Project Proposal

Human Interface for Robotic Control

SYSC 4907 – 4th Year Project

Team Members:
Brandon To 100874049
Minh Mai 100845949
Yuzhou Liu 100853392

Supervised By:
Richard Dansereau

October 2 2015
Carleton University

Table of Contents

Objective

The project aims to develop a robotic arm system that mimics human movement. Integration of feedback control will allow the robotic hand to pick up an egg without cracking it. Furthermore the system can be remotely controlled over the internet. Focus will be put on system performance with minimal latency and high accuracy.

Motivation

In the past, developing robotic systems with complex hardware designs were great challenges due to unavailability and high costs. The challenges with hardware limited the possibilities of developing robotics systems for time and budget constrained projects.

With recent advances in 3D printing technology and motion sensor capabilities, integrating complex robotics hardware designs are now practical and cost-effective. This greatly expands the opportunities available for students, researchers, and hobbyists.

The vision of this project is to develop a true embedded robotics platform that can serve as the foundations for developing robotic applications to replicate human movements. Possible applications for this project range from manufacturing plant assembly lines, to bomb diffusion, and remote surgery. Features such as remote control over internet and record/playback of user action were added to the platform to widen the scope of future development opportunities.

The work done in this project will be made open source to contribute back to the robotics community. Our individual goals are to apply the knowledge in both electronics and software development gained throughout undergraduate studies to develop a complete system. This exercise will provide experience in firmware development, socket programming and project management.

System Design Goal

In order to measure the success of our project, the following system requirements must be achieved at the end of the design and development life cycle:

- A robotic arm system with 4 degrees of freedom that can pick up an egg without harm
- An interface application to control the robotic arm using Leap Motion sensor
- Accuracy between user input and robotic system output should not differ by more than 1cm in any direction of the XYZ space
- A reliable P2P architecture to control the robot over the internet with latency under 200ms
- System operating voltage: 4.8 – 6.0 Volts, current: 2.5A max.

Technical Overview

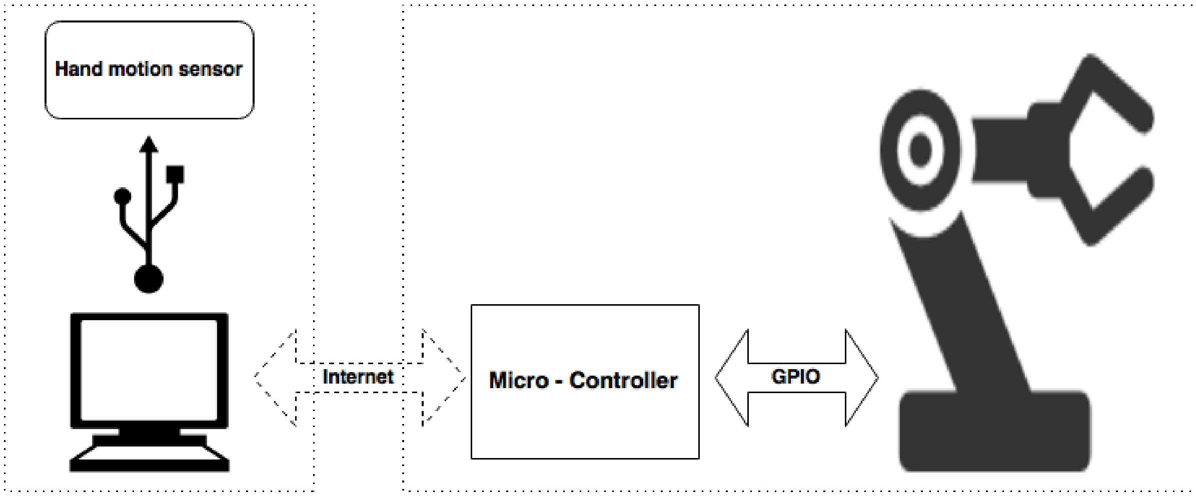

Figure 1: High Level System Architecture

In this project, the motions of the human hands are captured and sent to a computer. The program on the computer processes the data and sends it to the microcontroller via internet. Upon receiving the data, the microcontroller guides the robotic hand to replicate user movements. The movement of the robotic hand is driven by a set of servo motors. The controller controls the rotation and speed of each individual motor to achieve the user intended movement. Thus to accurately account for how the system is behaving in real time, it must have a feedback mechanism. The feedback system will determine when to increase/decrease the speed or stop the motor from operation. Figure 2 shows the concepts of a closed loop system used for the project.

