The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

A Navigation System for Family Indoor Monitor Mobile Robot

Fusheng Tan, Jun Yang, Jianming Huang, Tinggang Jia, Weidong Chen and Jingchuan Wang

Abstract—**The navigation system of family indoor mobile robot includes localization, path planning, collision avoidance. The hybrid localization method of straight line matching, corner matching and odometry is proposed. The hardware and software configuration is introduced. Robot detects environment using a 2D laser range finder. Line feature extraction process including area divided, iterative end point fit (IEPF) and a least square technique is introduced. Based on line feature, straight lines and corners as geometry features are obtained. The odometry localization algorithm, straight line localization algorithm and corner localization algorithm are discussed. Artificial Potential Field(APF) based path planning algorithms is implemented. As a result stable localization is achieved with position and orientation resolution as 50mm, 5 degree. A good performance for the method is also achieved with cycle time as 120ms. Experiment shows the effectiveness of the hybrid localization method.**

I. INTRODUCTION

THE concept of intelligent family living support system was proposed recently with the quick technological progress such as laser sensors, controllers and network [1]. The center of intelligent family living support system is the family indoor monitor robot (FIMR) which is developed in Central Academe of Shanghai Electric (***reference to attach video***). The robot can patrol anywhere indoor and acts as housekeeper to command domestic electrical appliances or robots such as electrical curtains, TV set, cooking robot, education robot, nursing robot to provide service to family members; it can provide services such as opening refrigerator directly by the two dexterous arms. To finish the above tasks, the navigation of mobile robots including path planning, collision-free movement, and localization should be designed. One important and basic function is the localization, to obtain robot position and orientation (pose) in the environment.

For the environment of indoor service robot such as office, supermarket, family living house is mostly structured with plat floors, vertical walls, standard entrance hall and corridors [2]. In such environment, robot will often interact with human and some furniture can be moved; human also is dynamic. Robot shuttles back and forth in the environment and should know its pose in real time.

Manuscript received March 9, 2010. The project (No. 2007AA041604) was supported by the National High Technology Research and Development Program of China.

Fusehng Tan, Jun Yang, Jianmin Huang and Tinggang Jia are within Central Academe of Shanghai Electric Group co. LTD, No. 960 Zhongxin Rd, Shanghai 200065, China (corresponding author to provide phone: +86-021-26027844; fax: +86-021-26027848; e-mail: tanfusheng@yahoo.cn).

Weidong Chen (IEEE member) and Jingchuan Wang are within School of Electronics, Information and Electrical Engineering, Shanghai Jiaotong University, No. 800 Dongchuan Rd, Shanghai, China.

Great efforts were done for robot localization using odometer as proprioceptive senor and vision sensor or laser range finder (LRF) as exteroceptive sensors and data estimation is carried out through Extended Kalman filter [3-5]. For structured environment, straight line matching method had been used for mobile robot location [6] with vision algorithm. But straight lines had limitation that the lines must be parallel or vertical with coordinate system axis X or Y, otherwise the lines can not be used for line matching. The corner composed by straight lines or non-straight lines can be used for mobile robot localization as a supplement to the straight line localization.

In this paper, the map is geometry constrained as corners and straight lines using odometer and LRF with analyzing the indoor structured environment. Software configuration is implemented using National Instrutment Labview platform to gain satisfied real time performance with CompactRIO support.

The paper is organized as followings. The first part introduces the background and related works with FIMR. The second part describes the hardware and software configuration; the definition of the corner and straight line is also proposed in this part. The third part gives the detail localization matching method including odometry, corner matching and straight line matching and this part also discusses the hybrid localization method by weight distribution of the error model. The fourth part will discuss APF algorithm. Finally the experiment and summary are given in the last part.

