Using Reflexes to Speed ANN Learning in an Autonomous Mobile Robot

ALEXANDER FERWORN KOSTAS PLATANIOTIS
Network-Centric Applied Research Team
School of Computer Science
Ryerson Polytechnic University
350 Victoria St
Toronto, Ontario, Canada, M5B2K3
aferworn@scs.ryerson.ca kplatani@scs.ryerson.ca

Abstract: - We introduce a novel control architecture for Autonomous Mobile Robots called the Reflexive Instructor (RI) with Deliberate Apprentice (DA). The architecture employs simple reinforcement signals provided by the RI component to train the DA. The DA is responsible for providing control signals to the agent's actuators based on received sensor input. The RI provides a measure of safety in this respect as it is responsible for taking over control of the mobile robot if the DA makes a mistake as well as providing an appropriate feedback signal to the DA.

The RIDA interaction is advantageous because it protects the vehicle from its own incompetence and has the potential to accelerate learning in the DA. We illustrate this by simulating a vehicle employing a simple RI coupled to a rapid reinforcement artificial neural network as a DA. The DA learns to use sensors while successfully interacting with its environment.

Key Words: - Neural Networks, Autonomous robotics, Control Architecture

1 Introduction

Artificial Neural Network (ANN) architectures have proven to be effective tools for developing systems capable of learning control tasks [1]. Their application to robotics is a natural extension of these successes. ANNs have been applied to problems as diverse as reverse kinematics [2], trajectory acquisition[3] and task planning[4][5]. ANNs have enjoyed similar success in Autonomous Mobile Robotics (AMR).

One of the recurring themes in AMR research is the inherent unpredictability of the environments that AMRs are forced to operate in. To a certain extent unpredictability can be addressed through learning. For example, an AMR might learn how to interpret sensor data coming from a sensed object while exploring an unfamiliar environment and respond with appropriate control signals to its actuators. Of course, initial responses will tend to be inappropriate with improved performance demonstrated over time.

One of the perennial difficulties in applying ANNs to AMR systems have been their abysmally slow learning rates. While in other applications this can be annoying, it is devastating to many AMR learning tasks. This is not difficult to understand. If an ANN requires one hundred repetitions to learn that the sensor data presented to it indicates that it has fallen off a cliff, it is unlikely that the other 99 repetitions will occur. The inability to improve performance in a timely manner has lead to disastrous failure. The semi-autonomous research robot Dante II [6] suffered a catastrophic system failure when it was unable to adapt to a surface consisting of mud rather than frozen snow while attempting to walk over it.

This problem has been addressed in several ways. Most notably is the avoidance of learning altogether--where learning a task is either accomplished off-line or is instilled using a carefully constructed ANN [7][8].

A different approach has been the application of computationally expensive ANN models in a suitable environment coupled with a certain degree of instilled behavior. This approach is perhaps best exemplified in the driving task discussed in [9].

Various investigators have suggested so-called reactive approaches to AMRs in which sensed aspects of the AMR's environment are acted on immediately through carefully engineered reflexes. Most notably, Brooks [10] has argued that “speed” concerns can be avoided by employing carefully constructing reactive systems based on Finite State Machines (FSMs). By carefully connecting these FSMs the AMR can anticipate all aspects of the environment it is likely to encounter. His subsumption approach has proven quite successful for a wide range of low-level tasks such as wall following and collision avoidance [11], however critics have correctly indicated that the AMR's designer must anticipate every eventuality and design the robot's reactive systems to address them. This is problematic, one cannot send an AMR to a distant planet and continue to tweak it into correct behavior.

We propose a model that can take advantage of the learning and control characteristics of certain ANNs and can make use of the reflexive response associated with reactive systems to compensate for relatively slow learning.

We call this control architecture the Reflexive Instructor with Deliberate Apprentice (RIDA) and discuss it in more detail in the following section. We go on to describe a RIDA implementation employing a reinforcement ANN as a learning component (DA) coupled with a RI employing reflexes. We apply this design to a simple collision avoidance problem. Finally we discuss some of the results of implementing the design.

2 The RIDA Architecture

RIDA consists of two independent, yet related sub-systems,

1. Reflexive Instructors (RI), and
2. Deliberate Apprentice (DA).

The sub-systems interact in a way best thought of in a pedagogical sense. The DA attempts to send control signals to the actuators it is intended to control. The RI, in turn, monitors the control signals and either does nothing if a control signal is appropriate or intervenes by over-riding the DA’s signal and injecting its own signal which it deems to be more appropriate--much as a teacher might correct the spelling of an errant pupil. At the same time, the activated RI informs the DA that it did something wrong.

