Proceedings of the 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control - CESCIT 2012
3-5 April 2012. Würzburg, Germany

Pose Estimation By Multisensor Data Fusion Of Wheel Encoders, Gyroscope, Accelerometer And Electronic Compass

Ahmad Kamal Nasir, Hubert Roth

Lehrstuhl für Regelungs- und Steuerungstechnik (RST), Department Elektrotechnik und Informatik, Fakultät IV Universität Siegen, 57068, Siegen Germany, (email: {ahmad.nasir, hubert.roth}@uni-siegen.de).

Abstract: In this paper we describe a method for combining data from multiple onboard sensors to determine a mobile robot pose. An error model for a gyroscope, a dual-axial accelerometer and wheel encoders are derived for estimating the mobile robot's pose. A tri-axial magnetometer measures the magnetic field strength which is used as a criterion for acceptance of electronic compass readings to correct the azimuth of the mobile robot's orientation. The errors in each sensor are estimated mutually rather than independently considering each sensor error model. The final estimated pose is used to navigate, guide and control a mobile robot in an indoor environment.

Keywords: Localization, Navigation, Pose Estimation, Data Fusion.

1. INTRODUCTION

Mapping and pose estimation are very closely related to each other. Accurate pose estimation is fundamental to mobile robots' navigation, guidance, localization and mapping. Dead reckoning is a primitive and well known method for pose estimation but depends on wheel traction, alignment, diameter and distance between wheels (Borenstein, 1996). Some mobile robot's kinematic parameters can be estimated in an offline mode using least square methods but others such as bumps, friction coefficients, and slippage are random in nature and therefore difficult to predict. Inertial Navigation system is a mature version of dead reckoning system, which comprises of gyroscope and accelerometers, measures motion with respect to inertial space. Gyroscope sensor has been used (Barshan, 1995; Borenstein, 1996) to find the mobile robot's orientation by integrating angular velocity about rotation axis but it suffers from the problem of gyroscope drift, random bias and scale factors, furthermore the gyroscope scale factor is dependent on temperature and linear within a limited range as stated by Ojeda (2001). Accelerometer has been used (Komoriya, 1994) to find the velocity of the mobile robot by integrating the acceleration but it also suffers from the problem of drift, random bias and scale factors. Electronic compass limit the growth of the unbounded errors in robot's orientation estimation (Hoshino, 1995; Zhangjun, 2010). Electronic compass finds the absolute orientation of the robot using the Geo-magnetic field but it suffers from the problems of nearby ferromagnetic materials and electric, magnetic sources (Sun, 2009).

To compensate the characteristic deficiencies of individual sensor measurements and to merge measurements from redundant sensors, data fusion can be performed to get the optimal estimate of the mobile robot pose. Recursive and non-recursive algorithms can be applied to fuse the data. Furthermore, there are different strategies to fuse the data which depends upon the configuration and availability of sensor model. Kalman filter is a recursive algorithm which has been widely used to fuse the data from redundant sensors in various formulations (Zhang, 2010; Von der Hardt, 1996; Xiang, 2005).

In this research work we have derived and implemented a multisensor data fusion algorithm to estimate the pose of the robot through multiple onboard sensors (wheel encoders, gyroscope, accelerometer, magnetometer, and electronic compass). To correct the orientation of the mobile robot electronic compass is used. The confidence of the electronic compass measurements are judged by the tri-axial magnetometer. The methodology of the data fusion framework is as follows, first of all, the entire error model for wheel encoders is derived which is used to estimate the errors in linear and angular velocity of the mobile robot. The error model for gyroscope and accelerometer are also derived with error perturbation method to estimate the errors in linear and angular velocities. Later the indirect Kalman filter is applied on the error model to estimates the errors in the pose and parameters. The estimated state vector is feedback to compensate the parameters and pose calculations. The final mobile robot pose calculation is outside the Kalman filter recursive estimation loop therefore comparatively low iteration frequency of the filter is required, because the dynamics of the robot is outside the filtering loop. For the comparison of the estimated results against the real results a rotating PMD (Photon Mixed Device) camera and ultrasonic sensor are used to validate the end robot position. PMD is a time of flight 3D camera which can be used to measure the distance information (Pursak, 2008).

2. ROBOT KINEMATIC MODEL

The robot, TOM3D (Tele Operated Mobile 3D), used in this research work has a differential drive wheel base with two wheels and one castor wheel as shown in fig. 1. The dynamic

978-3-902661-97-5/12/$20.00 © 2012 IFAC

10.3182/20120403-3-DE-3010.00068

model of the robot is not used because various parameters are not available furthermore process nonlinearities complicate the model development process.