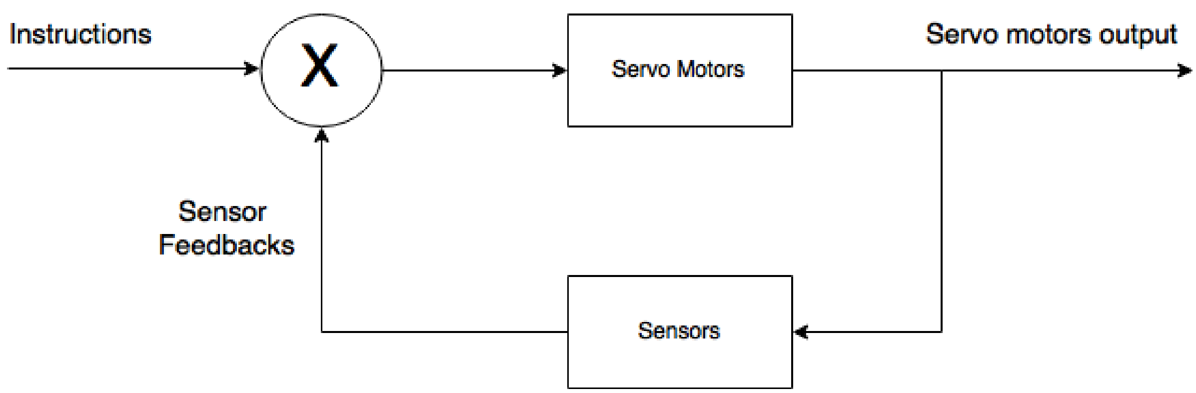

Figure 2: Closed Loop Feedback System Controlling Motors Movement

In figure 2, there are a set of sensors placed on the fingertips of the robot arms which gives pressure feedback to the controllers. Based on these feedback, the controllers adjust the operations of the motor to achieve the correct movement.

The hardware design consists of four main components: a hand motion sensor, a computer, a microcontroller, and a 3D printed robot hand. The hand motion sensor used in this project is the

Leap Motion. A Texas Instrument CC3200 SimpleLink Wi-Fi LaunchPad is chosen as the microcontroller. The hand will come from the InMoov open source robot project.

Leap Motion Controller

The Leap Motion controller recognizes and tracks hands and fingers. The sensors give high precision position tracking of each individual finger joint as seen in Figure 3.

