IEEE International Conference on Emerging Technologies & Factory Automation, UPC, Barcelona, Spain, 18-21 October 1999

Landmark-based Navigation of Mobile Robots in Manufacturing

Huosheng Hu † *and* ***Dongbing Gu*** ✠

† Department of Computer Science, University of Essex, Wivenhoe Park,
Colchester CO4 3SQ, England

✠ Department of Electronics, Changchun Institute of Optics and Fine Mechanics,
Changchun, Jilin, 130022, China

Abstract

Landmark-based navigation of autonomous robots has been widely used in manufacturing industry. Such a navigation strategy relies on identification and subsequent recognition of distinctive environment features or objects that are either known a priori or extracted dynamically. This process is inherently difficult in practice due to noise in sensors and changes in the real world. This paper presents a navigation algorithm that locates the robots and updates landmarks in a dynamic manufacturing environment. A key issue being addressed is how to improve the localization accuracy for mobile robots in a continuous operation, and how to initialize and recalibrate the robot position when necessary. Kalman filter algorithms are adopted to integrate odometry data with scanner data to achieve the required robustness and accuracy. Kohonen neural networks are used to recognize landmarks in the process of initialization and re-calibration of the robot position.

Key words: Mobile Robots, Kalman Filter, Localization, Landmarks, Triangulation, Neural Networks.

1. INTRODUCTION

To be useful in a wide range of industrial applications, autonomous mobile robots need the capability to explore and navigate in dynamic environments. In the past two decades, a number of different approaches have been proposed to develop flexible and efficient navigation systems for manufacturing industry, based on different sensor technologies such as odometry, laser scanners, inertial sensors, sonar and vision [1] [11]. However, since there is huge uncertainty in the real world and no single sensor is perfect, it remains a great challenge today to build robust and intelligent navigation systems for mobile robots to operate continuously in the real world.

In general, the methods for locating mobile robots in the real world are divided as two categories: relative positioning and absolute positioning. In relative positioning, odometry (or dead reckoning) and inertial navigation [3] are commonly used to calculate the robot position estimates from a start reference point at a high updating rate. As we know that odometry is one of most popular internal sensors for position estimation because of its easy to use in real time. However the disadvantage of odometry is that it has an unbounded accumulation of errors. Therefore, frequent correction made from other sensors becomes necessary [4]. In contrast, absolute positioning relies on detecting and recognizing different features in the robot environment in order for a mobile robot to reach a destination and implement specified tasks. These environment features are normally divided as two types: one is feature-based, such as active beacons [14], artificial [5] and natural [15] landmarks. Another is map-based such as different geometrical models [7] [18]. Among these, natural landmark navigation is flexible as no explicit artificial landmarks are needed, but may not function well when landmarks are sparse and often the environment must be known a priori. Although artificial landmark and active beacon approach are not flexible, the ability to find landmarks is enhanced and process of map building is simplified.

To make the use of mobile robots in the daily deployment feasible, it is necessary to reach a tradeoff between costs and benefits. Often, this prevents the use of expensive sensors such as vision systems in favor of cheaper sensing devices such as odometry and laser, and calls for efficient algorithms that can guarantee real-time performance in the presence of insufficient or conflicting data. Therefore, the focus of this paper is on the use of a laser scanner and artificial landmarks for the position estimation of a mobile robot. The natural landmarks are also considered for its extension.

The localization system based on the laser scanner and artificial landmarks is a promising absolute positioning technique in terms of performance and cost [2]. Using this technology, the artificial landmarks in the environment are scanned and their bearing relative to each other is measured. Then the position estimation of the mobile robot is normally calculated by using two distinctive methods: triangulation [16] [17] and Kalman filtering algorithm [6]. However, the accuracy of the position estimate is affected by the unexpected changes in landmark's positions and noises in laser measurements. When the robot moves, the accuracy of robot positioning degrades gradually, and sometime becomes unacceptable during a continuous operation. Therefore, re-calibration is needed from time to time and it becomes a burden for the practical applications. To solve this problem, a navigation algorithm is proposed in this paper, which is the combination of localization and feature-based map-building capabilities.

A navigation system based-on a rotating laser scanner and artificial landmarks is described in next section. Traditional triangulation methods for calibrating the mobile robot position are also described. Then the Kohonen neural networks are proposed in section 3 for the mobile robot to recognize the landmarks automatically in order to initialize and recalibrate its position by means of triangulation. In section 4 the localization system based on the Extended Kalman Filter (EKF) is presented, which integrates date from both the scanner and odometry. Experiment results are given to show its applicability. Section 5 presents a navigation system that can locate the robot and update landmarks in the robot internal model simultaneously. Both artificial and natural landmarks are considered. Finally, a brief summary is given in section 6.