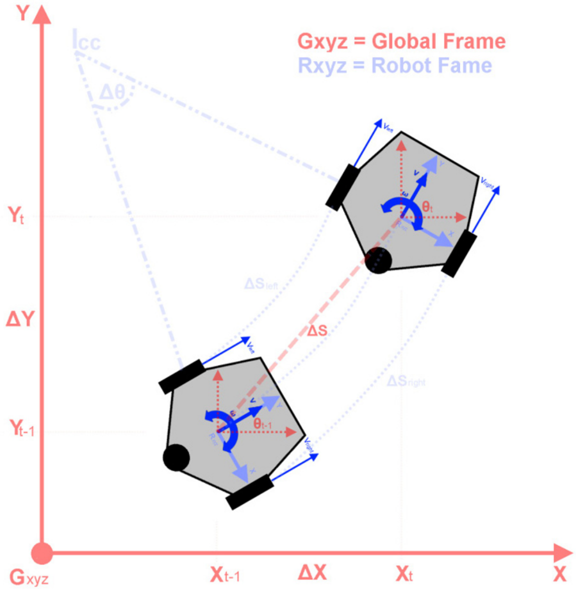

Fig. 1. Kinematic Model of the TOM3D

The kinematics of the TOM3D in navigation frame is depicted in the figure 1, which is used to calculate the pose error model. The error state model (Jian, 2008) of the mobile robot to estimate robot's pose compose of motion and measurement model. The measurement and observation model are derived using perturbation in encoder, gyroscope and accelerometer measurements and parameters. Each sensors error model is briefly described as follows:

Encoder Velocity Error Model: The real and ideal velocity equations for robot are as follows:

$$V_{idl}(t+1) = \left(\frac{V_L(t) + V_R(t)}{2}\right) \tag{1}$$

$$\omega_{idl}(t+1) = \left(\frac{V_L(t) - V_R(t)}{L}\right) \tag{2}$$

$$Vx_{idl}(t+1) = V_{idl}(t) \cdot \cos\left(\omega_{idl}(t) \cdot \Delta T\right) \tag{3}$$

$$Vx_{idl}(t+1) = V_{idl}(t) \cdot \sin\left(\omega_{idl}(t) \cdot \Delta T\right) \tag{4}$$

$$V_{act}(t+1) = \left(\frac{V_L(t) + V_R(t) + S_L(t)V_L(t) + S_R(t)V_R(t)}{2}\right) \tag{5}$$

$$\omega_{act}(t+1) = \left(\frac{V_L(t) - V_R(t) + S_L(t)V_L(t) - S_R(t)V_R(t)}{L + S_D(t)L}\right) \tag{6}$$

$$Vx_{act}(t+1) = V_{act}(t) \cdot \cos\left(\left(\omega_{act}(t) + \Delta\omega_{act}(t)\right) \cdot \Delta T\right) \tag{7}$$

$$Vy_{act}(t+1) = V_{act}(t) \cdot \sin\left(\left(\omega_{act}(t) + \Delta\omega_{act}(t)\right) \cdot \Delta T\right) \tag{8}$$

The error models of encoder velocities after simplification are:

$$\Delta Vx_e(t+1) = Vx_{act}(t+1) - Vx_{idl}(t+1) \tag{9}$$

$$\Delta Vy_e(t+1) = Vy_{act}(t+1) - Vy_{idl}(t+1) \tag{10}$$

$$\begin{aligned}\Delta Vx_e(t+1) = {} & \left(\frac{S_L(t)V_L(t) + S_R(t)V_R(t)}{2}\right) \cdot \cos(\Delta\omega_e(t) \cdot \Delta T) \\ & - \left(\frac{V_L(t) + V_R(t)}{2}\right) \cdot \Delta\omega_e(t) \cdot \Delta T \cdot \sin(\Delta\omega_e(t) \cdot \Delta T)\end{aligned} \tag{11}$$

$$\begin{aligned}\Delta Vy_e(t+1) = {} & \left(\frac{V_L(t) + V_R(t)}{2}\right) \cdot \cos(\Delta\omega_e(t) \cdot \Delta T) \\ & + \left(\frac{S_L(t)V_L(t) + S_R(t)V_R(t)}{2}\right) \cdot \Delta\omega_e(t) \cdot \Delta T \cdot \sin(\Delta\omega_e(t) \cdot \Delta T)\end{aligned} \tag{12}$$

$$\Delta\omega_e(t+1) = \left(\frac{V_L(t) - V_R(t)}{L}\right) \cdot S_D(t) - \left(\frac{S_L(t)V_L(t) - S_R(t)V_R(t)}{L}\right) \tag{13}$$