II. SYSTEM OVERVIEW

A. Hardware configuration

FIMR's chassis has two drive wheels in the middle, two casters in the front and two casters in the back. The chassis is duo-wheel differential driven with 135KG loading capacity. Its controller is National Instruments (NI) CompactRIO (CRIO) programmable automation controller. The controller is an advanced embedded control and data acquisition system designed for applications that require high performance and reliability. With the system's open, embedded architecture, small size, extreme ruggedness, and flexibility, LabVIEW FPGA and LabVIEW Real-Time technologies can be easily used to design, program, and customize the CRIO embedded system with easy-to-use graphical programming tools [7].

As for FIMR's localization and navigation hardware system, it composes of NI CRIO with IO card 9264 and 9403, laser range sensor (SICK LMS111), driving unit (DSP), encoders (OMRON) and motors in Fig. 1.

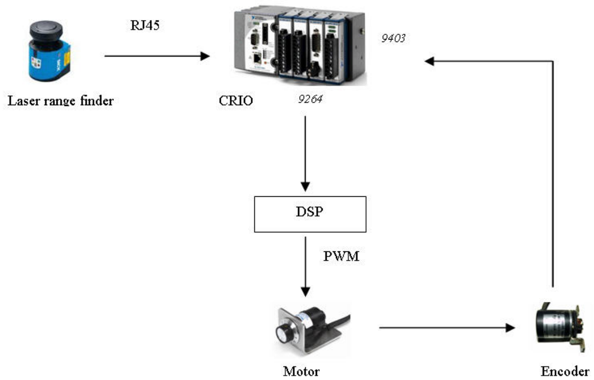


Fig.1. Hardware configuration of FIMR's navigation system.

1) LMS111: Has compact size (102×152×105mm3) and provide external sensor input for localization and navigation system and the communication between LRF and CRIO with TCP/IP protocol. The scan angle is 270° with resolution set as 0.5°. According to the manufacturer's specifications [8], the scanner error is up to ±30mm within the maximum range 18 meters.

2) NI CRIO: NI CRIO acts as the central controller reading encoder data by 9403 module, reading LRF data by RJ45 interface. With localization and navigation algorithm, CRIO calculates the robot pose in real time. When the tasks pose is given, CRIO gives movement instructions (PWM wave) to the driven motors by 9264 module.

3) Encoders: Two encoders are installed around the driven wheel by adhesion wheel to provide information for odometry. The other side of the structure is installed with spring to keep the constant pressure force. Because the encoders are not installed in the shaft, the ratio between each pulse and wheel moving distance can not be directly calculated with encoder resolution; the ratio is gained manually using weight average method with robot moving forward and backward at given times.

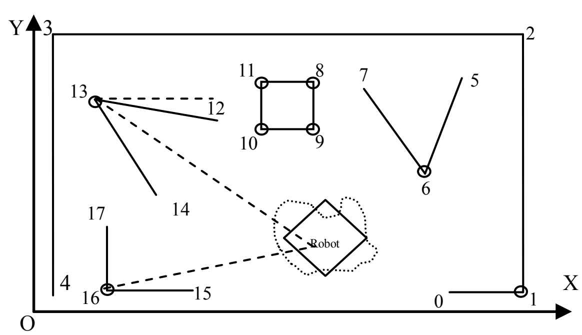


Fig.2. Straight lines and corners definition of indoor structured environment.

B. Geometry features definition

As Fig.2 shows, the indoor family environment can be treated as structured environment. The geometry features such as straight lines and corners can be abstracted from desks, walls, gates, corridors. The geometry features is collected from natural environment and act as given marks or mapping information for robot localization. The diamond shape is the robot and left part is robot's front area. Because of the outer covering issues, the LRF can only scan front part of the efficient scanning area. Only the geometry features 2-3-4, 12-13-14, 17-16-15 can be observed by LRF in counterclockwise rotation direction.

