Effective Method for Autonomous Simultaneous Localization and Map Building in Unknown Indoor Environments

Y.L. Ip, A.B. Rad, and Y.K. Wong
Department of Electrical Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon,
Hong Kong

1. Introduction

Many published papers [17, 44] on the map building of Autonomous Mobile Robots (AMRs) do not consider the question of autonomous exploration at all. This is, of course, often just a choice of research focus; effort is expended on the mechanics of map construction from sensor data without worrying about how the sensing positions were selected. Or the map is provided by the operator [7, 11] for any other applications. In our view, the autonomous exploration skill is an extremely important capability for a truly AMR. For example: as it is desired to build a map of unknown environments without human intervention, AMRs should be equipped with a skill of autonomous exploration which includes the competence of path finding, obstacle avoidance and monitor progress towards reaching a goal location or target.

Several possible strategies for exploration of unknown environment are described in the robotics literature. The following categorization is taken from Lee [24]:

1. Human Control – mobile motion is controlled by human operator.
2. Reactive Control – the mobile robot movement is relied on the perception system.
3. Approaching the unknown – the mobile robot move into the region that it knows least in the environment.
4. Optimal search strategies – the approach is focused on to search the shortest path for seeking the goal.

In the first category, the robot is guided around the environment by a human operator. This requires human intervention in the map building process. Therefore, it is not suitable for an autonomous exploration mobile robot.

For reactive exploration approach (2nd category), the sensory data (perception space) is used to calculate or determine the control actions (action space). The sensory data may be the distance information from infrared, sonar or laser range finder type sensors, visual information or processed information obtained after appropriate fusion of multiple sensor outputs. The control actions are usually a change in steering angle and setting a translation velocity of the robot that will avoid collisions with the obstacles on its way and reach the desired target. Pre-designed or adaptive systems based on fuzzy logic [15, 26, 28, 31, 40-41, 45-46, 48-49], neural-networks [9, 33-35, 38, 51] or combination of them [27] are designed by this reactive navigation

Source: Mobile Robots: Perception & Navigation, Book edited by: Sascha Kolski, ISBN 3-86611-283-1, pp. 704, February 2007, Plv/ARS, Germany

approach. Since a totally reactive behavior uses only locally available environment information, without any memory of the previously encountered situations, the autonomous robots are found to suffer from local minima situations. For that reason, a reactive approach cannot be applied to autonomous exploration robot independently.

In the strategy of approaching the unknown (3rd category), a mobile robot tries to move towards the regions of its environment about which it knows least. A mobile robot uses the perception sensor information to search the new territory and move towards that area. This process is repeated until the whole environment has been covered. Global grid model is used in some of map building system to represent the environment. Thrun [44] developed a system trained by an artificial neural network to translate neighboring groups of sonar readings onto occupancy values in the grid and then control the mobile robot to explore directed towards to an areas of high uncertainty in global grid map. However, this system required offline training by the robot simulator though neural network and the algorithm depended on the assumption that environment was rectilinear. In terms of topological maps approach for exploration, Edlinger and Weiss [12] developed a robot equipped with laser range finder to detect obstacle-free segments from the scans and it created topological relations between those scans. A similar approach proposed by Yamauchi [50] required an accurate laser range finder sensing to detect the open space. Recently, Duckett [10] proposed an exploration system to build a topological map which is augmented with metric information concerning the distance and angles between connected places. A trained neural network was used to detect an open space in the environment via sonar sensors and infrared sensor. The open space areas were added to a stack of unexplored locations which were visited in turn until the whole environment had been covered by the robot. This system was tested successfully in a middle-scale indoor environment with Nomad 200 mobile robot [36]. However, this approach relies on a set of sonar sensors and infrared sensors mounted on the rotating robot's turret (which can be rotated independently relative to the base of the robot). It needs to stop the robot on every 1m place in environment to scan and search the possible areas of uncharted territory in all directions. This means that the system would work with Nomad 200 mobile robot only and not suitable for the mobile robot which without rotating turret.

For the fourth category (Optimal search strategies), many researchers have provided mathematical analyses of strategies which are minimized the length of the path traveled by the robot during exploration. It is similar to the well-known traveling salesman problem [14].

In the last two decades, many researchers proposed robust and successful reactive navigation controller, such as behavior-based method [32, 40, 45, 49] and model-based method [2, 21-22, 30]. However, while reactive navigation approaches are often very robust, they cannot be guarantee to navigate all areas in an unknown environment. Therefore, the approaches based on reactive control (2nd category) and approaching the unknown (3rd category) would seem the most promising for autonomous exploration via mobile robot. Therefore, a novel mixing approach combined the reactive approach with approaching the unknown strategy will be presented in this paper.

Our approach is closest in spirit to that of Edlinger and Weiss [12] and Duckett [10], though it does not require an accurate laser range finder for perception system, a rotating robot's turret and a set of training for setting the open space detection neural network system. A simple and real-time system is designed for detecting an open free space via a Bayesian update theory [1] instead of pre-trained system or accurate sensing system. Reactive navigation scheme is applied to start the exploration by using a predefined Hierarchical Fused Fuzzy System (HFFS) [4, 20, 23]. Those proposed algorithm would generate a metric topological map model after the exploration.

In studying the problem of feature based SLAM, a number of specific problems should be considered. These include feature extraction, data association, map management and computation complexity.

In recent times, a number of research groups have attempted to implement real-time SLAM approach successfully with SICK laser scanner [3, 8, 16, 29] in indoor environments. The main advantage of the SICK laser scanner is that the sensor measurements from one robot position can be directly correlated to measurements taken from a nearby position. In contrast, the sonar sensor measurements are usually too noisy, ambiguous and spurious and hence it is difficult to apply the above technique to work properly. Also, it is a common belief that mobile robot navigation and mapping in indoor environments is far more difficult with sonar than with laser measurements. An alternative methodology [5, 6] to overcome limitations of the sonar sensor is to develop advanced custom-made sonar arrays that allow extracting and initializing geometric features from a single robot position. Most recently, some researchers [25, 43] have addressed this issue with a ring of sonar sensors successfully. Tardos et al. (2002) [43] proposed a new technique for perceptual feature extraction technique by using Hough Transform and map joining technique with two statistically independent and uncorrelated maps. On the other hand, Leonard et al. (2002) [25] proposed a feature initialization technique from multiple uncertain vantage points and focused on its state estimation aspects. Both these two algorithms were successfully applied to solve the SLAM problem with 24 sonar sensors in a circular array. The SLAM problem for non-circular or restricted sonar sensor array has not been addressed. This particular problem is addressed by the authors' pervious work [19].

The rest of this chapter is organized as follows: Section 2 states an overall framework of the proposed SLAM algorithm is described. Section 3 presents the complete implementation of the proposed autonomous exploration and mapping system. The performance of the proposed algorithm will be tested via robot simulator and physical mobile robot Pioneer 2DX in Section 4. Finally, the chapter will be concluded in Section 5.

2. Overall framework of the proposed SLAM with feature tracking algorithm

The authors' pervious proposed SLAM algorithm [19] which incorporate with conventional SLAM (which is introduced in appendix), enhanced adaptive fuzzy clustering EAFC feature extraction scheme [17], overlapping comparison technique and feature tracking scheme. The overall framework of this novel SLAM algorithm with feature tracking scheme is shown in the Figure 1. A step-by-step procedure for updating the estimated robot pose and map feature are presented as follows:

1. Obtain the control action and sensor measurements from the odometric sensor and sonar sensor respectively.
2. Store the sensor measurements in Overlapping sliding window sonar buffer.
3. Use the control action (odometric measurement) and present estimated robot pose to perform a SLAM prediction process (it is calculated by equation (A.3) to (A.5)).
4. Check the buffer is full (go to step 5). If the buffer is not empty (go to step 7).
5. To extract the two overlapping segments in the different group of buffer and calculate the comparison values.
6. To perform a "Feature Initialization" process (the map state matrix $X_m(k|k)$ and covariance matrix $P(k|k)$ are updated by equation (A.11) and (A.12), respectively) if the related overlapping segments comparison values are satisfied.

7. To perform a "Data Association" process to calculate sensor measurement prediction value. If the sensors can match with the existing feature in the map, go to step 10. Otherwise, go to next step.
8. Feature Tracking Scheme: Extend the current tracking feature in the map and repeat the "Data Association" again if the related comparison values are satisfied.
9. If the related sensor is matched, store the extended feature in the map. If not match, just forgo it.
10. Use the prediction value obtained in step 3 to perform the SLAM updating process (it is calculated by equation (A.6) to (A.10)).
11. The estimated robot pose and map are updated and their covariance as well. Go back to step 1.

3. Autonomous exploration strategy

In the proposed exploration system, two core components are required: a reactive navigation scheme by using hierarchical fused fuzzy system (section 3.1) and a navigation point generation system (section 3.2). When the mobile robot starts to navigate the environment, a point-mark will be placed on the acquired map sequentially. The map can be learned by either one of standard segment-based map building technique [13, 17, 43].

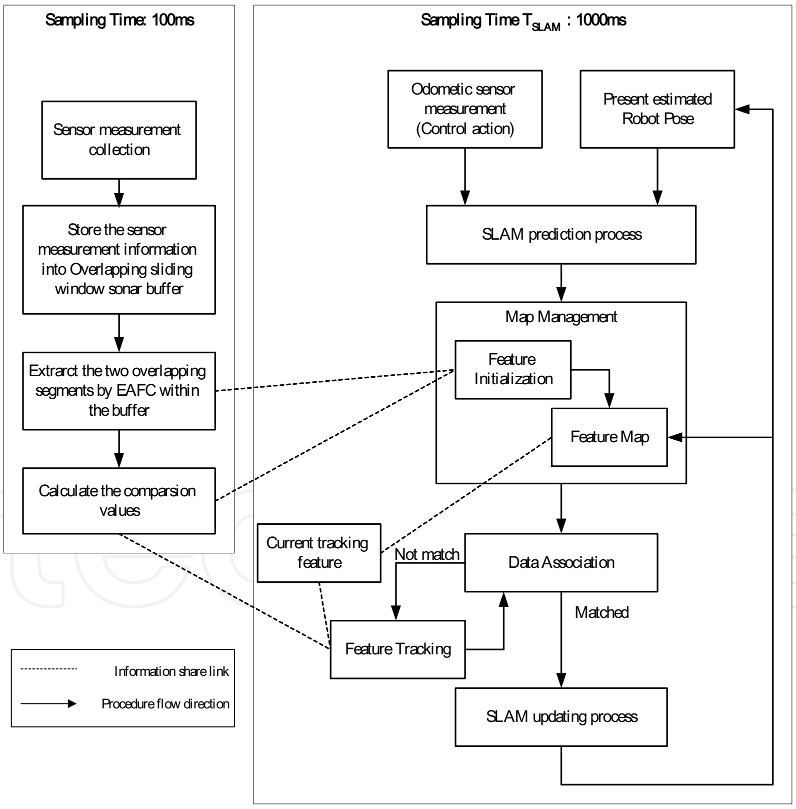

Sampling Time: 100ms
Sensor measurement collection
Store the sensor measurement information into Overlapping sliding window sonar buffer
Extranct the two overlapping segments by EAFC within the buffer
Calculate the comparsion values
Sampling Time T_SLAM : 1000ms
Odometic sensor measurement (Control action)
Present estimated Robot Pose
SLAM prediction process
Map Management
Feature Initialization
Feature Map
Current tracking feature
Not match
Data Association
Matched
Feature Tracking
SLAM updating process
Information share link
Procedure flow direction

Fig. 1. The overall framework of the proposed SLAM.

