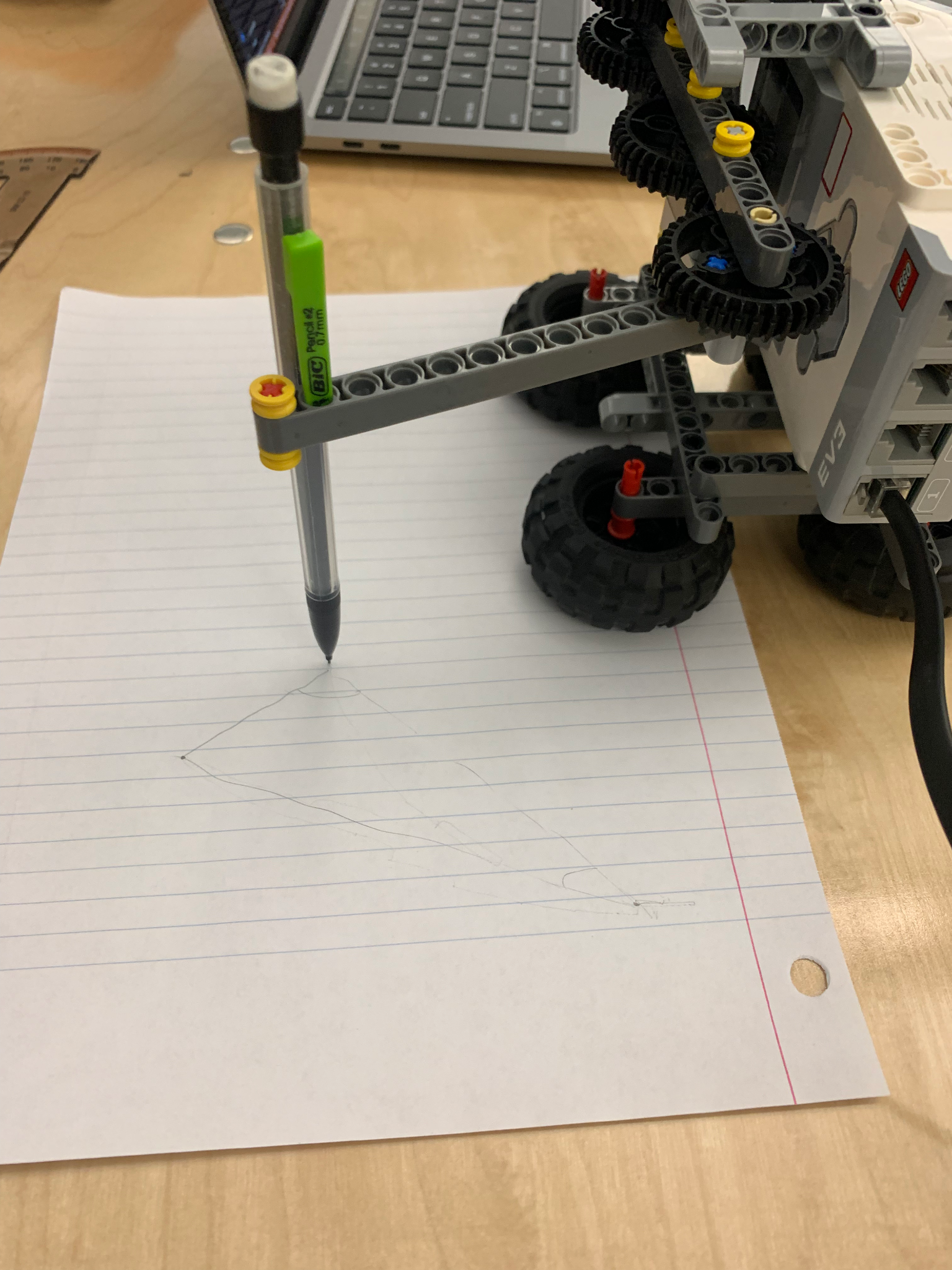
# LAB 2: Motor Control, Robot Arm Kinematics and Path Planning

Group: 2

Members: Daniel Merryweather (1580298), Qasim Khawaja (1571051),

Nathan Pelletier (1480408)

## 2-DOF Robotic Arm



**a) Propose a 2-DOF Arm LEGO design and build it. Take into account the following requirements:**

* **Explain the design of the robot**

The design of our 2-DOF arm was one out of simplicity, trying to keep both motors in a central location to control our arm. The unique thing about our design is that the second arm is geared to be rotationally independent of the first, so rotating the first linkage will not rotate the second linkage/end effector.

