CSE 423: Final Report

Armed and Dangerous

December 12th, 2012

Team Sieben

Craig Cassinat
Max Shirvanifar
Joseph Babb
Scott Baker
Michael J. Astrauskas
Ashoraya Mezdo

Sponsor: Dr. John Black

ARIZONA STATE UNIVERSITY

Table of Contents

Table of Figures 3

Executive Summary 4

Introduction 5

Project Description 5
Project Motivations 5
User Identification 6

Body 7

Scope 7
Project Plan 7
Development Approach 10
Notable Design Decisions 10
Overview of System Architecture 11
Technology and Tools Used 14
Preliminary Results 17
Problems and Risks 18

Summary of Tasks Completed by Each Team Member 19

Conclusion 21

Success of the project so far 21
Lessons learned 21
Future work 22

Citations 23

Table of Figures

Table 1: fall 2012 semester deliverables 8

Table 2: spring 2013 semester deliverables 9

Table 3: Sub-teams 10

Figure 1: Schedule for fall 2012 Semester 7

Figure 2: Schedule for spring 2013 Semester 8

Figure 3: Complete System Architecture [5] [6] [7] [8] [9] 12

Figure 4: The LynxMotion AL5D robot arm 14

Figure 5: A wired 18-sensor CyberGlove 15

Figure 6: A terminal showing plaintext communication with the CyberGlove 15

Figure 7: A Microsoft Kinect 15

Figure 8: An ArduIMU+ V3 and Arduino Uno microcontroller 16

Figure 9: A sleeve with 3 ArduIMUs 16

Figure 10: A simulation of the remote video environment 17

Figure 11: Test operation of the robot arm with gesture control implementation. 18

Executive Summary

Intuitive robotics interfaces have many benefits, such as for bomb disposal or as an aide for individuals with motor control disabilities. Team Armed & Dangerous is interested in designing an intuitive interface for controlling a human-arm-like robot arm in real time, while keeping the software modular and extensible.

The team set out to control a small, heavy-duty robot arm's (LynxMotion AL5D) motion using a Microsoft Kinect – a camera that provides the 3D location of joints of the human body – and control the robot arm's claw using a CyberGlove motion-tracking glove. The project could logically be broken into components. Thus, the team of six split into 3 sub-teams of 2 each: Michael and Craig working on the CyberGlove, Joe and Scott working on the Kinect, and Max and Ash working on the robot arm. Using Microsoft Visual Studio 2012 as the development environment, git (hosted on github) as the code repository, and Google Drive for document sharing, the teams developed these modules in parallel and then tied together by a publish-subscribe bus system.

The team was initially behind schedule, during the design and development phases. However, ultimately integration was in fact finished ahead of schedule. The spring 2013 semester will be used to make some hardware changes to the project, as well as add to and refine the existing software. One change is the addition of a remote video setup that will allow the user to see and operate the robot arm when it is physically in a different location. This will limit feedback. Data from the Kinect was unreliable, with even a still arm occasionally registering as moving, and thus a second change will be the use of inertial measurement units (IMUS – 3-axis accelerometer, magnetometer, 3-axis gyroscope) on each rigid part of the arm to provide motion and orientation information. This replacement for the Kinect will be developed in collaboration with CUbiC's Dr. Troy McDaniel, who is also studying the use of IMUs for tracking human arm motion. Software-wise, the team will develop a graphical user interface for calibrating the hardware and a debugging module for better testing hardware and hardware interaction.

Besides Kinect reliability issues, the servos would occasionally overheat (likely due to forcing them into invalid positions via poor Kinect data), causing the offending servo itself and the entire controller board to stop responding. Additionally, several nylon servo gears were stripped and needed to be replaced. To help give the user more control over the robot arm, especially if they were not adjacent to the computer due to standing in front of the Kinect, hand gestures were added (to be detected via the CyberGlove module) to turn the robot arm on and off.

Currently, all code has been documented per official Microsoft Documentation standards for in-line code documentation. The Kinect, CyberGlove, and robot arm components are interacting reliably, and the robot arm can indeed mimic basic arm motion – albeit lagged -- to the point of picking up and putting down small objects, such as pens. Next semester, the team hopes to reduce the latency between input and output, improve motion control with the IMUs, and provide a more elegant user interface.

Introduction

As robotic industry is playing important factor in individual’s lives, team Sieben decided to choose a topic that can be a revolutionary experiment in the robotics industry. The project can be described as controlling a robotic arm to mimic the motion of arm and hand movements. The primary goal of the team was to develop a highly modular and extensive software to not only demonstrate the novelty of our system in the near term, but to also expand on it in the future into a system with non-trivial implications for society at large.

In this document, the contents are categorized in specific order to share and explain the details and results of this project over the past semester, and the project plans for the next semester. The document is organized as follows: initially, the report discusses the scope and motivation of the project and identifies the potential users; then it identifies the scope, project plan, and development approach followed by a brief overview of design decisions and the technology used; lastly, this paper discusses the problems and risks that the team encountered throughout the semester, as well as the overall results thus far, lessons learned, and future development plans.

Project Description

The robot arm project can be defined as developing and enabling an environment to control a robotic arm to mimic human arm and hand movements. This project is being developed in the Center for Cognitive Ubiquitous Computing (CUbiC) at Arizona State University, headed by Dr. John A. Black, Jr. The primary goal of this project is to use a CyberGlove motion-tracking glove along with accelerometers or orientation sensors to allow a user to control a Lynx Motion AL5D robot arm remotely through natural arm movements. These sensors include using a Microsoft Kinect and motion trackers to monitor the movement of the human arm and transfer the desired data to the designated control software for calculations and operations.