The moving actions are relied on the reactive navigation system to control the mobile robot to navigate in the environment by wall following and collision avoiding technique until the mobile robot travels back to the traveled point-mark. On every point-mark, the open space evaluation system examines the eight given directions (45° apart each direction) for detecting possible areas of uncharted. An open space probability value is assigned and stored in each given direction at each traveled point-mark. In addition, the acquired map will be used to update if the open space is free or not. If the open space in a given direction at a traveled point-mark is occupied by a segment (map model), then this direction will be stated as not open space, otherwise it is registered as open space (the corresponding variables are described in section 2.2). When the mobile robot travels back to the traveled point-mark, subsequent movements by the mobile robot is required to validate whether the open space in a given direction at each point-mark actually free or not. Therefore, "A*" heuristic search algorithm [36] is used for planning routines in all traveled point-marks. This process guarantees finding the shortest path to the target location (which is the traveled point-mark still contain an open space in those given direction) from all the other traveled point-marks. Note, when the shortest path is generated by a path planning algorithm, the robot's heading is steered by the current robot direction to the next node on the path to the target location with constant velocity at a fixed sampling time. If the given direction at the traveled point-marked is stated as open space, a reactive navigation scheme will be activated again to explore in this given direction. The whole process is repeated until all the open space in a given direction at each traveled point-mark in the map are traveled or cleared by the robot or map feature, respectively.

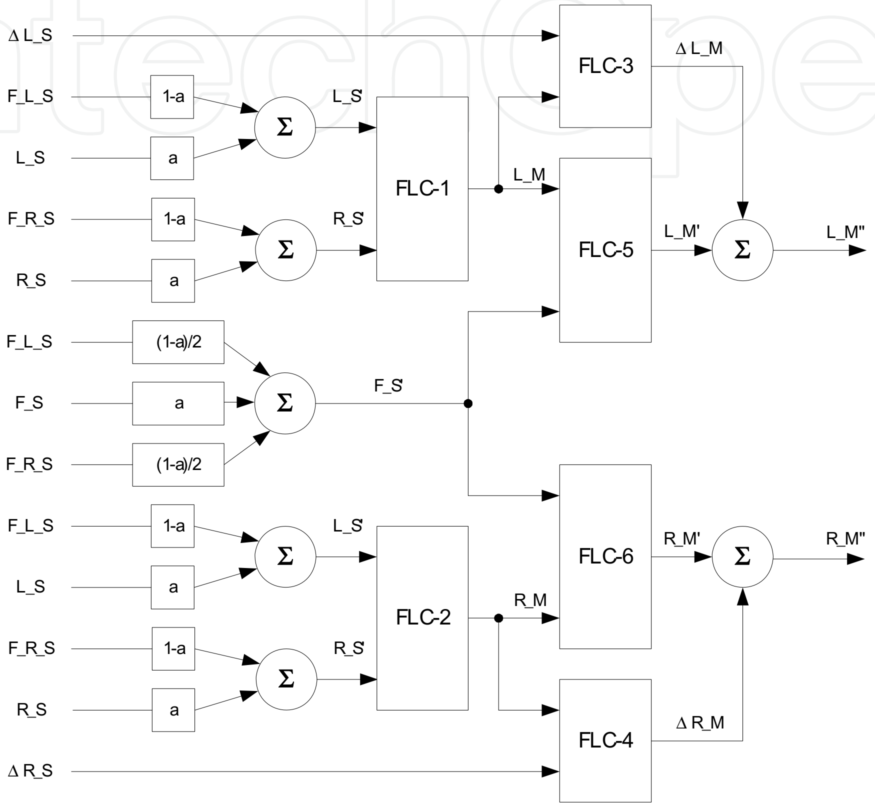
3.1 Reactive navigation via HFFS

HFFS [4, 20, 23] uses smaller number of rules to represent the same amount knowledge, have higher mutually related interactions due to their cross-coupling between each element and level. Due to these properties, this structure is applied to design a reactive navigation controller to control the mobile robot to achieve some task, such as: keep off obstacles and wall or corridor following.

The schematic diagram of the HFFS is shown in Figure 2. In this system, six 2-input / 1-output fuzzy system are used. This HFFS consist 7 inputs and 2 outputs and the total number of rules are 60 only.

The strategy of creating the fuzzy rules is that decision should try to move forward, navigate along a corridor and at corner, keep off and parallel to wall and avoid obstacle. The antecedent variables to the HFFS are 5 sides' sonar sensors reading (i.e. left and right side, left and right front corner and front side) and 2 changes of side sensor readings (left and right).

The input sensor readings are fuzzified using the fuzzy set definitions as shown in Figure 3a. The variable is partitioned in three fuzzy sets namely, VN (very near), NR (near) and FR (far). The sensor readings are normalized between 0 and 1. The input membership function for the two changes of side sensor readings is shown in Figure 3b. Furthermore, the variable is partitioned in three fuzzy sets namely, N (negative), Z (zero) and P (positive). The change of side sensor reading is normalized between –1 and 1. The input L_M and R_M inside the HFFS are fuzzified using the fuzzy membership functions as shown in Figure 3c and they are normalized between –1 and 1. The labels of each subset are N (negative), Z (zero), VS (very slow), M (medium) and F (fast). Since normalized input variables are used. Four input scaling factors should be used, such as L_S' and R_S' sensor reading scaling factor σ_S, F_S' sensor reading scaling factor σ_F, change of side sensor reading (ΔL_S and ΔR_S) scaling factor $\sigma_{\Delta S}$ and L_M and R_M velocity scaling factor σ_V.

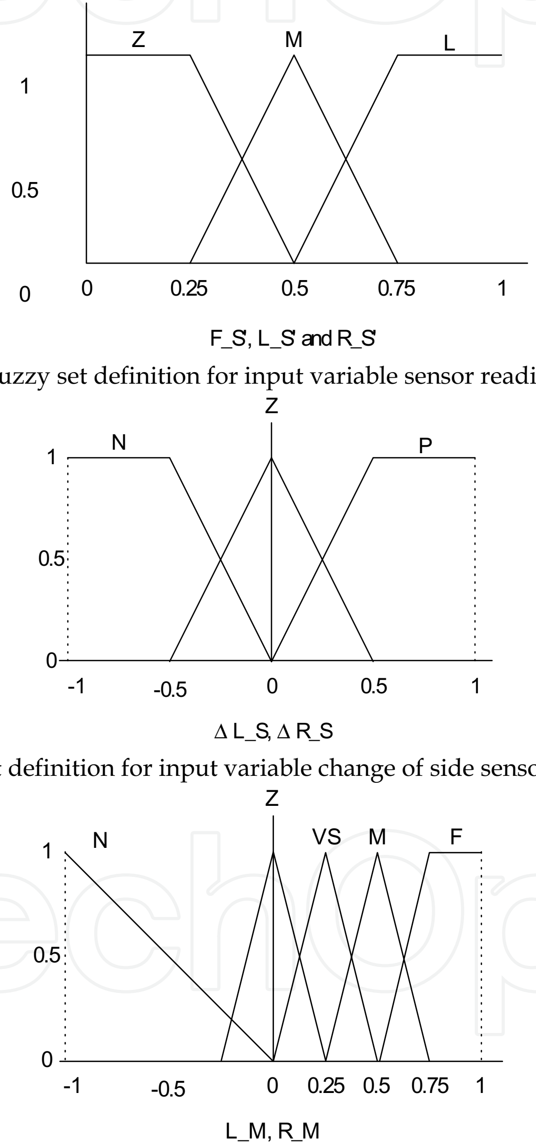


where a is weighting factor for fusion the sensor readings (0.8 in these experiments),
L_S is left side sensor reading,
R_S is right side sensor reading,
F_S is front side sensor reading,
F_L_S is front-left corner sensor reading,
F_R_S is front-right corner sensor reading,
ΔL_S is the change of left side sensor reading, i.e. ΔL_S = (L_S(t)-L_S(t-1)),
ΔR_S is the change of right side sensor reading, i.e. ΔR_S = (R_S(t)-R_S(t-1))
L_M" is the output velocity for left motor and
R_M" is the output velocity for right motor.

Fig. 2. The HFFS reactive navigation structure.

The singleton fuzzy set is used for the all output variables (L_M, L_M′, R_M, R_M″, ΔL_M and ΔR_M) in HFFS and it is shown in Figure 4. The fuzzy partition names for L_M, L_M′, R_M and R_M″ are same as that used in input L_M or R_M. The normalized ΔL_M and

ΔR_M are partitioned into five fuzzy sets, namely NL (negative large), NS (negative small), Z (zero), PS (positive small) and PL (positive large). All the output variables are normalized between –1 and 1. The output scaling factor for the motor velocity L_M′ & R_M′ and the change of motor velocity ΔL_M & ΔR_M are stated as σ_V (same as the input scaling factor used in L_M and R_M) and $\sigma_{\Delta V}$, respectively. In the present implementation the center average de-fuzzifier [47] is used for its fast computation.

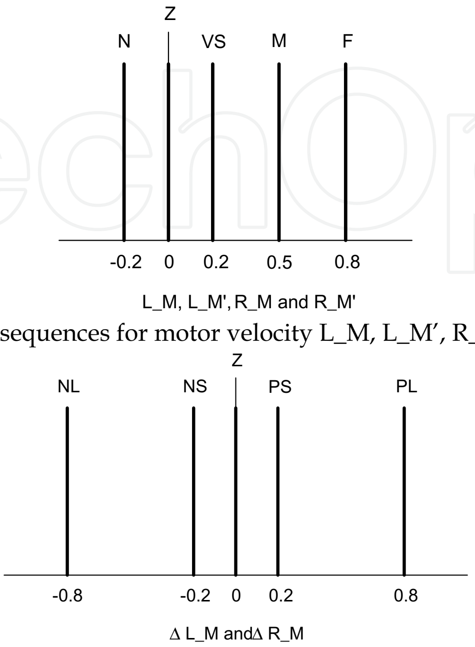


a) Fuzzy set definition for input variable sensor reading.

b) Fuzzy set definition for input variable change of side sensor readings.

c) Fuzzy set definition for input variable motor velocity.

Figure 3 Input membership functions corresponding to HFFS.



a) Output consequences for motor velocity L_M, L_M′, R_M and R_M′.

b) Output consequences for the change of motor velocity ΔL_M and ΔR_M .

Fig. 4. Output consequences corresponding to the HFFS.