2. NAVIGATION SYSTEM

2.1 Laser scanner and artificial landmarks

Research here is based on an industrial robotic system that uses artificial landmarks to the robot localization. A rotating laser scanner is used to measure the angle between base line and the beam line from either the leading or the falling edge of artificial landmarks in the horizontal plane. As shown in figure 1, the laser scanner, GCS, is situated on the top of the physical center of the robot, scanned in azimuth up to 50m range at a constant speed of 2Hz anti-clockwise. Note that an infrared laser beam (870nm) from a HeNe laser diode emits energy of 0.5 mw which is eye safe. As can be seen, there are 6 landmarks in this environment, namely T_1, T_2, T_3, T_{35}, T_{36}, and T_{37}. Another laser scanner, SICK, is located at the front of the robot, scanning 180° horizontally (see shadow part) to detect obstacles in the front of the robot.

The landmarks are currently in the form of single strip for easy detection at a long distance, instead of traditional bar-coded. All landmarks have an identical size of 50cm in length and 10cm in width. The positions of landmarks are surveyed in advance and pre-stored into the robot memory in a format of the looking-up table, represented by the coordinates in the world frame:

$$\Re = [T_1,...,T_i,...,T_N] = [(t_{1x},t_{1y}),...,(t_{ix},t_{iy}),...] \qquad (1)$$

where (t_{ix}, t_{iy}) -- the coordinates of the *i'th* landmarks.

N -- the total number of the landmarks in use.

These landmarks can return strong reflective signals to the scanner, i.e. the area inside dotted lines in figure 1. The reflected light from these landmarks is measured by a photodetector inside the scanner. The scanner outputs the relative angles (with respect to the robot frame) measured by the scanner encoder at the falling edge of each landmark. Figure 2 shows 921 angle measurements

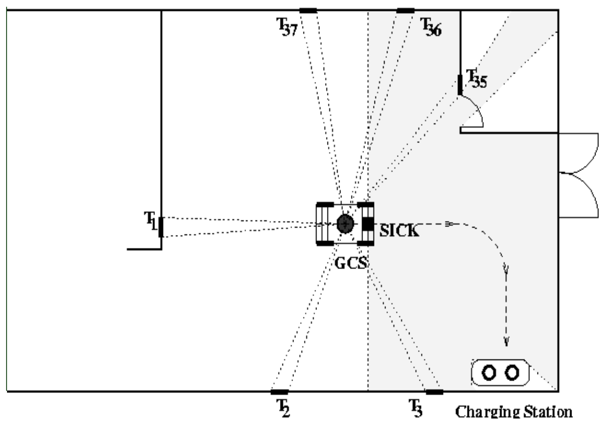


Figure 1 Landmarks and a laser scanner

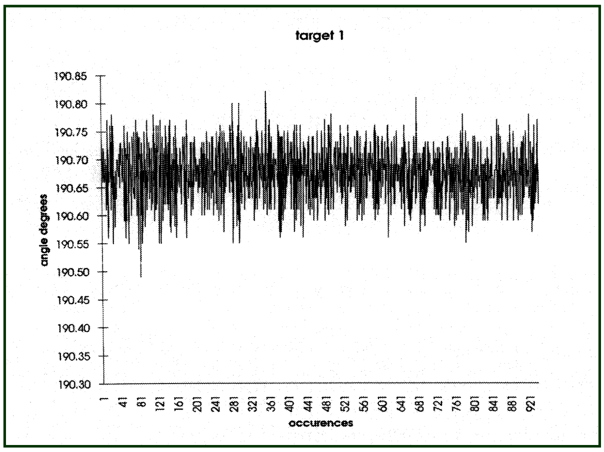


Figure 2 Angle measurements by the scanner

from the falling edge of the landmark T_1. It should be noticed that these results were obtained when the mobile robot was stationary. The variance in the measurements is about 0.3 degree. The measurement variance would increase when the mobile robot moves around. This is because the vibration of the laser scanner would appear when the floor surface is not smooth.

2.2 Triangulation algorithm

In the case of a stationary robot, the laser scanner senses all six landmarks, as shown in Figure 1, from a single location continuously. Then these data can be used to calculate the initial position and heading of the robot by the triangulation algorithms proposed in [16] [17]. The basic problem is to choose any three measurements to do triangulation. As shown in figure 3, it is actually identical to the "3 point problem" in land surveying. The laser scanner detects the falling edges of the observed landmarks and in turn provides angle measurements:

$$\beta_i = \arctan\frac{t_{iy} - y}{t_{ix} - x} - \theta \qquad (2)$$

where β_i is the angle measurement of the *i'th* landmark relative to the heading of the mobile robot, and (x, y, θ) is the robot position vector, as shown in figure 3.

