MY0101649

MALAYSIAN SCIENCE & TECHNOLOGY CONGRESS '98
Symposium C: Computer Science & Information Technology
Universiti Sains Malaysia, Pulau Pinang, 10 -11 November 1998

BEHAVIOR-BASED OBSTACLE AVOIDANCE CAPABILITY FOR BIOLOGICALLY INSPIRED EIGHT-LEGGED WALKING ROBOT

Izzeldin Ibrahim Mohd — izzeldin@suria.fke.utm.my
Shamsudin H.M. Amin — sham@suria.fke.utm.my
Adel Ali S. Al-Jumaily — adel@suria.fke.utm.my

Center for Artificial Intelligence and Robotics (CAIRO)
Faculty of Electrical Engineering
University of Technology Malaysia
Locked Bag 791,
80990 Johor Bahru, Malaysia
Fax: + 607 5566272

ABSTRACT

Behavior-based approach has proven to be useful in making mobile robot working in real world situations. Since the behaviors are responsible for managing the interaction between the robots and its environment, observing their use can be exploited to model these interactions. A real-time obstacle avoidance algorithm has been developed and implemented. This algorithm permits the detection of unknown obstacle simultaneously with the steering of the mobile robot to avoid collisions and advance toward the target. In our approach the robot is initially given a set of "behavior-producing" modules to choose from, and the algorithm provides a memory-based approach to dynamically adapt the selection of the behaviors according to the history of their use. We developed a set of algorithms, which uses Subsumption Architecture (SA) for controlling an eight-legged walking robot operating in closed vicinity. This paper describes a successful application of these algorithms to Oct-Ib robot and experimental results of the robot navigating in complex environment.

Key Words: Behavior-Based, Navigation, Collision avoidance

1 Introduction

Obstacle avoidance is one of the key issues to successful applications of the mobile robot systems. All mobile robot features some kind of collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot a short of it in order to avoid a collision through sophisticated algorithms, that enable the robot to detour the obstacles. The latter algorithms are more complex, since they involve not only the detection of obstacles, but also some kind of quantitative measurements concerning the obstacle's dimensions. Once these have been determined, the obstacle avoidance algorithm needs to steer the robot around the obstacle and resume the motion toward the original target [1].

A key issue in applying an autonomous robot to new fields is robot navigation in uncertain and complex environment. If a mobile robot moves among unknown obstacles to reach a specific target without collisions, sensors must be used to acquire information about the real world. Using such information, however, it is very difficult to build a precise and complete model in the real-time for the pre-planning a collision-free path. On the basis of the stimulus-response behavior in bio-systems, behavior-based control [2] has been proposed for mobile robot navigation. Since this control approach does not need building an exact world model and complex reasoning process, it is suitable for robot navigation in dynamic environment [3].

Robots operating close to each other in an unstructured environment require fast and decisive control to avoid hitting each other or stationary objects. Traditionally, the scientific method used to deal with such a problem has been Control System Theory. In recent years, AI techniques have been involved [4] but both control system theory and AI-based control been effective only in situations where the operational environment does not change or changes only a

INIS-MY-130

little with respect to time and space. This greatly limited the applicability of such approaches to practical real world problems. Recently a new wave of research called Behavior-Based has been introduced [5, 6]. In section 2 behavior-based system approach is presented. General description of Oct-Ib system overview is presented in section 3. Oct-Ib behavior control systems is presented in section 4. In section 5 we presented the Oct-Ib obstacle avoidance algorithm. Then we presented experimental results in section 6. The paper is concludes with a brief summary and a discussion on future work.

2 Behavior-Based Systems

Behavior-based system architectures are now generally accepted as an effective basis for high performance mobile robots. Its main characteristics are active perception of the robot's dynamically changing environment recognition and evaluating its current situation, and dynamic selection of behavior appropriate for the actual situation. Learning in mobile robot domain is very challenging tasks, especially in non-stationary conditions. The behavior-based approach has proven to be useful in making mobile robot working in real world situations. Since the behaviors are responsible for managing the interaction between the robots and its environment, observing their use can be exploited to model these interactions. In our approach the robot is initially given a set of "behavior-producing" modules to choose from, and the algorithm provides a memory-based approach to dynamically adapt the selection of the behaviors according to the history of their use.

