We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Behaviour Based Mobile Robot Navigation with Dynamic Weighted Voting Technique

Shamsudin H.M. Amin, Rosbi Mamat & Tan Chee Kwong

1. Introduction

Designing a mobile robot that operates in our everyday environment is a challenging task. Complexity of the environment set strict requirements for both the hardware and software components of the robot. A robot needs sensors and sensor data processing to keep updating the environment. Furthermore, the robot has to integrate task execution with fast reaction to unexpected situations. These problems are fundamental to embodied autonomous systems that have to interact with unknown dynamic environments. To overcome this problem, various types of architectural framework of mobile robot have been introduced. These methods range from centralized sense-model-plan-act architectures to distributed behaviour based architectures.

Behaviour-based architecture has emerged as an alternative to traditional approaches for designing autonomous mobile robots (Maes, 1989). It consists of a collection of task-achieving modules or behaviours, which achieve goals and run independently. Each behaviour can take inputs from the robot sensors and send outputs to the robot actuators. Intelligence emerges from the interaction of the behaviours of the system. Thus the coordinator plays an important role to combine the outputs from several conflicting behaviours. This is known as the action selection problem (ASP) and following is the definition of ASP: "How can such an agent select 'the most appropriate' or 'the most relevant' next action to take at a particular moment, when facing a particular situation?" (Pirjanian, 1998).

A dynamic weighted voting technique is introduced to solve the problem in multiple behaviour coordination. It proposes a satisfactory mechanism for action selection that covers the three criteria, namely capability of dealing with multiple problems, multi-valued behaviour, and dynamic priority. The use of voting technique for command fusion allows the mobile robot to deal with multiple problems. It takes a shared control approach where each behaviour module concurrently shares control of the mobile robot by generating votes for every possible motor command.

A centre arbiter will then perform command fusion to choose the most appropriate action. Besides, the generated votes are between 0 and 1, with vote zero being the least desired action and vote one is the most desired action. It employs the concept of multi-value rather than simple binary value. The votes are generated in this manner to show the possibility for each action to achieve behaviour's goal. With the weight generation module, the behaviours' weights are generated based on the readings from various sensors. In different situations, the behaviours will have different weights. Therefore, the priority of each

behaviour changes dynamically according to the situation by modifying their relative importance.

Various techniques have been proposed for behaviour coordination to solve the action selection problem. From the literature, the action selection mechanisms had been divided into two main groups, known as arbitration and command fusion respectively (Brooks, 1986). Behaviour arbitration is having the command of one behaviour completely overriding the commands of other behaviours. These include priority-based (Kosecka & Bajcsy, 1993), state-based (Arkin & Mackenjie, 1994), and winner-take-all (Pirjanian, 1998) approaches. Meanwhile command fusion mechanisms coordinate the activities of the set of behaviours that are active simultaneously by selecting the action that best satisfies the system's goal. These can further be divided into superposition (Khatib, 1986; Arkin, 1989; Saffiotti, et al., 1995), fuzzy (Tunstel, 1995; Pirjanian & Mataric, 1999) and voting (Brooks, 1986; Rosenblatt, 1995; Riekki & Roning, 1997) approaches.

In this paper, a dynamic weighted voting technique is proposed for mobile robot goal directed navigation in unknown dynamic indoor environment. Section 2 deals with the robot behaviour analysis. Section 3 discusses the implementation of this goal directed navigation. Section 4 describes the design of behaviour modules and weight manager. Section 5 presents the experimental results and discussion. Conclusion and suggestions for further work is given in section 6.

2. Robot Behaviour Analysis

The analysis of behaviour is an important aspect. The main goal of the mobile robot must be analyzed in detail. It is a top-down approach that involves decomposing the main objective into simpler ones, in such way that the main objective is achieved as a result from the execution of simpler behaviours and from their interaction. In short, a main objective like navigation can be decomposed into simple objectives like obstacle avoidance, and goal seeking. These simple objectives are going to be the basic behaviours for the mobile robot.

However, from an engineering point of view it is more appropriate to construct a system with a specified performance and functionality (Pirjanian, 1998). A top-down breakdown of the given task into a set of objectives identifies a set of relevant behaviours that when coordinated appropriately can achieve that given task.

Each objective module constructed from the value-tree is implemented as a single behaviour. Behaviour is actually a mapping from perception to action. However, it could be not only to define the mapping from perception to action but also to associate with each alterative action value that reflects its desirability. This is the key feature in voting technique, where behaviours vote for the desirability of the possible action set. The vote value is actually representing the preferences of each action in an interval of [0, 1].

Behaviour is treated as a specific objective function. Behaviours directly calculate the objective functions for particular objectives. The votes are generated in order to meet the behaviours' objectives. These votes generate assignments of preferences to action set. The assignments can be a fuzzy membership function, a probability distribution, or other pattern that suitable for the application. Each behaviour may use whichever representation and algorithm is most appropriate for that specific task. This representation does not exclude implementation using a look-up-table, a finite state machine, a neural network, or any other approach.

Meanwhile, the weight for each behaviour also needs to be defined. The weights will reflect the priority of the behaviours. Behaviour with higher priority will have bigger

weight. The weight manager takes the sensor input and decides the weight based on the situation. The weight for each behaviour is chosen with the value to show their priority and normalised to 1. The exact value needs not to be accurate because only the relative values are important (Rosenblatt, 1997).

The calculation of objective functions and assignment of weights are heuristic processes. There is no general guideline on these design issues due to the constraint of the unpredictable dynamic environment that the robot needs to deal with. Because from an engineering point of view, the dynamic changes of the environment are uncontrollable factor. Therefore, the design is done empirically and experiments have to carry out to improve the quality. This is actually a key feature of the embodiment in behaviour-based approaches. The mobile robot needs to experience the world directly, thus the idea of designing a mobile robot can only be proved by having a real robot that deals with the real environment.

2.1 Performance Analysis

The performance of the robot should be evaluated in quantitative terms. However, it is quite complicated due to the non-determinism of the robot trajectory that navigates in unknown dynamic environments (Fabiani et al., 1998). For dynamic domains this is notoriously difficult as they themselves resist modelling and characterization (Arkin, 1995). Comparing two sets of behaviours, even within the same task, is complex and the domain-dependent nature of the solutions can cause these systems to be basically incommensurate – one may fail some times, one may fail at other times and comparison is difficult (Manikonda et al., 1995). This is due to the inherent limitation in the system that deals with unknown dynamic environment. Thus various evaluation parameters are used by different researchers in mobile robot experiments (Gat, 1995; Rosenblatt, 1997; Pirjanian, 1998; Fabiani et al., 1998). In the work presented here, two parameters are suggested to be taken into consideration. These are reliability index and time index.

Reliability index, I_R is the ability to complete the task. It represents the percentage of completed task on the population. The formula is,

$$I_R = \frac{S_{success}}{S_{total}} \tag{1}$$

$S_{success}$ number of success

S_{total} number of tests

Meanwhile, the time index I_T is the completed-task time. It is the average time necessary for the mobile robot to perform the task. It will only be measured when the robot reaches goal. The formula is,

$$I_T = \frac{\sum_n T}{N} \tag{2}$$

T   time to complete the task
N   total number of completed task

The performance is analysed and evaluated from the performance indices. If the performance is not satisfied, the designers should find the root cause and improve the design from the behaviour analysis stage.

