Jurnal Teknologi, 36(D) Jun 2002: 55–70

USING VOTING TECHNIQUE IN MOBILE ROBOT BEHAVIOR COORDINATION FOR GOAL-DIRECTED NAVIGATION

TAN CHEE KWONG[1], SHAMSUDIN H.M. AMIN[2], ROSBI MAMAT[3] & JOZSEF K. TAR[4]

Abstract. In behavior-based mobile robot, the control strategy is distributed among a set of specialized behaviors. Each behavior with particular objective runs completely independently to send commands to control the mobile robot. However, behavior with different objectives may generate conflicting command. Therefore, behavior coordination is an important issue. An intelligent voting technique is implemented to solve this problem. Each behavior votes for a set of possible actions, with vote zero is the least desired action and vote one is the most desired action. The behaviors send votes as a possibility for each action set to achieve the objectives of the behaviors. An arbiter then performs command fusion and selects the most favored action that is pareto-optimal. This will solve the action selection problem and improve the probability to succeed. This technique has been implemented on UTM AIBOT mobile robot. The experimental results are presented and the reliability of the technique is shown.

Keywords: Voting technique, behavior coordination, action selection, mobile robot, goal directed navigation

Abstrak. Dalam robot mudah alih dengan kelakuan, pengawalan robot dibahagikan kepada sekumpulan kelakuan yang tertentu. Setiap kelakuan dengan objektif tertentu beroperasi secara bebas untuk menghantar arahan bagi mengawal robot. Namun begitu, kelakuan dengan objektif yang berlainan mungkin berkonflik. Oleh hal yang demikian, koordinasi di antara kelakuan merupakan suatu isi penting. Suatu teknik mengundi telah digunakan untuk mengatasi masalah ini. Setiap kelakuan mengundi untuk suatu set tindakan. Undi sifar menandakan tindakan yang tidak diingini manakala undi satu menandakan tindakan yang paling diingini. Kelakuan menghantar undi untuk mewakili kebolehan tindakan itu untuk mencapai objektif kelakuan. Suatu penentu akan mengumpulkan semua tindakan ini dan pilih tindakan yang paling sesuai dengan memenuhi syarat pareto-optimal. Ini dapat menyelesaikan masalah pemilihan tindakan dan meninggikan kebarangkalian untuk berjaya. Teknik ini telah digunakan dalam robot mudah alih UTM AIBOT. Keputusan eksperimen dan kebolehpercayaan teknik ini telah ditunjukkan.

Kata Kunci: Teknik mengundi, koordinasi kelakuan, pemilihan tindakan, robot mudah alih, pelayaran berpandukan matlamat

[1, 2 & 3] Department of Mechatronics and Robotics Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor. e-mail: ckt77@hotmail.com; sham@fke.utm.my; rosbi@fke.utm.my

[4] John von Neumann Faculty of Informatics, Budapest Polytechnic,H-1081 Budapest, Nepszinhaz utea 8, Hungary.

1.0 INTRODUCTION

In order to achieve a goal-directed navigation task in an unstructured, unknown, or dynamic environment, a mobile robot must be able to perceive its surroundings and generate appropriate actions. Mobile robots need to gather and combine the information from various sources. They will then generate an appropriate action based on the information. This is the main challenge in mobile robot navigation where uncertainty and real-time responses are the major issues. 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 behavior-based architectures.

Behavior-based architecture has emerged as an alternative to traditional approaches for designing autonomous mobile robots. The methodology is based on a biologically inspired philosophy that favors parallel, decentralized architectures, and allows some freedom of interpretation. It enables fast real-time responses through simple behaviors, each of which achieves a specific goal. It also distributes the representation and computation over more sophisticated concurrent behavior process rather than employing a centralized representation. Based on the sensory information, each behavior produces commands to control the robot. However, behavior with different objectives may produce conflicting actions. The objective of one behavior might be in contrast to the objectives of others. Thus, it creates the problem of how to coordinate multiple behaviors together and decision-making at each point in time. This is known as the action selection problem. The following is the definition of Action Selection Problem (ASP) from [1]:

"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?"

