F.O.C.U.S

Field Oriented Control Utilization System

Group 10 Authors:

Hani Bdeir - *Computer Engineering*

Emanuel Alvarez - *Electrical Engineering*

Hailey Gorak - *Computer Engineering*

Jackson Jacques III - *Electrical Engineering*

Nkunu Nuglozeh - *Electrical Engineering*

Mentor, Sponsors, and Major Contributors:

Dr. Chung Yong Chan - *Mentor/Coordinator*

Dr. Mike Borowczak - *Review Committee*

Professor Mark Maddox - *Review Committee*

Chapter 2: Project Description

2.1 Motivation and Background

In the age of AI and automation, we stand on the cusp of a transformative era where robotics is set to revolutionize industries, streamline operations, and enhance human capabilities. Yet, as we look around, we find that there is a significant gap in the market for robotic hardware solutions that are truly dexterous. This means capable of nuanced, precise, and flexible movements that mirror the capabilities of a human. Along with dexterity, the standard industry models lack basic safety features that make these robots extremely dangerous to work alongside as engineers. This is where our mission comes into play. We aim to implement safety features and dexterity in our design.

We are working on developing an industrial automation arm that aims to push the boundaries of current robotic capabilities. By leveraging advanced control techniques and state of the art hardware, our goal is to achieve a high level of dexterity and functionality. Our arm is designed to interact with the world in a sophisticated manner, enabling it to perform tasks with a degree of precision and adaptability that we believe is necessary for future applications. At the heart of this system lies the FieldOriented Control (FOC) motor controller, a critical technology that enables precise and responsive control of motor movements.

2.2 Current Existing Products & Inspiration

FANUC's M2000iA series of industrial arms are designed to hold heavy payloads and survive harsh environmental hazards. These arms are manufactured and sold to companies that focus on production, and require no other internal sensors or programming that integrates machine learning or robot vision. The company FANUC produces many more of these robotic arms with varying specific uses, such as their M800iA series, which holds a considerably lighter payload while boasting a higher level of agility. None of these series incorporate any machine learning or sensors to function properly.

ABB is another leading company in the production of robotic arms similar to what our project aims to achieve. Their IRB 1200 was used to assist in COVID19 vaccine production. The primary function of the arm is to be compatible with the needs of the buyer, since the end effectors of these products are variable according to the demand.

Similar products are constantly being updated by technology focused corporations, such as Intel, who are attempting to integrate machine learning and computer vision into robotic arms for safer and more precise movement. However, these practices are not industry standard and are not being implemented by many companies. These products are often deployed next to human employees, where they assist their engineering counterparts with various automated tasks.

Our product was inspired by these devices, primarily due to the lack of safety standards for these robots that has resulted in many unfortunate accidents for the humans that work alongside them. Our design is not aimed to just mimic these arms, but to add a solution to the safety concerns of these devices that are often overlooked, or simply not implemented. While many companies have achieved some truly incredible feats in regards to their machine's dexterity, compatibility, and adaptability, there is still an overwhelming majority that do not take the proper safety precautions.

These safety issues could be greatly reduced by the addition of FOC servos, which give the engineers full control over position, velocity, and torque. The implication of this is the control over the maximum amount of force that our arm would be able to use, including adjustments over time in response to positional changes that would add a "braking" feature to our arm when a collision happens. This ensures that any living body within the path of the arm would be far less likely to experience larger or fatal injuries.

2.3 Goals and Objectives

The main objective of this project is to design and build a dextrous, safe, and dynamic robotic arm. Below we have listed the main goals of our project in order of complexity and necessity to be completed.

- **Basic Goals:**
    - Our robot will have precise position, velocity, and torque control of all motors.
    - The robotic arm will detect its own position and have the ability to correct itself accordingly.
    - A joystick teleoperation device will be used for movement input with a record and replay button, emergency (E-Stop) off-button, and end effector activator button.
    - Resistive braking system allows the arm to come to a slow stop and not fall limp, avoiding injury to the device as well as any objects or humans near it.

