GENETIC PROGRAMMING OF FUZZY COORDINATION BEHAVIORS FOR MOBILE ROBOTS

Edward Tunstel* **and Tanya Lippincott**
NASA Center for Autonomous Control Engineering †
Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131 USA
tunstel@robotics.jpl.nasa.gov

ABSTRACT

Intelligent robot navigation can be achieved using a control system comprised of a collection of special-purpose motion routines, or behaviors. An approach to behavior coordination in multi-behavior systems is described with emphasis on evolution of fuzzy coordination rules using the genetic programming (GP) paradigm. Both conventional GP and steady-state GP are applied to evolve a fuzzy-behavior for sensor-based goal-seeking to be used in a hierarchical fuzzy navigation controller. The usefulness of GP is demonstrated by simulating performance of evolved coordination rules for autonomous navigation.

KEYWORDS: fuzzy-behavior, genetic programming, behavior modulation

INTRODUCTION

Real-time intelligent controllers are required to achieve the level of autonomy necessary in unstructured or "non-engineered" domains such as office settings, automated factories, and natural terrain. Traditional approaches based on sequential task decomposition have met with difficulty in achieving real-time response. Controllers are now being developed that are modeled after natural processes and that advocate a behavioral decomposition of tasks. These behavior-based systems facilitate real-time intelligence by decomposing high-level motion control into a set of special-purpose behaviors that achieve distinct tasks when subject to particular stimuli while operating concurrently. We have focused our research efforts on developing an architecture that incorporates fuzzy control theory into the framework of behavior control for mobile robots. A mobile robot behavior is encoded as a fuzzy rule-base that maps relevant sensor inputs to control outputs according to a desired control policy. Autonomy is achieved within a hierarchical structure of fuzzy-behaviors in which low-level navigation behaviors are realized as fuzzy logic controllers, while higher-level coordination behaviors are implemented as fuzzy decision systems. In order to exhibit *robust* autonomous performance, a strategy for coordinating the fuzzy-behaviors must be adopted. A suitable approach in the context of behavior hierarchies was proposed in [1].

Experience thus far has revealed that in many practical instances fuzzy control alone is insufficient for addressing complex intelligent control problems of robotics. Furthermore, humans often find it difficult to design knowledge-based control systems with interacting rule-bases, particularly in the absence of experts or sufficient knowledge of the problem. In

*On leave from NASA Jet Propulsion Laboratory, Pasadena, CA. USA
†This work was supported in part by NASA under contract # NCCW-0087.

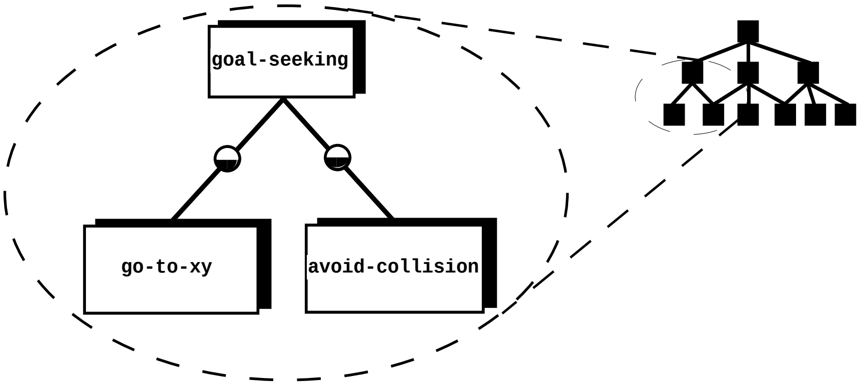

Figure 1. Goal seeking portion of a behavior hierarchy.

this paper, we address the problem of automatic discovery/learning of coordination rules for use in the proposed coordination strategy. This problem has been previously approached in the contexts of other coordination schemes by using reinforcement learning [2] and hybrids of reinforcement and neural networks [3, 4]. In [5], we demonstrated the potential of the genetic programming paradigm for learning fuzzy rule-bases for low-level regulation and tracking types of problems. Using this earlier work as a foundation, we apply our GP approach here to higher-level behavior coordination. Results are presented on coordination behaviors evolved for the goal-seeking portion of a mobile robot behavior hierarchy.