This approach differs from classical reinforcement learning [12] by actually correcting a mistake made by the system that is learning. However, the learning system is not corrected--the AMR is. For example, if an AMR were about to launch itself off a cliff, it would be stopped by the RI component and would also inform the DA about its error.

The RI is responsible for performing a correct action (which may not be optimal) based on its reaction to the situation. This reaction is similar to the way our reflexes might protect us from severe burns. We are forced to withdraw a hand from an open flame and we are given an opportunity to learn about flame through pain.

The figure below illustrates the relationship between the RI and DA components. Note that the sensor signals going to the RI and DA need not come from the same sensor.

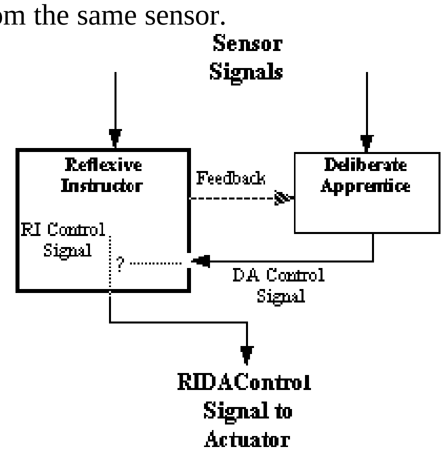

Figure 1 The RIDA Architecture

The RIDA architecture presents a hierarchical control scheme that places the RI in ultimate control of the actuator. The RI defers to the control signals of the DA so long as the DA’s control signals do not result in the RI's activation. The DA is free to “discover” control signals that result in the DA receiving optimum feedback from the RI.

The RI would normally consist of a highly reliable safety controller which would avoid the cliff problem alluded to previously. This frees the designer to select any higher level DA that is appropriate for the task.

3 The Problem

As an illustration of RIDA we selected the classic problem of designing an AMR that is capable of avoiding collisions with walls while remaining in motion. A collision is defined as any physical contact with a wall. Our goal was that the DA component of the AMR would learn how to use its sensor inputs to better control the vehicle than the RI could. To accomplish this we had to select a RI component that would promote the learning of the DA component. While the task is relatively simple, we use it here to illustrate the supervision potential of the RI.

4 The AMR and RIDA Designs

The model AMR vehicle we employed is illustrated below.

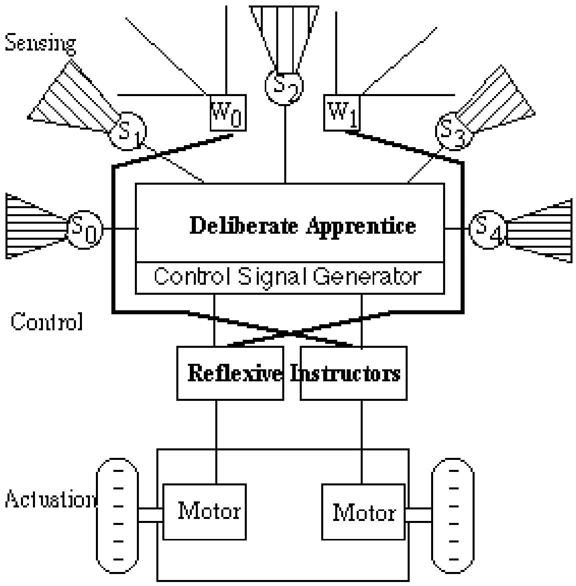

Figure 2 The Model AMR

Sensors on the vehicle consist of both sonar and whisker contact sensors. The left (W_0) and right (W_1) whisker sensors activate on contact and are cross-connected to the right and left RIs. The RIs were selected on the basis of reliability. This is appropriate for controllers that are the "final line of defence" against failure of the entire system. They consist of a simple reversal circuit as describe in [13], attached to the drive motors of the differentially steered AMR. When a whisker is depressed the current to the attached motor is reversed for a short period. The cross-connection is suggested in [14]. As the AMR contacts a wall the drive motor on the opposite side of the sensor will temporally reverse thus pulling the AMR away from the wall.