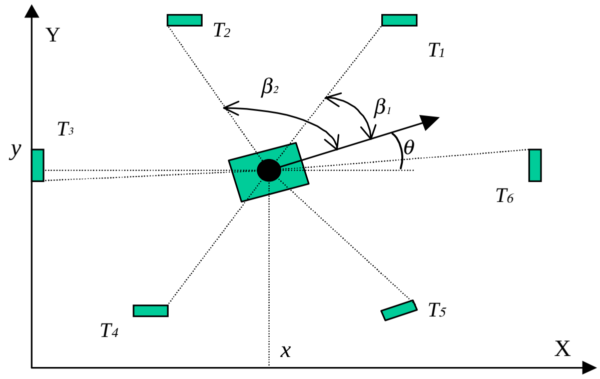


Figure 3 Triangulation example

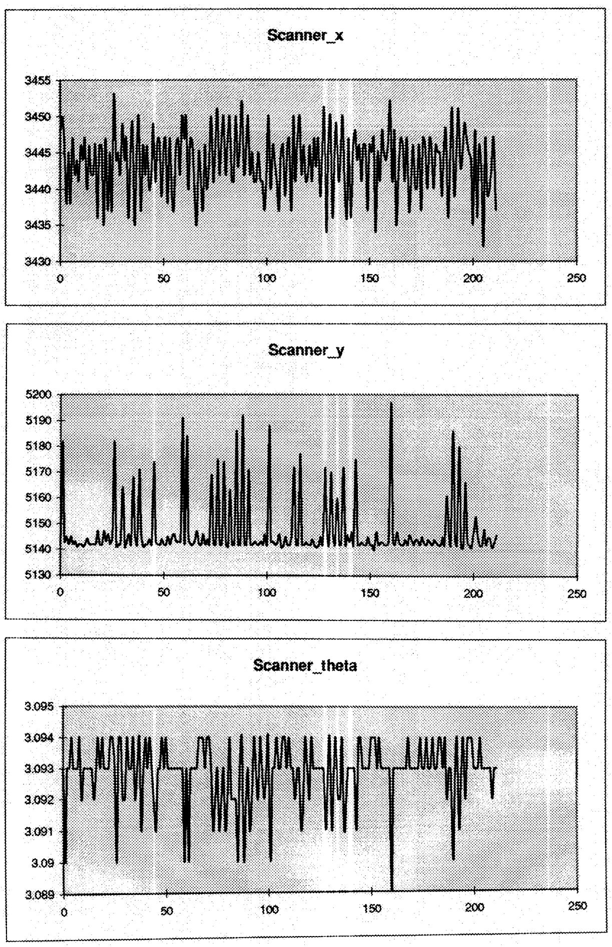


Figure 4 Locating the robot by triangulation

Based on the trigonometric identity, the equations for calculating the robot position and orientation are easy to derive from equation (2), see [17] for more details. It should be noticed that triangulation can be recursively implemented by choosing three landmarks from those 6 landmarks in figure 1 in turn when the mobile robot is stationary. The localization results are shown in figure 4, where the unit for *scanner_x* and *scanner_y* is *millimeter* and the unit for *scanner_theta* is *radian*.

There are two problems in this triangulation process. Firstly, the calculated y coordinate of the mobile robot jumped a lot, as can be seen, since the triangulation algorithm is normally sensitive to the positions of three landmarks being used. When three targets are in optimal position (about 120 degrees apart), the results are very accurate. Otherwise, the robot position and orientation have big variances with respect to an optimal value [16]. Secondly, it is very difficult to identify which landmark is been detected since all landmarks are identical. Mismatch is more likely to happen in practice when obstacles obscure one or more landmarks in the robot surroundings.

3. KOHONEN NEURAL NETWORK FOR LANDMARK DETECTION

Since the laser scanner can only measure the angles to the different landmarks, and cannot distinguish one landmark from another, a key problem is how to determine the correspondence between the measured angle and the landmark [2]. Therefore, the initialization of the robot position is normally done manually. Also, when the mobile robot gets lost, re-calibration is done manually. This is not convenient for practical applications. It is necessary to find a feasible way to initialize the position of a mobile robot automatically, which is addressed in this section.

3.1 Kohonen neural networks

The Kohonen neural networks are adopted here to recognize the landmarks using the measurements of the laser scanner in order to provide landmark's coordinates for triangulation [8][13]. Figure 5 is a schematic diagram for the Kohonen neural network. The inputs for the network are chosen as the angles of two landmarks for eliminating the effect of the robot's heading on the neural network, i.e.

$$\alpha_i = \beta_{i+1} - \beta_i \qquad (3)$$

So, there are five inputs (a pattern) in the network for six landmarks during each scanning circle. Therefore, there are six input patterns that the robot can get by choosing different angle readings in turn.

The Kohonen neural networks are then built up, which correspond to the six input patterns. When the neural networks mature, the input patterns can be recognized by the corresponding neural network that has minimum summary of the Euclidean distance between the network's weights and the input pattern. The nodes in the Kohonen neural network reflect the robot's position in the operating environment. At the current stage, only 30×30 nodes are used, which may be expanded more nodes later for high accuracy.

