# A Correction System of Odometry Error for Map Building of Mobile Robot Based on Sensor fusion

Woong Keun Hyun, *Member, KIMICS*

***Abstract*—** **This paper represents a map building and localization system for mobile robot. Map building and navigation is a complex problem because map integrity cannot be sustained by odometry alone due to errors introduced by wheel slippage, distortion and simple linealized odometry equation. For accurate localization, we propose sensor fusion system using encoder sensor and indoor GPS module as relative sensor and absolute sensor, respectively. To build a map, we developed a sensor based navigation algorithm and grid based map building algorithm based on Embedded Linux O.S. A wall following decision engine like an expert system was proposed for map building navigation. We proved this system's validity through field test.**

***Index Terms*—map building, localization, sensor fusion, odometry, navigation**

## I. INTRODUCTION

The goal of this work is to implement an autonomous mobile robot capable of navigation in an unknown indoor environment using position sensors. For this , the robot requires the capability to build a map of the environment, which is a cyclic process of moving to a new position, sensing the environment, updating the map and planning subsequent motion. Map building and navigation is a complex problem because map integrity cannot be sustained by odometry alone due to errors introduced by wheel slippage, distortion and simple linealized odometry equation. Any sensing is subject to random errors. Hence, neither odometry nor absolute sensory data to the map gives perfect estimation of the robot's position. This paper employs absolute and relative positioning for accurate position estimation in the map building process. Relative positioning is usually based on odometry, that is, computing a vehicle's relative motion from the measurement of wheel revolution and/or steering angle[1][9]. In most mobile robots, odometry is implemented by means of optical encoders that monitor the wheel revolutions and/or steering angle of the robot's wheels[5]. The encoder data is then used to compute the vehicle's offset from a known starting position. A positioning by odometry is simple and fast method to estimate the position and orientation of the moving robot. However, there are some systematic and non-systematic errors in detecting the correct position. Moreover, the errors continuously accumulate. Therefore, the odometry needs to be correct from time to time[2]. The disadvantage of odometry is its unbounded accumulation of errors. Because of the accumulation of errors. Absolute position corrections and often necessary after as little as 10 m of linear, and they are usually based on external measurements such as indoor GPS or landmarks[4][5]. However, the indoor GPS also has some drawback such as the error range of the indoor GPS is too wide to be used in a small size robot.

This paper presents an accurate localization method by encoder of relative sensor and absolute sensor such as indoor GPS sensor fusion for map building system of mobile robot. The simple GPS system described in this paper consists of 4 transmitter and 2 receivers which have RF and ultrasonic devices respectively. Indoor GPS systems use transmitter located at high elevation as actual GPS or DGPS, or on the ground which irradiates RF or IR as synchronizing signal providing the receivers at opposite location with distance data. Such systems calculate locations with the distance data measured with 3 or more receivers [3].

The paper is organized as follows: Section 2 describes properties of odometry errors. Section 3 presents a sensor fusion system using indoor GPS and odometry. Section 4 shows the Linux based navigation system for map building using proposed sensor fusion system and section 5 provides conclusions and suggests future research to improve this method.

## II. PROPERTIES OF ODOMETRY ERRORS

The differential-drive design of two wheel mobile robot, incremental encoders are mounted onto the two drive motors to count the wheel revolution. Using simple geometric equations, it is straight-forward to compute the momentary position of the vehicle relative to the known starting position. The basic odometry equations by using incremental encoder data are given as Eq.(1), (2) and (3).

Manuscript received October 4, 2010; revised October 28, 2010; accepted November 6, 2010.

woong keun Hyun is with the Department of Electronics, Honam University, Gwang-ju, 506-714, Korea (Email: wkhyun@honam.ac.kr)

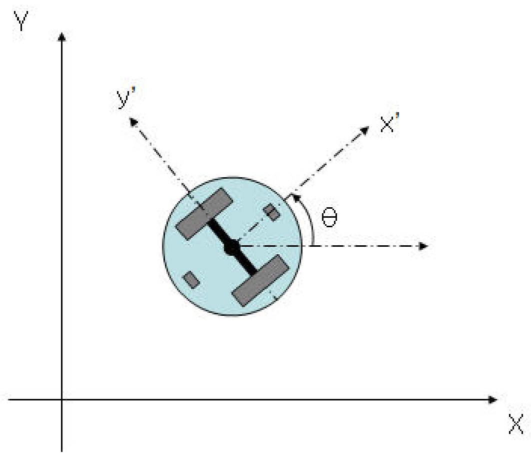



Fig.1. A differential drive mobile robot.

