The Development of a Computer Software to Control a Robotic Arm to Draw

University of Findlay

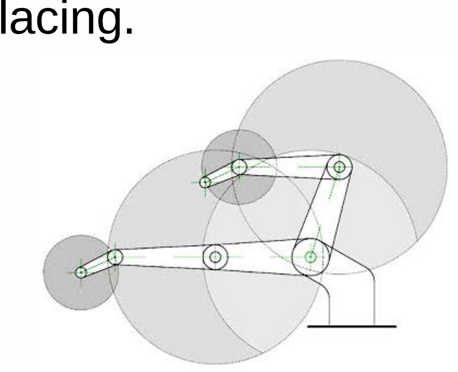
Presenters: Mahdi Khansari, Arash Mahmoudian

Sponsored by Prof. Heda Samimi

1. Introduction

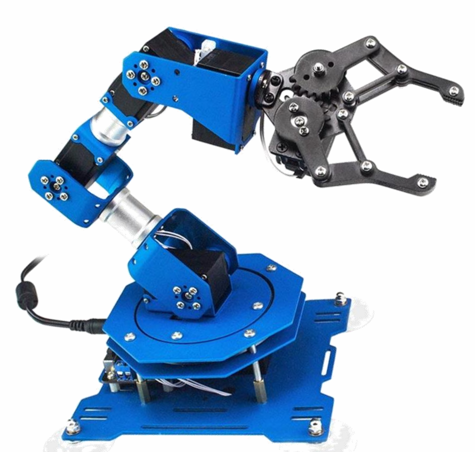
A computer controlled mechanical arm is called a robotic arm. They are known for their high accuracy and speed. They are being used for a wide range of purposes in different industries to perform tasks which need high precision such as welding or laser cutting or tasks which need to be repeated for many times like picking and placing.

Most of the robotic arms use electric motors in their joints to move the arm, also usually there is an actuator at the tip of arm which performs a specific task, like a gripper to pick objects. The 3D volume space that the arm can reach is called the work envelope. Finding the exact position of the actuator head is complex and is the main purpose of this project.



Work Envelope

In this project we used a 6 servo motors LewanSoul xARM robotic arm which could only rotate each motor individually.



LewanSoul xARM Robotic Arm

2. Objective

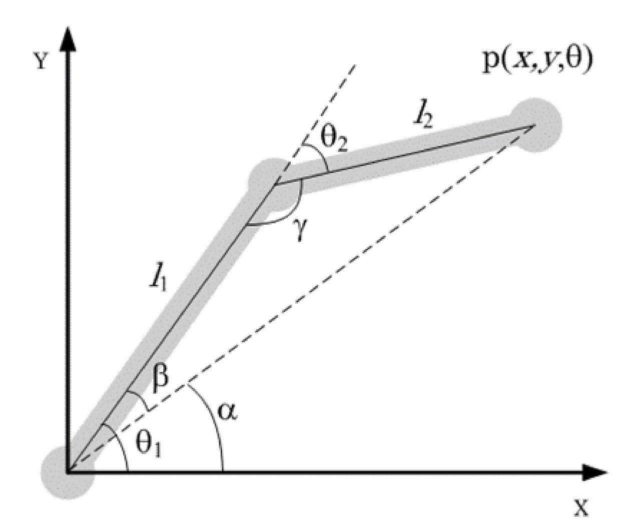
Robotic arm control systems run each motor as a separate element on a joint, which for practical activities requires multiple motor rotations with specific timing. In order to make the xARM robotic arm able to draw a shape using computer software, we need the following requirements:

- A formula to project the 3D work envelope of the xARM robotic arm onto a 2D canvas
- Communication protocol for xARM servo motors
- Software to draw and convert shapes to commands for xARM
- A pen at the gripper of xARM and a canvas

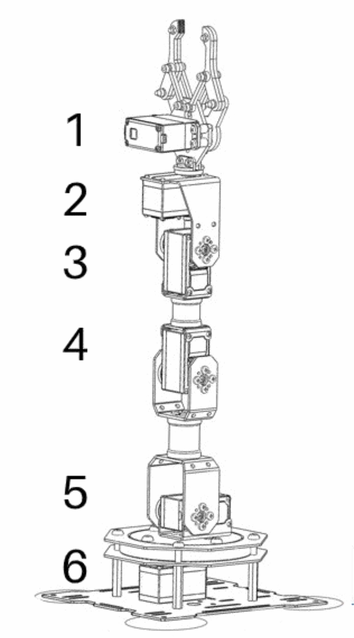
3. Methods

3.1 Projection Formula

In order to move the pen on the canvas, we used a specific formula to keep the xARM gripper on a 2D surface. This formula could be used for a two-jointed robot, so we kept four of the motors (Motors 1,2,3,6) in a fixed position.



2 Joints Robotic Arm Schematic



xARM Motors

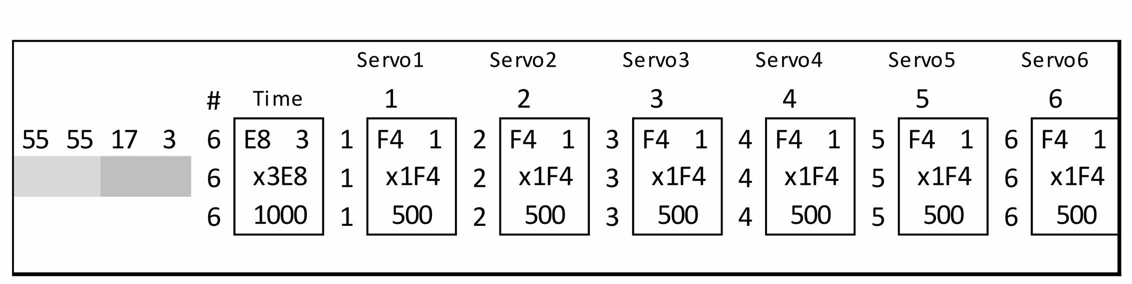
$$\theta_1 = \arcsin\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2+y^2}}\right) + \arctan\left(\frac{y}{x}\right)$$

$$\theta_2 = \arccos\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right)$$

