Project 1 Resource Document
EECS 498: Green Team

2014 Maize Team: Crane Robot

https://wiki2.eecs.umich.edu/hrb/images/3/33/BrainstormingPresentation_14Maize.pdf

This link refers to the Crane Robot from the 2014 Maize team. They used a linkage slider mechanism. They also used a thread to pull and push the crane, which always stays in the air. The base of the robot moves to and fro on the guided rails. The ends of the guided rails have markers to protect the robot from moving outside the arena. This design has a discrete x and y motion. The usage of linkage slider is popular in terms of dynamics.

2014 Blue Team: Rotational Robot

https://wiki2.eecs.umich.edu/hrb/images/e/e5/Brainstorm_Presentation_14Blue.pdf
https://wiki2.eecs.umich.edu/hrb/index.php/File:BlueTeam-P1-2014.pdf
https://wiki2.eecs.umich.edu/hrb/images/a/aa/Construction_Document_14Blue.pdf

These documents detail the 2014 Blue Team's design for this project, which used two wheel arms that could be lifted and rotated around a main platform containing the laser pointer without changing the laser's orientation. We considered a similar design for one of our alternatives. These documents are important because they provided us with a design idea, but also because they showed what the obstacles and challenges to implementing the design are. The construction brought to our attention the issues of straight-line drift and calibration of the lifting arms. The description of the P-Day performance in the final report caused us to realize how much the system relies on its mechanisms functioning perfectly, and how it does not allow for much software correction.

2016 Green Team: Extendable Arm Robot

https://wiki2.eecs.umich.edu/hrb/images/7/79/FinalReport_%281%29.pdf

This document explains the 2016 Green Team's robot for this project. It gave us the idea of keeping the main body of the robot in the staging area and simply extending an arm with the tag on it. This basic design influenced two of our three design alternatives. This document also shows how a rack and pinion system can be manufactured and used in the context of this class. This suggests that our own rack and pinion robot design is a plausible solution to the problem of Project 1.

Arduino Robot

http://robotics.stackexchange.com/questions/10019/should-i-use-gyro-or-encoders-for-robot-moving-in-straight-line

A flaw in the 2014 Blue Team's design is that it relies on the assumption that the robot will drive in a straight line at all times. However, as their construction document shows, it was subject to a certain amount of straight-line drift. This Stack Overflow link lists a number of ways to sense whether the motion of a robot is along a straight line, including servo encoders, gyroscopes, magnetometers, and fusion sensors.

Omni Wheel Mobile Robot
https://www.youtube.com/watch?v=5vJCucpVdX0

For the design where the crane pushes the tag on a wheeled body, we needed a way to allow the wheels to slide in both the x- and y-axes. This link shows a robot that uses omni-wheels, which are wheels that have smaller wheels around the rim, allowing the wheel to slide parallel to its axis of rotation. The robot in the video moves in many different directions but does not change orientation, leading us to conclude that our wheeled crane design is plausible. It also led us to consider moving the servos off from the crane arm and putting all of the control on the wheeled body, using the crane as a passive element to maintain orientation.

Kalman Filter
https://wiki2.eecs.umich.edu/hrb/index.php?title=File:Kalman-Filter-Intro.pdf

Using a Kalman filter would be one low-overhead way to help predict the robot's current state. This will help us to use low computing power while also giving us a mostly accurate way in a linear system of representing our robot's position, orientation, and allow us to predict which action will lead to the most success based on our last perceived state and measurement. The trade off with this method is that we sacrifice some accuracy for speed. A unscented kalman filter would give us a middle ground between the speed of the kalman filter vs the high performance of a particle filter.

Particle Filter
https://youtu.be/aUkBa1zMKv4

Using a particle filter would be a high-accuracy exhaustive method that will help us predict the robot's current state. This will give us similar results to a Kalman filter except this method can be used more reliably with non-linear and noise filled systems. This is more computationally expensive, but performs with more accuracy.

Pid Controller
https://en.wikipedia.org/wiki/PID_controller
https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf

A PID controller is essentially the software which we will be building. We will make a program which, given a previous state and a measurement, can calculate the error of each state and thus likelihood of each state. We will then choose the state with the lowest error as our assumed state. This will help the robot to make calculated movements even without direct human intervention. These sources gave us background on what a PID controller is, how to implement one, and common sources of error inside PID controllers.