1) Corners: 12-13-14 is a corner. 12-13 is the start line of the corner and 13-14 is the end line of the corner. Point 13 is the angular vertex. So the corner 12-13-14 is uniquely defined as $C(X, Y, \alpha, \beta)$. (X, Y) is coordinate value in global coordinate system. α, β is the angle between line 13-12, 13-14 and OX. The angle direction is counterclockwise rotation direction from 0° to 360°. *2)Straight lines:* Another geometry is straight lines. Line 2-3, 3-4, 17-16, 16-15 can be pick up as straight lines as mapping information. The straight line picking criterion is that the line's slope is 0 or infinity in the global coordinate system. The line feature is defined as $(k, b, L, Xs, Ys, Xe, Ye)$. The slope of the line is k; 0 is defined for the parallel lines and 500 is defined for the vertical lines. Value b comes from line's function

$$y = kx + b \qquad (1)$$

(X_s, Y_s) is the start point and (X_e, Y_e) is the end point.

C. Software configuration

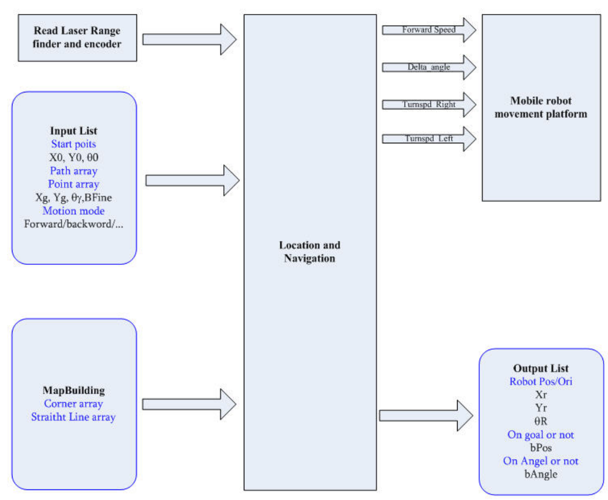


Fig. 3. Localization and navigation software configuration of FIMR.

The software configuration shows in Fig.3. The center is location and navigation module; it reads start point, goal point, motion mode data and encoder data; it also read LRF data and corner and straight line array mapping information and matching the calculated feature with given feature. When the local feature is matched, the matching algorithms will output the robot pose as $P(X_r, Y_r, \theta_r)$. With the task information with goal pose, the module calculates the driving instructions as (Forward Speed, Delta angle, Turnspd Right, Turnspd Left) and the status such as (bPos, bAngle) to judge whether the robot is on the target pose. Here Forward Speed is instruction to move robot forward; Delta angle is instruction to turn; Turnspd Right or Turnspd Left is the steer power parameters.

III. Hybrid Localization Method

Hybrid localization method integrates odeometry localization method, straight line matching method and corner matching method. According to the error model, weight average method is used to get the optimization pose by distributing different weight value. In this part, odometry model is introduced firstly. Secondly, line feature extraction and filtering from LRF raw data is introduced. Data association algorithm for straight line and corner feature is introduced later. Finally, error model and weight distribution is given.

A. Odometry localization algorithm

Odometry localization model data is obtained from encoder installed in the double differential driven wheel. Here ΔS_{Rk} , ΔS_{Lk} are the right wheel and left wheel distance during time period K. According to the differential drive model, therefore,

$$\Delta S_k = (\Delta S_{Lk} + \Delta S_{Rk})/2 \quad (2)$$

$$\Delta\theta_k = (\Delta S_{Rk} - \Delta S_{Lk})/W \quad (3)$$

where ΔS_k , $\Delta\theta_k$ are average distance and turn angle of the robot, W is the width of the between right wheel and left wheel.

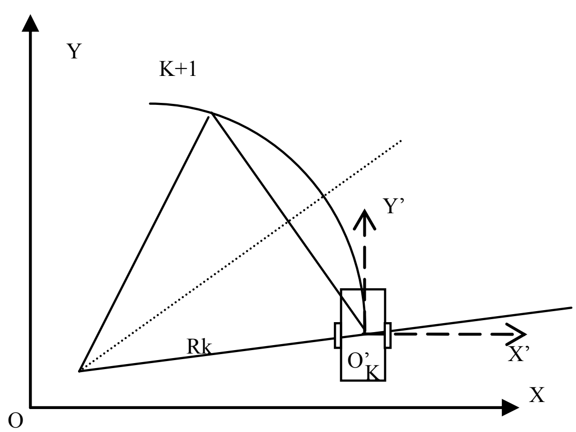


Fig. 4. Odometry model during time period K.

