Path Following Fetch

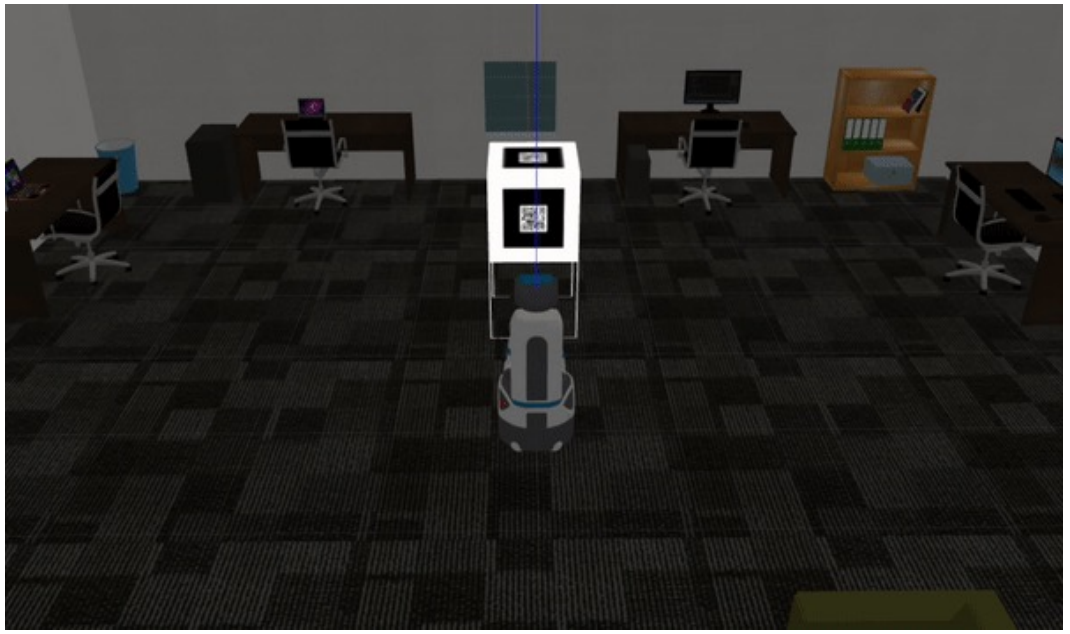
Group 2

Divjot Babra
Lee Madden
Zhifeng Huang

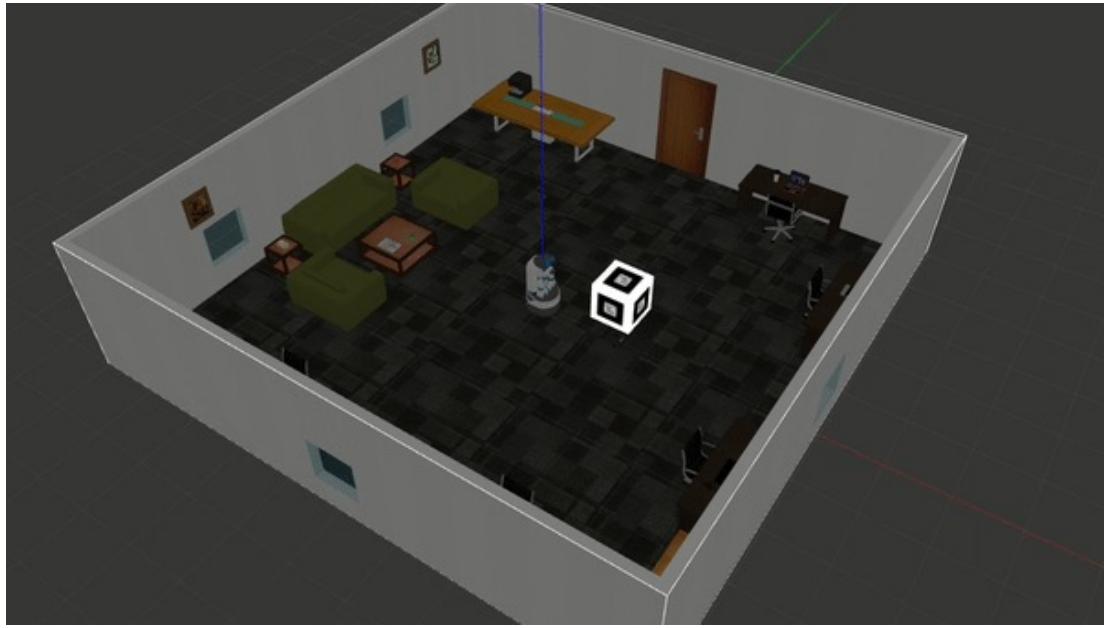
Sensors and Control for Mechatronic Systems