Thus, there is uncertainty about the angle at which the obstacle was measured which is about as large as the opening angle of the sonar.

$$\Delta\theta_0 = \frac{2\pi R}{N}\frac{\Delta enc_R(t) - \Delta enc_L(t)}{D} \quad (1)$$

$$\Delta X = \frac{2\pi R}{N}\frac{\Delta enc_R(t) - \Delta enc_L(t)}{2} \cos\left(\theta_0 + \frac{\Delta\theta_0}{2}\right) \quad (2)$$

$$\Delta Y = \frac{2\pi R}{N}\frac{\Delta enc_R(t) - \Delta enc_L(t)}{2} \sin\left(\theta_0 + \frac{\Delta\theta_0}{2}\right) \quad (3)$$

where R, N and D are the radius of wheel, gear rate and distance between wheels, respectively. $\Delta enc_L$ and $\Delta enc_R$ are incremental encoder values of right and left wheel, respectively.[5][6]. When investigating odometry errors, one should realize that there are two substantially different categories: (1) systematic and (2) non-systematic error sources. Systematic errors are caused by unequal wheel diameter, simple linealization of the odometry equations as Eq(1) thru (3). Non-systematic errors are caused by slippery floors. Systematic errors are particularly grave because they accumulate constantly. On most normal indoor surfaces systematic error is larger than non-systematic error. However, on rough surfaces with significant irregularities, non-systematic errors may be dominant.

## III. SENSOR FUSION SYSTEM USING INDOOR GPS AND ODOMETRY

A localization by odometry is simple and fast method to estimate the position and orientation of the moving robot. However, there are some systematic and non-systematic errors in detecting the correct position. Moreover, the errors continuously accumulate[5]. Therefore, the odometry needs to be correct from time to time. We propose a sensor fusion method to estimate the robot position using indoor GPS and encoder sensor.

The indoor GPS system consists of 4 transmiter and 1 receiver which have RF and ultrasonic devices respectively. The indoor GPS can detect the position of the robot in the workspace surrounded by 4 I-GPS transmitter poles. It has some drawback that its average position error is much larger than that of odometry , and the absolute value of estimated error is changed with respective to the position of I-GPS receiver in the workspace. The position error sensed from indoor GPS is bounded as much as 12cm, its error is larger than that of encoder sensor when a mobile robot follows the straight line path. On the other hand, when the robot moves in corner or circular path, The sensed error from encoder is larger than that from indoor GPS. To overcome the short coming of the two sensors and increase the advantages, we propose a sensor fusion method by using absolute and relative sensors.

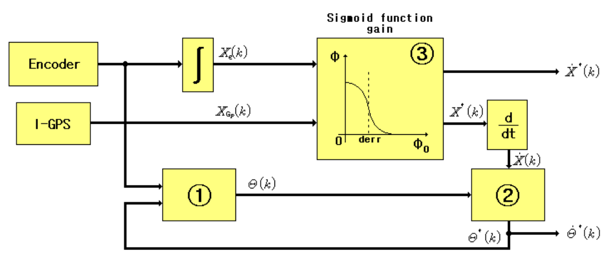



Fig 2. The block diagram of the proposed sensor fusion system

The Fig 2 represents the block diagram of the proposed sensor fusion method using encoder sensor and indoor GPS. The encoder sensor and indoor GPS detect the relative position and absolute position of the mobile robot, respectively . The two sensor data are fused in ③ of Fig 2 by sigmoid function filter. $\dot{X}(k)$ is a estimated position velocity after sensor fusion. The orientation of the robot can be found in ② by Eq(7). And the final estimation of the robot orientation can be found in ① by Eq(8). Used an equation (4), (5), (6) by sensor harmony at these papers in order to reduce a localization error.