In addition, one of the desired goals of this project was to achieve all of the designated tasks in real time. In other words, the goal is to create a system which drives a robotic arm to mimic human arm movement as it occurs. The goal for the first semester was to perform the basic movement of the robotic arm and disregard the complex movements and calculations.

Project Motivations

In today’s world the robotic industry is playing an important role in individuals' lives. The primary motivation of this project is essentially built around the benefits and contributions of the robotics industry. In other words, the product that has been developed for this project and the associated technology have a variety of beneficial real-world applications. The ability to accurately read human arm and hand movement and translate that data into a form that allows a machine to mimic that motion requires complex calculations. Exploring these techniques has significant potential to contribute to the robotics industry by enabling the production of more efficient and user-friendly human-robot interfaces. This is one of the main motivating factors behind this project.

Some of the real-word applications for this technology include, but are not limited to: bomb disposal robots; potential robotic replacement arms for people with paralysis or other nervous system disabilities; various industrial and manufacturing applications, such as the automobile manufacturing plants. Furthermore, in the healthcare industry the ability to enable people with paralysis to reach and grasp objects using computer interfaces has tremendous potential to improve their quality of life. Another potential user group would be workers who are asked to perform tasks in hazardous environments or situations in which replacing substituting robotic arms for human ones can significantly reduce the risk of serious injury or death. Therefore, researching and developing an intuitive, effective interface for a human to control a robotic arm will be beneficial to many industries and individuals, and is the primary motivation for this project.

User Identification

This section will identify potential users and beneficiaries of this project and related technology. For instance, in military or law enforcement fields, a bomb disposal team could benefit from remote robotic arm control. They are regularly in hazardous situations in which they must defuse live bombs in public areas, and take on the tremendous risk of having the bomb explode while they are attempting to defuse it. One of the potential goals for following semester will be creating and developing a control interface that will allow the robot arm to be operated remotely over arbitrarily large distances.

Workers in certain fields of manufacturing who regularly deal with dangerous chemicals could benefit from technology similar to that developed in this project. Many workers must handle chemicals and other materials which would cause extreme injury if they came into contact with human skin. The robot arm would allow these workers to transport hazardous materials without the risk of physical harm.

Finally, the robot arm could also be used for entertainment or educational purposes by children and teenagers. For instance, the robot arm could be placed in a crane game which is commonly found in arcades, with the crane being replaced by the robot arm. This would create a game that requires significantly more user interaction than current crane games which use a joystick to move the crane. The robot arm could have educational value by sparking young students' interest in computer programming, engineering, and robotics.

Body

Scope

The original scope of the project was to create and develop an environment to mimic human arm movement. The primary goal was to develop an intuitive, user friendly interface to remotely control the motion of a robotic arm. In order to achieve such a complex task, the original scope was focused on developing a fundamental interface to perform basic tasks such as lifting small objects by using data received from the Kinect and CyberGlove to control the movement of the robot arm servos. The initial scope also included the use of a webcam to allow a user to control the robot arm remotely (theoretically, from anywhere with an internet connection).

As project development progressed over the semester, the project scope was refined slightly. During the testing phase of the project it became clear that some of the hardware requirements needed to be changed. For instance, the Microsoft Kinect and its inaccurate motion tracking proved problematic. As a result, future iterations of the project will use ArduIMU motion sensors to track the motion of the user's arm, and the Kinect will be phased out entirely. The team also added gesture control to the interface, which allows the user to engage or disengage control of the arm by performing certain hand gestures with the CyberGlove. The scope has also been expanded to include plans for a graphical user interface to provide for easier user configuration and calibration of the robot arm.

Project Plan

The team developed a project plan at the beginning of the semester as for how the team would develop the final product. The schedule for fall 2012 semester is shown in Figure 1 and a list of dates and deliverables is show in Table 1.