The sonar sensors (S_0 through S_4) communicate with the DA. The sonar was calibrated to send discrete signals indicating close, middle and distant objects. Since the number of potential actions of an AMR is limited by its actuators' ability to implement them, it is common practice to attempt to map what the sensors perceive to actions the actuators can actually perform. In our case the five sonar sensors with three discrete distance measurements were mapped to five drive/steering actuators capable of moving the vehicle left, left forward, forward, right forward or right. The vehicle was not permitted to stop as this was the trivial "safe state" and would not accomplish the goal of remaining in motion. Reversing was not allowed as no sensors were provided to the rear of the vehicle. This was done to avoid backing up in a blind state.

[15] have suggested a Rapid Reinforcement (RR) ANN for performing this mapping employing a modified feed-forward, winner-take-all neural network to perform the selection of the next action and using a punishment/reward signal to act as a reinforcement generator.

The ANN architecture is not particularly complex, employing only 15 input and 5 output units, however it does rely on a relatively simple feedback mechanism that is appropriately provided by the RI. The network architecture is shown in figure 3. For a detailed description of the learning algorithm see [16]. Other potential DA candidates were [17] and [18].

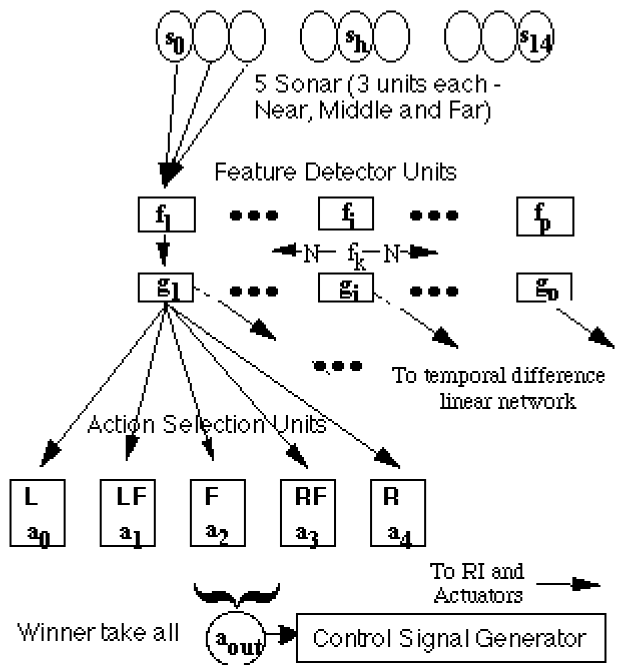

Figure 3 RR Network Architecture

Essentially, the RR attempts to send the correct output to the control signal generator whose signals are interpreted by the RIs. When the DA achieves correct output, the RI does nothing. When the output produces a control signal resulting in a collision, the RI intervenes--producing a corrected control signal and sends feedback to the DA.

In our starting configuration, the DA does not have knowledge of how to control the vehicle correctly and must rely on the RI quite heavily.

5 The AMR and RIDA Simulation

The AMR was placed in a simulated environment as illustrated in figure 4. The vehicle is identified by the number 5 which indicates its orientation relative to the "arena". 0 indicates straight up, 2 indicates right, 4 straight down and 6 left. In this case the vehicle is facing to the middle of the left side of the arena. The vertical and horizontal bars represent walls.

```
+-0--1--2--3--4--5--6--7--8--9+
9|.  .  .  .  5  .  .  .  .  .|9
8|.  .  .  .  .  .  .  .  .  .|8
7|.  .  .  .  .  .  .  .  .  .|7
6|.  .  .  .  .  .  .  .  .  .|6
5|.  .  .  .  .||.  .  .  .  .|5
4|.  .  .  .  .||.  .  .  .  .|4
3|.  .  .  .  .||.  .  .  .  .|3
2|.  .  .  .  .||.  .  .  .  .|2
1|.  .  .  .  .||.  .  .  .  .|1
0|.  .  .  .  .||.  .  .  .  .|0
+-0--1--2--3--4--5--6--7--8--9+
```

Figure 4 The Arena

Two vehicles were simulated. The first was the AMR employing the RIDA architecture as discussed. The second did not use the RI portion and relied only on the RR network as implemented in [15] relying on simple reinforcement learning to correct its future behaviour.

