David Schulz & Andrew Ruswick

Lab 2: Forward and Inverse Kinematics

CS-4981-021

9/21/2021

**Lab 2: Forward and Inverse Kinematics**

**Abstract**

In this lab, we gained familiarity programming and working with the EV3 robots, implemented forward and inverse kinematics, and programmed the movement of a robotic arm to draw simple shapes. From our experiments, we concluded that our kinematics equations that we derived are correct, but many factors that cause the arm we built to move inaccurately, such as instability and friction, need to be fixed or minimized to improve the accuracy of the arm. Also, in the case of making straight lines, our algorithm could be improved to make the arm move its servos at the same time and more smoothly.

**Methods**

Forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters. The reverse process that computes the joint parameters that achieve a specified position of the end-effector is known as inverse kinematics.

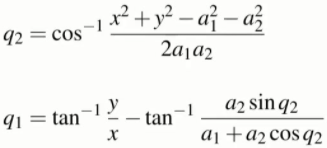
For our robot arm with two degrees of freedom, the forward kinematic equations are computed as

x = a1cos(q1) + a2cos(q1+q2)

y = a1sin(q1) + a2sin(q1+q2)

where a1 is the length of the first arm from its axle to the next arm’s axle, a2 is the length of the second arm from its axle to the point of the end-effector, q1 is the angle of the first servo with respect to the x axis and q2 is the angle of the second with respect to the angle of the first.

The inverse kinematic equation(s) are computed as



where the variables are the same as those in the forward kinematic equations.

**Results**

**Task 1**

Shoulder to elbow (a1): 13.5 cm

Elbow to pen (a2): 10.5 cm

When substituting the measured values for a1 and a2 into the forward kinematic equations, we get

x = 13.5cos(q1) + 10.5cos(q1+q2)

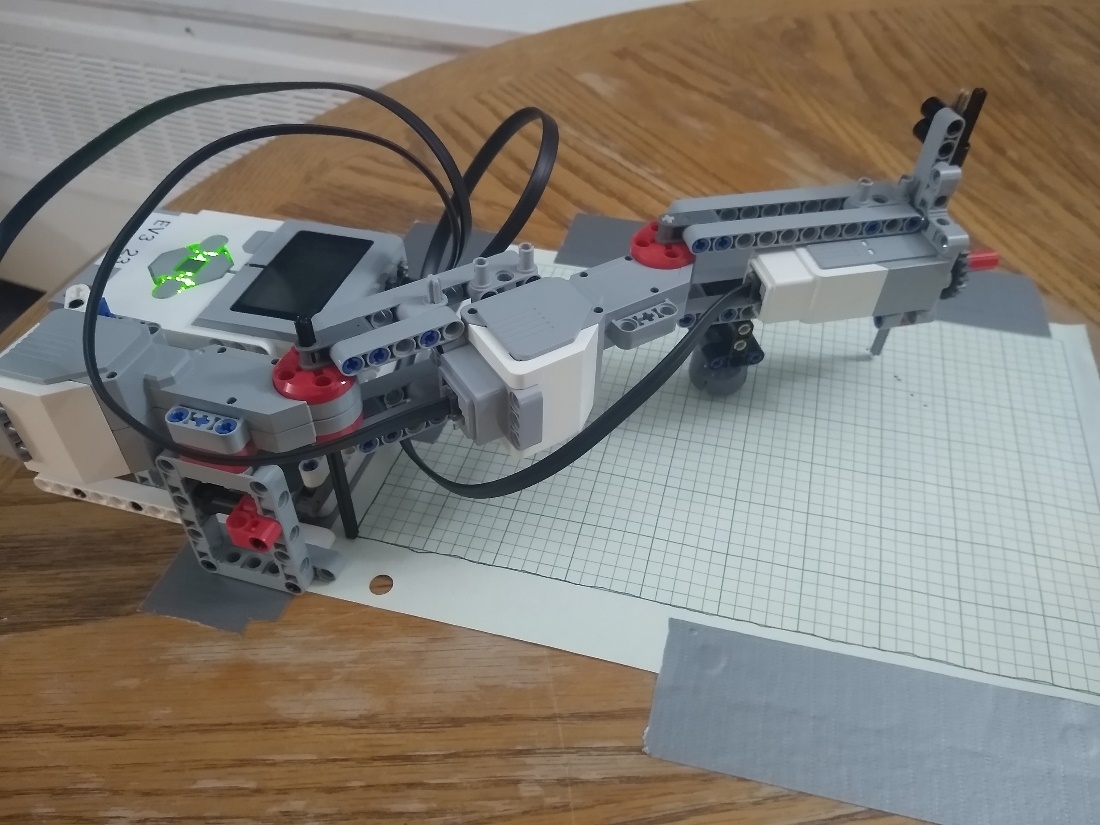
y = 13.5sin(q1) + 10.5sin(q1+q2)