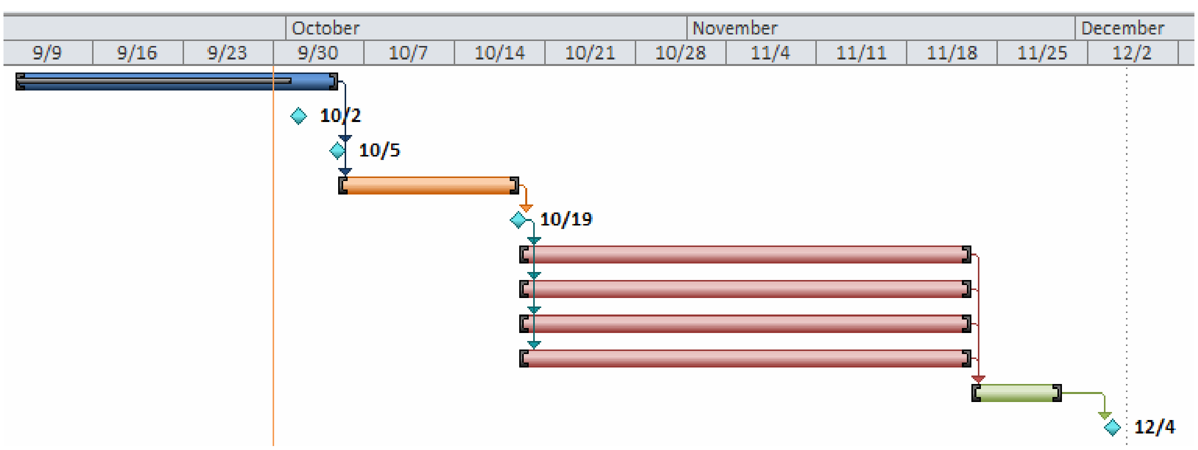

Figure 1: Schedule for fall 2012 Semester

Table 1: fall 2012 semester deliverables

Fall 2012 Semester			
	Scheduled Completion	Completed	Deliverables
Project Initiation	Oct 4	Oct 4	Initiation Documents
Design	Oct 18	Oct 28	Team/Duty Assignments Module Interface Specification
Development	Nov 22	Nov 29	CyberGlove Module Kinect Module Arm Control Module Integration and Control System
Integration	Dec 10	Dec 7	Generic Motion using all modules
Refinement	Spring 2013		

The schedule above shows the products that were delivered at the different stages for fall 2012 semester. As the team determined a scope change was needed for our project based on the quality of data that was being produced by the Kinect, spring 2013 semester plan was modified from the original plans as well.

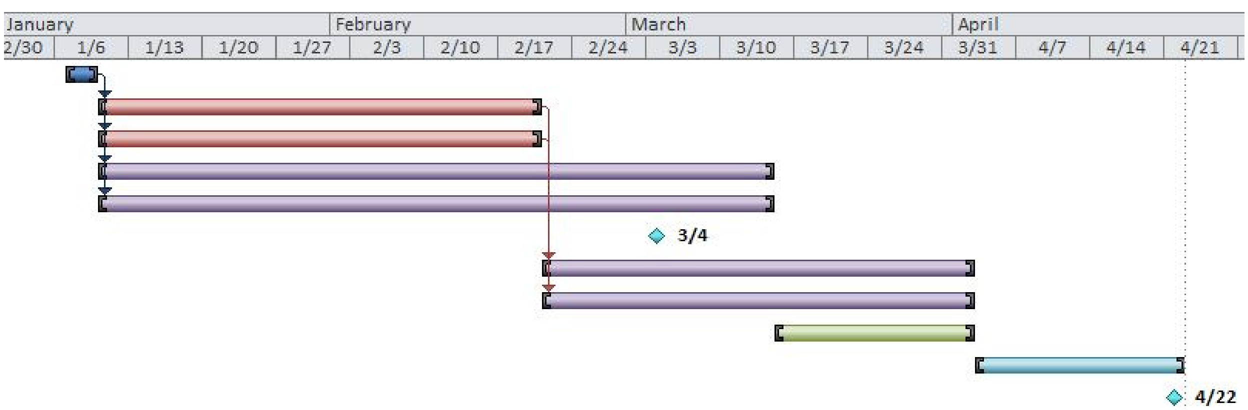

Figure 2: Schedule for spring 2013 Semester

Table 2: spring 2013 semester deliverables

Spring 2013 Semester		
	Scheduled Completion	Deliverables
Project Review	Jan 9th	Team assignments
Development	Feb 20th	IMU module Video Feedback module
Deliverable 2	March 4th	Video feedback partially operational IMUs functional (not calibrated) New components not integrated
Refinement	April 2nd	Calibrate all modules
Integration	April 2nd	Combine all modules while refining motion
Testing	April 22nd	Find all possible errors
Deliverable 3	April 22nd	All components Integrated All components calibrated Documentation

The schedule for spring 2013 semester is shown in Figure 2 and a list of dates and deliverables is show in Table 2. The original deliverables for semester two consisted of refining the work from semester one. With the addition of the new modules our deliverables for the team itself are as shown in the table. For deliverable 2 the team will be able to show a partially operational video feedback module. This module will not be complete in its integration. The IMU module will be able to produce data, but most likely will not be fully calibrated.

Refinement will take place on the existing two modules, the Arm Control module and the CyberGlove module, while the new modules are being developed. The final deliverable, 3, will have all four modules fully integrated and calibrated to function as a single working product. To recap the modules are the IMU module, Arm Control module, CyberGlove module and the video feedback module. To be a single working product the robot arm should move is a similar motion to the users arm.

Development Approach

The approach the team took for this project was to split the main team into sub-teams. Associated with each team was a single module for this first semester. The iterative development model allowed the team to develop these modules in parallel. Each sub-team had the ability to develop requirements, Analyze and implement, and test the functionality of their specific product prior to integration.

Table 3: Sub-teams

Module Name	Team Members
Kinect Module	Joseph Babb and Scott Baker
CyberGlove Module	Michael Astrauskas and Craig Cassinat
Arm Control Module	Ashoraya Mezdo and Max Shirvanifar

As will be discussed, the second semester will involve additional modules being added to give more functionality to our project. The three modules developed this first semester were the Robot Arm module, the Kinect module and the CyberGlove module. As stated, all modules were developed in parallel. The teams were closely involved with each other to minimize problems that could have become present at integration.

The team had different types of meetings throughout the semester. Each sub-team had meetings to work on their respective tasks. The entire team also met to coordinate the goals that each team should be accomplishing between meetings.

There is also a fourth component to this modular design that was developed by the team to help combine the modules. This component is called the Integration and Control System. This system allows each team to remain fairly separate from one another. The Integration and Control System will allow the team to replace modules as needed. As will be discussed in more depth later, the Kinect will be replaced by another motion tracking system. Since this system is able to integrate a number of components (modules) replacing the Kinect will be much easier compared to different design.

