PRELIMINARY DESIGN REPORT

Assistive Robots | Bobda Bots

Jonathan Feitel, Dylan Falzone
Jonathanfeitel@ufl.edu, Dylan.falzone@ufl.edu

TABLE OF CONTENTS

1. Abstract

The Assistive Robot is an autonomous system capable of omni-directional movement through an environment, locational and visual sensing, as well as precise and controlled object manipulation.

Figure 1. Fetch robot front, back and side views.

The system is based on hardware from the startup company Fetch Robotics. The Fetch robot uses a variety of sensors and motors distributed among its base, arm, spine, and head that allow for these functions.

The custom programs running in the Fetch and on an external server facilitate the intelligent usage of these features in such a way that enables the robot to autonomously complete menial tasks, such as cleaning and organizing. In this configuration, the Assistive Robot may be employed in domestic, household, or otherwise assistive roles to aid its human counterparts in the completion of tasks.

2. Objectives

Project Objective

While the Assistive Robot will have a wide array of uses, for the scope of this project we aim to implement one key feature: the ability to autonomously navigate an environment in search for a simple, plastic water bottle. Once it has located the bottle, it will be able to approach and accurately grab the bottle with its arm. It will then navigate to some receptacle and release the bottle before starting its search again. This task was chosen as a proof of concept for the types of functions the robot will one day be able to perform.

Looking Forward

Successful completion of the bottle task will effectively prove the system capable of work in a variety of fields. Domestically, the robot will be able to assist in household chores like cleaning, organizing or even cooking. More significantly, the robot could be used to help elderly or disabled people who have weakened motor functions in performing basic tasks, allowing them to retain their independence.

Figure 2. Two Fetch Robots working with boxes.

It is even feasible that the Assistive Robot will make an impact in locations like fulfillment centers and warehouses, as its object detection and navigation features make it superior in many ways to current assembly line style robots.

3. Project Features/Specifications

This project features four main hardware components: robot arm, wheels and distance sensors, RGBD camera, and processing server.

