A Hierarchical SLAM/GPS/INS Sensor Fusion with WLFP for Flying Robo-SAR's Navigation

S.M. Mirzababaei[i *] and M.K. Akbari[ii]

ABSTRACT

In this paper, we present the results of a hierarchical SLAM/GPS/INS/WLFP sensor fusion to be used in navigation system devices. Due to low quality of the inertial sensors, even a short-term GPS failure can lower the integrated navigation performance significantly. In addition, in GPS denied environments, most navigation systems need a separate assisting resource, in order to increase the availability and reliability of the device. When the GPS service/information is available, the integrated SLAM system arranges for a landmark-based map using a GPS/INS feature. But in case of inaccessibility of GPS information, the latest formerly produced map plays an important role in decreasing the INS errors. In addition, a Wireless Fingerprinting (WLFP) mechanism helps us limit the errors in the system. The results of the proposed method decreases the average estimation precision on the order of 2.6m, without any performance degradation and in different experiments, which is the maximum sustainable error (below 2.66m) for flyer landing on the base. The mentioned method could be used in computer networks to schedule the services too.

KEYWORDS

Inertial Navigation System, GPS, Car Navigation, Unmanned Autonomous Vehicle, Wireless Fingerprint

1. INTRODUCTION

The Global Navigation Satellite System (GNSS) is a space-borne, radio navigation system with world-wide coverage, accessibility, and high accuracy. Its paired characteristics to the Inertial Navigation Systems (INS) make it a brilliant assisting basis for INS. There are theoretic and useful research activities aiming at the Integration of Global Positioning System (GPS) and INS to improve the accuracy and decrease the price of navigation systems [1]-[4]. Wireless signals can be also used in the environment as a fingerprint to ensure the correct position estimation of the control system. The current trends are incorporating a lower cost or equivalently a lower quality, inertial sensor with a higher performance GNSS sensor. The low-cost, light, and compact-sized GPS/INS/WLFP system is an ideal navigation system for the Unmanned Autonomous Vehicle (UAV) platform, which requires a high maneuverability, and a limited payload capacity.

The GPS/INS based systems are more reliable on the accessibility and quality of GPS data. Even a short period of satellite signal blockage can cause significant deviations in navigation results. You can imagine the conditions such as the Iceland’s volcanic eruption and also the conditions in movie recoding by small helicopters in the cities or even national disasters that can happen in megacities like Chicago, NY or LA. When our flying autonomous robot for Search and Rescue (SAR) gathers the data of the disaster scenes, it has to fly about 30 feet above the ground and pass through tall buildings in which there would be poor GPS signals, or move in the shadow of the buildings to look for things of interest and there may not be any GPS signals at all. In this situation, the GPS subsystems suspend the data delivery and report less than 4 (needed for the altitude) or 3 communicable satellites (only for the coordinates without altitude) and the integrated system needs other sensors. The smoke and dust also deny image processing in such conditions also.

The Terrain Aided Navigation System (TANS) can decrease the dependency on GNSS. This type of navigation systems typically makes use of onboard sensors and a preloaded terrain database [5][6]. For example, a Terrain Contour Matching (TERCOM) system has been applied in missile navigations. The airborne 6DoF SLAM algorithm was firstly demonstrated in [7]. The ability to estimate both the vehicle location and the map is due to the statistical correlations that exist within the estimator between the vehicle and the

[i *] Corresponding Author, S.M. Mirzababaei is the Department of Computer Engineering and IT Department, Amirkabir University of Technology, Tehran, Iran (e-mail: {mirzababaei,akbarif}@aut.ac.ir).

[ii] M.K. Akbari is with the Department of Computer Engineering and IT Department, Amirkabir University of Technology, Tehran, Iran (e-mail: {mirzababaei,akbarif}@aut.ac.ir).

landmarks.

Here the SLAM algorithm uses the GPS/INS and WLFP navigation system to provide more consistent INS/WLFP aiding information in GPS-denied situations. Figure 1 presents the SLAM/GPS/INS/WLFP integration architecture. The sensor fusion filtering works as either landmark-tracking filter or SLAM filter depending on the accessibility of GNSS observation. If GNSS provides unfailing observations, then the on-board vision or radar observations are used to build the landmark map and the SLAM/GPS/INS/WLFP filter estimates the errors in INS and map, which results in a landmark (or target)-tracking system. However, in the GPS-denied condition, the vision or radar observations are solely used to estimate the errors in INS and the map, which has consequences in the SLAM system [8][9].

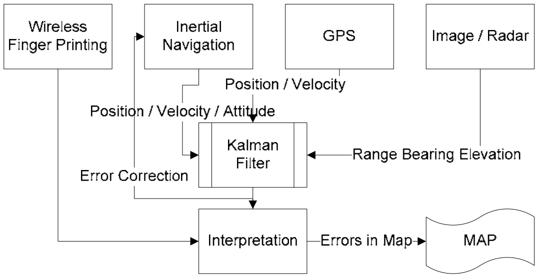


Figure 1: SLAM/GPS/INS/WLFP integrated system

The sensor fusion is a widespread and useful method used by several researchers such as [10]-[12]. In [10], Weiss and et al. used the sensor fusion of GPS and laser-scanner to restrict the location of vehicle. Their laser-scanner looks like the BTS landmarks in our research, but their vehicle moves on the earth and only needs the yaw rotation, in contrast the fliers (such as our case) have two other rotations, roll and pitch. Weiss also did not consider the GPS denial situations in their studies. In 2007, Miettinen and et al. [11] studied the forest harvesters using 2D and 3D laser-scanners that differs from our approach both in dimensions and obstacle avoidance and that they not deliberate the GPS denial conditions.

There is also another fascinating research by Schleicher and et al. [12] based on stereovision and GPS fusion. In their method, they used 3D imaging and mixed it with 3D GPS data. Although they included the GPS uncertainty in their study, they only treated it as a computational error in the formula, which differs from ours in the GPS denial situation where the data is totally noisy and out of action.

Nowadays, there has been a growing deployment wave of wireless local area networks (WLANs) by many individuals and organizations inside their homes, offices, buildings, and campuses. The acceptance of WLANs opens a new opportunity for extending the location fingerprinting in location-based services. This type of positioning system does not require specialized hardware other than the common wireless network interfaces with received signal strength measurement capability; thus, it is relatively simple to deploy compared to other techniques. Any existing WLAN infrastructure can be reused for this positioning system. Such positioning systems are viewed as the most effective and feasible solution for the indoor environments [13]-[15], and have thus become the main focuses of many explores.