$$X(k) = (1+\Phi)\cdot X_{GP}(k) + \Phi \cdot X_e(k) \quad (4)$$

$$\phi = 1 - \frac{1}{1+e^{-(perr-derr)\bullet k}} \quad (5)$$

$$\text{where} \quad perr = \sqrt{(X_E - X_{GP})^2 + (Y_E - Y_{GP})^2} \quad (6)$$

$$\dot{\theta}(k) = \cos+\left(\frac{\dot{x}}{\sqrt{\dot{x}^2+\dot{y}^2}}\right) + \sin+\left(\frac{\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}\right) - \theta(k) \quad (7)$$

$$\dot{\theta}(k) = \begin{cases} \Delta\theta_0, where\,|\,\dot{\theta}(k) - \Delta\theta_0\,| < \delta\theta \\ \dot{\theta}(k), where\,|\,\dot{\theta}(k) - \Delta\theta_0\,| < \delta\theta \end{cases} \quad (8)$$

Eq(4) thru Eq(8) represent the sensor fusion equations of encoder and indoor GPS sensors. Where $X_{GP}$ and $X_E$ are measured position from the indoor GPS and encoder sensor, respectively. Eq(4), Eq(5) and Eq(6) work in ③ of Fig 2. Eq(7) and Eq(8) work in ② and ① of Fig 2 respectively. Eq(5) shows weights of Eq(4) as a type of sigmoid function. Where *perr* is a distance between positions from encoder and indoor GPS. The sensor fusion system described by Eq(4),(5) and (6) is designed by following idea.

(1) if the distance error *perr* is larger than the limit value of indoor GPS(*derr*), then the reliable value is that from indoor GPS.
(2) If the *perr* is smaller than *derr*, then the appropriate localization value is mixed one of indoor GPS and encoder.
(3) The smaller *perr* is, the more reliable encoder position localization is. The larger *perr* is, the more reliable indoor GPS localization is.

## IV. SOFTWARE STRUCTURE AND WALL FOLLOWING ENGINE FOR MAP BUILDING

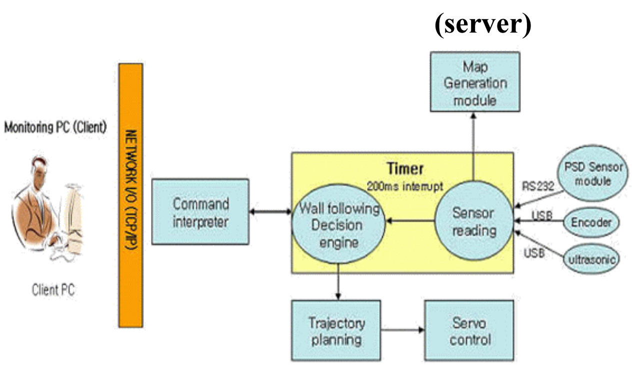



Fig. 3. The overall framework of the robot system

The robot controller is run on common Embedded Linux. Fig 3 illustrates the overall framework of the proposed navigation scheme based on Embedded Linux. The navigation software structure consists of 6 processes such as command interpreter, map generation process, wall following decision engine, sensor reading process and trajectory planning process. All process works based on time scheduler by timer interrupter of Embedded Linux O.S. At every 200msec interrupted by timer, sensor read process runs at first time and then this process gives position and distance data to the wall following decision engine. Wall following decision engine determines the moving direction and velocity of the robot by using these information. These commands are sent to trajectory planning process. The robot detects the obstacle at forward direction and decides the moving direction and velocity by proposed wall following decision engine. The wall following decision engine is represented as fallows;