The corresponding fuzzy rule tables used in HFFS are shown in Table 1. From the rule table (Table 1c), we defined that the robot will turn right when both sides' sensor readings are equal or a nearer obstacle occupied in left side.

O/P: L_M / R_M		R_S'		
		VN	NR	FR
L_S'	VN	Z / Z	VS / Z	M / N
	NR	Z / VS	M / M	F / M
	FR	N / M	M / F	F / F

a) Fuzzy rule base for FLC-1 and FLC-2 in HFFS.

O/P: ΔL_M or ΔR_M		L_M or R_M				
		N	Z	VS	M	F
ΔL_S	N	X	X	X	PS	PL
or	Z	X	X	X	Z	Z
ΔR_S	P	X	X	X	NS	NL

b) Fuzzy rule base for FLC-3 and FLC_4 in HFFS.

O/P: L_M' / R_M'		L_M / R_M				
		N	Z	VS	M	F
F_S'	VN	N	N	Z	A_1 / B_1	A_1 / B_1
	NR	N	VS	Z	A_2 / B_2	A_2 / B_2
	FR	N	VS	VS	M	F

c) Fuzzy rule base for FLC-5 and FLC_6 in HFFS.

Table 1. Fuzzy rule tables for the proposed HFFS reactive navigation controller.

3.2 Navigation point generation system

The objective of this system is to place and store a point-mark on the traveled robot trajectory and hence to search all possible open space on the mapping environment. Before we describe the system, few types of points and terms are introduced in Table 2. The main feature of this navigation point generation system is to arrange the navigation point in unevenly distribution. Different to the navigation system proposed by Duckett [10]. Duckett [10] suggests adding a "place" (equivalent to navigation point used in here) at every certain distance (1 m is used in their experiment). Therefore, the resulting topological path was distributed evenly and some of redundant "place" or navigation point was occurred. In contrast, we suggest adding a confirmed navigation point in a required region. For example: if the mobile robot navigates in a long corridor, few navigation points are required to represent the free path. Therefore, 8 possible free space directions and its related free space probability are assigned at each confirmed navigation point. We can compare their state (8 free space probability) of navigation point to past navigation point to verify their similarity when the mobile robot navigated in an unknown environment. The calculation of free space probability in a given direction at each navigation point is discussed in section 3.2.3

Symbol	**Definition**
T_NPn	n^{th} Test Navigation Point
P_NP	Potential Navigation Point
Cd_NP	Confirmed Navigation Point
L_Cd_NP	Last Cd_NP
N_Cd_NP	The Nearest Cd_NP to the current robot location
T_Cd_NP[i] where i = 0, 1, 2, 3	ith recent Traveled Cd_NP (The most recent Cd_NP is represented by i = 0)
E_Cd_NP	The Extra Past Cd_NP, i.e. At time k E_Cd_NP ≠T_Cd_NP[i] for i = 0, 1, 2, 3.
FreeSpaceCoverRadius	The maximum free space coverage radius
MinNavTravelDis	The minimum specified traveled displacement

Table 2. The Various symbols in Navigation point generation system.

In addition, three terms will be attached in each given direction at each Cd_NP, such as "FreeSpaceProbability (= 0 ~ 1)", "IsOpenSpace (= True or False)" and "IsExplored (= True or False)".

Figure 5 is a flow chart of the entire navigation point generation system. It is designed to allow direct translation into an implementation. In Figure 5, two types of line are used, i.e. dashed-line and solid-line. The dashed-line stated that the process should go to next step within same sampling interval. In contrast, the solid-line stated that the next step would be executed at next sampling interval. The flow of the algorithm is regulated by the state variables described in Table 2. The significance of some parts of the flow chart necessitates discussion. These regions have been labeled in Figure 4 and are described as following:

A The state of the last confirmed navigation point (L_Cd_NP) is updated by 10 sampling robot steps before it go to next step. The state of Cd_NP updating process will be discussed in section 3.2.1.

B When Condition 1 (which stated in Figure 4) is satisfied, a new confirmed navigation point (Cd_NP) is registered and reset the variable *n* (which states the total number of test navigation point) to 1. And then calculate and update the state of this new registered Cd_NP in next sampling interval.

C When the current test navigation point (T_NPn) is similar to the extra past confirmed navigation point (E_Cd_NP), then reset the current test navigation point and set *n* to 1. After that, register a new test navigation point in next sampling interval.

D If the current test navigation point is not similar to the current potential navigation point, then search the nearest confirmed navigation point (Cd_NP) in the list relative to the current potential navigation point. If this resulting point is similar to the current potential point and their distance is smaller than "MinNavTravelDis", then reset the current test navigation point and set *n* to 1.

3.2.1 Process for updating the state of confirmed navigation point (Cd_NP)

As mentioned that as before, three variables are attached in each given direction of each confirmed navigation point (Cd_NP). One floating point variable, i.e. "FreeSpaceProbability". The updating process of this variable will be introduced in section 3.2.3. Two Boolean variables, i.e. "IsOpenSpace" and "IsExplored", and the updating condition are discussed as follows:

"IsOpenSpace" This Boolean variable can be stated that the given direction of confirmed navigation point is open or not. And it can be determined by the "FreeSpaceProbability". If the free space probability is more than or equal to 0.5, then this given direction is open (IsOpenSpace = True). Otherwise, "IsOpenSpace" is equal to "False". On the other hand, this Boolean variable can be also updated by the acquired map model. Since a segment-based map is extracted by either one of segment-based map building technique simultaneously. (The corresponding map building will be discussed in section 4.) A simple logical updating algorithm is applied. If the given direction of confirmed navigation point is occupied by a segment, then the Boolean variable can be stated as "False". Otherwise, it is registered as "True".

"IsExplored" This Boolean variable states that the given direction of confirmed navigation point is explored or not. Three conditions are used to update this variable and are shown as following:

1. If the distance between two confirmed navigation points is smaller than FreeSpaceCoverRadius, then these Boolean variables (in the corresponding given direction at the two confirmed navigation points are registered as "True" (i.e. IsExpored = True).
2. If the distance between the current robot position and the corresponding confirmed navigation point is smaller than FreeSpaceCoverRadius, then it is stated as "True".
3. If the term "IsOpenSpace" is stated as "False" in a given direction at the corresponding confirmed navigation point, then the term "IsExplored" is equal to True in the same direction at the corresponding confirmed navigation point.

3.2.2 Redundant confirmed navigation point removing process

As we want to reduce the redundant confirmed navigation point in the map, a redundant point removing process is needed. In this process, we compare the T_Cd_NP[0] and T_Cd_NP[1]. If it is similar and the distance between T_Cd_NP[0] and T_Cd_NP[2] is smaller than FreeSpaceCoverRadius , then remove the T_Cd_NP[1]. The visualization of the proposed exploration algorithm is shown in Figure 6.

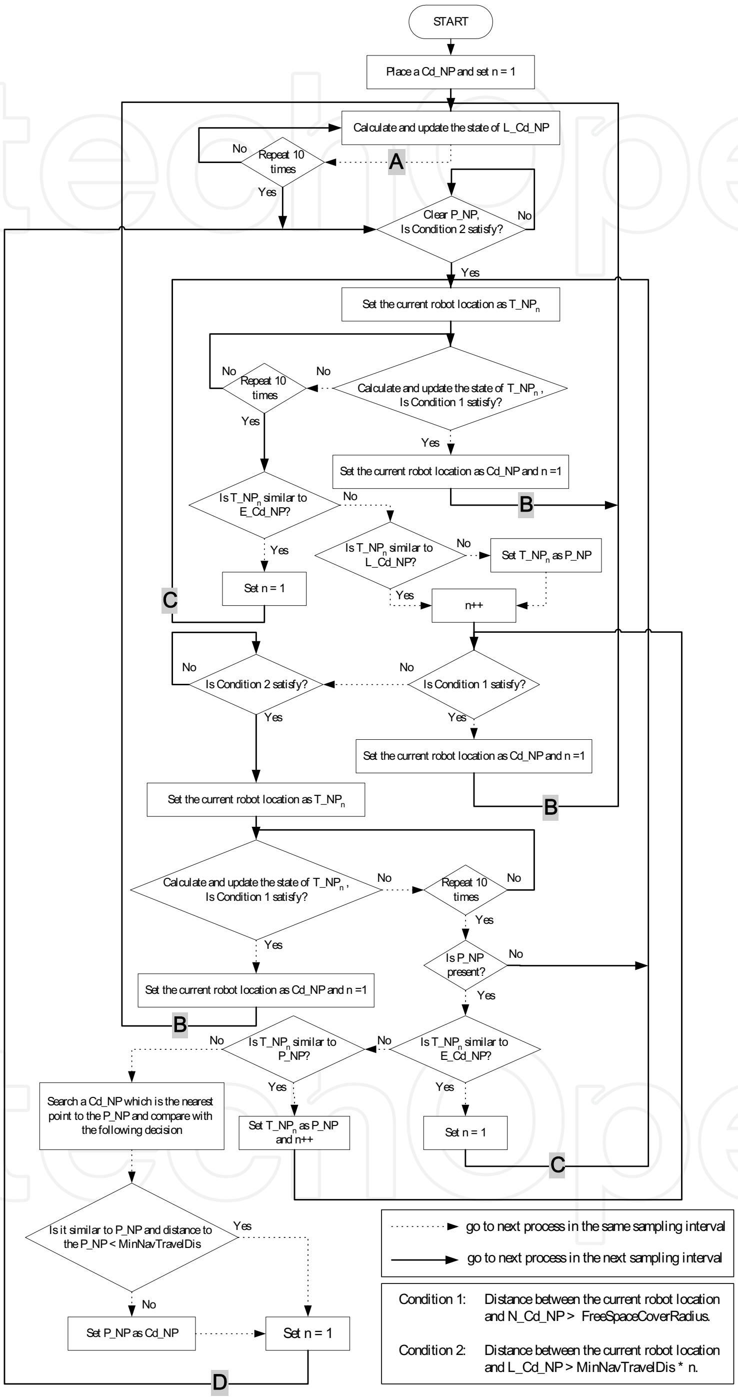

START
Place a Cd_NP and set n = 1
Calculate and update the state of L_Cd_NP
No
Repeat 10 times
A
Yes
Clear P_NP, Is Condition 2 satisfy?
No
Yes
Set the current robot location as T_NP_n
No
Repeat 10 times
No
Calculate and update the state of T_NP_n, Is Condition 1 satisfy?
Yes
Yes
Set the current robot location as Cd_NP and n =1
B
Is T_NP_n similar to E_Cd_NP?
No
Is T_NP_n similar to L_Cd_NP?
No
Set T_NP_n as P_NP
Yes
Set n = 1
C
Yes
n++
No
Is Condition 2 satisfy?
No
Is Condition 1 satisfy?
Yes
Yes
Set the current robot location as Cd_NP and n =1
B
Set the current robot location as T_NP_n
Calculate and update the state of T_NP_n, Is Condition 1 satisfy?
No
Repeat 10 times
No
Yes
Yes
Is P_NP present?
No
Set the current robot location as Cd_NP and n =1
Yes
B
No
Is T_NP_n similar to P_NP?
No
Is T_NP_n similar to E_Cd_NP?
Search a Cd_NP which is the nearest point to the P_NP and compare with the following decision
Yes
Set T_NP_n as P_NP and n++
Yes
Set n = 1
C
Is it similar to P_NP and distance to the P_NP < MinNavTravelDis
Yes
No
Set P_NP as Cd_NP
Set n = 1
D
go to next process in the same sampling interval
go to next process in the next sampling interval
Condition 1: Distance between the current robot location and N_Cd_NP > FreeSpaceCoverRadius.
Condition 2: Distance between the current robot location and L_Cd_NP > MinNavTravelDis * n.