3. Implementation: Goal-Directed Navigation

Research in the field of robotics is accompanied by experimental investigation of ideas, validation of theoretical results and so on. Simulations are not well suited for generating conclusive test data and results. The belief, among the opponents of simulation, is that it is only in the real world that certain phenomena manifest themselves and thus studies should be based on real-world experimentation. While simulations are quite useful for the proof-of-concept stage of research, when the feasibility of algorithms needs to be tested, they do not suffice as proof of algorithm functionality in the real world. If an algorithm fails in simulation it will certainly not work in the real world, but the opposite is not necessarily true. Robotics is an experimental science and must have realistic experiments as a central component for verifying hypotheses. In order to test the algorithm and avoid repeated work in simulation, a physical robot, AIBOT, was used for testing and debugging.

3.1 Mobile Robot Goal-Directed Navigation

The motivation behind the presented work was to build a mobile robot for indoor goal-directed navigation. The goal is to implement behaviour-based approach using dynamic weighted voting technique to achieve navigation as a result of a collection of interacting behaviour. The mobile robot must successfully navigate around obstacles, reach its goal and do so efficiently.

Goal-directed navigation problem is a classical problem in mobile robotics. In its original sense, the term navigation applies to the process of directing a ship to its destination. For a navigating mobile robot, the process can be seen as answering the following three question: 1) "Where am I?" 2) "Where are other places with respect to me?", and 3) "How do I get to other places from here?" (Levitt & Lawton, 1990). Sometimes, it does not require knowledge of the starting position. Thus the most important question is "How do I reach the goal?" In short, the definition of navigation was taken as the process of determining and maintaining a course or trajectory to a goal location (Gallistel, 1990). However, to achieve a successful navigation, an appropriate navigation scheme is needed. A navigation scheme is needed to find the lowest cost path from the robot's start state to the goal state. Cost can be defined to be distance travelled, energy expended, time exposed to danger and so on. In the past decade, various kinds of navigation schemes are introduced, ranging from simple reactive search navigation to complicated map-building navigation (Franz & Mallot, 2000). To perform an indoor goal-directed navigation, a simple navigation scheme with direction following is needed. In direction-following navigation, the mobile robot is required to follow a certain direction to find the goal. The goal itself needs not to be perceivable during approach. If direction following is coupled with distance information, then direction following becomes more efficient. Since the environment is unknown a priori, the mobile robot needs to navigate and store the information of the environment. A set of states of the environment is stored and a spatial map of the environment is built for further planning. Planning is required to avoid local minima problem and find optimal way while navigating in unknown environment.

3.2 The Mobile Robot - AIBOT

To test the proposed voting technique, a physical mobile robot, AIBOT (Autonomous Intelligent Mobile Robot) was used in the experiments. Two wheels mounted on a single axis are independently powered and controlled, thus providing both drive and steering for the mobile robot. Meanwhile, two additional passive wheels or castors are provided for support. It moves at an average speed of 0.3 m/s. AIBOT's computational hardware is located on the body. The processing is performed by a MIT Handy Board based on Motorola 68HC11 microprocessor that includes 32K of battery-backed static RAM, two L293D chips capable of driving four DC motors, both analog and digital inputs for a variety of sensors, and a 16x2 character LCD screen (Martin, 1999). An expansion board is added to the system, which provides additional inputs for sensors, digital outputs, and connector mount for Polaroid 6500 ultrasonic ranging system. The Handy Board runs Interactive C (IC in short), a multi-tasking version of the C programming language. It consists of a compiler and run-time machine language module. IC implements a subset of C including control structures, local and global variables, arrays, pointers, 16-bit and 32-bit integers, and 32-bit floating point numbers. The program is compiled to Motorola hex file and loaded into Handy Board's memory using 6811 downloader. AIBOT is equipped with various sensors. These are infrared sensors, sonar sensors, and odometer.

3.3 Sensor Modules

Three kinds of sensors are used in the navigation task. These are infrared sensors, sonar sensors, and odometer. Infrared sensors are a type of light sensors, which function in the infrared part of the frequency spectrum. However, they are preferable to visible light because it suffers a bit less from ambient interference since it can be easily modulated. They consist of an emitter and a receiver. In mobile robot's obstacle detection, infrared sensors are used as reflectance sensors. The emitter provides infrared signal and the signal will be reflected to the receiver if there are obstacles. Infrared sensors can only detect the presence of obstacle but not able to measure the distance of the obstacle. SUNX CX-22-PN infrared sensor is used in AIBOT. The sensing range is up to 0.8 meter. Seven infrared sensors are placed in front of AIBOT. They are directly connected to the digital input ports of Handy Board.

Another common sensor technique in robotics for proximity detection is time-of flight measurement (TOF). This is commonly achieved by using sonar (sound navigation and ranging). In sonar, the detection is based on the propagation of waves between the target and detector. The sensing is initiated by first creating a sonic ping at a specific frequency. These transitions are fed to the transducer at around 50 kHz. As this chirp falls well out of the range of human hearing, the ping is not audible. The chirp moves radially away from the transducer through the air at approximately 343.2 m/s, the speed of sound. When the chirp reaches an object, it is reflected. This reflected chirp then travels back towards the transducer, again at the speed of sound. As the reflected signal hits the transducer, a voltage is created which is fed to a stepped-gain amplifier. The Polaroid 6500 series ultrasonic ranging system is used in AIBOT. It is an economical sonar ranging module that is widely used in robotics researches. This module is able to measure distances from 6 inches to 35 feet. Three transducers are placed in front of AIBOT to sense the environment. To move to a goal point, mobile robots need to know its relative position from the goal location. Dead reckoning (derived from "deduced reckoning" from sailing) is a simple mathematical procedure for determining the present location of a vehicle by advancing

some previous position through known course and velocity information over a given length of time. The simplest form of dead reckoning is often termed as odometry. This implies that the vehicle displacement along the path of travel is directly derived from some on-board odometer. A common means of odometric measurement involves optical encoders directly coupled to wheel axles. In AIBOT, the odometer sensors are connected to digital input port of Handy Board. They provide the rotational count of the wheel and thus the vehicle velocity can be calculated.

3.4 Possible Action Set

If both drive wheels turn in tandem, the robot moves in a straight line. If one wheel turns faster than the other, the robot follows a curved path. In order to set the possible set as candidates, a discrete number of circular trajectories are chosen. Eight possible actions are as shown in Fig. 1 and named as "Hard Left", "Left", "Soft Left", "Forward", "Soft Right", "Right", "Hard Right", and "escape" respectively. "Escape" action is different from others. It is an action where one wheel move forward and the other wheel move backward. Thus the robot turns instantaneously on a point. It enables the robot to turn in a narrow space.
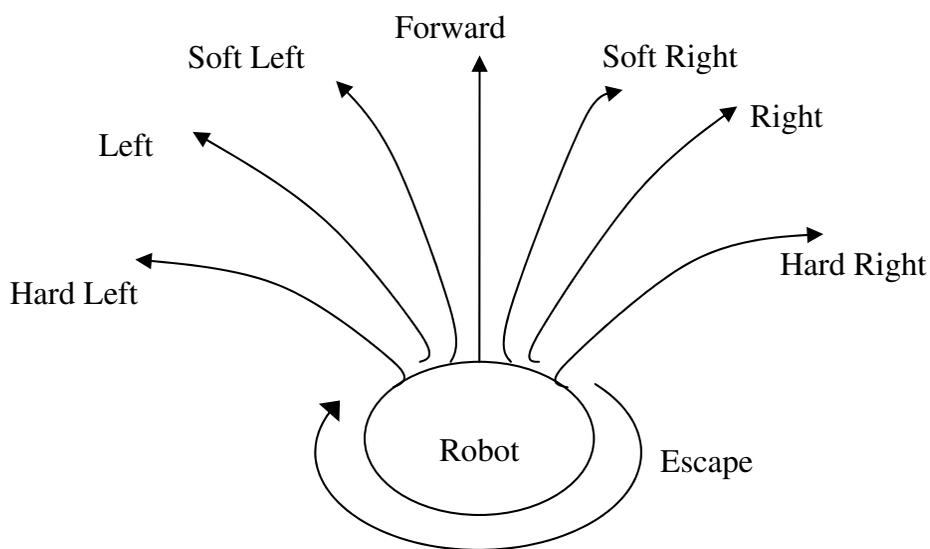

Figure 1. Possible actions for the robot

4. Design of Behaviour Modules and Weight Manager

In this stage, design of each behaviour modules is discussed. Each behaviour votes for every possible action base on the sensor reading. They vote in the pattern to achieve the behaviours' objective. Meanwhile, weight manager is designed to generate weight value for each behaviour.