2. Error Analysis

Here, the inertial navigation equation can be expressed as [16]:

$$\begin{bmatrix} \dot{\underline{r}}^n \\ \dot{\underline{v}}^n \\ \dot{C}_b^n \end{bmatrix} = \begin{bmatrix} D^{-1}\underline{v}^n \\ C_b^n \underline{f}^b - \left(2\underline{\omega}_{ie}^n + \underline{\omega}_{en}^n\right) \times \underline{v}^n + \underline{g}^n \\ C_b^n\left(\Omega_{ib}^b - \Omega_{in}^b\right) \end{bmatrix} \tag{1}$$

where

$$D^{-1} = \begin{bmatrix} \dfrac{1}{M+h} & 0 & 0 \\ 0 & \dfrac{1}{(N+h)\cos\phi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2}$$

$\dot{\underline{r}}^n$: the derivative of coordinates,

$\dot{\underline{v}}^n$: the derivative of velocities,

$\dot{C}_b^n$: the derivative of accelerations,

and M, N are radii of curvature in the meridian and prime vertical, and they considered as constant.

The error analysis utilizes perturbation methods to linearize the nonlinear system differential equations [17]. For example, the perturbation of the position, velocity, altitude, and gravity can be written as:

$$\hat{\underline{r}}^n = \underline{r}^n + \delta\underline{r}^n \tag{3}$$

$$\hat{\underline{v}}^n = \underline{v}^n + \delta\underline{v}^n \tag{4}$$

$$\hat{C}_b^n = \left(I - E^n\right)C_b^n \tag{5}$$

$$\underline{\gamma}^n = \underline{g}^n + \delta\underline{g}^n \tag{6}$$

The hat ^ denotes the calculated value, δ represents the error, $\underline{\gamma}^n$ is the normal gravity vector and E^n is the skew symmetric (7) (or cross product) form of the attitude errors.

$$E^n = (\varepsilon^n \times) = \begin{pmatrix} 0 & -\varepsilon_D & \varepsilon_E \\ \varepsilon_D & 0 & -\varepsilon_N \\ -\varepsilon_E & \varepsilon_N & 0 \end{pmatrix} \tag{7}$$

A. Position Error Dynamics

The linearized position error dynamics can be obtained by perturbing the dynamics equations for the geodetic positions. Since the position dynamics equations are functions of position and velocity, the position error dynamics equations can be obtained using the partial derivatives:

$$\delta \dot{\underline{r}}^n = F_{rr}\delta \underline{r}^n + F_{rv}\delta \underline{v}^n \tag{8}$$

where from [7]:

$$F_{rr} = \begin{bmatrix} 0 & 0 & \dfrac{-v_N}{(M+h)^2} \\ \dfrac{V_E \sin\phi}{(N+h)^2 \cos\phi} & 0 & \dfrac{-V_E}{(N+h)^2 \cos\phi} \\ 0 & 0 & 0 \end{bmatrix} \tag{8.1}$$

and

$$F_{rv} = \begin{bmatrix} \dfrac{1}{M+h} & 0 & 0 \\ 0 & \dfrac{1}{(N+h)\cos\phi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{8.2}$$

B. Velocity Error Dynamics

The velocity error dynamics equations have been derived in [16] and the F_{vv} and F_{vr} are defined as:

$$\delta \dot{\underline{v}}^n = F_{vr}\delta \underline{r}^n + F_{vv}\delta \underline{v}^n + \left(\underline{f}^n \times\right)\underline{\varepsilon}^n + C_b^n \delta \underline{f}^b \tag{9}$$

$$F_{vr} = \tag{9.1}$$

$$\begin{bmatrix} -2v_E\omega_e\cos\phi - \dfrac{v_E}{(N+h)\cos^2\phi} & 0 & \dfrac{-v_N v_D}{(M+h)^2} + \dfrac{v_E^2\tan\phi}{(N+h)^2} \\ 2\omega_c\left(v_N\cos\phi - v_D\sin\phi\right) + \dfrac{v_E v_N}{(N+h)\cos^2\phi} & 0 & \dfrac{-v_E v_D}{(M+h)^2} - \dfrac{v_N v_E\tan\phi}{(N+h)^2} \\ 2v_E\omega_e\sin\phi & 0 & \dfrac{v_E^2}{(N+h)^2} + \dfrac{v_N^2}{(M+h)^2} - \dfrac{2\gamma}{(R+h)} \end{bmatrix}$$

$$F_{vv} =$$

$$\begin{bmatrix} \dfrac{v_D}{M+h} & -2\omega_e\sin\phi - 2\dfrac{v_E\tan\phi}{N+h} & \dfrac{v_N}{M+h} \\ 2\omega_c\sin\phi + \dfrac{v_E\tan\phi}{N+h} & \dfrac{v_D + v_N\tan\phi}{N+h} & 2\omega_e\cos\phi + \dfrac{v_E}{N+h} \\ -2\dfrac{v_N}{M+h} & -2\omega_e\cos\phi - 2\dfrac{v_E}{N+h} & 0 \end{bmatrix} \tag{9.2}$$

C. Attitude Error Dynamics

The attitude error dynamics equation can be written as [16]:

$$\dot{\underline{\varepsilon}}^n = F_{er}\delta \underline{r}^n + F_{ev}\delta \underline{v}^n - \left(\underline{\omega}_{in}^n \times\right)\underline{\varepsilon}^n - C_b^n \delta \underline{\omega}_{ib}^b \tag{10}$$

$$F_{er} = \begin{bmatrix} -\omega_e\sin\phi & 0 & \dfrac{-v_E}{(N+h)^2} \\ 0 & 0 & \dfrac{v_N}{(M+h)^2} \\ -\omega_e\cos\phi - \dfrac{v_E}{(N+h)\cos^2\phi} & 0 & \dfrac{v_E\tan\phi}{(N+h)^2} \end{bmatrix} \tag{10.1}$$

$$F_{ev} = \begin{bmatrix} 0 & \dfrac{1}{N+h} & 0 \\ \dfrac{-1}{M+h} & 0 & 0 \\ 0 & \dfrac{-\tan\phi}{N+h} & 0 \end{bmatrix} \tag{10.2}$$

3. GPS/INS KALMAN FILTER

A continuous system equation can be expressed as:

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} \tag{11}$$

where F is the dynamics matrix, $\underline{x}$ is the state vector, G is a design matrix, $\underline{u}$ is the forcing vector:

$$F = \begin{bmatrix} F_{rr} & F_{rv} & 0 \\ F_{vr} & F_{vv} & \left(\underline{f}^n \times\right) \\ F_{er} & F_{ev} & \left(-\underline{\omega}_{in}^n \times\right) \end{bmatrix} \tag{11.1}$$

$$\underline{x} = \begin{pmatrix} \delta \underline{r}^n \\ \delta \underline{v}^n \\ \underline{\varepsilon}^n \end{pmatrix} \tag{11.2}$$

$$G = \begin{bmatrix} 0 & 0 \\ C_b^n & 0 \\ 0 & -C_b^n \end{bmatrix} \tag{11.3}$$

$$\underline{u} = \begin{bmatrix} \delta \underline{f}^b \\ \delta \underline{\omega}_{ib}^b \end{bmatrix} \tag{11.4}$$

The elements of $\underline{u}$ are white noise whose covariance matrix is given by:

$$E\left[\underline{u}(t)\underline{u}(\tau)^T\right] = Q(t)\delta(t-\tau) \tag{12}$$

where the operator δ denotes the Direc delta function whose unit is 1/time [18]. Q is called the spectral density matrix and has the form:

$$Q = diag\left(\sigma_{ax}^2 \quad \sigma_{ay}^2 \quad \sigma_{az}^2 \quad \sigma_{\omega x}^2 \quad \sigma_{\omega y}^2 \quad \sigma_{\omega z}^2\right) \tag{13}$$

where σ_a and σ_ω are standard deviations of accelerometers and gyroscopes, respectively. Because strap down inertial systems are usually implemented with sampled data, (11) is transformed into its discrete time form:

$$\underline{x}(t_{k+1}) = \Phi(t_{k+1}, t_k)\underline{x}(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)G(\tau)\underline{u}(\tau)d\tau \tag{11.5}$$

or in an abbreviated notation

$$x_{k+1} = \Phi_k \underline{x}_k + \underline{\omega}_k \tag{14}$$

where Φ_k is the state transition matrix, and $\underline{\omega}_k$ is the driven response at t_{k+1} due to the presence of the input white noise during the time interval (t_k, t_{k+1}). Because a white sequence of zero-mean random variables that are uncorrelated time wise, the covariance matrix associated with $\underline{\omega}_k$ is [19]:

$$E\left[\underline{\omega}_k \underline{\omega}_i^T\right] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \tag{15}$$

The analytical method to find the state transition matrix is:

$$\Phi_k = L^{-1}\left[(sI - F)^{-1}\right] \tag{16}$$

where L^{-1} represents the inverse Laplace transform and s is the Laplace transform parameter. However, for the implementation of INS, because the sampling time interval $\Delta t = t_{k+1} - t_k$ is very small, the following simple numerical approximation is preferred:

$$\Phi_k = \exp(F\Delta t) \approx I + F\Delta t \tag{17}$$

In this research Q_k is calculated using the first order approximation of the transition matrix as [20]:

$$Q_k \approx \Phi G Q G^T \Phi_k^T \Delta t \tag{18}$$

If the norm of Q_k is larger than the true one, the Kalman filter trusts the measurements more than the system. Then, the resulting estimates will be noisy due to the free passage or the measurement noise. However, the estimate does not have time lag. If the norm of Q_k is smaller than the true one, the time lag will show up. When the norm of Q_k is much smaller than the true one, the filter diverges, which results in numerical instabilities. Hence, for low cost inertial systems, Q_k must be selected pessimistically so that the trajectory can follow that of the GPS. In this paper, the elements of Q_k are increased until the filter is stabilized and the trajectory follows that of the GPS. Adaptive calculation methods can be applied to help in the tuning of Q_k [21], [22]. The derivation of the Kalman filter - a recursive, unbiased and minimum-variance estimator - starts from the random process model, and the following observation equations:

$$z_k = H_k \underline{x}_k + \underline{e}_k \tag{19}$$

where $\underline{z}_k$ at time t_k as a linear combination of the state vector, $\underline{x}_k$, plus a random measurement error, $\underline{e}_k$ [19]. The covariance matrices for the $\underline{\omega}_k$ and $\underline{e}_k$ are given by:

$$E\left[\underline{e}_k \underline{e}_i^T\right] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \tag{20}$$

$$E\left[\underline{\omega}_k \underline{e}_i^T\right] = 0, \quad \forall i, k \tag{21}$$

The implementation of the Kalman filter can be divided into two stages, the update stage and the prediction stage. In the former, the Kalman gain K_k is computed first, and then the state and the covariance are updated using the prior estimate, $\hat{\underline{x}}_{\bar{k}}$, and its error covariance, $P_{\bar{k}}$:

$$K_k = P_{\bar{k}} H_k^T \left(H_k P_{\bar{k}} H_k^T + R_k\right)^{-1} \tag{22}$$

$$\hat{\underline{x}}_k = \hat{\underline{x}}_{\bar{k}} + K_k\left(\underline{z}_k - H_k \hat{\underline{x}}_{\bar{k}}\right) \tag{23}$$

$$P_k = \left(I - K_k H_k\right) P_{\bar{k}} \tag{24}$$

In the prediction stage, the estimate and its error covariance are projected ahead:

$$\hat{\underline{x}}_{\overline{k+1}} = \Phi_k \hat{\underline{x}}_k \tag{25}$$

$$P_{\overline{k+1}} = \Phi_k P_k \Phi_k^T + Q_k \tag{26}$$

The position and velocity from the GPS can be considered as measurements. The straightforward formulation or the measurement equation can be written as:

$$\underline{z}_k = \begin{pmatrix} \underline{r}_{INS}^n - \underline{r}_{GPS}^n \\ \underline{v}_{INS}^n - \underline{v}_{GPS}^n \end{pmatrix} = \begin{bmatrix} \phi_{INS} - \phi_{GPS} \\ \lambda_{INS} - \lambda_{GPS} \\ \frac{h_{INS} - h_{GPS}}{\underline{v}_{INS}^n - \underline{v}_{GPS}^n} \end{bmatrix} \tag{27}$$

$$H_k = \begin{pmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} \end{pmatrix} \tag{28}$$