COORDINATION BY BEHAVIOR MODULATION

In our hierarchical approach a collection of primitive behaviors reside at the lowest level. These are simple, self-contained sets of rules that serve a single purpose, e.g. collision avoidance. They perform nonlinear mappings from different subsets of the sensor suite to subsets of control actions. Acting alone, each would be insufficient for achieving complex navigation tasks. However, their capabilities can be modulated through synergistic coordination to produce composite behaviors suitable for goal-directed operations. Behaviors at different levels of the hierarchy are generally interconnected. For example, consider the simple two-level structure illustrated in Figure 1 consisting of primitive behaviors, `go-to-xy` and `avoid-collision`, and composite behavior, `goal-seek`. This is only the particular portion of a larger behavior hierarchy that is responsible for goal seeking. The remainder of the hierarchy may consist of additional fuzzy-behaviors, the number and purposes of which are indicative of the problem complexity. The interaction between `go-to-xy` and `avoid-collision` produces the desired task-oriented function of `goal-seek`. The circles on the adjoining lines in the figure represent weights of the associated primitive rule-bases that fluctuate according to their applicability. In general, these weights can be used for threshold activation. The fluctuations in these weights are at the root of the intelligent coordination of the primitive behaviors. Coordination is achieved by weighted decision-making embodied in a concept called the *degree of applicability* (DOA), which is a linguistic measure of the instantaneous level of activation of a behavior. Fuzzy rules for coordination are formulated to include weighting consequents which modulate the DOAs of lower-level primitives. The DOA, $\alpha_j \in [0,1]$, of primitive behavior j is specified in the consequent of rules of the form

$$IF\ x\ is\ \tilde{A}_i\ THEN\ \alpha_j\ is\ \tilde{D}_i \tag{1}$$

where x is an input linguistic variable and $\tilde{A}_i$ is a fuzzy subset representing its linguistic value. $\tilde{D}_i$ is a fuzzy set specifying the linguistic value (e.g. "*HIGH*") of α_j for the situation prevailing during the current control cycle. The DOAs cause the control policy to dynamically change in response to goals, sensory input, and internal state. Fuzzy rules of each applicable primitive behavior are processed yielding respective output fuzzy sets. Each fuzzy output is weighted by its corresponding DOA, the weighted sets are aggregated by fuzzy union and then defuzzified to yield a crisp output that is representative of the intended coordination.

GENETIC PROGRAMMING OF FUZZY CONTROLLERS

The GP paradigm [6] computationally simulates the Darwinian evolution process by applying fitness-based selection and genetic operators to a population of parse trees of a given programming language. It departs from the conventional genetic algorithm primarily with regard to its representation scheme. Structures undergoing adaptation are executable hierarchical programs of dynamically varying size and structure, rather than numerical strings. In our GP system a population comprised of fuzzy rule-bases, that are candidate solutions to the problem, evolves in response to selective pressure induced by their relative fitnesses for implementing the desired behavior.

Rule Discovery

In the process of learning fuzzy rules, GP manipulates the linguistic variables directly associated with the fuzzy-behaviors. The dynamic variability of the representation allows for rule-bases of various sizes, thus enhancing population diversity which is important for the success of the GP system. It also increases the potential for discovering rule-bases of smaller sizes than necessary for completeness but sufficient for realizing desired behaviors. The search space is contained in the set of all possible rule-bases that can be composed recursively from a set of *functions* and a set of *terminals*. The function set consists of components of the generic fuzzy *if-then* rule and common fuzzy logic connectives, i.e. functions for antecedents, consequents, fuzzy intersection, rule inference, and fuzzy union. Each rule-base is an executable program that evaluates to an output fuzzy set resulting from fuzzy inference. The terminal set is made up of the input and output linguistic variables and pre-specified membership functions associated with the desired behavior.