In behavior-based systems, the robot controller consists of a collection of behaviors, each of which achieves and/or maintains a specific goal. For example, the "avoid-obstacles" behavior maintains the goal of preventing collisions with objects in the environment, and "go-home" behavior achieves the goal of reaching some home region. Each behavior is a processing element or a procedure, also called a control law in the engineering field of Control Theory.

Behaviors are activated in response to external and/or internal conditions, sensors input and external state. The system as a whole activates entire subsets of behaviors so that parallelism can be exploited, both in the speed of computation and in the resulting dynamics. Inspired by biological organisms, designer of behavior-based systems exploited these dynamics to create repeatable stable, and, ultimately, intelligent behavior without relying on top-down, centralization and often even hierarchical control [5, 7].

A key issue in the behavior-based systems concerns the coordination of the multiple behaviors, thus making "arbitration", i.e., deciding what behavior to execute at each point in time. For the sake of the simplicity most implemented systems use a built-in, fixed priority ordering behaviors [7].

3 OCT-Ib System Overview

OCT-Ib is an eight-legged walking robot, with each leg approximately 114 mm long. The width of the robot is 394 mm with the leg stretch out, when is standing erect its total height become 168 mm and 610 mm long, including the whisker. It can be used for legged navigation over rugged terrain, entertainment/educational applications and research into legged locomotion and evolutionary robotics experiments. Its normal operation weight is approximately 4 Kg with a maximum payload of 1.8 Kg. It travels at approximately 0.5ft/sec and can climb over an obstacles about 1.5 inches height. Figure (3.1) is the schematic diagram of the Oct-Ib robot [8].

Each leg of OCT-Ib is driven by the Futaba S9402 servomotor, one gives the lifting movement and the other gives the swinging motion. The lifting torque is designed to be twice that of the swinging torque. Left and right primary whiskers stretch diagonally in the front of the robot and are used to detect obstacles ahead of its path. Each whisker is spring loaded and has its own open/close circuitry. Stainless steel wires "secondary whiskers", each approximately 60 cm long, were attached in a circular fashion between the head and the tail of the robot, the two contact "head and tail" are each equipped with a micro switch which detects contact at any point on the wire. There is also small wire loop in the very rear of the robot, which forms a bumper to detect a contact from the behind.

Light sensors are used to detect the level of the ambient light surrounding the robot in every direction, a large numbers of CDs photo sensors were installed.

Infrared (IR) sensors are used to detect signals emitted from humans and other warm bodied creatures and are also used to detect other robots when operating as a group.

Motorola MC68332 single chip controller is used as the "brain" of the robot. MC 68332 are a 32 bit 16.78 MHz processor with on-board memory, counters, A/D converters and other control circuitry allow it to operate as stand-alone control computer. It is accomplished by 64 Kbytes EPROM and up to 1 Mbytes of RAM.

The OCT-Ib robot was programmed fully in a behavior-based manner. This means that the entire software structure consists of a numbers of simple behaviors as opposed to a rigidly structured algorithm as in most computer software today. The OCT-Ib control program consists of two main parts, the low level time driven functions and high level or coordinate functions. The time functions implement the basic behaviors such as walk, seek food, seek darkness, and watch for the enemy. The high level behaviors activates or suppresses lower level behaviors depending on the external and internal stimuli.

The executed code was developed using GNU C and associated tools "assembler, linker, etc". Currently, a limited number of library routines have been included in the on-board ROM. These routines handle initialization, serial I/O, sensor readings, setting leg positions, and interrupt handling. The main hardware features of the Oct-Ib robot are given in the table (3.1). [8].