An intelligent voting technique is introduced to solve the problem in multiple behavior coordination. The main distinction of this method is that it takes a discrete set of possible motor commands into consideration, rather than computing only a single motor command. Multiple behavior modules concurrently share control of the robot by generating votes for every possible motor command. The generated votes are between 0 and 1; vote zero being the least desired action and vote one is the most desired action. The voting scheme is actually a mapping from perception to action. Each behavior generates the votes in a manner to show the possibility for that action to achieve behavior's goal. This will enable the robot to deal with uncertainty in perception and incomplete information about the environment. A center arbiter will then perform command fusion to choose the most appropriate action that is pareto-optimal.

2.0 RELATED WORKS

Various techniques have been proposed for behavior coordination to solve the action selection problem. From the literature, the action selection mechanism had been divided into two main groups, known as arbitration and command fusion respectively [2]. Arbitration mechanisms are used for arbitrating between the set of active behaviors at any moment according to the system's changing objectives. These include priority-based, state-based and winner-take-all approaches. Meanwhile, command fusion mechanisms coordinate the activities of the set of behaviors that are active simultaneously by selecting the action that best satisfies the system's goal. These can be further divided into superposition, fuzzy and voting approaches.

The subsumption architecture is the most well known technique that employs a priority-based arbitration mechanism [3]. In this architecture, priorities are assigned to each behavior. Behaviors with higher priorities are allowed to override the output of behaviors with lower priority via an inhibition link. In state-based arbitration, systems are modeled in terms of finite state automata (FSA). At each state, a distinct behavior is activated and perceptual triggers cause transitions from one state to another. These mechanisms include Discrete Event System [4] and Temporal Sequencing [5]. In winner-take-all mechanisms, action selection results from the interaction of a set of distributed behaviors that compete until one behavior wins the competition and takes control of the robot [1]. The behaviors also receive activation and inhibition based on the current state of the world and on the current goals of the system, effectively determining a dynamic prioritization of behaviors. All these arbitration methods solve the problem of conflicting behavior by having one behavior's command completely override another's. However, they do not provide an adequate means for dealing with multiple goals that can be satisfied simultaneously.

Potential field approach presents a method of action selection based on vector sums of the potential fields, such as repulsive fields from obstacles and attractive fields from goals [6]. Work on motor schemas has been inspired by the potential field approach [7]. The motor schema framework allows multiple behaviors to be instantiated and their outputs combined for a variety of different tasks. However, it suffers from the problem of local minima and command averaging. Fuzzy command fusion mechanisms use fuzzy logic to generalize the behavior module in classical logic to allow a conclusion to be drawn from a fuzzy if-then rule [8, 9]. Each fuzzy conclusion is then defuzzified, resulting in a control command. These work allow the sensory data to be interpreted into a multi-value logic and therefore deal with environment uncertainties. However, fuzzy defuzzification may sometimes generate counterintuitive results for action selection [10].

Distributed Architecture for Mobile Navigation (DAMN) is the most well known architecture that using voting technique for behavior coordination [11]. It is similar to fuzzy logic approach. However, DAMN is more general because it is not rule-based but rather behaviors evaluate input and produce votes for every possible action. Each behavior generates a vote between –1 and +1 for every possible steering command, with negative votes being against and positive votes for a particular command option. An arbiter will then select the most appropriate action with the maximum vote value.

SAMBA is another voting approach that is largely influenced by the ideas of DAMN [12]. Primitive behaviors produce reaction in form of primitive action maps. An action map specifies preferences for each possible action. The action maps are forwarded to a command arbiter and action with the highest gain will be chosen. In Multiple Objective Action Selection (MOAS), the principles of Multiple Objective Decision Making are used to formulate command fusion techniques for behavior-based control [12]. Each behavior calculates an objective function over a set of possible actions. Multiple behaviors are blended into a single more complex behavior that seeks to select the action that satisfies the objectives as good as possible.

3.0 VOTING ALGORITHM

Voting technique provides a framework for the behaviors to generate multi-valued output. Traditional behavior design produces a single-valued output that is optimal from the point of each behavior. If the behaviors are in conflict then it is difficult to make a good decision based on single-valued behavior. Multi-valued output provides a better solution for this problem. A set of possible actions would be taken into consideration in the decision-making. The preferences of each action are represented by the vote value. Meanwhile, multi-valued behaviors also take into account the uncertainty of environment, sensors and actuators.

