The Robotic Multi-Agent Development System: Simulating the Physical World

Rebecca Morrison
rmorriso@scrippscollege.edu

August 24, 2007

Abstract

The Robotic Multi-Agent Development System (RoMADS) is a project investigating the theory of single and multi-agent learning. Using an experimental platform, the goal of the project is to reach an analytical theory of learning by studying one or more agents as they adapt to an environment. My goal was to build a computer simulation of the RoMADS robots that would mimic as closely as possible the physical experiment, which involves small robots moving on a table within the arena walls. This simulation will allow the RoMADS team to predict the behavior of the actual robots in situations surpassing the physical limits of the experiment.

1 Background

The Dynamics of Learning project at UC Davis, part of the Science and Engineering Center, is investigating the the process of learning using the methods of dynamical systems, computation and information theory, statistical mechanics, and chaos theory. The RoMADS project uses an experimental foundation to develop the theory of single and multi-agent learning. It is quite unique in that most agent-based engineering projects rely on specific situations and determined environments to control the behavior of their agents. The RoMADS project, in contrast, is concerned with what happens when an agent is placed in some environment with very limited initial knowledge and left to explore on its own. The goal is to reach an analytical theory of learning that can predict the behavior of both natural and synthetic systems.

1.1 What is learning?

To look any deeper into this project, we must first understand what it means to learn. Learning is, put simply, taking the experiences, or interactions through whatever inputs an agent may have, and adapting the behavior according to the input patterns, as well as according to some objective. This idea of an objective is an important and very subtle point. First of all, these robots should learn without depending on human preconceptions. However, when they are learning, they must have some sort of objective, some notion of what is "good" or "productive." Therefore, there must be some subjective input of what this objective should be.

Another way to describe learning is to think of an agent in some world, and every time it needs to make a decision, it is presented with an array of probabilities describing the possible "goodness" of every action. Learning is using the results of past actions and changing the probabilities of the choices that it has for future actions. At the most basic situation, with a single robot and its environment, the agent should adapt its behavior to achieve an objective, starting with random motion and moving towards maximizing some value, such as distance traveled before hitting a wall. From this point, we may add complexity, such as additional robots and more complicated arenas.

1.2 RoMADS: Stage I

In the previous stage of the project, the RoMADS team built and studied the first models of the robots. These were cube-like robots about six inches in height, with four sensors, one on every bottom corner. In a typical experiment, the agents were placed in a rectangular or elliptical arena, starting without any descriptive state; that is, they entered their world in a null state. An agent began moving only by making random decisions, and its sensors were triggered when it hit a wall or another robot. It subsequently would decide which way to go, although it could not tell the difference between a wall and another robot. Equivalent behavior, as a more natural example, is a newborn: it can wiggle legs and arms and move its eyes from side to side, but it does this without any clear objective.

From the initial null state, the agent was able to recognize and add more states as it had more experiences. However, at this stage of the project, this adding of states was ad hoc: there was no objective function followed in order to add them. In many cases, the process was still effective since, for example, an agent could learn to follow a wall instead of repeatedly bumping into one. However, there was no way to measure whether or not this was optimal learning.

In several cases, a robot with more detailed states and directions was able to move through the environment more effectively. As figure 1 shows, the more ideally engineered robot traveled more distance in a given amount of time than the one who was left to learn. Looking at this graph, an obvious question arises, "Why let the robots learn instead of engineering them in a certain way if that is more successful?" In answer to that, consider sending a robot to Mars. With a more determined engineering approach, we would need to know the whole set of parameters to describe the environment, but of course, we don't. There may be aspects such as wind, solar radiation, or the rockiness of the terrain that we cannot foresee or cannot describe. If we can send the robot there with the ability to learn on its own, then this is not a problem since we are not required to describe beforehand all the possible states.