Fig. 4 shows the little arc of robot in time period K from $P_k(X_k, Y_k, \theta_k)$ to $P_{k+1}(X_{k+1}, Y_{k+1}, \theta_{k+1})$. Fig.4 also shows the global coordinate system XOY and local mobile coordinate system X'O'Y'. Therefore, the central task is how to get data P_{k+1} from P_k , ΔS_k and $\Delta\theta_k$.

So $P_{k+1}(X_{k+1}, Y_{k+1}, \theta_{k+1})$ is denoted as

$$X_{k+1} = X_k - 2\frac{\Delta S_k}{\Delta\theta_k}\sin\frac{\Delta\theta_k}{2}\sin(\theta_k + \frac{\Delta\theta_k}{2}) \quad (4)$$

$$Y_{k+1} = Y_k + 2\frac{\Delta S_k}{\Delta\theta_k}\sin\frac{\Delta\theta_k}{2}\cos(\theta_k + \frac{\Delta\theta_k}{2}) \quad (5)$$

$$\theta_{k+1} = \theta_k + \Delta\theta_k \quad (6).$$

The above odometry arc model is based on that the threshold value $\Delta\theta_k$ is more than 0.025. If $\Delta\theta_k < 0.025$, the odometry model is simplified as line model. The pose $P_{k+1}(X_{k+1}, Y_{k+1}, \theta_{k+1})$ is denoted as

$$X_{k+1} = X_k - \Delta S_k \sin(\theta_k + \Delta\theta_k) \quad (7)$$

$$Y_{k+1} = Y_k + \Delta S_k \cos(\theta_k + \Delta\theta_k) \quad (8)$$

$$\theta_{k+1} = \theta_k + \Delta\theta_k \quad (9)$$

Obviously the localization error using odometry model grows with distance accumulation. So the external sensor LRF must be used to avoid the error growing.

B. Line feature extraction and filtering

To get geometrical information of the data of the LRF, a segmentation process is required. The method can be divided into the following processes: coordinate system transformation, area divided, iterative end point fit (IEPF) based line feature extraction [9], line parameters calculation using a least square technique [11].

C. Straight line matching localization algorithm

Straight line matching condition is realized as the following in the order of priority.

1) Contrast the line length between local line and global line of the map: Firstly the local line length is checked; if it is smaller than a present threshold value L, the line is omitted to go to next step. If the line length passes the threshold, only the local line whose length is smaller than global line length can go to next step.

2) Coordinate system transformation for the local lines: With the reference pose by hybrid localization method, the local line parameters are transformed to global coordinate system.

3) Angle similarity: The angle θ^l between local line and global coordinate X axis was calculated using K parameter from step 2. Also θ^g between global line and global coordinate X axis is obtained using global line parameters. The difference between global angle θ^g and local angle θ^l is checked; if it is smaller than a present threshold value θ_{min} , go to next step.

4) The scope of start point and end point: To satisfy the matching condition, the scope of local line should be within the scope of global line. For parallel and vertical line, there is different condition. For parallel line, the matching condition is denoted

$$(X_s^l - X_s^g)(X_s^l - X_e^g) \le 0 \;\&\&\; (X_e^l - X_s^g)(X_e^l - X_e^g) \le 0$$

where $(X_s^l, Y_s^l),(X_e^l, Y_e^l)$ is the transformed coordinate value of the local line's start point and end point and $(X_s^g, Y_s^g),(X_e^g, Y_e^g)$ is the global line's start point and end point. For the vertical line, the matching condition is denoted $(Y_s^l - Y_s^g)(Y_s^l - Y_e^g) \leq 0 \,\&\&(Y_e^l - Y_s^g)(Y_e^l - Y_e^g) \leq 0$.

