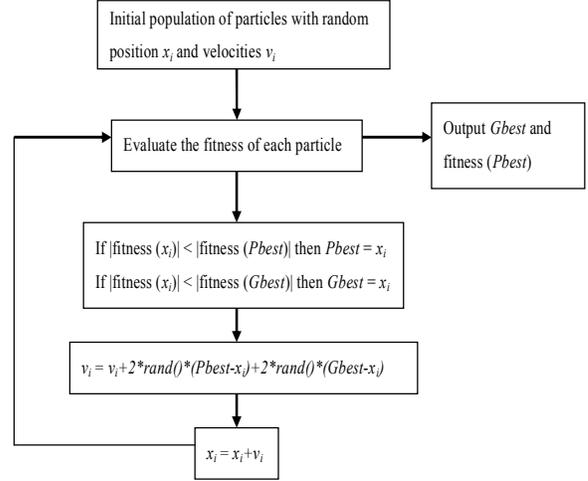


Fig. 1. Flowchart for the PSO algorithm

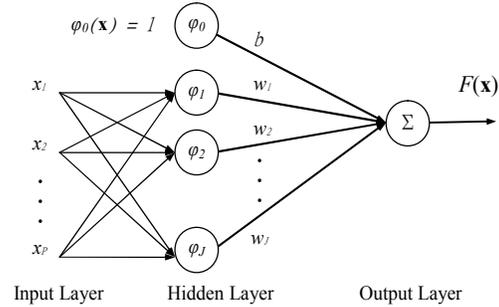


Fig. 2. Architecture of an RBF network

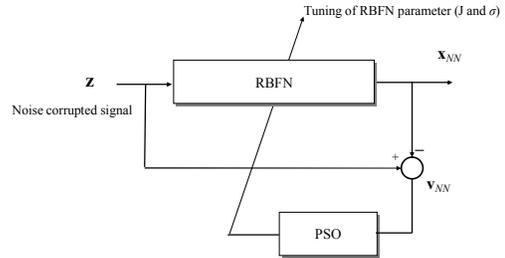


Fig. 3. PSO-RBFN functional block diagram

4.1 Fitness function for the PSO-RBFN mechanism

The RBFN is employed for filtering out the high frequency noise while the PSO is employed for obtaining the suitable network parameters so as to obtain good filtering performance.

Expressing the measurement model for certain time interval:

$$\mathbf{z} = \mathbf{x} + \mathbf{v} \quad (11)$$

where \mathbf{z} is the measurement vector, $\mathbf{z} = [z_k \ z_{k+1} \ \dots \ z_{k+n}]$; \mathbf{x} is the state vector, $\mathbf{x} = [x_k \ x_{k+1} \ \dots \ x_{k+n}]^T$; and \mathbf{v} is the measurement noise vector, $\mathbf{v} = [v_k \ v_{k+1} \ \dots \ v_{k+n}]^T$. It should be noted that the ‘vector’ here is formed by a group of n samples at n time epochs for certain state variable. Furthermore, the measurement noise vector taken at the previous epoch and is denoted as $\mathbf{v}_f = [v_{k+1} \ v_{k+2} \ \dots \ v_{k+1+n}]^T$. Taking the difference between ‘current’ epoch and ‘previous’ epoch, we have

$$\Delta \mathbf{v} = \mathbf{v} - \mathbf{v}_f = [v_k - v_{k+1} \ v_{k+1} - v_{k+2} \ \dots \ v_{k+n} - v_{k+n+1}]^T$$

Taking the variance for $\Delta \mathbf{z}$ leads to

$$\text{var}[\Delta \mathbf{z}] = \text{var}[\Delta \mathbf{x} + \Delta \mathbf{v}]$$