However, this approach causes numerical instabilities in calculating $\left(H_k P_{\bar{k}} H_k^T + R_k\right)^{-1}$ for the Kalman gain, K_k because Φ and λ are in radians and therefore they are very small values. This problem can be resolved if the first and second rows are multiplied by *(M+h)* and *(N+h)* cos*Φ*, respectively. Hence, the measurement equation will take the form [16]:

$$\underline{z}_k = \begin{bmatrix} (M+h)(\phi_{INS} - \phi_{GPS}) \\ (N+h)\cos\phi(\lambda_{INS} - \lambda_{GPS}) \\ \frac{h_{INS} - h_{GPS}}{\underline{v}^n{}_{INS} - \underline{v}^n{}_{GPS}} \end{bmatrix} \tag{29}$$

$$H_k = \begin{bmatrix} (M+h) & 0 & 0 & & \\ 0 & (N+h)\cos\phi & 0 & 0_{3\times3} & 0_{3\times3} \\ 0 & 0 & 1 & I_{3\times3} & 0_{3\times3} \\ & 0_{3\times3} & & & \end{bmatrix} \tag{30}$$

and the following measurement noise matrix will be used:

$$R_k = diag\left(\sigma_\phi^2 \quad \sigma_\lambda^2 \quad \sigma_h^2 \quad \sigma_{vn}^2 \quad \sigma_{ve}^2 \quad \sigma_{vd}^2\right) \tag{31}$$

which can be obtained from the GPS processing. To start a Kalman filter, the initial estimation uncertainty standard deviations must be first given. If an Inertial Measurement Unit (IMU) is initialized in stationary mode, the position uncertainly will be that of the GPS solution and the velocity uncertainty will be almost zero. The attitude uncertainty is totally dependent on the accelerometer and gyroscope biases. The GPS and IMU measurements are usually made in different times. So, the IMU's position and velocity can be interpolated using the data before and after the GPS measurements made to compose the vector $\underline{Z}_k$. Let us assume that IMU measurements are made at t_{k-1} and t_k and the GPS measurement is made at t_{GPS} . Then, the following simple linear interpolation equation can been applied (from a previous implemented version in the system) to get the position and velocity of IMU at the GPS measurement time (that is appropriate for real time implementation):

$$\underline{r}^n(t_{GPS}) = \underline{r}^n(t_{k-1}) + \frac{\underline{r}^n(t_k) - \underline{r}(t_{k-1})}{t_k - t_{k-1}}(t_{GPS} - t_{k-1}) \tag{32}$$

$$\underline{v}^n(t_{GPS}) = \frac{t_k - t_{GPS}}{t_k - t_{k-1}}\underline{v}^n(t_{k-1}) + \frac{t_{GPS} - t_{k-1}}{t_k - t_{k-1}}\underline{v}^n(t_k) \tag{33}$$

For high dynamic applications, a higher order interpolation is needed. For example, the Lagrange interpolation equation can be used [23]:

$$\underline{r}^n(t_{GPS}) = \sum_{i=k-m-1}^{k+m} \underline{r}^n(t_i) \prod_{j=k-m-1}^{k+m} \frac{t_{GPS} - t_j}{t_i - t_j} \tag{34}$$

$$\underline{v}^n(t_{GPS}) = \sum_{i=k-m-1}^{k+m} \underline{v}^n(t_i) \prod_{\substack{j=k-m-1 \\ j\neq i}}^{k+m} \frac{t_{GPS} - t_j}{t_i - t_j} \tag{35}$$

where *2m + 1* is the order of interpolation. When *m=0*, (34) and (35) are identical to (32) and (33), respectively. Since both sensors cannot be installed at the same place in the host vehicle, the position and velocity of the IMU are different from those of the GPS. This is called the level-arm effect. The level-arm correction for the position and velocity, considering $\Delta\underline{r}^b$ as the offset vector of the GPS antenna from the center of the IMU in the body frame, we will have:

$$\underline{v}_{IMU}^{n} = \underline{v}_{GPS}^{n} - C_b^n \Omega_{nb}^b \Delta\underline{r}^b \tag{36}$$

$$\underline{r}_{IMU}^{n} = \underline{r}_{GPS}^{n} - \begin{bmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\phi} & 0 \\ 0 & 0 & -1 \end{bmatrix} C_b^n \Delta\underline{r}^n \tag{36.1}$$

4. Landmark Model

When the flying Robo-SearchAndRescue has to gather the data and flies through tall buildings or stop in the shadow of the buildings. The Buildings may block the GPS signals. So the GPS suspends the data delivery and report less than 4 (needed for the altitude) or 3 (only for the x-y coordinates without altitude) so the integrated system has to refer to other sensors. There can be some landmarks or wireless signals that can be used to guess the true location. Hence the ith landmark simply becomes: $m_i^n(k) = m_i^n(k-1)$. If a new landmark is observed, this external map is the dynamically augmented with new landmark position.

5. Complementary Algorithm

Now we focus on the complementary SLAM/GPS/INS/WLFP Algorithm. In this work, the Kalman Filter (KF) is used as the state estimator and the WLFP will be explained in the next section.

A. Augmented Error State

In the paired SLAM, the state is now defined as the error state of the vehicle on the map:

$$\delta x(k) = \left[\delta x_v(k), \delta x_m(k)\right]^T \tag{37}$$

The inaccurate state of the vehicle $\delta x_v(k)$ comprises the errors in the INS indicated position, velocity and attitude expressed in the navigation frame:

$$\delta x_v(k) = \left[\delta p^n(k), \delta v^n(k), \delta\Psi^n(k)\right]^T \tag{38}$$

The error state of the map $\delta x_m(k)$ includes the errors in 3D landmark positions in the navigation frame. The size of state is also dynamically changes with the new landmark error after the start,

$$\delta x_m(k) = \left[\delta m_1^n(k), \delta m_2^n(k), \ldots, \delta m_N^n(k)\right]^T \tag{39}$$

where *N* is the current number of registered landmarks and each one consists of a positioning error in 3D [24].