4.1 Obstacle Avoidance Behaviour

The objective of this behaviour is to move the robot at a safe distance from the obstacles. Obstacle avoidance techniques range from primitive algorithms that detect an obstacle and stop the robot in order to avoid a collision, to sophisticated algorithms that enable the robot to detour obstacles. Sonar sensors and infrared sensors are used separately to meet the objective of avoiding obstacles.

Since infrared sensors are only detecting the presence of obstacles, the real distances of the obstacles are not available. Therefore, they only trigger upon detection of obstacles. For

each possible path, three infrared sensors are responsible to determine the vote for that path. If all three infrared sensors detect the presence of obstacle, obstacle avoidance behaviour with infrared (b_{IR}) will vote 0 to that path and vote 1 if the opposite situation holds. If only one or two of the infrared sensors detect the obstacle, it will vote for a value in between 0 and 1. This can be shown by the equation below,

$$b_{IR}(x_m) = \begin{cases} 1.0 & \text{if no obstacle} \\ 0.7 & \text{if one sensor is triggered} \\ 0.3 & \text{if two sensors are triggered} \\ 0.0 & \text{if three sensors are triggered} \end{cases} \quad (3)$$

For "Escape" action, the voting process is different. The robot only needs to escape when it goes in to a space with a lot of obstacles, where the passage is too narrow and it is not allowed to move forward or turn in circular path. This condition is represented by the density of obstacles. When the number of infrared sensors that sense the presence of obstacle is increased, the obstacle density is high. Therefore, the vote is a function of number of sensors being triggered, as shown in the equation below,

$$b_{IR}(\text{Escape}) = \frac{\text{number of sensors being triggered}}{\text{total number of sensors}} \quad (4)$$

So, when the mobile robot goes into a narrow space, the escape action will receive higher average vote.

The obstacle avoidance behaviour with sonar (b_{sonar}) votes for the possibility to bring the mobile robot to a path free from obstacles,

$$b_{sonar}(x_m) = \begin{cases} 1 & \text{if } D_{obs} > D_{max} \\ \dfrac{(D_{obs} - D_{min})}{(D_{max} - D_{min})} & \text{if } D_{min} < D_{obs} \leq D_{max} \\ 0 & \text{if } D_{obs} \leq D_{min} \end{cases} \quad (5)$$

Dobs   the detected obstacles distance

Dmax the maximum distance to detect

Dmin the minimum distance to detect

4.2 Goal-Seeking Behaviour

The task for goal-seeking behaviour is to look for a goal point and try to approach it. In a completely known environment, an optimal algorithm is used to search a state space to find the lowest cost path for the mobile robot. However, the work presented here focuses on the robot with no information about its environment before it begins its traverse. Thus the goal-seeking behaviour needs to direct the robot toward the goal point from an incomplete and uncertain model of the environment. "How to locate the goal-point" is a main issue in this behaviour. Two different techniques are used to achieve the target of this behaviour. These are goal-seeking behaviour with odometer and goal-seeking behaviour with planning.

This behaviour is designed to assign high values to actions that will cause the robot to face the target and low values to action that do the opposite. To use the odometer and dead-reckoning method, the coordinate system for the robot's world must be defined. The robot operates in a two-dimensional world. This means that at any given time its position can be defined in terms of 2-D Cartesian coordinates and the direction it is facing can be defined as an angle measured from one of the axes, as illustrated in Fig. 2. The orientation of the robot (theta) is measured counter clockwise from the x-axis. The position coordinates, x and y, are in meters and the orientation angle, theta, is in radians.
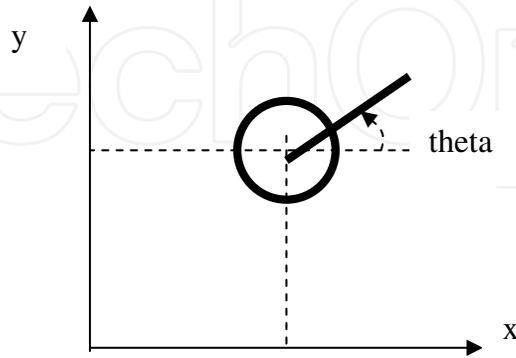
Figure 2. Coordinate System of the Mobile Robot

This behaviour generates the votes in three steps. First, the robot receives signals from the odometer. It counts the pulses from the odometer. These pulses are used to calculate the robot's position as well as its heading. The robot's current position and heading is calculated as below,

$$x(t) = x_o + \frac{d(V_R + V_L)}{2(V_R - V_L)} \left[\sin\left(\frac{t(V_R - V_L)}{d} + \theta_o \right) - \sin\theta_o \right] \qquad (6)$$

$$y(t) = y_o + \frac{d(V_R + V_L)}{2(V_R - V_L)} \left[\cos\left(\frac{t(V_R - V_L)}{d} + \theta_o \right) - \cos\theta_o \right] \qquad (7)$$

$$\theta(t) = \frac{t(V_R - V_L)}{d} + \theta_o \qquad (8)$$

V_R Velocity of right wheel
V_L Velocity of left wheel
t   time
d   distance between wheels
θ_o initial orientation of the robot

Secondly, with the robot's current position and heading, the relative position to the goal-point is calculated. The robot will then locate the target point direction. Finally, it votes for each possible action according to their angle to the goal point. A trajectory that will lead the robot toward the goal will get a higher vote and vice versa. This will give the robot more flexibility in navigation and take the uncertainty into consideration. The vote

evaluation in this goal-seeking behaviour with odometer (b_{odo}) is represented as an objective function of $\prod$–function (Pirjanian, 1998),

$$b_{odo}(x_m) = \frac{1}{1 + \left(\dfrac{\theta_m - \theta_{goal}}{\beta}\right)^2} \tag{9}$$

θ_m angle of the candidate path

θ_{goal} relative angle to the goal

β width of window, which determines the value

Fig. 3 illustrates an example of the vote evaluation in this behaviour. In Fig. 3a, the mobile robot detects the goal on 60o to the right. It broadens the target direction and generates the votes as shown in Fig. 3b.
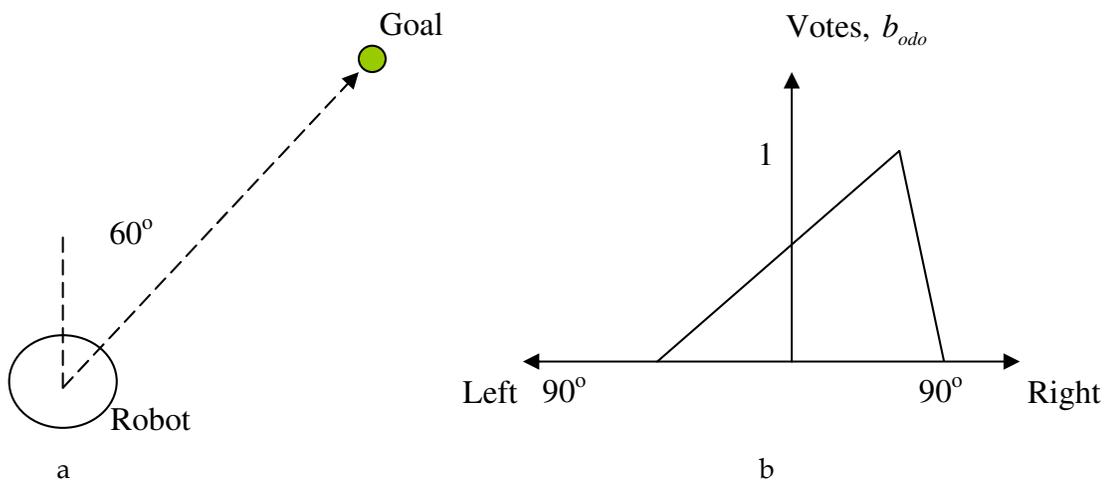
Figure 3. Vote evaluation in goal-seeking behaviour