For $\Delta \mathbf{x}$ and $\Delta \mathbf{v}$ being mutually independent, we have

$$\text{var}[\Delta \mathbf{z}] = \text{var}[\Delta \mathbf{x}] + \text{var}[\Delta \mathbf{v}] \quad (12)$$

where $\Delta \mathbf{v}$ stands for the difference of measurement noise vector \mathbf{v} for two successively epochs. Taking the variance of $\Delta \mathbf{v}$ leads to

$$\begin{aligned} \text{var}[\Delta \mathbf{v}] &= \text{var}[\mathbf{v} - \mathbf{v}_f] \\ &= \text{var}[\mathbf{v}] + \text{var}[\mathbf{v}_f] - 2E[\mathbf{v} \cdot \mathbf{v}_f] + 2E[\mathbf{v}] \cdot E[\mathbf{v}_f] \end{aligned} \quad (13)$$

Since \mathbf{v}_f and \mathbf{v} in fact belongs to the same sequence (but taken at two successively epochs), we have

$$\text{var}[\mathbf{v}] \cong \text{var}[\mathbf{v}_f] \neq 0$$

and it is seen that

$$\text{var}[\Delta \mathbf{v}] = 2 \text{var}[\mathbf{v}] - 2E[\mathbf{v} \cdot \mathbf{v}_f] + 2E[\mathbf{v}] \cdot E[\mathbf{v}_f] \quad (14)$$

For \mathbf{v}_f and \mathbf{v} being mutually independent with the same probability density function (PDF), we have

$$E[\mathbf{v} \cdot \mathbf{v}_f] = E[\mathbf{v}] \cdot E[\mathbf{v}_f] \quad (15)$$

and the relation can be obtained:

$$\text{var}[\Delta \mathbf{v}] = 2 \text{var}[\mathbf{v}] \quad (16)$$

If the objective value of RBFN for training is \mathbf{z} and the outputs is \mathbf{x}_{NN} , the difference between the two is the noise vector:

$$\mathbf{v}_{NN} = \mathbf{z} - \mathbf{x}_{NN} \quad (17)$$

where the subscript ‘NN’ denotes the RBFN output. The RBFN output is the smoothed version of the noise corrupted signals.

Combining Equations (12), (16) and (17) leads to

$$\text{var}[\Delta \mathbf{z}] = \text{var}[\Delta \mathbf{x}_{NN}] + 2 \text{var}[\mathbf{z} - \mathbf{x}_{NN}]$$

Therefore, the fitness function for the condition of network convergence is chosen to be

$$\text{var}[\Delta \mathbf{z}] - \text{var}[\Delta \mathbf{x}_{NN}] - 2 \text{var}[\mathbf{z} - \mathbf{x}_{NN}] = 0 \quad (18)$$

For \mathbf{v}_f and \mathbf{v} being mutually independent and being zero mean, Equation (15) becomes

$$E[(\mathbf{v}_{NN}) \cdot (\mathbf{v}_{NN})_f] = 0 \quad (19)$$

where $(\mathbf{v}_{NN})_f$ basically represents the same vector as

\mathbf{v}_{NN} but only taken at the previous epoch. Equation (6) is used as the criterion of convergence for the neural network.

4.2 Feasibility check for the proposed fitness function

Validation on Equations (16), (18) and (19) will be performed. A set of 3600 sample points corrupted by the Gaussian white sequence were generated using the following function:

$$x = 200 \sin(t) \exp(-t/10) + \text{randn}$$

where $t = 0 : \pi/900 : 4\pi$ and randn stands for the unity Gaussian white sequence. Table 1 provides summary of the statistics for various sequences of concern. Equations (18) and (19) both hold and $\text{var}[\Delta \mathbf{v}]$ is approximately equal to $2 \text{var}[\mathbf{v}]$.

Table 1. Statistics for various sequences of concern.

$\text{var}[\Delta \mathbf{z}]$	2.0398
$\text{var}[\Delta \mathbf{x}]$	0.1052
$\text{var}[\Delta \mathbf{v}]$	1.9346
$2 \text{var}[\mathbf{v}]$	1.9644
$\text{var}[\Delta \mathbf{z}] - \text{var}[\Delta \mathbf{x}] - 2 \text{var}[\mathbf{v}]$	-0.0299
$E[\mathbf{v} \cdot \mathbf{v}_f]$	0.0150

4.3 Roles of the PSO-RBFN mechanism

The design strategy of the PSO-RBFN mechanism is presented.

