CONF-9605145--3

ADDING MEMORY PROCESSING BEHAVIORS TO THE FUZZY BEHAVIORIST-BASED NAVIGATION OF MOBILE ROBOTS

François G. Pin and Steven R. Bender
Robotics and Process Systems Division
Oak Ridge National Laboratory*
P.O. Box 2008
Oak Ridge, TN 37831-6305

Contact: Dr. François G. Pin, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6305, Telephone: (423)574-6130, Fax: (423)574-4624, E-mail: pin@ornl.gov

Submitted to: ISRAM '96, Sixth International Symposium on Robotics and Manufacturing, Montpellier, France, May 27–30, 1996

*Oak Ridge National Laboratory, managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under contract number DE-AC05-96OR22464.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

ADDING MEMORY PROCESSING BEHAVIORS TO THE FUZZY BEHAVIORIST-BASED NAVIGATION OF MOBILE ROBOTS

François G. Pin and Steven R. Bender
Robotics and Process Systems Division
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6305

ABSTRACT

Most fuzzy logic-based reasoning schemes developed for robot control are fully reactive, i.e., the reasoning modules consist of fuzzy rule bases that represent direct mappings from the stimuli provided by the perception systems to the responses implemented by the motion controllers. Due to their totally reactive nature, such reasoning systems can encounter problems such as infinite loops and limit cycles. In this paper, we proposed an approach to remedy these problems by adding a memory and memory-related behaviors to basic reactive systems. Three major types of memory behaviors are addressed: memory creation, memory management, and memory utilization. These are first presented, and examples of their implementation for the recognition of limit cycles during the navigation of an autonomous robot in *a priori* unknown environments are then discussed.

INTRODUCTION

In previous papers, an approach, which we named the Fuzzy Behaviorist Approach (FBA), has been proposed for the development of navigation systems for approximate sensor-equipped robots [1,2,3]. This approach uses the Fuzzy Sets theoretic framework to embody approximate reasoning schemes that imitate human-type navigation behaviors. Under this approach, a robot's decision-making module consists of a fuzzy rule base built through the superposition of elemental fuzzy behaviors that represent direct mappings from the perception systems to the motion controllers. This essentially produces a purely reactive system, which of course has no possibility for long-term reasoning or global optimization. Similar approaches have been proposed by others (e.g., see [4], [5], [6], [7]) which, although benefiting from the "qualitative" aspect of the reasoning, also resulted in purely reactive systems. Due to their totally reactive nature, such qualitative reasoning schemes can encounter problems such as infinite loops and limit cycles. Adding a memory to the existing reactive system is one method for remedying such undesirable phenomena. In this paper, we propose an approach for the addition of memory-related behaviors to a fuzzy behaviorist-based robot control system. Three major types of memory behaviors are addressed: memory creation, memory management, and memory utilization. These new behaviors are developed in such a way that they conform to the formalism of the Fuzzy Behaviorist Approach and preserve the parallelism of the existing control system's architecture. The next section presents this approach. As an example and to illustrate the approach, a memory-related rule base is discussed which deals with the recognition of limit cycles during the navigation of an autonomous robot in *a priori* unknown environments. In this example, the robot identifies a limit cycle when it recognizes that its current position and motion direction approximately correspond to some previously encountered. Other schemes could of course be developed using the proposed approach. In Section 3, experiments involving *a priori* unknown environments in which the robot would typically encounter limit cycles or infinite loops are presented to illustrate the resulting control system's ability to identify and prevent such unrecoverable situations. Our conclusions and possible directions for enhancing the creation, management, and utilization of memory in fuzzy behavior-based control systems are discussed in Section 4.

APPROACH

Figure 1 shows the architecture of our proposed approach and a corresponding typical flow of control. The lower portion of the diagram corresponds to the basic reactive system in which, at each new time step, the robot collects raw data from its sensors and the reasoning module transmits commands to the motion controllers or actuators. Memory is added to the basic reactive system, and memory feeding behaviors are responsible for preprocessing raw data and sending the results to the memory. This data is stored and maintained in memory in an orderly, useful manner by the memory management behaviors. When necessary, data is used by the memory utilization behaviors to perform various memory-dependent functions, for example, identify previously visited positions and orientations and recognize limit cycles or infinite loops so they may be avoided. When these functions are activated, a command is sent to modify the input or output of the basic reactive system in order for the particular memory-related condition to alter the effect of the basic reactive mode. (Note that the reactive behaviors themselves are not modified, but only their input or output data are affected.) As an example of this latter process in the case of limit cycles, virtual obstacles are created to modify the robot's raw sonar readings and to force the robot out of local minima areas. In the following subsections, a sample implementation of this approach for limit cycle detection and avoidance are presented to illustrate the functioning of the proposed architecture in a very simple case.