    - An end effector will be added to perform movements involving object manipulation, such as grabbing or picking up.
    - The robot should halt when attempting to manipulate an object outside of its specified parameters, restricting the robot to only interact with correctly weighted objects and stop if obstructed.
- **Stretch Goals:**
    - Use deep learning and PyBullet/ROS Sim environment (Some Computer vision required) for visual object recognition.
    - Add force-feedback to our teleoperational device, allowing the user to feel resistance as the arm encounters it.
- **Objectives:**
    - We aim to detect the position of our robotic arm using rotary encoders as positional sensors.
    - The movements of our robotic arm will be controlled using a teleoperational device.
    - Precise position, velocity, and torque control will be achieved with a FOC algorithm.
    - The data received from the robotic arm's hardware will be sent through a CAN bus to communicate with our software.
    - A braking system will be implemented using a resistive load.
    - We will program the robotic arm with the ability to distinguish the weight of the object it is manipulating using the torque feedback of the arm.

2.3.1 Detailed Description of Features and Functionalities:

Our project will include precise and accurate motor control. It will consist of multiple motors, which will drive the joints and movement of our robotic arm. Each motor will utilize a motor control PCB. Included in the functionality of the arm will be realtime teleoperation controls, designed to be operated from a custom controller. Using advanced kinematic algorithms, we will be able to simulate smooth and flexible movements in the arm.

With all of these basic goals completed, we then aim to implement feedback from the arm's operation through the joystick movement. This is done by adding resistance to the operator when the arm encounters an object. The potential end effector of our robotic arm will be determined by the end usage, as the end effector should be interchangeable and compatible with many different products.

2.3.1-1 Motor Control PCB Hardware Requirements for Features and Functionality:

The Motor Control PCB is engineered to provide high precision and accuracy essential for controlling the motors of our robotic arm. This PCB features several critical hardware components and functionalities designed to meet the specific needs of our application, ensuring both operational excellence and safety in human interactive environments. Below is a table of the features and the required hardware we need to support the features.

Feature	Required Hardware
High Precision and Accuracy in Control	● High Frequency Microcontroller ● Position Sensor ● Current Sensor ● Motor Driver
High Speed Communication Across Boards and Communication for Teleoperation	CAN bus Transceiver IC such as mcp2515
High Power Handling	● Power Input/Output Ports ● Regulators for motor and microcontroller power
Firmware Programmability	Programming Interface
Precise Motor Feedback	Rotary Encoder Inputs Current Sensor
Future Sensor Integration	Provisions for Additional Sensor Inputs
Emergency Power Loss Handling	Resistive Load Breaker with same logic level as MCU

2.3.1-2 Teleoperation PCB:

The Teleoperation PCB is a critical component of the F.O.C.U.S. project, designed to enable intuitive and precise control of the robotic arm through a custom-built teleoperation unit. This unit, inspired by the control systems of heavy machinery such as

excavators, will utilize dual joysticks to provide the operator with a natural and efficient interface for manipulating the arm in three dimensions. The left joystick will control the stick (forward/backward) and swing (left/right), while the right joystick will control the boom (up/down) movements. The addition of purpose built end effectors, in this case a parallel claw apparatus alongside an accompanying button will increase the capabilities of our robot.

To ensure seamless operation, the Teleoperation PCB must interface reliably with the central computing unit via a low-latency communication protocol, such as the CAN bus. This communication is vital for promptly transmitting commands from the operator to the robotic arm. The PCB will process inputs from the joysticks and buttons and convert them into precise control signals for the robotic arm's motors. Safety is also a paramount concern for this project. Therefore, the Teleoperation PCB will incorporate an essential safety feature, that being an emergency shut-down button. Furthermore, the system will include a record and replay feature, allowing the operator to record sequences of movements and replay them as needed. This functionality is particularly useful for repetitive tasks, improving efficiency and consistency.

In terms of hardware, the PCB will be equipped with a high-performance microcontroller capable of handling input processing, communication, and safety features. During the prototype phase, an Arduino will be used to validate the design. Once the functionality is confirmed, a custom PCB with only the necessary components will be fabricated. The PCB will also include power management systems to handle the power requirements of the joysticks and microcontroller, ensuring stable operation and effective management of power loss scenarios. Input ports for connecting the joysticks and output ports for communication with the central computing unit and other robotic arm components will also be part of the design.

Software integration will involve developing firmware to run on the Teleoperation PCB. This firmware will process joystick inputs, manage communication with the central computing unit, and implement safety protocols. Initially, this will be developed on the Arduino platform, with plans to adapt it to the custom PCB later. The communication protocol will use the CAN bus to ensure efficient and reliable data exchange.