1.3 RoMADS: Stage II

In the current stage of RoMADS, a more advanced batch of robots are being built with several hardware

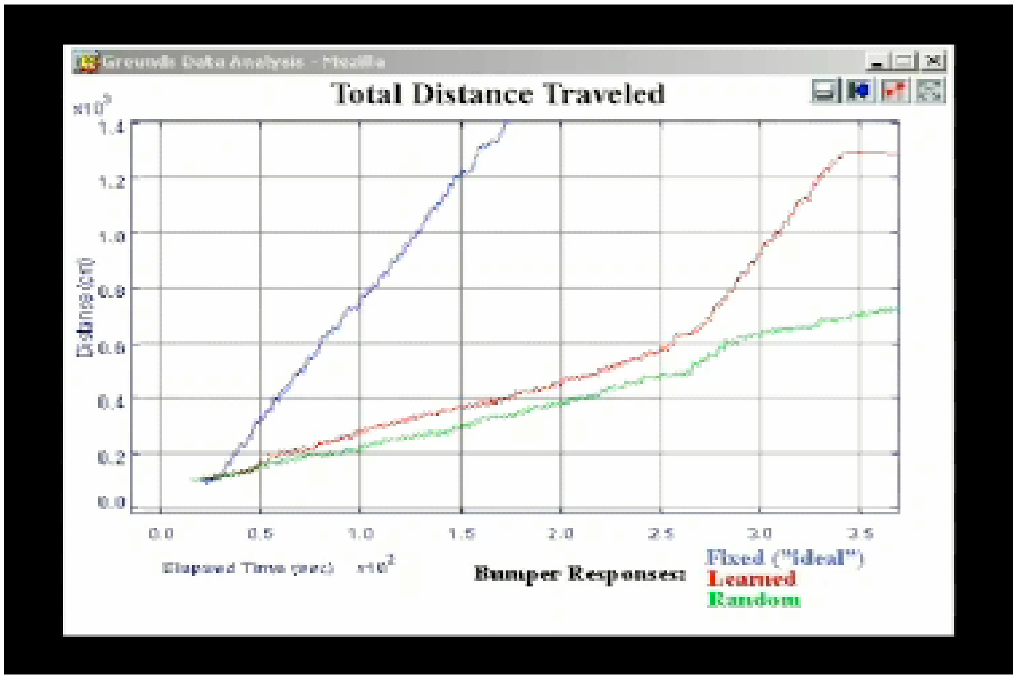


Figure 1. The learning robot initially behaves similarly to one making random decisions. As it adapts to the environment, the distance per time it travels approaches the "ideally" engineered robot. For more information on this project, see [2].

improvements over the last. For example, they are slightly smaller, move much more smoothly, and can distinguish between a wall and another robot. They use six infrared sensors to identify one another, and four infrared sensors around their base to detect the walls of the arena. This is important because it allows for the distinction between something static and something moving.