2 Joints Robotic Arm Angle Formula

3.2 Communication Protocol

The xARM connection to a PC is through USB ports. We used a USB analyzer to find how the protocol works, so by performing many experiments, finally we succeeded to model the protocol to communicate with each servo motor. The protocol is shown below:



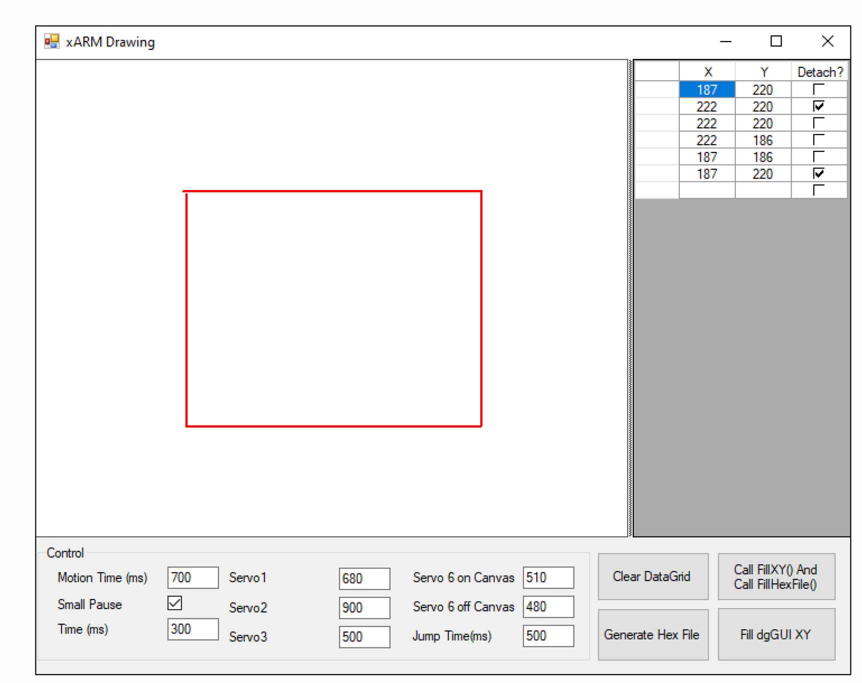
Motor communication frame protocol

3.3 Computer Software

The main purpose of this project is to develop computer software with the following features:

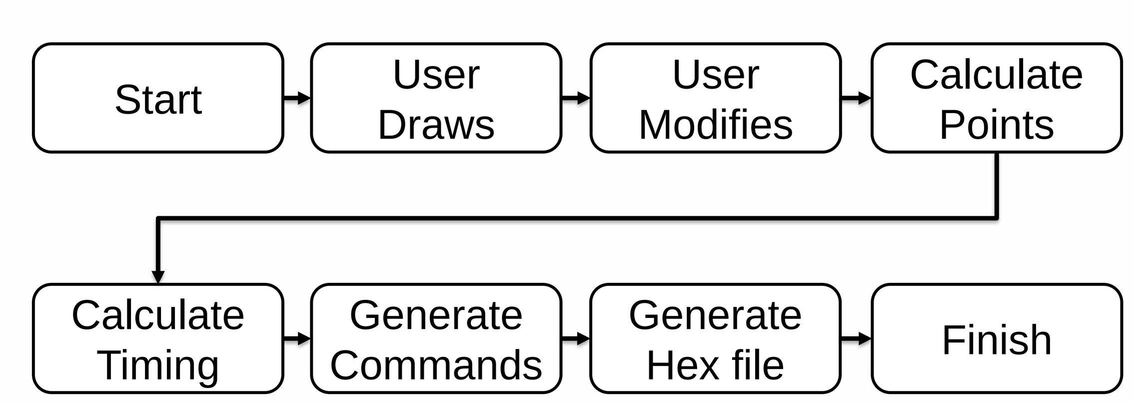
- A screen for the user to draw shapes by using lines
- A list of points with the ability to change
- Timing calculation and pause tuning
- Converting the drawing into the motors' angle
- Creating commands compatible with the motor communication protocol

We developed the software using C# in Visual Studio 2019. It consisted of four classes and about 1,500 lines of code.



Developed Software

This software uses the following flowchart to perform calculations and generate the commands:



3.3 Hardware

We fixed a pen at the gripper position of the xARM so it stands perpendicular to the canvas. We also placed the canvas in a location where the movements of the robotic arm would be parallel to the surface of the canvas.

Pen at gripper

4. Results

After hundreds of tries and calibrations, we finally succeeded in drawing the shape with the robotic arm. Considering that this robot is made for academic or entertainment purposes, the gearboxes of the servo motors had some vibration through the movements, which we minimized by proper timing.

xARM drawing

5. References

- InTech Open

www.intechopen.com/books/robot-arms/kinematics-of-adeptthree-robot-arm

- LewanSoul Documents

http://www.lewansoul.com/download/list/

ACKNOWLEDGEMENTS

- **Special thanks to Prof. Heda Samimi of the Computer Science Department for his financial and technical support.**
- **Dr. Mary Jo Geise for editing**