1. Set 1
   1. q1=135 degrees q2=-20 degrees
   2. x = 13.5cos(135) + 10.5cos(135-20) = -13.98 cm
   3. y = 13.5sin(135) + 10.5sin(135-20) = 19.06 cm
   4. Trial Locations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (-12,19.4) | (-12.5,18.9) | (-13.2,18.4) | (-13.5,18.3) | (-15,17.1) |

* 1. Errors:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (1.98,0.34) | (1.48,-0.16) | (0.78,-0.66) | (0.48,-0.76) | (-1.02,-1.96) |



1. Set 2
   1. q1=100 degrees q2=90 degrees
   2. x = 13.5cos(100) + 10.5cos(100+90) = -12.68 cm
   3. y = 13.5sin(100) + 10.5sin(100+90) = 11.47 cm
   4. Trial Locations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (-12.3,11.2) | (-11.7,11.7) | (-11.5,11.6) | (-11.5,11.5) | (-10.9,11.9) |

* 1. Errors:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (0.38,-0.27) | (0.98,0.23) | (1.18,0.13) | (1.18,0.03) | (1.78,0.43) |

**Task 2**

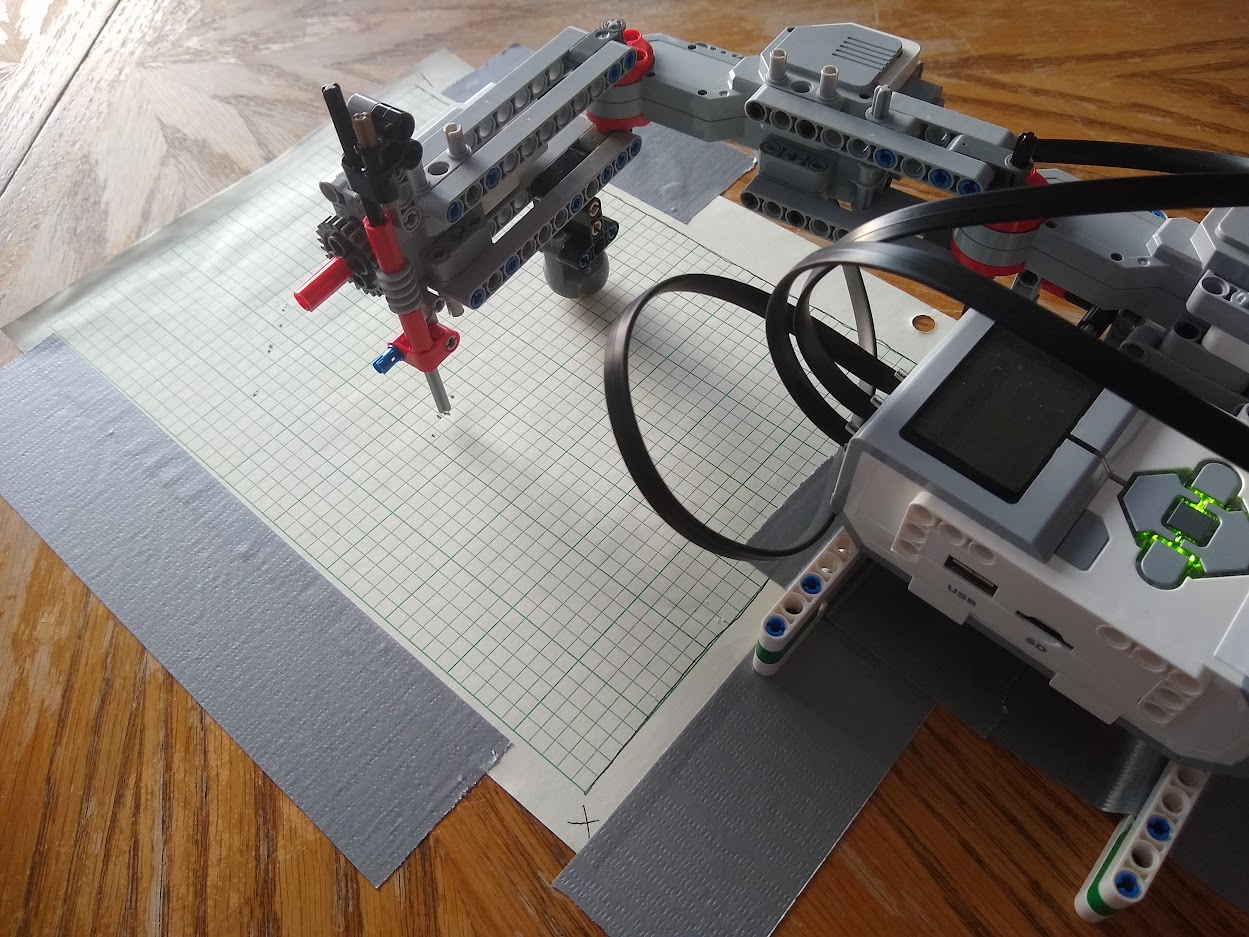
We then checked if the inverse kinematic equations were correct by using the coordinates we got from task 1 as our inputs for the equations and compared the calculated angles. We then tested the equations on our robot arm and measured the error.