B. SLAM Error Model

The linearized SLAM system in discrete time can be written as:

$$\delta x(k+1) = F(k)\delta x(k) + G(k)w(k) \tag{40}$$

where $\delta x(k)$ is the error state vector, *F(k)* is the system transition matrix, *G(x)* is the system noise input matrix and *w(k)* is the system noise vector with noise variance *Q(k)*. The continuous time SLAM/Inertial error model is based on misalignment angle dynamics and stationary landmark model which is a random constant [25]:

$$\left[\begin{array}{c} \delta \dot{p}^n \\ \delta \dot{v}^n \\ \delta \dot{\Psi}^n \\ \hline \delta \dot{m}_m^n \end{array}\right] = \left[\begin{array}{c} \delta v^n \\ C_b^n f^b \times \delta\Psi^n + C_b^n \delta f^b \\ -C_b^n \omega^b \\ \hline 0_m \end{array}\right] \tag{41}$$

where f^b and ω^b are acceleration and rotation rates measured from IMU, δf^b and $\delta\omega^b$ are the associated errors in IMU measurement, C_b^n is the direction cosine matrix formed from the quaternion. The matrices *F(k)*, *G(k)* and *Q(k)* are given by:

$$F(k) = \begin{bmatrix} I & \Delta t f^n & \left(\Delta t^2/2\right) f^n & 0 \\ 0 & I & \Delta t f^n & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I_{m\times n} \end{bmatrix} \tag{42}$$

$$G(k) = \begin{bmatrix} 0 & 0 \\ \sqrt{\Delta t} C_b^n(k) & 0 \\ 0 & -\sqrt{\Delta t} C_b^n(k) \\ 0_m & 0_m \end{bmatrix}$$

$$Q(k) = \begin{bmatrix} \sigma_{\delta f}^2 & 0 \\ 0 & \sigma_{\delta\omega}^2 \end{bmatrix}$$

with $\sigma_{\delta f}$ and $\sigma_{\delta\omega}$ representing noise variance of acceleration and rotation rate respectively [26].

C. Observation model

The linearized observation model can be obtained in terms of the observation residual, or measurement differences, $\delta z(k)$ and the error states, $\delta x(k)$,

$$\delta z(k) = H(k)\delta x(k) + v(k) \tag{43}$$

with *H(k)* being the linearized observation Jacobian and *v(k)* being the observation noise with noise strength matrix *R(k)*. The error observations are generated by subtracting the measured quantity, *z(k)*, from the INS predicted quantity $\hat{z}(k)$:

$$\delta z(k) = \hat{z}(k) - z(k) \tag{44}$$

The onboard sensor provides relative observations between vehicle and landmarks. The non-linear observation equation relates these observations to the state as follows:

$$z(k) = h(x(k), v(k)) \tag{45}$$

where *h(.)* is the non-linear observation model at time *k*, *v(k)* and is the observation noise vector. Since the observation model predicts the range, bearing, and elevation for the i^{th} landmark, it is only a function of the i^{th} landmark and the vehicle state. Therefore (45) can be further expressed as:

$$z_i(k) = h(x_v(k), x_{mi}(k), v_i(k)) \tag{46}$$

with $z_i(k)$ and $v_i(k)$ being the i^{th} observation and its associated additive noise in range, bearing and elevation with zero mean and variance of *R(k)*. The initial landmark position in the navigation frame is then computed:

$$m_i^n(k) = p^n(k) + C_b^n(k)p_{bs}^b + C_b^n(k)C_s^b p_{sm}^s(k) \tag{47}$$

where $p_{bs}^b(k)$ is the lever-arm offset of the sensor from the vehicle's center of gravity in the body frame, C_s^b is a direction cosine matrix which transforms the vector in the sensor frame (such as camera installment axes) to the body frame, and $p_{sm}^s(k)$ is the relative position of the landmark from the sensor expressed in the sensor frame which is computed from the observation:

$$p_{sm}^s(k) = \begin{bmatrix} \rho\cos\phi\cos v \\ \rho\sin\phi\cos v \\ \rho\sin v \end{bmatrix} \tag{48}$$

with ρ, ϕ and v being the range, bearing and elevation angle respectively, measured from the onboard sensor. Hence the predicted range, bearing and elevation between the vehicle and the i^{th} landmark in (46) can now be obtained by rearranging (48):

$$z_i(k) = \begin{bmatrix} \rho \\ \phi \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2 + z^2} \\ \tan^{-1}(y/x) \\ \tan^{-1}\left(z/\sqrt{x^2 + y^2}\right) \end{bmatrix} \tag{49}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = p_m^s(k) = C_b^s C_n^b(k)\left[m_i^n(k) - p^n(k) - C_b^n(k)p_{bs}^b\right] \tag{50}$$

The observation model is non-linear and has two composite functions; a coordinate transformation from the navigation frame to the sensor frame, and a transformation from Cartesian coordinates to polar coordinates. By calculating Jacobian of this equation, a linearized discrete model is obtained.

$$H(k) = \begin{bmatrix} \frac{\partial p}{\partial p^n} & \frac{\partial p}{\partial v^n} & \frac{\partial p}{\partial \Psi^n} \\ \frac{\partial \omega}{\partial p^n} & \frac{\partial \omega}{\partial v^n} & \frac{\partial \omega}{\partial \Psi^n} \\ \frac{\partial \theta}{\partial p^n} & \frac{\partial \theta}{\partial v^n} & \frac{\partial \theta}{\partial \Psi^n} \end{bmatrix}, R(k) = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \tag{51}$$

If vision or radar information is available, *δz(k)* is obtained by subtracting the range, bearing and elevation of the sensor from the INS indicated range, bearing and elevation, then it is fed to the integrated fusion filter to estimate errors in the vehicle and on the map. If the GPS position/velocity observation is used, the observation model simply becomes a linear form with:

$$H(k) = \begin{bmatrix} I & 0 & 0 & 0_m \\ 0 & I & 0 & 0_m \end{bmatrix}, R(k) = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \tag{52}$$

If the GPS information is available, *δz(k)* is formed by subtracting the position and velocity of the GPS from the INS indicated position and velocity, then they are fed to the fusion filter to estimate the errors in the vehicle and in the map [24] .

D. Kalman Filter Prediction

With the state transition and observation models defined in (40) and (43), the estimation procedure is then performed. The state and its covariance are predicted using the process noise input. The state covariance is propagated using the state transition model and the process noise matrix by:

$$\delta x(k \mid k-1) = F(k)\delta x(k-1 \mid k-1) = 0 \tag{53}$$

$$P(k \mid k-1) = F(k)P(k-1 \mid k-1)F^T(k) + G(k)Q(k)G^T(k) \tag{54}$$

E. Kalman Filter Estimation

When an observation occurs, the state vector and its covariance are updated according to

$$\delta x(k \mid k) = \delta x(k \mid k-1) + W(k)v(k) = W(k)v(k) \tag{55}$$

$$P(k \mid k) = P(k \mid k-1) - W(k)S(k)W^T(k) \tag{56}$$

where the innovation vector, the Kalman weight, and the innovation covariance are computed as,

$$v(k) = z(k) - H(k)\delta x(k \mid k-1) = z(k) \tag{57}$$

$$W(k) = P(k \mid k-1)H^T(k)S^{-1}(k) \tag{58}$$

$$S(k) = H(k)P(k \mid k-1)H^T(k) + R(k) \tag{59}$$

Once the observation estimation has been processed successfully, the estimated errors are now fed to the external INS loop and the map for correction.