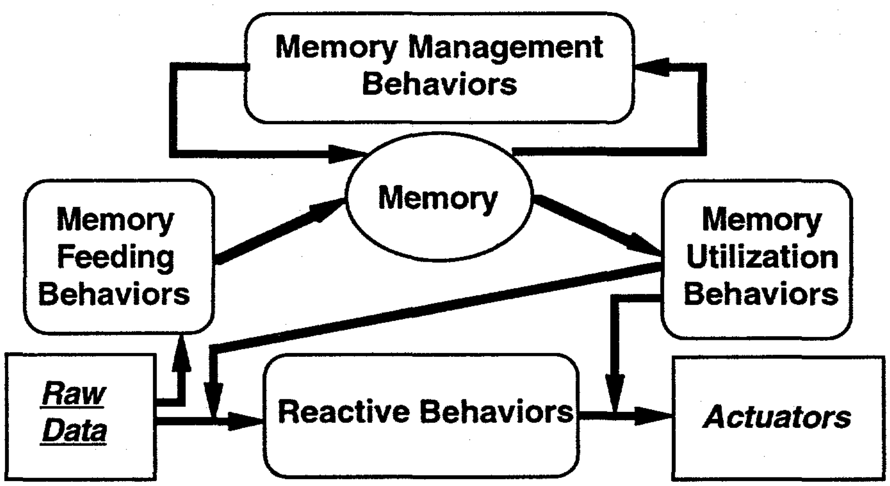

Figure 1. Flow of control in Fuzzy Behaviorist-based robot control system including new memory creation, memory management, and memory utilization behaviors.

Memory Forming Behaviors

The raw sensor data, e.g., sonar and wheel encoder readings, are acquired by the memory forming behaviors. These behaviors calculate the current x, y coordinates and orientation of the robot using the simple odometry calculations, and send these to the memory management behaviors for storage.

Memory Management Behaviors

To manage and ease the retrieval of previous robot positions in memory, each new x, y coordinate is "hashed" into memory according to the following formula

$$hashvalue = \tfrac{1}{2} X^2 + XY + \tfrac{1}{2} Y^2 + \tfrac{1}{2} X + \tfrac{3}{2} Y \quad (1)$$

where $X = x / <\text{size of grid}>$ and $Y = y / <\text{size of grid}>$

Position memory is split into four sections representing the four quadrants in a two-dimensional Cartesian plane. Each quadrant is broken into grids in accordance with the above formula (1) as shown in Figure 2. A linked list holds all positions within a particular grid. These linked lists are stored in an array indexed by the hash values. These indices represent the different grids sketched in Figure 2.

The main navigation goal, specified as an x, y position and orientation of the robot with respect to the starting point of the navigation experiment, is stored in the bottom of a stack. When the stack is empty, the robot has achieved its main goal and the navigation stops.

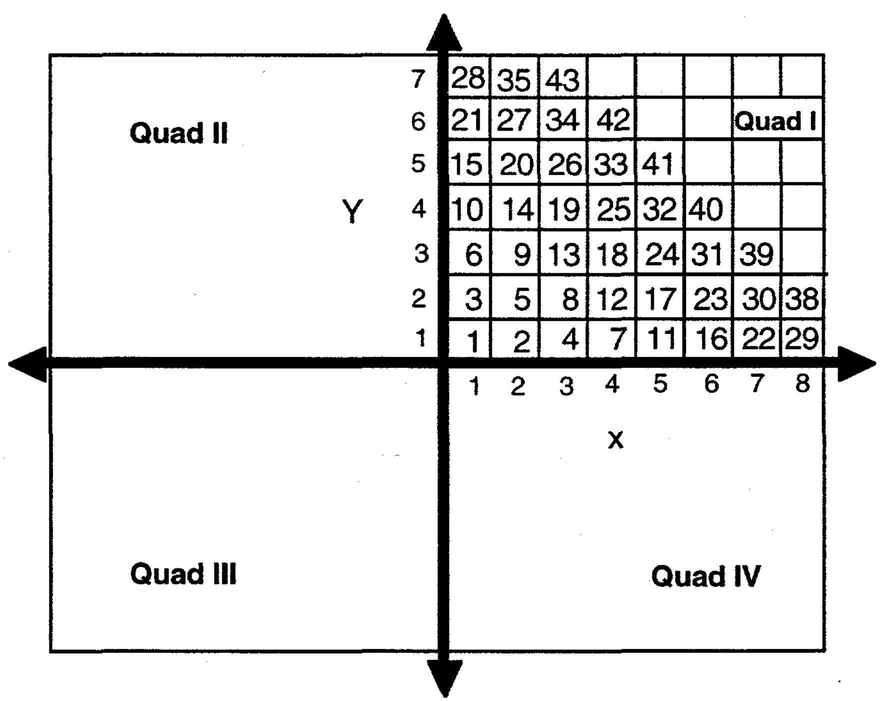

Figure 2. The robot's surrounding environment is represented as a two-dimensional Cartesian plane broken into grids in accordance with a hash formula.

