

Ankur Desai
Weekly Mentorship Log
Week 3: August 14 – August 18

Goals for the week:

- Learn the details of how the AIBO connects wirelessly in order to get permission to set up a connection in the lab
- Investigate possible methods of localization of the AIBO
- Create a wrapper interface so that the artificial physics C library can be accessed using Python

Wireless networking permission:

In order to get clearance to set up a wireless network to communicate with the AIBOs in the lab, I needed to figure out the exact nature of the connection between the host computer and the AIBO. To do this, I used a port scanner to determine where and how the connection was occurring. I also went through some of the source code to figure out how to change the ports used.

Localization:

In any robotics project, localization is a major issue. The problem is that robots need to know where they are at all times in order to do anything. Figuring out where they are can be rather difficult when they are equipped with only an infrared rangefinder and a low quality camera. With the AIBOs, the most likely method of localization is through the odometry data. That is, if we know where the robot was when he started, and we know where he moved, we should be able to figure out where he is now. However, this method is a bit unreliable because of errors accumulated while moving. It will probably be necessary to implement some sort of basic particle filter in the future.

C/Python Interface:

In order to turn the artificial physics C library into a usable Python module, it was necessary to create a wrapper for the pertinent C functions. This process was partially automated using a program called SWIG, which allows users create a slightly simpler

file, and then uses that file to automatically create the wrapper. Once this wrapper was created, I compiled all of the necessary binary files and the wrapper into a shared library. This library can be dynamically loaded into the Python interpreter at any time. I also created a Makefile to automate the process of compiling the PyAP module.