Testing and validation of the Teleoperation PCB will be comprehensive. Initial bench testing will verify the functionality of the joysticks and communication interfaces using the Arduino-based prototype. Following this, integration testing will be conducted to ensure the PCB works seamlessly with the motor control PCB and the central computing unit, guaranteeing smooth and precise operation of the robotic arm. Key performance

metrics such as response time, communication reliability, and input processing accuracy will be rigorously tested to ensure the system meets the project's specifications.

By carefully designing, implementing, and validating the Teleoperation PCB, we aim to create an intuitive, reliable, and safe control system for the F.O.C.U.S. project, significantly enhancing its functionality and ensuring it meets the needs of its users.

2.3.1-3 Software:

A primary computing device, such as a Raspberry Pi or laptop, will be interfaced with our motor's PCB via a communications bus, such as a CAN bus. This main computer will be responsible for conducting kinematic calculations, which include processing inputs like the current position, motor torque/current readings, and motor ID. Based on these inputs, it will compute the target position and ensure safe current levels throughout the movement, according to the task requirements and positional feedback. Additionally, teleoperation inputs are received from the controller PCB, which communicates with the main computer either through the CAN bus or an alternative protocol.

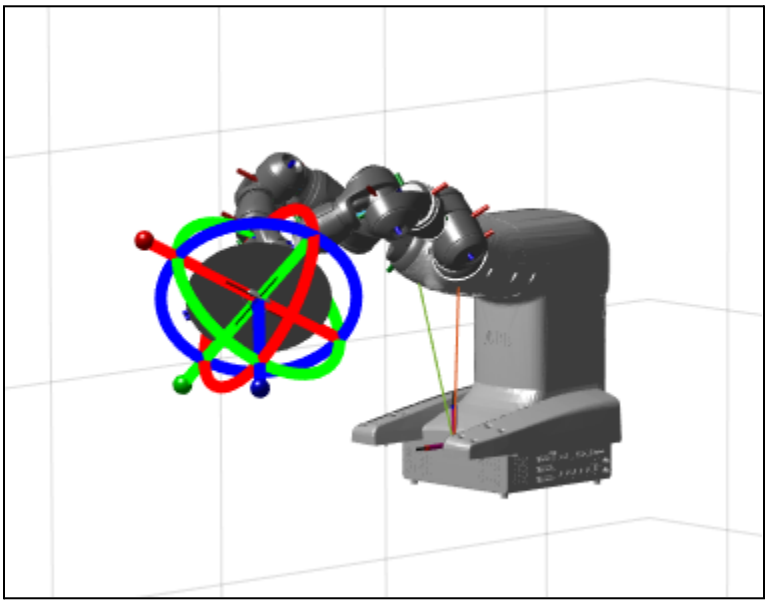

Figure 2.3.1-1 - Robotic Arm Simulation

Software can also be used for simulation purposes. By defining a virtual environment

including directional gravity, mass, and friction coefficients, the movements of a virtual robot can be simulated. The figure above shows an ABB Yumi robot simulated in Mathworks as an example. An advantage to using this method is that the physics environment can be defined with any parameters we need and we can make tests using several kinds of robots. This way we can adapt our simulations as our design evolves. By simulating a robotic arm in this way, we can perform tests on the functionality of control systems software to address issues before they materialize in a hardware implementation. Since control systems including FOC can be designed in the same virtual environment that the simulation and embedded implementation utilize, a simplified workflow between design, simulation, testing, and implementation can be constructed.

Simulation software can also be used to gather parameters for the movement of the robotic arm to be dictated by the microcontroller. This has the potential to save significant development time as it reduces the trial and error involved in the process via manual implementation.