Memory Utilization Behaviors

Memory Utilization Behaviors are developed to determine whether or not the robot has visited a particular position before. In this illustrative example, these memory utilization behaviors form a rule base which contains only two fuzzy rules. These rules compare the current position and orientation (motion direction) of the robot with those stored in memory. One rule establishes recognition of the x, y position of the robot while the other determines whether or not the same motion direction exists. Inferencing determines whether the robot recognizes having been in this position before or not, with or without the same orientation. The interesting aspect of using fuzzy rules here, just like in the basic reactive system [1], [2], [3], is that the full strength of approximate reasoning using membership functions can be utilized. For example, the threshold of recognition of "having been here before" can be adjusted based on the precision (uncertainty) of the odometry sensors and the length of the already executed journey (during which position uncertainty increases).

In this illustrative example, virtual obstacles are created to prevent limit cycles and infinite loops. The robot detects such undesirable phenomena through consecutive recognitions of a previously visited positions. A virtual obstacle is formed by taking the minimum and maximum *x, y* coordinate values of all positions collected between two consecutive encounters of a previously visited position with the same orientation, i.e., while in a limit cycle. These values form the points representing the four corners of the virtual obstacle. Once the robot has recognized a limit cycle, it establishes a subgoal based on the position of the virtual obstacle just created, and stores it on the goal stack. The coordinates of the subgoal are calculated from the midpoint of the side of the virtual obstacle opposite that of the previous goal, one meter from the wall of the virtual obstacle, as is shown in Figure 3.

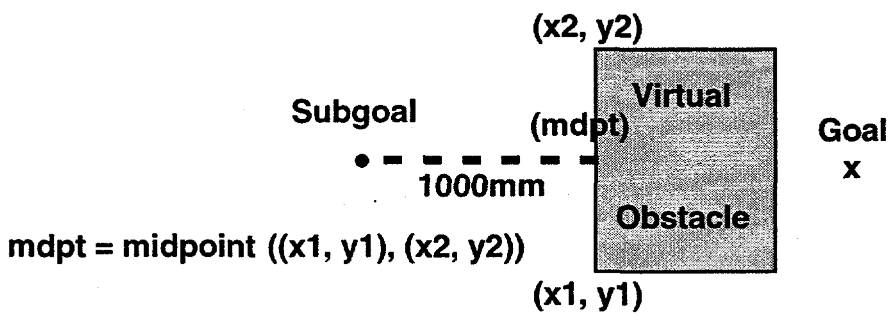

Figure 3. Computation and location of a subgoal created based on the position of the virtual obstacle just created.

EXPERIMENTAL RESULTS

Experiments were performed in which the new memory behaviors were added to an existing totally reactive navigation system. For the preliminary tests and for better assessing the resulting effect of the new behaviors, the simulator of the actual robot (e.g., see [1], [2], or [3]) was used. Tests of the memory creation and memory management behaviors showed the efficiency and accuracy of using a hash table to store the break-down of the robot's surrounding environment, although we are also investigating other schemes for storage and optimal retrieval of this and other information in memory. Memory utilization rule bases, each with a different fuzzification of the qualitative variables, along with different membership functions in the rules were also tested. These confirmed the intrinsic dependence of the scheme on the odometric sensor quality.

Figure 4 shows a navigation run without the use of the memory behaviors. The robot quickly finds itself oscillating back and forth in a limit cycle against the far wall of a local minimum area. The robot will remain in a limit cycle in this strong local minimum area due to its unconditionally reactive behavior.

Figure 5 shows another navigation run in the same environment conditions, however, with the memory-related scheme and behaviors added to the control system. After several instances of noting similar positions and orientations, the robot recognizes it is in a limit cycle. A virtual obstacle is created based on the minimum and maximum *x, y* coordinate values collected. A subgoal is created according to the position of the new virtual obstacle and the robot navigates towards that subgoal. As seen in Figure 5, the robot reaches the subgoal. Navigation then continues except that now the real sonar readings are overridden by the distances to the virtual obstacle(s) when necessary. A very similar process, including a limit cycle and creation of another virtual obstacle and another subgoal, occurs in which the control system repeats the above process, progressively forcing the robot out of the area of local minimum. The "dead end" chamber is eventually "filled" with virtual obstacles, preventing entrance of the robot, which can then successfully continue toward its original goal.