Project Overview

- Fetch robot following the path of a guider robot within Gazebo
- Guider robot is represented by a Turtlebot Waffle which is manually controlled
- Developed path following algorithm to achieve following motion
- Camera video data is processed by a visual tracker which provides information to the algorithm to directly control Fetch



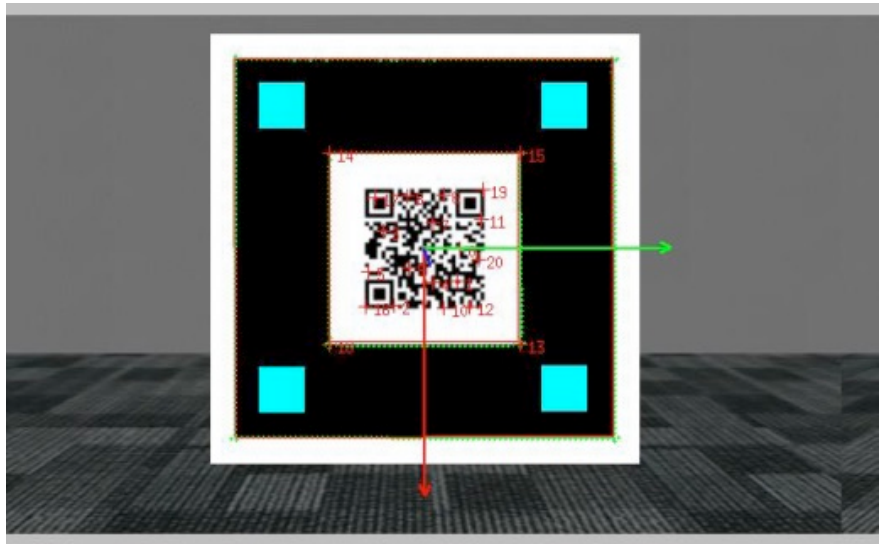
Simulation Setup



- Gazebo map includes a small office, multiple pieces of furniture and several office items
- Marker image was overlayed onto a cube model using Blender
- Marker cube was then attached to Turtlebot model as a link component
- All models (including robots) were configured using a World file for importing into Gazebo
- Launch file developed for automatically starting simulator and several ROS nodes

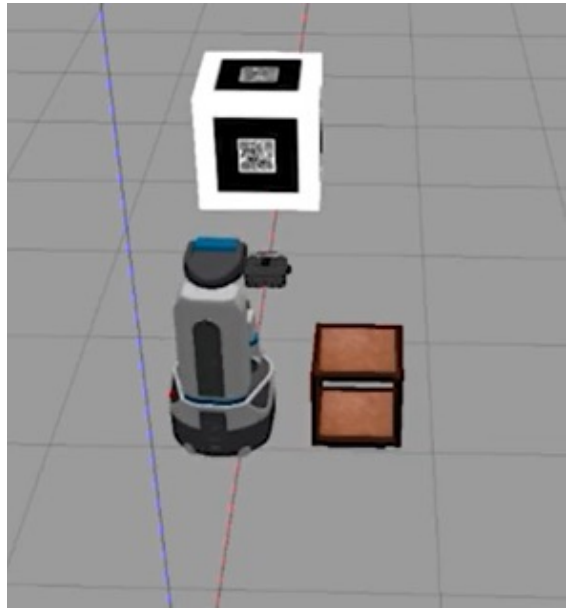
Image Processing

- Used Fetch's RGBD camera
- ViSP – Visual Servoing Platform: (visp_auto_tracker)
- Visual tracking and servoing of visual features
- Access to states and location data from the QR Code