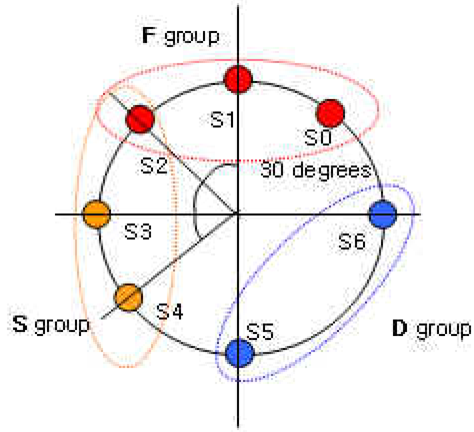



Fig. 4 Sensor allocation for the navigation robot

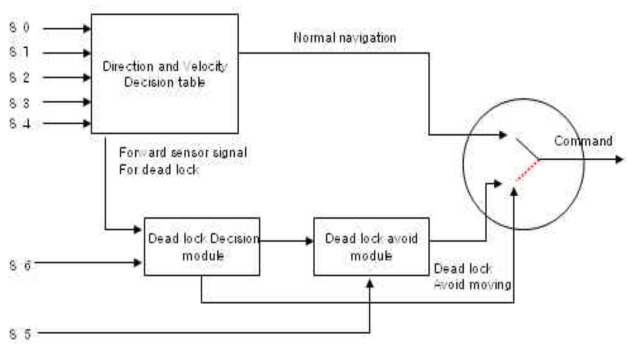



Fig. 5 Block diagram of wall following decision engine

Figure 4 shows the sensor allocation for the wall following. There are 7 sensors where 3 sensors in group F are for detecting the forward direction and sensors in group S are for detecting of the distance of the side wall and sensors in group D are for dead locking avoidance, respectively. The direction and velocity command of the robot for wall following are determined by using sensors in group F and S. When the robot meets a dead lock situation, 2 sensors in group D are used for finding the escape direction. The sensors in group F and group S recognize the shape of the wall to be followed. The decision to control the mobile robot is determined in 3 cases as follows;

(i) when S2, S3, S4 sensors of S group detect plate wall and S0 and S1 detect safe distance.
(ii) when S2, S3, S4 sensors of S group detect corner and S0 and S1 detect safe distance.
(iii) when S0, S1, S2 sensors of F group detect obstacle

(i) when S2, S3, S4 sensors of S group detect plate wall and S0 and S1 detect safe distance.

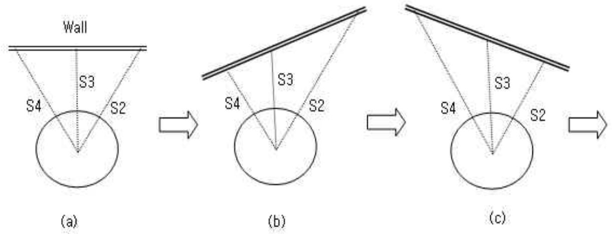



Fig.6 The case when sensors in S group detect the plate wall

In case of (a) in Fig.6, sensors S2, S3 and S4 measure the distance as M, S and M, respectively. The moving direction of robot is perpendicular to the wall. In case (b) and (c), sensors S2, S3 and S4 measure L, M and S or S, M and L, respectively. In case (a) the robot move forward and in case (b) and (c) the robot move forward and turn left and move forward and turn right slightly, respectively.

(ii) when S2, S3, S4 sensors detect corner and S0 and S1 detect safe distance.

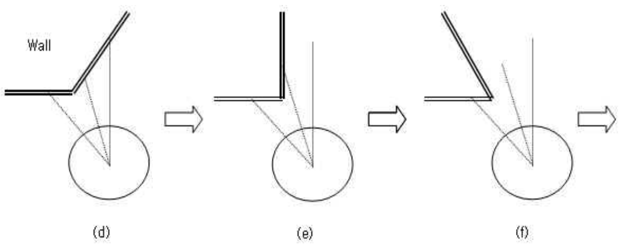



Fig.7 The case when robot sensors a corner wall by sensor S2, S3 and S4.

In case of fig 7(d),(e) and (f), the distances from S2, S3 and S4 are M, S and M or L, M and M or L, L and M, respectively. Then the robot move turn left large, medium or small according to each case without moving forward.