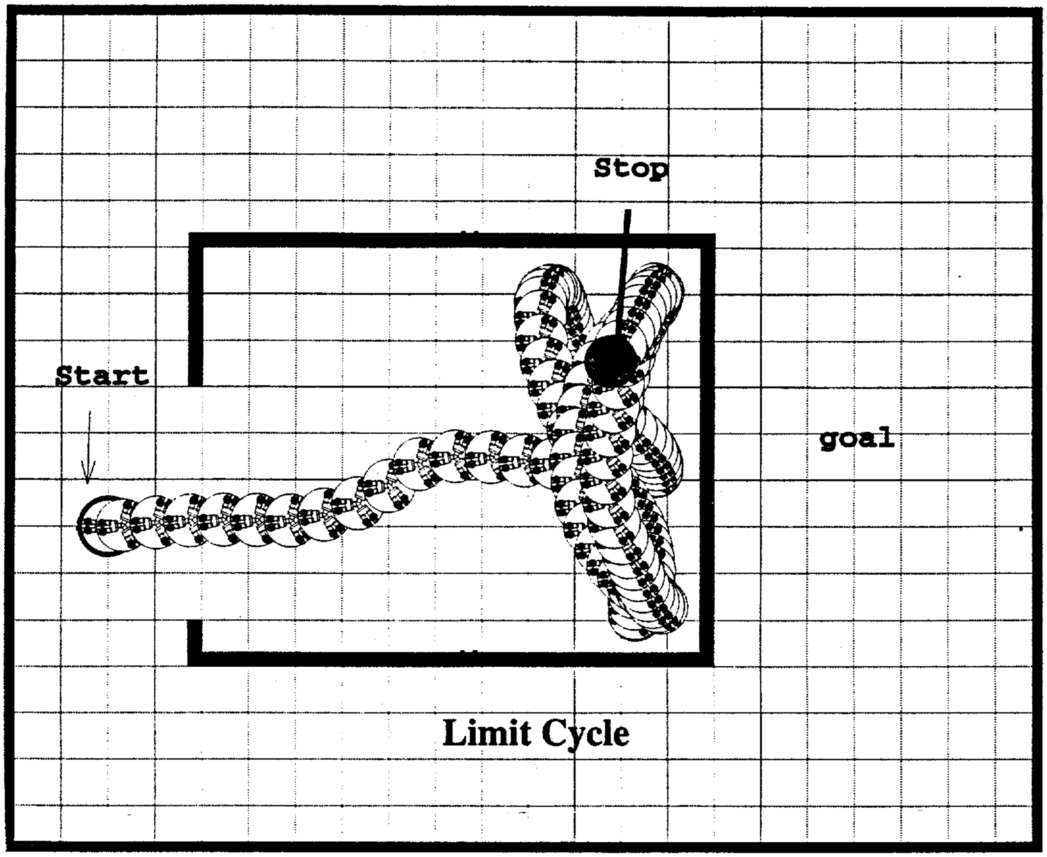

Figure 4. Sample run without the memory behaviors. The robot rapidly enters a limit cycle oscillating against the far wall of a local minimum nearest the goal.

CONCLUSION

A novel approach to remedy some of the shortcoming of purely reactive systems has been proposed. The corresponding architecture calls for the addition of memory to the system, however, along the principles of the Fuzzy Behaviorist Approach to utilize the strengths of both Approximate Reasoning and the Behaviorist Theory in uncertainty-prone decision-making conditions. Three forms of memory-related behaviors, memory creation, memory management, and memory utilization, have been discussed for addition to existing totally reactive Fuzzy Behaviorist-based robot control systems, preserving the existing system's architecture and its parallelism. The approach has been illustrated through a sample implementation for the avoidance of limit cycles in the sensor-based navigation of an autonomous robot in *a priori* unknown environments. Sample results involving limit cycles in strong local minima areas have been presented to illustrate the control system's ability to identify and prevent such unrecoverable situations. Our current work focuses on the development and test of the approach for other memory-based behaviors including local and global optimization and on refinement of the detection behaviors through an adaptive scheme.