3.2 Training the network

The Winner-Take-All learning algorithm is adopted here to train the parameters for the Kohonen neural networks to locate the robot. Assume that the network weights are W_{ij}, and the input data is $A = \{a_i\}$, where i=1,2,…,5 and

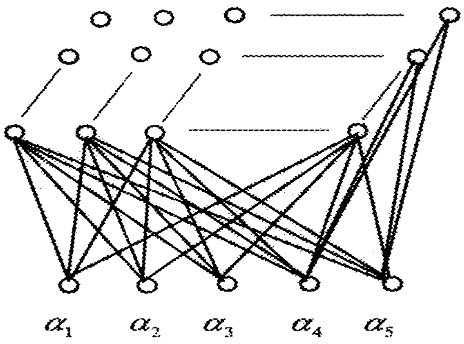


Figure 5 Kohonen neural network

j=1,2,…,30 for six landmarks situation in this research. The winner node W_N is determined by

$$\|A - W_N\| = \min_{i,j} \|A - W_{ij}\| \tag{4}$$

The winner node and its neighbors are updated by

$$W_n(k+1) = W_n(k) + \eta_\varepsilon (A - W_n), \quad n \in N_\varepsilon \tag{5}$$

where ε is the epoch of the learning, η_ε is the learning rate parameter and N_ε is the neighborhood size at the given epoch.

The learning rate and the neighborhood size are adapted with the epoch as follows

$$\eta_\varepsilon = \left(\frac{\eta_0 - \eta_f}{\varepsilon_{\max}} \right) \varepsilon - \eta_0 \tag{6}$$

$$N_\varepsilon = \begin{cases} 5 & \varepsilon \geq \frac{1}{2}\varepsilon_{MAX} \\ 3 & \frac{1}{4}\varepsilon_{MAX} < \varepsilon < \frac{1}{2}\varepsilon_{MAX} \\ 1 & \varepsilon < \frac{1}{4}\varepsilon_{MAX} \end{cases} \tag{7}$$

where η_0 is the initial rate, η_f is the final rate and ε_{MAX} is the maximum epoch.

3.3 Recall

After training, there are six mature Kohonen neural networks for six possible input patterns. Recall procedures are as follows:

1) To get a measurement data $A = \{a_i\}$ from the laser scanner.
2) To find the winner node W_k for the *k'th* Kohonen neural network.
3) To calculate the Euclidean distances between the measurement data $A = \{a_i\}$ and the weight of the winner node W_k.
$$D_k = \|A - W_k\| \tag{8}$$
4) To find the minimum Euclidean distance among $Dk\{k$=1,2 .... 6}.
$$D_{\min} = \min_k \{D_k\} \tag{9}$$
5) To obtain the winner network to which the input pattern is corresponding.

By the input pattern being identified, the robot then finds out the correspondence between the measured angle and the actual landmark. The self-localization can then be implemented by means of triangulation.

Table 1 Simulation results

EPOCH	THE NUMBER OF INPUT DATA	THE NUMBER OF CORRECT RECOGNIZED
500	10	7
	50	39
	100	87
1000	10	9
	50	48
	100	94

Table 1 presents the recognized results by the neural network with parameters as follows:

$$\eta_0 = 0.5 \quad , \quad \eta_f = 0.01$$

It shows that the neural network can properly recognize the landmarks after 1000 times training. Note that this is only preliminary study and only six landmarks are considered so far. The next stage of our research is to extend this algorithm into more landmarks and the real mobile robots.

4. KALMAN FILTER BASED NAVIGATION SYSTEM

The triangulation algorithm is difficult to implement when the robot moves around. This is because it is necessary to compensate the time frame since each of landmarks is detected at different robot positions. Skewis and Lumelsky proposed a triangulation algorithm to attack this problem [17]. However, there was no satisfactory result being obtained in our testing, mainly due to the following reasons:

- Each of landmarks is in a single strip and not encoded, i.e. indistinguishable one another.
- Noisy readings come from the laser scanner as some of data is caused by random objects.
- In general, the robot environment is complex, and not all landmarks can be seen by the laser scanner. Moreover some landmarks may be obscured by dynamic objects such as humans and other robots.

Therefore, the Extended Kalman Filter (EKF) has been used in the localization process, which is addressed here.

The Kalman filter algorithm is a natural choice for robot localisation since it provides a convenient way to fuse the data from multiple sensors, e.g. the laser scanner and odometry in this work. However, it normally requires a linear dynamic model and a linear output model. In this research, both models are nonlinear as follows:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k) \tag{10}$$

$$\mathbf{z}(k+1) = \mathbf{h}(\beta_i, \mathbf{x}(k)) + \mathbf{v}(k) \tag{11}$$

where **f**(**x**(k), **u**(k)) is the nonlinear state transition function of the robot. **w**(k) ~ **N**(**0**, **Q**(k)) indicates a Gaussian noise with zero mean and covariance **Q**(k). **h**(βi, **x**(k)) is the nonlinear observation model and **v**(k) is again Gaussian noise **N**(**0**, **R**(k)).