(iii) when S0, S1, S2 sensors of F group detect obstacle

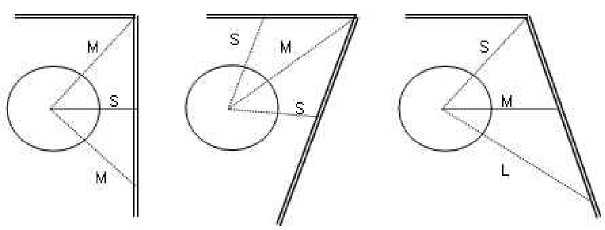



Fig.8 The robot sensors obstacle in forward moving direction by F group sensors

When the distances from S0, S1 and S2 are as Fig.8, the robot meets obstacles or wall on the forward area. Then the robot should turn right without any forward moving. The turning degree is determined as M, L and S according to the case (g), (h) and (i) in Fig 6, respectively. Where S, M and L mean Small, Medium and Large distance, respectively. From this idea, we can decide direction and velocity as in table I. In table 1, FRS means "move Forward Right direction with Small speed.

TABLE I. DECISION MAKING TABLE OF WALL FOLLOWING ENGINE.

| S0 | S1 | S2 | S3 | S4 | Command | S0 | S1 | S2 | S3 | S4 | Command |
|---|---|---|---|---|---|---|---|---|---|---|---|
| L/M | L/M | S | S | S | RS | L/M | L/M | M | L | S | RS |
| L/M | L/M | S | S | M | FLM | L/M | L/M | M | L | M | FRS |
| L/M | L/M | S | S | L | FLM | L/M | L/M | M | L | L | F |
| L/M | L/M | S | M | S | RM | L/M | L/M | L | S | S | RS |
| L/M | L/M | S | M | M | FRS | L/M | L/M | L | S | M | FRS |
| L/M | L/M | S | M | L | FLM | L/M | L/M | L | S | L | F |
| L/M | L/M | S | L | S | RS | L/M | L/M | L | M | S | RS |
| L/M | L/M | S | L | M | ELS | L/M | L/M | L | M | M | FRS |
| L/M | L/M | S | L | L | FLL | L/M | L/M | L | M | L | F |
| L/M | L/M | M | S | S | FRS | L/M | L/M | L | L | S | RS |
| L/M | L/M | M | S | M | F | L/M | L/M | L | L | M | FRS |
| L/M | L/M | M | S | L | F | L/M | L/M | L | L | L | F |
| L/M | L/M | M | M | S | FRS | S/M | S | S | S | - | B/RM |
| L/M | L/M | M | M | M | FRS | S/M | M | S | * | - | B/RM |
| L/M | L/M | M | M | L | F | S/M | M | M/L | * | - | FRS |

## V. EXPERIMENT RESULTS

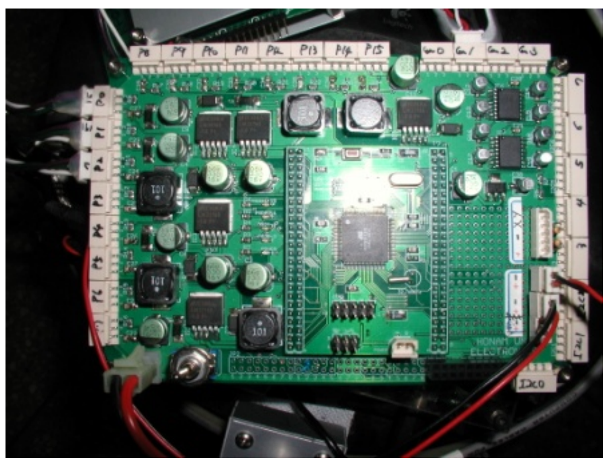


Fig 9 The developed sensor controller

The developed sensor controller for multiple sensor and mobile robot[6] are shown in Fig.9 and Fig.10, respectively. Localized trajectories by indoor GPS sensor and encoder data are shown in Fig 11 and Fig 12, respectively. The desired trajectory is rectangle path of 50cmX50cm. The trajectory drawn with bold line is desired one and the trajectory drawn with yellow line is resultant path estimated by odometry.