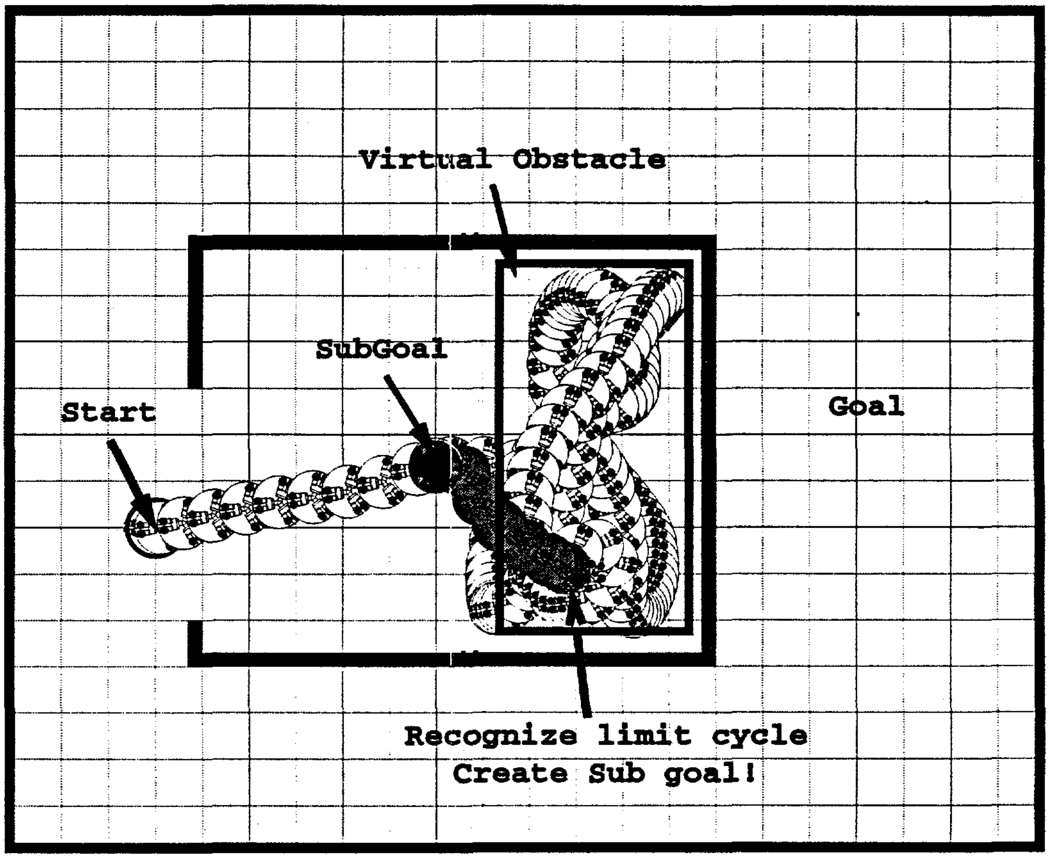

Figure 5. Sample run using the memory behaviors showing the recognition of a limit cycle and the creation of a virtual obstacle and a subgoal.

REFERENCES

1. Pin, F. G. and Y. Watanabe, "Navigation of Mobile Robots Using a Fuzzy Behaviorist Approach and Custom-Designed Fuzzy Inferencing Boards," *Robotica* **12**(6), 491–503 (1994).
2. Pin, F. G. and Y. Watanabe, "Steps Toward Sensor-Based Vehicle Navigation in Outdoor Environments Using a Fuzzy Behaviorist Approach," *International Journal of Intelligent and Fuzzy Systems* **1**(2), 95–107 (1993).
3. Pin, F. G. and Y. Watanabe, "Automatic Generation of Fuzzy Rules Using the Fuzzy Behaviorist Approach: The Case of Sensor-Based Robot Navigation," *Intelligent Automation and Soft Computing* **1**(2), 161–178 (1995).
4. Yen, J. and N. Pfluger, "A Fuzzy Logic Based Robot Navigation System," in *Proceedings of the AAAI Symposium on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots*, Cambridge, Massachusetts, October 23–25, 1992, pp. 195–199.
5. Takeuchi, T. et al., "Fuzzy Control of a Mobile Robot for Obstacle Avoidance," *Information Science* **45**, 1988, pp. 231–239.
6. Sugeno, M et al., "Fuzzy Algorithmic Control of Model Car by Oral Instructions," *Fuzzy Sets and Systems* **32**, 1989, pp. 207–219.
7. Reignier, P., "Fuzzy Logic Techniques for Mobile Robot Obstacle Avoidance," *Robotics and Autonomous Systems* **12**, 1994, pp. 143–153.