University of South Florida

Introduction to Robotics
CAP 4662

Assignment 5 Report

Submitted By:
William Gonzalez-Guzman
U98299955

Due Date: 12/06/22
Submit Date: 12/04/22

Table of Contents

Abstract	3
Introduction	3
List of Ideas	5
Robotic System Schematic Drawing	6
Detailed Device List	7
Software Architecture Schematic Drawings	8
Running Flowcharts	10
List of Anticipated Challenges	13
Conclusion	14

Abstract

This report illustrates a theoretical robot designed to complete 3 of the 6 tasks in the Robotic Grasping and Manipulation Competition (RGMC). This report offers an individual submission of a theoretical robot that could complete the competitions 1, 2, and 3 tasks. The report displays the theoretical robots' “robotic system schematic drawing”, its “detailed device list”, and “software architecture schematic drawings” and “running flowcharts” for those 3 tasks. This report also includes 5 ideas incorporated into the robot's design and 5 challenges this interaction of the theoretical robot won't be able to overcome.

Introduction

While there are 6 tasks in the RGMC, this report only illustrates “software architecture schematic drawings” and “running flowcharts” for 3 tasks. Namely **tasks 1**, **2**, and **3**. Before a task is started there is an **initial setup** required for the team's robot.

Initial Setup

Based on the work area specified by the contestants, the following objects are randomly placed in the workspace defined by the team before the beginning of the competition.

- A stack of five dinner plates, a stack of five salad plates, a stack of five soup bowls, a stack of five cloth napkins
- A silverware tray with five sets of silverware including salad forks, diner forks, soup spoons, table knives, and tablespoons. and 1 pair of ice cube tongs
- Five wine glasses, five water glasses, five teacups, and five saucers for the teacups
- A pitcher (1 liter), a wine bottle (750 ml), and a coffee Carafe (1 liter)
- An ice cube bucket with about 30 ice cubes
- A cup of 10 salt/sugar packets
- A measuring cup

Before starting, the team will specify a dinner set area.

Task 1

Set down a dinner plate at the center (left/right) of the defined dinner set area and one inch from the table edge. Place a salad plate inside the middle of the dinner plate, then place a soup bowl inside the middle of the salad plate.

Task 2

Place the dinner fork on the left of the dinner plate. Place the dinner knife on the right side of the dinner plate and have its blade facing the plate. Place the salad fork on the left of the diner fork and the tablespoon on the right side of the diner knife. Place the napkin in a rectangle on the left-hand side of the plates and outside of the forks.

Place a teaspoon right above the dinner plate with the handle pointing to the right. The forks and knives should be also vertical to the table edge, and one inch from the table edge. Keep an even distance between the two forks as there is between the dinner fork and the dinner plate, not too much and not too little, but there is not a specific amount. Just make sure you can see the whole utensil and that none of it is hiding under the dinner plate.

Task 3

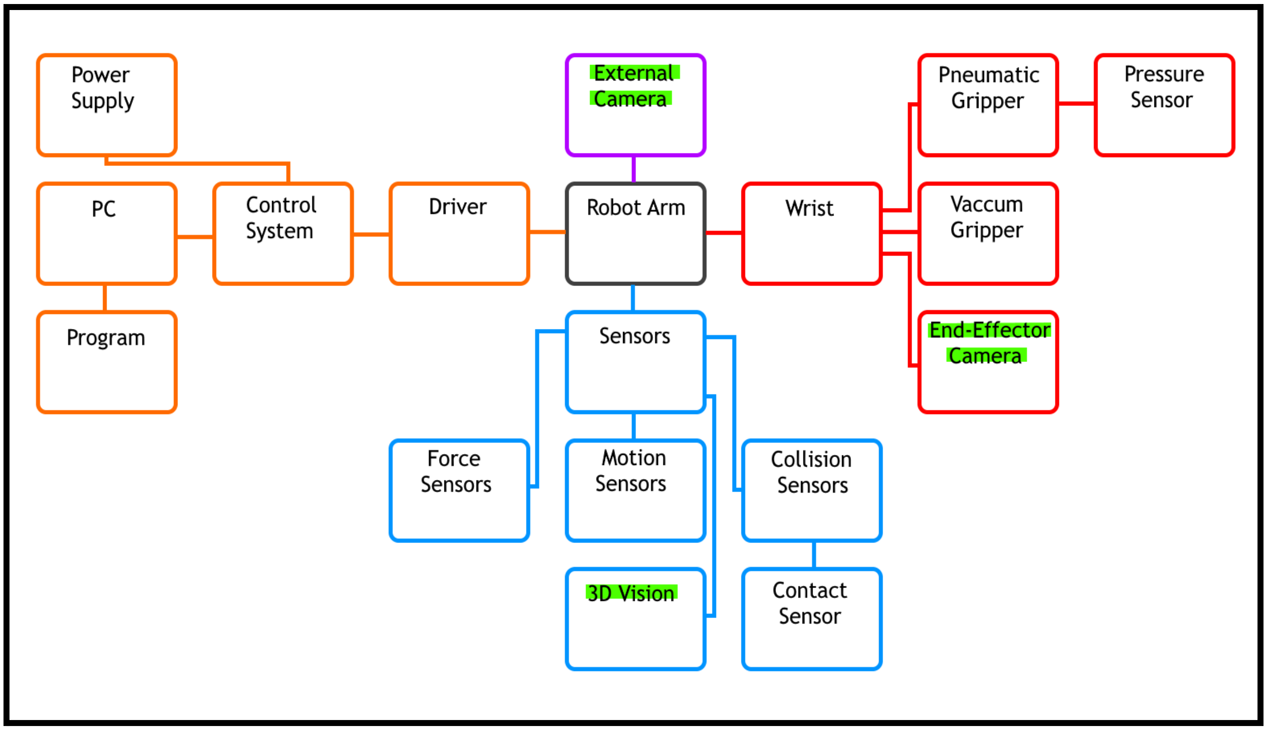
Place a water glass straight above the table knife. Place a wine glass up and to the left of the water glass. Place the cup and saucer to the right of the spoon directly out from the middle of the dinner plate. The base of the cup will sit in the inner ring of the saucer. Place the cup of salt/pepper/sugar above the teaspoon.