The diagrams below illustrate the results of a typical trial run conducted employing the RIDA equipped AMR (figure 5) and an AMR equipped with only a RR network (figure 6). At the end of the trial the RIDA equipped vehicle had collided with walls 11 times out of 100 time steps allocated for the trial.

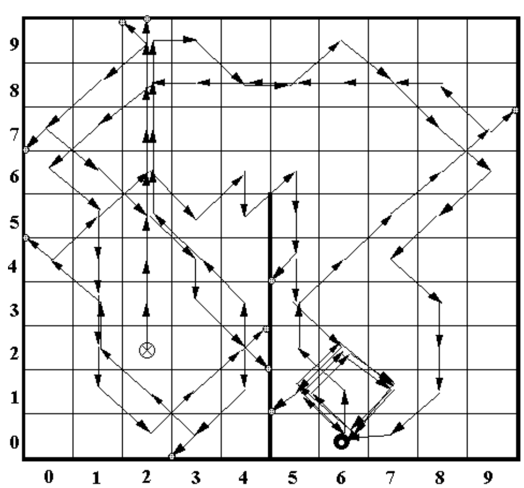

Figure 5 Path of the RIDA AMR

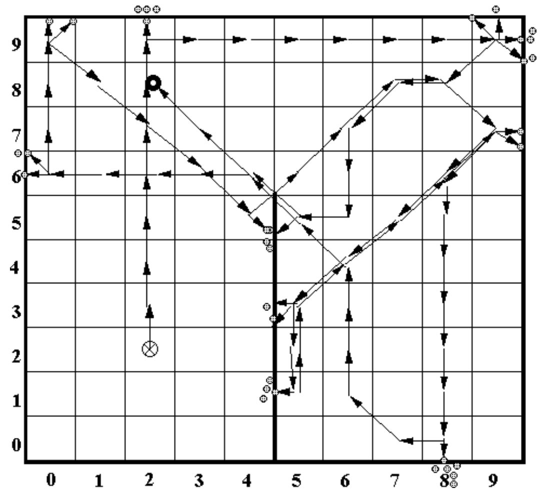

Figure 6 Path of the RR AMR

No further collisions occurred the last quarter of the trial. In contrast the RR equipped AMR collided 64 times with walls and continued to do so throughout the trial with no evidence of improvement over time.

The RI/DA vehicle incurred far fewer collisions and managed to traverse more of the actual arena in the same number of time steps than the non-RIDA AMR which was fettered by a series of collisions which lead to other collisions substantially hindering its progress.

Of interest is the pattern that the RI/DA vehicle exhibits. One can clearly see that the vehicle has actually discovered a form of wall following as it moves about the arena. This could be due to the RI policy that has a tendency to place the vehicle parallel to the wall at which a collision occurred.

The circular motion evident in the lower right quadrant of figure 5 occurred at the end of the trial and is a result commonly observed in reinforcement learning methods and was also observed in [13]. The AMR seems to have found a "safe" path that avoids collisions and keeps the AMR in Motion but is not the most effective means of traversing the arena.

Figure 7 further illustrates the superiority of the RIDA approach. After an initial “training set” of collisions (curve marked with squares), the RI/DA controller stopped employing the RI and became wholly reliant on the DA for guidance. In effect the AMR had learned to control the vehicle using the sonar sensors instead of relying on the more primitive whisker sensors. The RR network (curve marked by triangles) simply kept colliding with walls.

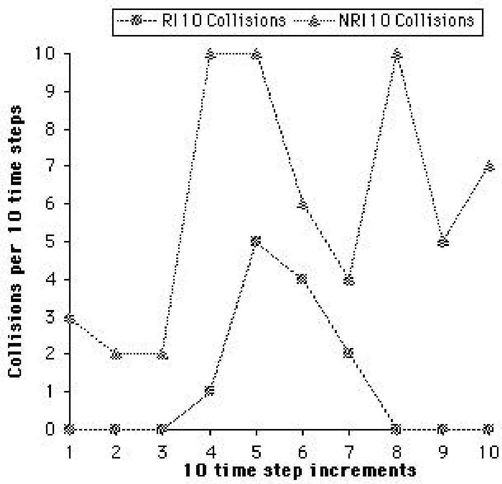

Figure 7 RIDA vs. RR Performance

6 Conclusion

We have demonstrated that by providing a safety mechanism in the form of the RI component to an AMR we are able to outperform a similar vehicle without this component. The RIDA interaction provides a richer feedback than is possible with more traditional reinforcement learning networks and was able to contribute to faster learning. In addition, there is the benefit an AMR gains by avoiding unnecessary--and possibly disastrous failure while learning a new task. This provides a certain cushion in which the DA component can continue to fail and learn and the AMR, as a whole, remains viable.