Satisfying with the above straight line matching condition, the parallel lines and vertical lines array is obtained. The object of the straight line matching localization algorithm is to obtain calibrated parameter $\Delta P(\Delta x, \Delta y, \Delta\theta)$. As for the parallel line matching case, only $(\Delta y, \Delta\theta)$ is calibrated. The same with vertical line matching case, $(\Delta x, \Delta\theta)$ is calibrated. In the parallel matching case, $(\Delta x, \Delta\theta)$ is denoted

$$\Delta y = (Y_s^g + Y_e^g - Y_s^l - Y_e^l)/2 \quad (10)$$

$$\Delta\theta = \theta^g - \theta^l \quad (11).$$

For the vertical matching case, $\Delta\theta$ is denoted as (11). Δx is denoted

$$\Delta x = (X_s^g + X_e^g - X_s^l - X_e^l)/2 \quad (12).$$

D. Corner matching localization algorithm

According to the corner definition $C(X, Y, \alpha, \beta)$, two parameters corner vertex coordinate value $V(X, Y)$ and corner angle $\varphi = |\alpha - \beta|$ are considered as matching parameters.

With the reference pose by hybrid localization method, the local corner parameters are transformed to global coordinate system as $V^l(X^l, Y^l)$ and φ^l; in contrast with the global corner parameters $V^g(X^g, Y^g)$ and φ^g, the corner distance difference value Δd between V^l and V^g is denoted

$$\Delta d = \sqrt{(X^l - X^g)^2 + (Y^l - Y^g)^2} \quad (13).$$

The corner angle difference value $\Delta\varphi$ is devoted

$$\Delta\varphi = |\varphi^l - \varphi^g| \quad (14).$$

The corner matching condition ΔC is devoted,

$$\Delta C = K_d \Delta d + K_a \Delta\varphi \quad (15)$$

where K_d and K_a comes from experience ratio between corner distance and corner angel difference. By (15), for a given local corner feature, the minimum value ΔC is obtained therefore the minimum index i in global array is obtained. By the minimum index i, the matching condition is that Δd_i and $\Delta\varphi_i$ must be smaller than threshold value Δd_{min} and $\Delta\varphi_{min}$, otherwise there is no global corner matching with the local corner.

When the two matching corners are found as local corner vertex $V^r(X^r, Y^r)$ and corner's line direction angle α^r or β^r (the value is in local coordinate system without transformation) and global corner vertex $V^g(X^g, Y^g)$ and corner's line direction angle α^g or β^g. The localization algorithm for the corner matching is denoted,

$$X = X^g + X^r\cos\phi - Y^r\sin\phi \quad (16)$$

$$Y = Y^g + X^r\sin\phi + Y^r\sin\phi \quad (17)$$

where (X, Y) is the updating position and $\phi = \alpha^g - \alpha^r$ or $\phi = \beta^g - \beta^r$ depending which line is longer in the local corner (using α if start line is longer or β if end line is longer).

To ensure the matching effectiveness, the gate threshold array is used for the coordinate system increment value. If the coordinate system increment value is smaller than the threshold, the matching is valid otherwise the matching is invalid.

E. Weight value form error model

The pose of the robot P_r is obtained by the LRF with the assistance of the odometry. Therefore P_r is calculated using data fusion with odometry localization data P_o, straight line localization data P_{line} and corner localization data P_c by different weight value according to its error model.

The error of odometry is linear growing with distance accumulation. So W_o is denoted

$$W_o = K_o / \Delta D_o \quad (18)$$

where K_o is constant value, ΔD_o is the distance of robot movement recording from odometry model.

