Worcester Polytechnic Institute

Major Qualifying Project Proposal

MIRA: Modular Interchangeable Robotic Arm

Submitted on

November 15, 2017

Submitted by

Chris O'Shea, RBE
Alex Taglieri, RBE/CS
Ben Titus, RBE/ECE

Contents

1 Introduction

The goal of this project is to create a adaptable control system whose purpose is to drive a variety of robotic arms. To prove the usefulness of our system, we plan on taking an existing arm from the RBE 3001 course and retrofit it with our control system. In addition to this, we will modify the existing arm to include an interchangeable link on the end of our arm, controlled by our system. To interface with our system, we will create a GUI for easy configuration and basic control of the arm. The base will communicate with a computer running control code either through the software application or code library.

The existing RBE 3001 arm currently has its own central control system. We aim to remove this and replace it with our own distributed one. To accomplish this, we will design a motor control board that communicates with a base controller. Additionally, we will design and implement a new link on the arm. This new link will be interchangeable and implement our control board.

We will create a software application to interface with a constructed arm. The user will input how they have constructed their arm into this application and then be able to do some simple control. Another feature of this application will be the ability to record a series of poses for the arm to perform. In addition to this software, we will also create some programming libraries to allow users to control the arm with an actual program.

In this document, we will outline some existing robot arms and highlight the differences between these arms and our arm kit. We then discuss what work there is to be done on this project. After discussing the work to be done, we will state how this work will satisfy the capstone design requirements for each of the three disciplines represented by our group members. Then, we state the constraints we expect going forward with this project. Next, the acceptance criteria for any deliverables at the end of this project will be outlined. Finally, we will state an estimated timentire line for this project.

2 Background

This section begins with an overview of some existing robotic arms that are about the same size as a fully assembled arm from our kit will be. Next we discuss some existing modular robotic arms. Finally, we highlight how our kit will be different from the discussed prior art.

2.1 Robot Arms Currently in Use

There are a plethora of industrial robotic arms. Since our kit will be relatively small compared to most industrial arms, we will begin with an overview of existing desktop industrial arms. Arms that fit this description have a reach of less than 1000mm. Industrial robot arms typically cost between $50,000 and $80,000 new and $25,000 and $40,000 used [3]. Some manufacturers of these industrial arms include ABB Robotics, Universal Robots, and KUKA Robotics.

2.1.1 ABB Robotics

ABB Robotics makes many small industrial arms. The ABB IRB 120 boasts a 580mm reach, 3kg payload, and 25kg weight. It has 6 degrees of freedom and can be mounted at any angle. The ABB IRB 1200 comes in two varieties, one with a reach of 703mm reach and 7kg payload, and one with a 901mm reach and 5kg payload. Both of these arms have 6 degrees of freedom. The weights are similar at 52kg and 54kg respectively [3].

2.1.2 Universal Robots

Universal Robots makes two robot arms in this size range. The UR3 is the smaller of the two with a reach of 500mm, payload of 3kg, and 11kg weight. A step up is the UR5 which has a 850mm reach, 5kg payload, and 18.1kg weight. Both of these arms have 6 degrees of freedom. Universal boasts that these arms are easy to implement and re-implement due to compact and lightweight construction, and simple programming interface [3].

2.1.3 KUKA AG

KUKA makes two robot arms in this size range. The KR3 R540 has a reach of 541mm, payload of 3kg, and weight of 26kg. It can be mounted on the floor, wall, or ceiling for added utility. The K5 sixx R650 is larger with a reach of 650mm, payload of 5kg, and weight of 127kg. It can only be mounted on the floor or ceiling. Both of these arms have 6 degrees of freedom [3].

2.2 Small Industrial Robotic Arm Comparison

Table 1 shows a comparison of all the robotic arms discussed in this section.

Name	Reach (mm)	Payload (kg)	Weight (kg)	Axes
IRB120	580	3	25	6
IRB1200-7/0.7	703	7	52	6
IRB1200-5/0.9	901	5	54	6
UR3	500	3	11	6
UR5	850	5	18.1	6
KR3 R540	541	3	26	6
K5 sixx R650	650	5	127	6

Table 1: Comparison of <1000mm reach industrial robot arms

2.3 Modular Arms

While there are many industrial arms in production, there are very few modular robotic arms. There is one commercially available robotic arm, the Robolink, made by igus. A few modular robot arms have been developed, including the reconfigurable modular manipulator (RMM), made by TRACLabs [1], and a single joint for the Modular Robotic Arm project MQP at WPI [4].