While the collision avoidance task is--in some sense--a "toy problem", it should be possible to employ the RIDA technique in selected circumstances where more traditional approaches are impossible. Potential mission environments might include planetary exploration or providing assistance in dangerous environments. Both settings do not lend themselves well to extensive modeling or prediction of their environments.

We have conducted other trials involving various ANN DAs engaged in various low-level tasks, and have achieved similar results. Learning is faster and the vehicle is protected from destruction. In addition we have replaced the learning DA with a reactive DA and have achieved a hierarchy of control very similar to that achieved by subsumption.

We wish to be very clear, under conditions where uncertainty exists about the nature of the environment, or an AMR's response to that environment, the RIDA architecture provides much needed support for learning and is especially well suited to enhancing the performance of reinforcement ANNs involved in Relevant AMR tasks.

References:

[1] Guez, A., Eilbert, J.L., and Kam, M., Neural Network Architecture for Control, *IEEE Control Systems Magazine*, pp. 22-25, April (1988)

[2] Iberall, T., A Ballpark Approach to Modeling Human Prehension, IEEE *Conf. on Neural Networks*, Vol. 4, pp. 535-544, (1987)

[3] Jorgenson, C.C., Neural Network Representation of Sensor Graphs in Autonomous Robot Path Planning, *IEEE Conf. on Neural Networks*, Vol. 4, pp. 507-516, (1987)

[4] Albus, J., Data Storage in the Cerebellar Model Articulation Controller (CMAC), *J. Dynamic Systems*, Measurement, and Control, pp. 228-233, (September 1975)

[5] Yueng, D. and Bekey, G., Adaptive Load Balancing Between Mobile Robots Through Learning in an Artificial Neural System, *IEEE Conf. on Decision and Control*, pp. 2299-2304, (1987)

[6] Gunther, J.A., Dante's Inferno, *Popular Science*, pp. 66-68, 96, (November 1994)

[7] Quinn, R.D., and Espenschield, K.S., Control of a Hexapod Robot Using a Biologically Inspired Neural Network, in *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, R.D. Beer (Ed), Academic Press, (1993)

[8] Beer, R.D., and Chiel, H.J., The Neural Basis of Behavioral Choice in an Artificial Insect, *Proc. of the 1st Intl. Conf. on Simulation and Adaptive Behavior*, (1991)

[9] Moravec , H.P., *Robot Rover Visual Navigation*, UMI Res. Press, (1981)

[10] Brooks, R.A., A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, (March 1986)

[11] Brooks, R.A., Elephants Don't Play Chess, *Robotics and Autonomous Systems* 6, pp 3-15, (1990)

[12] Kaelbling, L.P., Littman, M.L. & Moore, A.W., Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research* 4, 237-285, (1996)

[13] Jones, J.L. & Flynn, A.M., *Mobile Robots--Inspiration to Implementation*, A.K. Peters publishers, (1993)

[14] Braitenberg, V., *Vehicles, Experiments in Synthetic Psychology*, MIT press, (1986)

[15] Fagg, A.H., Lotspeich, D., Hoff, J. & Bekey, G.A., Rapid Reinforcement Learning for

Reactive Control Policy Design in Autonomous Robots, *submitted draft to WCNN*, (1994)

[16] Fagg, A.H., Lotspeich, D. & Bekey, G.A., "A reinforcement Learning Approach to Reactive Control Policy Design for Autonomous Robots", in *Proceedings of the IEEE Conf. on Robotics and Automation*, 39-44, (1994)

[17] Nehmzow, U., Hallam, J. & Smithers, T., "Really Useful Robots", *Intelligent Autonomous Systems* 1, T. Kanade, F.C.A Graen & L.O. Hertzberger (ed) , 284-293, (1989)

[18] Nehmzow, U., Smithers, T. & McGonigle, B., "Increasing the Behavioural Repertoire in a Mobile Robot", *From Animals to Animats, Proceedings of the 2nd Int. Conf. on Simulation of Adaptive Behavior*, MIT Press, 291-297, (1993)