(-13.98, 19.06)

* Negative q2 = -cos-1(((-13.98)2 + (19.06)2 - (13.5)2 - (10.5)2) / (2\*13.5\*10.5)) = -0.35 radians = -20.11 degrees
* q1 = tan-1(19.06/-13.98) - tan-1(10.5sin(-0.35) / (13.5 + 10.5cos(-0.35))) = 2.36 radians = 135 degrees
* Trial Location: (-12.8, 18.2)
* Trial Error: (1.18, -0.86)

(-12.68, 11.47)

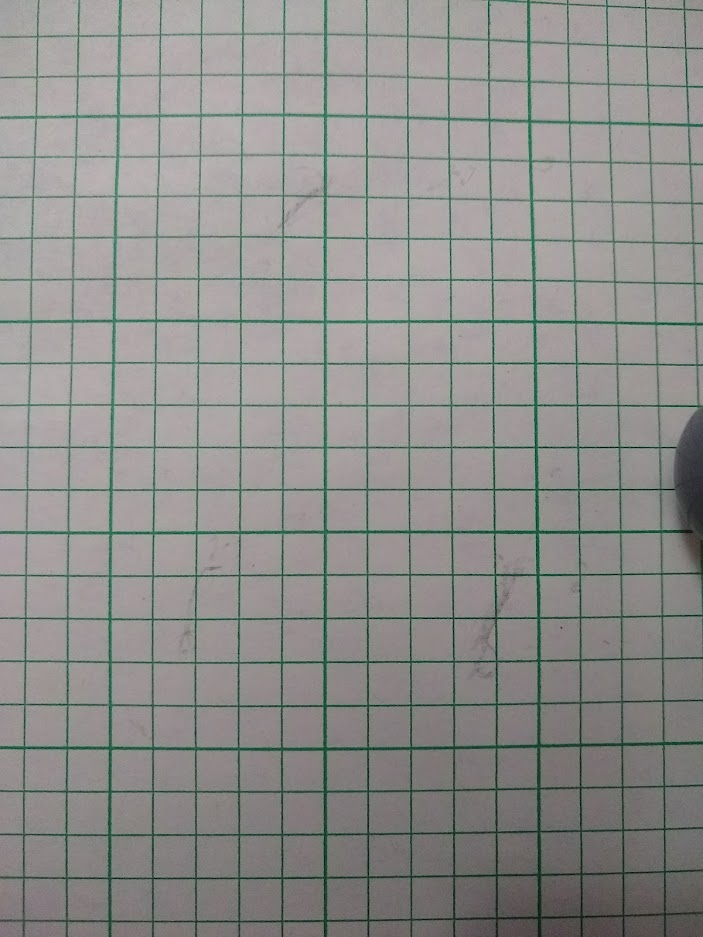
* Positive q2 = -cos-1(((-12.68)2 + (11.47)2 - (13.5)2 - (10.5)2) / (2\*13.5\*10.5)) = 1.57 radians = 90.03 degrees
* q1 = tan-1(11.47/-12.68) - tan-1(10.5sin(1.57) / (13.5 + 10.5cos(1.57))) = 1.74 radians = 99.98 degrees
* Trial Location: (-11.5, 11.4)
* Trial Error: (1.18, -0.07)