Fig. 5. The flow chart of the proposed navigation point generation system.

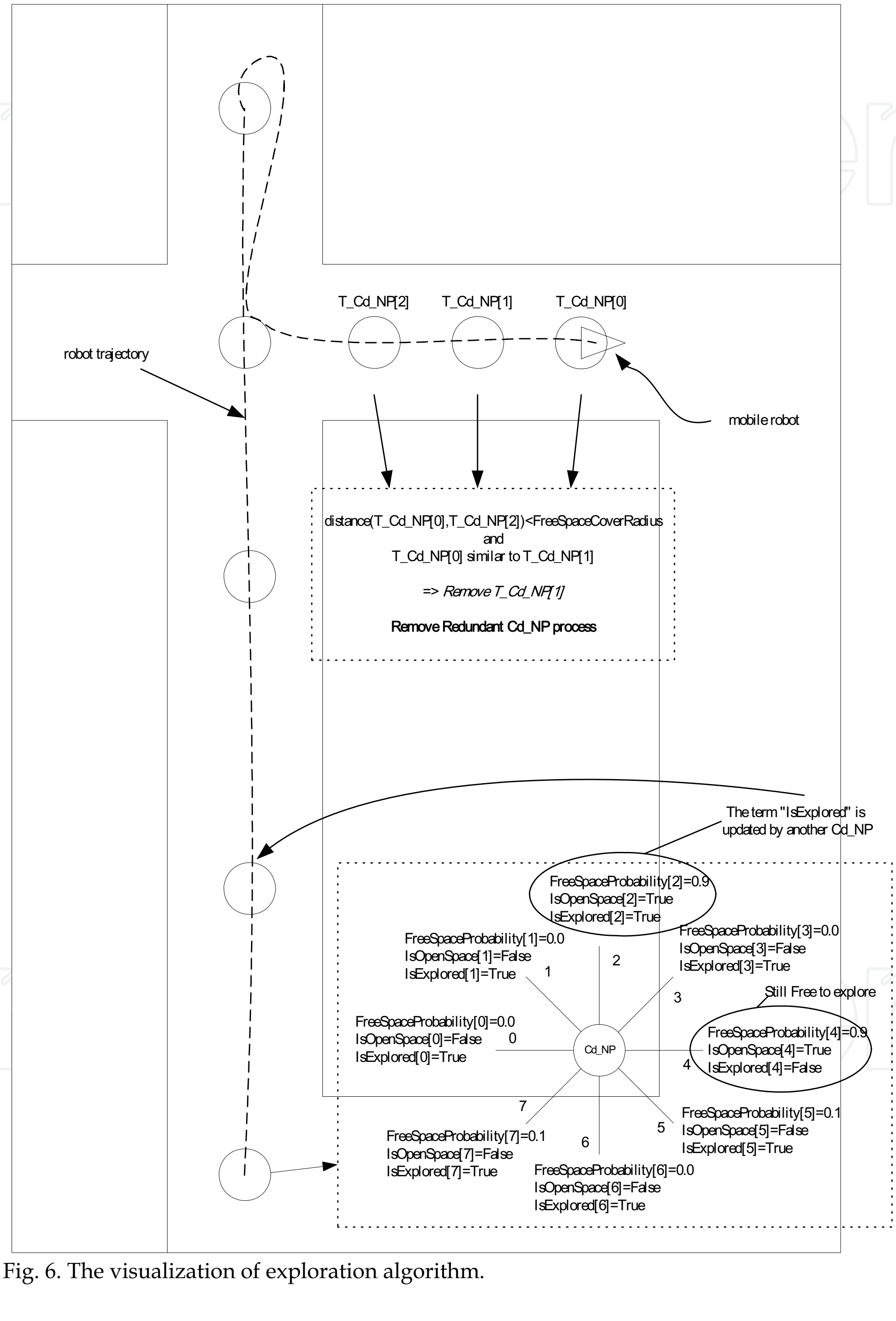


Fig. 6. The visualization of exploration algorithm.

3.2.3 Open space evaluation system

To evaluate areas of free space, we have developed a simple and efficient mechanism which uses a Bayesian update rule and sonar sensor model *pdf* to calculate the value of "FreeSpaceProbability" in a given direction at each point-mark (T_NP or Cd_NP). From equation (A-2), the sonar probability model $p(r \mid z, \theta)$ can be calculated. r is the sensor range reading. z represents the true parameter space range value (here we use the value of "FreeSpaceCoverRadius"). θ represents the azimuth angle measured with respect to the beam central axis (here we calculated as sonar beam angle θ_S subtract the given direction at a point-mark, all angles are referred to the global reference frame). In this application, the value k_ε and k_o become a constant in equation (A-2). i.e. k_ε = zero or very small real number and k_ε = 0.6. The variance of radial σ_r and angular σ_θ is equal to 250mm and 12.5°, respectively. After a sampling for all sonar sensors (16 sonar sensors used here), the probability is fused together by Bayes's theorem [1]. The value of "FreeSpaceProbability" in given direction is calculated as below:

$$P[NavDir_j = FREE \mid r] = \frac{p[r \mid NavDir_j = FREE]P[NavDir_j = FREE]}{\sum_{NavDir_j} p[r \mid NavDir_j]P[NavDir_j]} \qquad (1)$$

for *j = 0, 1, 2, …, 7.*

where $p[r \mid NavDir_j = FREE]$ corresponds to sensor *pdf* (for calculate the probability of free space, it is calculated as $(1 - p(r \mid z, \theta))$),

$P[NavDir_j = FREE]$ is prior presented free space probability of a given direction at a point-mark $NavDir_j$,

$P[NavDir_j = FREE \mid r]$ is new presented free space probability of $NavDir_j$ based on r observations.

The whole process is repeated until the robot has traveled far away from the point-mark with certain distance (300mm in these experiment). After the open space evaluation process is complete, 8 probability values (see bottom right in Figure 6) are attached in the point-mark (T_NPn or Cd_NP) for representing the free space probability values in given direction. Furthermore, another function for this open space evaluation system is aimed to a mobile robot escaping from a "U-trap" (see the upper part in Figure 6). We can detect a "U-trap" situation from the following 2 conditions:

1. L_S'<600mm and R_S'<600mm
2. An "IsOpenSpace[i]" in point T_Cd_NP[0] is equal FALSE, where i is equal to the direction (0 ~ 7) which is pointed to the current position of mobile robot.

If a "U-trap" is detected, the reactive navigation system will be disabled until the mobile robot is rotated for certain degree (120° in these experiments).

4. Experimental Results

In order to evaluate the performance of the proposed autonomous exploration strategy, several experiments were conducted. In the first two experiments, we applied the proposed reactive navigation system only to control a Pioneer 2DX mobile robot to navigate along a corridor and at corners. For third experiment, the complete autonomous exploration strategy was conducted in a localization error free (no slippage or wheel drift error) mobile robot simulator platform (Pioneer simulator). For fourth experiment, the autonomous SLAM experiment was conducted in a well constructed unknown indoor environment with Pioneer 2DX mobile robot.

All of experiments were implemented in robot software "Saphira" [42] on a Pentium 4 1.6 GHz with 256RAM PC computer and it is communicated to the Pioneer 2DX mobile robot via wireless modem. Also, all sonar sensors measurements were limited up to 3 meters to reduce the uncertainty for map building process. In our proposed exploration scheme, two sides sensors and front sonar sensors array are used only. They are arranged in 5 input groups as shown in Figure 7 (i.e. L_S, L_F_S, F_S, R_F_S and R_S). The 10 sensors were categorized into 5 inputs for HFFS reactive navigation system and it is allocated as follows:

Left side: L_S = min(S_{16},S_1)
Left front corner: L_F_S = min(S_2,S_3)
Front side: F_S = min(S_4,S_5)
Right front corner: R_F_S = min(S_6,S_7)
Right side: R_S = min(S_8,S_9)

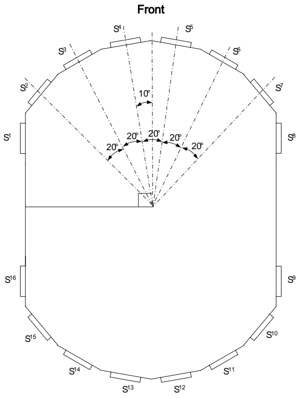


Fig. 7. Sensors arrangement for HFFS reactive navigation system.

The values (range) from direction left to right (i.e. L_S, L_F_S, F_S, R_F_S and R_S) were used as the input of the HFFS reactive navigation system and the outputs of the system were the linear velocity of the left (L_M") and right (R_M") driving motor of the mobile robot. For scaling factor σ_S, a simple adoption scheme was used and it is formulated as follows:

$$\sigma_S(t)=\begin{cases}\sigma_S(t-1)+\lambda(\sigma_{MAX}-\sigma_S(t-1)) & \textit{if } (\mathrm{R\_S}>=\sigma_{MAX})\,\&\,(\mathrm{L\_S}>=\sigma_{MAX})\\ \sigma_S(t-1)+\lambda(\mathrm{L\_S}-\sigma_S(t-1)) & \textit{if } (\mathrm{R\_S}>=\sigma_{MAX})\,\&\,(\mathrm{L\_S}<\sigma_{MAX})\\ \sigma_S(t-1)+\lambda(\mathrm{R\_S}-\sigma_S(t-1)) & \textit{if } (\mathrm{L\_S}>=\sigma_{MAX})\,\&\,(\mathrm{R\_S}<\sigma_{MAX})\\ \sigma_S(t-1)+ & \\ \lambda((\mathrm{R\_S}+\mathrm{L\_S})-\sigma_S(t-1)) & \textit{if } (\mathrm{L\_S}<\sigma_{MAX})\,\&\,(\mathrm{R\_S}<\sigma_{MAX})\end{cases} \qquad (2)$$

where σ_{MAX} is the maximum defined range value and is equal to FreeSpaceCoverRadius and

λ is a forgetting factor (0.7 in these experiments).

The initial value $\sigma_S(0)$ is defined as 1500mm. The setting of reminding input and output scaling factors are stated as follows:

σ_F is defined as FreeSpaceCoverRadius.

$\sigma_{\Delta S}$ is defined as 100mm (designed by human experience).