Notable Design Decisions

The initially proposed scope of the project has provided a number of different possibilities for the design and implementation of the Natural Motion Robotics Controller. The initial proposal did not specify requirements on system flexibility, language and system compatibility, the full extent of user input and its collection methods, and feedback provided to the user. Due to this, the system architecture has been developed based on a number of decisions made regarding these factors. Many of the notable decisions are discussed below.

- The system architecture should emphasize flexibility and extensibility. In order to accomplish this, the system has been designed to use a highly modular publish / subscribe architecture. This will not only simplify subsequent stages of the project, but will also provide a method in order to develop and debug each piece of functionality separately in order to better utilize development resources.
- The system will be targeted primarily towards the Windows operating system and will be implemented in C#. This will provide us with a well-developed API for building a graphical user interface and will provide us with additional SDKs for several of the peripherals of the peripherals we will be working with.

- The system will not only take input from the CyberGlove peripheral device, as specified by the project proposal, but will ultimately use an additional sensor to track the state of the users arm as well as a GUI in order to collect additional user input. Initially, the Kinect was to be used to track the user's arm, which will allow us to rapidly develop an initial prototypical system. The Kinect's performance was to be evaluated in order to determine if the functionality provided was adequate.

- Due to limitations of the Kinect (specifically, limited data accuracy), the Kinect will be replaced by the use of IMUs in the near future. These IMUs will be attached to the rigid portions of the user's arm and hand in order to track their orientation.

- Finally, the system will ultimately provide the user with augmented graphical feedback, including information regarding the state of the system and its peripherals, the inputs being acquired by peripherals, and a remote video feed displaying the real-world position of the robotic arm as it moves to match the user's position.

Overview of System Architecture

The architecture of the Natural Motion Robotics Controller is designed in order to maximize extensibility and versatility for future applications. In order to support this, the system uses a fully modular publish / subscribe architecture. In this architecture, the functionality of the system is split into atomic components, each of which is encapsulated in a single module. The modules are almost completely autonomous and their interactions are controlled through a managed bus interface. A complete view of the system can be seen in Figure 3.

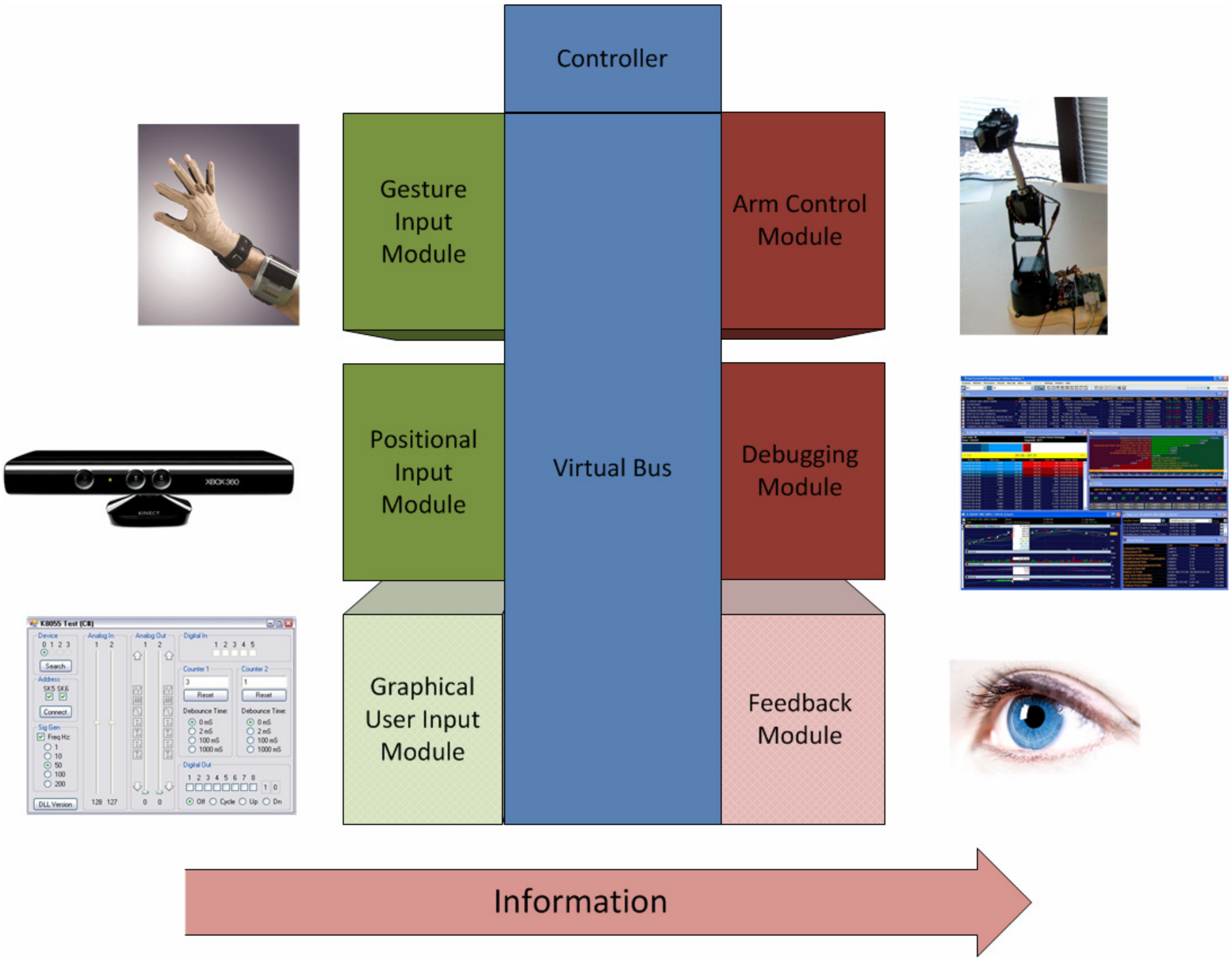

Figure 3: Complete System Architecture [5] [6] [7] [8] [9]