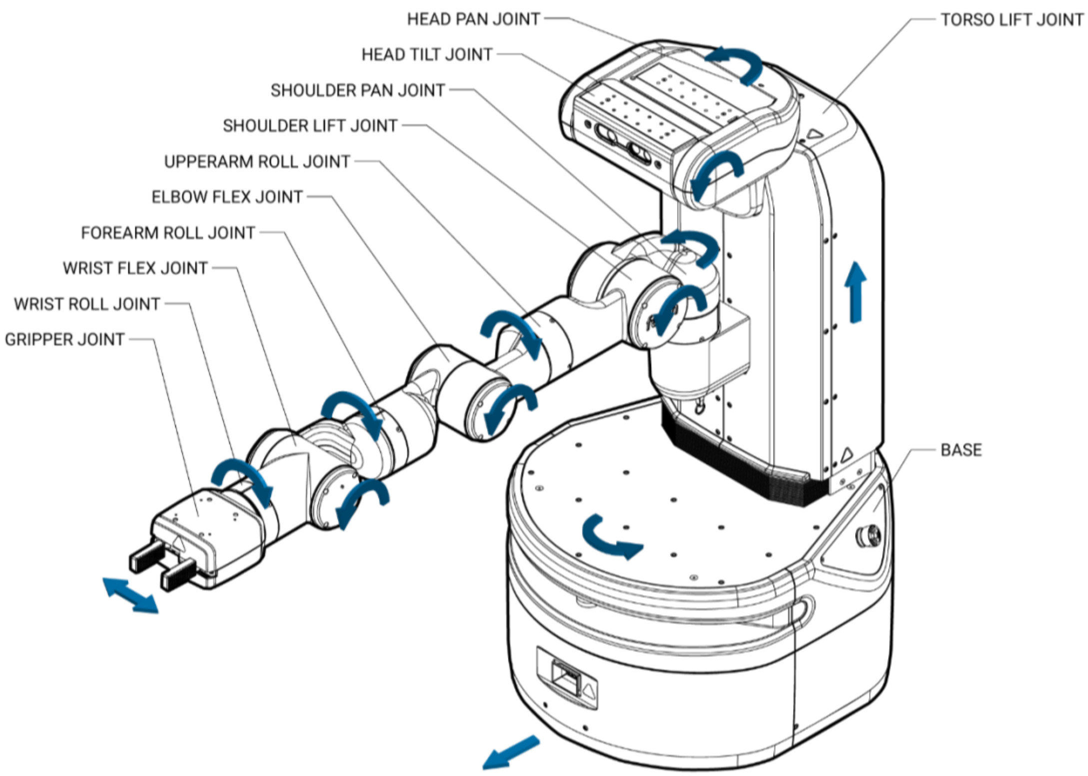

Figure 3: Fetch Robot's axis of rotation

The robot arm, as seen in Figure 3, will have eight control motors allowing six degrees of freedom and the ability to grab an object. Along the base, there are six laser distance sensors and two wheels allowing for obstacle avoidance. At the top, there is a set of optical cameras arranged so that stereo imaging is possible. This gives Fetch the ability to create a 3-D map of its environment and send image data back to the artificial intelligence object detection software running on the processing server.

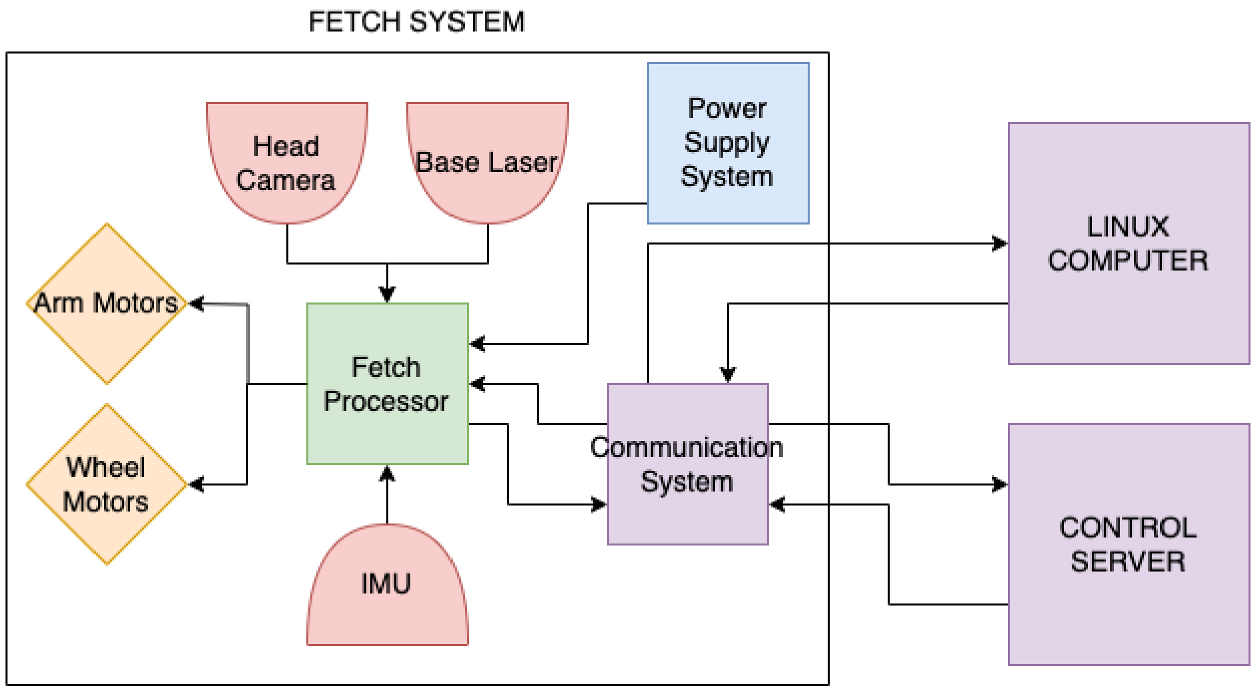

Figure 4: Hardware Block Diagram

A block diagram gives an excellent overview of how our hardware systems integrate together to accomplish our deliverable. The Linux computer seen in Figure 4 is a separate laptop just used to for programming purposes and will serve no function during the demo.

The software side also has four main components: ROS Melodic software, internal Fetch software, image processing, and simultaneous localization and mapping (SLAM). The ROS software will be used to control the robots movements and sensor operations while the internal Fetch software will translate ROS commands into actual wheel and arm movements. Figure 5 shows a generalization for how the image processing and SLAM processing will take place in the system.