2.3.1 igus Robolink

Robolink is a modular robotic arm kit produced by the plastics manufacturing company igus. The kit contains parts to make an arm that is up to 6 Degrees of Freedom (DOF), with belt driven linkages powered by stepper motors that reside in the base of the robot. Robolink offers 7 individual links, ranging from 1-2 DOF and differing based upon their kind of motion (pivoting, rotating, swiveling). Each link is made of a lightweight and strong plastic or carbon fiber with cables inlaid in them, resulting in a low cost and weight arm. The cables used to control these links are made of a high strength synthetic fiber with has a tensile strength of 4,000N. Separating the actuation of each link from the joint allows the arms to be easily maneuverable with its lightweight and strong joints.

Purchasers of the kit are able to combine the links in different ways, allowing for a flexible, modular solution to robotic arms. Igus also offers their Robolink software for programming articulated arms that facilitates the programming of individual arms through the use of a simple, intuitive control software. The total cost of a kit to make a 6 DOF arm is $6000, and buying individual links will cost anywhere from $370 to $750 per link. While this price may be low cost compared to other arms such as the ABB robotic arm which can cost up to $200,000 in total, it is still not low enough for either hobbyists or people interested in learning about robotic arms but prevented from doing so by the high entry cost. In addition to this, the belt system actuating each link requires the user to thread belts attached to the actuators to each link in order to set up the robot. The long assembly time and intricacy also detracts from the idea of a modularity because the time involved in switching configurations can inhibit users from really exploring the different workspaces and combinations this kit can create [2].

2.3.2 Reconfigurable Modular Manipulator

The reconfigurable modular manipulator developed by TRACLabs for NASA is a fully modular 7-DOF robot arm. Each joint and end effector have the same connector that provides power and control lines throughout the arm. Internal power and control circuitry take in these lines and convert them into movement. Joints can be swapped out by hand in a matter of seconds. Joints accept position or velocity data from the central communication lines and store physical characteristics about the joints in memory. This robot arm is not commercially available [1].

2.3.3 Modular Robotic Arm

This project aimed to close the market gap between inexpensive toy robot arms and expensive professional grade industrial arms. The group aimed to do this by designing a single joint that could be used to assemble a robot arm. Ultimately, a single DOF joint that was heavy, difficult to manufacture, and expensive to produce was designed and constructed. In their future recommendations section, the group stated that the goal of designing a modular robot arm was possible but their design was not the solution [4].

2.4 Our Robotic Arm System

Our modular robotic arm kit aims to offer a completely different experience compared to existing products and projects. The system maintains a low cost while providing a versatile platform for beginner engineers or rapid prototyping professionals. This is achieved by avoiding expensive proprietary software and subtractive manufacturing; favoring off-the-shelf parts, 3D-printed structures, and freely available software. Providing custom-built software for controlling the arm creates a plug-and-play environment suitable for most any skill level.

2.5 Communications

We looked at several types of communications for the purposes of controlling our robotic arm. These include serial UART, SPI, I2C, and CAN.