Components displayed in blue are responsible for the management of the individual modules and their interactions. These components are the top-level controller and managed virtual bus. The top level controller is responsible for managing the life-cycle of each of the individual modules, namely initializing them as needed and finalizing them as they become obsolete. Meanwhile, the primary purpose of the virtual bus is to facilitate safe communication between modules, insulating each module from the implementation of each other module. An example of this insulation is ensuring that all interactions between modules are completely thread safe.

The components displayed in green are the data producing modules. Their job is to poll various input sources, such as the Kinect, CyberGlove, or IMUs, and preprocess the data received into an expected form. They then publish the resulting data, which includes information like joint rotations and player position, to the virtual bus for use elsewhere in the program. In particular, the purpose of each module is as follows:

- The Gesture Input Module (GIM) is responsible for tracking the position of the user's fingers, hand, and wrist. This is accomplished by polling the CyberGlove (currently done about 10 times a second), which is worn by the user. The CyberGlove then provides flexion and orientation information about many of the joints in the hand, which are then used to obtain a number of useful positional and gestural metrics used to control the system.

- The Positional Input Module (PIM) is responsible for tracking the users position and the orientation of the user's upper and lower arm. Currently, this is done by interfacing with a Microsoft Kinect, which provides rough skeletal tracking data that is processed to obtain the three dimensional relative orientation of each joint in the users upper body.

- [Implementation Pending] The Graphical User Input Module (GUIM) is a future module that will allow the user to configure the behavior of the system via a graphical user interface. Possible configuration options include: the COM ports used to communicate with the peripherals, initial arm position, arm movement speed, debugging mode toggle, and module management.

The components displayed in maroon (and grey) on the right side the diagram are data presentation and processing modules. These modules are primarily responsible for reacting to the input of the data producing modules in order to present the resulting information to the user by various means. Among them, the Arm Control Module and Debugging Module have been implemented to date, while the Feedback Module still remains for next semester. The purpose of each module is as follows:

- The Arm Control Module (ACM) is the primary output module. It is responsible for manipulating the physical position and orientation of the arm and its components in order to mimic the position of the user, as observed by the input modules. This is done primarily by listening for updates along the virtual bus, performing additional processing on the values in order to better model the capabilities of robotic arm, and communicating with an embedded servo controller on the arm itself to position each servo.

- The Debugging Module is mean to facilitate debugging one or more modules by providing an internal view of the communication between each module. This is done by quietly listening to all data being published on the bus and updating a view

- [Implementation Pending] The Feedback Module (FM) is meant to provide the user with additional visual feedback from the system, this can include indications as to the state of the system, such as whether it is currently tracking the user, pre-processed input reflection, such as how feedback showing how the system sees the user's current position / orientation, and remote visual information on the arm, such as a direct video feed of the robot arm.

Technology and Tools Used

The project goal is to develop a closed-loop system where a user can intuitively control the robot arm while getting some level of feedback. The robot arm used was a LynxMotion AL5D [4] (Figure 4) arm with 5 degrees of freedom (DoF), allowing it to roughly imitate human arm motion.

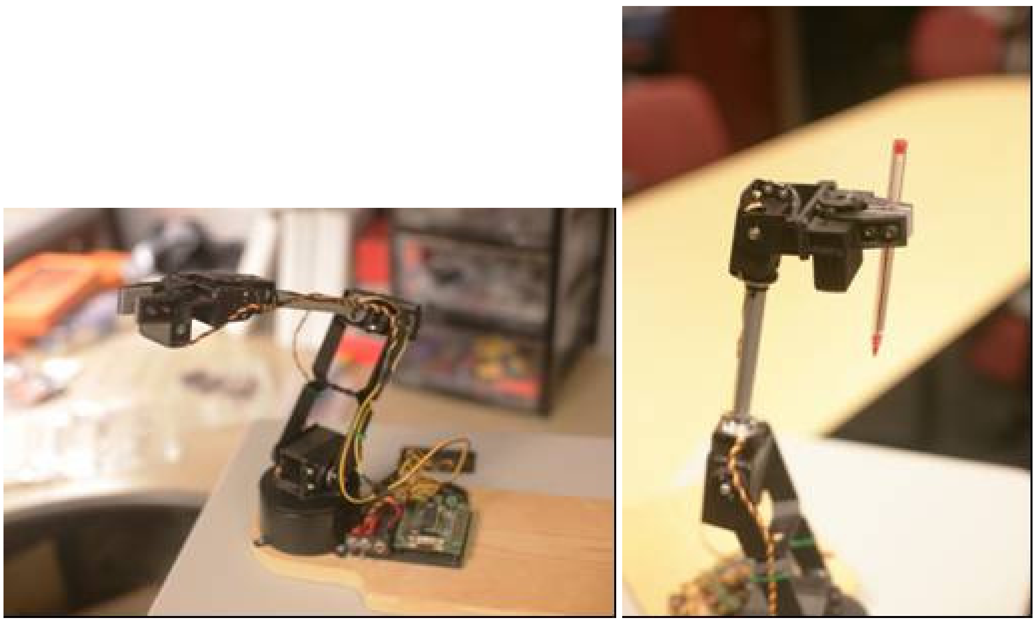

Figure 4: The LynxMotion AL5D robot arm

Two means of input were used to control the robot arm. First is a CyberGlove [1] (Figure 5) motion-tracking glove to control the gripper and wrist of the robot arm. This is a cloth glove with 18 or 22 flex sensors (both models are available to the team) that return information such as angle of individual finger joints, finger abduction (angle between fingers), and wrist flexion/extension (wrist up and down movement).