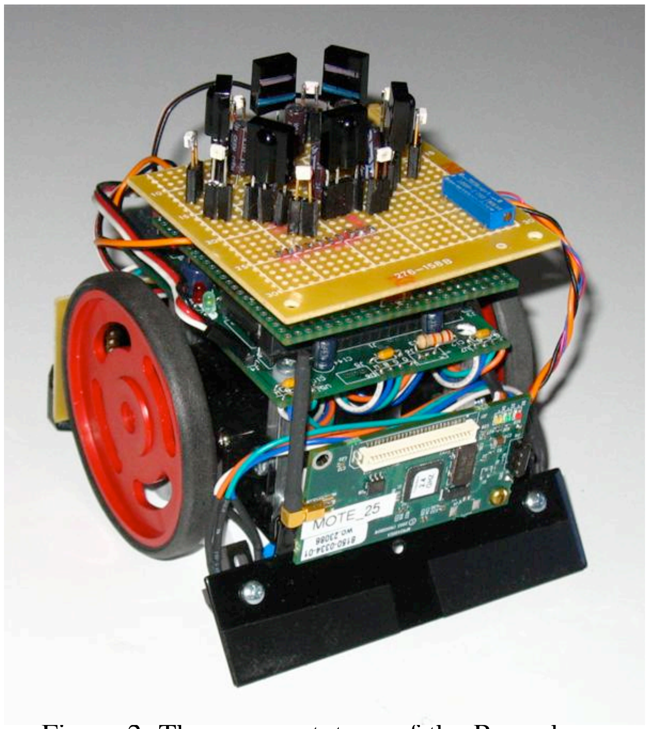


Figure 2. The new prototype of the Romads robots rely on six infrared sensors around the top to identify one another. The four wall sensors are located underneath at each corner of the base of the robot.

As the learning methods are improved, there are some techniques we would like to pursue:

1. Collective learning. Currently we do not know how to share the information accumulated from one robot to another. Two robots in disparate environments may both have knowledge of some state or object, let's call it K. However, they may have learned about K through very different experiences. For example, consider robot A who goes to a cafe every afternoon and eats a cookie, and so associates the concept of cookie from that of cafe. Next consider robot B, who eats cookies every time it is at grandma's house. Then they both know what a cookie is, but they have learned this through completely different associations. Thus, if robot A is at the cafe and B is at grandma's house, how do they share the information that there are cookies at each?

There is another issue at hand. We not only want the robots to share information when they both have knowledge of object or state K, but we want them to be able to learn from the experiences of one another. For example, suppose A lives in the desert and B lives in a forest. Then, B decides to travel to the desert. We want to determine how A may transmit to B the knowledge of favorable and unfavorable desert states so that B may benefit from the knowledge that A already has.

This possibility of sharing information creates a sort of super-organism, in that each individual doesn't experience everything, but by sharing knowledge, we no longer require that to happen. It becomes a sort of super robot that can gather data from disparate non-local environments simultaneously. Clearly, there is a much broader model of its world when sharing is possible. Another everyday example of this is the knowledge that parents try to pass on to their children.

2. Active Learning. Another path we are interested in following is active learning. Typically, machines learn passively, that is, they sit in some state and are fed data, like a teacher finding the pupil material to study. In a contrasting scenario, a student graduates from college and seeks out what is "interesting" on his or her own. With this process of active learning, we must decide on what we deem "interesting." In this project, we define it as the information between random fluctuations of noise and completely predictable static information. For example, if we were to look at some binary data stream, a completely random output is uninteresting. On the other hand, a completely determined pattern, such as 110110110, is also uninteresting. So we switch channels and look for something new. Moving to the context of the robots, we would say that if it finds a straight line to follow, it is not learning and this is boring. And of course, if it bumps back and forth, always fluctuating without structure, this is also boring and there is nothing intelligent here. By using information theoretic measures, we quantify what is interesting. This is also where this project becomes more rigorous than the last. An attractive model of active learning is called Reinforcement Learning, as we'll see later. As the project continues, this will be the method to determine how the robots will choose what to do for every action.

2 Simulation

As a robot moves about its environment, we gather a stream of data of what it senses and the actions it makes. A fundamental point of this project is that there is some process going on in order to create the data stream. We, in turn, are taking the data stream and reverse engineering. As an analogy, suppose I were to listen to what you say and watch what you do. Can I, then, reconstruct your mind? In other words, we are receiving all of the information; can we reconstruct the mechanism that is producing it?

More concretely, consider a single robot in its environment. Then, the environment is the bigger mechanism feeding back the data. By deciphering the information that returns, we then try to reconstruct the arena. In some sense, we may reach a concept of space solely by which sensors get triggered and how long the motors have run.