2.5.1 Controller Area Network

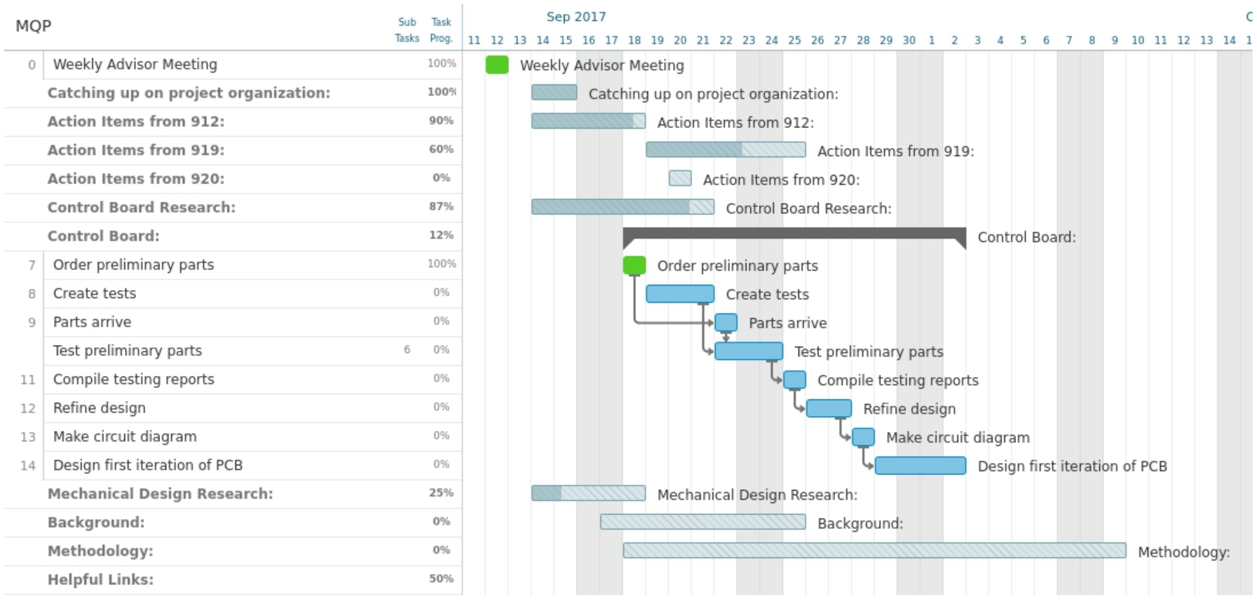
A controller area network (CAN) is a system for sending data reliably between distinct subsystems with reasonably low danger of transmission errors. CAN buses are widely used in the automotive industry to allow various computerized parts of the car to talk to one another.

2.6 Control board

We decided to use a control board that consists of a microcontroller, analog-to-digital converter, quadrature encoder counter, motor driver, current sensor, CAN controller, and CAN transceiver. These devices will allow each control board to control one motor.

Current sensing can be done in multiple ways. The most common way is by using a small resistor in series with the motor and amplifying the voltage across this resistor. This method can cause lots of noise in the system and takes some power from the motor to use. As the value of resistor used increases, so does the power taken from the motor. However, this also causes a larger voltage across the resistor, meaning less amplification is needed and reducing the output noise.

Other methods of current sensing include hall effect sensors. These sensors measure current non-intrusively by using the motor current to induce a small voltage in a conductor, then amplifying that small voltage and outputting it. The insertion loss caused by these sensors is very small. Since these sensors measure current by induction, they can only measure current in a specific frequency band. Most of these frequencies are DC to some upper limit determined by the physical characteristics of the sensor, usually around 100kHz.



	MQP	Sub Tasks	Task Prog.
0	Weekly Advisor Meeting		100%
	Catching up on project organization:		**100%**
	Action Items from 912:		**90%**
	Action Items from 919:		**60%**
	Action Items from 920:		**0%**
	Control Board Research:		**87%**
	Control Board:		**12%**
7	Order preliminary parts		100%
8	Create tests		0%
9	Parts arrive		0%
	Test preliminary parts	6	0%
11	Compile testing reports		0%
12	Refine design		0%
13	Make circuit diagram		0%
14	Design first iteration of PCB		0%
	Mechanical Design Research:		**25%**
	Background:		**0%**
	Methodology:		**0%**
	Helpful Links:		**50%**

Figure 1: Functional block diagram of the system

3 Description of Work

There are many different ways to accomplish the goals we set. We decided to discretize common robot arm configurations into smart joints and their connectors, a base, and end of arm tooling. Each joint will have a motor/servo actuating it and its own control board with various sensors used to control the actuation. We also plan to control the end of arm tooling using another joint control board.

Each joint control board allows joints to connect to the main communications bus running through the system and control the motor/servo on the joint. This board will have all of the necessary components for controlling one motor and communicating to the main controller. The main controller will act as an interface between the computer that is performing all of the complex calculations and the Joints that are controlling their positions.

To exemplify the modularity of our controls system and the interchangeability of our arm we are going to be adding a link onto the existing 3001 arm. By doing this, we will show how our control system can adapt to a variable arm configurations. By taking an already existing arm and modifying it, then applying our control system, we prove that our system can be used to modify most arms that conform to the specs of our system.

We will develop a software application for controlling the arm with a GUI. This software will have a simple control interface for moving the arm, and some configurable settings to act as inputs for the kinematics equations. In addition to this we will develop a code library which will serve as an interface for the end user for use in their own code.