The input vector is measured by two optical encoders measuring rear wheel motion at each cycle time k:

$$\mathbf{u}(k) = [\Delta d, \Delta\theta]^T \tag{12}$$

where Δd is the distance traveled and $\Delta\theta$ is the orientation change.

The state transition function of the robot is

$$\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k) = \begin{bmatrix} x(k) + \Delta d\cos\theta \\ y(k) + \Delta d\sin\theta \\ \theta(k) + \Delta\theta \end{bmatrix} \tag{13}$$

For the laser scanner, the observation model is

$$\mathbf{h}(\beta_i, \mathbf{x}(k)) = \tan^{-1}\frac{t_{iy} - y(k)}{t_{ix} - x(k)} - \theta(k) \tag{14}$$

Since both model (13) and model (14) above are nonlinear, the EKF must be used here to integrate the laser measurements and readings from optical encoders. Note that the EKF is recursively implemented as follows:

Step 1: **Prediction**

$$\mathbf{x}(k+1/k) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \tag{15}$$

$$\mathbf{P}(k+1/k) = \nabla\mathbf{f}\mathbf{P}(k/k)\nabla\mathbf{f}^T + \mathbf{Q}(k) \tag{16}$$

where ∇**f is** the Jacobean matrix of the transition function, and is obtained by linearization

$$\nabla\mathbf{f} = \begin{bmatrix} 1 & 0 & -\Delta d(k)\sin\theta(k) \\ 0 & 1 & \Delta d(k)\cos\theta(k) \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

Step 2: Observation

The measurement of the laser scanner is

$$\mathbf{z}(k+1) = \beta_i \tag{18}$$

The predicted angle measurement is

$$\hat{\mathbf{z}}(k+1) = \mathbf{h}(\beta_i, \hat{\mathbf{x}}(k+1/k)) \tag{19}$$

To calculate the innovation, then use

Figure 6 Experimental mobile robot

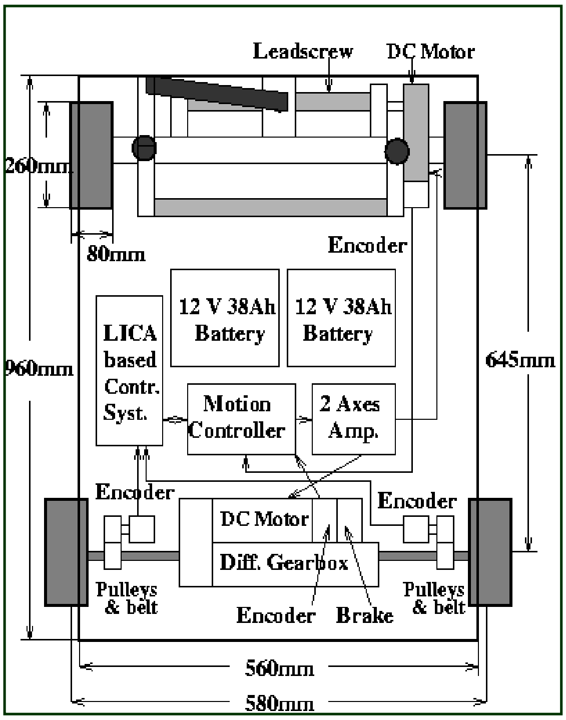


Figure 7 Schematic Diagram of the vehicle base

$$\mathbf{v}(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1) \tag{20}$$

The innovation covariance is:

$$\mathbf{S}(k+1) = \nabla\mathbf{h}\mathbf{P}(k+1/k)\nabla\mathbf{h}^T + \mathbf{R}(k+1) \tag{21}$$

where ∇**h** is the Jacobean matrix of the measurement function:

$$\nabla\mathbf{h} = \begin{bmatrix} \frac{\partial\mathbf{h}}{\partial\mathbf{x}}, & \frac{\partial\mathbf{h}}{\partial\mathbf{y}}, & -1 \end{bmatrix} \tag{22}$$