2.4 Engineering Specification

Based on the goals and features in section 2.3 we needed measurable specifications for our design decisions and have listed them below:

Specification	**Measure**
Motor Control Accuracy	±1 degree
Load Capacity (W)	0.5kg<W<3.5 kg
Load Sensing Accuracy	±0.5 nm
Response Time of Autonomous Operation	< 100 ms
Response Time Teleoperation	< 200ms
Arm Reach	< 1 Meter
Active Compliance/ Backdrivable Minimum Input	±1 nm
Peak Power Draw	<400 watt

2.5 Block Diagrams

To demonstrate the flow of our project and purpose of each component, we have created diagrams for both the hardware components and the software that we will be utilizing in the project. This includes a diagram for our Field Oriented Control in the software section,

2.5.1 Hardware Block Diagram:

The hardware block diagram below provides a clear overview of the system design. It features three main sections: Joystick PCB, Motor PCB, and Power Supply. The Joystick PCB Housing includes components like dual joysticks and an interface circuit for operator input, which sends signals to the central CPU. The Motor PCB Housing contains motor controllers and sensors to manage the robotic arm's movements. The Power Supply Housing ensures all parts receive stable power. Each section is color-coded to show which team member is responsible and the status of each component, whether it needs to be acquired, is under investigation, is being designed, prototyped, or completed. This layout helps in understanding the project and keeping track of progress.

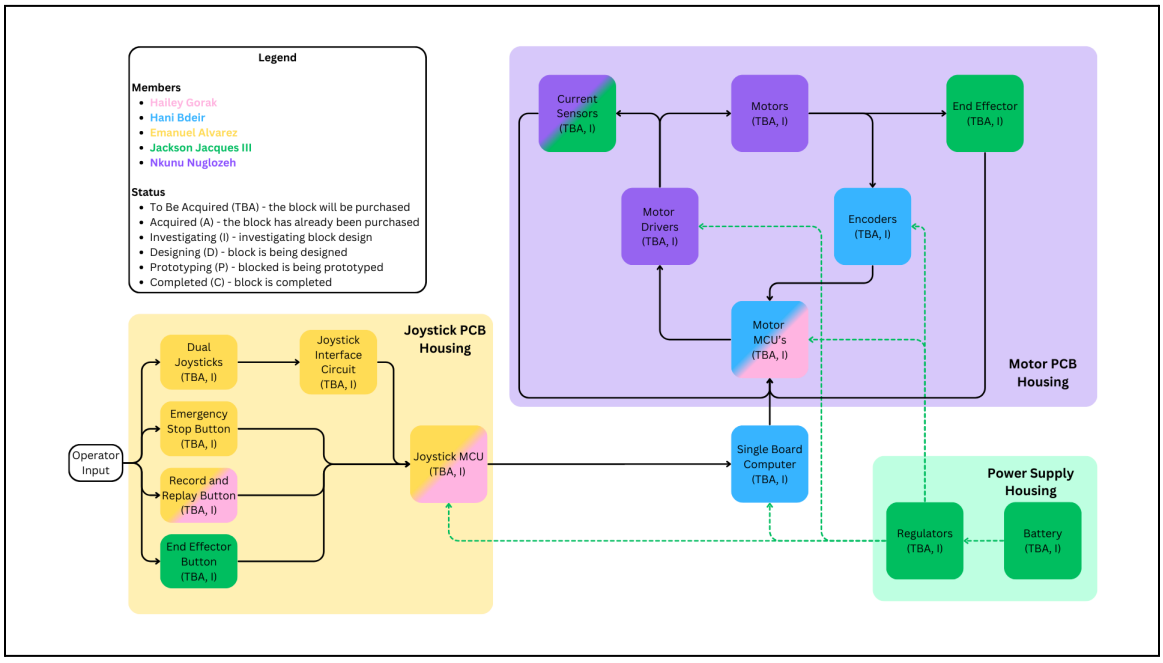

Figure 2.5.1 - Hardware Block Diagram

2.5.2 Software Block Diagrams:

Below is a color-coded description of the division of labor for the intended software and coded processes of our project, not including simulations used for testing. The movement of our arm will be determined by the analog input of our joystick, which will also have the capability to record and replay movement from the joystick. Most operations programmed into our robotic arm will be done with the ROS language, which is designed specifically for interaction in robotic systems. The programming done for our arm will also involve solving mathematical operations based on sensor input to correctly judge the robotic arm's positional velocity, giving us further control over how much current it uses. This greatly improves its functionality in terms of safety, as it helps our arm to judge how much force it needs to use to move from its current position to its desired position.