* **Are there different types of linkages that can provide the same motion? Consider the pros/cons of different motor mountings and arm linkage constructions (see lecture notes).**

The robot could also have been designed to make rotation dependent on the first arm by adding another gear; this would have been a more traditional 2-DOF arm robot. The only advantage of gearing the arm to have independent rotation like we did was to simplify the math for positioning as we would not need to compensate for the first linkage rotation. The motors were also mounted beside each other around the origin of motion for stability, another strategy for this could have been mounting a motor at the second joint instead of gearing it to it, this would make it more stable and simple. The main disadvantage of mounting the motor at the axis would be the fact that the first linkage would need to support the motor and the rest of the arm requiring very thoughtful design to be stable as inertia would be more larger.

**b) Determine the workspace of your robot. Include a figure and dimensions of your workspace in the report.**

**Dimensions**

Each link is 96mm long.

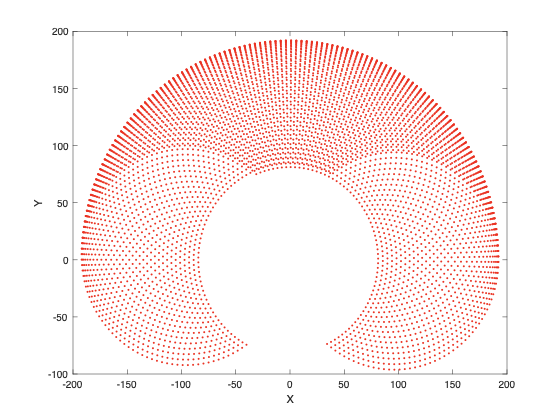
The maximum distance reachable from the center of the robot is 192mm.

The first linkage can rotate about the X-axis for a maximum of 180.

The second linkage can rotate about its X-axis for a maximum of 270.

Code to plot workspace

| a1 = 96; a2 = 96; DH(1) = Link([0 0 a1 0]); DH(2) = Link([0 0 a2 0]); th1 = (0:0.05:pi); th2 = (degtorad(-130):0.05:degtorad(130)); q = {th1,th2, 0}; plotworkspace(DH,q) |
| --- |

****

Mohammad Al-Fetyani (2021). plotworkspace - Plot workspace of n-DOF planar robot (https://www.mathworks.com/matlabcentral/fileexchange/71136-plotworkspace-plot-workspace-of-n-dof-planar-robot), MATLAB Central File Exchange. Retrieved October 6, 2021.

P.I. Corke, “Robotics, Vision & Control”, Springer 2017, ISBN 978-3-319-54413-7. [[bibtex](http://www.petercorke.com/RVC/rvc.bib)]

## Forward Kinematics

**Define an initial position for your robot, such that the robot base is located on (0,0) Cartesian coordinates.**

The initial position of the robot arm is at (192,0) such that the arm is parallel to the x-axis.

**Measure the accuracy/repeatability of these movements to include in your report.**

We marked 3 points on our workspace to repeatably test our angle and distance function for forward kinematics.



Distance Measurements were off by 1cm on average.

Angle Measurements were off by 1-2 degrees on average.

### Inverse Kinematics

**Explain, compare, and analyze both approaches in your report.**

**Newton's Method:**

Newton's method uses forward kinematics over decreasing distances as it hones in on a desired x, y coordinate.

It is harder to set up and to compute than the analytical method but it can be used for scenarios that the analytical method cannot. Newton's method can be used in scenarios where there are more joints on the robot than degrees of freedom of the end effector without cheating and locking a joint. Newton’s method barely becomes more complicated with each added joint, allowing us to program a snake like machine similarly to how we would program an arm.

As the last statement suggests Newton’s method is ideal for complicated robots in complicated environments where speed is not a factor.

**Analytical solution:**

The analytical solution to inverse kinematics requires trigonometry based on the current joint configuration and the hypotenuse formed by the given coordinates.

The analytical solution is easy to set up and it is easy for the machine to compute. Given that there are a few limitations. Even in this two joint, two degrees of freedom system there are two solutions to pick from. For each joint added the trigonometry becomes harder to solve, until the system contains more joints than degrees of freedom in rotation and position. In that case the analytical solution offers infinite solutions unless an unnecessary joint is taken out of the equation.

The analytical solution is ideal for simple robots like our two dimensional drawing robot, for robots that require fast operation and for robots that have small CPU’s.