Fishing Robot

Team 13

Alysa Kataoka, Cheryl Lee, Harry Sandstrom, Jacob Sayono, Jingran Meng, Jinyoung Kim

Presentation Overview

1. Introduction
2. Design
3. CAD
4. Physical Parts
5. Forward Kinematics
6. Inverse Kinematics
7. Control
8. Future Work

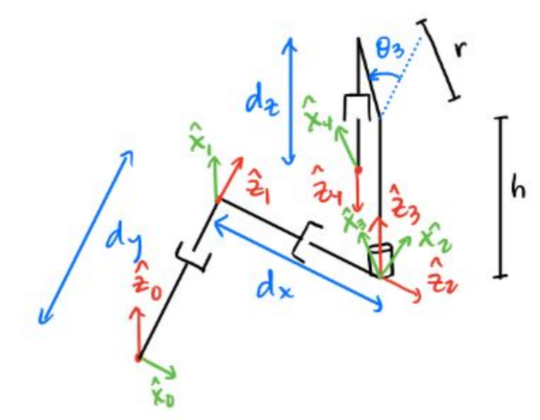
Introduction

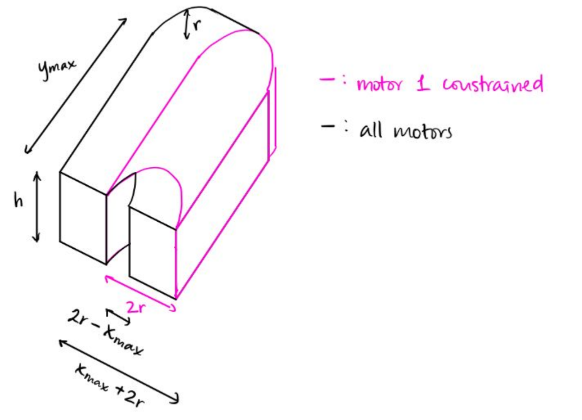
Design

Purpose: Get target located in 3D space using magnetic end effector

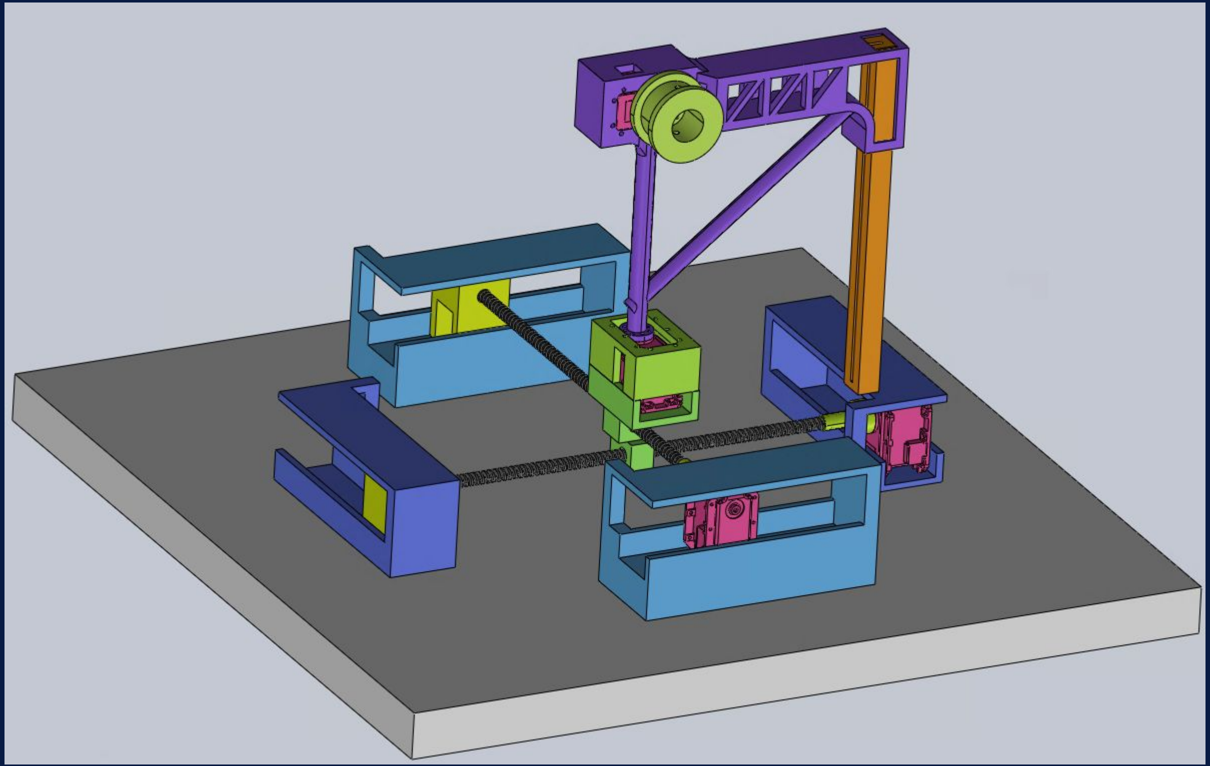
Model DOF: 4 (PPRP) → Extra DOF to reach additional space