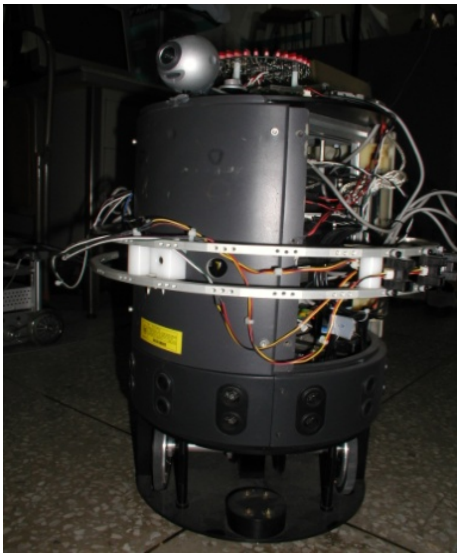


Fig. 10 A mobile robot with developed sensor system

Fig.11 shows a localized trajectory by encoder sensor and odometry equation Eq(1) thru Eq(3) after robot made a same square path as many as three times. There We can observer that the error is accumulating while robot follow a rectangle path and the error is larger when robot turn at a edge than move straight. If robot makes a curve or circular, or straight motion, the odometry localization by encoder sensor is superior to the other sensor such as I-GPS as in Fig.11, Fig.12 and Fig.13. The localized trajectory by indoor GPS sensor is shown in Fig.14. The trajectory has error like noise. The errors were distributed in whole trajectory, but the errors were not accumulated. The errors in line path of trajectory localized by encoder sensor were much smaller than that by indoor GPS sensor. To overcome the short coming of the two sensors and increase the advantages, we propose a sensor fusion method by Eq(4) thru Eq(8). Fig.15 and Fig.16 show the result of the proposed sensor fusion method. We can see that the localized trajectory by proposed method better than each single sensor's method.

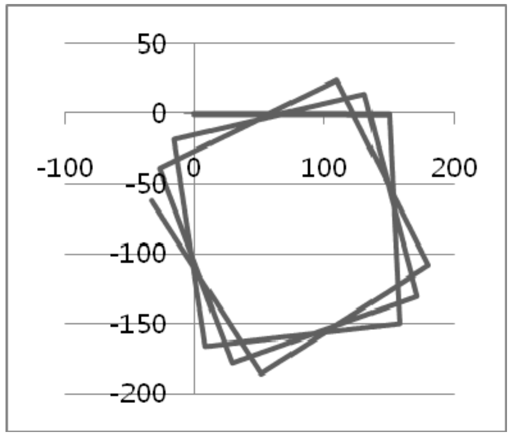



Fig.11 a square path experiment : a localized trajectory by odometry after robot made a same square path as many as three times.

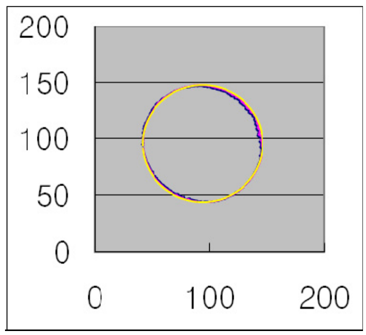



Fig.12 a localized trajectory after robot make one revolution.
(a) yellow line: trajectory by encoder sensor
(b) red line: trajectory by I-GPS sensor
(c) blue line: trajectory by sensor fusion