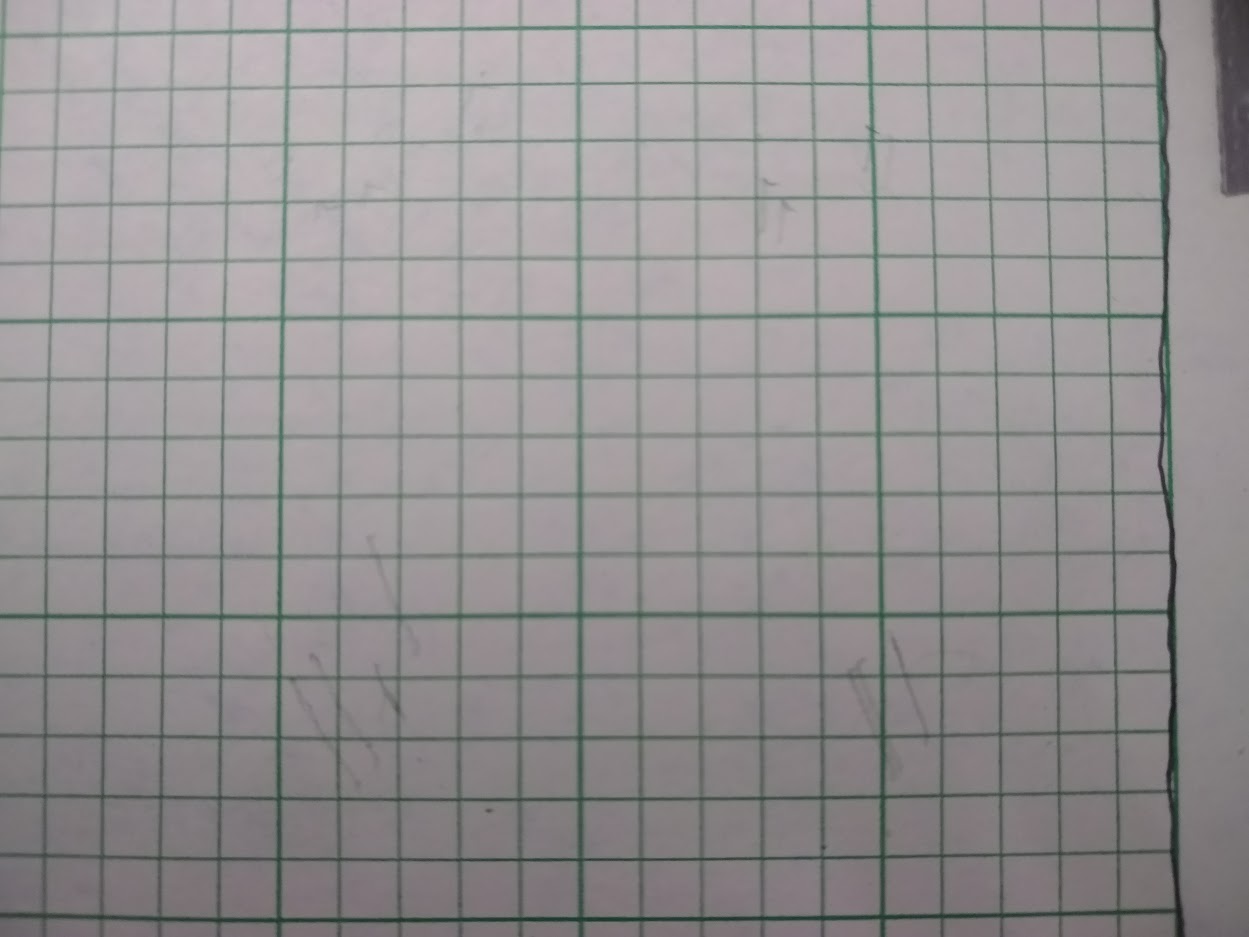
**Task 3**

For the rest of the lab, we taped some 0.9mm pencil lead to the end effector. Unfortunately, the marks it makes are not easily visible due to the small amount of force being applied by the arm. In task 3, the marks are at least visible enough to tell where the vertices of each shape are.