6. Location Finger Printing with RBF Network

The approximation of functions is one of the most general uses of artificial neural networks. A radial basis function (RBF) neural network is usually trained to map one vector into another vector, where the pairs form the training set. From this point of view, learning is equivalent to finding a surface in a multidimensional space that provides the best fit to the training data. Generalization is therefore synonymous to interpolation between the data points along the constrained surface generated by the fitting procedure as the optimum approximation to this mapping. The principle of RBF networks is to fit a weighted sum of radial function φ to the function *f* to approximate. Such function φ depends only on the norm of the difference between their argument and a center called cancroids, and they generally can be tuned by a width factor:

$$f(x)=\sum_{j=1}^{p} w_j \varphi_j\left(\left\|x-u_j\right\|\right) \tag{60}$$

The most common radial function in practice is a Gaussian Kernel given by (61) where σ_j is the width factor of kernel j.

$$\varphi_j\left(\left\|x-u_j\right\|\right)=e^{\frac{-(x-u_j)^2}{\sigma_j^2}} \tag{61}$$

Generally in location fingerprint, a fingerprint f is labeled with location information l. The location fingerprints and their labels (e.g., location information) are maintained in database and are used during the on-line phase to estimate the location. The label and fingerprint are usually denoted as a tuple of *(l, f)*. It is commonly acknowledged that the radio signal strength (RSS) is the simplest and most effective RF signature for location fingerprints because it is readily available in all WLAN interface cards. The RSS [13] is more location-dependent than the signal-to-noise ratio (SNR) because the noise component is rather random in nature. However, the RSS itself fluctuates over time for each access point and location [13]. Each RSS element can be considered as a random variable; therefore, it can be captured by recording its descriptive statistics parameters.

Battiti et al. [27] point out that the location information l for indoor location can be recorded in two forms as either a tuple of coordinates (Regression) or an indicator variable (Classification). The tuple of real coordinates can vary from one dimension to five dimensions which include the three dimensions for space and two dimensions for orientation variables expressed in spherical coordinates [15]. For instance, a location information of a two-dimension system with an orientation could be expressed as a triplet $l = \{(x, y, d) \mid x, y \in R^2, d \in \{North, East, South, West\}\}$. In the case of indicator variable, the scope of location covers a wide area. An example is given by $l = \{-1, 1\}$ [27]. With inclusion of the above parameter in (60), the output can be expressed by:

$$\hat{L}(f)=\sum_{j=1}^{p} w_j \varphi_j\left(\left\|f-u_j\right\|\right) \tag{62}$$

and

$$\varphi_j\left(\left\|f-u_j\right\|\right)=e^{\frac{-(f-u_j)^2}{\sigma_j^2}} \tag{63}$$

where f : R x Q matrix of Q input measures of RSS signal vectors and l : S x Q matrix of Q target location vectors. In our method we consider the following parameters: the number of the samples in the training set, the spread of Gaussian Kernel, and the number of kernels that could be designed by user. According to these parameters the parameter u_j and w_j will change and get adjusted in such a way that the global mean-squared error between the desired outputs l_i and the estimated outputs $\hat{L}(f_i)$ would decrease to less than a desired objective value (the default objective parameter value is 0).

$$Goal=0.5\sum_{i=1}^{Q}\left(L_i-\hat{L}(f_j)\right)^2 \tag{64}$$

To effectively evaluate the application of RBFN approach for location fingerprinting, the data set of Wilma project was used [27],[28]. This system was based on a wireless LAN using the IEEE802.11b (Wi–Fi) standard. The LAN was composed of six AVAYA WP-II E access points, equipped with external omni directional antennas. The RSS measurements consist of 257 sampling points [29]. Therefore, the fingerprint is a vector of RSS measured from six access points for each location and the location vector arranged by two coordinates *(x,y)*. We use these vectors for regression analysis, and adapt them to location vector indicator for classifier. We use Leave-One-Out method in which every other samples are skipped in each row of grid points, making a new data set (training set) with others, and the RBFN is trained with this training set and examined by the skipped sample fingerprint. Then implementation results for location by RBF network (RBFN) are compared with real values of location to compute the error.

A. Evaluation

As we mentioned earlier, the data set in Wilma Project was used to implement several experiments using the proposed methods. Earlier, several basic methods in location fingerprinting were fed with this data set by which we can evaluate our methods and outcomes. Five different previous studies with the same data set are chosen to compare with our methods and to evaluate the proposed models in all dimensions. The first algorithm is called the SVM (Support vector machine) which benefits from the SRM (structural risk minimization) principle, by minimizing a boundary on the basis of VC-dimension (Vapnik-Chervonenkis dimension, interested reader can refer to [30] for a details). The next algorithm is called WKNN (weighted K nearest neighbors) which follows the following steps for location fingerprinting:

(1) Find within the training set the k indices i_1, ..., i_k whose radio strength arrays: f_{i1}, … ,f_{ik} that are nearest (according to a given radio space metric) to the given f vector.

(2) Calculate the estimated position information by the following average, weighted with the inverse of the distance between signal strength tuples.

$$\hat{L}(f)=\frac{\displaystyle\sum_{j=1}^{k}\frac{l_{ij}}{d\left(f_{ij},f\right)+d_0}}{\displaystyle\sum_{j=1}^{k}\frac{1}{d\left(f_{ij},f\right)+d_0}} \tag{65}$$

where $d(f_{i\,j}, f)$ is the radial distance between the two n-tuples (for example the Euclidean distance) measured in dBm, and d_0 is a small real constant (d_0 = 0.01dBm in our tests) used to avoid division by zero. Another approach was implemented before on the data set is the Bayesian modeling (BAY), which is a method based on the conditional probability estimation, and it requires the knowledge of the signal propagation model, either in the form of an empirical distribution from repeated observations on each physical point in a given set, or by selecting a suitable radio propagation model and by estimating its parameters on the basis of empirical observations [34]. The following model was implemented on this data:

$$l_i = b_0^{(i)} + b_1^{(i)} d_{AP_i}(f) + b_2^{(i)} w_{AP_i}(f) \qquad (66)$$

where $d_{APi}(f)$ represents the vector of distances of the physical point y from the access points (logarithms are computed component wise), $w_{APi}(y)$ represents the number of walls and b_0 becomes a constant term. This probability distribution can be used in turn to calculate the position in many ways. The derived results for comparing, achieved by two possible position estimators be used to determine the users location estimate $\hat{L}$, namely the average position (67) and the maximum likelihood estimator (68) [27].

