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

The Final Report for CSEE 4840 Embedded System Design
Keyboard-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 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 Methodology

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 the desired position and orientation of the end effector. The algorithm is planned to be implemented using the C language on the FPGA.

3 Mechanical Design

3.1 CAD design

We will design the manipulator structure using 3D modeling software (Solidworks) and 3D printing technology, allowing us to build lightweight and sturdy structures. Preliminary CAD design is showed in fig. 1. The manipulator consists of 4 servos which are used to control the movement of the links, opening and closing of the end effector, and the rotation of the manipulator's body relative to the base.

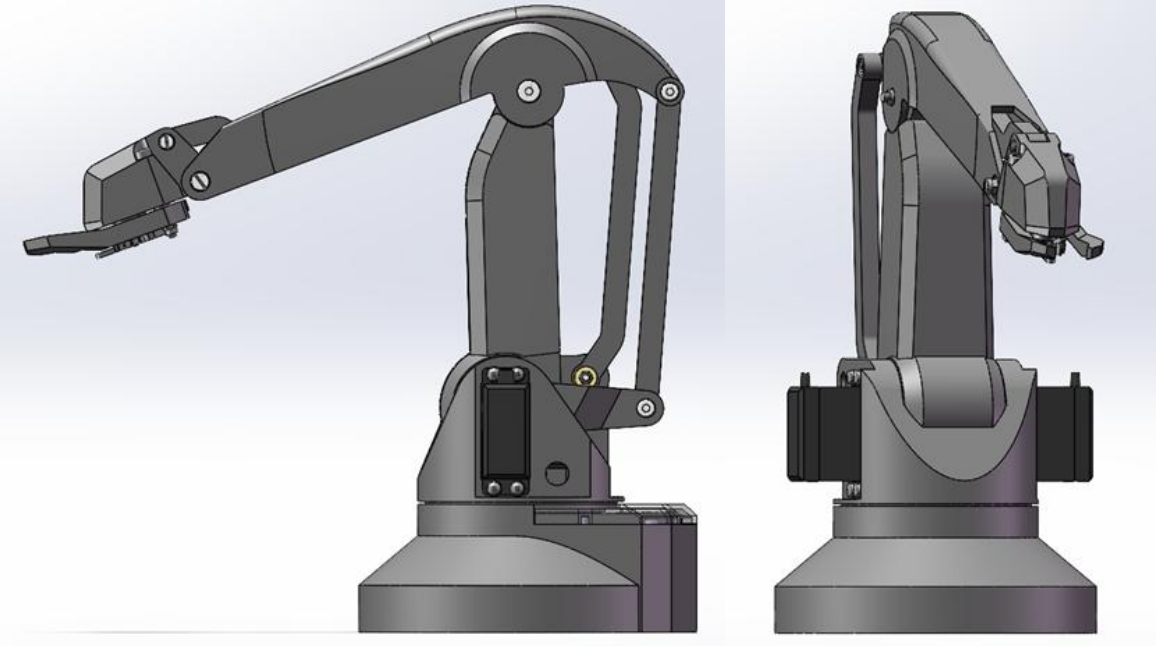

Fig 1. Sketch of CAD design

3.2 Inverse Kinematics:

In the inverse kinematics calculation for the 3-DOF robotic manipulator, the goal is to find the joint angles (theta1, theta2, theta3) given a desired end-effector position (x, y, z). The robotic manipulator consists of two links with lengths l1 and l2.

