# Towards an Adaptable Robot Language

**Holly Yanco**
MIT Artificial Intelligence Laboratory
545 Technology Square, Room 741
Cambridge, MA 02139
(617)253-7884
holly@ai.mit.edu

### Abstract

Robot communication is necessary to complete collaborative tasks efficiently. The robots will need an adaptable language if we want them to function in new and changing environments. This paper reports steps already taken towards the creation of an adaptable robot language and discusses extensions to the work that are being examined.

## 1 Introduction

Cooperation requires communication. If multiple agents are working towards a common goal, they must interact. Even if the goal could be reached without communicating, the performance will not be as expedient as a communication-based result.

For example, imagine a situation where two robots need to move a leaking container of hazardous materials and neither robot has the ability to move the container by itself. If each robot is told that the container needs to be moved, each robot will make an attempt to move the container; however, the robots will only be successful when both try to move it at the same time in the same direction. The long possible delay in reaching the goal could result not only in problems for humans, but also the destruction of the robots before the task is complete. In this hypothetical situation, communication between the robots could expedite the process.

Simply providing the robots with a means for communicating with other robots is not the solution. What happens when the robots encounter a new situation that the programmer had not envisioned? Only when the robots are able to adapt their language to novel situations will robots be able to be truly robust.

This paper describes two robots that can learn a simple two signal language and discusses possible extensions to the work. Reinforcement learning is being used to teach the robots. Much of the work done in reinforcement learning is in simulation only; moving this work to robots provides a richer testbed for the algorithms. (For an overview of research in reinforcement learning, see [Sutton, 1992].) The world is its own best simulation – running algorithms in a toy world can overlook problems in the real world, no matter how well-intentioned the designer of the simulation. Also, since memory and time constraints are much greater on robots than on computers, reinforcement learning algorithms may need to be redesigned for use on robots or the robot learning problem will need to be tailored to the size and time constraints.

## 2 Robots that learn a two signal language

Before the robots can adapt a language to meet changing needs, they must first have a language. Typically, robots are provided with language hard-coded into their software. However, the programmer has little idea what language will be most appropriate to the tasks or to the robots themselves. While a programmer can anticipate the communication needs of the tasks the robots will perform, the robots may see a more natural way to accomplish the goal. For instance, if one robot wants to tell another robot to move off to the left at a 45 degree angle, but the robots have only been provided with signals for "straight ahead" and "turn left", the instruction requires two signals and results in a less than optimal path. However, given the ability to learn a language, the robots could develop a signal in their language that means "go off to the left at a 45 degree angle".

Additionally, the process of learning the original language is similar to the process of relearning concepts or learning new ideas. So, once the robots can learn a language, they can also adapt their language.

### 2.1 The robots

Bert and Ernie, the two robots used in this research, are "Sensor Robots" designed by Fred Martin at the Media Laboratory at the Massachusetts Institute of Technology [Martin and Sargent, 1991]. Each robot is approximately $9''l \times 6''w \times 4''h$, with a single circuit board containing most of the computational and sensory resources of the robot. A 6v battery strapped to the underside of the chassis supplies the power for the robot. The robots are shown in figure 1.

The primary computational resource is an on-board Motorola 6811 microprocessor. The programming environment is IC, an multi-tasking interactive C compiler and interpreter developed by Randy Sargent [Sargent and Martin, 1991]. IC allows the sensor robot to be addressed through a serial line from a host computer as well as the downloading of programs for autonomous activity. The work described in this paper was implemented with the robots under autonomous control.

Locomotion is controlled by a dual geared-wheel drive stripped from a Radio Shack Red Fox Racer. The direction of the robot is controlled by varying the speed of both the left and right motors (with negative speed moving the motor backwards). The two motorized wheels are at the rear of the robot chassis and a caster is on the front.

Communication from human to Bert is through an infra-red remote control transmitter. Bert receives the infra-red