(1) Use of RBFN as a low pass filter. RBFN is employed to filter out high frequency noises. Since the pseudo-inverse matrix approach is employed, there are two parameters (J and σ) to be determined in the RBFN. With various combinations of J and σ , it can be seen that when the value of $E[(\bar{\mathbf{v}}_{NN}) \cdot (\bar{\mathbf{v}}_{NN})_f]$ is approaching zero, so will the RMS values be, which indicates that the noise has been ideally mitigated.

(2) Optimization of RBFN parameters through PSO optimization searching process. The PSO is employed to search for the optimal RBFN parameters for filtering out high frequency noises. When $E[(\mathbf{v}_{NN}) \cdot (\mathbf{v}_{NN})_f]$ approaches zero, the RBFN output approaches the optimal values. Therefore, the trained RBFN is employed as the PSO fitness function, and the PSO will be used to search the J and σ parameters in the RBFN, as presented in Fig. 3. The other parameters for the PSO is: $w_j = 1, c_1 = c_2 = 2$. The required number of iterations (epochs) is dependent on the number of particles. To verify the effectiveness of the PSO-RBFN, an experiment was conducted. The function $y = 100 \sin(x) \exp(-x/5) + \text{randn}$ was employed. Here, 20 iterations, 5 particles were employed. The results are shown in Fig. 4. Fig. 4(a) shows the PSO-RBFN output

as compared to the actual trajectory and noise corrupted trajectory and Fig. 4(b) gives the errors for the PSO-RBFN outputs. Table 2 provides various combinations of J and σ , the fitness, and RMS error for various iteration epochs.

(3) Derivation of noise variances from the RBFN outputs. By defining the windowed version of noise $\mathbf{v}_{NN}^* = \mathbf{z}^* - \mathbf{x}_{NN}^*$, the measurement noise standard deviation derived from the RBFN outputs is based on the relation $\sigma_{v-NN} = \sqrt{\text{var}(\mathbf{v}_{NN}^*)}$, where the superscript ‘*’ stands for the windowed version (windows size is to be determined) of a signal. The process noise standard deviation derived from the RBFN outputs can be performed based on the relation $\sigma_{w-NN} = \sqrt{\text{var}(\Delta(\mathbf{x}_{NN}^*))}$.

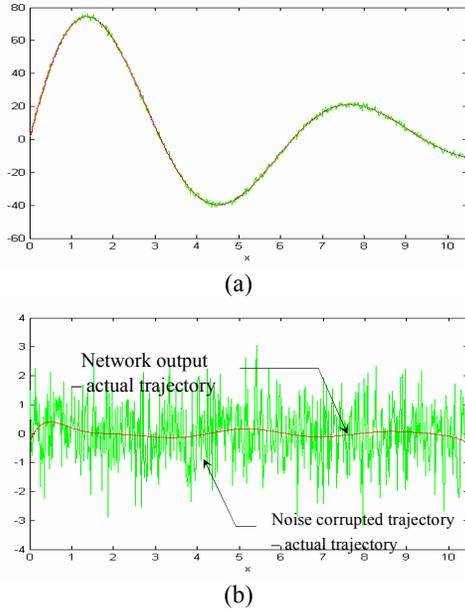


Fig. 4. (a) PSO-RBFN output as compared to the actual trajectory and noise corrupted trajectory; (b) Errors for the PSO-RBFN outputs

Table 2. Various combinations of J and σ , the fitness, and RMS error for at various iteration epochs.

Epoch	Fitness ($E[(\mathbf{v}_{NN}) \cdot (\mathbf{v}_{NN})_f]$)	RMS	Converged values	
			J	σ
1	174.3581	13.265449	5	4.0801
2	7.7724992	2.8005671	7	7.0289
3	0.0620789	0.0891904	9	4.5987
4	0.0620789	0.0891904	9	4.5987

5. Application of the PSO-RBFN for GPS/INS navigation design

Application of the PSO-RBFN mechanism to

GPS/INS navigation filter design is presented. The loosely-coupled GPS/INS architecture is investigated. Fig. 5 provides the strategy for the GPS/INS architecture aided by the PSO-RBFN mechanism.

