In-Class Design #1 Feedback:

Learning the Paddle Ball Task

(In-class exercise on 2/16/2006)

This exercise required you to design a learning approach to the paddle ball task, using either reinforcement learning or neural networks. You were to assume you had a humanoid robot performing the task, and could assume you had access to any human-like sensing capability or motion, or any sensing capability that is easy to engineer (such as measuring the distance to the ball).

As we've learned by now, any learning approach involves many design decisions, and there is no single correct solution. This exercise was to meant primarily to stir your thinking about how to formulate learning problems, using the techniques we've studied so far.

Here are some thoughts about the solution approaches discussed:

- State:
  - Including a state variable that measures the position of the ball relative to the paddle is a good idea.
  - It is also a good idea to use the elastic force pulling on the paddle be part of the state. However, keep in mind that the elastic force pulling on the paddle will be 0 throughout the downward trajectory of the ball, so it can't distinguish between the ball being near the paddle and the ball just coming back from its furthest point.
  - Because of the points mentioned below (in Actions), the *x*, *y* position of the paddle and the tilt angle of the paddle would probably need to be part of the state in order to deal with some non-vertical ball motion.
- Actions:
  - It is a very good idea to define a "strike" action, which encapsulates a series of motions for one strike of the ball. However, this will probably need to be combined with a paddle motion (see next bullet).
  - Since we defined the robot to be "human-like", this means that the robot will also have some error in executing its motions (so it can't necessarily perform exactly repeatable actions every time, or move exactly in a plane). So, your solution would need to be able to address some degree of error. This means that the paddle would need to be able to move to correct for the ball going off-center; so, actions would need to involve tilting the paddle and moving it in the plane parallel to the floor. Certainly, as pointed out by some group members, the elastic string will tend to move the ball back toward the paddle. But, there will still certainly be drift over time.

Reinforcement Learning:

- Reward:
  - One obvious state for receiving negative reward would be the state when the ball misses the paddle. How this negative reward is propagated is important. It should be propagated at least to the last hit, but perhaps not too much further back into the future.
  - Some positive reward is needed, or else the robot would just learn to rest the ball on the paddle (to avoid negative rewards). So, a positive reward could be given every time the ball was hit successfully. This reward could vary, with hits that lead to a large number of future hits rewarded more. Again, this would be an experimental design decision.

Neural Networks:

- Output:
  - When using neural networks to learn control, the output is typically an action. Recall the ALVINN example, in which the outputs corresponded to how to turn the steering wheel – varying from hard left to hard right, and values in between. So, you would have to encapsulate each possible combination of motions and have an output unit that represents each of these motions.