While a CyberGlove API is available, the team opted to read raw data streamed from the CyberGlove's serial interface. This removed any OS dependency, as the API is only available for Windows, and removed a layer of abstraction that may have introduced undesired latency into this real-time project. Figure 6 shows some example communication with the CyberGlove, including raw data from the 18 flex sensors (toward the bottom).

Figure 5: A wired 18-sensor CyberGlove

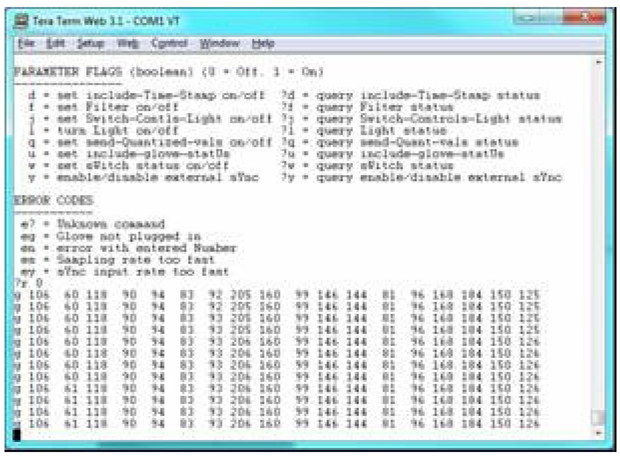

Figure 6: A terminal showing plaintext communication with the CyberGlove

The second input method used a Microsoft Kinect [3] (Figure 7) to track the user's body. The Kinect, in short, is a series of cameras that can provide a 22-point 3D skeleton of a user standing in front of it, allowing ease in finding the joints of the user's arm.

Figure 7: A Microsoft Kinect

For software development, the team used Microsoft Visual Studio 2012. Source code was stored and exchanged in a git repository hosted by github, and Google Drive was used for sharing other documents, such as scheduled, reports, and slideshows.

An input method planned for this semester but ultimately held off for next was the use of inertial measurement units (IMUs) to sense arm movement and orientation. An IMU contains a 3-axis accelerometer, a magnetometer (to account for accelerometer drift), and a 3-axis gyroscope to measure orientation. The model in mind is the ArduIMU+ V3 [2]. The team has one and is currently testing its precision. One IMU would be placed on each rigid part of the arm, meaning the hand, forearm, and upper arm.

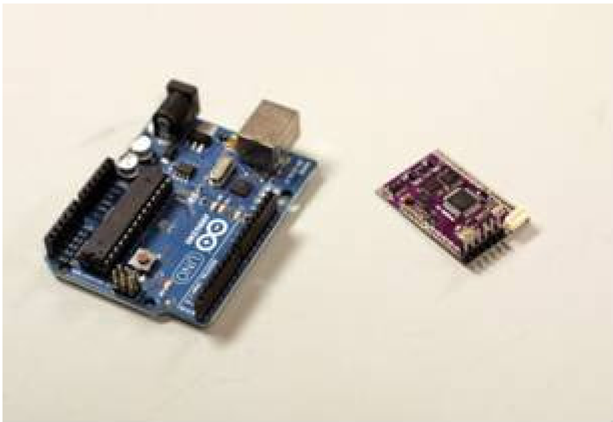

Figure 8: An ArduIMU+ V3 and Arduino Uno microcontroller

The IMUs transmit data over a serial UART, so the team will either connect all 3 to a PC with serial-to-USB adapters or route them through a USB Arduino Uno, which can also offload some of the processing. Dr. Troy McDaniel of CUbiC is planning a spring 2013 study involving the use of IMUs in tracking of arm motion and has expressed an interest in collaborating with the team. A prototype sleeve of Dr. McDaniel's with 3 ArduIMU+ V3s can be seen in Figure 9.

Figure 9: A sleeve with 3 ArduIMUs

Possibly in parallel with the integration of the IMUs into the project the team plans to set up a remote viewing environment so that the user of the robot arm will have to operate it remotely via a video screen. Sound may or may not be included, but the user will no longer be able to move around to get a more complete perspective of the robot arm’s environment.

Figure 10: A simulation of the remote video environment

Preliminary Results

The current implementation of the Natural Motion Robotics Controller allows a user limited control of the robot arm. Specifically, the robot arm will roughly mimic the movement of the user's upper arm and forearm by utilizing the Kinect's skeleton-tracking feature and the Positional Input Module (PIM), and will fairly accurately track the pitching movements of the user's wrist through the use of the CyberGlove and the Gesture Input Module (GIM). The CyberGlove and GIM also provides the user with the ability to control the pincers of the robot arm by bringing the thumb and forefinger together or apart. All of these tracking motions experience significant latency between the user's motion and the arm's response (roughly 250-500ms delay).

