COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

The Design Document for CSEE 4840 Embedded System Design

Gesture Controlled Manipulator Using FPGA

Group Members: Fan Wu (FW2392), Jiamiao He (JH4593),

Tailai Zhang (TZ2550), Yi Wang (YW395)

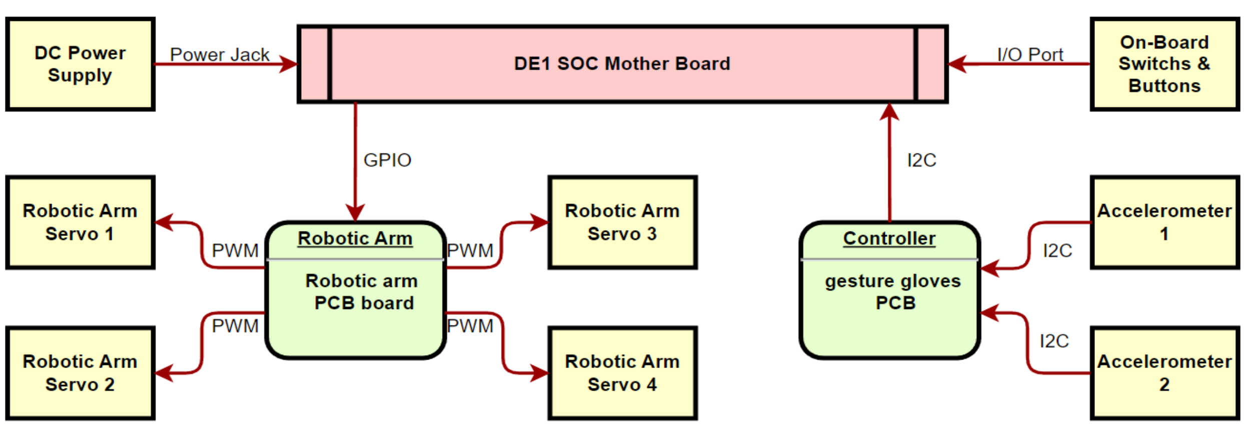
Contents

1 Introduction

Manipulators are used in various applications such as manufacturing, healthcare, and transportation. The precise control of manipulators is essential for their safe and efficient operation. Field-Programmable Gate Arrays (FPGAs) are increasingly being used for controlling manipulators due to their high-speed processing capabilities, low power consumption, and programmability. In this proposal, we propose to develop a manipulator control system using FPGA.

2 System Block Diagram

In our upcoming project, we aim to develop a highly innovative gesture-controlled manipulator system. To achieve this, we will be utilizing two accelerometers, integrated into a specially designed glove, to collect real-time gesture data from the wearer. This data will then be analyzed by the powerful DE1-SOC platform, which will imitate the movement of the wearer's hand on a 3-axis robotic arm. Through the use of this control glove, users will have complete control over the robotic arm's movements, enabling them to grasp, transport, place, and adjust the posture of objects in its field of view with ease. On the firmware side, we will use factory PCB and cables to build the electrical links for two accelerometers, four servo motors, a DE1-SOC board. Furthermore, we will be leveraging 3D printing technology to design and fabricate the mechanical support structure of the entire system, ensuring maximum durability and stability. On the software side, we will be using advanced programming languages such as System Verilog and C to develop robust firmware drivers and core algorithms, enabling seamless communication between the various hardware components and ensuring optimal system performance.



3 Algorithms

We planned to use a keyboard to control the robot arm, not only to make it move forward, backward, left and right, but also to move the arm to specific positions using specific keys. Then, we need to implement an Inverse Kinematics algorithm to calculate and move the arm to the desired position. Inverse Kinematics is a method of determining the motion of a system of interconnected objects, to find the values of the joint parameters that will achieve a desired position and orientation of the end effector. For the Inverse Kinematics on FPGA, we are planning to use the Jacobian-based approach, which involves calculating the Jacobian matrix of the robot arm and then using it to iteratively adjust the joint angles to approach the desired end effector position and orientation. The algorithm is planned to be implemented using the C language on the FPGA.