3.1 Uncertainty Handling

Mobile robots that navigate in dynamic environments may encounter with several problems, such as incomplete knowledge of the environment, unpredictable environment changes, imperfect sensors and imperfect actuators. Therefore, uncertainty handling is a major issue in mobile robot navigation to improve the reliability.

Probability methods have been widely used in mobile robot for uncertainty handling especially for localization, mapping, path planning, and navigation [13]. Most of the probability methods are used to build a reliable internal model for the mobile robot. However, voting technique provide a model free method. Thus, the computational complexity in model building can be avoided. The voting scheme is actually a direct mapping from perception to action. Each behavior generates the votes in such manner to show the possibility for that action to achieve behavior's goal. The relation between action set, X, and the behavior, b, are shown below,

$$b : X \quad \rightarrow \quad [0,1] \tag{1}$$

The action space, $X = \{x_1, x_2, ..., x_m\}$, is a finite set of possible actions. The action is described in $x = (\nu, \omega)$, where the control parameters consists of translation and angular velocities. The mapping in equation (1) assigns to each action $x \in X$ a preference, where the most appropriate actions are assigned 1 and undesired actions are assigned 0. Each vote represents the possibility for that action to achieve the behavior's objective.

This method is similar to fuzzy logic approach that is described by the fuzzy membership function. Membership values in the interval [0, 1] indicate a measure of uncertainty or imprecision associated with the element and the belief that "The world is not black and white but only shades of grey" [14]. Commonly used function shapes are triangular, trapezoidal, singleton, positively sloped ramp, and negatively sloped ramp. Several researchers had also combined voting techniques with fuzzy logic approach to improve DAMN and MOAS techniques [15, 16]. However, the assignments of preferences to action set in voting technique are not limited in a membership function. For example the obstacle avoidance behavior can generate the votes as inversely proportional to the distance to the nearest obstacle.

3.2 Voting Selection Scheme

As stated in the proceeding section, the behavior will vote for a finite set of possible actions. Thus, the problem in behavior coordination is to find the most appropriate action that can achieves the overall goal. In MOAS, the problem is formulated as finding the most appropriate action that maximizes the objective function,

$$\text{Max } \,|\, o_1(x),\, o_2(x),\, \ldots,\, o_n(x) \,| \tag{2}$$

where $x \in X$, and behaviors are designed as objective functions, $o_1(x)$, $o_2(x)$, ..., $o_n(x)$. In voting technique, the problem of finding the most appropriate action can be defined as finding the action that takes all behaviors' goals into consideration based on the votes generated by each behavior.

In [17], an overview of various classes of voting schemes is given. The most general voting schemes are majority voting and m-out-of-n voting, which belong to the group of weighted consensus voting. In majority voting, an action is chosen if it receives more than half of the total number of votes. In m-out-of-n voting, an action is selected if it receives m or more votes out of n. Plurality voting has a higher probability of choosing the correct action. In plurality voting, the candidate with the most first place votes win. The winner does not have to receive a majority of the first place votes. In behavior-based mobile robot, each behavior is allowed to vote for a set of possible actions. This is known as approval voting in which voters can vote for as many candidates as they wish. In order to reflect the priority of each behavior in controlling the robot, weighted voting system is applied. A weighted voting system is one in which the preferences of some voters carry more weight than the preferences of other voters.

In short, the voting scheme used here can be concluded as weighted voting system with approval voting for each voters and plurality voting in selecting the winner. Therefore, the most appropriate action can be found by solving the following equation,

$$x = \arg\max \textstyle\sum_i^n W_i * b_i(x), \text{ with } x \in X \tag{3}$$

W_i are normalized weights with the total sum of 1,

$$\sum_i^n W_i = 1 \tag{4}$$

Therefore, x is a pareto-optimal solution if x is a unique solution for (3).

4.0 VOTING TECHNIQUE IMPLEMENTATION

The voting technique takes an approach where multiple modules concurrently share control of the robot. It uses a scheme where each behavior votes for a discrete set of motor commands and each set of motor command consist of the robot turn radius together with its speed. Figure 1 shows the architecture for the intelligent voting technique.