First, the z-coordinate is adjusted to account for the manipulator's base height. To calculate theta1, the cosine law is applied to the triangle formed by the two links (l1, l2) and the distance from the end-effector to the first joint in the yz-plane. The angle is then corrected for the inverse tangent of the ratio of z to y, as well as for the angle between the second link and the yz-plane. Finally, a constant value is subtracted to account for the specific robotic manipulator's configuration. For theta2, the same cosine law is used with a similar triangle, but this time considering the angle between the two links. The inverse tangent of the ratio of z to y is then added to the result. To determine theta3, the inverse tangent of the ratio of y to x is calculated.

$$L_3 = \sqrt{y^2 + z^2}$$

$$\tan(a) = z/y$$

$$\measuredangle a = \arctan(a)$$

$$\cos(b) = \frac{L_3^2 + L_2^2 - L_1^2}{2L_2L_3}$$

$$\measuredangle b = \arccos(b)$$

$$\cos(d) = \frac{L_1^2 + L_2^2 - L_3^2}{2L_1L_3}$$

$$\measuredangle d = \arccos(d)$$

$$\measuredangle e = 158 - \measuredangle d$$

$$\measuredangle c = 180 - \measuredangle e$$

$$\measuredangle \alpha = 180 - \measuredangle a - \measuredangle b - \measuredangle c$$

$$\measuredangle \alpha = 202 - \arctan\left(\frac{z}{y}\right) - \arccos\left(\frac{y^2+z^2+L_2^2-L_1^2}{2L_2\sqrt{y^2+z^2}}\right) - \arccos\left(\frac{L_1^2+L_2^2-y^2-z^2}{2L_1L_2}\right)$$

$$\measuredangle \beta = \measuredangle a + \measuredangle b$$

$$= \arctan\left(\frac{y}{x}\right) + \arccos\left(\frac{y^2+z^2+L_2^2-L_1^2}{2L_2\sqrt{y^2+z^2}}\right)$$

$$\measuredangle \gamma = \arctan\left(\frac{y}{x}\right)$$

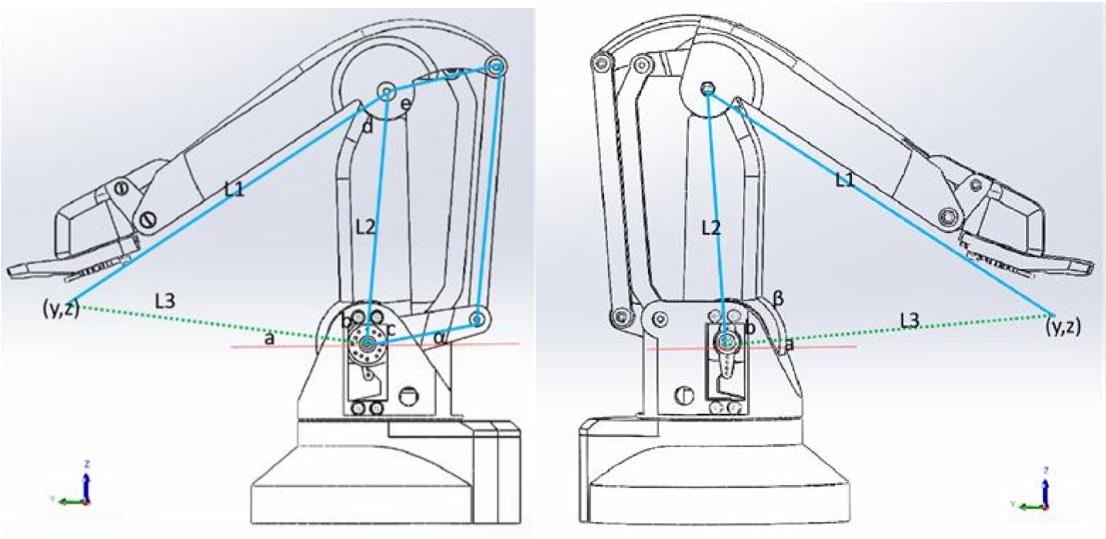

Fig 2. Manipulator geometry

After finding the joint angles (theta1, theta2, theta3), adjustments are made based on the initial angles of the robotic manipulator. These adjustments are made by finding the difference between the calculated joint angles and the initial angles. The input angles for the robotic manipulator are then computed by adding or subtracting these differences from an initial angle.