In mobile robot navigation, the use of planning is a critical issue. While planning is useful for mobile robot to escape from the trap and avoid local minima problem, it is computationally expensive. This is mainly due to the use of knowledge representation. The knowledge could be a map, an analytical representation of surface features, or a semantic description of the world. Traditional systems build symbolic maps of the world for navigational reference. In the work presented here, knowledge is represented as a local spatial memory to minimize the use of memory. The addition of a local spatial memory allows the mobile robot to avoid areas that have been visited.

The concept of this local spatial memory is equivalent to leaving chemical trail in ants (Balch & Arkin, 1993). The memory is a two dimensional array of integers, which corresponds to the environment to be navigated. Each element of the grid records the number of times the corresponding square patch in the world has been visited. Every time the grid point is visited, it will be given a value of +1. The more often an area is visited, the larger the value. This recorded "trail" is used later for planning.

The planning function uses the memory with current positional information to generate the vote to run away from an area that has already been visited. It compares the mark point of the area around the robot's current position and finds the freest space, which is the least visited. The direction is recorded and set to be the temporary goal point. The planning function will vote for this temporary goal point in the same manner as the algorithm in goal-seeking behaviour with odometer. That is, it votes for each possible action according to their angle to the temporary goal point. The formula for goal-seeking behaviour with planning (b_{plan}) is,

$$b_{plan}(x_m) = \frac{1}{1 + \left(\dfrac{\theta_m - \theta_{goal}}{\beta}\right)^2}$$ (10)

θ_m angle of the candidate path

θ_{tem} relative angle to the temporary goal point

β width of window, which determines the value for half vote

4.3 Weight Manager

The weight manager plays an important role to change the behaviour's weight dynamically to enable the mobile robot to deal with the complexity of the environment. From the discussion above, it shows that there are two levels of arbitration in the mobile robot, that is the arbitration in behaviour team and the arbitration in centre arbiter. Therefore, the weight manager needs to generate three weight functions, one for arbitration in obstacle avoidance behaviour team, one for goal-seeking behaviour team, and one for centre arbiter. In obstacle avoidance behaviour team, there are two behaviours. Although the two behaviours use different sensors to achieve the same function, their priorities are the same in every situation of the environment. No matter what is the situation of the environment, both the behaviours will vote for the path to avoid obstacles. To represent their priorities in the system, these two behaviours have the same weight value. In other words, the weight values for these two behaviours will not change; they both are assigned with the value of 0.5. That is,

$$W_{IR} = E_{sonar} = 0.5$$ (11)

where W_{IR} and W_{sonar} are the weight for obstacle avoidance with infrared and the weight for obstacle avoidance with sonar respectively.

In goal-seeking behaviour team, the two behaviours have different priorities in different situations. In example, when the mobile robot in a free state, the goal-seeking behaviour with planning will have lower priority than goal-seeking behaviour with odometer. On the other hand, when the mobile robot is trapped in a local minima, the goal-seeking behaviour will have higher priority such that its vote could bring the mobile robot out of the visited area. Therefore, the weight values are the function of the change in environment situation, while the environment situation is the degree of belief in a trapped

area. It takes an assumption where the belief in trapped area is proportional to the sum of mark point value. If the mark point value is high, the belief in a trap area is high. Thus the weight value for goal-seeking behaviour with planning will increase, as shown below,

$$W_{plan} = \frac{\text{mark\_sum - mark\_min}}{\text{mark\_max - mark\_min}} \tag{12}$$

$$W_{odo} = 1.0 - W_{plan} \tag{13}$$

W_{plan} Weight value for goal-seeking behavior with planning

W_{odo} Weight value for goal-seeking behavior with odometer

mark_sum   total sum for mark point

mark_min   minimum sum value for mark point to trigger planning

In the centre arbiter, there are two weight values, which are weights for obstacle avoidance behaviour team and goal-seeking behaviour team respectively. The obstacle avoidance behaviour's weight must be larger to reflect that avoiding obstacles is more important than approaching the goal (Rosenblatt, 1997). Therefore, the weight value for obstacle team is set in an interval of [0.6, 0.9] while the goal team is [0.4, 0.1]. The weight values change according to the density of obstacles. An assumption is taken where the obstacle density is proportional to the number of infrared sensors being triggered. If the number of the infrared sensors being triggered is high, the obstacle density is high. It means that the mobile robot is in an obstructed situation with cluttered of obstacles. Thus the obstacle avoidance behaviour team should be given higher weight value relative to the condition when obstacle density is low. This is shown in the equation,

$$W_{obs} = 0.6 + \left(\left(\frac{N}{N_{total}} \right) * 0.3 \right) \tag{14}$$

$$W_{goal} = 1.0 - W_{obs} \tag{15}$$

W_{obs} Weight value for obstacle voidance behavior team

W_{goal} Weight value for goal-seeking behavior team

N Number of infrared sensors being triggered

N_{total} Total number of infrared sensors

5. Experimental Results and Discussion

Experiment is the way to prove the theoretical concept in real world. In the previous chapter, the implementation of dynamic weighted voting technique is discussed. The proposed technique is targeting on solving the problem in action selection for behaviour-based mobile robot. It was implemented on AIBOT for indoor goal-directed navigation. Experiments were carried out to validate the design of the proposed technique. The experimental results, comparisons and discussions are covered in this chapter.

5.1 Experimental Procedure

The proposed idea is evaluated and tested across a variety of different types of environment and behaviour combinations. The mobile robot went through the tests in 14 experimental fields. The objective of the experiment is to test the proposed dynamic weighted voting technique for behaviour-based mobile robot goal-directed navigation in an unknown dynamic environment. AIBOT was designed with four behaviours. The experiments were designed to show the results of the navigation with different behaviour combinations. Therefore, four experiments were carried out with the behaviour combination implemented shown as described below. In each experiment, AIBOT is tested in all of the 14 experimental fields. For each field, AIBOT was run for 50 times to get an average result. Thus in each experiments, 700 experimental runs were conducted. While running the experiment, the battery level for the motor is very important as the difference in battery level may affect the speed of AIBOT and thus affect the performance. To avoid this problem, the battery is fully charged before starting a new experiment set. Data is collected and the performance indices are reliability index and time index.

5.2 Navigation Results

The results of these navigation experiments are presented and discussed in this section.

Experiment 1 - Obstacle avoidance behaviour with infrared and goal-seeking behaviour with odometer

This was an experiment with only two behaviours, which are obstacle avoidance behaviour with infrared and goal-seeking behaviour with odometer. The objective was to test the concept of voting in solving action selection problem. The two behaviours with different objectives may generate conflicting action. Thus the dynamic weighted voting technique should solve the problem and choose the most appropriate action. The results are shown in Table 1.