Various behaviors are designed for the mobile robot. These behaviors can be ranged from low-level behavior such as obstacle avoidance behavior, goal-seeking behavior to high-level behavior such as landmark detection behavior, path-planning behavior, and learning behavior. Each behavior is responsible for achieving some particular tasks. Each of them runs completely independently and asynchronously while generates a vote between 0 and 1 for every possible motor command, with vote zero is the least desired action and vote one is the most desired action. Each behavior is assigned a weight reflecting its priority in controlling the vehicle. An arbiter then performs command fusion to select the most appropriate action.

To implement the voting technique in a mobile robot, the design procedure are shown as follows,

i. Determining the types of behaviors and objectives of each behavior, that is determining the number of voters and the weights of each voter.
ii. Determining the possible command set, that is determining the number of candidates.

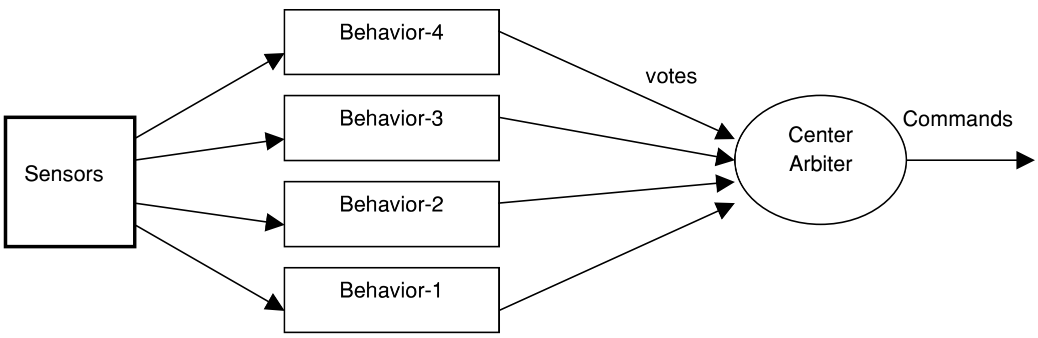


Figure 1 The mobile robot architecture for intelligent voting technique

iii. Design of behaviors, that are designing the mapping function in sending votes for each behavior.
iv. Design of center arbiter, that is designing the module for selecting the most appropriate action by solving equation (3).

4.1 Determining the Types of Behaviors and Objectives of Each Behavior

Behaviors are the basic module in mobile robots with behavior-based architecture. The type of behaviors and the objective of each behavior have to be decided. Behaviors can vary from primitive reactive behaviors to deliberative planning behaviors, $B = \{b_1, b_2, ..., b_n\}$ with B is the behavior set with total n amount of different behavior module b. Each behavior has a specific objective to achieve in the system. In order to achieve a goal-directed navigation, two behaviors have been designed for the mobile robot. These behaviors are obstacle avoidance behavior and goal-seeking behavior. The main objective of the obstacle avoidance behavior is to avoid collision with any obstacles while the goal-seeking behavior responsible in looking for the goal point and try to approach it. These two behaviors have their own objectives and operate completely independently without any communication between them. The critical point here is how to determine the weight for each behavior. In Subsumption Architecture, obstacle avoidance behavior has the first level of competence to ensure the safety of the mobile robot [3]. Therefore, to reflect the priority of obstacle avoidance behavior, it should have a weight greater than goal-seeking behavior. That means $W_{obs} > W_{goal}$ where W_{obs} is the weight for obstacle avoidance behavior and W_{goal} is the weight for goal-seeking behavior. Meanwhile the total weights for these behaviors are normalized to 1.

4.2 Determining the Possible Command Set

In voting technique, the possible command set is actually the candidate, $X = \{x_1, x_2, ..., x_m\}$ with X is the command set with m amount of different motor command x. Different kind of robot may have different kind of command set. In this paper, a differential steering vehicle is used. Therefore, the command set is in the domain of differential steering vehicle. A differential steering vehicle is a vehicle with two wheels mounted on a single axis and independently powered and controlled by different motors. Additional passive wheel, usually casters, is provided for support. If both drive wheels turn with same velocity, the robot moves in a straight line. If one wheel turns faster than the other, the robot follows a curved path. Each control command set thus describes a circular trajectory. For example, the command set can be the path shown in Figure 2 and named as HardLeft, Left, SoftLeft, Forward, SoftRight, Right, and HardRight respectively.