My work this summer has been to complete a computer simulation, written in Python, that behaves as closely as possible to the physical system that we see on the table. Initially, I worked with previously written code to create a more complex and realistic simulation. The point of the simulations is to allow for predictions and studies that surpass the physical limits of the experiment. In this project, the goal is to build twenty-four of the new RoMADS robots. However, it is significantly more interesting as we allow the number of robots, n, to increase. Emergent behavior, such as flocking and cooperative behavior, is much more likely at large n. Therefore, if we can align the behavior on the table with the computer simulation for twenty-four robots, then once we pass our physical experimental limit for n, we can more confidently make predictions about what would really happen on the table with large n using the computer simulation.

A natural question, then, becomes, "Why have the physical experiment at all?" This answer goes back to what I mentioned earlier about unintentionally adding too much information or our own biases to the simulation, as this may distort what we are really trying to study. These distortions can have huge and incorrect consequences. The computer simulations are running within set capabilities that do not necessarily recreate what is going on in the physical world. There are always unforeseeable unknowns in the real world which may escape a computer simulation. For example, when we are dealing with the robots on the table, they have no knowledge of their position in space- their x and y coordinate on the table. However, in the computer simulation, there is built-in information in the objects moving around on the screen such as internal position. Without even trying, just by creating an object on the screen, we have already empowered it with more knowledge than we will give the physical robots on the table. Of course, as the programmers, we can decide how we will use this information, but that level of decision and knowledge is what this project is trying to avoid, and what makes it unique.

Another significant aspect of my work was to implement the graphics of the simulation. Although the graphics really function as a representation of the simulated robots, they are necessary in the early stages of building the simulations in order to verify that the simulation is behaving as it should.

3 Results

Currently, the computer simulation well mimics the physical robots when they are behaving according to random decisions. Several unrealistic effects of the original programs were eliminated. For example, the first simulated robots would overlap with each other or escape through the arena walls. One of the first steps of my summer project was to fix bugs such as these. Also, the order of such processes such as detecting another robot, compiling all the sensory data, and transmitting this data to the decision-making mechanism is analogous to those of the actual robots. At this time, the initial stage of a more rigorous learning method is being developed, and the overall structure of the simulations were built so that this may be easily added to the current program.