The error of straight line matching localization and corner matching localization is the same for the matching parameters are both denoted by line segmentation. W_{line} is denoted,

$$W_{line} = K_{line} / n^2 \quad (19)$$

where K_{line} is experienced constant value of LRF. So the finally optimization robot pose P_r is denote

$$P_r = \frac{P_{line} \times W_{line} + P_c \times W_c + P_o \times W_o}{W_{line} + W_c + W_o} \quad (20)$$

where W_c is calculated by (19).

IV. APF Based path planning algorithm

The task of APF based path planning is to plan a collision free path between the start pose and goal pose that satisfies certain optimization criteria. In the APF approach used [10-12] for a trajectory generation problem of a mobile robot, the goal is represented by an attractive artificial potential and the obstacles are represented by repulsive ones, so that the trajectory from the start pose to the goal pose can be generated by using a gradient vector of the potential filed. However there is local minima problem. In this paper APF combined with sub-goal method is used to avoid the deadlock problem.

A. APF path planning algorithm

Suppose robot current position is X, goal position is X_g. The attractive force from current position is denoted as

$$F_{at}(X) = -k(X - X_g) \quad (21).$$

There is one obstacle around the robot. Suppose ρ is the distance from robot to the obstacle. ρ_0 is the influence distance for the obstacle. η is constant value related to distance. So the repulsive force is denoted as

$$F_{re}(X) = \begin{cases} F_{re1} + F_{re2} & \rho \le \rho_0 \\ 0 & \rho > \rho_0 \end{cases} \quad (22)$$

where F_{re1}, F_{re2} is denoted as

$$F_{re1}(X) = \eta(\frac{1}{\rho} - \frac{1}{\rho_0})\frac{1}{\rho^2}(X - X_g)^2 \quad (23)$$

$$F_{re2}(X) = \eta(\frac{1}{\rho} - \frac{1}{\rho_0})^2(X - X_g) \quad (24)$$

where the direction of F_{re1} is from obstacle to robot current pose and the direction of F_{re2} is from robot current pose to goal pose. With the summary of attractive force and all repulsive force in coordinate X and Y, the robot moving direction can be calculated.

If the local minima problem happens, valid possible sub goal is calculated as Fig 5. and the goal is replaced with the subgoal. Whenever the minim problems disappears, the goal is pop up again.

B. Collision avoidance strategy

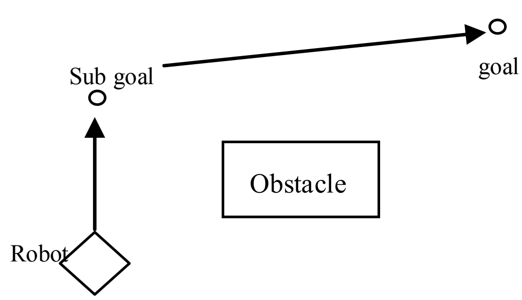


Fig. 5. Sub goal meothod for APF path planning algorithm.

In concern with the dynamic obstacles, collision avoidance strategy is implemented. Firstly three districts are defined. Right district is from LRF 0 to 60 degree. Middle district is from LRF 60-120 degree and the left district is from LRF 120 to 180 degree. If the distance value from one district is below a threshold value, set the bool value of the districe as 1 other as 0. The detail collision avoidance strategy is organized in table 1.

TABLE I
COLLISION DETECTION STATEGY

Bool value for the each district			Robot Action
Right	Middle	Left	
1	1	1	Stop
1	0	1	Stop
0	0	0	Navigate
0	0	1	Turn right
0	1	1	Turn right
0	1	0	Turn right
1	0	0	Turn left
1	1	0	Turn left

Navigate means robot will go forward or turning around with the path planning. Turn right(left) means the robot will turn around with clockwise(anticlockwise) direction.