The velocity scale factor errors (S_L, S_R) and wheel distance error (S_D) are assumed to very slow time invariant, therefore

$$S_L(t+1) = S_L(t) \tag{14}$$

$$S_R(t+1) = S_R(t) \tag{15}$$

$$S_D(t+1) = S_D(t) \tag{16}$$

Accelerometer Velocity Error Model: The accelerometer error model for real and ideal linear velocities along with the scale (S_{ax}, S_{ay}) and bias factors (B_{ax}, B_{ay}) are as follows:

$$Vx_{idl}(t+1) = Vx_{idl}(t) + S_{ax}(t) \cdot Ax(t) \cdot \Delta T + B_{ax}(t) \tag{17}$$

$$Vy_{idl}(t+1) = Vy_{idl}(t) + S_{ay}(t) \cdot Ay(t) \cdot \Delta T + B_{ay}(t) \tag{18}$$

$$\begin{aligned}Vx_{act}(t+1) = {} & Vx_{act}(t) + \left(S_{ax}(t) + \Delta S_{ax}(t)\right) \cdot Ax(t) \cdot \Delta T \\ & + \left(B_{ax}(t) + \Delta B_{ax}(t)\right)\end{aligned} \tag{19}$$

$$\begin{aligned}Vy_{act}(t+1) = {} & Vy_{act}(t) + \left(S_{ay}(t) + \Delta S_{ay}(t)\right) \cdot Ay(t) \cdot \Delta T \\ & + \left(B_{ay}(t) + \Delta B_{ay}(t)\right)\end{aligned} \tag{20}$$

$$\Delta Vx_a(t+1) = Vx_{act}(t+1) - Vx_{idl}(t+1) \tag{21}$$

$$\Delta Vy_a(t+1) = Vy_{act}(t+1) - Vy_{idl}(t+1) \tag{22}$$

$$\Delta Vx_a(t+1) = \Delta Vx_a + \Delta S_{ax}(t) \cdot Ax(t) \cdot \Delta T + \Delta B_{ax}(t) \tag{23}$$

$$\Delta Vy_a(t+1) = \Delta Vy_a + \Delta S_{ay}(t) \cdot Ay(t) \cdot \Delta T + \Delta B_{ay}(t) \tag{24}$$

$$S_{ax}(t+1) = S_{ax}(t) \tag{25}$$

$$B_{ax}(t+1) = B_{ax}(t) \tag{26}$$

$$S_{ay}(t+1) = S_{ay}(t) \tag{27}$$

$$B_{ay}(t+1) = B_{ay}(t) \tag{28}$$

Gyroscope Error Model: The gyroscope error model for real and ideal angular velocities along with the scale (S_{gz}) and bias factor (B_{gz}) are as follows:

$$\omega_{idl}(t+1) = S_{gz}(t) \cdot \Omega(t) + B_{gz}(t) \tag{29}$$

$$\omega_{act}(t+1) = \left(S_{gz}(t) + \Delta S_{gz}(t)\right) \cdot \Omega(t) + \left(B_{gz}(t) + \Delta B_{gz}(t)\right) \tag{30}$$

$$\Delta\omega_g(t+1) = \omega_{act}(t+1) - \omega_{idl}(t+1) \tag{31}$$

$$\Delta\omega_g(t+1) = \Delta S_{gz}(t) \cdot \Omega(t) + \Delta B_{gz}(t) \tag{32}$$

$$S_{gz}(t+1) = S_{gz}(t) \tag{33}$$

$$B_{gz}(t+1) = B_{gz}(t) \tag{34}$$

Compass Angle Error Model: The compass error model for actual and ideal azimuth angle along with the bias factor (B_c) are as follows:

$$\theta_{idl}(t+1) = \theta_{idl}(t) + B_c(t) \tag{35}$$

$$\theta_{act}(t+1) = \theta_{act}(t) + \left(B_c(t) + \Delta B_c(t)\right) \tag{36}$$

$$\Delta\theta_c(t+1) = \theta_{act}(t+1) - \theta_{idl}(t+1) \tag{37}$$

$$\Delta\theta_c(t+1) = \Delta\theta_c(t) + \Delta B_c(t) \tag{38}$$

$$B_c(t+1) = B_c(t) \tag{39}$$

The final measurement and observation model equations used for data fusion in state space model form are as follows:

$$\Delta X(t+1) = A(t)\cdot\Delta X(t) + w(t) \tag{40}$$

$$\Delta Y(t) = C(t, V_L, V_R, \omega_e, Ax, Ay, \Omega)\cdot\Delta X(t) + v(t) \tag{41}$$