The GPS navigation solution based on the least-squares (LS) is solved first. The measurement is the residual between GPS LS and INS derived data, which is used as the basis of KF parameter adaptation for the PSO-RBFN mechanism. The PSO-RBFN is employed for deriving parameters σ_{w-NN} and σ_{v-NN} through the relations:

$$\sigma_{w-NN} = \sqrt{\text{var}(\Delta(\mathbf{x}_{NN}^*))}; \sigma_{v-NN} = \sqrt{\text{var}(\mathbf{v}_{NN}^*)}$$

As mentioned before, the subscript ‘NN’ stands for the outputs from PSO-RBFN and the superscript ‘*’ stands for the windowed version of the signals.

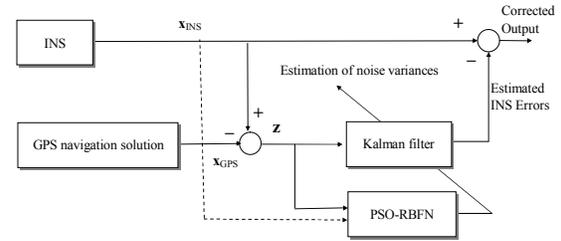


Fig. 5. The proposed loosely-coupled GPS/INS strategy

Simulation experiments have been carried out to evaluate the performance of the proposed method in comparison with the conventional methods for GPS/INS navigation processing. The computer codes were constructed using the Matlab® software. The commercial software Satellite Navigation (SATNAV) Toolbox, INS Toolbox and Navigation System Integration and Kalman Filter Toolbox by GPSof LLC was employed.

Satellite constellation was simulated and the error sources corrupting GPS measurements include ionospheric delay, tropospheric delay, receiver noise and multipath. The positioning result obtained from GPS is based on the least-squares approach. It assumed that there is no GPS failure during simulation. The INS errors are assumed to be follows: initial east velocity error: 2 m/s; initial north velocity error: 2 m/s; body-x tilt error = 0.1 milli-radian; body-y tilt error = 0.1 milli-radian; x-acceleration bias = 500 micro-g; y-acceleration bias = 500 micro-g; acceleration noise = 0.00001 km/hour²; gyro bias = 0.0015 deg/hr; gyro noise = 0.00001 deg/hr².

Fig. 6 shows the vehicle trajectory and Fig. 7 gives the velocity components and Euler angles for the vehicle for simulation. The trajectory of the aircraft can be approximately divided into three zones according to the dynamic characteristics: (1) At Zone 1, the aircraft performed highest dynamic flight, during the time interval 0-11 min; (2) At Zone 2, the aircraft performed medium dynamic flight, during the time interval 11-22 min; (3) At Zone 3, the aircraft performed straight line

flight, during the time interval 22-32 min. The characteristics of the trajectory can be approximately divided into three zones, in which low dynamic motion is involved during 698-1314 second; medium dynamic maneuvering is involved during 1314-1914 second; high dynamic maneuvering is involved during 0-698 second. The initial position of the aircraft was located at the position North 39° degrees and West 82° degrees at an altitude of 1000 meters.

