EECS 467
Autonomous Robots

Goalkeeper

Adam Dziedzic, Jingliang Ren, Ryan Wunderly, Jonas Hirshland
{dziedada, meow, rywunder, jhirshey}@umich.edu

Project Inspiration and Overview:
Goalkeeper is an autonomous robot that senses incoming balls and blocks them from passing. It was inspired by goalies in various sports, but mainly soccer. We demonstrate what ability is required in order to react to a human rolling a ball in the robot's direction.

FYI - How to Interact with the Robot:
Put a ball in motion towards the robot and it should move to block the ball if in range.

Arm Design

The arm mimics a human arm playing air hockey, projected to a two-dimensional space. We first used Solidworks to mock up the configuration of Rexarm parts, and then designed a custom end effector and 3D printed the end effector. The arm uses one AX-12+ and two Dynamixel MX-28 servos to move.

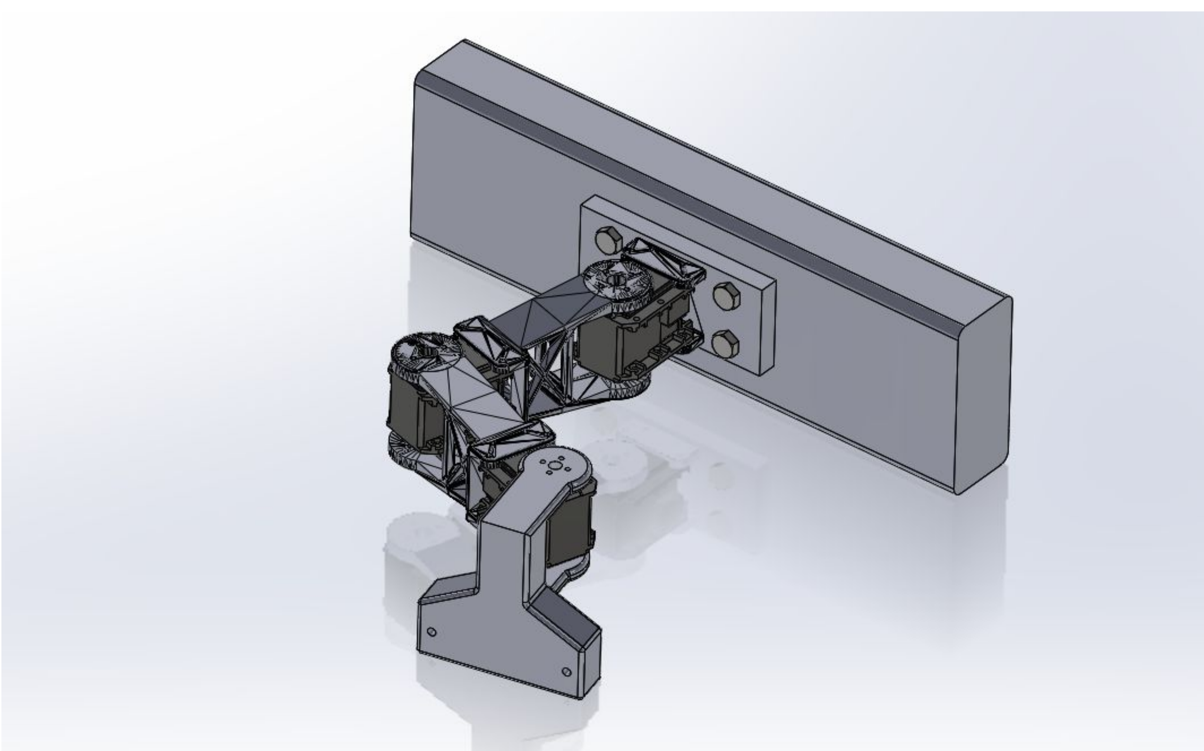
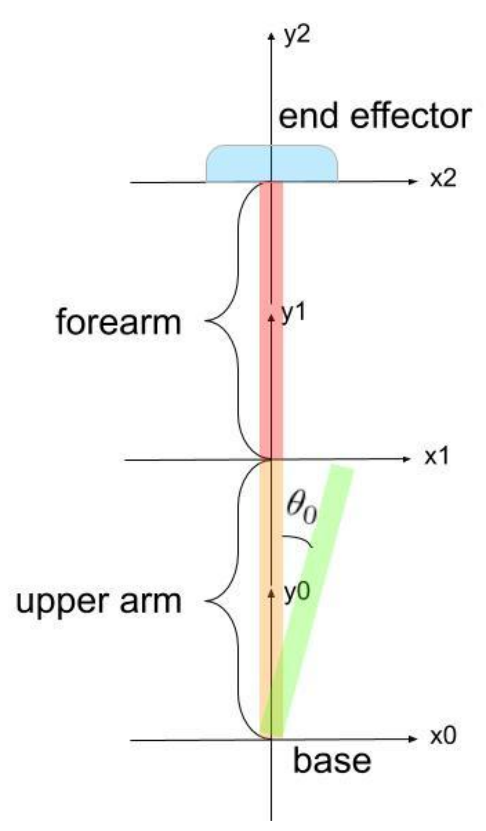