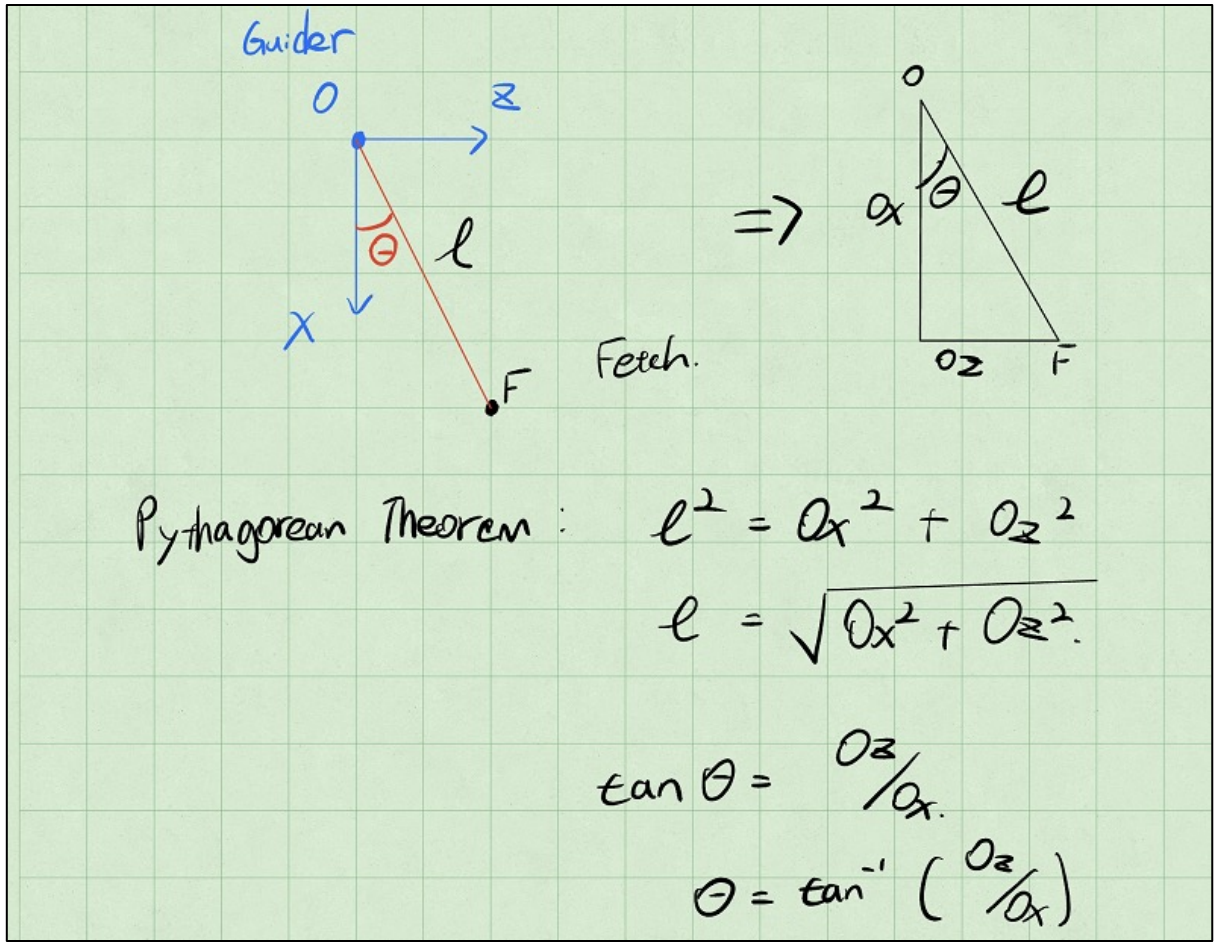
Obstacle Avoidance

- Used the Fetch's base scanner – laser range finder
- Obtained 220 degrees of ranges
- Found the closest range to avoid obstacle



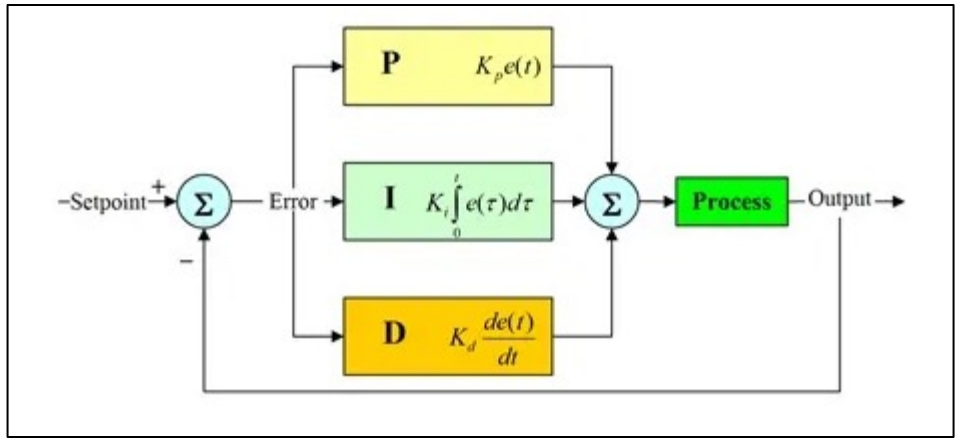
Path Following

- Target Position
  - Too far, move closer
  - Too close, move away
  - At the left, turn left
  - At the right, turn right
- Pythagoras's Theorem
  - Perpendicular Distance
  - Bearing



PID Control

- Proportional Controller (P)
  - Linear constants between input and output
- Integral Controller (I)
  - Address steady state error
- Derivative Controller (D)
  - Prevent overshooting



Velocity = P x Error + D x Derivative + I x Integral

- *Error is the difference between current position and target position*
- *Derivative is the difference between last error and current error*
- *Integral is the modulated error*