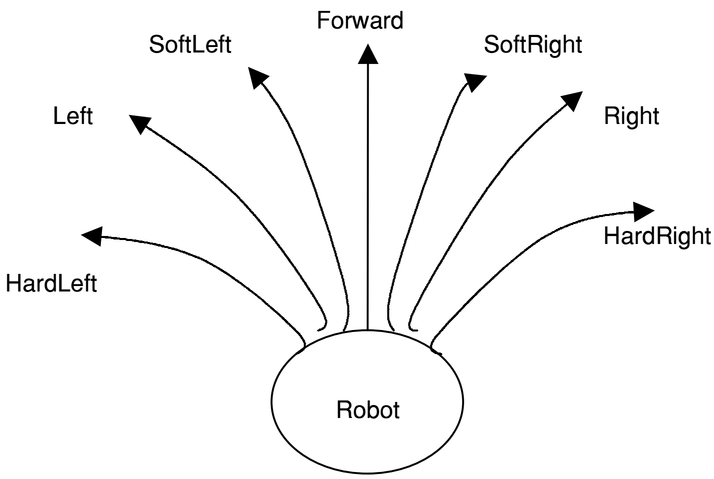


Figure 2 Seven motor commands for the robot

4.3 Design of Behaviors

Behavior is actually a mapping from perception to action. Equation (1) shows that each behavior will vote for all the possible command set. The preferences are in the interval of [0, 1]. Two behaviors are designed here, which are obstacle avoidance behavior and goal-seeking behavior.

4.3.1 Obstacle avoidance behavior

The objective of this behavior is to move the robot at a safe distance from the obstacles. Sonar sensors provide reading of the current obstacle distance. The sonar readings are directly generated as vote in a probability manner rather than building a local map. Let s be the free state and s' be the obstacle state, the probability for the robot to be in a free state from the view of a sonar reading is,

$$p(s) = \begin{cases} 1 & \text{if} \quad D > D_{\max} \\ (D - D_{\min})/(D_{\max} - D_{\min}\} & \text{if} \quad D_{\min} < D < D_{\max} \\ 0 & \text{if} \quad D < D_{\min} \end{cases} \tag{5}$$

where D is the distance detected by that sonar sensor. Meanwhile, $D_{\max}$, the maximum distance to detect, set as free state, and $D_{\min}$, the minimum distance to detect, set as obstacle state. The behavior will vote for the probability to move to a free state. Each sonar reading will denote a weight to every motor command. Thus, the probability for each motor command to bring the robot to a free state is,

$$P_m(s) = \sum_n W_{mn} \ p_n(s) \tag{6}$$

m motor command index from M motor command
n sonar index from N sonar
$P_m(s)$ probability for motor command m to bring the robot to a free state
W_{mn} weight for sonar n from the view of motor command m
$p_n(s)$ probability for the robot to be in a free state from the view of sonar n

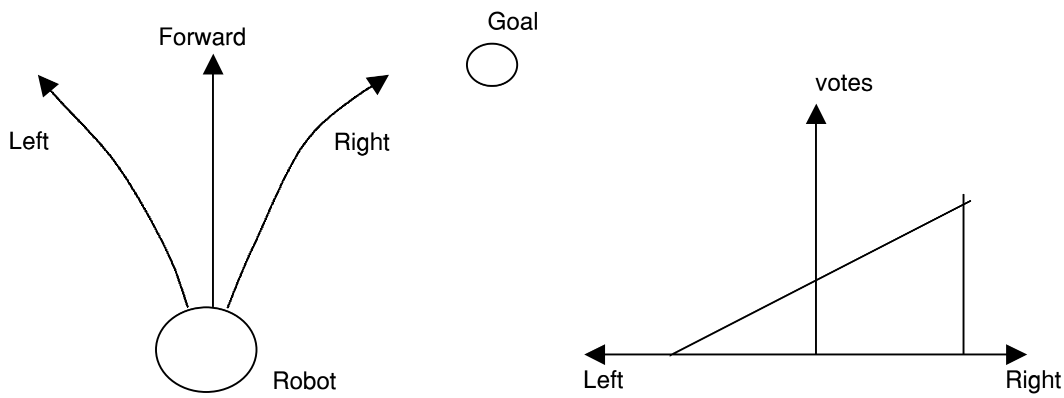
The vote generated in the obstacle avoidance behavior can be shown as the probability of that motor command to bring the robot to a free state,