Figure 1: A SolidWorks Assembly file of the robot arm.

Kinematics (Forward & Reverse)

The movement of the arm is controlled by three servos. By giving three angles $(\theta_0, \theta_1, \theta_2)$ to three servos, the arm will move to a certain position.

Inverse kinematics can calculate the angles given a target position in the frame of base system. Forward kinematics is mainly used for debugging. It can calculate the coordinates of the end effector in the frame of base system by giving angles of the servos.



$$p^0 = A_0(\theta_0)A_1(\theta_1)p^2$$

Figure 2: The Arms coordinate frame and homogenous transformation.

Vision System

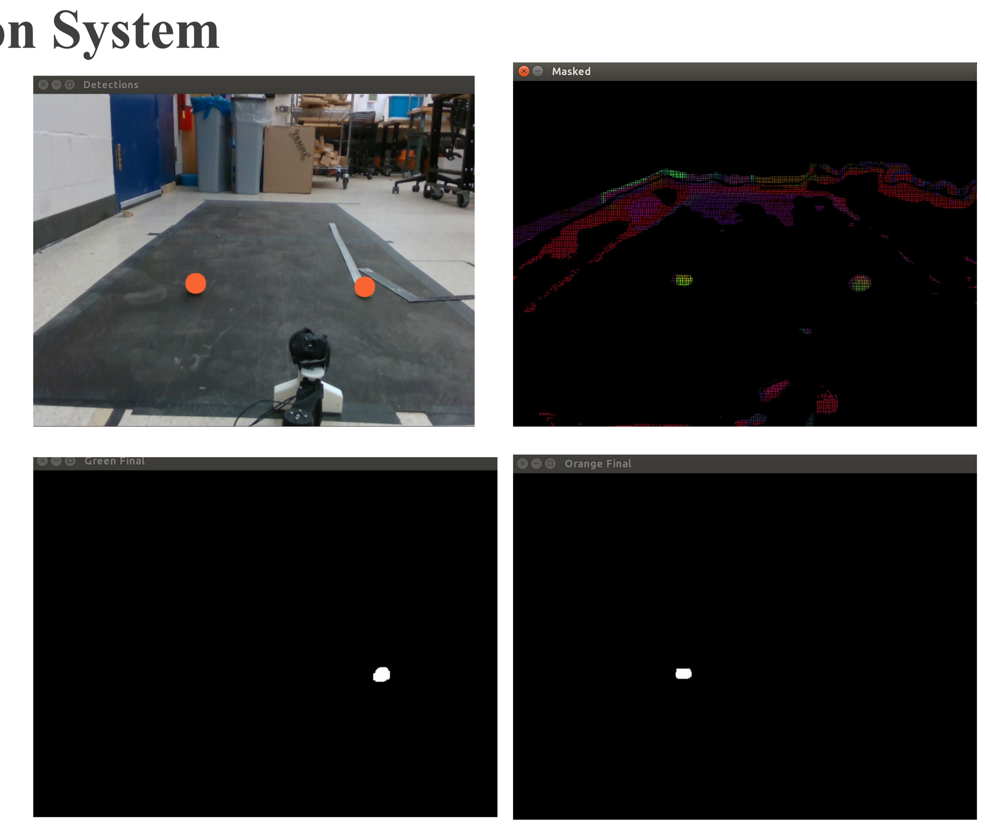


Figure 3: Visualizes the point cloud detection in the top right, the ball masks in the bottom frames, and the full detections in the top left frame.