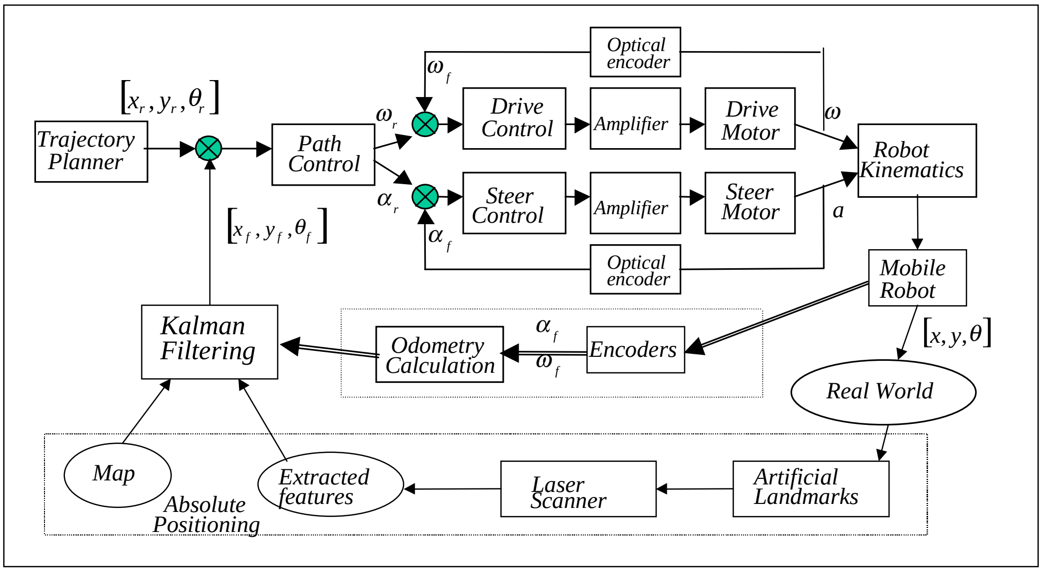


Figure 8 Block diagram of the motion control system

Step 3: Matching

For each measurement, a validation gate is used to decide whether it is a match or not:

$$\mathbf{v}(k+1)\mathbf{S}(k+1)\mathbf{v}^T(k+1) \le \mathbf{G} \qquad (23)$$

If it is true, the current measurement is accepted. Otherwise, it is disregarded.

Step 4: Updating

The filter gain is updated by:

$$\mathbf{W}(k+1) = \mathbf{P}(k+1/k)\nabla\mathbf{h}^T\mathbf{S}^{-1}(k+1) \qquad (24)$$

The robot state is then calculated by:

$$\mathbf{x}(k+1/k+1) = \hat{\mathbf{x}}(k+1/k) + \mathbf{W}(k+1)\mathbf{v}(k+1) \qquad (25)$$

The covariance is updated by:

$$\mathbf{P}(k+1/k+1) = \mathbf{P}(k+1/k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1) \qquad (26)$$

Step 5: **Return** to **Step 1** to recursively implement four steps above.

The proposed algorithm was implemented in a car-like mobile robot that is shown in figure 6. Figure 7 shows more detailed configuration of the robot base.

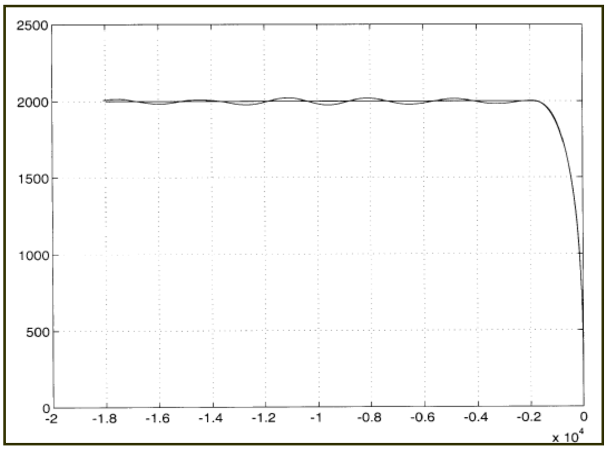
As can be seen, the mobile robot has four wheels with pneumatic tyres. The robot measures 100cm×60cm×120cm (length×width×height) [12]. Two front wheels are driven by a DC motor, serving for steering. Two rear wheels are driven by a DC motor and differential gearbox, serving for traction. Each motor has an incremental optical encoder for close-loop velocity control. Another two incremental encoders (500 counts per revolution) are fixed onto two rear wheels via 3:1 ratio pulley and belts, serving as odometry calculation. The robot has a weight of 100kg and a maximum speed of 1m/s.

Figure 8 shows the block diagram of the motion control system used in our mobile robot. In the outer control loop, the trajectory planner is to generate a continuous-curvature trajectory for the robot to travel [10]. The path controller is to keep the robot on the planned path using a PID control algorithm [9]. It generates the speed and steering demands to two inner loop controllers based on errors between the desired trajectory and the actual trajectory estimated by the EKF. PID control algorithms have been used for maintaining stable speed and steering angle.

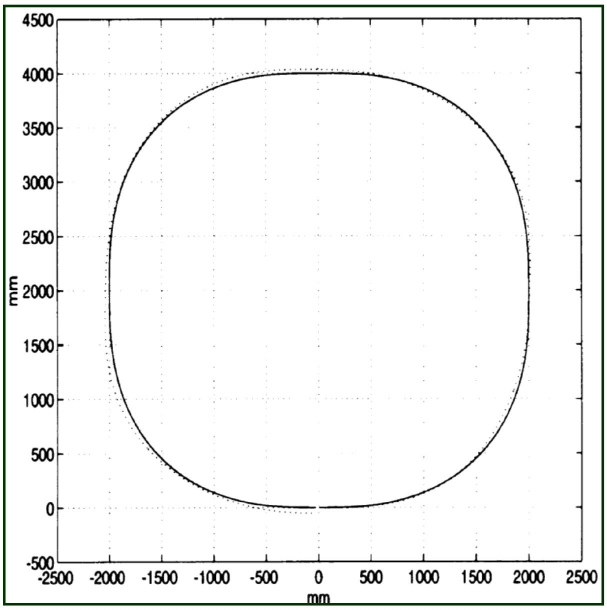
Figure 9(a) presents the result gathered when the mobile robot travels along a straight-line path, then turns 90 degrees. As can be seen, the actual trajectory traveled by the mobile robot is estimated by the EKF, which is slightly deviated from the reference trajectory.

