ESE-519 Final Project Report

Project Title:
GestureBot: Gesture-controlled robotic arm

Team Members:
Aniruddha Rajshekar (arajs@seas.upenn.edu)
Aditya Pinapala (pinapala@seas.upenn.edu)
Lanlan Pang (planlan@seas.upenn.edu)

1. Introduction:

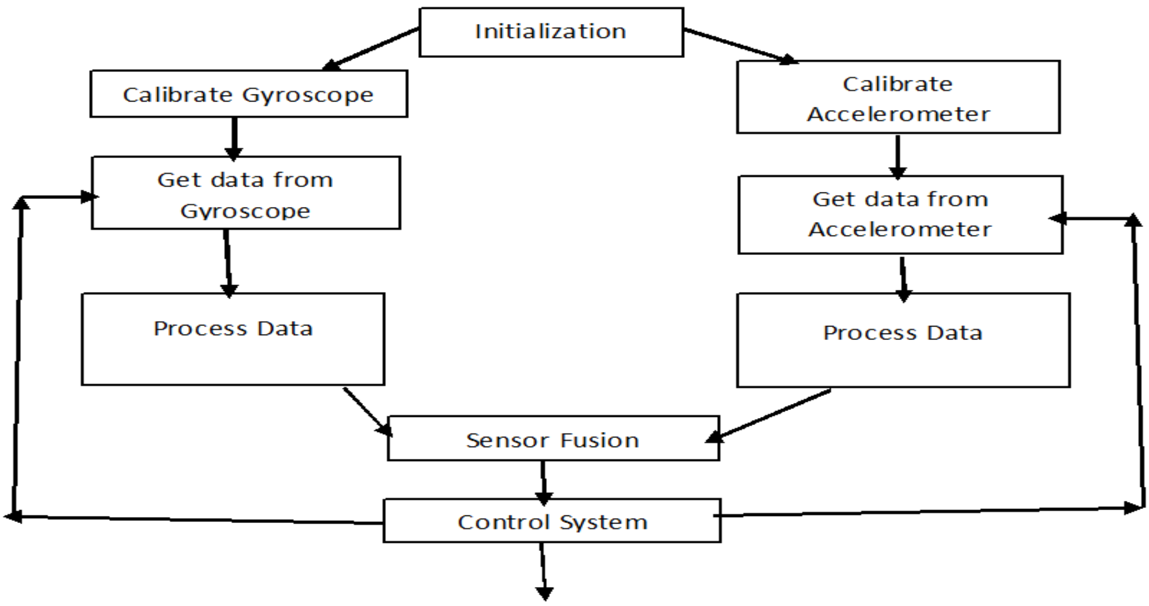
Our project aims to design a gesture-based control system which lets a person use the motions of his arm to control the robotic arm. We built a master system which uses an IMU signal to register the human arm movements and interpret that to create control signals, which is then transmitted via a wireless module, to the slave system (i.e., the robotic arm), which then executes the corresponding actions.

2. Motivation:

The methods commonly used for humans to direct the actions of robots are cumbersome and insufficient for complex tasks. There is growing interest in using hand and arm gestures to supervise robots. Current research in gesture-based interfaces typically rely on computer vision for observing and classifying the human gestures. Vision-based approaches have the advantage of avoiding human-worn equipment. However, cameras often requiring consistent lighting. So, our project aims to develop a gesture-based human interface for robotic arm control, based on various sensors.

3. Steps:

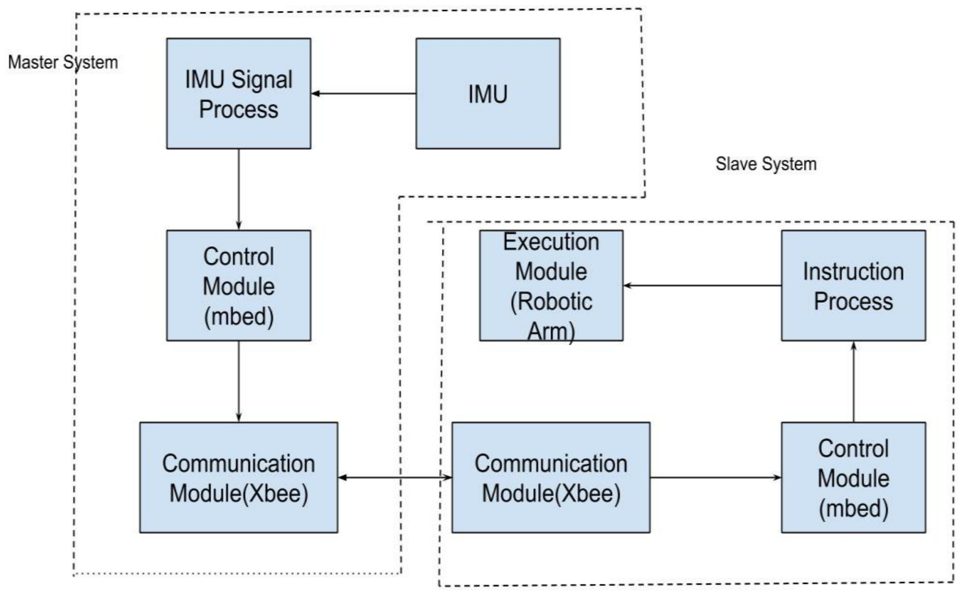
1. We firstly designed a gesture based control system. We wrote a program that learns to differentiate between different hand movements based on the readings of the IMU sensor.
Below is the software architecture diagram of our gesture based control system.