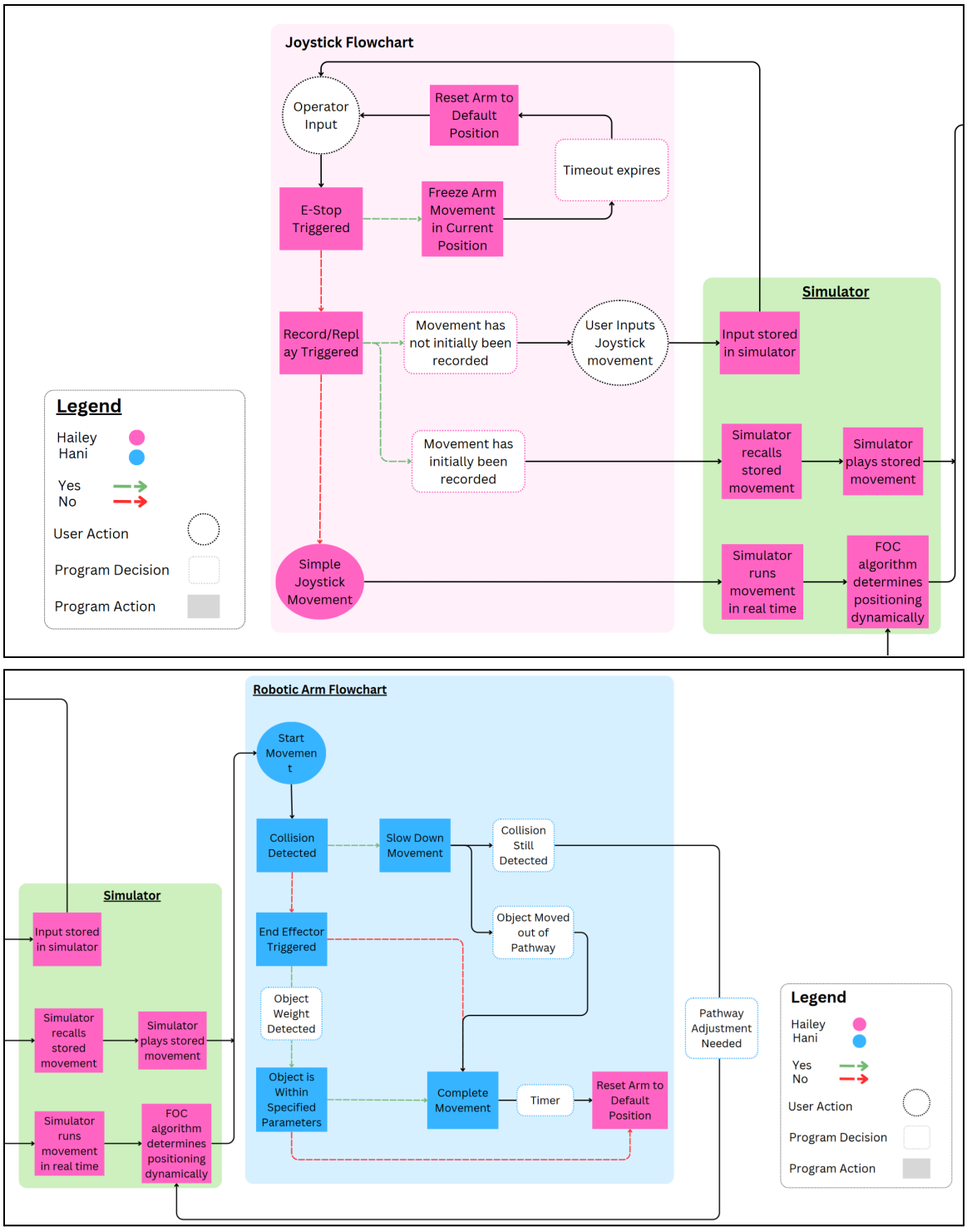

Figure 2.5.2-1 - Software Block Diagram

The FOC operations pictured in our software diagram will be handled internally with code, and externally with encoders on the motors to act as positional sensors. Below is a diagram of how these FOC operations should function alongside the robotic arm's motors and power source. The green blocks indicate the physical components of our FOC operations, which primarily include our power sources and our motors.