Figure 9(b) shows that the robot traveled along a close-loop route planned by the trajectory planner. Plots from both odometry data and the EKF data look very close to the planned trajectory since the trajectory plotting is scaled down too much. In fact, the odometry data will deviate further away from the desired trajectory if the robot travels along this route continuously.

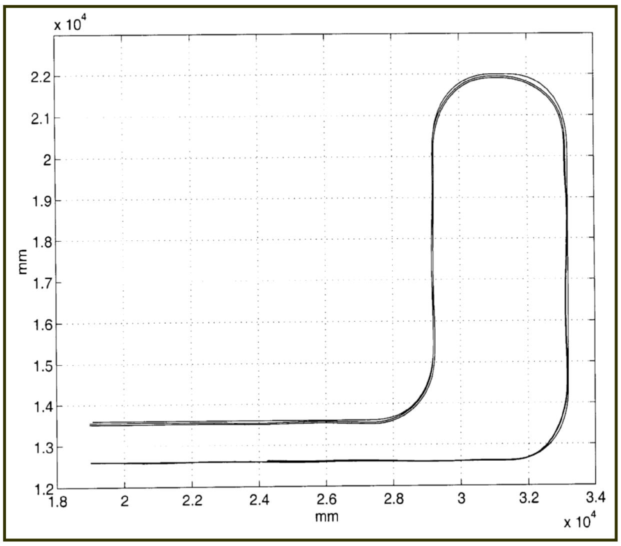
Figure 9(c) shows the results from the experiment that the robot travels along a trajectory consisting of straight lines and curves. Again the trajectory is generated by the global path planner. As you can see, the mobile robot follows the commanded trajectory very well.



(a) A straight-line motion and 90 degree turn



(b) Navigating a close-loop route



(c) Travel along a trajectory with line & arc segments

Figure 9 Experiment results

5. UPDATING LANDMARK MAP

Until now the landmark map we used is built *a priori*, not being updated at all. In other words, the initialization and updating of the robot position is an open-loop navigation process. If some landmarks change their positions or being damaged accidentally, then the navigation system is unable to work well since its internal map doesn't match the landmarks in the environment. At the same time, updating is also necessary in order to register new and natural landmarks into the map if they are to be used in the navigation process.

Map building and maintenance is a dynamic process that involves providing an interpretation of observed sensor information in terms of physical features in the environment. This is the typical correspondence problem or data association problem, which makes perception and sensor-based control difficult. This is because we face both uncertainties: one is in noisy sensors and another in the robot's environment that is dynamically changed.

Figure 10 shows the block diagram of the proposed navigation system that is able to implement concurrent localization and map building automatically. The system consists of three parts:

- an initialization and re-calibration part
- an EKF localization part
- a map-building part.

The initialization and recalibration is implemented by the Kohonen neural networks and the triangulation algorithm, and the EKF navigation part is detailed in section 4. In contrast, the task of the map building is to update and maintain the internal world model of the mobile robot, realizing a feedback loop for the robot navigation. The main purpose of the feedback loop is to verify the correspondence between the abstracted landmarks and the actual landmarks in the robot environment.

To do this, a recursive least square algorithm is adopted to estimate the landmark position during operation of the mobile robot [2]. The key idea is to optimize the internal landmark model during the robot operation and add any new landmark that is consistently detected by the laser scanner and the SICK scanner into the internal world model. The choice of the least square criteria is of course based on assumption that measurement errors are Gaussian. More details will be presented in a forthcoming paper.

6. CONCLUSIONS

This paper addresses the problem of landmark-based navigation for mobile robots to operate in manufacturing industry. A navigation system that locates the robots and updates landmarks in a dynamic environment is proposed. To improve the localization accuracy for mobile robots in a continuous operation, the EKF has been adopted in the navigation system.

Both artificial and natural landmarks are considered in the algorithm in order to provide a useful solution toward real-world applications. A Kohonen neural network based algorithm is proposed in the paper to enable mobile robots to initialize their positions autonomously and re-calibrate their positions in case they get lost. The preliminary experiment results are presented to verify its applicability. The proposed navigation system has potential applications for service robots in home, office and hospitals where there are many useful features and dynamic changes in the environment.