TABLE (3.1) HARDWARE FEATURES

Actuators	8 servo motors type S9402 "lifting motion" up/down 8 servo motors type S9402"swinging motion" front/back
Sensors	8 infrared sensors (IR) 15 light sensors 11 whisker switches 16over-current sensors
Batteries	4Ni-Cd cell, 4.8 volt battery for electronic 2 of 4Ni-Cd cells, 4.8 volt battery in parallel for motors
Computer system	Motorola MC68332 micro controller, 32 bit, 16.78 MHz processor with 64 K bytes EPROM and 1 M bytes RAM

4 Behavior Based Control

Behavior based control is a method for controlling robots in a complex environment. The basic idea is to break up the problem in terms of the behaviors that should exhibited by the robot instead of by the stages of information flow within the controller. These behaviors are directly connected to sensors and actuators. A behavior can, thus, be thought of as a transfer function, which transform its sensory input to an actuator output as shown in Fig. 4.1. Information coming from the sensors need not go through a decision making process; an action is taken in contrary to the hierarchical model. This of course increases the time response of the robot.

4.1 Subsumption Architecture

Subsumption Architecture is a special case of the behavior-based control for robotics. The term "Subsumption Architecture" has many several meanings, which can be very specified or quite general. In its most general sense the term subsumption architecture refers to the idea of behavior based decomposition of control. More specifically, it usually means a layered system with higher layer over riding or subsuming lower levels. Finally, it can be the specific language for building controller described by [5].

A behavior-based controller is divided into modules, each of which is responsible for one type of behavior or task to be performed by the entire system. The behavior based controller has its module connected by "parallel wiring" This reduce the vulnerability to bottlenecks. Each modules uses only the sensor information and assumption about the world that its needs for its specific task Fig. 4.2.

4.2 Specific Architecture

The subsumption architecture includes a mechanism for implementing behavior-based control with subsumtion the lower level behaviors by high level ones. The architecture consists of a language for building behavior out of liked finite state machines, and a mechanism for implementing subsumption by allowing behaviors to suppress or inhibit the links used by lower level behaviors.

A network of finite state machine implements a behavior, in the subsumption architecture. The lowest level modules are argument FSAs, which typically contain ten states or less. All modules operate independently at all times;

they communicate by asynchronous links, which connected the outputs of some modules to the inputs of others. Any module can send a message through any of its output links and can read the last message received on any of its input links at any time. All modules can read any of the robot's sensors, though in general only the lowest level behavior will manipulate the control directly.

4.3 The Subsumption Compiler

A compiler has been written which takes a description of a subsumption controller and produces a "C" programs that implements it. "C" was used instead of Lisp because this was the language used in the simulator and because fast run time performance was needed. The description file contains definitions of all the modules and their connections. It may also contain "C" code.[9].

5. Oct-Ib Algorithms

The algorithm employed by each behavior is the following. Each behavior starts from a "minimal" perception list. More specifically, only conditions that are necessary in order to be able to execute the processes of the behavior are presented.

5.1 Obstacle Avoidance Algorithm

In a first approach, a few assumptions are made regarding the obstacles. First, static obstacles are considered. Second, the robot can detect obstacles with its front and back sensors and front, side and back whiskers. Finally, there is enough space around the obstacle for the robot to move in. Figures (5.1-a) and Figure (5.1-b) show obstacle avoidance algorithm by activating IR sensors and whiskers.