The Kinect has proven to be incapable of providing accurate and consistent tracking of the user's arm, which has made it difficult to fine-tune the calibration of the Arm Controller Module (ACM), and makes it challenging for the user to precisely control the motion of the robot arm. However, the CyberGlove has proven very capable of providing precise, reliable tracking of the user's wrist and hand motions. The result is that if a user can manage to move the robot arm into the desired position, it is fairly easy to then pitch the wrist joint up and down and pick up or release a small, lightweight object. Figure 11 shows such a test performed by the team.

The team has also implemented gesture control which allows the user to engage or disengage control of the arm using the CyberGlove. The user does this by moving the middle and ring fingers (3rd and 4th fingers, if the thumb is considered finger 1) down to touch the base of the palm, while keeping all other fingers extended. This works as a toggle switch, so if the user performs this gesture while motion tracking is active, the GIM detects this gesture and signals the ACM to stop sending movement commands to the robot arm, resulting in the arm holding its current position regardless of any other movements of the users arm. If the user repeats the gesture, the ACM will resume sending commands to the robot arm.

Figure 11: Test operation of the robot arm with gesture control implementation.

Problems and Risks

During development over the course of the semester, the team encountered some problems, some which were expected and others which were not, and recognized new sources of potential risk that could hinder future development.

First, the Kinect was found to be much less precise in trcking the user's arm movements than anticipated. It requires the user to stand in one place and hold the right arm within a relatively narrow range (approximately 4 to 11 ft within a 57.5 degree arc in front of the Kinect) in order for the Kinect to track any motion at all. Furthermore, the Kinect has proven to be largely incapable of tracking the full range of pitch and yaw of the user's wrist, and completely incapable of tracking any rolling motion of the wrist/forearm. While this rolling motion is not currently required due to the current robot arm hardware, the planned addition of an extra servo to the robot arm to provide a rolling wrist capability will require accurate tracking of the full range of motion of the user's wrist.

To better track wrist movements, the team decided to use the Cyberglove, rather than the Kinect, to track the motion of the wrist. Furthermore, it was decided that future implementations should rely on the ArduIMU sensors to track the motion of the user's upper arm and forearm. This means that development of future iterations of the product will drop the Kinect entirely (assuming the team is able to acquire the sensors).

Another major problem encountered during development, and one that poses a continuing source of risk for future development, is that the servos are vulnerable to overheating and failing if they attempt to move against a physical impediment. This occurred during our testing when the servo controlling the robot's upper arm was attempting to move the arm lower than the base would allow. This caused that servo to overheat and eventually shutdown, rendering the entire system inoperable, and also lead to stripping of the plastic horns which attack the arm to the servo.

The stripping issue has been solved by replacing the plastic horns with metal ones, but the risk of overheating the servos by sending them bad commands remains. A servo failure could significantly setback the development schedule, and if this failure occurred immediately before a scheduled demonstration of the hardware, the consequences could be dire. As a precautionary measure, the team will try to ensure that extra servos are available to replace any that fail. At the software level, bounds have been set redundantly to try to prevent sending any servo an "out-of-bounds" movement command. Furthermore, now that the team is aware of this issue, all members will be more vigilant while operating the robot arm and more likely to notice if the servos are straining against excessive resistance.

Lastly, a continuing risk stems from the uncertainty regarding the team's budget and ability to acquire additional hardware (such as the ArduIMU sensors and an additional wrist joint/servo for the arm) which will be required to further refine the user interface and robot arm functionality. If the team is unable to acquire these items, it will be difficult to deliver a product with the desired functionality next semester.

Summary of Tasks Completed by Each Team Member

Scott Baker

- Explored the Microsoft Kinect hardware features and capabilities to determine appropriate use for the project
- Explored the Microsoft Kinect API and software library to help implement the gesture input module
- Assisted with debugging and testing of the Positional Input Module
- Aided in general group testing and debugging of the integrated control interface

Max Shirvanifar

- Developed the initial source code to move all servos

- Developed the serial bus configuration to communicate with robot arm
- Defined all the allowed ranges for each servo and developed all the safety checkpoints
- Developed subscription method to get data from both Kinect and cyber glove and helped to develop proper transition functionality in math and calculations.

Craig Cassinat
- Developed the CyberGlove Module with Michael
- Developed Gesture inputs to help control the robot arm
- Assisted in integration and calibration of the CyberGlove and Arm Control Module
- Disassembled and reassembled the robot arm moving servos to lessen wear and strain on a faulty servo

Joseph Babb
- Designed system architecture
- Created interface specification for modules
- Developed the integration and control systems (controller and virtual bus)
- Developed Positional Input Module with Scott
- Developed Debugging Module to test integration and control systems
- Assisted with final integration of robotic arm module
- Spearheaded final calibration efforts

Ashoraya Mezdo
- Refined the initial source code developed by Max Shirvanifar to gain an understanding of the operational range of each servo
- Developed code to test how well the robot arm handles multiple servos moving at once
- Helped develop the code which controls the movement of the robot arm
- Helped develop code to reduce the passing of redundant data to the robot arm from other modules
- Tested and helped diagnose the cause of the robot arm freezing up after a few seconds of operation
- Aided in general group testing and debugging of the control interface

Michael Astrauskas
- Co-wrote project proposal
- Aided robot arm team with getting started
- Ordering of replacement robot arm parts and some robot arm repair
- Half of CyberGlove code and calibration
- Set up team workspace (software and hardware)
- Researched IMUs

Conclusion

Success of the project so far

The team has had a large variety of successes this semester. Firstly, all sub-teams have successfully completed and integrated their modules. This includes the Kinect, CyberGlove, and Arm Control modules. The entire team worked together extensively to iron out as many known bugs as possible from the integrated modules.