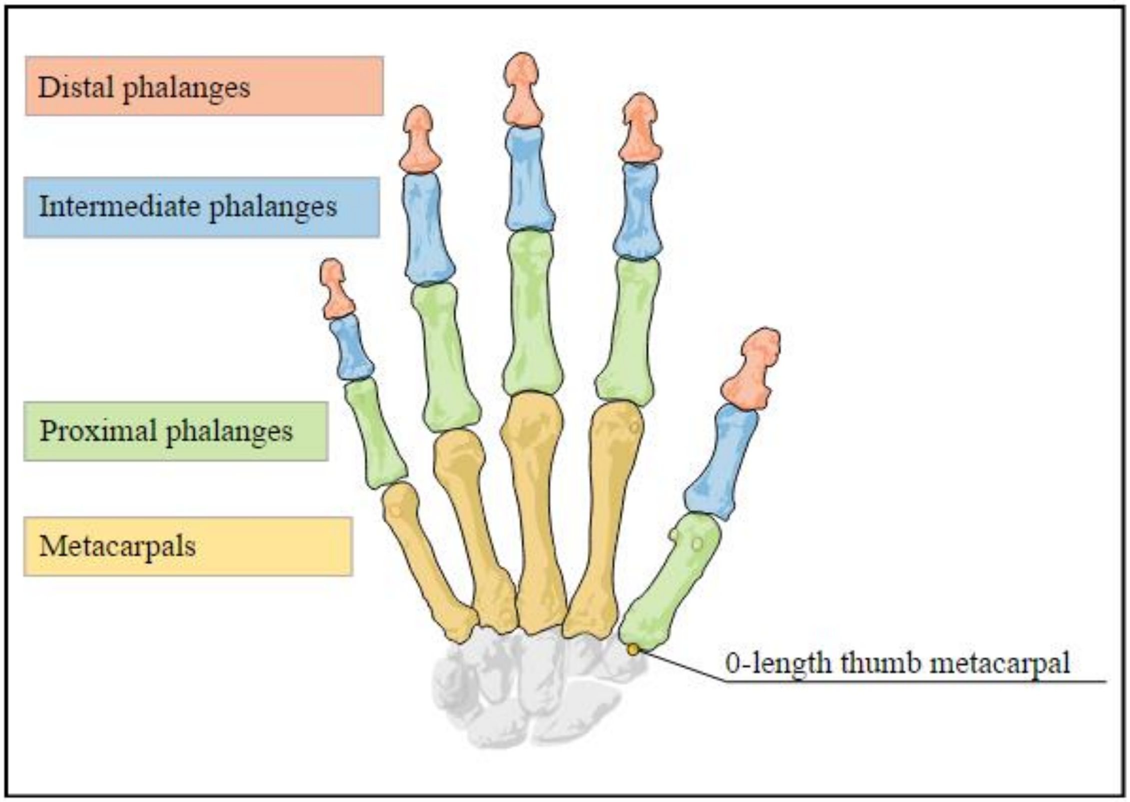

Figure 3: Hand showing individual finger joints

The software API provides methods to track the normal, basis, and direction vectors of the individual finger joints and hands, as well as the Cartesian coordinates in 3D space. The API also provides velocity vectors for the palm. At a higher level, the leap motion detects predefined gestures made by the hand. Motions like waving, swiping, and circling are all tracked by the Leap Motion. For the purpose of our project, gesture detection will not be needed. We primarily want to focus on capturing position and velocity data and translating it into PWM waves to be fed into the servo motors controlling InMoov.

In order to interface with the controller, a separate service must be run in the background that fetches the information and distributes it to applications that are registered to it. This allows many different applications to receives data from the Leap Motion controller. Our interface application will register to this service and request information at a frequency high enough to be perceived as smooth motion.

Interface Application

The Leap Motion controller is a small USB peripheral device that provides accurate hand and finger tracking through the use of the sensor embedded in the device. A C++ program will be written which query the Leap device for relevant data, serialize the data, and stream it across the network. The decision of whether to use TCP or UDP for this has not yet been finalized. We would like to avoid using UDP if possible, to save ourselves the trouble of implementing our own reliability and network congestion protocol. Knowing this, we will initially develop our system under the assumption that we will be using TCP, and only switch to UDP if the transmission rate does not meet our expectations.

Here are some problems that we have identified that must be addressed in the communication across IP:

- The IP address of the microcontroller would have to be static and port forwarded on the router in order for an application to communicate with it from a different network.
- One group member who used the CC3200 WiFi shield (the WiFi module on our microcontroller) for a different project last year and mentioned that the shield does not work with enterprise WiFi.

To work around both of these limitations, we are planning on bringing our own router (set to bridging mode) to the poster fair demonstration.

Firmware

The microprocessor controls the robotic hand upon receiving the relevant data from the Interface Application. Firmware will be written to use the received data to calculate the speed at which the joints are moving, and will configure the PWM accordingly in order to simulate that movement using the servo motors.