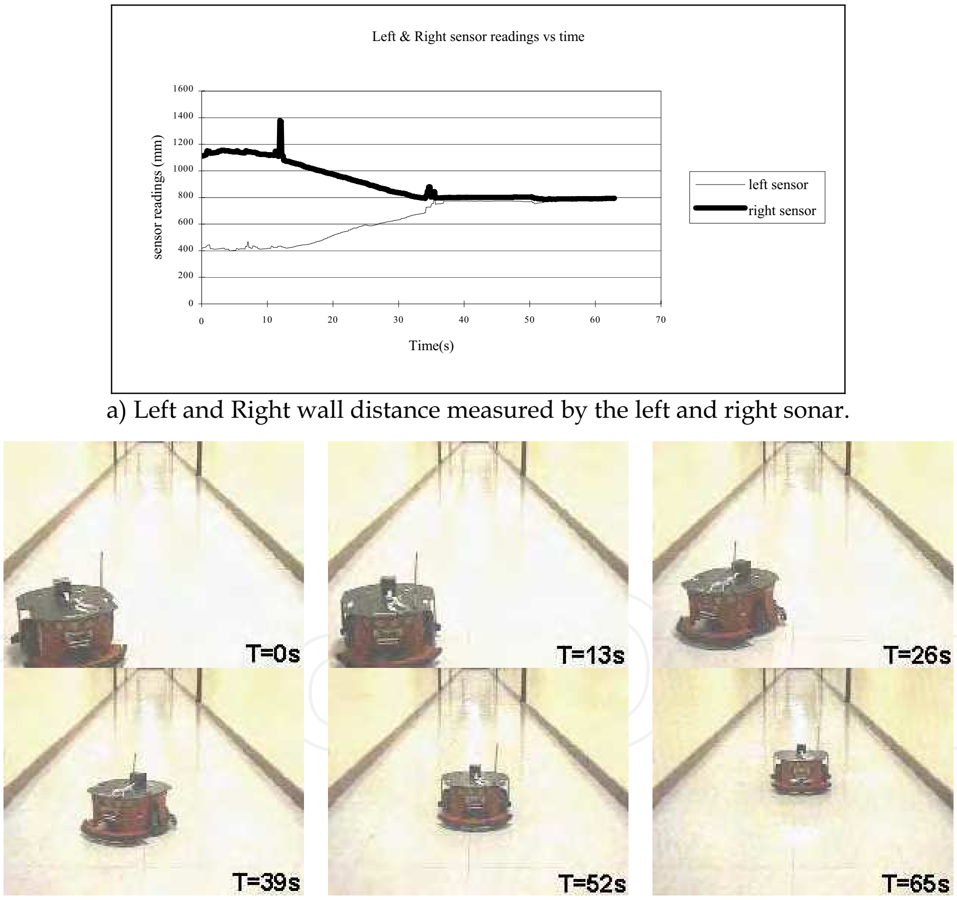
σ_V is defined as 200mm/s (predefined maximum translation velocity).

$\sigma_{\Delta V}$ is defined as 20mm/s (designed by human experience).

The sampling time for the proposed autonomous exploration strategy is 1s in all the following experiments.

4.1 Experiments with a real robot

In the first experiment, the system was tested in a long corridor with 1.5m widths. The objective of this experiment was to verify the performance when a mobile robot navigated along a corridor. Therefore, the minimum range value of the left and right side group sensors are plotted against time and it is shown in Figure 8a. In Figure 8a) and b), shows that the Pioneer 2DX mobile robot navigated along towards the middle of corridor with a smooth trajectory.



a) Left and Right wall distance measured by the left and right sonar.

b) Snap shots of a Pioneer 2DX navigating along a corridor.

Fig. 8. The performance of the proposed HFFS reactive navigation system while navigates along a corridor.

In the second experiment, the proposed reactive navigation system was used to control a Pioneer 2DX navigating in a more complex area where is located at the outside of our research laboratory in the university. Figure 9 shows the robot's information and the robot trajectory during navigation. At starting of the navigation (low bottom left in Figure 9b), the mobile robot traveled along a corridor. Then the mobile robot turned to right side when the robot's front sensor detected an obstacle (at time 70s, see Figure 9a). Then the mobile robot started to follow a set of lockers (by wall following behavior) until it's front sensor detect an obstacle again. Finally, it started to follow right hand side object at time 140s.

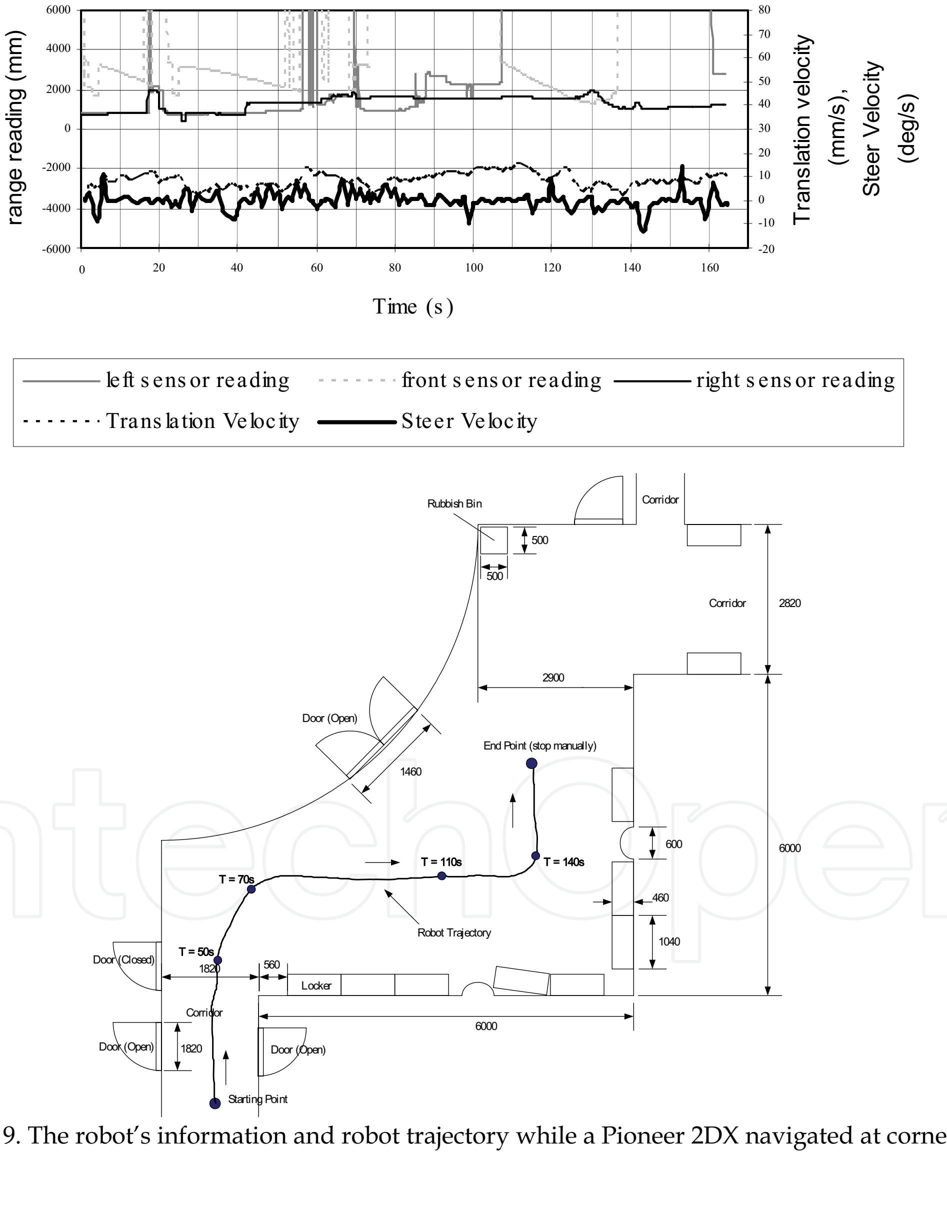


Fig. 9. The robot's information and robot trajectory while a Pioneer 2DX navigated at corner.

From the above two experiments, it can be demonstrated that the proposed HFFS reactive navigation system can achieve the goal of multi-behavior (such as: navigate along a corridor and at corner, keep off and parallel to wall and avoid obstacle) mobile robot controller. In the next experiment, the complete autonomous exploration strategy is applied to control a mobile robot for navigating in an unknown environment via robot simulator.

4.2 Experiment with a robot simulator

In this experiment, the EAFC segment-based map building algorithm [15] was adopted to extract the map information from raw sonar data. This map building algorithm is the authors' pervious work [17]. Other than that algorithm, we can also apply fuzzy sonar maps [13] (which was proposed by Gasos and Martin 1996) or Hough transform with sonar arc (which was proposed by Tardos *et. al.* 2002) for extracting a segment-based map. For the parameters setting in autonomous exploration strategy, it was selected as follow: "FreeSpaceCoverRadius" = 2500mm and "MinNavTravelDis" = 800mm.

The advantage for using a robot simulator to verify our proposed autonomous exploration strategy is that the localization error can be disabled or neglected. Since the localization problem will arise an error or affect the accuracy in the planning process. The Pioneer Simulator [42] can simulate several different types of typical noise that occur during robot navigation and sensor perception. To achieve the goal of this experiment, the percentage of encoder jitter, angle jitter and angle drift in robot simulator is reduced to zero. Nevertheless, the sonar sensor uncertainty is still occurring in the system. Figure 10 shows the navigation point-marks and the unexplored direction at each Cd_NP superposed on the actual map when the Pioneer 2DX navigates in the simulation world. We can see that the mobile robot can navigate in all regions in the unknown environment. Also, the navigation point-marks are distributed unevenly in the navigation environment. The raw sonar data and extracted map by EAFC during the autonomous navigation are shown in Figure 11 a) and b), respectively.

4.3 Autonomous SLAM experiment

In this experiment, the autonomous exploration strategy was combined with the SLAM algorithm [19] to form an effective SLAM algorithm. Basically, this effective SLAM algorithm is similar to the algorithm that was tested in section 4.2 except the map information (for aiming the navigation point generation system) is replaced by the SLAM map. An overview of the system architecture is shown in Figure 12. Since this was a real-time experiment, it was difficult to obtain a ground truth robot trajectory. Therefore, we used the authors' previous proposed fuzzy tuned extended Kalman filter FT-EKF model-based localization algorithm [18] to measure the robot trajectory during the autonomous SLAM process for comparison. The system was tested in our research office (8 × 8 m) and the floor plan.. The total trajectory of the mobile robot was around 30m, lasting around 20 minutes

The sampling rate of SLAM process and autonomous exploration strategy was 1000ms. The parameters settings for the autonomous exploration strategy were selected as: "FreeSpaceCoverRadius" = 2000mm and "MinNavTravelDis" = 700mm.