	Reliability Index, I_R	Time Index, I_T
Field 1	1	10.31
Field 2	1	12.15
Filed 3	0.94	17.89
Field 4	0.88	14.75
Field 5	0.92	15.36
Field 6	1	14.32
Field 7	0.86	19.03
Field 8	0.64	26.25
Field 9	0.70	24.17
Field 10	0.66	27.12
Field 11	0.54	29.87
Field 12	0.30	34.71
Field 13	0.14	80.43
Field 14	0.16	84.59

Table 1. Results of Experiment 1

In field 1, the mobile robot showed reliable navigation to achieve the goal point with an average of 10.31s. Experiments in field 2 to field 5 indicated that the dynamic weighted voting technique was able to handle the conflict between the obstacle avoidance behaviour and goal-seeking behaviour. While avoiding obstacles in the test field, the mobile robot was able to maintain heading to the goal target. An example is illustrated in Fig. 4 for

navigation in field 3. In the beginning, the obstacle avoidance behaviour will vote equally for each path because it detects no obstacle. Meanwhile, the goal-seeking behaviour will vote for the direction of the goal point. So, AIBOT will move to the direction of the goal point. When the obstacle is detected, the obstacle avoidance behaviour will vote for a free path. Although the goal-seeking behaviour will vote for the forward move, AIBOT still take a turn because of the greater weight of the obstacle avoidance behaviour. By the time AIBOT pass over the obstacle, all the paths are free and will get equal vote from obstacle avoidance behaviour. Therefore, it will take a right turn to go to the goal point as voted by the goal-seeking behaviour. However, the results show that the mobile robot may sometimes fail to achieve the goal point. It may collide with the obstacles during navigation. This is due to the obstacle angle that is not detected by the infrared, especially at the edges of obstacle.
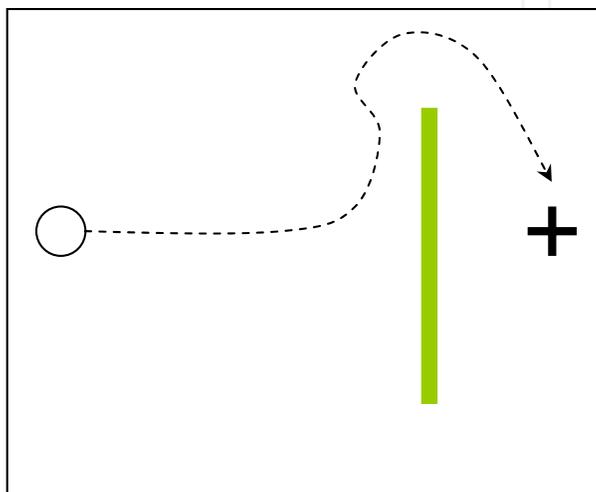
Figure 4. An Example for Navigation in Field 3

For navigation in corridor-like environment, the mobile robot shows high reliability in field 6. For corridor in field 7, there is some failure in the infrared sensors due to the obstacle angle too. In these two experiments, the time index showed an interesting issue where the mobile robot sometimes perform non-optimal path. Fig. 5 shows an example of this condition. Since the navigation is reactive, the mobile robot may in a position and sense that there are too many obstacles around. Thus the obstacle density increases. The obstacle avoidance behaviour then votes for "escape" action. As a result, the mobile robot turns around in the corridor or wobbles while the path is free.
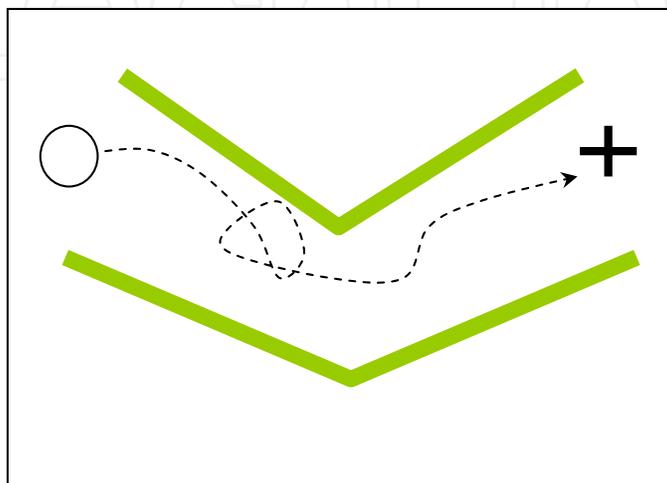

Figure 5. A condition of generating non-optimal path for corridor navigation in Field 7

The test in field 8 to field 11, the mobile robot produced low reliability index. Since the infrared sensors could only sense the presence of obstacles, it is not able to locate the distance of the obstacles. This has reduced the reliability for the mobile robot to find a narrow passage. Even if it can find the passage, it may waste some time to navigate and turn around. This is shown in Fig. 6 for field 9.
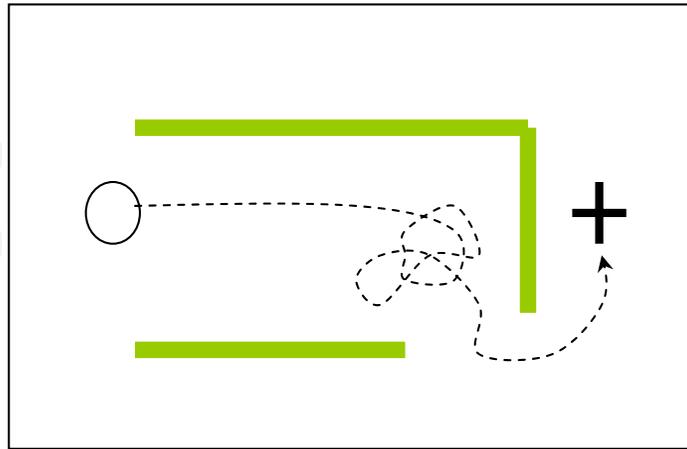

Figure 6. Mobile robot navigate turn around near the narrow passage

For the local minima problem, the mobile robot with only the reactive behaviours will navigate repeatedly in the same area. It can only come out from the trap by chance based on the interaction with the environment. Therefore, the reliability index is very low.

Experiment 2 - Obstacle avoidance behaviour with sonar and goal-seeking behaviour with odometer

This was an experiment of obstacle avoidance behaviour with sonar and goal-seeking behaviour with odometer. It was similar to experiment 1 in that to test the concept of voting in solving action selection problem. However, in this case the sonar sensor was used for obstacle avoidance behaviour. This experiment also provided a comparison regarding the difference in using infrared and sonar for obstacle avoidance. The results were shown in Table 2.

Navigation in field 1 indicated a high reliability to achieve the goal point with an average of 10.45s. The results for experiments in field 2 to field 5 showed a high reliability with only a few failures as experiment 1. With the dynamic weighted voting technique as the backbone, the mobile robot was able to achieve the goal point, to handle the conflict between the obstacle avoidance behaviour and goal-seeking behaviour. The failure, again, was due to the obstacle angle that was not detected by the sonar, especially at the edge of obstacles.

The result of navigation in corridor-like environment was similar to experiment 1. The mobile robot was confronted with the same problem of sensor failure.

For the test of narrow passage, as in field 8 to field 11, the mobile robot showed higher reliability relative to experiment 1. The mobile robot was more reliable in finding the narrow passage. It could find the way faster than it was in experiment 1 as indicated by the time index. This was because of the use of sonar sensors, which enabled the mobile robot to calculate the distance of obstacles and thus generate the vote based on the distances. Fig. 7 illustrates an example of smooth path through the narrow passage. However, there was sometimes some collision with obstacle due to the obstacle angle.