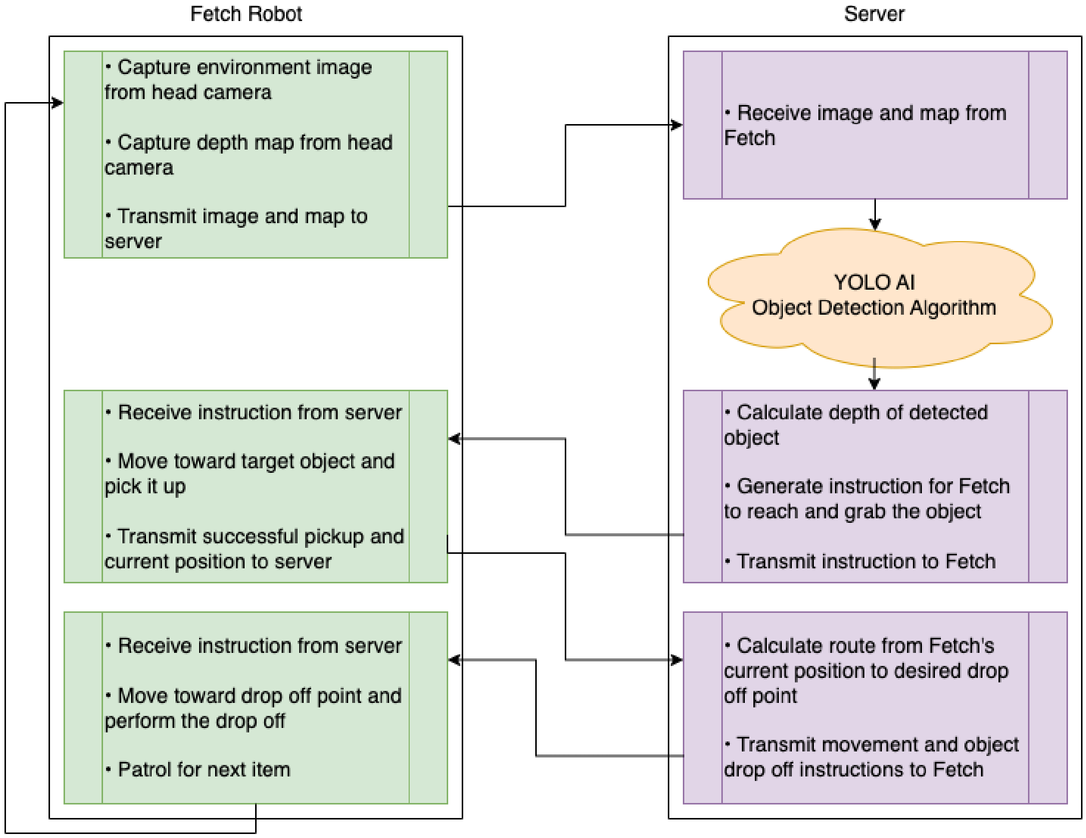

Figure 5: Software Block Diagram

The stereo camera data from the RGBD camera is processed internally on the Fetch, allowing the robot to create a 3-Dimensianl map of its surroundings and locate itself in that map. This camera data will also be used to identify an object, a water bottle, in the robots surroundings. To do this, the image data is sent to the processing server, which is running an algorithm known as You Only Look Once or YOLO. This software uses artificial intelligence to identify objects: in this case, it will identify water bottles in the image. This data will allow Fetch to map the water bottle to a coordinate in the 3-D map of its surroundings. From then, Fetch will use its wheels and distance sensors to navigate within grabbing distance of the water bottle. More image data will be used to navigate the arm to the bottle and grab it. After successful acquirement of the bottle, instructions will the be generated for the robot to navigate to a drop-off location and dispose of the bottle.

4. Design Decisions

While we are still early in the development process, we have already been faced with several choices regarding the implementation of the Assistive Robot's functionalities. The first was in relation to the scope of the project as a whole. As previously mentioned, the end goal of the robot, as a generalized autonomous helper, is beyond the scope of a 1-semester, 2 student design project. Because of this, we chose to set our goal solely on the aforementioned bottle retrieval task, as it seemed a lot more attainable with the time and resources at our disposal.

Another key decision relates to the allocation of the system's software. Our current plan, as reflected in our block diagrams, is to perform all calculations and computations on the server, rather than the robot's internal processor. We came to this decision before we have gotten the chance to experiment with the robot, so it is subject to change, however it does come with some advantages. Beyond the simplicity of having all of the processing on a single device rather than two, we expect the server to have more processing power than the robot, allowing for a quicker and more efficient system overall.