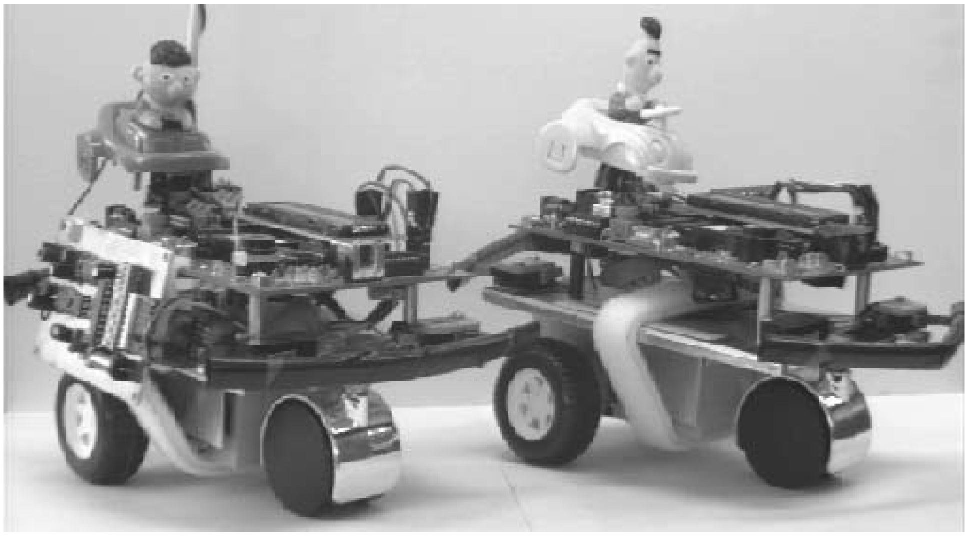

Figure 1: Ernie and Bert

signals using infra-red receivers similar to those found in televisions and VCRs. While Ernie also has infra-red receivers on board, they are not being used in this work – only Bert hears signals from humans. Robots communicate between themselves using a pair of radio transmitter and receiver boards similar to those used in garage door openers. The robots are able to send and receive four-bit packets through the boards. (The transmitter and receiver each run off of a separate 9v battery.) In addition, each robot has a speaker and a 16-character LCD, both used primarily for debugging and monitoring of the robot's activity.

In addition to the infra-red and radio receivers, the sensor robots contain four (front and rear, left and right) bump sensors, left and right shaft encoders, an inclination sensor, photosensitive cells, a microphone, and infra-red emitters. The additional sensory abilities of the robots were not substantively used in the experiments described here.

### 2.2 Description of initial work

In [Yanco and Stein, 1992], we report work in which Bert and Ernie develop a simple two signal language. The robots are given the task of coordinated movement (either *both spin* or *both go straight*). So that communication is required between the robots, only Bert receives the task signal from the outside world.

Upon receiving a signal, Bert has to choose an action to perform (either *spin* or *straight*) and a signal to send to Ernie (either high or low). When Ernie hears a signal from Bert, he must select an action to perform (again, either *spin* or *straight*). Initially, the robots select actions randomly.

The robots learn the task using a reinforcement learning algorithm. Currently, the interval estimation method [Kaelbling, 1990] is being employed. Tallies are kept of the number of times a particular action has been performed on an input and the number of times that positive reinforcement has been received. An optimization function is used to select the next action to be performed on a given input. The robots only receive good reinforcement (+) when the task is performed correctly. If the robots have been told to *both spin* but only one spins, the robots receive negative reinforcement (−). (Understanding of reinforcement signals is hard-coded into the robots.) This *task based reinforcement* is discussed in [Yanco and Stein, 1992].

The robots learn the task and a private robot language after five to fifteen instruction–action–reinforcement cycles. A sample run is shown in figure 2.

Already we have developed an adaptable language. Once the robots have converged upon a robot language, they are able to relearn what the signals mean. For example, if we change the input that originally meant *both spin* to *Bert spin and Ernie go straight*, the robots can relearn the task. While this new training is going on, Bert may decide to send Ernie the signal that they have already agreed upon as *straight* or the robots may reassign the meaning *straight* to the signal that Ernie used to interpret as *spin*. The relearning process generally takes a number of steps equal to the number of training runs (since negative reinforcement must be received enough times to make the robots want to try a new behavior rather than sticking with the behavior that used to provide positive reinforcement).

## 3 Looking forward

There are many possible extensions to the work described above. Two extensions are discussed below.

### 3.1 Compositional language

The space of possible actions and signals in the initial work was intentionally kept very small. In the reinforcement algorithm, two variables must be kept for each possible action on each possible input. Thus, the required variables grow exponentially with each additional action added. In a simulation, memory and time may not matter; however, this is a real issue for autonomous robots with limited memory that we want to act in real-time.

Our current goal is to have the robots develop a compositional language. In a compositional language, there are