Given suitable function and terminal sets GP proceeds by randomly generating an initial population of behaviors. This is followed by evaluating each behavior in the current population and applying, primarily, reproduction and crossover to behaviors selected with a probability based on fitness. Crossover is done by swapping portions of parse trees from two parent behaviors such that proper syntax is preserved. Upon satisfaction of termination criteria, the GP result is the best-fit behavior that appeared in any generation. In addition to this generational process, a steady-state evolution can be applied as in the Steady State Genetic Algorithm (SSGA) [7] which has recently been applied in GP for behavior evolution [8, 9]. In the SSGP approach the concept of generations does not exist. Instead, on each iteration following creation of the initial population only two new offspring are produced; the offspring replace two of the worst individuals in the population. In our approach, parent behaviors are selected to produce offspring by tournament, while behaviors to be removed are chosen randomly from the set of below-average behaviors in the current population.

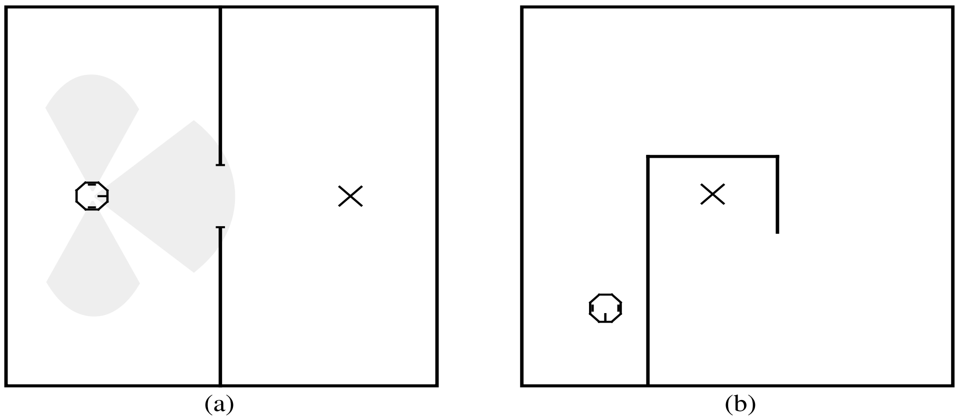

Figure 2. Example fitness cases.

EVOLUTION OF COORDINATION

During any sufficiently complex navigation task, individual behaviors frequently compete for control of the robot by recommending different control actions for common actuators. The problem becomes one of coordinating multiple task-achieving behaviors and resolving conflicts amongst them. In order to formulate suitable coordination rules, one must first decide what the DOAs of low-level behaviors should be in all practical situations perceived from sensory input. Formulation of such rules for coordination is not entirely intuitive, and expert knowledge about how to concurrently coordinate primitive behaviors is not readily available. We apply GP for this purpose. A fuzzy coordination behavior for goal seeking, comprised of rules in the form of Equation (1), is what we wish to evolve.

In the current implementation, applicability rules used by `goal-seek` to modulate the underlying primitive behaviors consider three instantaneous input states — the range to the nearest obstacle, the distance from the goal, and the angular heading to the goal. The consequents of these rules prescribe a DOA for each primitive behavior.

Behavior Fitness Evaluation

During evolution each behavior in the current population is evaluated via simulation in a number of indoor *fitness cases* subject to an upper time limit of 200 seconds. In this work, $n_f = 5$ fitness cases are used; the simplest and most difficult of these are illustrated in Figure 2(a) and Figure 2(b) respectively. Goal locations in the figure are indicated by an X, the robot is depicted as an octagonal icon with a radial line designating its initial heading, and the range sensor horizon is indicated by the shaded regions of Figure 2(a). In each case, the dimension of the indoor space is $10m \times 10m$. Each fitness case was chosen to represent situations likely to be encountered in indoor environments.