2. Then we assembled a robotic arm. The robotic arm performs basic up, down, left and right motions. The gripper on the arm opens and closes. The arm is controlled by a receiver module that instructs the robotic arm to perform the above mentioned motions.

3. Finally, we had the robot arm move to different directions according to the arm gestures. Based on the reading of the IMU sensors, the transmitter module create a message that is sent to the receiver module. The receiver module that controls the robotic arm then decodes the message and actuates the motion of the arm accordingly.

3. Overview of Subsystems:



System Architecture Overview

Our project is done using two mbed MCUs and consists of two parts. First is the master/transmitter system, where we have the IMU sensors located on the arm of user (we used a glove to package our master system). The readings from the IMU sensor determine the direction of the movement of the arm. This signal is processed and sent to the communication module. The transmitter communication module sends corresponding instruction to receiver system. The receiver system recognizes the current gesture and processes the instruction in real-time to drive the robotic arm move according to the human arm gestures.

4. Master System:

a. IMU sensors

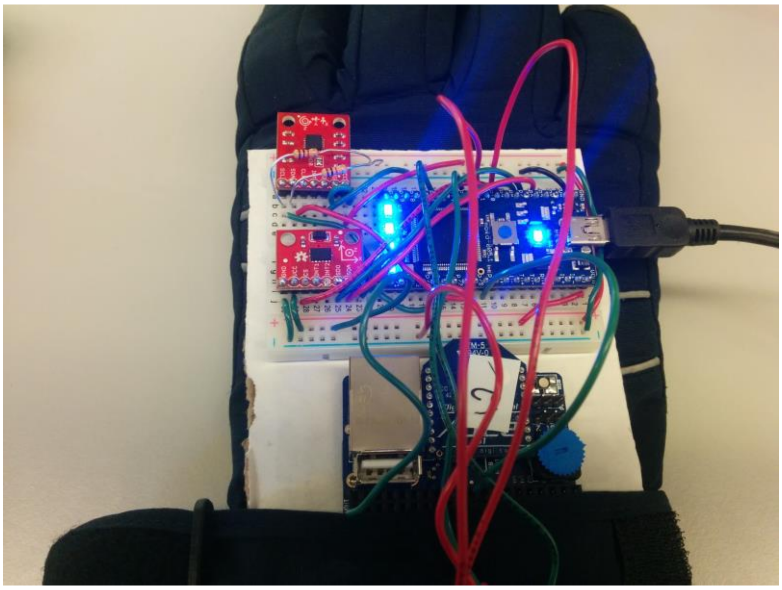
The IMU sensors consists of a Gyroscope and an Accelerometer. The gyroscope measure the instantaneous motion in the X, Y or Z direction. The accelerometer measures the acceleration in the X, Y and Z directions. These values are sampled and taken after filtering. Depending on the values given by the sensor, the direction of motion of the wearer’s arm is determined.

b. Communication Module (Xbee sender)

The Xbee sender module uses the mbed application board to interface with the mbed on the master/transmitter system sends a characters representing left, right, up, down and stationary ('l', 'r', 'u', 'd' and 's' respectively). These are received by the second Xbee receiver module which sends the sender to the receiver system