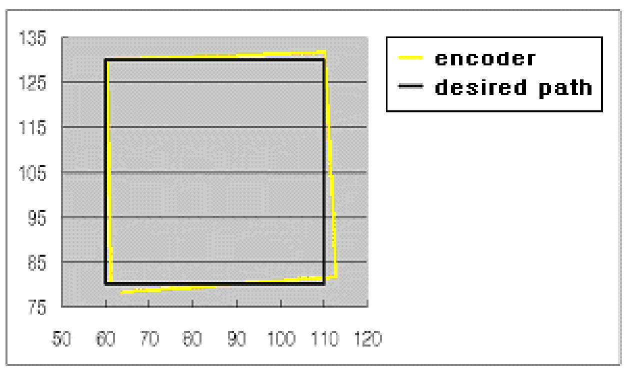



Fig 13 Localized trajectories by encoder sensor

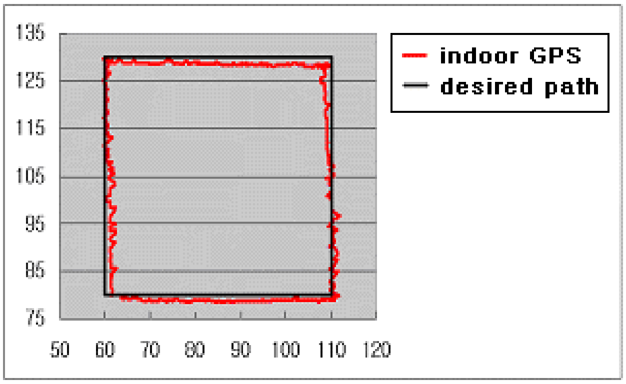



Fig 14 Localized trajectories by indoor GPS sensor

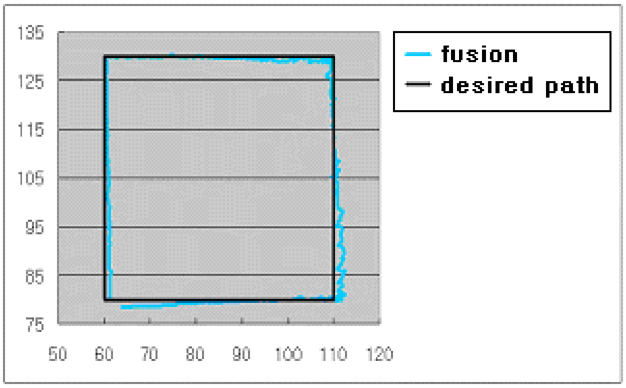



Fig 15 Localized trajectories by sensor fusion method

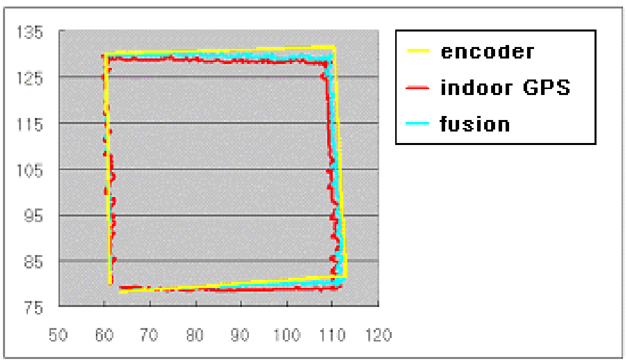
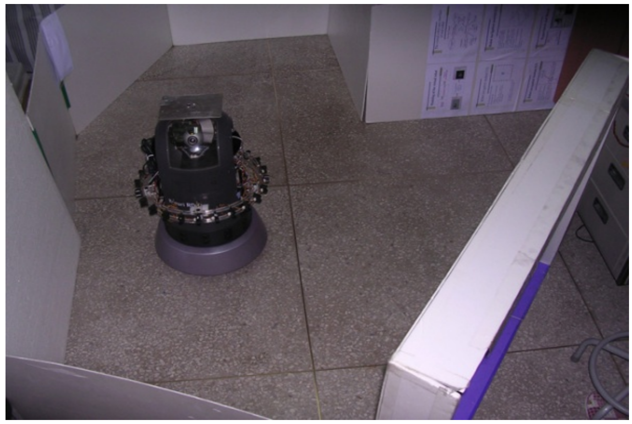
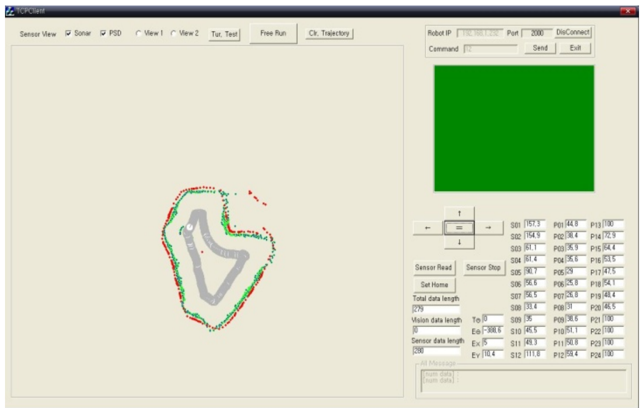