In order to verify that each of these systems is functioning as expected, we will also design a series of tests to perform on them. Tests will be performed and documented for each subsystem and the entire system as a whole.

4 Capstone Design Requirements

Our project will fulfill the Robotics capstone design requirement by touching on all of the major cornerstones of robotic technology. Mechanical Engineering will be fulfilled through designing a new link for the 3001 arm and developing the software required to control that link. We will need to use methods such as DH parameters, inverse and forward kinematics, and motor torque requirements to ensure the mechanical ability of this arm and control it. Dynamic control algorithms will be explored. We will also need to use computer programs such as ROS and Code Composer Studio in order to control the arm. Additionally we will design a software package and create an interactive GUI so that end-users are able to easily interact with the arm. We will need to write lower level embedded code in order to control the electronics on the master controller and each link's microcontroller. With all of the electrical component design in addition to the communications protocols we cover most topics in microelectronic circuit design and implementations. Finally, Systems engineering will be used for the designing the electrical, mechanical and computer science portions of the project. We will need to use systems engineering skills that we have learned through our various robotics projects in order to complete the project.

4.1 Computer Science

In order to properly communicate with the arm, we will need to implement communication protocols. The communications will need to be efficient. The problem of sending information to each of the links of the Robotic Arm will rely heavily on Information Theory.

In addition to communications, we will be developing a set of programs to interface with the arm. The suite will include a library to offload the computationally heavy task of solving kinematic equations to direct the motion of the arm. It will also include an application so the user can interact with the functions inside the mathematics library. This application will have a GUI so the user can specify the arm's configuration. It will also provide the user with some basic control functions, such as jogging the arm's position (either joint-by-joint or linearly with respect to the base coordinate frame) and opening/closing the gripper. Another feature will be the ability to record poses for the arm to go to.

As the arm is designed with modularity in mind, the software will need to be dynamic enough to leave the user with plenty of room for configuration. One of the deliverables of this project will be a set of files that can be included in other people's work so that anybody has access to code that they can use to control the robotic arm.

Finally, we'll need to write firmware that will run on each of the arm's joints, and libraries that people will be able to use to interact with those joints.

4.2 Electrical and Computer Engineering

The design and implementation of each joint's control board involves core topics from Electrical Engineering. The controller board designed for each link must be carefully designed and tested rigorously. Electrical systems design skills will be used to select compatible parts to produce the desired functionality.

Once constructed, the controller board will need embedded code to control the motor. This code will need to be tested and optimized for better performance. Some communications protocols will be used to send data from chip to chip and board to board.

5 Constraints

Budget will likely be the biggest constraint with this project. The stipend given to students by WPI may not cover all of the costs incurred when constructing this robot, and the rest will be paid out of pocket by the student team. Another large constraint will be access to a 3D printer for prototyping. It will be crucial to start prototyping early to accomplish all of the design goals. Time will also be against a major concern, since there are many time consuming aspects to this project.

6 Acceptance Criteria

Acceptance criteria for this project will be broken into 5 major categories: Joints, End effector, Base, Software application, Code library

6.1 Joint Board

- Receive initialization information and joint angles from base
- Moves joint to angles
- Send position updates back to base
- Pass power and signal buses

- Capable of powering logic without powering motors
- Control board is the same for each joint

6.2 Modify RBE3001 Arm

- Remove control system and replace with our own
- Add a link to the existing arm
- Replace currently implemented servo motors with brushed DC motors

6.3 End Effector

- Receives power and signal buses
- Keyed connection
- One input connector
- Terminate CAN bus
- Uses a joint board

6.4 Base

- Sends and receives joint angles to/from Personal Computer (PC)
- Receives initialization information from PC, then sends it to all joints on signal bus
- Outputs power and signal buses
- Converts AC wall power to system power bus
- Power supply and arm on/off switch
- Capable of powering logic without powering motors
- Array of indicator LEDs

6.5 Software Application

- Sends configuration information to the Code Library
- Sends individual joint angles or pose commands to robot through the Code Library
- GUI to adjust current arm configuration parameters
- Record and play back sequence of poses
- Acts as a front-end for code library
- Stretch goal: 3D model of arm moving in real-time

6.6 Code Library

- Receive configuration information from user, selects control constants, sends to base
- Able to control the robot: Receive joint status, send joint angles
- Calculate joint angles using kinematics
- Stretch Goal: Written so that it can interface with multiple languages

7 Testing

7.1 Motor Driver

The DC motor driver selected was the DRV8872. This driver takes in two inputs, in1 and in2, which affect the output much like inputs to an H-bridge.The exception is when both inputs are high. In this case, the inputs are pulled together as a motor break.