	Reliability Index, I_R	Time Index, I_T
Field 1	1	10.45
Field 2	1	11.97
Filed 3	0.86	15.53
Field 4	0.92	14.12
Field 5	0.92	16.26
Field 6	1	13.25
Field 7	0.78	16.46
Field 8	0.68	19.31
Field 9	0.74	19.67
Field 10	0.74	18.89
Field 11	0.66	20.12
Field 12	0.26	42.28
Field 13	0.20	76.31
Field 14	0.08	86.49

Table 6.2 Results of Experiment 2
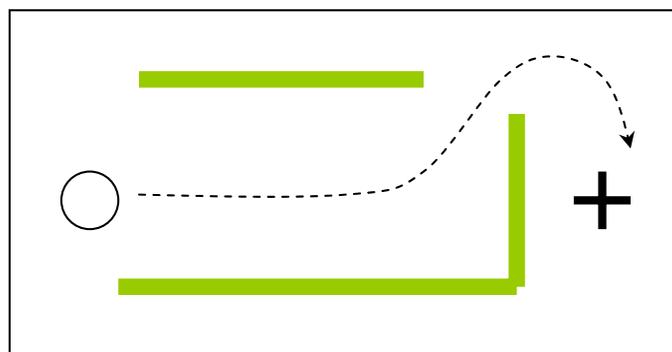
Figure 7. Smooth Navigation through the Narrow Passage

For the local minima problem, the results were similar as in experiment 1 because the mobile robot had only the reactive behaviours. Therefore, the mobile robot will navigate repeatedly in the same area. It can only come out of the trap by chance based on the interaction with the environment.

Experiment 3- Obstacle avoidance behaviour team (infrared and sonar) and goal-seeking behaviour with odometer

This was an experiment with three behaviours, namely obstacle avoidance behaviour with infrared, obstacle avoidance behaviour with sonar and goal-seeking behaviour with odometer. These experiments were designed to test the use of dynamic weighted voting technique for command fusion in homogeneous behaviour team. Besides, the performance of the mobile robot with the use of homogeneous behaviour team was studied. The results are shown in Table 3.

For navigation in field 1, the results were similar as in the previous experiments. AIBOT navigated with high reliability to achieve the goal point in average of 10.41s. In experiment for field 2 to field 5, the mobile robot achieved higher reliability compared to experiments 1 and 2. With the homogeneous behaviour team, both the obstacle avoidance behaviours with infrared and sonar generate voted to achieve the objective of avoiding obstacles.

	Reliability Index, I_R	Time Index, I_T
Field 1	1	10.45
Field 2	1	11.23
Filed 3	0.94	16.85
Field 4	1	13.94
Field 5	1	13. 76
Field 6	1	13.75
Field 7	0.94	14.52
Field 8	0.80	16.19
Field 9	0.78	17.16
Field 10	0.82	16.87
Field 11	0.72	18.43
Field 12	0.34	40.54
Field 13	0.24	81.22
Field 14	0.10	93.50

Table 3. Results of Experiment 3

The dynamic weighted voting technique was able to combine the votes from the behaviours.

In corridor navigation, the results in reliability index and time index showed that the performance was relatively better. The time for achieving goal point was reduced, due to the reduction in non-optimal paths. The navigation was smoother as shown in Fig. 8. This was due to the use of two types of sensors that helped to increase the reliability, as stated in the belief of "uncertainty handling with homogeneous behaviour".

For the test in field 8 to field 11, the mobile robot achieved a higher reliability compared to experiments 1 and 2. It appeared that performance was better with the use of homogeneous behaviour team. This proved that the command fusion in dynamic weighted voting technique was able to be an alternative to traditional sensor fusion. For the experiments in field 12 to field 14, the mobile robot achieved low reliability results. It was trapped in the local minima due to inability to do planning.
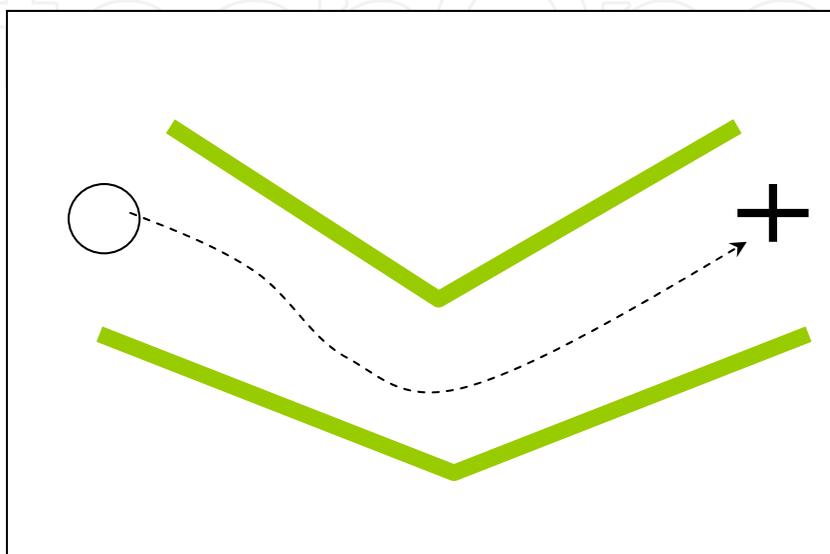

Figure 8. A Smoother Path in Corridor Navigation

Experiment 4 - Obstacle avoidance behaviour team (sonar and infrared) and goal-seeking behaviour team (odometer and planning)

This was an experiment with four behaviours. Two behaviour teams were built, which were obstacle avoidance behaviour team and goal-seeking behaviour team. The new added behaviour was goal-seeking behaviour with planning. The objective was to test the mobile robot with planning capabilities, to test the ability of the dynamic weighted voting technique to handle the use of planning behaviour, and to solve the problem in local minima. The results are shown in Table 4.

	Reliability Index, I_R	Time Index, I_T
Field 1	1	10.57
Field 2	1	12.03
Filed 3	0.90	17.48
Field 4	0.96	15.65
Field 5	1	14.11
Field 6	1	14.02
Field 7	0.94	15.03
Field 8	0.84	16.53
Field 9	0.82	16.77
Field 10	0.82	17.06
Field 11	0.76	19.14
Field 12	0.62	26.18
Field 13	0.54	65.32
Field 14	0.48	67.98

Table 6.4. Results of Experiment 4

For navigation in field 1 to field 11, the mobile robot was able to reach the goal point with the results similar to the previous experiments. This proved an easy way to add in new behaviours. With the design of reusable behaviour modules, a new behaviour could be easily added in without modifications to previous behaviours.

The main focus of this experiment was to test the quality of planning behaviour in field 12 to field 14. An example of navigation in field 13 is shown in Fig. 9. The mobile robot moved into the trap in the beginning because it had no knowledge about the environment. After some navigation in the same area, the mobile robot then marked the grid point and kept it as a local spatial memory. While the sum of mark point increased, the goal-seeking behaviour with planning will be given a higher weight than odometer. Meanwhile, this behaviour found a new goal to move the freest space at the bottom. It thus caused the mobile robot to go out of the local minima trap.
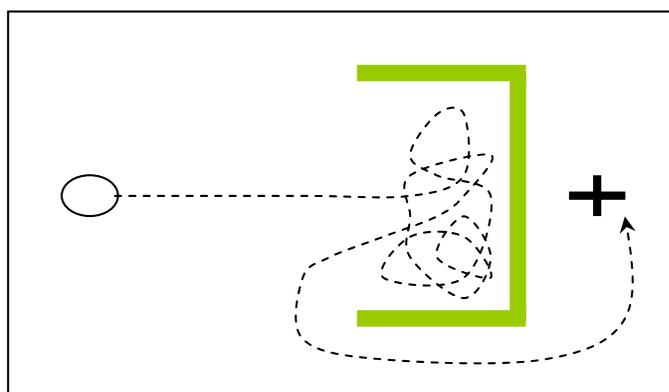

Figure 9. An Example of Navigation to Recover from Local Minima Problem

5.3 Discussions

The four experiments above showed the navigation of AIBOT using dynamic weighted voting technique. The robot navigated in various kinds of environments with various behaviour combinations. The experiments indicated that the proposed technique was able to solve the action selection problem and provide a successful behaviour design.