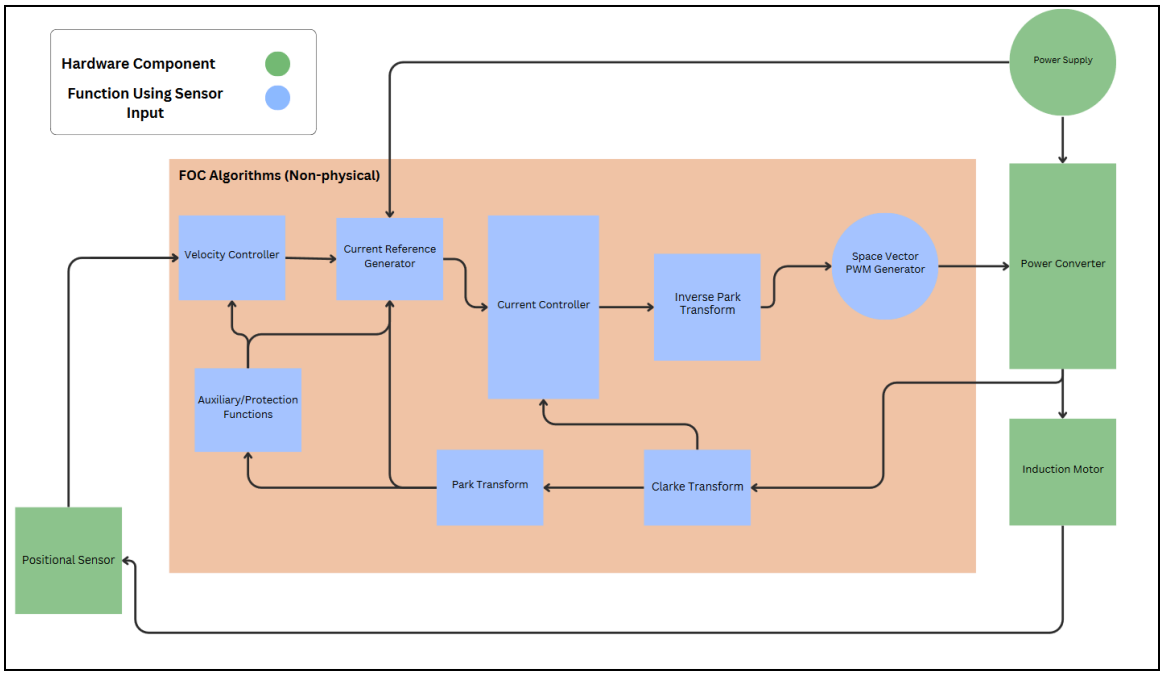

Figure 2.5.2-2 - FOC Operations Block Diagram

2.5.3 House of Quality:

Legend	
•	Maximize
○	Minimize
↑	Positive Correlation
↓	Negative Correlation
↑↑	Strong Positive Correlation
↓↓	Strong Negative Correlation

Customer Requirements ↓	Engineering Requirements →	Motor Control Accuracy	Maximum Load capacity	Load Sensing Accuracy	Response Time (Autonomous)	Response Time (TeleOp)	Arm Reach	Active Compliance/ Backdrivable Minimum Input	Peak Power Draw
		•	•	•	○	○	•	•	○
Precision/Accuracy	•	↑↑		↑↑	↓	↓	↑	↑	↓
Speed/Efficiency	•	↑		↑					↓
Durability	•		↑				↑	↑	
Ease of Use	•							↑↑	
Load Capacity	•		↑↑	↑↑					
Flexibility/Versatility	•	↑↑	↑↑					↑	↓
Cost	○	↓		↓					↓
Safety	•			↑			↑	↑↑	
Targets		±1 degree	>=3 kilograms	±0.5 nanometer	< 20 milliseconds	< 200 milliseconds	>= 1 Meter	±1 nanometer	<400 watts

Figure 2.5.3 - House of Quality Diagram

2.6 Budget and Financing

This project will be funded out of pocket.