For a given behavior, the score of a trial run through fitness case i is given by

$$S_i = \begin{cases} 100 & ; \; goal\ reached \\ \frac{100}{\gamma(1+10e_N)} & ; \; otherwise \end{cases} \tag{2}$$

where e_N is the normalized residual distance to the goal in the case of a time-out or collision. The parameter $\gamma = 2$ if a collision occurs; otherwise $\gamma = 1$. That is, the score for an unsafe trial is half of that for a collision-free trial with all else being equal. The overall fitness of

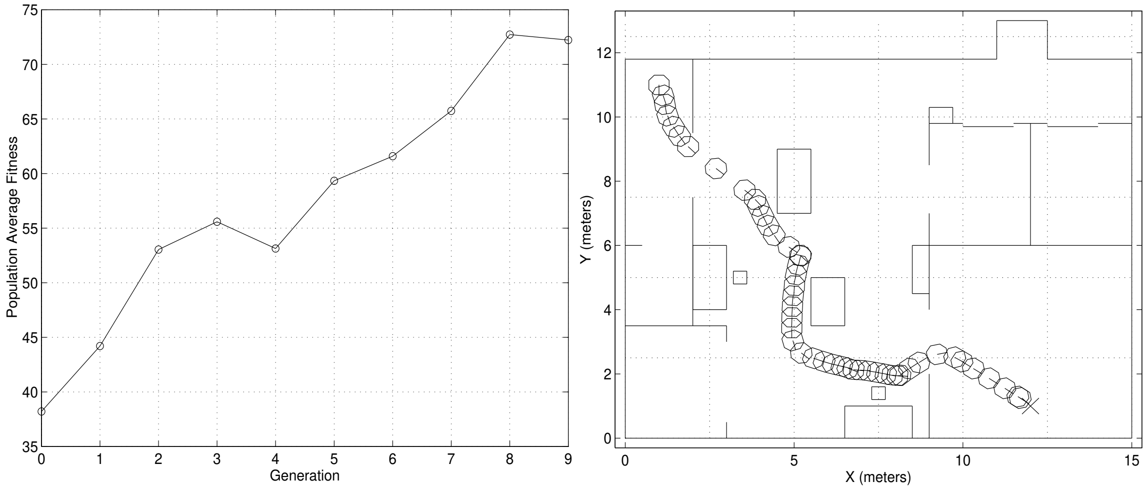

Figure 3. Performance of GP-evolved coordination.

the behavior is the average score over all n_f fitness cases:

$$F = \frac{1}{n_f} \sum_{i=1}^{n_f} S_i \tag{3}$$

Thus, the highest possible score, and hence fitness, is 100.

RESULTS

The GP system was run using population sizes of 10–20 rule-bases for a number of generations ranging from 10–15. In GP, genetic diversity remains high even for very small populations due to the tree structure of individuals [6]. Steady-state GP was also applied using a population size of 20. Results of runs using both approaches are presented here.

The simulated robot, modeled after a differentially steered $60cm$ base, senses range to obstacles using ultrasonic sensors. Pose information $(x\ y\ \theta)$ is computed from optical wheel encoders and its maximum speed was limited to $0.5m/s$. The simulated "world" is a hypothetical indoor environment of walls and furniture. It is considerably more general than any one of the fitness cases used during the evolution process and, thus, provides a suitable environment to test the generalization capability of the evolved behaviors. The mean performance of GP over five runs is shown, in the left half of Figure 3, as the progression of the population average fitness during the first ten generations. A trend towards higher fitness is evident. Evolved coordination behaviors were tested in the simulated environment (Figure 3) for sensor-based navigation from initial pose $(1\ 11\ -\frac{\pi}{2})$ to a goal located at $(12, 1)$. The right half of the figure shows a navigation run successfully coordinated by a GP-evolved behavior with a fitness of 71 and 12 rules. The best behavior discovered by GP over the five runs had a fitness of 86.5 and 11 rules.

Slightly better results were obtained using the non-generational SSGP. Figure 4 shows the progression of the average fitness of the current population at each iteration. Twenty behaviors were processed in the initial population; thereafter, two new behaviors evolved at each iteration. The figure also shows a navigation run coordinated by the best SSGP-evolved behavior which had a fitness of 87.8 and 9 rules. Only minor improvement in the path is observed, however, for this problem good regions of the search space were discovered with less processing by SSGP.