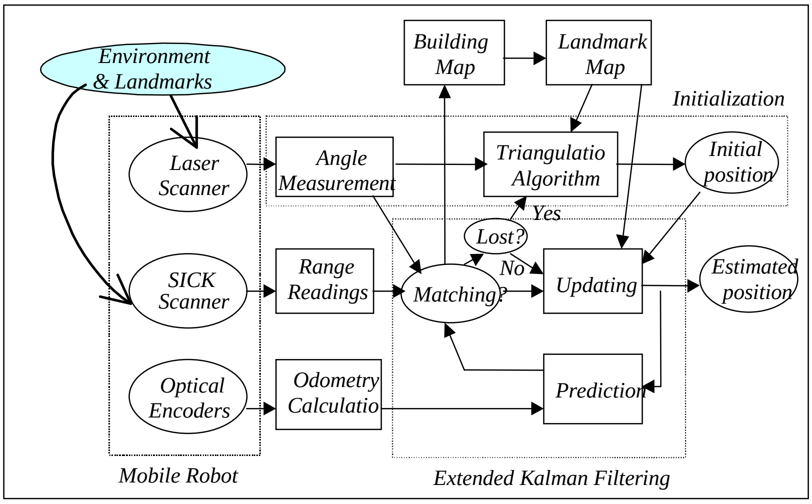


Figure 10 Concurrent localization and map building

Our future research is to pursue further investigation on the Kohonen neural network based algorithm by moving the mobile robot along different operating environments to training the networks. Also, some experiments shall be conducted to testing the concurrent localization and map building process in terms of the addition and eliminating of both artificial and natural landmarks.

REFERENCES

[1] R. Arkin and R. Murphy, Autonomous navigation in a manufacturing environment, IEEE Trans. On Robotics and Automation, Vol. 6, No. 4, pp. 445-454, August 1990

[2] K. Astrom, Automatic Mapmaking, Proc. IFAC Int. Workshop on Intelligent Autonomous Vehicles, pages 181-186, Southampton, 1993

[3] B. Barshan and H.F. Durrant-Whyte, Inertial sensing for mobile robotics, IEEE Transactions on Robotics & Automation, Vol. 11, No 3, 1995

[4] J. Borenstain, The CLAPPER: a dual-drive mobile robot with internal correction of dead-reckoning errors, Proc. of IEEE Int. Conference on Robotics and Automation, pages 3085-3090, 1994

[5] B-Y Chee and S.Y.T. Lang, A random sampling approach to landmark detection for mobile robot localization, Proc. Int. Conf. on Mechatronics’96, Vol. 1, pp. 41-44, Portugal, 18-20 Sept. 1996

[6] H.F. Durrant-Whyte, An autonomous guided vehicle for cargo handling applications, Int. Journal of Robotics Research, Vol. 15, No. 5, 1996

[7] A. Elfes, Sonar-based real-world mapping and navigation, IEEE Journal of Robotics and Automation, RA-3 (3), 1987.

[8] 1. K. Fukushima, S. Miyake, and T. Ito, "Recognition: A neural network model for a mechanism of visual pattern recognition", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13, No.5, pages 826-834, 1983.

[9] D. Gu, H. Hu, M. Brady, F. Li, Navigation System for Autonomous Mobile Robots at Oxford, Proc. Int. Workshop on Recent Advances in Mobile Robots, Leicester, U.K., pp. 24-33, 1-2 July 1997

[10] H. Hu, M. Brady, P. Probert, Trajectory Planning and Optimal Tracking for an Industrial Mobile Robot, Proc. of SPIE's International Symposium on Mobile Robots VIII, Vol. 2058, pp. 152-163, Boston, USA, September 1993

[11] H. Hu and M. Brady, Dynamic Global Planning with Uncertainty for Autonomous Mobile Robots in Manufacturing, IEEE Transactions on Robotics and Automation, Vol. 13, No. 5, pages 760-766, October 1997

[12] H. Hu, D. Gu, M. Brady, A Modular Computing Architecture for Autonomous Robots, in the International Journal of Microprocessors and Microsystems, Vol. 21, No. 6, pages 349-362, March 1998

[13] J.A. Janet, R. Gutierrez-Osuna, T.A. Chase, M. White, R.C. Luo, "Global self localization for autonomous mobile robots using self-organizing Kohonen neural networks", Proc. Int. Conf. on Intelligent Robots & Systems, pp. 5O4-509, 1995

[14] L. Kleeman, Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead reckoning, Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2582-2587, 1992

[15] J. Leonard and H. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation, Vol. 7, 376-382, 1991

[16] C.D. McGillem and T.S. Rappaport, Infrared location system for navigation of autonomous vehicles, Proc. of Int. Conf. on Robotics & Automation, pp. 1236-1238, 1988

[17] T. Skewis and V. Lumesly, Experiments with a mobile robot operating in a cluttered unknown environment, Int. Journal of Robotic Systems, 11(4), pp. 281-300, 1994

[18] S. Thrun and A. Bucken, Integrating grid-based and topological maps for mobile robot navigation, Proc. of the 13th National Conference on AI, Portland Oregon, August 1996