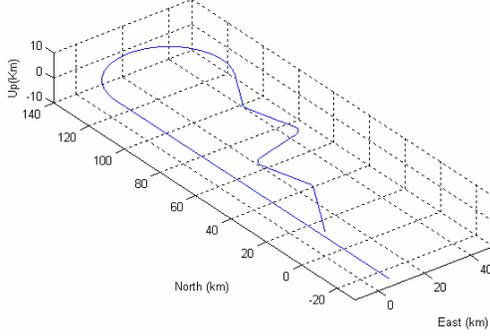


Fig. 6. The vehicle trajectory for simulation

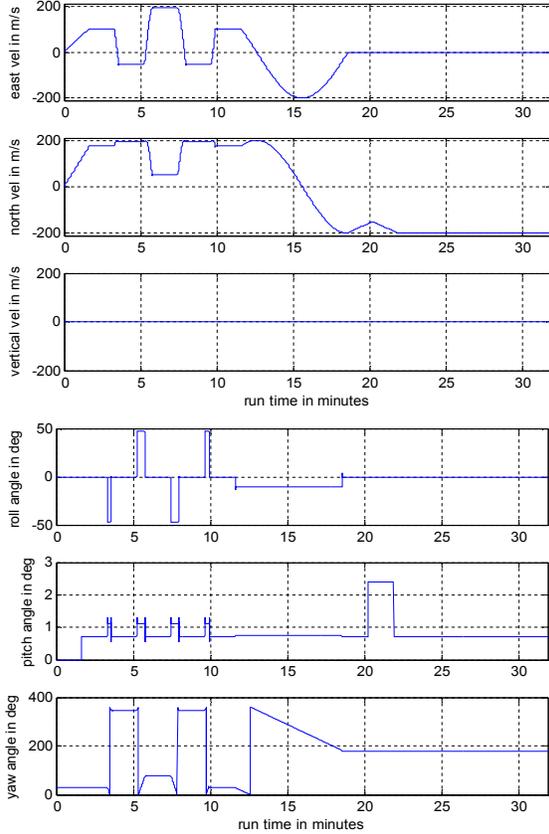


Fig. 7. Velocity components (top) and Euler angles (down) for the vehicle

The other setting on various parameters is

summarized as follows.

(1) PSO-RBFN parameters

The parameters employed in RBFN are not chosen by the designer but come from the PSO searching process when convergence is reached. There are not many parameters in PSO need to be adjusted. The parameters used in the PSO are as follows:

- Number of samples for the measurements: 300;
- Maximum J value: 30;
- Number of particles in each swarm: 5;
- Number of iterations (generations) for each searching process (epochs): 30;
- Window size for σ_{w-NN} derivation: 30;
- Window size for σ_{v-NN} derivation: 30.

(2) Kalman filter parameters

The parameters in the Kalman filter recursive loop include Φ_k , H_k , Q_k and R_k . Matrices Φ_k and Q_k are obtained through the setting:

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_a & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where $\sigma_a = 0.3 * 9.81e-9 * I$; $\sigma_g = 1e-8 * I$. Through W and G matrices, Φ_k and Q_k can be computed. In addition, $H_k = [I \ 0 \ 0 \ 0 \ 0 \ 0]$; $R_k = r_p * I$, where $r_p = 100$. The '0's in W , G and H_k are 3×3 matrices of zeros. The initial values of covariance matrix and state vector, \hat{x}_o^- and P_o^- , respectively, are

$$\hat{x}_o^- = \mathbf{0}_{18 \times 1}; \quad P_o^- = \begin{bmatrix} p_{1,1} & 0 & 0 & 0 \\ 0 & p_{1,1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & p_{18,18} \end{bmatrix}$$

where

$$\begin{aligned} p_{1,1} &= p_{2,2} = p_{16,16} = p_{17,17} = p_{18,18} = r_p; \\ p_{4,4} &= p_{5,5} = 4; \\ p_{7,7} &= p_{8,8} = p_{9,9} = 1e-8; \\ p_{10,10} &= p_{11,11} = p_{12,12} = (9.81e-4)^2; \\ p_{13,13} &= p_{14,14} = p_{15,15} = 5e-4; \\ p_{3,3} &= p_{6,6} = 0. \end{aligned}$$

The positioning errors based on the standard KF and as compared to the proposed method are shown in Figs. 8 and 9. Fig. 8 presents the result for which no differential correction is applied; Fig. 9 presents the result for which differential correction is applied. Substantial performance improvement is clearly seen for

both the modes without and with differential correction.