Where

$A(t) = [A_1 \quad A_2 \quad A_3]$

$C(t, V_L, V_R, \omega_e, Ax, Ay, \Omega) = [C_1 \quad C_2 \quad C_3]$

$Ax(t), Ay(t) =$ Accelerometer Readings

$\Omega(t) =$ Gyroscope Readings

$V_L(t), V_R(t) =$ Linear Velocity from Encoders

$\omega_e(t) =$ Angular Velocity from Encoders

$\Delta T =$ Samping Time

$$A_1 = \begin{bmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&1&0\\ 0&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix} \quad \Delta X = \begin{bmatrix} \Delta Vx_e(t)\\ \Delta Vy_e(t)\\ \Delta Vx_a(t)\\ \Delta Vy_a(t)\\ \Delta\omega_e(t)\\ \Delta\omega_g(t)\\ \Delta\theta_c(t)\\ \Delta S_{LW}(t)\\ \Delta S_{RW}(t)\\ \Delta L_{DW}(t)\\ \Delta S_{AX}(t)\\ \Delta B_{AX}(t)\\ \Delta S_{AY}(t)\\ \Delta B_{AY}(t)\\ \Delta S_{GZ}(t)\\ \Delta B_{GZ}(t)\\ \Delta B_C(t) \end{bmatrix} \quad \Delta Y = \begin{bmatrix} \Delta Vx_e(t) - \Delta Vx_a(t)\\ \Delta Vy_e(t) - \Delta Vy_a(t)\\ \Delta\omega_e(t) - \Delta\omega_g(t) \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \frac{-(V_L(t)+V_R(t))\Delta T\sin(\omega_e(t)\Delta T)}{2} & 0 & 0 & \frac{V_L(t)\cos(\omega_e(t)\Delta T)}{2} & \frac{V_R(t)\cos(\omega_e(t)\Delta T)}{2}\\ \frac{(V_L(t)+V_R(t))\Delta T\cos(\omega_e(t)\Delta T)}{2} & 0 & 0 & \frac{V_L(t)\sin(\omega_e(t)\Delta T)}{2} & \frac{V_R(t)\sin(\omega_e(t)\Delta T)}{2}\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&-\frac{V_L(t)}{L}&\frac{V_R(t)}{L}\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&A_x(t)\Delta T&1&0&0&0&0&0\\ 0&0&0&A_y(t)\Delta T&1&0&0&0\\ \frac{V_L(t)-V_R(t)}{L}&0&0&0&0&0&0&0\\ 0&0&0&0&0&\Omega(t)\Delta T&1&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0&0&-1&0&\frac{-(V_L(t)+V_R(t))\Delta T\sin(\omega_e(t)\Delta T)}{2}\\ 0&0&-1&0&\frac{(V_L(t)+V_R(t))\Delta T\cos(\omega_e(t)\Delta T)}{2}\\ 0&0&0&0&0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0&0&\frac{V_L(t)\cos(\omega_e(t)\Delta T)}{2}&\frac{V_R(t)\cos(\omega_e(t)\Delta T)}{2}\\ 0&0&\frac{V_L(t)\sin(\omega_e(t)\Delta T)}{2}&\frac{V_R(t)\sin(\omega_e(t)\Delta T)}{2}\\ 0&0&-\frac{V_R(t)}{L}&\frac{V_R(t)}{L} \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 0&-Ax(t)\Delta T&-1&0&0&0&0&0\\ 0&-Ay(t)\Delta T&-1&0&0&0&0&0\\ \frac{V_L(t)-V_R(t)}{L}&0&0&0&0&-\Omega(t)\Delta T&-1&0 \end{bmatrix}$$

3. ROBOT STRUCTURE

The TOM3D is a two wheel differential drive robot. Each wheel is equipped with a DC-motor integrated with gearbox and quadrature encoder. An electronic control board is designed for this robot based on 16-bit Infineon MCU. The robot's control board is equipped with various sensors for autonomous map building purpose. The robot's firmware is designed in a way that all the information are pre-processed at robot's control unit and reports are sent to the remote controlling PC. The PC transmits the high level user commands to the remote mobile robot with a wireless transceiver. The robot is also equipped with an embedded PC which is used to process the information acquired through PMD camera. The PMD camera acquires the real time 3D images of the environment. A software utility is running on the embedded PC to process the 3D image and get the distance to the obstacle from the robot. The distance information is sent to the remote controlling PC through the Wi-Fi communication channel built into the embedded PC. This distance information from PMD camera and ultrasonic sensors are used to validate the position estimated by the data fusion algorithm.