Triangle



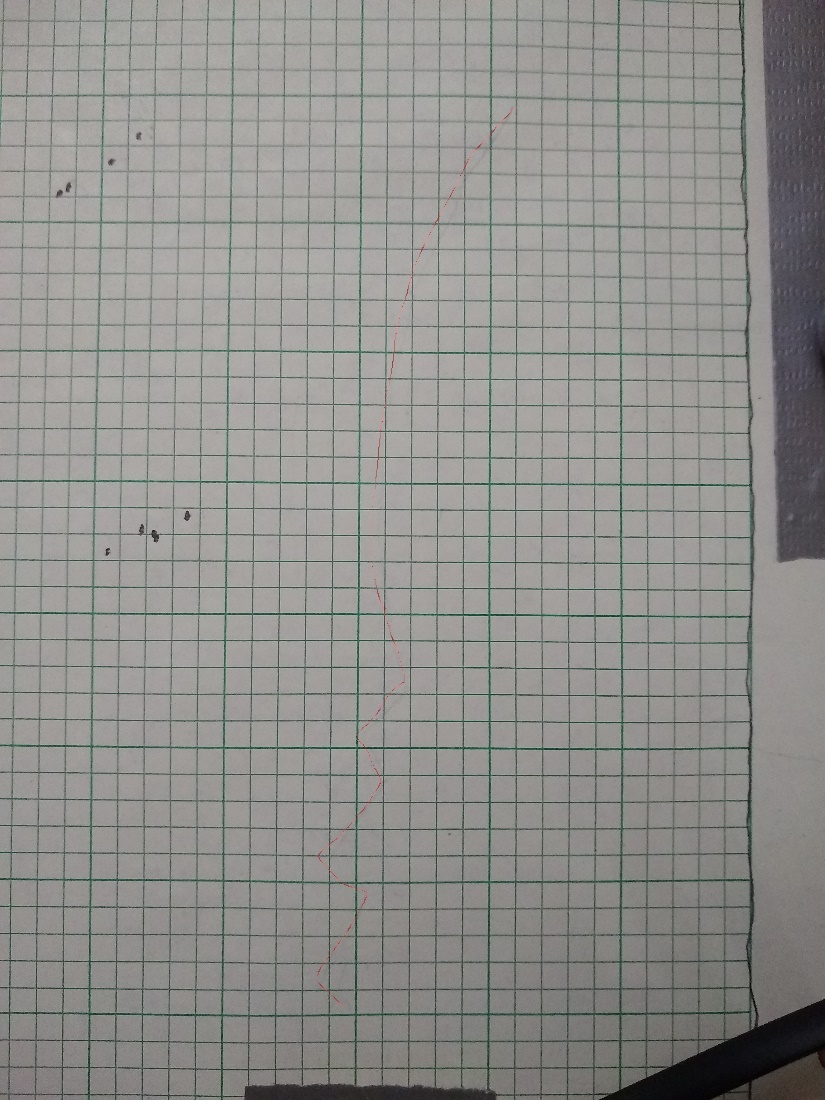
Square



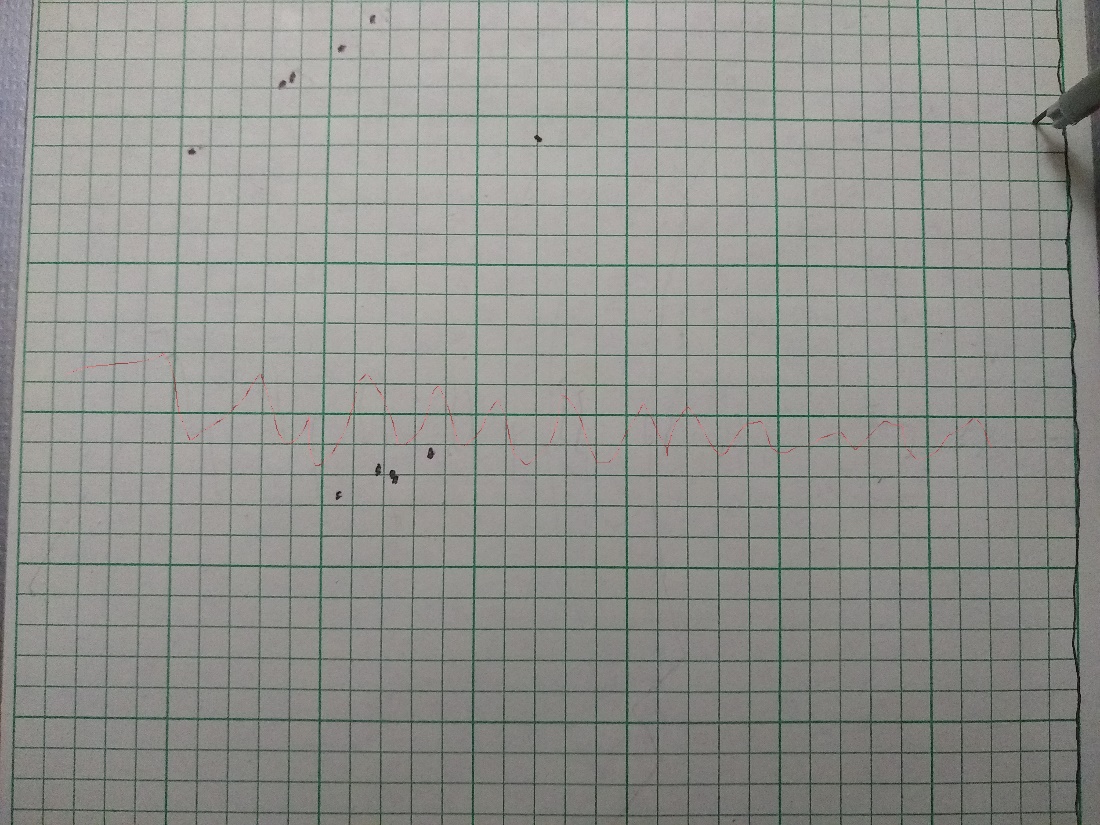
**Task 4**

Since maintaining the lead’s pressure on the paper while the arm was moving was extra difficult, we roughly traced the lines it made to clearly show the full paths. To make our lines, we created 10 points along the way from point A to point B that the arm had to move to before continuing along the path.