3.3 Workspace Analysis

The workspace of a robotic manipulator refers to the set of all reachable points that the end-effector can achieve within its operational range. The workspace is determined by the manipulator's geometry, joint limits, and kinematic configuration. After calculating the forward kinematics, the workspace of the manipulator end-effector is shown below.

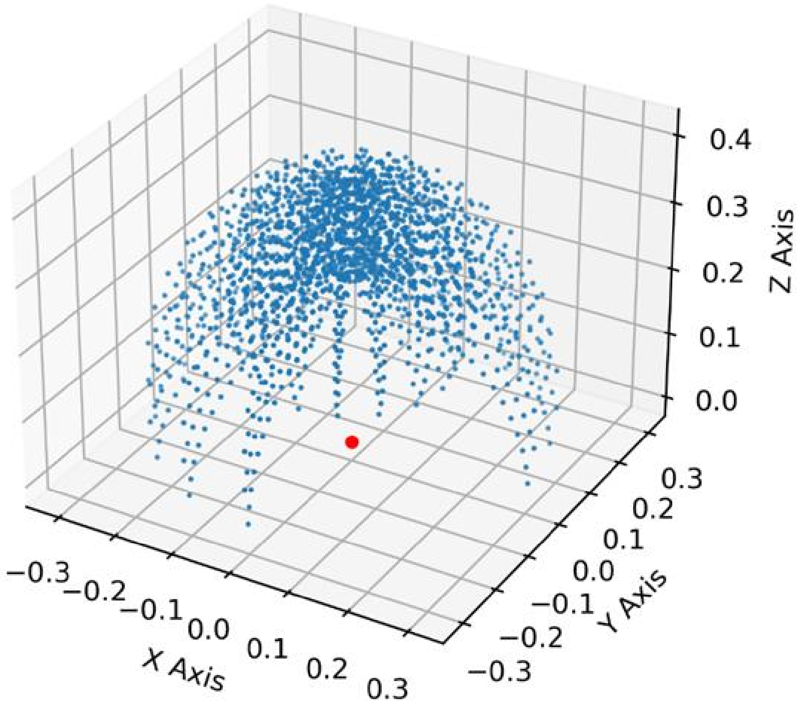

Fig 3. Workspace of the end effector

4 Hardware Design

4.1 System

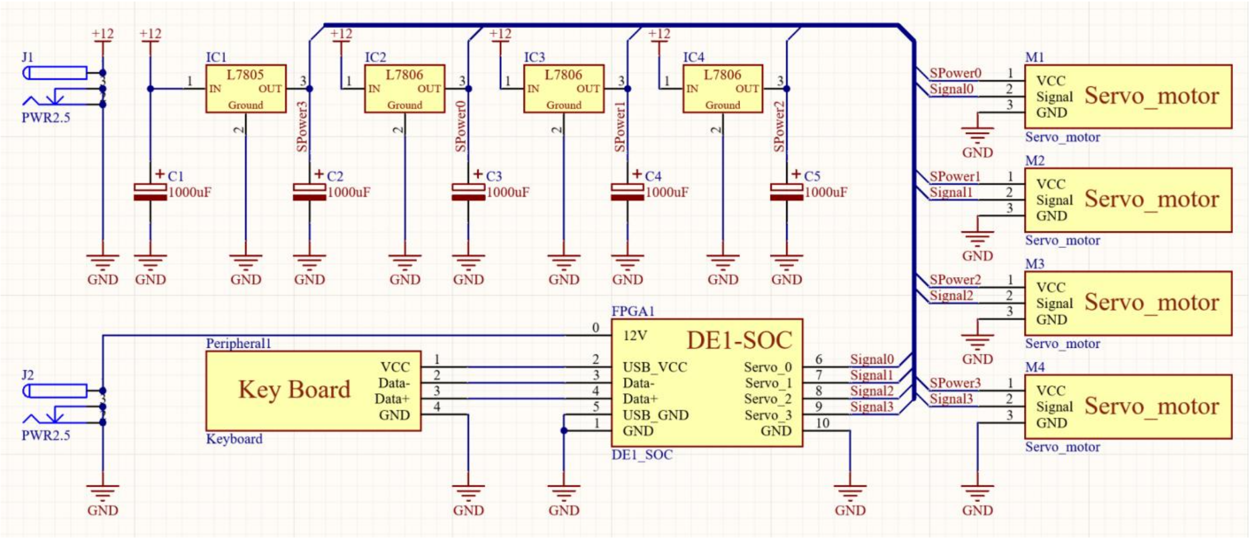

Fig 4. System Diagram

Our hardware setup was built around the DE1-SOC board, which employed the onboard 5CSEMA5F31C6N for processing, storage, and physical interface management. By utilizing the board's built-in GPIO and USB interface, we facilitated signal transmission for the keyboard and robotic arm. This setup provided comprehensive control over the manipulator, including the option to toggle between automatic and manual operation modes. It also enabled us to establish origin points, perform debugging, and seamlessly initiate or halt operations.

4.2 Peripherals

Our system was designed to acquire control signals for the robotic arm via an external keyboard using the onboard USB interface of the DE1-SOC board. The collected coordinate signals were processed and translated into four angular data points. These data points were transmitted to the servomotors in real-time through the General Purpose Input/Output (GPIO) interface. The servomotors were controlled using Pulse Width Modulation (PWM) for precise angular adjustments. Once the PWM signals were acquired, they underwent a process of signal isolation to avoid any potential interference. Following isolation, these signals were passed through a filter to be converted into analog signals. These analog signals were then compared with a reference voltage level. The resulting differential was calculated and integrated. The output of this operation was subsequently delivered to the DC reduction motor. This process ensured highly accurate and efficient control of the robotic arm, showcasing the system's sophisticated design and functionality.

4.3 Power Source

To safeguard the FPGA power supply from any interference caused by the servomotors, we incorporated an additional power adapter to supply power to the four servomotors independently. An external 12-watt power supply was used to drive four independent linear regulators (L7806/L7805), providing three sets of 6V power to the three main servomotors and one set of 5V power to the end effector of the robotic arm.