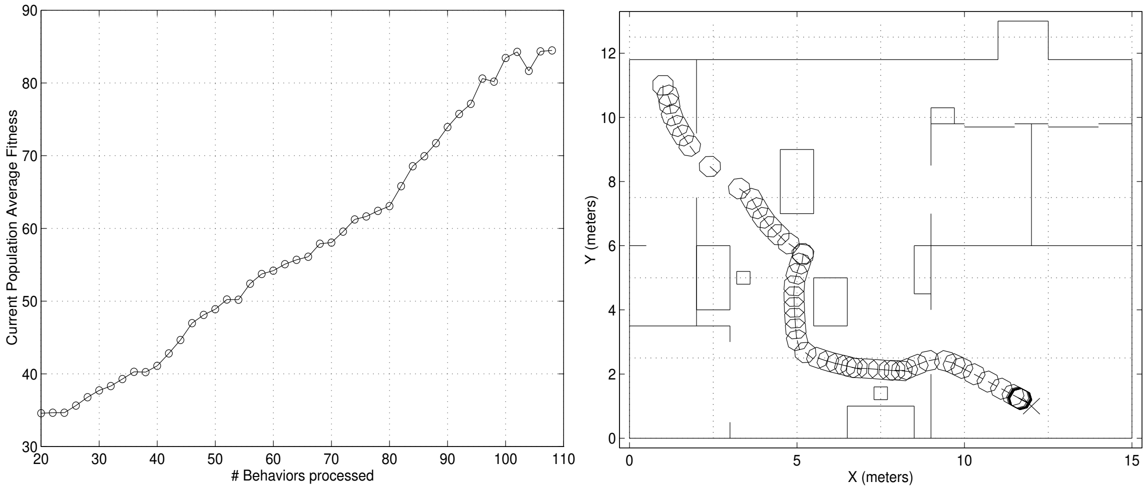

Figure 4. Performance of SSGP-evolved coordination.

SUMMARY AND CONCLUSION

An approach to applying genetic programming for learning fuzzy rules has been introduced. In particular, we have evolved rules for coordinating low-level fuzzy-behaviors within the context of a hierarchical architecture of mobile robot navigation behaviors. The coordination scheme is one of general behavior modulation which includes behavior selection as a special case. Here, we have focused on a goal-seeking portion of a behavior hierarchy; the GP approach is generally applicable throughout. Conventional GP and steady-state GP were applied, each yielding good results for small populations. Overall, SSGP yielded slightly better results for goal-seeking coordination. Using only five fixed fitness cases during behavior evolution modest generalization capabilities were exhibited by the highest fit behaviors. Additional improvement can be achieved with a richer set of fitness cases.

REFERENCES

1. Tunstel, E. "Coordination of Distributed Fuzzy Behaviors in Mobile Robot Control." *IEEE Intl. Conf. on Systems, Man and Cybernetics.* (October, 1995), 4009–4014.
2. Bonarini, A. "Learning to Coordinate Fuzzy Behaviors for Autonomous Agents." *2nd European Congress on Intelligent Techniques and Soft Computing EUFIT'94.* (1994), 475–479.
3. Beom, H.R., Koh, K.C. and Cho, H.S. "Behavioral Control in Mobile Robot Navigation Using Fuzzy Decision Making Approach." *IEEE Intl. Conf. on Robots and Systems.* (1994), 1938–1945.
4. Gachet, D., Salichs, M.A., Moreno, L. and Pimentel, J.R. "Learning Emergent Tasks for an Autonomous Mobile Robot." *IEEE Intl. Conf. on Robots and Systems.* (1994), 290–297.
5. Tunstel, E. and Jamshidi, M. "On Genetic Programming of Fuzzy Rule-Based Systems for Intelligent Control." *Intl. J. of Intelligent Automation and Soft Computing.* 2 (2) (1996), to appear.
6. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA (1992).
7. Syswerda, G. "A Study of Reproduction in Generational and Steady-State Genetic Algorithms." *Foundations of Genetic Algorithms.* G. Rawlins (Ed.), Morgan Kaufmann, San Mateo, CA (1991).
8. Reynolds, C.W. "Evolution of Corridor Following Behavior in a Noisy World." *3rd Intl. Conf. on Simulation of Adaptive Behavior.* (August, 1994), 402–410.
9. Nordin, P. and Banzhaf, W. "Genetic Programming Controlling a Miniature Robot in Real Time." Technical Report 4/95, CS Dept. University of Dortmund, Germany (1995).