4 Resource Budgets

Category	Size (Bits)	Number of Usage	Total Size (bits)
Keyboard Driver	64000	1	64000
Servo Motors Driver	12800	4	51200

5 The Hardware/Software Interface

5.1 Hardware Design

Our hardware implementation will utilize a DE1-SOC board to handle computing, storage, and physical interfaces. The board's onboard GPIO interfaces will enable the connection of sensors and the robotic arm. Onboard switches and buttons will allow for control over the manipulator, including the ability to switch between automatic and manual operation modes, initialize the origin, debug, and start and stop actions. Real-time attitude information from the accelerometer will be received by the FPGA board via the I2C communication protocol, which will then perform calculations to correct the current motion behavior of the robotic arm.

5.2 Software Design

We will use SystemVerilog and C language to develop the firmware drivers and core algorithm of the system. The firmware drivers will be responsible for the data transfer between the sensors, actuators, and FPGA board, while the core algorithm will control the manipulator's position, velocity, and acceleration accurately. We will develop trajectory planning and path following algorithms to enable the manipulator to move along a specified path with precision. We will also develop a user interface to enable users to interact with the system easily and intuitively.

5.3 Mechanical Design

We will design the manipulator structure using 3D modeling software (Solidworks) and 3D printing technology, which will allow us to build lightweight and sturdy structures. Preliminary CAD design is showed in fig. 1.

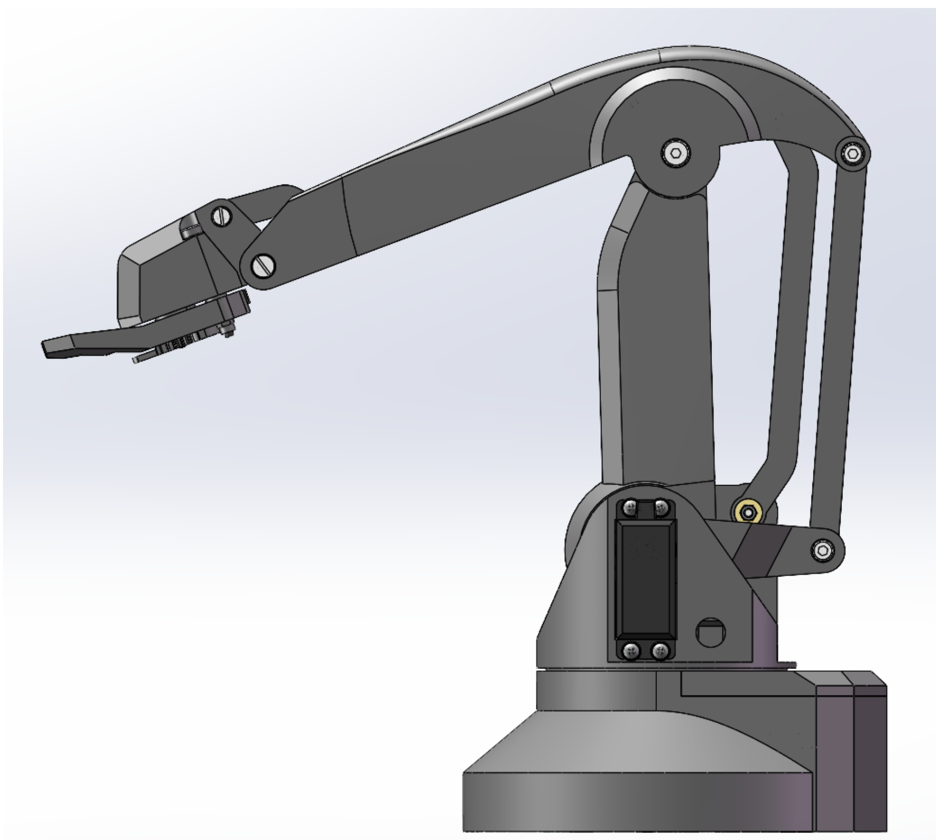


Fig 1. Preliminary CAD design