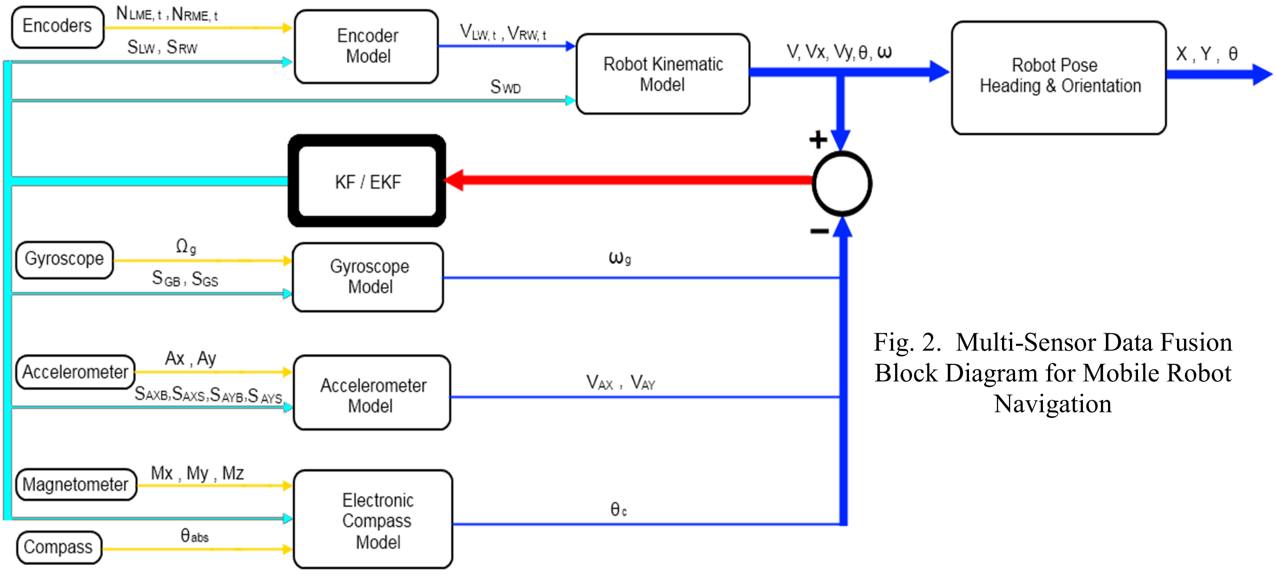

Fig. 2. Multi-Sensor Data Fusion Block Diagram for Mobile Robot Navigation

4. DATA FUSION

Multisensor data fusion (Hall, 1997) method reduces deterministic and stochastic errors during mobile robot operation hence provides the accurate pose information of mobile robot without the use of external positioning system for longer period of time. A robust data fusion algorithm must address the problems such as different sensors sampling rates, asynchronous sensors sampling and reliable availability of estimated data in the presence of sensor failures. Kalman filter (Simon, 2006) is widely used for pose estimation and data fusion. Since the scale factors are not measured by sensors therefore they need to be estimated through the model. The state estimation by Kalman filter is elaborated by the fig. 3. The Kalman algorithm consists of two phases; Time update and Measurement update. In each phase the mean and covariance of the system is updated and propagated. Therefore the motion and sensor measurement model of the mobile robot is required to estimate the mean value for Time and Measurement update phase respectively.

Time Update:

$$X_t = A_{t-1}X_{t-1} + B_{t-1}U_{t-1} + G_{t-1}w_{t-1} \quad (42)$$

$$\tilde{P}_t = A_{t-1}P_{t-1}A_{t-1}^T + G_{t-1}Q_{t-1}G_{t-1}^T \quad (43)$$

Measurement Update:

$$\tilde{Y}_t = C_t\tilde{X}_t \quad (44)$$

$$r_t = Y_t - \tilde{Y}_t \quad (45)$$

$$S_t = C_t\,\tilde{P}_t\,C_t^T + R_t \quad (46)$$

$$K_t = \tilde{P}_t\,C_t\,S_t^{-1} \quad (47)$$

$$X_t = \tilde{X}_t + K_t\,r_t \quad (48)$$

$$P_t = \tilde{P}_t - \tilde{P}_t\,C_t^T\,S_t^{-1}\,C_t\,\tilde{P}_t \quad (49)$$

Kalman Filter can be applied for the multisensor data fusion (Sasiadek, 2000) directly over the state vector or indirectly over the error in state vector. Therefore following Kalman filter data fusion schemes are possible: (1) Direct Pre-Filter, (2) Direct Filter, (3) Indirect Feed Backward Filter, (4) Indirect Feed Forward Filter.

In direct pre-filter scheme dead reckoning measurement and inertial navigation system measurements are filtered separately and the errors between these filtered measurements are used to correct measurement from any one method as shown in fig. 4. This is the basic idea of the Kalman filter to correct the estimate of the motion model by the measurements as shown in fig.3. Each Kalman filter requires separated measurement model and sensor model for estimation.