To measure the speed of the motor, a servo horn with 6 spokes was attached and a beam break sensor was mounted on the motor with the spoke traveling through the beam. The signal line of the beam break sensor was connected to an Arduino that measured the frequency by incrementing a count in an interrupt triggered on a pin change. The ISR just incremented a count that was printed out and reset every 5 seconds. Some issues arose with this system, however, since there was a small amount of bouncing on the rising and falling edges, leading to multiple readings for each beam break. This was solved by placing a 10nF capacitor from the signal line to ground. Since this number was triggered 12 times per revolution and printed out every 5 seconds, the actual printed value happened to be in revolutions per minute, as shown in 1.

$$rpm = \frac{1rev}{12ticks} * \frac{1}{5} * \frac{60seconds}{1minute} \tag{1}$$

During initial testing, the motor driver was wired up with V_m of 8.4V, logic voltage of 5V, a 10kΩ pull up resistor on nFault, Isen grounded, and the motor outputs connected to a DC motor. During this test, one of the inputs was connected to an Arduino Uno, outputting a constant PWM wave using the `analogWrite()` function with a duty cycle of approximately 50%. The motor turned, but very slowly and with a high pitched whine. When a 100μF capacitor was placed from V_m to ground, the motor spun up to full speed and the whining sound went away.

Further testing revealed a strange behavior when increasing the frequency of the input PWM signal. The motor spun normally at low frequencies of around 500Hz, but at around 1kHz the motor started slowing down and making a whining sound. The problem worsened with increasing frequency. Eventually, this problem was fixed by using a power supply that could output 3A and adding capacitors from in1 and in2 to ground.

Next, in order to determine whether the performance of the motor driver was dependent on input frequency or other factors, the input frequency was increase again from 200Hz to 20kHz. This time, the motor performed much better. The RPM actually increased with an increase in frequency, as can be seen in Table .

7.2 Hall Effect Current Sensor

7.3 DC Motor

7.4 Demultiplexer

The original demultiplexer selected for the joint board (SN74LVC1G19) did not output the correct values to drive the motor driver (DRV8872). The demultiplexer output can be seen in Table and the motor driver inputs can be seen in Table . When the EN pin was pulled high, both outputs would also be driven high. This effect is undesirable since, when given a PWM signal, this would cause the motor driver to turn then brake then turn again as opposed to the desired turn then coast then turn. To solve this problem, a different chip (SN74LVC1G18) was selected. The truth table for this chip can be seen in Table .

7.5 TM4C123

Several peripherals were needed to achieve the desired functionality from our microcontroller. A test board was set up in order to test and verify that each of these peripherals was setup properly and working as expected. The test board consisted of a potentiometer connected to an ADC pin, an SPI controlled ADC (MCP3202), the 1:2 demultiplexer (SN74LVC1G18), CAN transceiver (TC332), and some LEDs.

As a temporary stand in for the AS5055 absolute hall effect encoder to test the SSI peripheral, a MCP3202 12-bit, 2 channel ADC was used. Both devices use SPI to communicate their sensor data back to the MCU, and the packets are similar in structure. Some differences between the two that can be changed are a maximum sample rate for the AS5055 of 1ms as opposed to the few SCLK cycle delays for the MCP3202. The AS5055 has a maximum SCLK frequency of up to 10MHz at 3.3V while the MCP3202 has a limit of 900kHz at 3.3V.

The potentiometer was connected to PB? which was enabled at AIN3. The ADC was set to sample at 1kHz with hardware oversampling 16x enabled.

7.6 Hall Effect Encoder

References

[1] Reconfigurable modular manipulator (rmm).

[2] robolink robot components— igus.

[3] Robotworx: Expert industrial robot integrator.

[4] Derek Calzada-mariaca, Monica Preston, and Yihao Zhou. Modular robotic arm. Technical report, April 26, 2015.

Appendix A Gantt Chart

Figure 2: Gantt chart for project focused on the Control Board