$$\hat{L}_{\max} = \max \arg_f P(l \mid f) \qquad (67)$$

$$\hat{L}_{Average} = \int fdp(l \mid f) \qquad (68)$$

For the last method, Multi-layer Perceptron (MLP) with three-layer model was used, where the first layer (input) has six neurons, the second (hidden) layer has 8 neurons and the third (output) has two neurons in the regression problem. To have a fair evaluation, we designed four different algorithms for location fingerprinting on the basis of signal strengths. First, we provided a model based on nearest neighbor algorithm, applied it to each sample, picking the location of sample j from data set that has the shortest signal distance with it. So, the selected sample j has the condition:

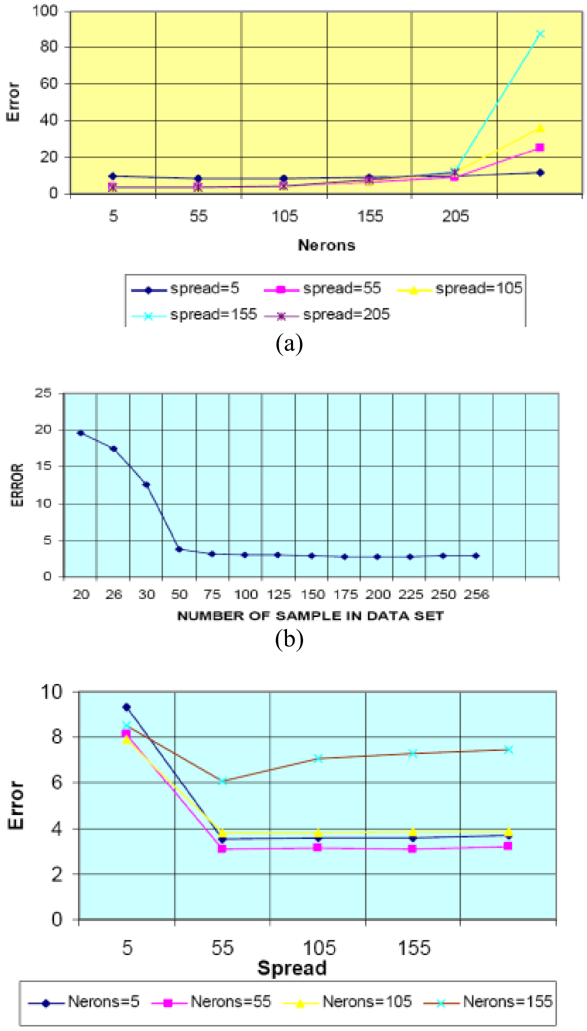
$$Dis(f,s_j) < Dis(f,s_k) \quad \forall K \neq j \qquad (69)$$

Another approach that we employed on this data set is the standard k nearest neighbors that instead of picking only the location of the closest sample, the average of K nearest neighbors' samples are selected. To find the best K, we apply the algorithm for different k values and compare the results according to the least average and the value for different k values.

B. Regression results

In the regression state, the location is a vector of real coordinates and the network trained by location vector of samples with two real coordinates *(x,y)*. The network first trained for achieving the optimum parameter according to the mean error of all samples.

Figure 2 illustrates the result of experiments based on the changes in mean errors with respect to these parameters. To come up with the results, the parameters are chosen as: number of samples in data set=257, spread=45 and number of neurons=30. As it can be seen in Figure 2-a, it is possible to reduce the data set to 70 samples or less without significant changes in the average error. Therefore, a combination of RBFN with the nearest dataset approach is proposed. In this approach, we design the neural network with a reduced number of samples. For each sample in the testing process, we arrange the training data set with samples that have the shortest RSS distance with an index (picking) sample. Like before, we calculate the optimum parameters and in the result, the following parameters are derived: number of samples in data set=70, spread=12 and number of neurons=10.



(a)

(b)

(c)

Figure 2: the change mean error with a) the size of data set, b) the neurons, c) the spread of radial basis function

Figure 3 and Table 1 provide statistics regarding the error distribution of the aforementioned techniques for the regression state. The first five data rows report experimental results of the previous known techniques [27] and the last four rows list the results of our methods. It is obvious that the proposed approaches outperform the other techniques, more specifically, the combinational approach of the RBFN with the *K* nearest data set. This method also presents an improvement in execution time due to the smaller training data set. The tests also showed that the Weighted *K* Nearest Neighbors and the Support Vector Machine (SVM) outcomes are comparable, and the two global models: neural networks and Bayesian inference (BAY) models suffer more than 30% performance degradation in compared with the best results. Although the average estimation precision is in the order of 2.6m, quantities reported in Table 1 show that three measures out of four have an error below 3.4m, and only one in 20 has an error higher than 5.5m with the RBF method with K nearest data set.

7. Conclusion

The model exploration showed that the SLAM system with a range, behavior, and elevation sensor can constraint the INS errors effectively, performing an on-line map building in unknown terrain environments. The SLAM augmented GPS/INS system possesses two capabilities of landmark tracking and mapping using GPS information, and more importantly, aiding the INS under GPS-denied situation. As a result, the SLAM augmented low-cost GPS/INS system can be effectively applied to various GPS-denied situations, such as urban canyons, indoor, or even underwater using multi-level location estimation system, by other context signals such as wireless finger printing [31]. The same sensor fusion approach could be used to schedule the services in the pervasive environment by combining the measured network parameters.

TABLE 1
LEAVE-ONE-OUT ESTIMATION ERROR DISTRIBUTION

ALGORITHM	AVERAGE	50%	75%	90%	95%	PROPOSED
SVM	3.04 ±0.10	2.75	3.96	5.12	6.09	Ref[27]
WKNN	3.06 ±0.10	2.84	3.93	5.16	5.79	Ref[27]
BAY (Average likelihood)	3.35 ±0.11	3.04	4.39	5.61	6.61	Ref[27]
MLP	3.18 ±0.11	2.82	4.01	5.4	6.73	Ref[27]
BAY (max. likelihood)	3.83 ±0.15	3.42	5.14	6.83	8.42	Ref[27]
RBF	2.85 ±0.10	2.61	3.85	5.21	6.09	Main method of this article
Nearest neighbors	3.61 ±0.10	3.81	5.28	6.11	7.4	Implement in this article for comparing
K Nearest neighbors (k=4)	3.1 ±0.10	2.99	4.02	5.11	5.77	Implement in this article for comparing
RBFN with K Nearest Data Set (K=70)	2.6 ±0.10	2.56	3.34	4.42	5.53	Main method of this article

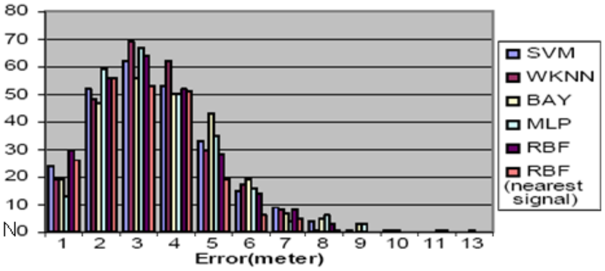


Figure 3: Leave-one-out estimation error distribution for the different algorithms

8. References

[1] A.B. Chatfield, ”Fundamentals of High Accuracy Inertial Navigation", American Institute of Aeronautics and Astronautics, 1997.