The only other decision we have had to make so far is related to the order of operations of the program, and the sample rate of the Fetch camera. As it currently stands, we plan to only require a single photo of the bottle to be taken before the robot travels to it and picks it up. Depending on testing, we may decide to have multiple pictures taken as it approaches the bottle if it proves to be more effective at the task at hand. Based on what we have been told about the robot's precision, one photo, taken from a distance, should suffice, but further testing will be required to know for certain.

5. Project Plan

The development schedule follows a 12-week plan. Starting off meeting with the undergraduate, graduate, and faculty members responsible for the Fetch Robot project, and installing all the software packages needed to develop the project.

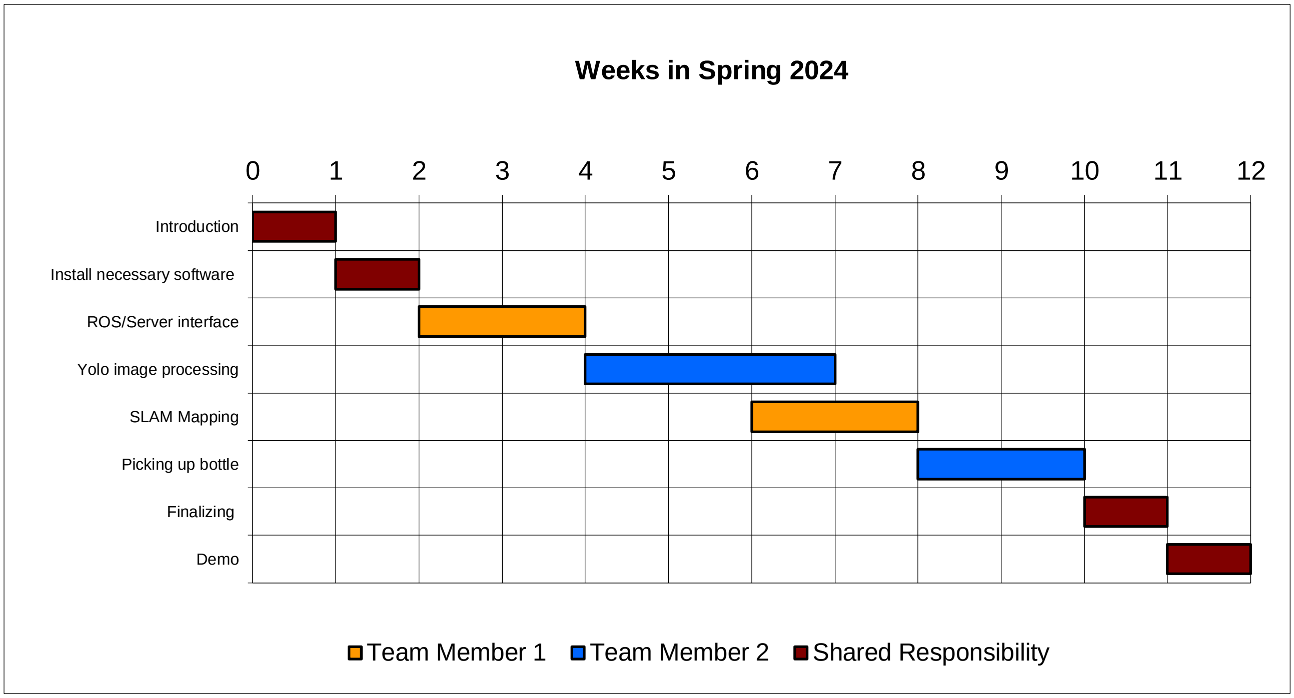

Figure 6: Gantt Chart

After this, the schedule moves onto setting up the interface between the robot and the server and getting image data across from robot to server. Then the data processing needs to be developed. This will involve using YOLO to detect water bottles and properly placing the bottles coordinates in the 3-D map. Flowing into the 3-D map generation, the Fetch's stereo cameras will be used to generate the 3-D map and accurately place itself within the coordinate system: this is known as SLAM mapping. Picking up the battle involves sending data back and forth between the robot and server and successful acquirement of the bottle. These intermediate tasks are broken into different responsibilities with team member 1 responsible for tasks in orange, team member 2 responsible for tasks in blue, and shared responsibilities for tasks in maroon.