The results from the four experiments above were rearranged in Fig. 10 and Fig. 11 from the view of reliability index and time index respectively. The experimental data showed that the performance of AIBOT was relatively better in experiment 4, which was the experiment with four behaviours. This behaviour combination with an additional planning behaviour provided a solution for the problem of local minima. Meanwhile, it maintained the performance quality in other simple experimental fields. However, the performance in experiment 1 and 2 was sufficient for simple experimental field with only a few obstacles, such as field 1 to field 6. From the view of cost, these two behaviour combinations were easier to design. For behaviour combination in experiment 3, it indicated better performance compared to experiment 1 and 2 in the experiment for narrow passage, from field 7 to field 11. This is the advantage of using homogeneous behaviour team of obstacle avoidance.
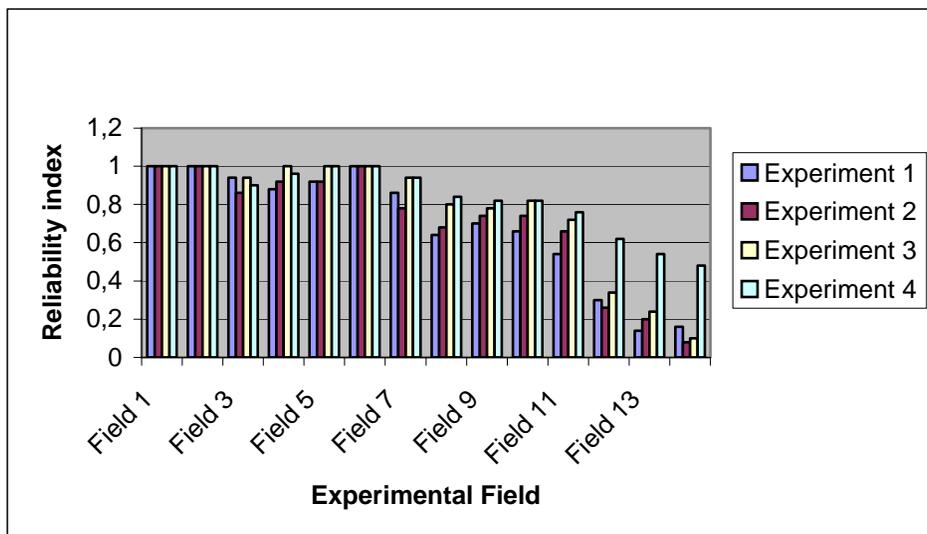

Figure 10. Reliability Indices for All Experiments
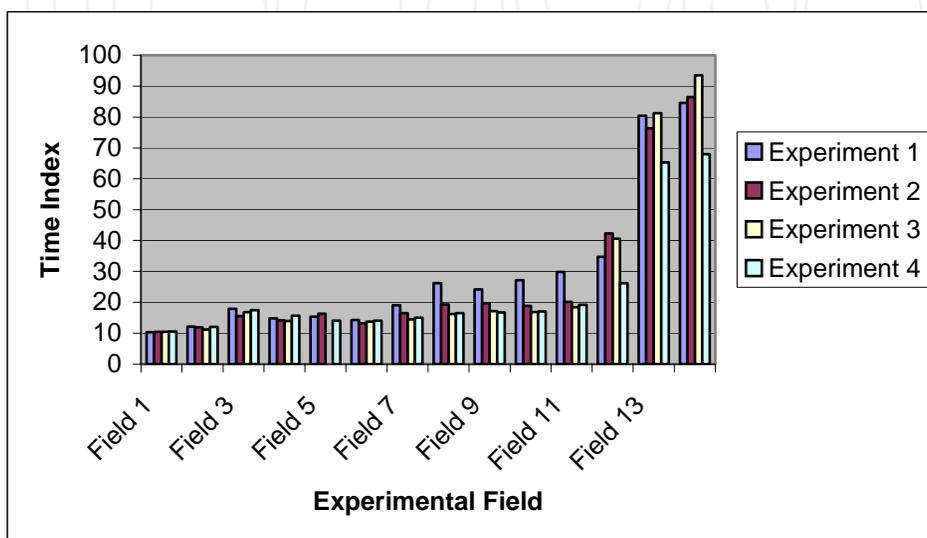
Figure 11. Time Indices for All Experiments

218

An obvious problem of the experiment above was the use of odometer as the only sensor for localisation. While odometer was easy to implement, it suffered from the problem of common dead reckoning error due to wheel slippage, unequal wheel diameter, and non-point wheel contact with the floor. This may have caused the error in calculation of the robot position. Therefore in some experiments, the mobile robot could not stop exactly on the position of the goal point.

5.4 Comparison with Other Techniques

A challenge in robotics is how to compare different architecture and architectural styles. Research papers on architectures are typically descriptive, describing what the system did and how it was organized but rarely provide metrics that would enable readers to determine what effect, if any, the architecture itself had on system performance. Thus comparison is done based on the theoretical analysis, rather than the runtime performance metrics due to the difference in mobile robot hardware, test field and others.

The dynamic weighted voting technique takes the advantages of previous action selection mechanisms while avoiding the shortcomings. The use of voting technique for command fusion allows the mobile robot to deal with multiple problems. Each behaviour module concurrently shares control of the mobile robot by generating votes for every possible motor command. This has overcome the shortcoming of behaviour arbitration technique that only deals with single problem at each point in time. Rather than choosing among behaviours, the dynamic voting technique allows the mobile robot to take different actions from all behaviours into consideration.

Meanwhile, the votes are generated between 0 and 1, with vote zero being the least desired action and vote one is the most desired action. This multi-value approach takes the idea of fuzzy logic with the belief that "the world is not black and white but only shades of gray". It enables the robot to deal with uncertainty in perception and incomplete information about the environment. The interval vote value shows the possibility for each action to achieve behaviour's goal rather than generate a single action such as behaviour arbitration and superposition command fusion. With the weight generation module, the behaviours' weights are generated based on the readings from various sensors. It modifies the relative importance of each behaviour by varying their weight value. In different environment situation, the behaviours will have different weights. Therefore, the priority of each behaviour changes dynamically according to the situation. This is the feature of winner-take-all behaviour arbitration, where the behaviours' priorities are changing dynamically. It will enable the mobile robot to deal with the complexity of the environment and avoid the discrimination against behaviours.

5.5 Advantages and Disadvantages Dynamic Voting Technique

The main advantage of the dynamic weighted voting technique is that it provides a reactive solution to deal with challenges in real-time responsiveness. The behaviour module directly maps the sensors reading to the vote value as an objective function. The computational process in sensor fusion is avoided. Meanwhile, planning modules is allowed to provide a better solution. With the dynamic weighted voting technique as the backbone, the performance of other behaviour modules will not be affected with the addition of planning modules. It employs a fully distributed architecture where all the behaviour modules run completely independently. The technique also enables the mobile robot to handle the challenges of uncertainties. The votes generated in an interval of [0, 1]

in order to take uncertainties into consideration. Furthermore, the dynamically changing weights allow the mobile robot to deal with dynamic changes of the environment. In addition, the proposed design guidelines provide an engineering way to design the mobile robot from initial stage to performance analysis.