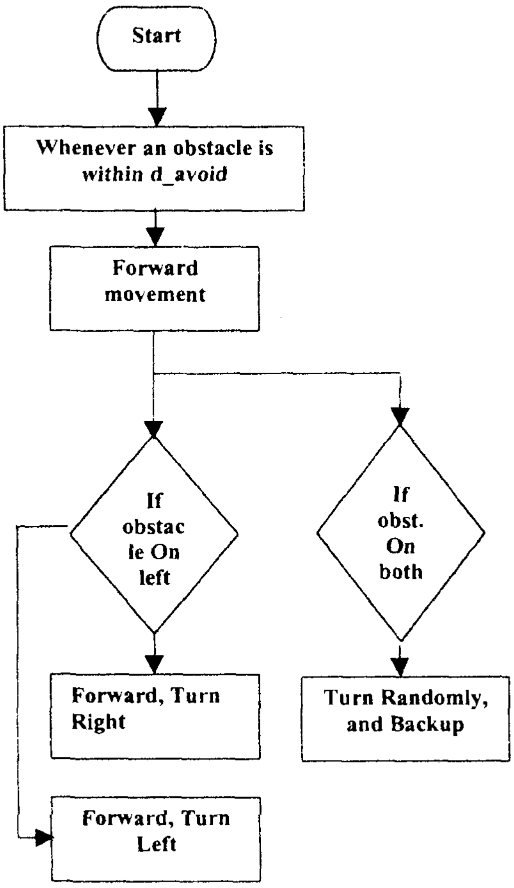

FIGURE (5.1-A) IR SENSORS ACTIVATED ALGORITHM

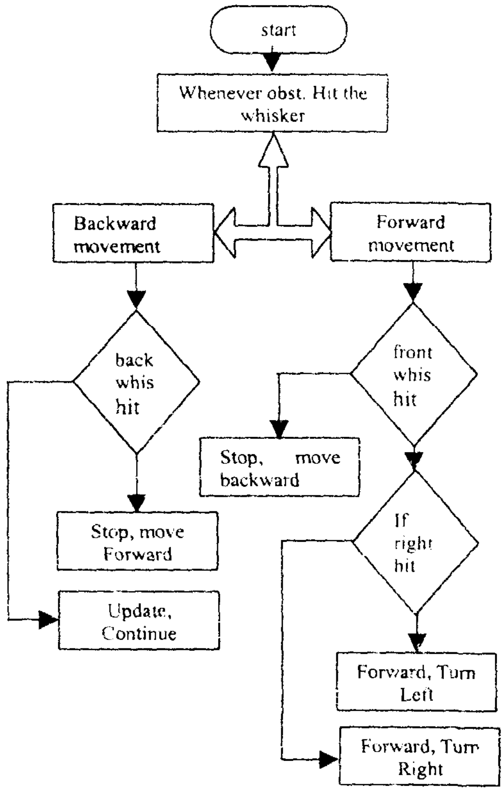

FIGURE (5.1-B) WHISKER ACTIVATED ALGORITHM

6. Experiments Setup

Our experiments were performed on Oct-Ib eight-legged walking robot. The goal of the proposed algorithm is to learn in the dynamic environments. It would have been inappropriate to start directly with the experiments where non-stationary conditions make analysis and validation of the properties of the algorithm extremely difficult. To gradually address this problem, we first used static environment conditions with increasing the complexity to verify that the algorithm could cope with stationary conditions from its interaction with the environment.

6.1 Experiment No.1: Robot Manoeuvers around a block

In this experiment the Oct-Ib moves down the center of the corridor guided by five forward facing infrared (IR) sensors. An obstacle lies halfway down the corridor blocking one side of the passage. When Oct-Ib reaches the block it manoeuvres around it and then continues on. No contacts are made between the robot and the obstacles at any time during the test run. Oct-Ib successful behavior set is shown in the Figure (6.1).

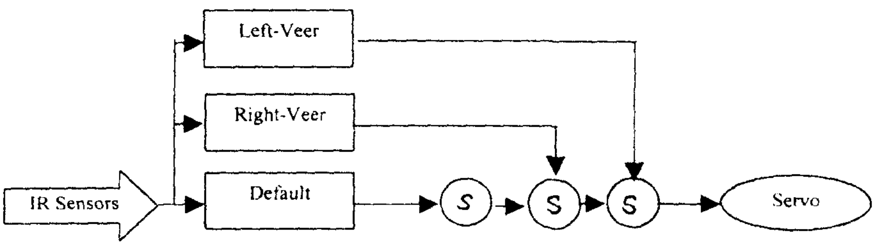

FIGURE (6.1) BEHAVIOR STRUCTURE (EXPERIMENT 1)

Also another experiments are applied for detecting inputs from primary and secondary whiskers, This group of behaviors generate evasive actions. Depending on the port of the whisker support mechanism, which is activated, the robot moves backward, forward, or turn left or right. Higher priority is given to forward and backward movement, and turning will be executed in the absence of those motion demands. If the secondary whiskers are touched either from left or right, the Oct-Ib robot will turn gradually in the direction away from the touch by tilting the body. Oct-Ib successful behavior set is shown in Figure (6.2).