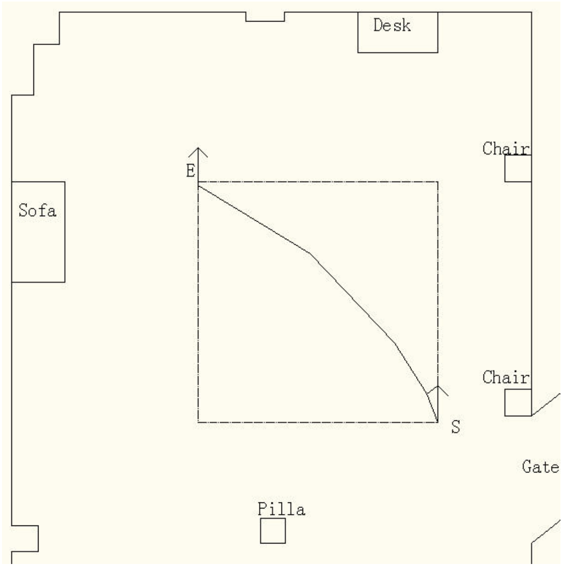
V. EXPERIMENT AND SUMAMARY

The experiment is designed to check the resolution of the hybrid localization method. A 6000mm $\times$ 6000mm district is used for the experiment; the district is equipped with two desks, two chairs, sofa, walls corridor and gate as the map of the environment in Fig.6.

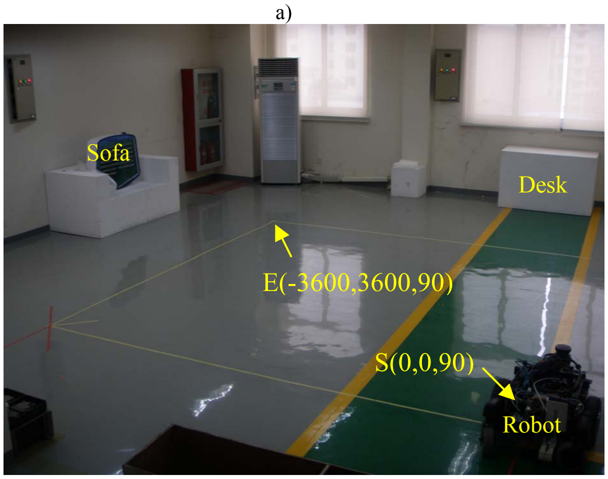
Robot detects environment using LRF with cycle as 25ms and obtains straight lines and corners from the natural features such as the wall, gate, desks, chairs, sofa as in Fig.6. Robot starts from $S(0,0,90^\circ)$ to $E(-3600,3600,90^\circ)$ following the broken line path as in Fig.6a) using odometry localization method and the hybrid localization method fusing straight line, corner and odometry localization data. The robot starts from the same start pose S to end pose T; the localization pose value and real measured pose value are recorded whenever robot stops around the end pose area. Pose error is obtained just comparing the above pose value.

Four groups of pose error are measured in the experiment. Fig. 7 shows the position error XY using hybrid localization method. Fig. 8 shows the position error XY using odometry localization method. Fig. 9 shows direction error using hybrid and odometry localization method.

For one case using the hybrid localization method, the maximum speed is 340mm/s; average speed is 230mm/s; travel distance is 5.5 meters; completing time is 23.9s. Stable localization is achieved with pose resolution as 50mm, 5 degree with average cycle time as 120ms. In contrast with other localization method, the method is practical and simple with high resolution for indoor structured environment. Task such as automatic battery charging is achieved using the method.



a)



b)

Fig. 6. a) Map information in CAD model b)real environment for the hybrid localization experiment (Robot starts from S to E following the broken line path. First robot moves to target position and turns around to reach the target orientation after robot reaches the target position).

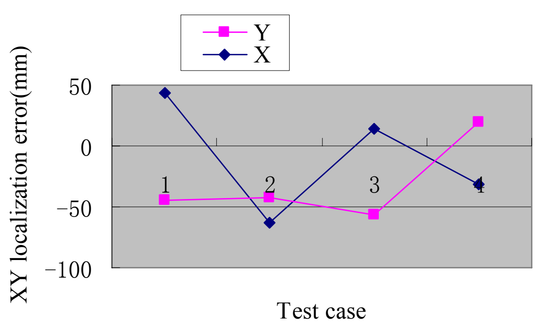


Fig. 7. XY localization error of the hybrid localization method.