We use Intel Realsense D435 RGBD cameras for rgb and depth images as well as point clouds of the environment. We use this for ball detection which proceeds as follows.

1. Convert Point Cloud from Camera to World Space
2. Convert RGB image to HSV image
3. Mask HSV image using x, y, and z of points
4. Color Segmentation of HSV Image
5. Extract point clouds for each contiguous color segment
6. Perform RANSAC sphere fitting on point clouds to purge false positives
7. Return point cloud centroids labeled by color as detections

The ball detection takes 14 milliseconds on the Lenovo P51 we use for testing, and the point cloud extraction and transformation takes 7 milliseconds, therefore resulting in a total detection latency of 21 milliseconds allowing us to process each camera stream at 30 frames per second.

Tracking

The robot predicts the ball's trajectory and acts to block the predicted path. The robot deals with noisy vision data by keeping an internal list of seen balls. If a detection can be matched to a ball (L2 norm), then the odds of the ball is incremented. If no detections are matched to a ball, then the odds go down. We remove a ball from the list once its odds drop below a threshold. This allows for a robust ball state estimates based on noisy vision data.

The trajectory and position of a ball is predicted by matching a line of best fit (least squares regression) to its past five detections. It's future trajectory is predicted to fall along the line of best fit.

We need to correspond the incoming detections to the existing balls, and we store the probability that the ball exists as log-odds.

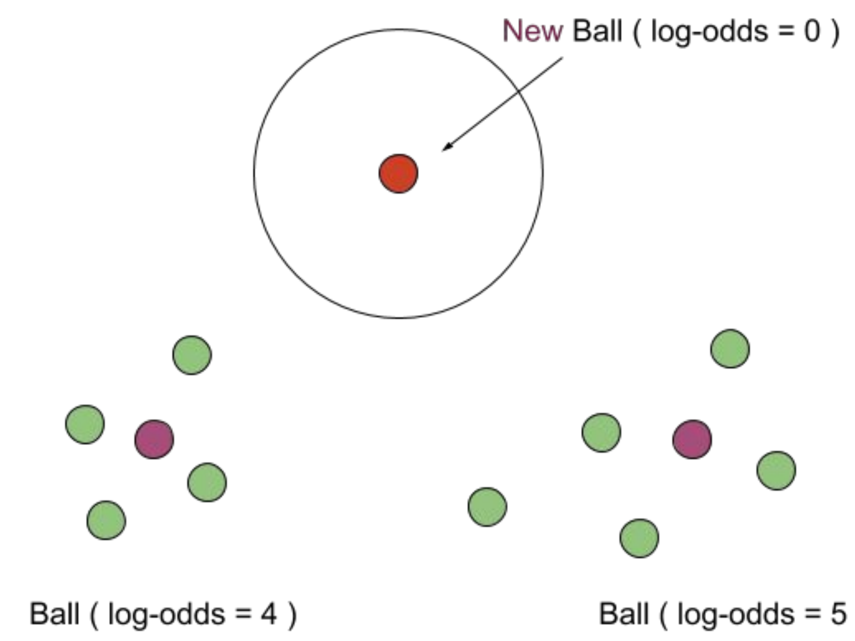


Figure 4: Ball objects in our state representation are depicted as purple dots, the detection history of that are the green dots that have been corresponded to that ball. The red ball is a new ball because it is not close enough to be corresponded to any of the existing balls. The log odds are also given.

