Goal
The objective of this project is to assist in the development of a robot software system for use in a human-robot interaction study. The purpose of this study is to investigate humans interacting with robots through imitation. This requires a robot that is capable of evaluating human poses, making human motions (from the torso up), and providing feedback. Much of the framework for the project has already been developed by my advisors - my task is to improve some existing parts and to extend the existing system by adding a visualization tool.

Project Overview
This project will be a part of a larger robotics project being conducted by Professor Scassellati’s lab. Members of the lab will conduct demonstrations with children from local schools where they interact with the robot. They will be playing a simple “Simon Says” type imitation game where the robot will make a pose and the child will have to mimic the robot’s position. The robot will then assess the child’s position and offer feedback depending on whether or not the child correctly imitated the pose.

The “imitation pipeline” is diagrammed below. I will be my project will be focused on the areas highlighted in orange:
All of this code is going to have to be fast - the robot needs to be able to mimic the human in near real-time. As such, it may be necessary to optimize the code for speed.

**Deliverables**

There are two main deliverables for this project:

- **joint angle code**
  - Given the positions of joints in three-dimensional space, return the joint angles
  - This will entail learning an inverse-kinematic method for inferring joint angles

- **visualization code**
  - Show a 3-D representation of the poses of both the human and the robot as well as a representation of the joint angles
  - This visualization is an important aspect project because it will help the children gain an intuition for how the robot works

**Secondary Objectives**

An optional secondary objective (time-permitting) is to incorporate more helpful feedback than just “your position is incorrect”. For example, the robot should be able to offer feedback like “move your right arm down a little”.

**Department Requirements**

Per CS dept requirements, I will complete an abstract, a written report, and a website describing the project.

**Skills Acquired**

Although I’ve been doing a lot of web/mobile programming for the past five years, I’ve done virtually nothing with robots or the Kinect. To complete the above objectives, the following skills are required:

- **Robot and sensor programming**
  - **Robot Operating System (ROS)**
    - *ros core* - the basic framework that allows for control of the various components of the robotic system
    - *ros tf* - a library that enables the coordinate transformations needed for working with robotic joints in 3-dimensional space
    - *ros actionlib* - a library that enables modeling of events and actions. This will be used when programming robotic movements
    - *ros smach* - a library for state machines that will be used to build up the robot’s behavior patterns
    - *rviz* - a visualization library that will be used in the visualization module
    - *ros bag* - a logging tool to help with debugging

- **Kinect**
  - *open ni* - basically an SDK for the Kinect, this is a platform that enables the extraction of a human skeleton model from a video feed. This will be used to assess human poses.
  - *pcl* - A point-cloud library that (time-permitting) may be used to enhance the visualization tool by doing some video filtering

- **Nao robot**
  - *nao_driver* - this enables a Nao robot to be controlled through ROS
  - *nao_led* - this is code to control the default eye LEDs on the Nao robot
- **turn_taking** - this is code that is part of the imitation pipeline described above
  - Application of mathematics relevant to human pose and imitation
    - Linear Algebra - used to determine joint angles from Cartesian coordinates
    - Probability - used to determine appropriate responses given joint angles

**Timeline**
I go to lab meetings on Mondays and meet one-on-one with Dr. Feil-Seifer on Tuesdays. The milestones below are due at the start of the week that they are listed under.

**Week of Jan 30**
Finalize proposal, start on ROS tutorials

**Week of Feb 6**
Base ROS proficiency

**Week of Feb 13**
Mathematics review and beginning algorithm development

**Week of Feb 27**
Joint angle prototype ready for evaluation

**Week of March 19**
Joint angle node revisions completed

**Week of March 26**
Visualization module project goals identified

**Week of April 2**
Prototype visualization system and evaluation

**Week of April 9**
Visualization module revisions completed

**Week of April 16**
Project writeup / website

**Week of April 23 - READING WEEK**
Final wrap-up