$$b_{obs}(x_m) = P_m(s) \tag{7}$$

4.3.2 Goal-seeking behavior

The task for goal-seeking behavior is to look for a goal point and try to approach it. This simple behavior directs the robot toward the goal point from an incomplete and uncertain model of the environment. The method is similar to fuzzy logic rules used in other robotics application. It determines the desired turning angle in three steps. First, the robot perceives the information of a goal point, and then locates a target point direction. Second, the behavior broadens the specific target direction into a general desired direction. This will give the robot more flexibility in navigation and also consider the uncertainty in environment. Third, the behavior will vote for each possible command options according to their angle to the goal. A trajectory that will lead the robot toward the goal will get a higher vote and vice versa.

The goal point is in the direction of right as shown in Figure 3*a*. The goal-seeking behavior will broaden the target direction as shown in Figure 3*b*. It will vote for traveling along the arc on the right. The other command options will have less desired vote depending on their direction from the goal point.



(*a*) (*b*)

Figure 3

4.4 Design of Center Arbiter

The center arbiter plays an important role in command fusion and action selection. It is actually a process to solve equation (3) to find a pareto-optimal solution. The arbitration process used in the intelligent voting technique is illustrated in Figure 4.

The process can be divided into few steps. First, each behavior will send their votes for every single motor command to center arbiter. The arbiter will collect the votes for all the command. At this stage, the votes from different behaviors will be fused together using weighting method. The formula is,

$$V_{\text{sum}}[a] = \sum_b V_b[a] * W_b \tag{8}$$

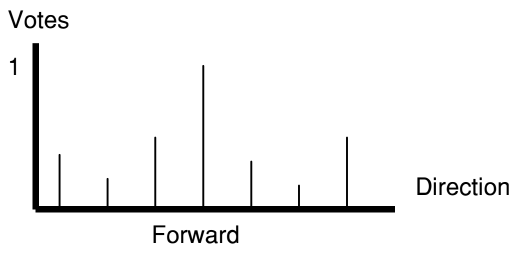


Figure 4*a* Behavior A, weight = 0.7, desired direction = forward

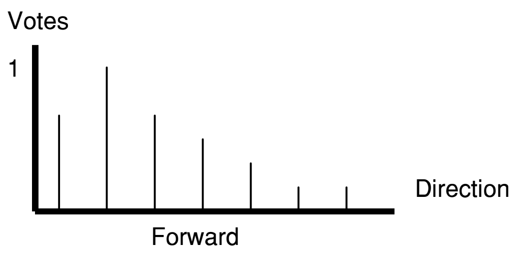


Figure 4*b* Behavior B, weight = 0.3, desired direction = left

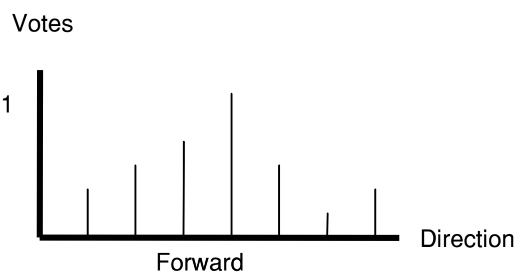


Figure 4*c* Weighted sum, maximum vote = forward

a	command index from A command
b	behavior index from N behavior modules
$V_b[a]$	vote for action a by behavior b
W_b	weight for behavior b

Now, the arbiter will find the command with maximum vote value V_M.

$$V_M = \text{argmax } \{V_{\text{sum}}[a]\} \tag{9}$$

Finally, this command will be chosen and sent to the motor controller for execution.

5.0 EXPERIMENTAL RESULTS AND DISCUSSION

The intelligent voting technique has been implemented and tested on the UTM AIBOT (Figure 5). AIBOT is a differentially steered mobile robot equipped with sonar sensors for obstacle avoidance behavior and light sensors for goal-seeking behavior. It can travel at an average speed of 20 cm/seconds. There are seven motor commands for the robot to vote for. The seven command directions are named as HardLeft, Left, SoftLeft, Forward, SoftRight, Right, and HardRight respectively.