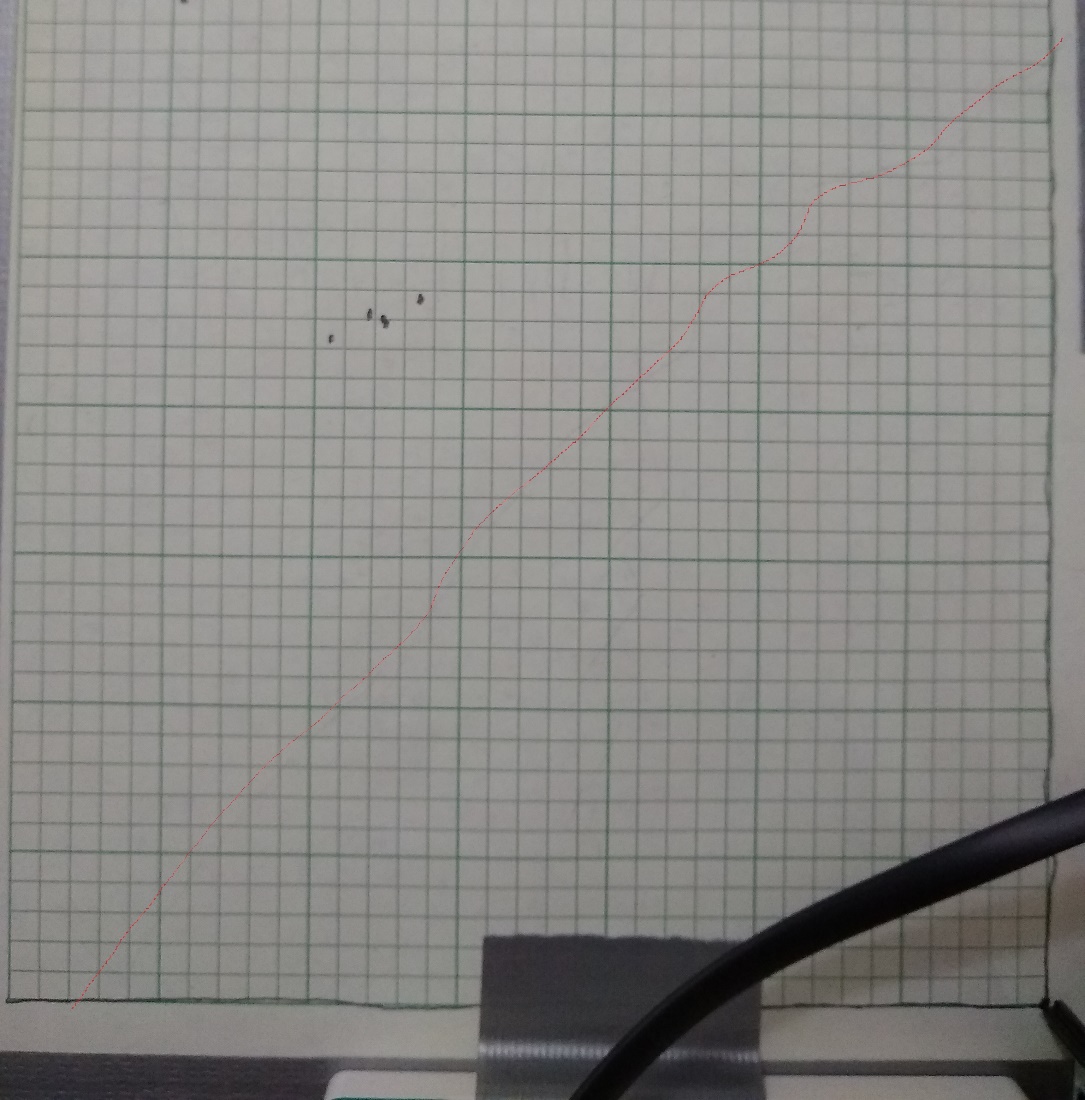
Vertical



Horizontal



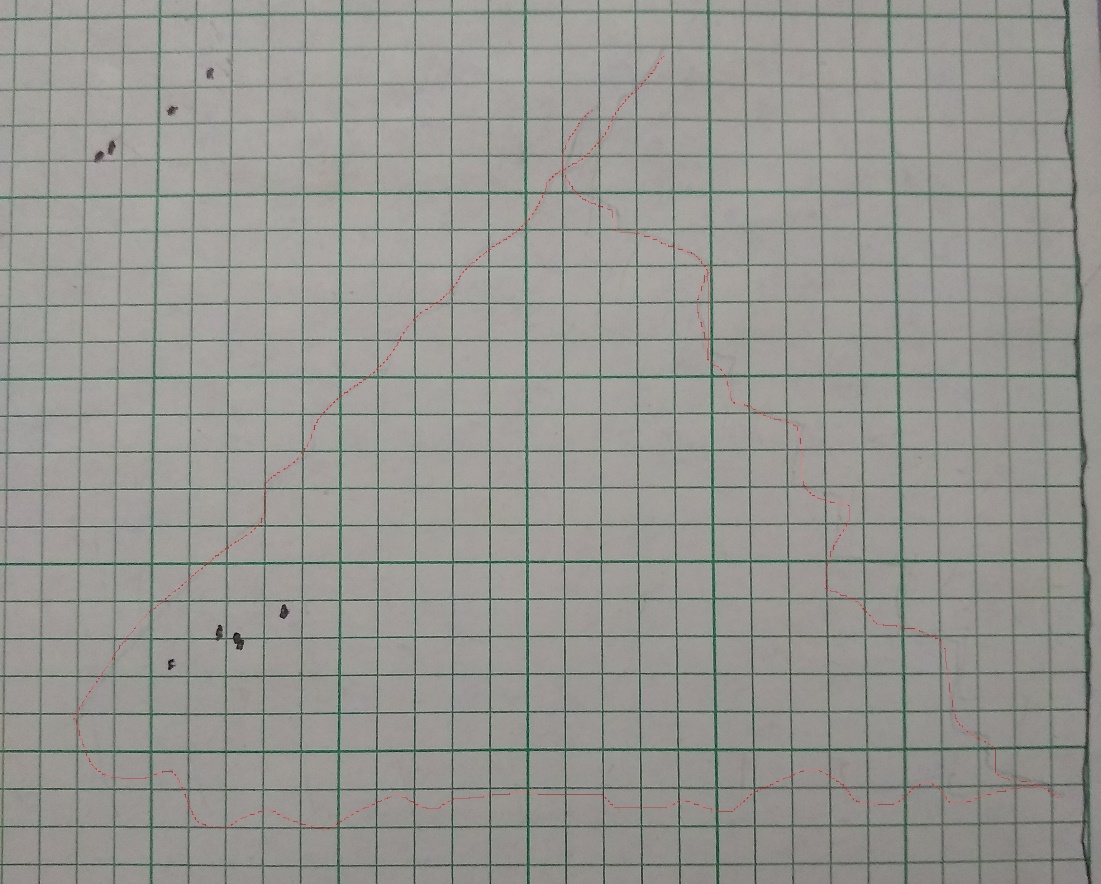
Diagonal



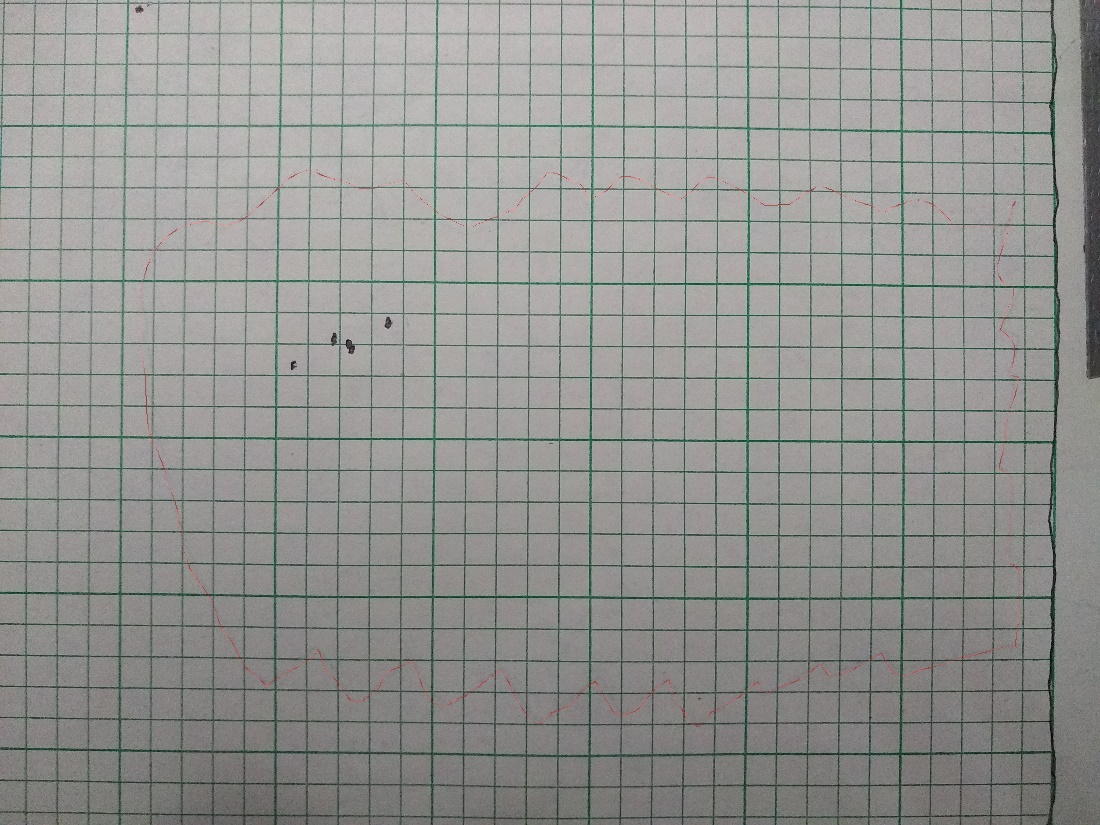
**Task 5**

The shapes with fully-drawn sides were traced as well for clarity. As we did in task 4, each side was made by putting 10 points along the path for the arm to get to.

Triangle



Square



**Discussion**

1. Why is drawing a straight line between two points non-trivial?
   1. Both servos have to be rotating at the same time to different angles at different speeds in order to maintain a straight line from one point to the other.
2. How did you deal with the hard constraints in your inverse kinematics?
   1. We limited the arm’s movement to the second quadrant, so all coordinates and angles we used in the lab were only in the negative x and positive y ranges.
3. For each desired target location, the analytical method produces two sets of angles. How did you decide which one to use?
   1. We chose whichever set was easier to get within the motors’ hard constraints. If both were possible, we just chose one at random.
4. Were the movements of the arm accurate? Why or why not?
   1. Not exactly. There are several different factors that cause the end-effector to not land perfectly on the desired locations, such as imperfect encoder sensors in the motors, weak structural stability of the arm, and inconsistent starting position of the arm.
5. What was the biggest difficulty in working with the robot arm?
   1. The biggest struggle for us was understanding how the orientation of the arm corresponded to the angles of the motors, with the issue stemming from misunderstanding why the starting position of the arm was set to the angles that we were told.