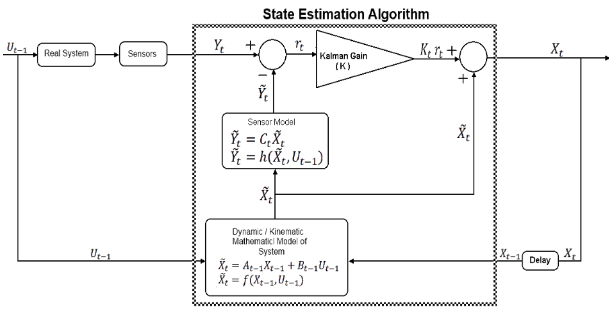

Fig. 3. Block Diagram of State Estimation Using Kalman Filter

The direct filter formulation uses the states such as position/velocity and orientation calculated from wheel encoders as state variables and the measurements are the inertial and other sensors outputs, fig. 5. In direct formulation Kalman filter is inside the navigation loop therefore filter has to suppress the noisy measurements from the INS as well as to estimate of mobile robots position/velocity and orientation. Due to accurate kinematics estimation and being inside the navigation loop the filter has to be update faster than the dynamics of the navigation system. This of course is a computational burden because Kalman filter gain calculation requires inverse and square operations on matrix which are very costly in term of computation. Another disadvantage is since encoder and gyroscope/accelerometer models are independent from each other therefore it suppress the errors exclusively according to respective sensor model. In general indirect Kalman filter formulation is divided in two parts, Navigation Equation and Navigation error equations.

Navigation equations are used to estimate the pose where Navigation error equations are used to compensate the pose errors.

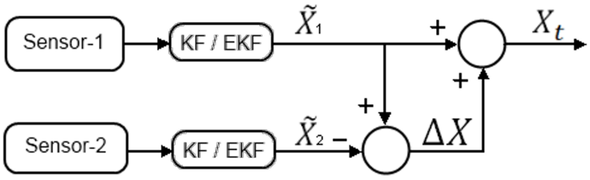

Fig. 4. Direct Pre-Kalman Filter

The navigation error equations are derived by linear perturbation in the dead reckoning and inertial navigation system model.

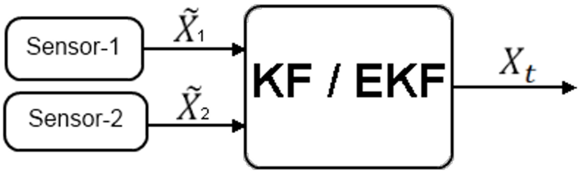

Fig. 5. Direct Kalman Filter

The indirect feedback Kalman filter feeds back the error estimates to one of the mobile robot's dead reckoning or inertial navigation algorithm to mutually compensate the errors as shown in fig. 6. The filter estimates the systematic errors of encoder (wheel scale factor, wheel distances) and stochastic errors of gyroscope (scale factor, bias) mutually and explicitly. These scale factor errors are fed back to compensate the respective sensor output. Furthermore, the pose errors are feedback into navigation system.

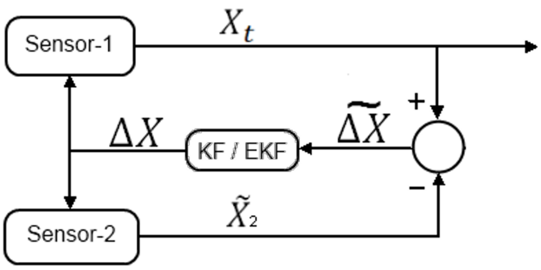

Fig. 6. Indirect Feed Backward Kalman Filter

In indirect feed forward formulation the signals measured from sensors are compared before fed into the Kalman Filter and the estimated error is added into one of the dead reckoning system or inertial navigation system as shown in fig. 7.

The proposed multisensor data fusion algorithm is elaborated by a block diagram is shown in the fig.2 it is based on indirect feedback Kalman filter data fusion methodology.