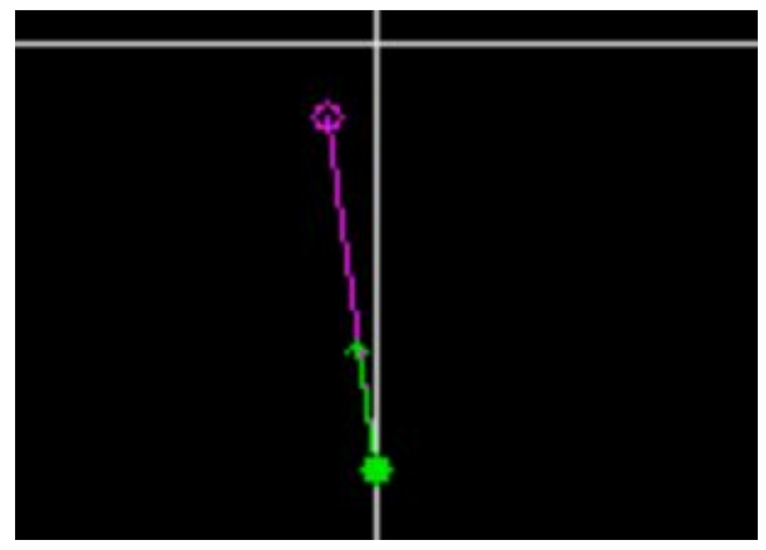


Figure 5: This is the visualization tool used to see what the arm planners world state representation looks like. The purple line is the predicted path to reach the arm. The green arrow is the velocity vector.

Control System:

The central planner predicts whether tracked balls will end up passing through the radius around the base of the arm where the arm can reach. For balls that will pass through the radius, where they will pass through it is computed and when they will pass through. The location where the first ball penetrates the radius of the arm is then sent to the arm controller to move the end effector to that position. The end effector is always at a 180 degree angle to the main direction balls come from to maximize the chance of blocking successfully.

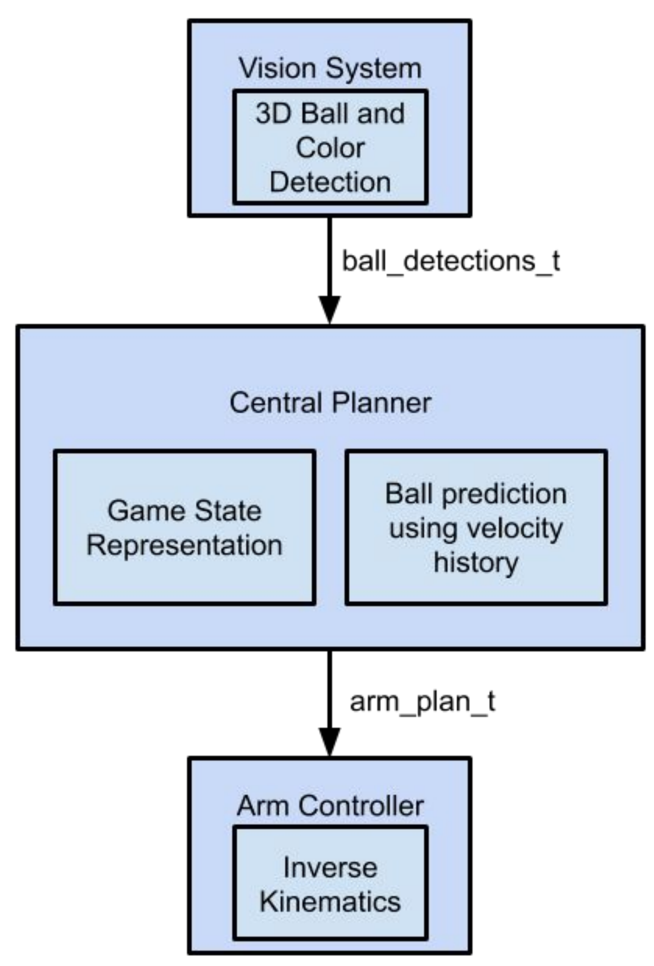


Figure 6: Depicts the communication between the logical units of our control system, and the message types between them.