[2] S. Kim, K. Choi, S. Lee, J. Choi, T. Hwang, B. Jang, and J. Lee, "A Bimodal Approach for Land Vehicle Localization", ETRI Journal, vol. 26, no. 5, Oct. 2004, pp. 497-500.

[3] S.Y. Cho, B.D. Kim, Y.S. Cho, and W.S. Choi, "Multi-model Switching for Car Navigation Containing Low-Grade IMU and GPS Receiver", ETRI Journal, vol. 29, no. 5, Oct. 2007, pp. 688-690.

[4] M.S. Grewal, L.R. Weill and A.P. Andrews, "Global Positioning Systems, Inertial Navigation, and Integration", John Wiley & Sons, Inc ., 2001.

[5] W.R. Baker and R.W. Clem, "Terrain contour matching (TERCOM) premier", ASP-TR-77-61, Aeronautical Systems Division, Wright-Patterson, 1997.

[6] S.B. Williams, M.W.M.G. Dissanayake and H. Durrant-Whyte, "Towards terrain-aided navigation for underwater robotics ", In Advanced Robotics, 15(5):533-550, 2001.

[7] J. Kim and S. Sukkarieh, "Autonomous Airborne Navigation in Unknown Terrain Environments", IEEE Transactions on Aerospace and Electronic Systems, 40(3):1031-1045, July, 2004.

[8] S.Y. Cho and W. S. Choi, "Performance Enhancement of Low-Cost Land Navigation System for Location-Based Service", ETRI Journal, vol. 28, no. 2, Apr. 2006, pp. 131-144.

[9] J. Guivant and E. Nebot, "Optimizations of the simultaneous localization and map building algorithm for real-time implementation", IEEE Transactions on Robotics and Automation, 17(3):242-257, 2001.

[10] T. Weiss, J. Spruck and K. Dietmayer, "A Scalable Sensor Fusion Framework for the Localization of a Vehicle on Detailed Digital Maps Using Laserscanners", Multisensor Fusion and Integration for Intelligent Systems, IEEE International Conference, pp.444–449, Sept. 2006.

[11] M. Miettinen, M. Ohman, A. Visala, P. Forsman, "Simultaneous Localization and Mapping for Forest Harvesters", IEEE International Conference on Robotics and Automation, pp.517-522., Apr. 2007.

[12] D. Schleicher, L.M. Bergasa, M. Ocana, R. Barea, M.E. Lopez, "Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion", Dept. of Electronics, Univ. of Alcala, Madrid, Spain; IEEE Transactions on Intelligent Transportation Systems, ISSN: 1524-9050 Sep. 2009.

[13] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system", in Proc. IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00), pp. 775-784, Mar. 2000.

[14] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On indoor position location with wireless LANs", in Proc. IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications Sept. 2002.

[15] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao, "A wireless LAN-based indoor positioning technology", IBM Journal of Research and Development, vol. 48, no. 5,6,(Sept, Nov) pp. 617-626, 2004.

[16] H. Arghavan, "Processing Unit Design and Implementation of Fault Tolerant Navigation System", Master of Science Thesis, Amirkabir University of Technology, 2003.

[17] K.R. Britting, "Inertial Navigation Systems Analysis", John Wiley & Sons, 1971.

[18] S. Eun-Hwan, "Accuracy improvement Of Low Cost INS/GPS for Land Applications", UCGE Reports Number 20156, The University of Calgary, Calgary, Canada, 2001.

[19] R.G. Brown and P.Y.C. Hwang, "Introduction to random signals and applied Kalman filtering", 3rd Ed., New York, Wiley, 1997.

[20] J.A. Farrell and M. Barth, "The Global Positioning System & Inertial Navigation", New York, 2000.

[21] O.S. Salychev, "Inertial Systems in Navigation and Geophysics", Bauman MSTU Press, Moscow, 1998.

[22] A.H. Mohamed, "Optimizing the Estimation Procedure in INS/GPS Integration for Kinematic Applications", UCGE Reports Number 20127, Dept. of Geomatics Engineering, The university of Calgary, 1999.

[23] S.D. Conte and C.D. Boor, "Elementary Numerical Analysis: An Algorithmic Approach", McGraw-Hill, 3rd edition, 1980.

[24] J. Kim and S. Sukkarieh, "6Dof SLAM aided GNSS/INS Navigation in GNSS Denied and Unknown Environments", Journal of Global Positioning Systems, v. 4, pp. 120, 2005.

[25] J. Kim, "Autonomous Navigation for Airborne Applications", Ph.D. thesis, Australian Centre for Field Robotics, The University of Sydney, 2004.

[26] J. Kim, M. Ridley, E. Nettleton, S. Sukkarieh, "Real-time Experiment of Feature Tracking/Mapping using a low-cost Vision and GPS/INS System on am UAV platform", Journal of Global Positioning Systems, v. 3, pp. 167, 2004.

[27] R. Battiti, M. Brunato, and A. Villani, "Statistical learning theory for location fingerprinting in wireless lans", Technical Report. Oct. 2002 , [Online]. Available: http://rtm.science.unitn.it/»battiti/archive/86.pdf

[28] Webpage of allied company, http://www.alliedworld.com/ servlets/Home.

[29] Work data, Available: at http://ardent.unitn.it/software/ data.

[30] R. Battiti, A. Villani, and T. Le Nhat, "Neural network models for intelligent networks: deriving the location from signal patterns", in Proc. of AINS2002, UCLA, 2002.

[31] K.W. Min, K.W. Nam, and J.W. Kim, "Multilevel Location Trigger in Distributed Mobile Environments for Location-Based Services", ETRI Journal, vol. 29, no. 1, Feb. 2007, pp. 107-109.