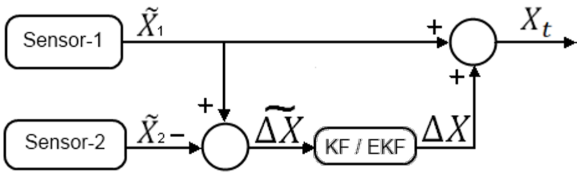

Fig. 7. Indirect Feed Forward Kalman Filter

5. EXPERIMENTAL RESULTS

During the Experiment TOM3D is commanded to move along a straight line at a speed of 28 cm/sec for 25 sec, for a linear trajectory of 700cm. Figure 8 shows the experiment environment in which the robot has moved along a wall. The straight wall provided a reference for PMD camera distance measurement. An application to acquire data from PMD camera was running in the embedded P|C. Data is acquired from robot controller and embedded pc through the wireless transceiver and Wi-Fi adapter respectively. The acquired sensors data is processed off-line.The effect of the nearby electronic sources and ferromagnetic materials can be seen clearly in fig. 9. The data shows the variations of the magnetic field intensity measured by magnetometer during the travel near ferromagnetic materials. The implemented algorithm monitors the earth magnetic field strength measured by magnetometer as a criterion to accept the compass measurements which is used to correct the robot's orientation. A set of waypoints trajectory is sent to robot by wireless transceiver. During the execution of the linear trajectory along the wall, the robot acquires the earth magnetic field strength at the start point of the linear trajectory and then uses it as a criterion for acceptance of compass measurements if the field strength varies less than the threshold value calculated from the start point value. The gyroscope drift is also prominent clearly in fig. 9.

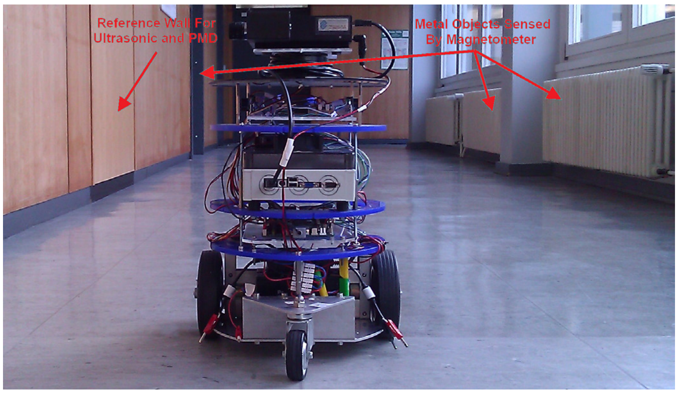

Fig. 8. Experiment Environment with ferromagnetic interferences and reference wall for measurements

Fig. 10 shows the variations of the robots angular velocity calculated from wheel encoders and gyroscope. Fig. 11 shows the trajectory of the TOM3D by wheel encoders, fusion algorithm and PMD camera. At the end of experiment a manual measurements of final robot position were taken which reported the final robot position is 4cm (Y-axis, toward wall) and 15cm (X-axis, along corridor) away from the desired end position.

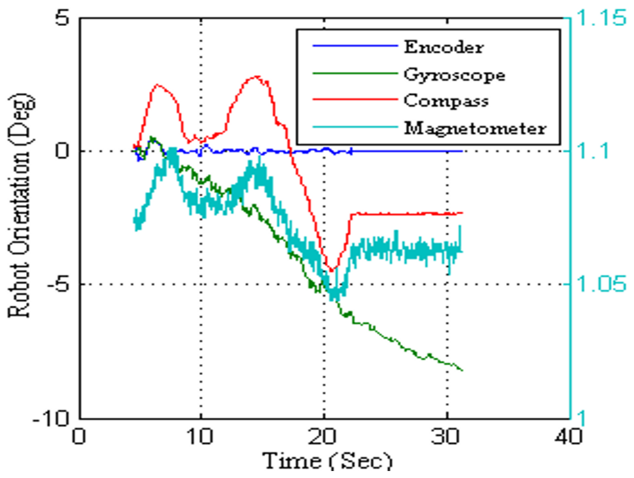

Fig. 9. Angle measured by wheel encoders, gyroscope and electronic compass. Magnetic field strength variation along the trajectory.

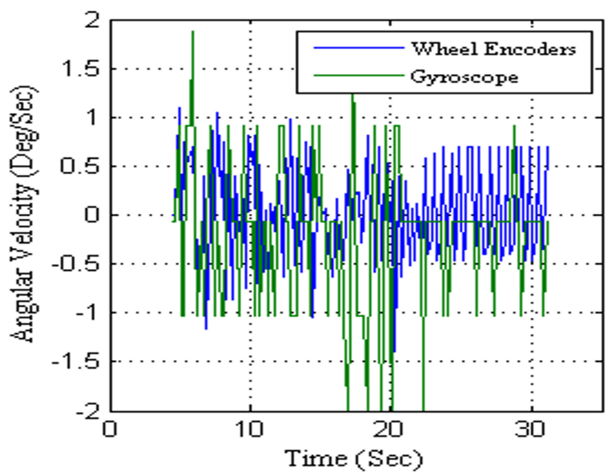

Fig. 10. Angular Velocity measurement from wheels encoder and Gyroscope.