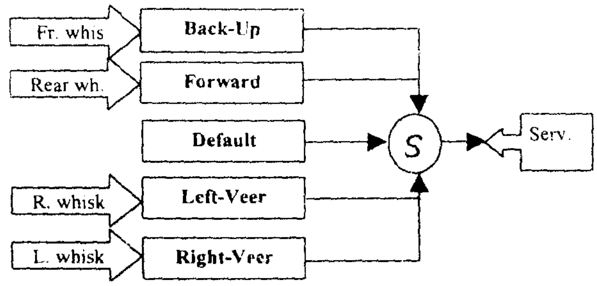

FIGURE (6.2) BEHAVIOR STRUCTURE (EXPERIMENT 1)

7 Conclusion

This paper presented a method for analyzing behavior-based approach for navigation in an unknown environment. We have developed an algorithm, which allows a behavior-based robot to learn when its behaviors should become active. We tested our algorithm by successfully teaching the Oct-Ib legged walking robot to walk forward and backward. Excellent collision avoidance performance was obtained. For further work we plan to test the generality of the algorithm by doing more experiments with the robot for different tasks and by studying the properties and limitation of the algorithm with a mathematical model.

Reference

[1] J. Borenstien and Y. Koren. IEEE Transaction on Systems, Man., and Cybernetics, vol. 19, No. 5, Sept./Oct. 1989, pp. 1179-1187.
[2] Roland C. Arkin and Robin R. Murphy. Autonomous Navigation in a Manufacturing Environment. IEEE Trans. On Robotic and Automation, vol. 6, No. 4, pp. 445-454, 1990.
[3] Wei Li. Fuzzy Logic Based "Perception-Action" Behavior Control of a Mobile Robot in Uncertain Environments. IEEE World Congress On Computational Intelligence, 1994.
[4] Nils J. Nillson. Problem Solving Method in Artificial Intelligence, New York: McGraw-Hill 1971.
[5] R. A. Brooks. A Robust Layered Control Systems for a Mobile Robot. IEEE Journal of Robotics and Automation, RA-2, 1986.
[6] Takashi Gomi and Jean-Christophe Laurence. Behavior Based AI Techniques for Vehicle Control. IEEE-IEE Vehicle Navigation & Information Systems Conference-VNIS'93.
[7] Maja J. Mataric. Behavior-Based Robotics as a Tool of Artificial Behavior and Analysis of Natural Behavior. In Cognitive Science, vol. 2, No. 3, 1998, 82-87.
[8] Genjirou Nishi, Shegeru Shinbori, Takashi Gomi, Koichi Ide, Patric Maheral. Mechanimal for Mechquarium. AAI Technical Report 96-03, 1996.
[9] Ralph Hartely and Frank Pipitone. Experiment with the Subsumption Architecture. Proceeding of the 1991 IEEE. International Conference on Robotics and Automation. California-April 1991.

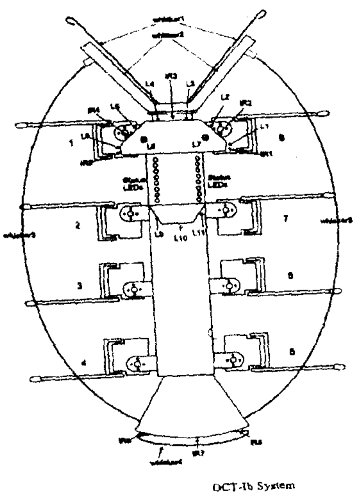

FIGURE (3.1) OCT-IB SCHEMATIC DIAGRAM