We will need one set of the items below perc robot Axis, This includes the motor driver and motor:

Component Category	Type	Quantity	Price Range (Each)
Power Supply	24V Power Supply	1	$20 - $70
Microcontroller	General-purpose microcontroller	1	$4 - $8
Sensors	Current sensors	3	$2 - $5

Component Category	Type	Quantity	Price Range (Each)
Sensors	Rotary Encoder	1	$10 - $50
Motor Driver	Gate driver IC	1	$5 - $15
Motor Driver	Power transistors	6	$1 - $3
Passive Components	Capacitors	20	$0.1 - $1
Passive Components	Resistors	50	$0.01 - $0.1
Communication	Communication transceiver	1	$1 - $3
Connectors	Various connectors	10	$0.5 - $2
Miscellaneous	Heat sinks, mounting hardware, etc.	-	$10 - $20
Motor	BLDC Motor	1	$20 - $60

We will need one controller PCB:

Component Category	Type	Quantity	Price Range (Each)
Power Supply	5V Power Supply	1	$10 - $20
Microcontroller	General-purpose microcontroller	1	$4 - $8
Communication	CAN Transceiver	1	$1 - $3
User Interface	Buttons	2	$0.5 - $2
User Interface	Joysticks	2	$5 - $15
Safety	Emergency Stop (E-stop) button	1	$5 - $15

Connectors	Various connectors	10	$0.5 - $2
Passive Components	Capacitors	10	$0.1 - $1
Passive Components	Resistors	20	$0.01 - $0.1
PCB	Custom PCB, 2-4 layers	1	$20 - $50
Miscellaneous	Heat sinks, mounting hardware, etc.	-	$5 - $10

2.7 Project Milestones

2.7.1 Milestones for SD1:

Milestone Category	**Milestones**
Project Planning and Initial Research	- Finalize project scope and objectives
	- Conduct a thorough literature review on current robotic arms and FOC motor controllers
	- Identify key technologies and components required for the prototype
Conceptual Design and System Architecture	- Develop the conceptual design of the robotic arm

	- Create detailed system architecture diagrams, including hardware and software components
	- Define the control algorithms and kinematics for the arm
Component Selection and Procurement	- Select and procure motors, sensors, controllers, and other essential components
	- Ensure all components meet the basic requirements for the prototype
Mechanical Design and Fabrication	- Design the mechanical structure of the multi-axis arm
	- Fabricate and assemble the mechanical components
	- Integrate the motors and sensors into the mechanical structure
Electronics and Control System Integration	- Design and build the electronic control system
	- Integrate the FOC motor controllers with the motors
	- Set up the initial wiring and connections
First Iteration of PCB Design and Fabrication	- Design the first iteration of the PCB for motor control and communication
	- Fabricate and test the PCB
	- Integrate the PCB with the existing hardware

Communication Setup	- Establish communication from all motors back to a central computer
	- Implement protocols for data exchange and control commands
	- Test the communication system for reliability and responsiveness
Software Development and Testing	- Develop the basic software for controlling the arm, including position, velocity, and torque control
	- Implement forward and inverse kinematics algorithms
	- Begin preliminary testing of the control system and software
Prototype Assembly and Initial Testing	- Assemble the complete 3-axis arm prototype
	- Conduct initial testing to ensure basic functionality
	- Identify and troubleshoot any issues
Functioning Multi-Axis Arm Prototype	- Finalize the assembly and testing of the 3-axis arm prototype
	- Ensure the prototype meets basic functional requirements, even if it does not hit all specifications
	- Document the performance and any deviations from the initial specifications

2.7.2 Milestones for SD2:

Milestone Category	**Milestones**
Central Computer Control	- Develop and implement software to enable the central computer to control all motors
	- Integrate Robot Operating System (ROS) for centralized control and communication
Full-Featured Software Development	- Build a comprehensive, compliant software system for the arm
	- Incorporate advanced features for precise control and adaptability
	- Ensure the software meets all functional and performance specifications
Implement Teleoperation(Advanced Goal)	- Develop and implement teleoperation capabilities
	- Test and refine the teleoperation system for real-time control and feedback
	- Ensure the system is reliable and user-friendly
Force Feedback for Teleoperation (Stretch Goal)	- Develop force feedback software and hardware mechanisms for teleoperation
	- Reference "Steer by Wire" research and implementations for force feedback and use automotive safety standards
	- Integrate force feedback into the teleoperation system
	- Test and refine the force feedback functionality to enhance user experience and control precision

Appendices

- **Appendix A – References**

FANUC America Corporation, *Fanuc Robots By Series,* M800iA/60 Robot | FANUC America, FANUC America Corporation 2024

ABB, *ABB Robots Aid Rapid Automated Testing For COVID19 Virus,* ABB robots aid rapid automated testing for COVID19 virus, ABB Corporation 2021

- **Appendix B – Copyright Permissions**
- **Appendix C – (Other as needed)**