Secondly, all of the existing code has been well documented so as to remove as many ambiguity in the code. A developer's manual has been created which details all of the functions that are used in the program. This will allow the team, and any future team which may continue working on this project, to quickly understand the code base without delay.

Finally, and most importantly, the robot arm is capable of performing basic arm movements. For example, the robot arm is capable of performing tasks such as picking up a pen and moving around. Thus, the project has been largely successful this semester with the team reaching its main goal of basic arm movement along with other accomplishments such as code documentation.

Lessons learned

Throughout the semester, the team has learned a number of valuable lessons. One of the biggest lessons learned is that extreme caution must be practiced when testing the robot arm with the Kinect. There was a situation where Joe was testing the robot arm with another member standing behind him. Instead of the Kinect locking onto Joe, the Kinect instead locked onto the person standing behind him and began feeding invalid data to the robot arm. After receiving this invalid data, the robot arm began to move around erratically and stripped one of its horns. From now on, whoever is testing the robot arm will make sure that the only person who remains in the field of vision of the Kinect is the person who is actually performing the test.

Another lesson that was learned was that the team must be ready to replace parts on the robot arm at any time. This is because the robot arm is simply not as robust as was originally assumed. Numerous horns have already been stripped unintentionally, and the delays due to shipping time of new parts can be disastrous when deadlines must be met. Extra horns and perhaps even servos will be kept aside just in case more failures happen. Furthermore, the robot arm has been stripped down numerous times and the group has a better foundational understanding of the mechanical workings of the robot arm. This will allow the team to quickly disassemble the robot arm and replace broken parts with replacements that have been kept in storage.

Finally, the team has learned that the Kinect is simply too unreliable for the purpose of performing fluid human arm motion with the robot arm. Earlier in the semester, it was assumed that the Kinect would provide reliable data; however, after testing the data that was provided by the Kinect, it was realized that the robot arm was behaving very erratically. It has proven to be very difficult to predict when the Kinect will fall short of the expectation of consistent data, thus making it difficult to work around this issue when the Kinect is being used. In conclusion, there have been numerous lessons learned which will help the team and future teams more effectively work on the robot arm.

Future work

There is much future work to be completed in order to be able to produce a fully functional final product. In the future, a visual feedback module will be added. This will provide a user with remote viewing capabilities, which will allow the user to control the robot arm without necessarily being within close proximity of the robot arm. The user will be able to view the robot arm through the use of a camera which will transmit a single point of view to a designated screen. This addition would be largely beneficial for future applications that involve the handling of hazardous materials.

Another part of the project which must be completed is the graphical interface module. This module will allow developers of the robot arm to have an easier configuring and calibrating the robot arm. As it stands now, calibration and configuration must be done by digging through the code and changing the appropriate values to adjust factors such as the range of motion of each servo. By developing a graphical interface, developers will be able to change a wide variety of settings while the robot arm is still moving. Currently, the software controlling the robot arm and the robot arm itself must be restarted every time an adjustment in calibration and configuration is made.

Additionally, the Kinect must be phased out in favor of IMUs which will provide data that is far more reliable. These IMUs will be able to detect a user's arm movement and provide the arm control module with data that is as reliable as other data provided by the CyberGlove. This new source of data will hopefully be sufficient enough to reach the goal of performing movements with the robot arm that very closely matches the movements of the user who is controlling the robot arm.

Finally, additional servos must be installed on the robot arm to add more degrees of freedom. While the robot arm is currently capable of performing movement which roughly mimics the motion of a human arm, it is still lacking in a fundamental area. Presently, the robot arm cannot perform any form of wrist rotation, which is a key aspect of many tasks that humans can do with their arms. For example, it would be exceedingly difficult for the robot arm to carry out tasks such as rotating an object. In the future, a wrist servo will be added to move past this issue, and give users of the robot arm a more natural experience when controlling the robot arm. Therefore, all of this additional work will be completed in the future to produce a robot arm that features many more capabilities and functionalities over the current product.

Citations

[1] "Cyberglovesystems," CyberGlove Systems LLC, 2010. [Online]. Available: http://www.cyberglovesystems.com/. [Accessed 30 09 2010].

[2] "ArduIMU+ V3," 3D Robotics Inc., 2012. [Online]. Available: http://store.diydrones.com/ArduIMU_V3_p/kt-arduimu-30.htm. [Accessed 30 09 2012].

[3] "Develop for Kinect | Microsoft Kinect for Windows," Microsoft, 2012. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/. [Accessed 30 09 2012].

[4] "Lynxmotion - AL5D," Lynxmotion, [Online]. Available: http://www.lynxmotion.com/c-130-al5d.aspx. [Accessed 30 09 2012].

[5] [Online]. Available: http://www.realsimple.com/health/preventative-health/eye-health-101-00000000015938/index.html.

[6] [Online]. Available: http://www.interactivedata.com/index.php/Contents/show/content/primeterminal-pro.

[7] [Online]. Available: http://wiki.squeak.org/squeak/5907.

[8] [Online]. Available: http://www.indiamart.com/cad-engineering/products.html.

[9] [Online]. Available: http://gdgt.com/microsoft/kinect/.