Fig 16 The comparison of the localized trajectories by 3 methods.

Fig.17 shows a built grid map by proposed wall following engine and sensor fusion method. To detect obstacle and wall, we developed PSD sensor controller shown in Fig.9. In the real experimentation, localization has been performed by proposed sensor fusion system. Figure 17(a) shows the indoor environment. Figure 17(b) shows the robot trajectory by proposed wall following engine and the grid map,respectively.



(a)



(b)

Fig.17 Experimental example and built map by proposed wall following engine.

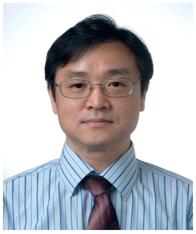
## VII. CONCLUSIONS

A localization and map building system for mobile robot was represented. We developed a sensor fusion system with odomerty and indoor GPS sensor. The odomerty error was broken out at turning motion and its error was accumulated while robot was navigating. But the error in line path is much smaller than that of indoor GPS. On the other hand, the position error of the indoor GPS is not accumulated and is bounded in 12 cm. To compensate the shortcoming of these two sensors, we proposed a localization method with fusion of the two sensors. A wall following decision engine like an expert system was also proposed for map building. We proved this system's validity through field test.

## VII. REFERENCES

[1] K. Chong and L.Kleeman. Accurate odometry and error modeling for a mobile robot. In *Proceedings of the International Conference on Robotics and Automation, 1997*

[2] Alonzo Kelly, Fast and easy systematic and stochastic odometry calibration. In *Proceedings of the International Conference on Robots and Systems, 2004*.
[3] S a D shiv Ghidary, Takahiro Tani, Toshi Takamori, Motofumi Hattori, " *A new Home Robot Positioning System (HRPS) using IR switched multi ultrasonic sensors*"
[4] Chenavier, F. and Crowley, j., 1992, "Position Estimation for a Mobile Robot Using Vision and Odometry." *Proceedings of IEEE International Conference on Robotics and Automation*, Nice, France, May 12-14, pp. 2588-2593.
[5] Feng, L., Borenstein, J., and Everett, B., 1994, "Where am I? Sensors and Methods for Autonomous Mobile Robot Localization." *Technical Report, The University of Michigan, available via anonymous FTP from* ftp,eecs,umich,edu/f/people/jogannb.
[6] Woong keun Hyun, Tae Cheol Jung and Chang wan Lee, "Aline based map building and virtual robot modeling for remote controlled mobile robot," ITC-CSCC, 2005, Vol. 1, p281-282
[7] Cang Ye, Johann Borenstein, "Characterization of a 2-D Laser Scanner for Mobile Robot Obstacle Negotiation", Proc. IEEE Int. Conf. On Robotics and Automation, Washington, DC, December,pp.2512-2518,2002
[8] T. Bailey, J. Nieto, and E. Nobot. Consistency of the FastSLAM algorithm. In *Proc. IEEE int. Conf. Robotics and Automation, 2006.*
[9] Johann Borenstein and Liqiang Freg, "Measurement and correction of systematic odometry errors in mobile robots," IEEE trans. On robotics and automation. Vol.12, No.6, DEC.,1996,pp869-880.



**Woong keun hyun**
He received BS, MS and Phd degree from Hanyang University. He has been a professor of department of electronics engineering of Honam University since 1993. His current researches are localization and navigation of robot.