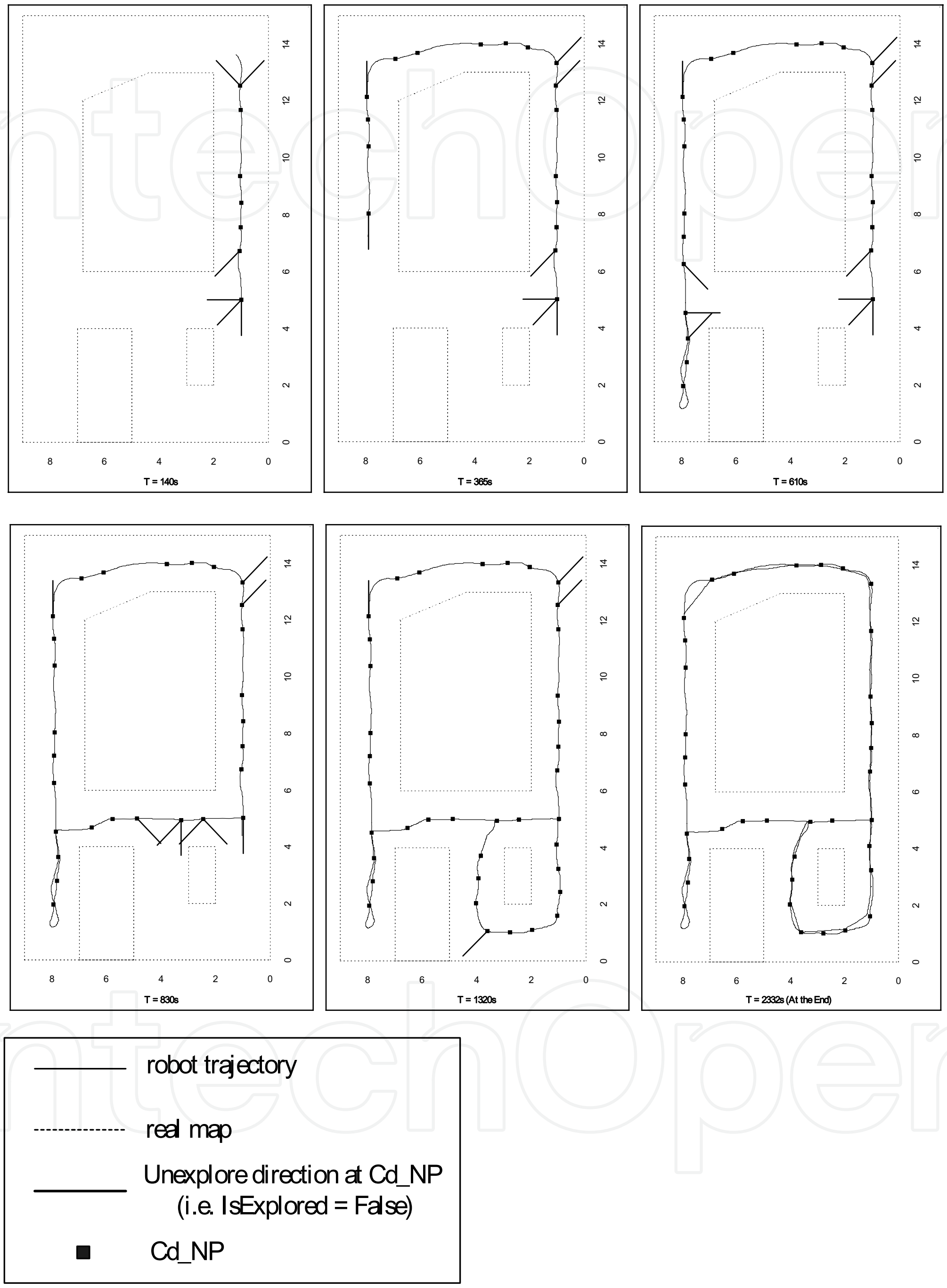
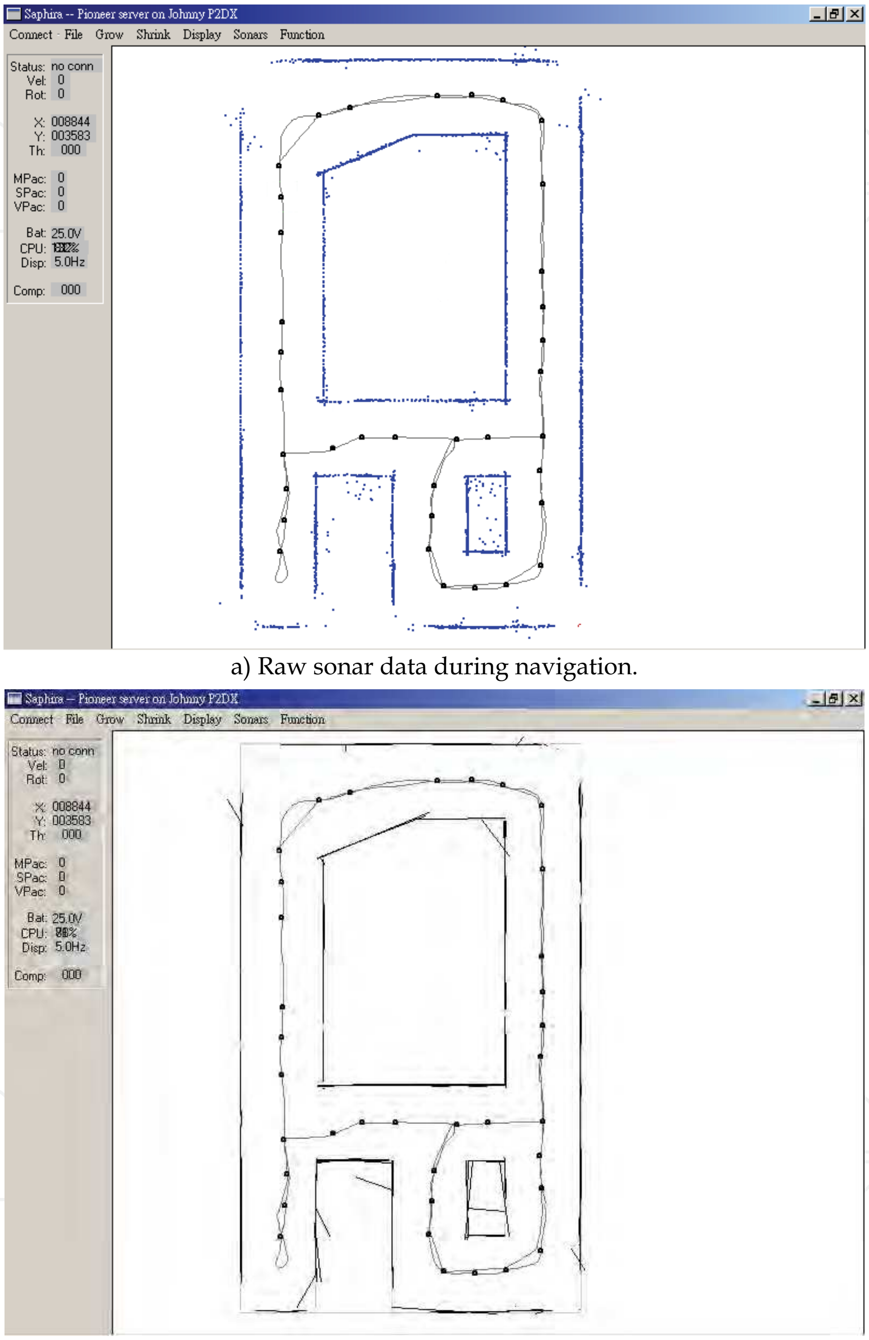


Fig. 10. Snap Shots for the Pioneer 2DX mobile robot navigating in the simulation world.



Status: no conn
Vel: 0
Rot: 0
X: 008844
Y: 003583
Th: 000
MPac: 0
SPac: 0
VPac: 0
Bat: 25.0V
CPU: [illegible]
Disp: 5.0Hz
Comp: 000

a) Raw sonar data during navigation.

Status: no conn
Vel: 0
Rot: 0
X: 008844
Y: 003583
Th: 000
MPac: 0
SPac: 0
VPac: 0
Bat: 25.0V
CPU: [illegible]
Disp: 5.0Hz
Comp: 000

b) (black line) Extracted line segments superposed on (gray line) real map.

Fig. 11. Robot trajectory, navigation point-marks, extracted map, raw data and real map captured from the robot software Saphira.

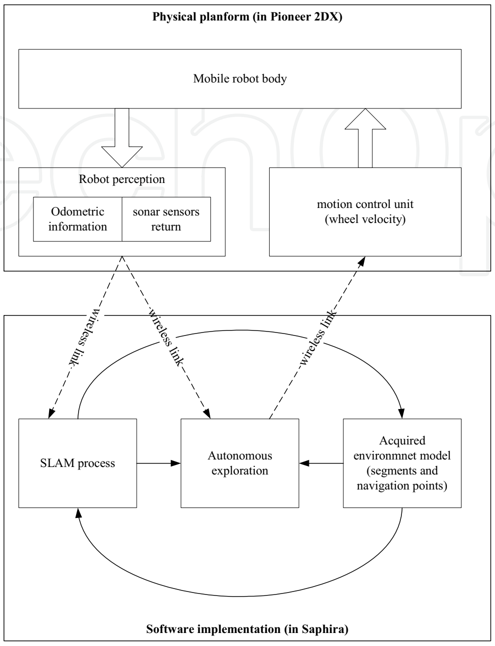
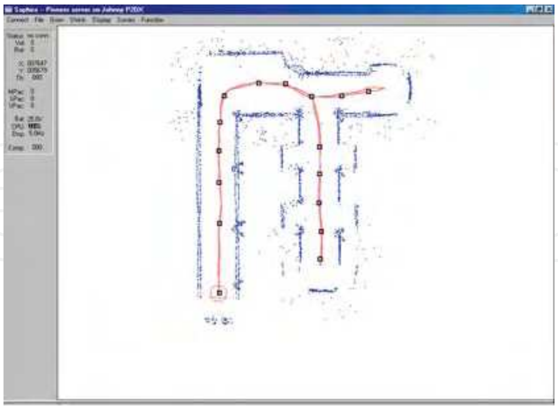
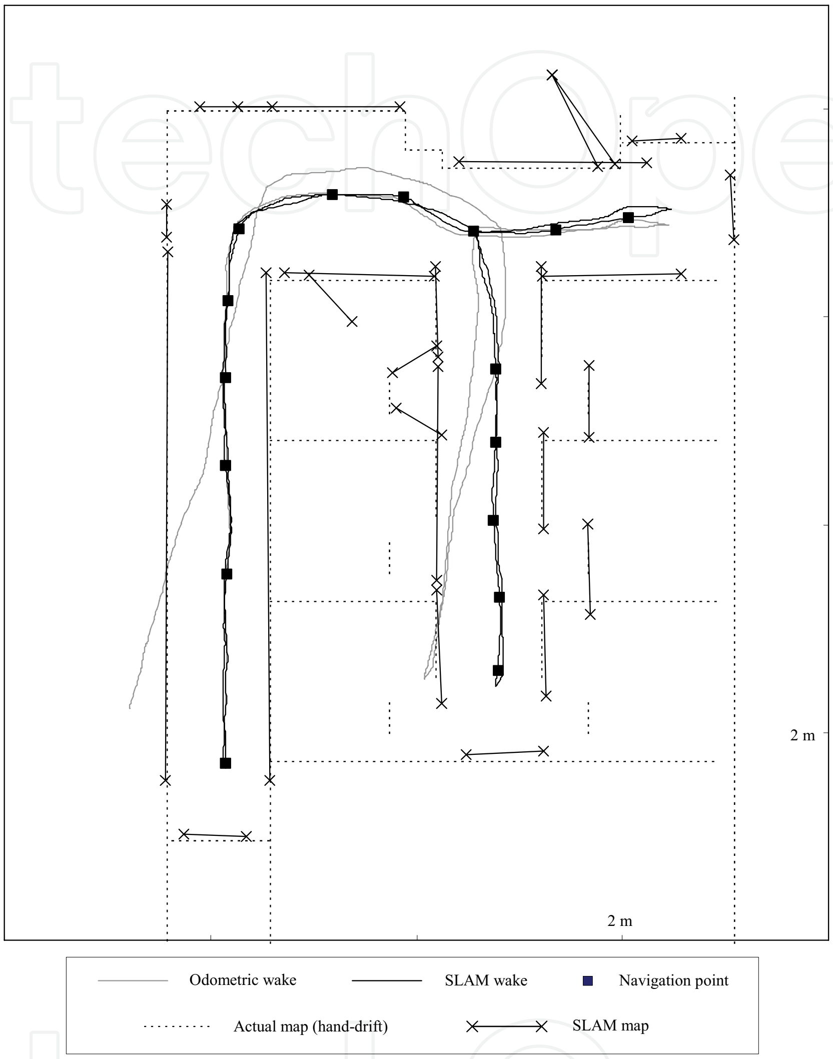