REFERENCES

[1] Jun Yang, Jianmin Huang, "Intelligent family living Support System," *Robot Technique and Application*, vol. 6; pp. 10-13, 2007.

[2] LU Xin-hua, ZHANG Gui-lin, "Summarization on indoor service robot navigation," *ROBOT,* vol. 25, no. 1, pp. 80-87, January 2003.

[3] XU Jun-yong, WANG Jing-chuan, CHEN Weidong "Omin-vision-based Simultaneous Localization and Mapping of Mobile Robots," *ROBOT,* vol. 30, no. 4, pp. 289-297, July 2008.

[4] Martinelli, A. and Siegwart, R. 2003, "Estimating the odometry error of a mobile robot during navigation," In European Conference on Mobile Robots (ECMR 2003), Warsaw, Poland, September 2003.

[5] Datta S, Banerji D and Mukherjee R, "Mobile robot localization with map building and obstacle avoidance for indoor navigation," IEEE International Conference on Industrial Technology, Mumbai, India, December 2006.

[6] Aider, O.A.Hoppenot, P.Colle, E., "A Model to Image Straight Line Matching Method for Vision-Based Indoor Mobile Robot Self-Location," in IEEE/RSJ International Conference on Intelligent Robots and System, Lausanne, Switland, pp. 460-465, Sep 30-Oct 4, 2002.

[7] NI CompactRIO Control and Acquisition System, http://www.ni.com/compactrio/whatis.htm

[8] LMS100/111/120 Laser Measurement Systems operating instructions Manual, 2008

[9] A. Siadat et al. "An Optimized Segmentation Method for a 2D Laser-scanner applied to Mobile Robot Navigation," 3rd IFAC Symposium on Intelligent Components and Instruments for Control Application-SICICA, pp. 153-158, June, 1997.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," International Journal of Robotics Research, vol. 5,no. 1, pp. 90–98, 1986.

[11] G. Dozier, A. Homaifar, S. Bryson, and L. Moore, "Artificial potential field based robot navigation, dynamic constrained optimization and simple genetic hill-climbing," in The 1998 IEEE International Conference on Evolutionary Computation, pp. 189–194, May 1998.

[12] P. Vadakkepat, K. Tan, and M. Wang, "Evolutionary artificial potential fields and their application in real time robot path planning," in Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 256–263, July 2000.

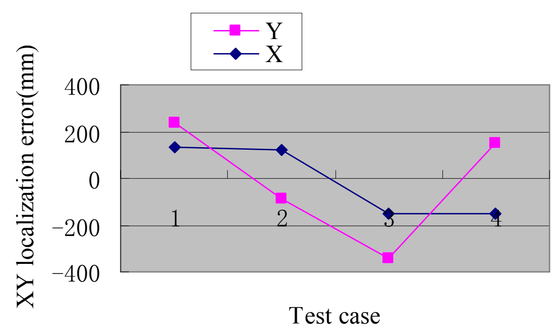


Fig. 8. XY localization error of the odometry localization method.

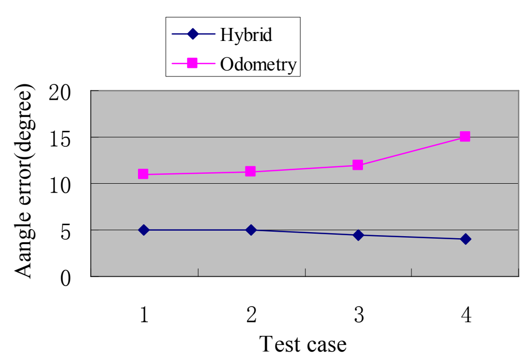


Fig. 9. Direction error of the hybrid localization method and odometry method.