Under non-overload conditions, each servomotor operated at an average power of 3 watts, which was well within the capacity of the external power adapter. This setup ensured that any voltage fluctuations due to overload on the servomotor power lines did not affect the power supply of the FPGA. As a result, we achieved a more secure and stable operating environment for the FPGA, demonstrating the effectiveness of our thoughtful and strategic system design.

5 Software Interface

The task of establishing an interface between the keyboard and the FPGA is accomplished using libusb 1.0. This C library is renowned for its capability to ensure smooth interactions with USB devices across a broad spectrum of operating systems.

Upon the pressing of a key, the system initiates a chain of actions. First, it determines and acquires the existing position from the manipulator. The position is then adjusted in accordance with the specific key that has been activated and the current position of the manipulator. Following the implementation of necessary alterations, the revised position values are conveyed back to the manipulator. This process enables us to modify the manipulator's position in alignment with real-time user inputs. Furthermore, the system retains its prior state to confirm that it responds only when a single key is pressed, effectively precluding the chances of conflicting keyboard inputs.

To establish a connection between the keyboard inputs and the manipulator, we integrated a miscellaneous device into our framework. This device is responsible for transmitting four distinct 8-bit unsigned integer values, which dictate the movements of the major arm, minor arm, base, and clamp of the device. These values are disseminated via the Avalon bus, subsequently reaching the memory registers of the FPGA board before being received by the manipulator.

6 Future Work

Currently, we have implemented keyboard and spatial coordinate control for our robotic arm using FPGA. In the future, we are considering adding a camera to our robotic arm. By extracting images from the camera and integrating machine learning and deep learning algorithms, we can directly extract the location information of an item in space from the images. This will enable us to issue instructions to grab the target object automatically.