| | Appropriate action | Leader's | | Follower's action | Reinforcement |
|---|---|---|---|---|---|
| | | action | signal | | |
| 1. | $\uparrow\uparrow$ | *spin* | low | *spin* | $-$ |
| 2. | $\bigcirc\bigcirc$ | *spin* | low | *straight* | $-$ |
| 3. | $\uparrow\uparrow$ | *straight* | high | *spin* | $-$ |
| 4. | $\bigcirc\bigcirc$ | *straight* | high | *straight* | $-$ |
| 5. | $\bigcirc\bigcirc$ | *spin* | low | *spin* | $+$ |
| 6. | $\uparrow\uparrow$ | *straight* | high | *spin* | $-$ |
| 7. | $\bigcirc\bigcirc$ | *spin* | low | *spin* | $+$ |
| 8. | $\bigcirc\bigcirc$ | *spin* | low | *spin* | $+$ |
| 9. | $\bigcirc\bigcirc$ | *spin* | low | *spin* | $+$ |
| 10. | $\uparrow\uparrow$ | *spin* | low | *spin* | $-$ |
| 11. | $\uparrow\uparrow$ | *straight* | high | *straight* | $+$ |
| 12. | $\uparrow\uparrow$ | *straight* | high | *straight* | $+$ |
| 13. | $\bigcirc\bigcirc$ | *spin* | low | *spin* | $+$ |

Figure 2: A sample run. The desired behavior is *both spin* on input $\bigcirc\bigcirc$, *both go straight* on input $\uparrow\uparrow$. After thirteen iterations, convergence is reached.

words and relationships between words. For example, the robots may learn a word for "go straight" and modifiers such as "quickly" and "slowly". The advantage of a compositional language is that the robots need only learn each concept once, rather than relearn it every time it reappears in a new sentence. This is similar to English; we understand words and how they fit together and need not relearn everything when presented with a new sentence.

The reuse of concepts on robots will save both learning time and memory. If the robots had to build a reinforcement table for each new sentence, they would soon run out of memory. However, if the robots learn the words separately first, much less memory is required to learn composed sentences. Also, the amount of time necessary to learn the composed utterance should be much smaller than the the time required to learn the meaning of the whole utterance without any clues to the meaning of the parts.

### 3.2 Autonomous reinforcement

Once the robots have developed a compositional language, they will be able to explore the world and talk to one another. If they encounter a new situation, they will either need to compose a new sentence or learn a new concept. At this time, all reinforcement is provided by a human signal; however, for the robots to be truly autonomous, they would need to reinforce one another's behaviors. For instance, if Bert wanted Ernie to help him move an object, Bert would send Ernie a signal that he means to be "help move object". If Ernie goes off and continues to explore the world, Bert should send Ernie negative reinforcement.

Thr robots would need to have additional capabilities to be able to reinforce one another. Most importantly, they would need to have an understanding that there are other robots in the world that they are able to influence. This understanding could come through a vision system that recognizes other robots and the actions the other robots are performing. It could also come through an action echoing system provided to the robots in software. For example, if Ernie went straight, he would broadcast the signal "I went straight". This is a simpler method of getting the robots to reinforce one another since adding vision to the robots would be no small task.

To reinforce one another, the robots could continue to use the built in understanding of reinforcement signals. Alternatively, the robots could make changes to the world to reinforce other robots. For example, suppose Bert wanted Ernie to turn right, but Ernie went straight instead. Bert could then move an obstacle in front of Ernie that could make Ernie turn right.

Self-reinforcement is necessary if we want the robots to act autonomously in novel situtations. In space, human reinforcement could not arrive quickly enough from Earth to allow the robots to react to changing situations in a reasonable amount of time. Also, self-reinforcement would be useful in situations that are hazardous to humans. If human–to–robot communication were destroyed for some reason, the robots would be able to carry on with their tasks by reinforcing one another.

## 4 Acknowledgements

The initial work has been directly inspired by work done by John Shewchuk [Shewchuk, 1991]. Portions of the research described here were done jointly with Lynn Stein, who has provided immeasurable guidance. I'd also like to thank Candy Sidner, Rod Brooks, Fred Martin, Ian Horswill and Jeanne Speckman.

The author is supported in part by a scholarship from the Digital Equipment Corporation. Support for this research was also provided by the Gordon S. Brown Fund of the Department of Electrical Engineering and Computer Science at MIT. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124.

# References


[Kaelbling, 1990] Leslie Pack Kaelbling. Learning in embedded systems. Technical Report TR–90–04, Teleos Research, June 1990.

[Martin and Sargent, 1991] Fred Martin and Randy Sargent. The mit sensor robot: User's guide and technical reference, October 1991.

[Sargent and Martin, 1991] Randy Sargent and Fred Martin. IC: Multi-tasking interactive C for the 6811. IC Version 2.5, October 1991.

[Shewchuk, 1991] John P. Shewchuk. Ph.D. thesis proposal, 1991.

[Sutton, 1992] Richard S. Sutton. Special issue on reinforcement learning. *Machine Learning*, 8(3–4), May 1992.

[Yanco and Stein, 1992] Holly Yanco and Lynn Andrea Stein. An adaptive communication protocol for cooperating mobile robots. MIT AI Memo 1379, MIT Artifical Intelligence Lab, August 1992. Submitted to SAB-92.