However, it too encounters with several problems. From the experiments, it is clear that the dynamic weighted voting technique is not allowed for task sequencing. This means that the mobile robot is not able to handle multiple tasks together and plan the sequence of these tasks in advance. In the other hand, although the design guideline provides a solution in behaviour design, the determination of the vote value is still primitive. A lot of experiments need to be carried out to find a better result.

5.6 Future Development

Current work on the dynamic weighted voting technique has some limitations because it is still on the development stage. There is still a room for improvement. Future development is suggested to focus on several directions, as discussed below.

1.) Learning capability

Currently, the objective function for the generation of vote value is determined empirically through experiments. The approach often taken is an iterative one of trial and error. The design of some behaviour may need several days of experimental debugging. Therefore, the process is time-consuming. Furthermore, once the vote value is fixed, it will not change. Further work may focus on studying the possible role of learning techniques to generate behaviours. Learning capability produces changes within a mobile robot to enable it to perform more effectively in its environment. It serves as an aid to fine-tune the vote value. With the learning capability, the mobile robot could learn during the navigation and tune the objective function to an optimum value.

2.) Additional level for task sequencing

The dynamic weighted voting technique does not support for task sequencing. An additional layer could be added into the architecture to enable task sequencing. This layer does not take control within the voting scheme. It plays the role of planning the sequences of tasks, such as going to the table, then getting the can and finally throwing it into rubbish bin. With the support of task sequencing, the mobile robot may perform more complicated tasks and serve as human assistance in our everyday environment.

3.) Scaling in the number of behaviour

The current dynamic weighted voting technique is implemented on mobile robot indoor navigation with four behaviours. However, the limit of the performance is still an unknown. An interesting extension will have to be in the direction of more complex behaviour. It is interesting to test the performance in complicated task with large composition of behaviours.

4.) Extension into multi agent mobile robots

Since the dynamic weighted voting technique shows a successful result in mobile robot navigation, the work is suggested to be extended into multi agent mobile robot. The work in multi agent mobile robot includes the decision making of

multiple mobile robots. This robot team must be able to coordinate their job together to perform a useful task. The problem is similar to the decision making of multiple conflicting behaviour modules in the work presented here. Thus extending the dynamic weighted voting technique into multi agent mobile robot is a possible work.

6. Conclusion

Experiments were carried out to prove the effectiveness of the proposed dynamic weighted voting technique. The experiments were carried on fourteen experimental fields with four different behaviour combinations. The results and comparison of the different experiments wee discussed. These results appear to show that the dynamic weighted voting technique was able to handle the problem in action selection. Meanwhile, the design also provided a way to design the mobile robot with dynamic weighted voting technique in an organised manner. Comparison with other behaviour-based approaches was briefly discussed, so as well the advantages and the disadvantages of the dynamic weighted voting technique.

7. References

Arkin, R.C. (1989). Towards the Unification of Navigational Planning and Reactive Control. AAAI Spring Symposium on Robot Navigation. 1-5.

Arkin, R.C. (1995). Just what is a Robot Architecture Anyway? Turing Equivalency versus Organizing Principles. Proceedings of the AAAI Spring Symposium: Lessons Learned from Implemented Software Architectures for Physical Agents. Stanford, CA. 21-24.

Arkin, R.C. and D. MacKenjie. 1994. Temporal Coordination of Perceptual Algorithms for Mobile Robot Navigation. IEEE Transaction on Robotics and Automation 10(3): 276-286.

Balch, T. and Arkin, R.C. (1993). Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation. Proceedings of IEEE International Conference on Robotics and Automation. Atlanta, GA. 678-685.

Brooks, R.A. (1986). A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation. 2(1). 14-23.

Fabiani, L., Regnier, S., and Monacelli, E. (1998). Performance Analysis Tool for Reactive Mobile Robots. Proceedings of the Third IFAC Symposium on Intelligent Autonomous Vehicles (IAV '98). Madrid, Paris. 25-30.

Franz, M.O. and Mallot, H.A. (2000). Biomimetic Robot Navigation. Robotics and Autonomous Systems. 30.133-153.

Gallistel, C.R. (1990). The Organization of Learning. MIT Press. Cambridge, MA.

Gat, E. (1995). Towards Principled Experimental Study of Autonomous Mobile Robots. Autonomous Robots. 2(3). 179-789.

Khatib, O. (1986). Real-time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research. 5(1). 90-98.

Kosecka, J. & Bajcsy, R. (1993). Discrete Event Systems for Autonomous Mobile Agents. Proceedings of the Intelligent Robotic Systems '93. Zakopane. 21-31.

Levitt, T.S. and Lawton, D.T. (1990). Qualitative Navigation for Mobile Robot. Artificial Intelligence. 44. 305-360.

Maes, P. (1989). How to Do the Right Thing. Connection Science. 1(3). 291-323.

Manikonda, V., Hendler, J., and Krisnaprsad P.S. (1995). Formalizing Behavior-based Planning for Nonholonomic Robots. Proceedings of the International Joint Conference on Artificial Intelligence. Montreal, Canada. 563-570.

Martin, F.G. (1999). The Handy Board Technical Reference. Massachusetts Institute of Technology, United State of America.

Pirjanian, P. (1998). Multiple Objective Action Selection and Behavior Fusion Using Voting. Aalborg University, Denmark: PhD Thesis.

Pirjanian, P. and M. Mataric. 1999. Multiple Objective vs. Fuzzy Behavior Coordination. In Lecture Notes in Computer Science, Fuzzy Logic in Mobile Robot Navigation.

Riekki, J. and Roning, J. (1997). Reactive Task Execution by Combining Action Maps. Proceedings of the International Conference on Integrated Robots and Systems (IROS). Grenoble, France. 224-230.

Rosenblatt, J.K. (1995). DAMN: A Distributed Architecture for Mobile Navigation. AAAI Spring Symposium on Software Architectures for Physical Agents. Stanford CA. 167-178.

Rosenblatt, J.K. (1997). DAMN: A Distributed Architecture for Mobile Navigation. Carnegie Mellon University, United States: PhD Thesis.

Saffiotti , A., Konolige, K., and Ruspini, E.H. (1995). A Multivalued Logic Approach to Integrating Planning and Control. Artificial Intelligence. 76 (1-2). 481-526.

Tunstel, E. (1995). Coordination of Distributed Fuzzy Behaviors in Mobile Robot Control. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Vancouver, BC Canada. 4009-4014.

Cutting Edge Robotics

Edited by Vedran Kordic, Aleksandar Lazinica and Munir Merdan

ISBN 3-86611-038-3

Hard cover, 784 pages

Publisher Pro Literatur Verlag, Germany

Published online 01, July, 2005

Published in print edition July, 2005

This book is the result of inspirations and contributions from many researchers worldwide. It presents a collection of wide range research results of robotics scientific community. Various aspects of current research in robotics area are explored and discussed. The book begins with researches in robot modelling & design, in which different approaches in kinematical, dynamical and other design issues of mobile robots are discussed. Second chapter deals with various sensor systems, but the major part of the chapter is devoted to robotic vision systems. Chapter III is devoted to robot navigation and presents different navigation architectures. The chapter IV is devoted to research on adaptive and learning systems in mobile robots area. The chapter V speaks about different application areas of multi-robot systems. Other emerging field is discussed in chapter VI - the human- robot interaction. Chapter VII gives a great tutorial on legged robot systems and one research overview on design of a humanoid robot.The different examples of service robots are showed in chapter VIII. Chapter IX is oriented to industrial robots, i.e. robot manipulators. Different mechatronic systems oriented on robotics are explored in the last chapter of the book.

INTECH

open science | open minds