Physical planform (in Pioneer 2DX)
Mobile robot body
Robot perception
Odometric information
sonar sensors return
motion control unit (wheel velocity)
wireless link
wireless link
wireless link
SLAM process
Autonomous exploration
Acquired environmnet model (segments and navigation points)
Software implementation (in Saphira)

Fig. 12. Overall architecture of the proposed autonomous SLAM mechanism.

At the start of the experiment, the Pioneer 2DX was placed at end of the corridor (shown in lower left corner in Figure 13a). After all the given directions at each navigation point were navigated, the mobile robot traveled back to the starting position. The final global map acquired at end of the experiment is shown in Figure 13b. In addition, 25 line features and 16 navigation points were extracted in the final map and the final absolute position error in X and Y is 50mm and 64mm (measured by hand and relative to actual position), respectively. For comparison purposes, the odometric wake, the SLAM wake, extracted navigation points and map model are superimposed on the hand measured map model.



a) Sonar returns, navigation points and autonomous SLAM estimated wake obtained during the experiment. (Captured from the robot software “Saphira”.) The range threshold of all sonar sensors is 1500mm. Therefore, a lot of ambiguous and noise measurements were filtered.



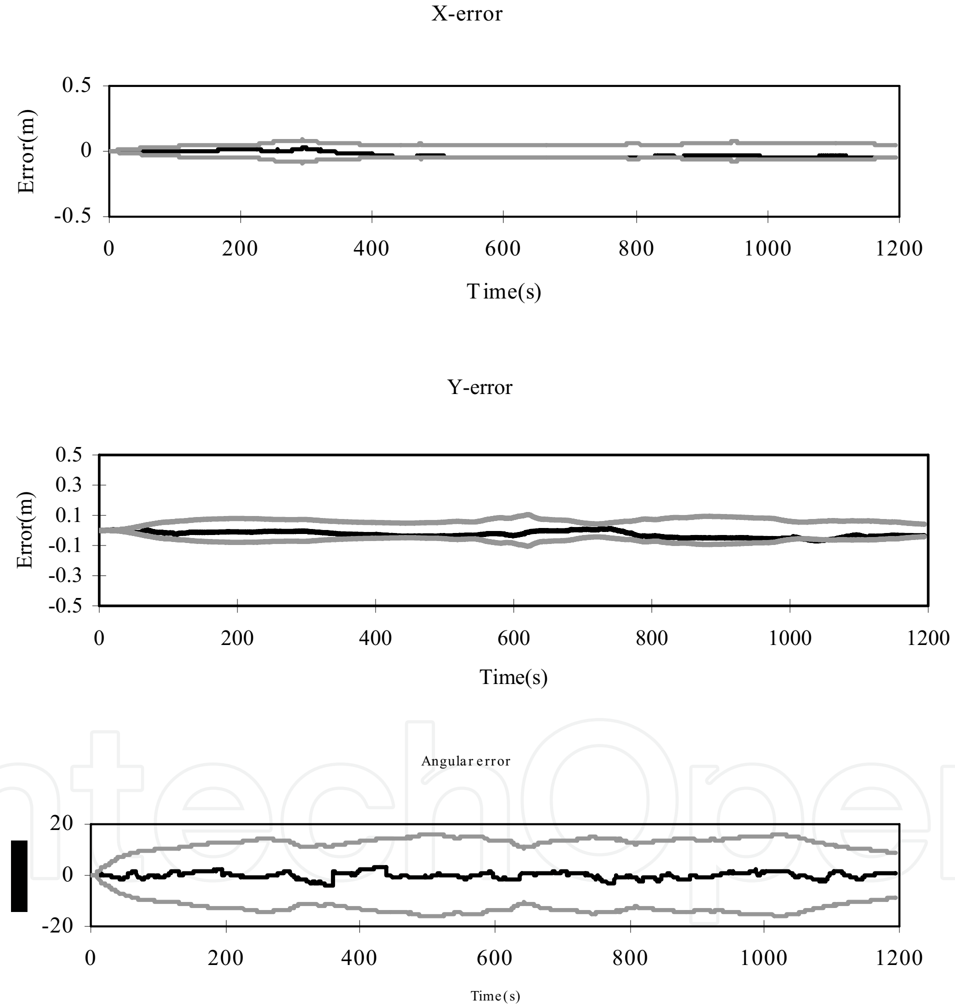
b) Extracted map model and navigation points superposed on the real map.

Fig. 13. Robot trajectory, navigation point-marks, extracted map, raw data and real map during the autonomous SLAM experiment.

To further analyze the consistency of our integrated approach, Figure 14 shows a comparison between the error in the autonomous SLAM pose versus model-based FT-EKF robot pose along with the 2-sigma (2σ) uncertainty bounds logged from the SLAM process. It is clearly demonstrated that those errors remain inside their 2σ uncertainly bounds at the most of time. From this on-line integrated experiment, we conclude that this approach can fulfill the three essential missions of mobile robot and those are operated in real time and simultaneously. Figure 15 shows snap shots captured from the robot software “Saphira”, during the experiment.

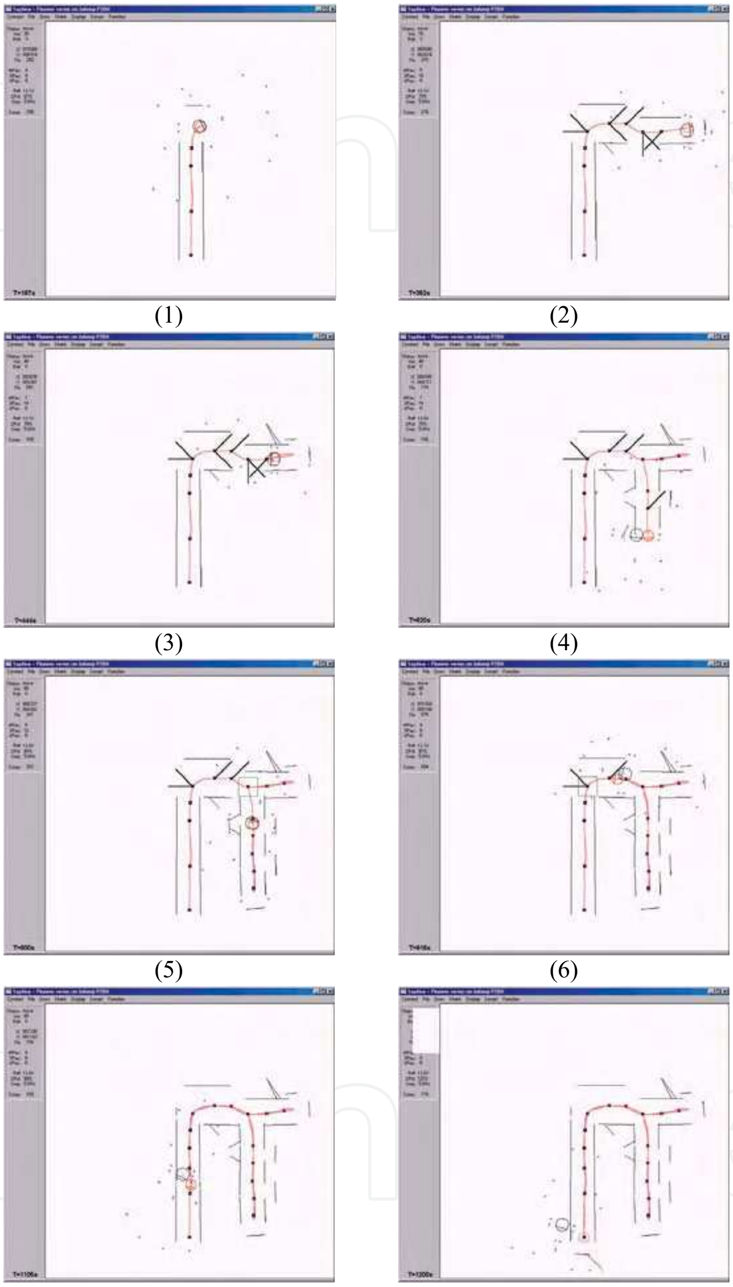
5. Conclusions

In this chapter, a new autonomous exploration strategy for mobile robot was presented and extensively tested via simulation and experimental trials. The essential mechanisms used included a HFFS reactive navigation scheme, EAFC map extraction algorithm, SLAM process, an open space evaluation system cooperating with probability theory and Bayesian update rule and a novel navigation point generation system. The proposed autonomous exploration algorithm is a version of combination of a robust reactive navigation scheme and approaching the unknown strategy which ensure that the mobile robot to explore the entire region in an unknown environment automatically.



Fi

Fig. 14. Estimated errors in robot location during the autonomous SLAM process with sonar. (Gray lines represent two-sigma (2σ) uncertainly bounds.)



(1) (2)

(3) (4)

(5) (6)

(7) (8)

Fig. 15. Snap shots during autonomous SLAM process via Pioneer 2DX mobile robot. (Captured from the robot software "Saphira".) The black robot (a bit bigger) represents the robot position estimated by odometric. The gray robot represents the robot position estimated by SLAM process.