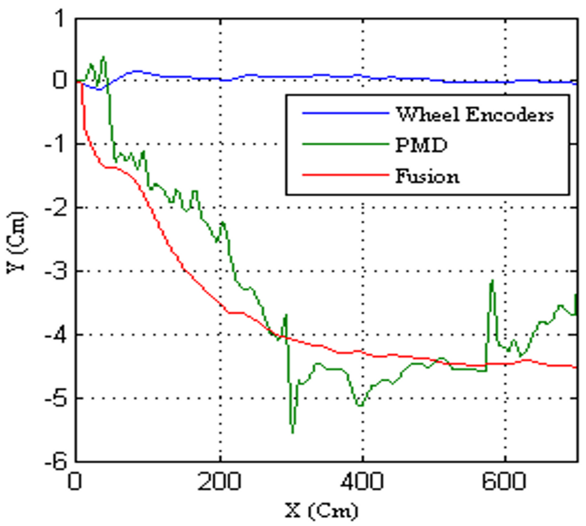

Fig. 11. Trajectory estimation by wheels encoders , PMD Camera and Fusion algorithm.

6. CONCLUSION

In this research work a detailed error state model for dead reckoning and inertial navigation system is derived using error perturbation theory and implemented for a indoor mobile robot. An indirect feedback based Kalman filter data fusion methodology is applied to estimate the robot pose. Various multisensor Kalman filter data fusion configuration are also discussed. The results are validated against the results obtained from the PMD camera. In future the filter will be tested in scenarios where robot is given waypoint commands for map building purpose. Furthermore the Kalman filter will be implemented in real-time within the robot control unit.

REFERENCES

Borenstein J. and Feng L. (1996). Measurement and correction of Systematic Odometry Errors in Mobile Robots. *IEEE Trans. On Robotic and Automation,* Vol.12, No.6, 869-880.

Billur Barshan, Huge F. Durrant-Whyte (1995). Inertial Navigation Systems for Mobile Robots. *IEEE Transactions on Robotics and Automation*, Vol. (11), 328-342.

Borenstein, J., Feng, L., (1996). Gyrodometry: A New Method for Combining Data from Gyros and Odometry in Mobile Robots. *IEEE International Conference on Robotics and Automation, 1996*, Vol. (1), 423-428.

Hakyoung Chung Ojeda, L. Borenstein, J. (2001). Sensor fusion for mobile robot dead-reckoning with a precision-calibrated fiber optic gyroscope, *International Conference on Robotics and Automation*, Vol. (4), 3588-3593.

K. Komoriya and E. Oyama. (1994). Position Estimation of a Mobile Robot Using Optical Fiber Gyroscope OFG. *International Conference on Inelligent Robots and Systems '94'*, Vol. (1), 143-149

Hoshino, M. Gunji, Y. Oho, S. Takano, K.(1996.) A Kalman filter to estimate direction for automotive navigation. *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 145-150.

Hua Sun Peike Yang (2009). Design and research on errors compensation of digital compass. *International Conference on Mechatronics and Automation*, 4742-4747.

Zhangjun Song Huifen Liu Jianwei Zhang Liwei Wang Ying Hu (2010). A localization algorithm for low-cost cleaning robots based on kalman filter. *Intelligent Control and Automation (WCICA)*, 1450-1455

Qingxin Zhang Xiaoyan Qiu Guangyan Xu Qinglong Di (2010). Azimuth estimation of mobile robot based on Federated Kalman Filter. *International Conference on Computer, Mechatronics, Control and Electronic Engineering*, Vol.4, 202-205.

Von der Hardt, H.-J., Wolf, D., Husson, R. (1996). The dead reckoning localization system of the wheeled mobile robot ROMANE. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 603-610.

Xiang, Z. Ozguner, U. (2005). A 3D Positioning System for Off-road Autonomous Vehicles. *IEEE* Intelligent *Vehicles Symposium Proceedings*, 130-135.

A. Prusak, O. Melnychuk and H. Roth (2008). Pose estimation and map building with a Time-Of-Flight-camera for robot navigation. *Int. J. Intelligent Systems Technolgies and Applications,* Vol.5, Nos. ¾.

Jian Rong, Haitao Zhang, Xiaochun Zhong, Ke Chen (2008). Research and Application of a Kalman Smoother Based on INS/GPS Integrated Navigation. *International Colloquium on Computing, Communication, Control, and Management*, 672-676.

David L. Hall, James Llinas (1997). An Introduction to Multisensor Data Fusion. Proceedings *of the IEEE*, Vol. (85) Issue No. 1, 6-23.

Simon, D. (2006). *Optimal State Estimation: Kalman, H∞, and Nonlinear* Approaches. Wiley, Hoboken, New Jersey.

J.Z. Sasiadek and P. Hartana, Sensor Data Fusion Using Kalman Filter (2000). Proceedings *of the Third International Conference on Information Fusion.* Vol. (2), 19-25.