Here are some problems that we have identified that must be addressed at this point in the system:

- The speed at which the robotic hand can move will likely be much slower than the potential speed of our hand. We must handle the cases where the tracked hand moves faster than the servos can handle. As of now, our group believes that we should implement a queuing system in order to buffer the movement input before InMoov mimics it.
- The Leap Motion controller only tracks hand and finger motion. In order to incorporate arm movement, we must improvise. One idea is to only allow 1D arm movement, that is, only allow the arm to rotate along one axis. We can then use one previously untracked measurement (for example, the height of the hand) in order to dictate the movement of the entire arm. This is an overly simplified solution to a very complex problem that stems from hardware limitations, but it could be a first step.

- For the robotic hand to physically pick up items, we would need to include pressure sensors on the fingers in order to detect that we have actually come into contact with the object. The readings from this sensor will notify the microcontroller to stop the robotic hand from gripping too tightly since the tracked hand will likely be out of sync with the robotic hand after it comes into contact with an object.

Milestones and Timeline

The project is divided into 3 milestones:

Milestone 1: Hardware Complete

The robotic hand and arm are assembled with functional servo motors.

Hardware is tested for required range of motion.

Milestone 2: Interface Established between Leap Motion and Robotic Hand

The robotic hand is able to perform basic hand movements with Leap Motion as input.

Test that the hand and forearm can be controlled using leap motion.

Milestone 3: Remote Control over IP and Implement Closed Loop System

The robotic system can be controlled remotely over Internet Protocol. Closed loop system is implemented using fingertip sensors.

Verify the latency over remote control over IP is smaller than requirement specifications. Test implementation of closed loop feedback system for ability to pick up egg successfully.

Milestone 4: Record motion feature and test entire integrated system.

Implement and test the system's ability to playback recorded movements. Final testing of entire integrated system. Fine tuning will be done here.

Please see Timeline attached in Appendix A.

Required Components

Listed below are the components that we need. These are subject to change.

Hardware	
InMoov 3D Printed Robot (hand, wrist and arm)	$55
Leap Motion Controller	-
TI CC3200-LAUNCHXL	$43
HS-311 Servo Motors (6)	$75
Fingertip Sensors	$25
Software	
C++/Python - interfacing with Leap Motion Controller	-
Berkeley Socket API - IP communication	-
C - Microcontroller programming	-

Table 1: Components List

Appendix A

Project Timeline

ACTIVITY	PLAN START	PLAN DURATION	PLAN END
Project Proposal (Draft)	10-Sep	8	18-Sep
Progress Report	23-Nov	15	8-Dec
Oral Presentation	4-Jan	14	18-Jan
Final Report (Draft)	22-Feb	21	14-Mar
Poster Fair	4-Mar	14	18-Mar
Final Report	19-Mar	20	8-Apr
3D Print Hand/Arm	3-Sep	22	25-Sep
Hand Assembly	11-Sep	14	25-Sep
Forearm Assembly	25-Sep	7	2-Oct
Test Hand/Forearm Range of Motion	2-Oct	7	9-Oct
[Milestone 1 - Hardware Complete]		9-Oct	
Hand Control using Microcontroller	24-Sep	22	16-Oct
Interface Leap Motion with Microcontroller	1-Oct	43	13-Nov
Test Wired control from Leap Motion to Hand	13-Nov	14	27-Nov
[Milestone 2 - Interface Established]		27-Nov	
Remote Control Over Internet	4-Jan	31	4-Feb
Test Latency of Remote Control over IP	26-Jan	17	12-Feb
Fingertip Sensors for Closed Loop System	4-Jan	31	4-Feb
Test Arm Movement and Closed Loop System (Pick up Egg test)	4-Feb	8	12-Feb
[Milestone 3 - Remote Control]		12-Feb	
Record Motion Feature	1-Feb	17	18-Feb
Test Entire Integrated System	13-Feb	16	29-Feb
[Milestone 4 - Verify Entire System]		29-Feb