Goal DOF: 3 (x, y, z)

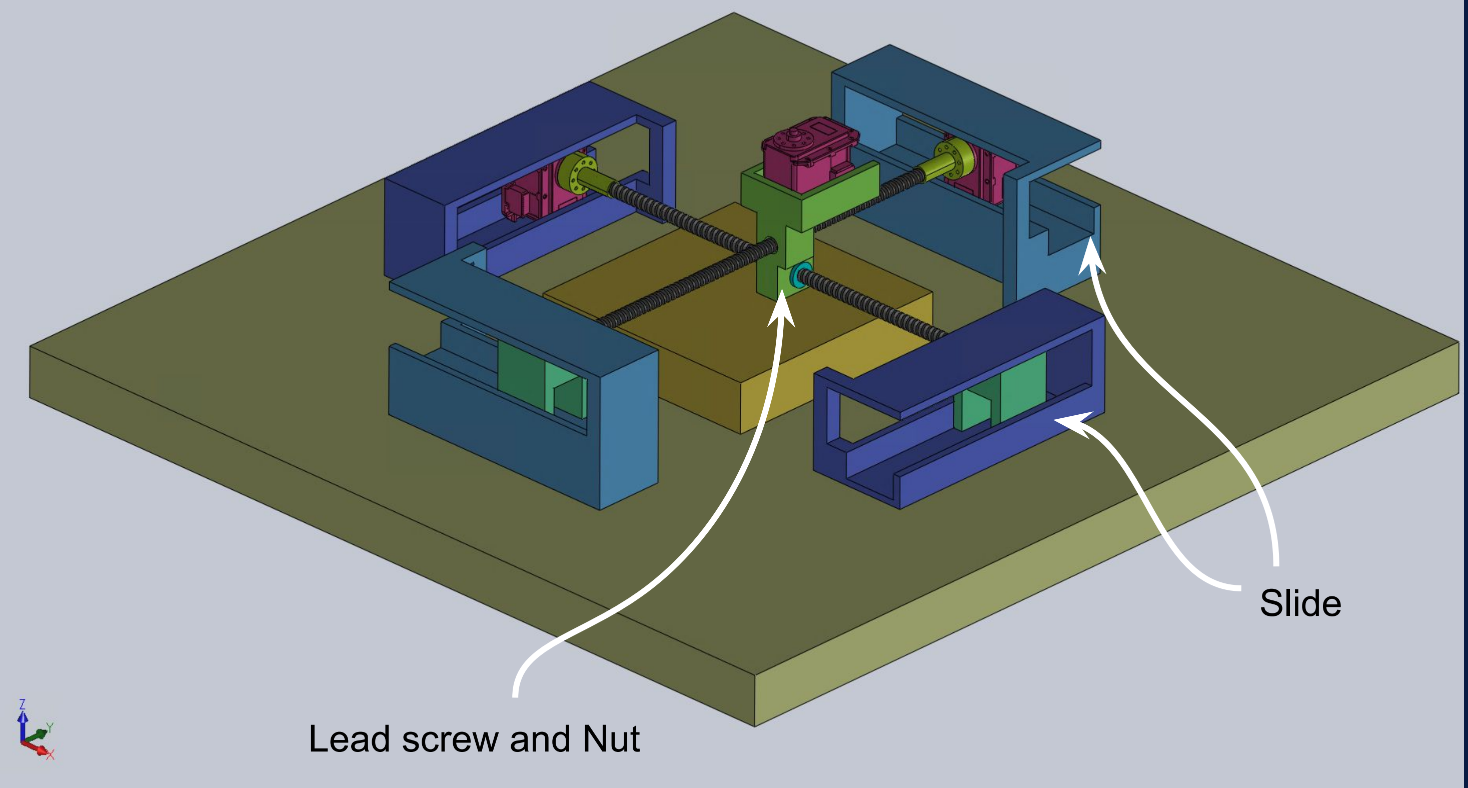




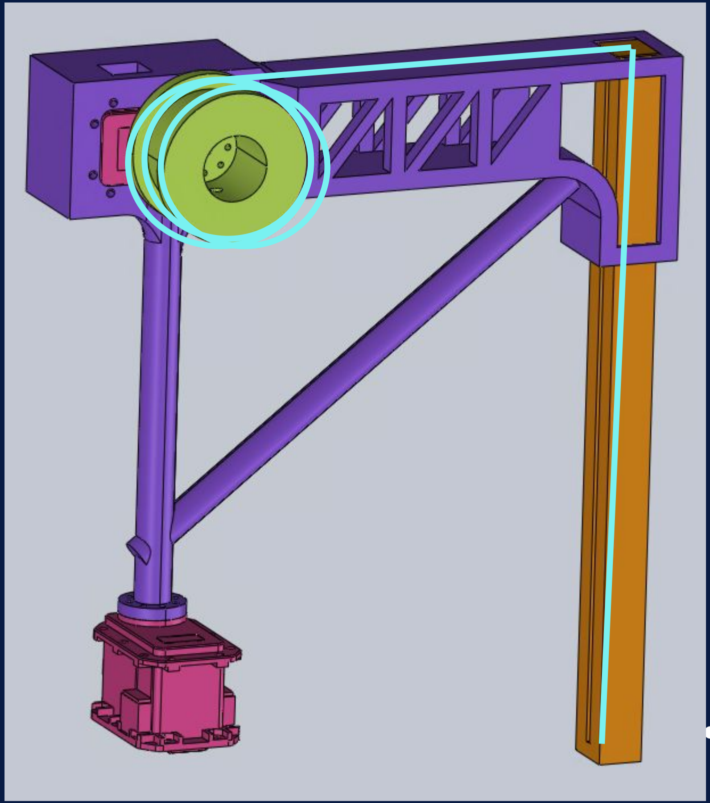
Full CAD



CAD Part 1 (2 Prismatic)

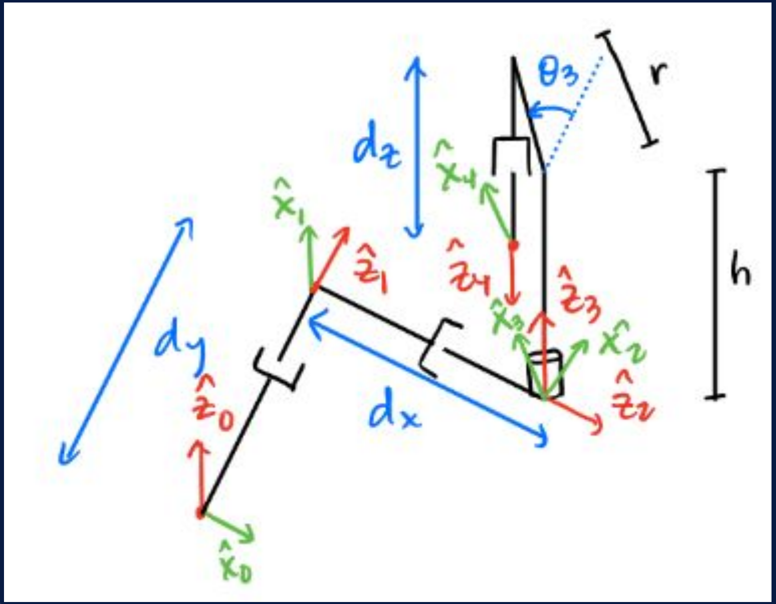


CAD Part 2 (1 Rotational + 1 Prismatic)



Physical Parts

Forward Kinematics - DH Parameters



i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	-90°	d_y	-90°
2	0	-90°	d_x	-90°
3	0	-90°	0	θ_3
4	r	180°	$h - d_z$	0

Forward Kinematics

$$ {}^0_1T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_y \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^1_2T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_x \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^2_3T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^3_4T = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & d_z - h \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Forward Kinematics

$$
{}^{0}_{4}T = \begin{bmatrix} -\sin(\theta_3) & \cos(\theta_3) & 0 & d_x - r\sin(\theta_3) \\ \cos(\theta_3) & \sin(\theta_3) & 0 & d_y + r\cos(\theta_3) \\ 0 & 0 & -1 & d_z - h \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Inverse Kinematics

$$d_x = \begin{cases} p_x - r & if\ p_x > 0 \\ p_x + r & if\ p_x < 0 \end{cases}$$

$$\theta_3 = \sin^{-1}\left(\frac{d_x - p_x}{r}\right)$$

$$d_y = \begin{cases} p_y - r & if\ \theta_3 = 0 \\ p_y - \frac{d_x - p_x}{\tan(\theta_3)} & otherwise \end{cases}$$

$$d_z = p_z - h$$

2 Cases:

Case 1: If distance from end effector to target is **less** than range limit, r, constrain dx = 0

Case 2: If distance from end effector to target is **more** than range limit, r, then include dx equation

Control

Prismatic Joints:
Motors 1, 2, & 4
(extended wheel mode)

Revolute Joints:
Motor 3
(wheel mode)

Motor Control:
MATLAB

Show demo! :D

Future Work

01

Finish physical assembly of printed parts with the motors

02

Conversions from prismatic distance to number of revolutions (rod pitch)

03

Velocity control (slow down to prevent jerkiness)

04

Creating simulation using MATLAB

Thank you!