In addition in this chapter, a metric topological map model is advocated for facilitating the path planning process during the autonomous exploration. Moreover, the map model extracted from an EAFC map building algorithm (metric map model) is aimed to generate the navigation point or node on the navigation path. Therefore, a hybrid map model is proposed for autonomous map building in an unknown indoor environment. An autonomous map building algorithm was tested in a simulation world (section 4.2). On the other hand, a successful on-line autonomous SLAM experiment (section 4.3) was conducted for a mobile robot to map an indoor and unknown environment.

Basically, this chapter concluded the pervious work: a SLAM problem solved by overlapping sliding window sonar buffer [Ip and Rad 2003] and EAFC feature initialization technique []Ip and Rad] combined with a novel autonomous exploration strategy to formulate an autonomous SLAM mechanism. Experimental studies demonstrated that the mobile robot was able to build a segment-based map and topological map (a list of navigation points) in real time without human intervention.

6. References

1. Berger, J.O., Statistical Decision Theory and Bayesian Analysis. Springer-Verlag, Berlin. Second Edition, 1985.
2. Borenstein, J. and Koren, Y., Obstacle avoidance with ultrasonic sensors, IEEE Transactions on Robotics and Automation 4 (1988) 213–218.
3. Castellanos, J.A. and Tardos, J.D.: *Mobile robot localization and map building: A multisensor fusion approach*. Boston, MA: Kluwer Academic Publishers (1999)
4. Chan, P.T. and Rad A.B., Optimization of Hierarchical and Fused Fuzzy Systems with Genetic Algorithms, Joint Conference on Intelligent Systems 1998 (JCIS'98), Oct. 23-28, Research Triangle Park, North Carolina, U.S.A., Proceedings Vol. 1, 240-243.
5. Chong, K.S. and Kleeman, L. "Feature-Based mapping in real, large scale environments using an ultrasonic array". *The International Journal of Robotics Research*, Vol. 18, No. 1, pp. 3-19 (1999a)
6. Chong, K.S. and Kleeman, L. "Mobile robot map building from an advanced sonar array and accurate odometry". *The International Journal of Robotics Research*, Vol. 18, No. 1, pp. 20-36 (1999b)
7. Crowely, J.L., Navigation for an Intelligent Mobile Robot. IEEE Journal of Robotics and Autonmation 1 (1), pp. 34-41, 1985.
8. Dissanayake, M.W.M.G., Newman, P., Durrant-Whyte, H., Clark, S. and Csorba, M. "A solution to the simultaneous localization and map building (SLAM) problem". *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp. 229-241. Dec. (2001)
9. Dubrawski, A. and Crowley, J.L., Learning locomotion reflexes: A self-supervised neural system for a mobile robot, Robotics and Autonomous Systems 12 (1994) 133–142.
10. Duckett, T.D., Concurrent map building and self-localization for mobile robot navigation, PhD. Dissertation, In The University of Manchester, 2000.
11. Drumheller, M., Mobile robot localization using sonar. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(2), pp. 325-332, 1987.
12. Edlinger, T. and Weiss, G., Exploration, navigation and self-localisation in an autonomous mobile robot. In Autonomous Mobile Systems, 1995.
13. Gasos, J. and Martin A., A fuzzy approach to build sonar maps for mobile robots, Computers in Industry 32 pp. 151-167, 1996.

14. Gibbons, A. Algorithmic graph theory. Cambridge, England: Cambridge University Press. 1985.

15. Goodridge, S.G., Kay, M.G. and Luo, R.C., Multi-layered fuzzy behavior fusion for reactive control of an autonomous mobile robot, in: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Barcelona, 1997, pp. 579–584.

16. Guivant, J.E. and Nebot, E.M. "Optimization of the simultaneous localization and map building algorithm for real time implementation". *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp 242-257 Dec. (2001)

17. Ip, Y.L., Rad, A.B., Chow, K.M. and Wong, Y.K.: Segment-Based map building using enhanced adaptive fuzzy clustering algorithm for mobile robot applications, *Journal of Intelligent and Robotic Systems* 35, (2002), pp 221-245.

18. Ip, Y.L., Studies on Map Building and Exploration Strategies for Autonomous Mobile Robots (AMR), PhD. Dissertation, In The Hong Kong Polytechnic University, 2003.

19. Ip, Y.L. and Rad, A.B. "Incorporation of Feature Tracking into Simultaneous Localization and Map Building via Sonar Data". *Journal of Intelligent and Robotic Systems* 39: 149-172, 2004.

20. Jamshidi, M., Large-scale systems modelling, Control and Fuzzy Logic, Prentice-Hall, Englewood Cliffs, U.S.A. (1996)

21. Konolige, K., A Gradient Method for Realtime Robot Control, In Proceedings of International conference on Intelligent Robots and Systems IROS 2000.

22. Koren, Y. and Borenstein, J., Potential methods and their inherent limitations for mobile robot navigation, in: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, 1991, pp. 1938–1404.

23. Lacrose, V. and Titli, A. Fusion and Hierarchy can Help Fuzzy Logic Controller Designers, Proc. of sixth IEEE Intern. Conf. of Fuzzy Systems. pp.1159-1165. (1997)

24. Lee, David. The Map-Building and Exploration Strategies of Simple Sonar-Equipped Mobile Robot. Cambridge University Press, 1996.

25. Leonard, J.J., Rikoski, R.J., Newman, P.M. and Bosse, M. "Mapping partially observable features from multiple uncertain vantage points". *The International Journal of Robotics Research*, Vol. 21, No. 10, pp.943-976 (2002)

26. Li, W., Fuzzy-logic based reactive behavior control of an autonomous mobile system in unknown environments, Engineering Application of Artificial Intelligence 7 (8) (1994) 521–531.

27. Li, W., A hybrid neuro-fuzzy system for sensor based robot navigation in unknown environments, in: Proceedings of the American Control Conference, pp. 2749–2753, 1996.

28. Lin, C.H. and Wang L.L., Intelligent collision avoidance by fuzzy logic control, Robotics and Autonomous Systems 20 (1997) 61–83.

29. Lu, F. and Milios, E. "Globally consistent range scan alignment for environment mapping". *Autonomous Robots* 4, pp 333-349 (1997)

30. Lumelshy, V. and Skewis, T., Incorporating range sensing in the robot navigation function, IEEE Transactions on Systems, Man and Cybernetics 30 (1990) 1058–1069.

31. Maeda, M., Maeda, Y. and Murakami, S., Fuzzy drive control of a autonomous mobile robot, Fuzzy Sets and Systems 39 (2) (1991) 195–204.

32. Mataric, M., Navigating with a rat brain: A neurobiologically inspired model for robot spatial representation. In From Animals to Animats 1, 1991.

33. Millan, J.R., Rapid, safe, and incremental learning of navigation strategies, IEEE Transactions on Systems, Man and Cybernetics 26 (3) (1996) 408–420.

34. Musilek, P., Neural networks in navigation of mobile robots: A survey, Neural Network World 6 (1995) 917–928.

35. Nagrath, I.J., Behera, L., Krishna, K.M. and Rajshekhar, K.D., Real-time navigation of a mobile robot using Kohonen's topology conserving neural network, in: Proceedings of the International Conference on Advanced Robotics, pp. 459–464, 1997.

36. Nilsson, N., Principles of Artificial Intelligence. Palo Alto, CA: Tioga, 1980.

37. Nomad 200 User's Manual. Mountain View, CA: Nomadic Technol., 1996.

38. Pal, P.K. and Kar, A., Mobile robot navigation using a neural net, in: Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan, pp. 1503–1508, 1995.

39. Pioneer 2 Mobile Robot Operation Manual. May 2001. ActivMedia Robotics, LLC, v7b.

40. Saffiotti, A., Autonomous robot navigation, in: E. Ruspini, P. Bonissone, W. Pedrycz (Eds.), Handbook of Fuzzy Computation, Oxford University Press, Oxford, 1998 (Chapter G6.1).

41. Saffiotti, A., Fuzzy Logic in Autonomous Navigation, In Driankov, D. and Saffiotti, A. eds., Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag Heidelberg, New York, pp.3-24, 2001

42. Saphira operations and programming manual V6.2, ActivMedia Robotics, 1999.

43. Tardos, J.D., Neria, J., Newman, P.M. and Leonard J.J., Robust Mapping and Localization in Indoor Environments using sonar data. The International Journal of Robotics Research Vol. 21, No. 4, pp.311-330, 2002.

44. Thrun, S., (1998) Learning Metric-topological Maps for Indoor Mobile Robot Navigation, Artificial Intelligence 99 21-71.

45. Tunstel E. and Jamshidi, M., (1997) Intelligent Control and Evolution of Mobile Robot Behavior, Ed. M. Jamshidi, Prentice Hall Inc. 1997, Chapter 1 pp1-24.

46. Tunstel E., H. Danny, Y. Lippincott and M. Jamshidi (1997) Adaptive Fuzzy-Behavior Hierarchy for Autonomous Navigation, Proceeding of the 1997 IEEE International Conference on Robotics and Automation Albuquerque, New Mexico – April 1997, pp829-834.

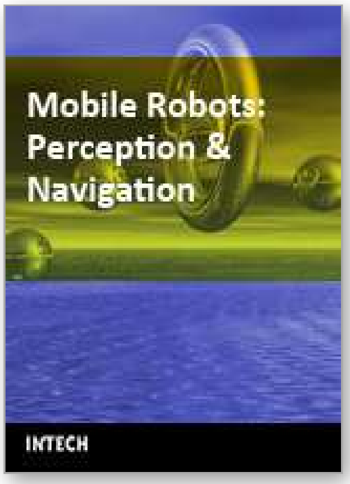
47. Wang, L.X., A Course in Fuzzy Systems and Control. Prentice-Hall International, Inc., 1997.

48. Xu, W.L., Tso, S.K. and Fung, Y.H., Fuzzy reactive control of a mobile robot incorporating a real/virtual target switching strategy, Robotics and Autonomous Systems 23 (1998) 171–186.

49. Xu, W.L. and Tso, S.K., Sensor-Based Fuzzy reactive navigation of a mobile robot through local target switching, IEEE Transactions on System, Man and Cybernetics-Part C: Application and Reviews, Vol. 29, No. 3, pp. 451-459 (1999).

50. Yamauchi, B., A frontier-based approach for autonomous exploration. In Proceedings of 1997 IEEE International Symposium on Computational Intelligence in Robotics and Autonmation, 1997.

51. Zhang, M., Peng, S. and Meng, Q., Neural network and fuzzy logic techniques based collision avoidance for a mobile robot, Robotica 15 (1997) 627–632.



Mobile Robots: Perception & Navigation

Edited by Sascha Kolski

ISBN 3-86611-283-1

Hard cover, 704 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, February, 2007

Published in print edition February, 2007

Today robots navigate autonomously in office environments as well as outdoors. They show their ability to beside mechanical and electronic barriers in building mobile platforms, perceiving the environment and deciding on how to act in a given situation are crucial problems. In this book we focused on these two areas of mobile robotics, Perception and Navigation. This book gives a wide overview over different navigation techniques describing both navigation techniques dealing with local and control aspects of navigation as well es those handling global navigation aspects of a single robot and even for a group of robots.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821