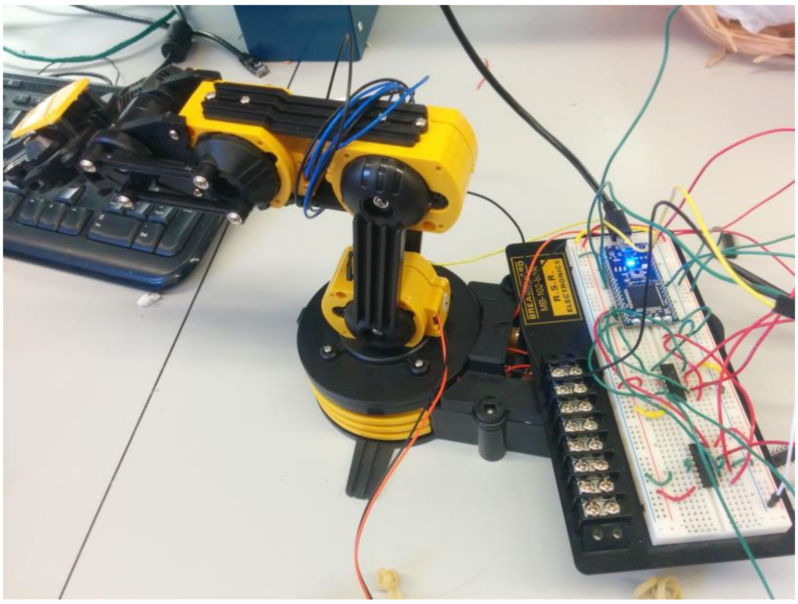
c. Joystick

The joystick was used to control the grippers. We used the joystick that is available on the Mbed application board. The joystick was programmed in a way such that the up and down movement of the joystick resulted in the open and close of the grippers respectively.



Final Package for our Master System

5. Slave System:



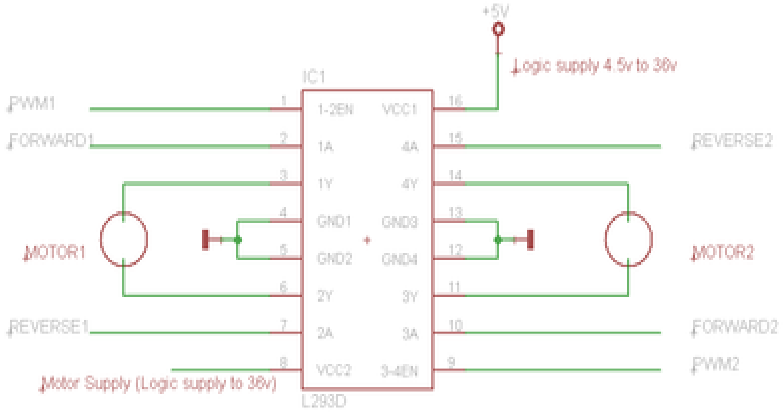
Final Package for our Slave System

a. Communication Module (Xbee receiver)

The Xbee receiver module uses the mbed application board to interface with the mbed on the receiver system and receives either an ‘l’, ‘r’, ‘u’, ‘d’ or ‘s’ and accordingly signals the motors to run in that particular direction.

b. Robotic Arm (Motors)

Our Robotic Arm is controlled by three DC motors, moving the robotic arm in the up/down, and left / right direction and also the open/close motion of gripper. The DC motor are driven using PWM signals from mbed and H-Bridge. We use DigitalOut pin to control the motors. Initially, all the motor DigitalOut pins are set to make them stop and the Pwm time period and duty cycle is set. Each time the control signal is received from the Xbee receiver, the mbed decodes the control signal and translates this control signal to change the value of the DigitalOut pins that control the 3 DC motors. This corresponds to the movement of the robotic arm in that particular direction.
Below is the H-Bridge connection diagram.



Connection Diagram of DC Motor to Mbed

6. Challenges and Solutions

The biggest challenge we faced was that the sensors were very sensitive. Even a slight movement of the hand was resulting in a large variation in gyroscope and accelerometer readings. We used restrictive samples for the gyroscope and used strict threshold values to counteract this problem. This helped us gain better control over the movement of arm.

7. Potential Future Extensions

We spent a lot of time on getting data from IMU sensor and processing the data from it. Maybe we can add EMG sensor in the future. Instead of using joystick to control the gripper, we could use the muscle sensors. The muscle sensors will be used to open/close the gripper of the arm and also control the up/down movements of the wrist of the robotic arm.

Individual Member contributions:

For this project all 3 team member met together in the lab and worked on all the parts of the project equally.

Name	Percentage	Work Done
Aniruddha Rajshekar	33%	Assembly of robot arm Design of master system Design of slave system Code for master and slave system Sensor data processing Sensor data fusion Co-ordination between master and slave system
Aditya Pinapala	33%	Assembly of robot arm Design of master system Design of slave system Code for master and slave system Sensor data processing Sensor data fusion Co-ordination between master and slave system
Lanlan Pang	33%	Assembly of robot arm Design of master system Design of slave system Code for master and slave system Sensor data processing Sensor data fusion Co-ordination between master and slave system