In the experiments, the weights were 0.7 and 0.3 for the obstacle avoidance and goal-seeking behavior respectively. The weights were set empirically. The obstacle weight was larger to reflect that avoiding obstacles is more important than approaching the goal. Figure 6 and Figure 7 illustrate the result of two experiments carried out

Figure 5 The UTM AIBOT

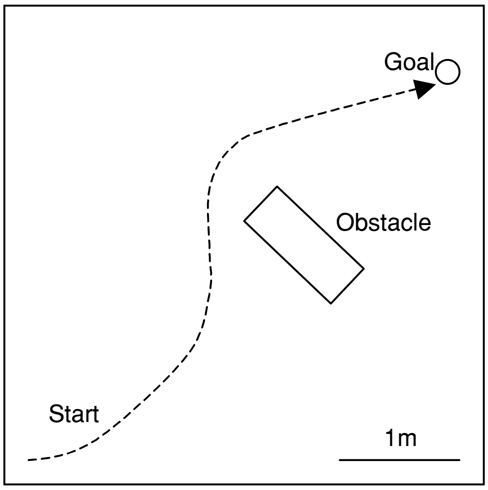


Figure 6 Navigation to goal point with an obstacle in the middle

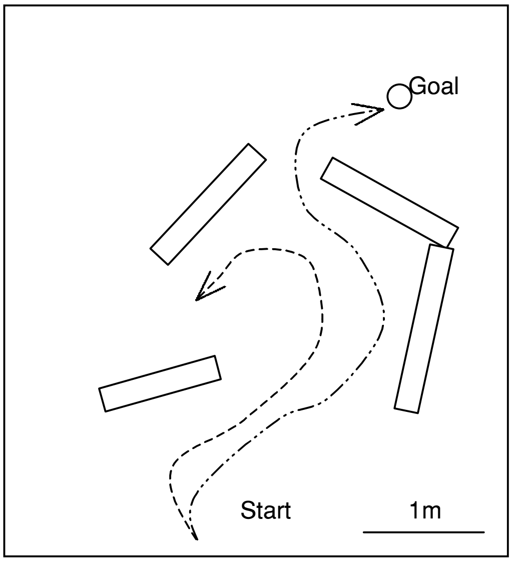


Figure 7 Navigation in a cluttered environment

for testing the intelligent voting technique. In each experiment, the robot will run in the field for 50 times. Thus, a total of 100 experimental runs were carried out using the same technique in different fields. The experimental results are shown in Table 1. The reliability of the navigation is calculated as the percentage of success navigation from the total navigation in each experiment.

Figure 6 shows an example of traveling along a path in a free environment towards a goal point with only one obstacle in the middle of the path. At the beginning, the obstacle avoidance behavior will vote equally for each path because it detects no

Table 1 Experimental results for robot navigation

Experiments	**Total tests**	**Success navigations**	**Reliability**
Navigation with an obstacle	50	42	84%
Navigation in cluttered environment	50	36	72%

obstacle. Meanwhile, the goal-seeking behavior 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 behavior will vote for a free path. Although the goal-seeking behavior will vote for the forward move, AIBOT still take a turn because of the greater weight of the obstacle avoidance behavior. By the time AIBOT passes the obstacle, all the paths are free and will get equal vote from obstacle avoidance behavior. Therefore, it will take a right turn to go to the goal point as voted by the goal-seeking behavior. The robot will be considered fail if it cannot achieve the goal point or collides with the obstacle during navigation.

Figure 7 shows another example of AIBOT goal-directed navigation in a cluttered environment. Two navigation paths have shown in the figure. AIBOT could achieve the navigation from starting point to the goal point without collision with any obstacle. However, it will sometimes miss the narrow passage due to the inherent limitation in the obstacle avoidance behavior. Since the obstacle avoidance behavior generates its vote with no map about the environment, the uncertainties of the environment may sometimes suggest the robot to take a turn although there is a narrow passage in front. The robot will only be considered success if it can navigate to the goal point through the narrow passage without any collision with the obstacles.