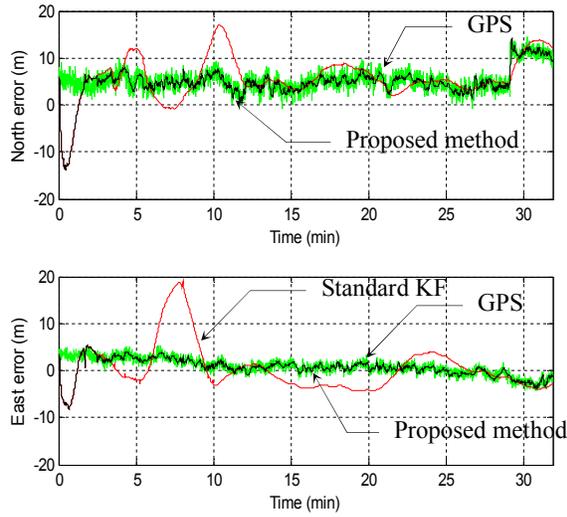


Fig. 8. Positioning errors – without differential correction

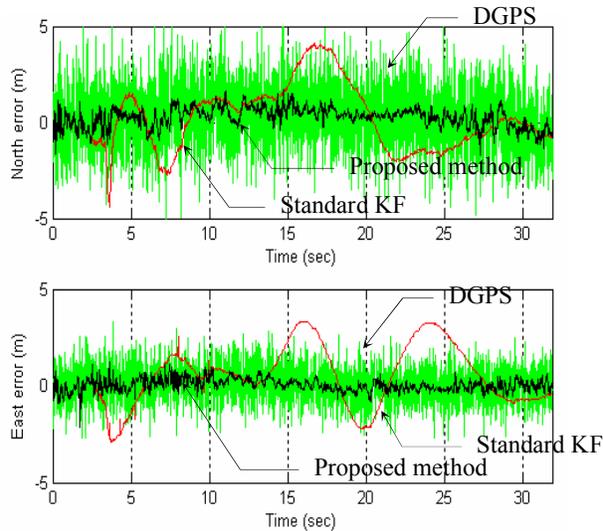


Fig. 9. Positioning errors – DGPS mode

6. Conclusions

Incorporation of PSO-RBFN mechanism into the Kalman filter design has been presented. Fitness function for the PSO-RBFN mechanism has been proposed and the feasibility has been checked. The PSO is employed to search for the optimal RBFN parameters such that the PSO-RBFN is able to filter out high frequency noises (act as a low pass filter). The PSO-RBFN filtered outputs are utilized for derivation of noise variances (or equivalently, standard deviations) in the Kalman filter. Using the proposed method, the covariance matrices for both the process dynamic and measurement models in the Kalman filter have been estimated on-line and the nonlinear dynamics errors has been identified so that the modeling error can be

compensated. The PSO-RBFN has been applied to the loosely-coupled GPS/INS navigation filter design. The results using the proposed approach have demonstrated significant positioning performance improvement in comparison with the standard Kalman filtering method.

7. Acknowledgment

This study is supported by the National Science Council of the Republic of China through Grant NSC 95-2221-E-019-026.

8. References

1. Brown R G and Hwang P Y C 1997 *Introduction to random signals and applied Kalman filtering*, John Wiley & Sons, New York, 3rd edn.
2. Hide C Moore T and Smith M 2003 Adaptive Kalman Filtering for Low-cost INS/GPS *Journal of Navigation*, 56 143-152
3. Eberhart R C and Shi Y 1998 Comparison between genetic algorithms and particle swarm optimization, 1998 annual conf on evolutionary programming.
4. Farrell J 1998 *The Global Positioning System and Inertial Navigation*, (McCraw-Hill).
5. Gelb A 1974 *Applied optimal estimation*, (MA: MIT Press).
6. GPSSoft LLC 2005 *Navigation System Integration and Kalman Filter Toolbox User's guide*
7. Haykin S 1994 *Neural Networks: A Comprehensive Foundation* (N Y: Macmillan).
8. Kennedy J and Eberhart R C 1995 Particle swarm optimization Proc. IEEE international conf. neural network 1942-1945
9. Kong X, Nebot E M and Durrant-Whyte H 1999 Development of a non-linear psi-angle model for large misalignment errors and its application in INS alignment and calibration, Proc. 1999 IEEE International Conference on Robotics & Automation, 1430-1435.
10. Mehra R K 1970 On-line identification of linear dynamic systems with applications to Kalman filtering, *IEEE Trans. Automat. Control*, AC-16 12-21.
11. Mehra R K 1972 Approaches to adaptive filtering *IEEE Trans. Automat. Control*, AC-17 693-698.
12. Mohamed A H and Schwarz K P 1999 Adaptive Kalman filtering for INS/GPS *Journal of Geodesy*, 73 193-203.