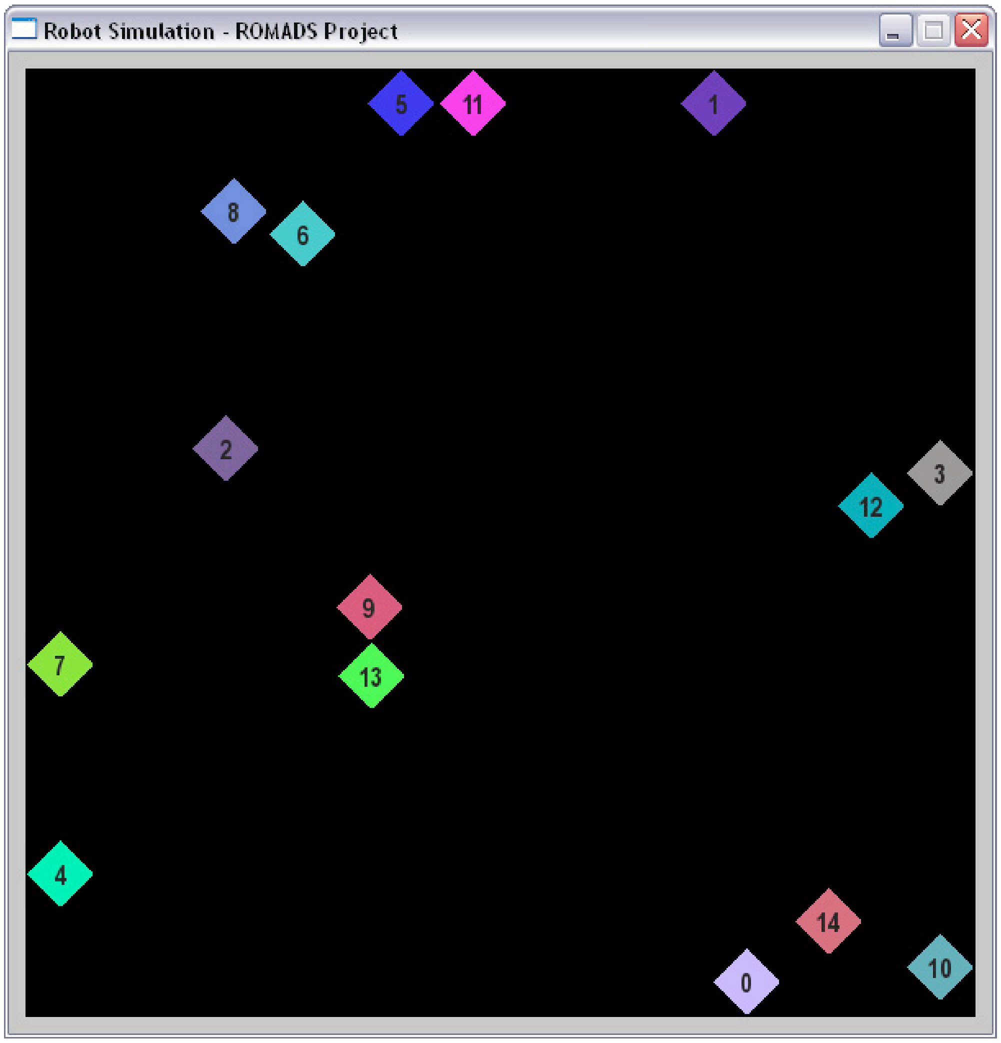


Figure 3: An example of the graphics representing one of the simulations. The numbers are the identification tags of each robot.

4 Future Work

The simulation and graphics will surely be modified as the project continues. Also, as mentioned above, the learning process is yet to be connected to the program. This is the next immediate step after what I have done this summer, and it will implement the learning process called Reinforcement Learning [1].

The four parts of Reinforcement Learning are states, actions, rewards, and a policy. In this process, the agent interacts with the environment and the environment provides rewards. For an agent in a particular state, we want to know what the optimal action to get the most reward is, or the least amount of punishment. Also, we think of the states as the foundation for making decisions, the actions are the choices, and the policy is the set of rules the agent follows to make every decision. Figures 4 and 5 show a simple example of Reinforcement Learning.

To conclude with an interesting question: What does it mean to say that I know someone? Then when he’s in a new situation, I’ll say, “I know what he’ll decide to do.” I have built a model based on our interaction and have inferred how he’ll act. I have, to some extent, reconstructed what the mechanism is behind his decision-making process. Like the robots, the better the model, i.e. better I know him, the better the prediction will be and hence the better connection between the past and the future. This connection between interpreting the past and successfully predicting the future is a crucial theoretical concept and practical concern for all areas of physics, mathematics, and computer science and in some sense, everything we, as thinking, learning human beings depends on our ability to understand the connections between the past and the future.

Figure 4: The set up of a simple Reinforcement Learning example:

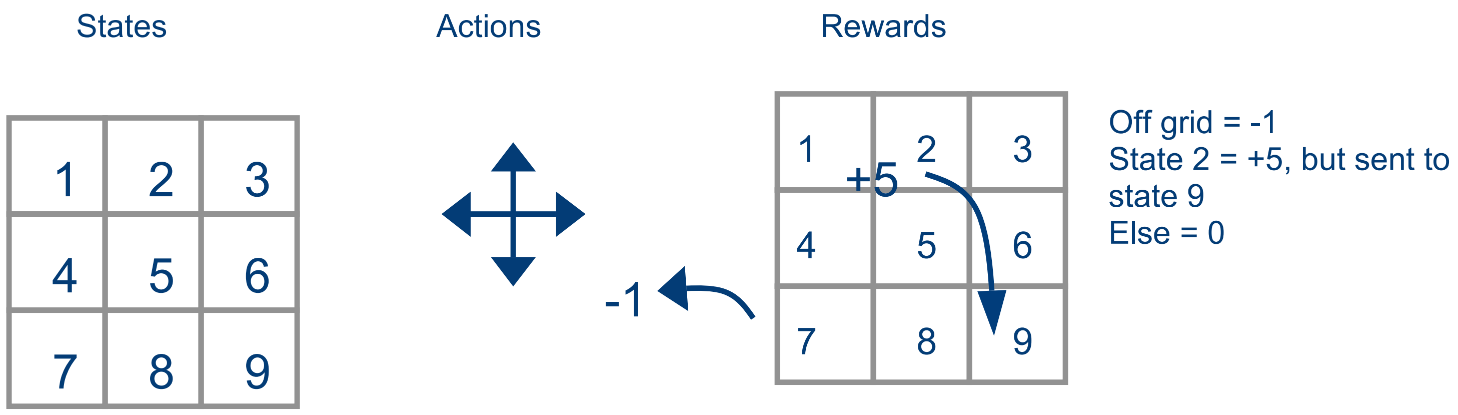
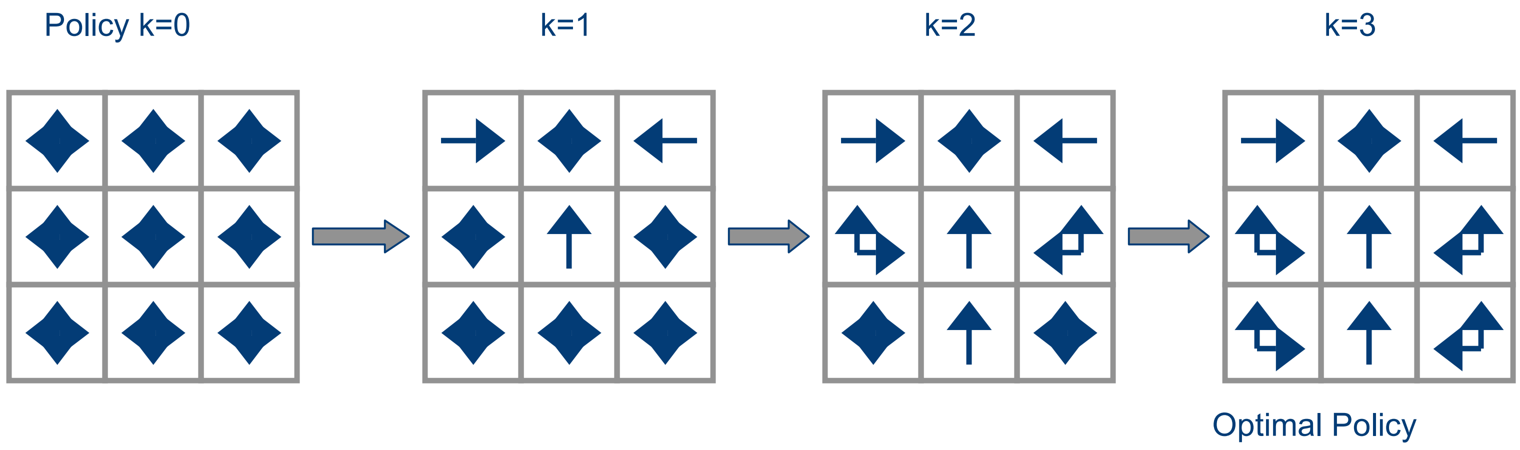


Figure 5: At every step, the policy is changed as the agent adapts to received rewards. At the beginning, every direction looks equal. As the agent moves through the environment, it adapts to move towards state 2 in order to receive the maximum reward.



References

[1] A. G. Barto and R. S. Sutton. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[2] RoMADS Project. Website, 2002. http://shell.cse.ucdavis.edu/~dynlearn/robots.htm

[3] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, 2000.