Two experiments had shown a successful action selection technique in a behavior-based mobile robot in achieving an indoor goal-directed navigation. This technique has some advantages such as the ability to control the speed and direction simultaneously, handle trade offs between safety and goal-directedness, and enable navigation in dynamic environment without a priori map. Furthermore, local map is not needed and sensor fusion is not necessary. This can avoid bottlenecks in sensor fusion and decision-making and is therefore able to respond in real-time to external events. The voting technique also has the advantage of the ability to account for uncertainty in the interpretation of sensor data. The evaluation of votes enables the robot for safety navigation in unknown and uncertain environment.

However, from the experimental results shown in Table 1, navigation in cluttered environment has a lower reliability. This is caused by some problems in the voting technique. Both the obstacle avoidance and goal-seeking behavior do not hold any memory of the environment. This might cause information loss in the process and therefore perform non-optimal path navigation. Besides, the motor command has been discretized in several sets and might cause the robot to miss the best way. If the best way is not lie on the command set, the robot will miss it. Sometimes, it may also trap in a local minima problem caused by information loss.

6.0 CONCLUSION

An intelligent voting technique for behavior coordination in mobile robot goal-directed navigation for indoor environment has been presented. With the voting technique, uncertainties tolerating basic behaviors are designed and able to coordinate execution of multiple behaviors to achieve overall goal. For indoor goal-directed navigation, the robot architecture is divided into two behaviors, which are obstacle avoidance behavior and goal-seeking behavior. Each of them operates asynchronously to send vote to the center arbiter for command fusion. Future works based on this method can be carried out by adding a short-term memory behavior for planning a smoother navigation with optimal path. The concept of homogeneous behaviors suggested in [2] may be implemented to increase the reliability in navigation.

ACKNOWLEDGEMENTS

The research reported here was done at the UTM Robotics Laboratory. Support for this research was provided by Malaysian Ministry of Science, Technology and the Environment under the Intensification of research in Priority Areas (IRPA project number: 09-02-06-250).

REFERENCES

[1] Maes, P. 1989. How to Do the Right Thing. *Connection Science Journal*. 3(1): 291 – 323.

[2] Pirjanian, P. 1998. Multiple Objective Action Selection and Behavior Fusion Using Voting. *PhD thesis, Laboratory of Image Analysis, Department of Medical Informatics and Image Analysis, Aalborg University, Denmark*.

[3] Brooks, R. A. 1986. A Robust Layered Control System for a Mobile Robot. *IEEE Journal Robotics Auromat*. 7(2): 14 – 23.

[4] Kosecka, J., and R. Bajcsy. 1993. Discrete Event Systems for Autonomous Mobile Agents. *In Proceedings Intelligent Robotic Systems '93 Zakopane*. 21 – 31.

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

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

[7] Arkin, R. C. 1989. Towards the Unification of Navigational Planning and Reactive Control. *In Proceedings of the AAAI Spring Symposium on Robot Navigation*. 1 – 5.

[8] Saffiotti, A., K. Konolige, and E. H. Ruspini. 1995. A Multivalued-logic Approach to Integrating Planning and Control. *Artificial Intelligence*. 1-2(76): 481 – 526.

[9] Tunstel, E. 1995. Coordination of Distributed Fuzzy Behaviors in Mobile Robot Control. *In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. 4009 – 4014.

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

[11] Rosenblatt, J. 1995. DAMN: A Distributed Architecture for Mobile Navigation. *In Proceedings of the AAAI Spring Symposium on Software Architecture for Physical Agents*.

[12] Riekki, J., and J. Roning. 1997. Reactive Task Execution by Combining Action Maps. *In International Conference on Integrated Robots and Systems (IROS)*. 224 – 230.

[13] Thrun, S. 2000. Probabilistic Algorithms in Robotics. *Technical Report, CMU-CS-00-126*, Computer Science Department, CMU.

[14] Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control*. (12): 338 – 353.

[15] Yen, J., and N. Pflugger. 1995. A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation. *IEEE Transaction on System, Man, and Cybernetics*. 6(25): 971 – 978.

[16] Pirjanian, P., and M. Mataric. 1999. A Decision Theoretic Approach to Fuzzy Behavior Coordination. *In Proceedings IEEE International Symposium on Computational Intelligence in Robotics & Automation (CIRA99)*.

[17] Parhami, B. 1994. Voting Algorithm. *IEEE Transaction on Reliability*. 4(43): 617 – 629.