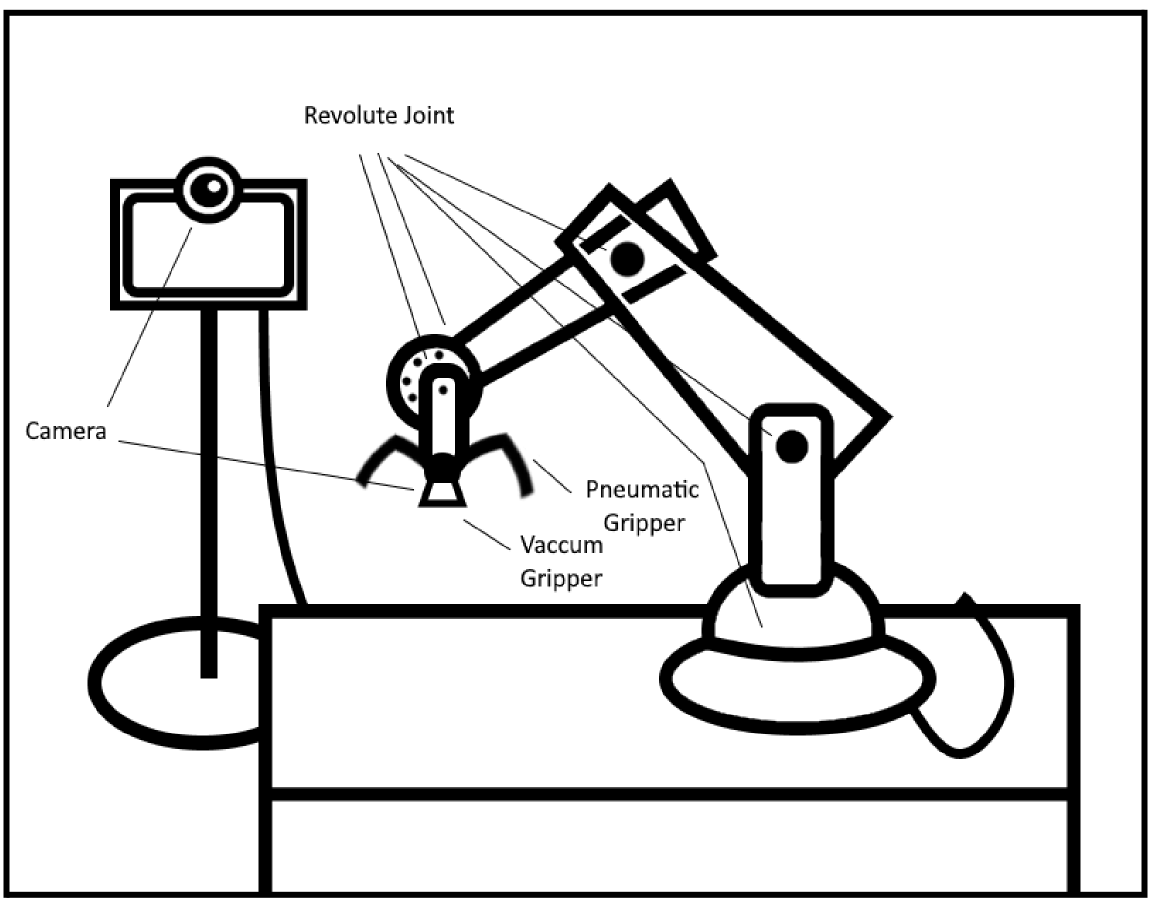
List of Ideas

This is a list of five ideas

1. **The End-Effector is a Combination of Grippers and Sensors**. The gripper is a pneumatic and vacuum gripper with a camera. The pneumatic gripper can be used to un-suction whatever the vacuum gripper has suctioned. There can be a camera in the middle that is used for 3D vision and enable it to calculate its trajectory in its environment. The two grippers also have force sensors that detect how much force they exert on an object. With two different grippers, they can have a larger range of applications. For example, a vacuum gripper is favorable to pick up a plate with compared to a pneumatic gripper.
2. **Use Machine Learning to Recognize Objects**. I can create a machine learning algorithm and apply that algorithm to several algorithmic bots, then make a test with images that are forks and aren't forks. I can test the bots and see how many they get right. The ones you get the most right, live on to make slight changes to their own algorithms with the goal that they get the most right. If this is done to billions of bots, over and over again, then it's feasible that there comes a bot that will recognize a fork, or a plate, or a sugar packet in an image 99% of the time.
3. **The Camera Captures Images in Black and White**. Ideally, I'd want the robot to be able to recognize objects in its environment based on a database/machine learning bot that can match pictures and objects. If the image is in black and white then there are fewer distractable factors that may confuse the robot's programming. For this reason, the workspace's background (aka table) would need to be of a contrasting color.
4. **The Camera Surveys the Environment to Create a 3D environment for Path Planning**. A camera can be attached to the end-effector and it can survey its workspace capturing imaging from multiple angles to create a 3D environment with depth. It can also locate the desired objects' locations and store them for later use.
5. **Audio Output for Error Detection**. Whenever there is an error being made, such as the robot's path being blocked by an unknown object, or its unable to detect a desired object in the workspace, then it outputs an error message depicting what is going wrong.

Robotic System Schematic Drawing





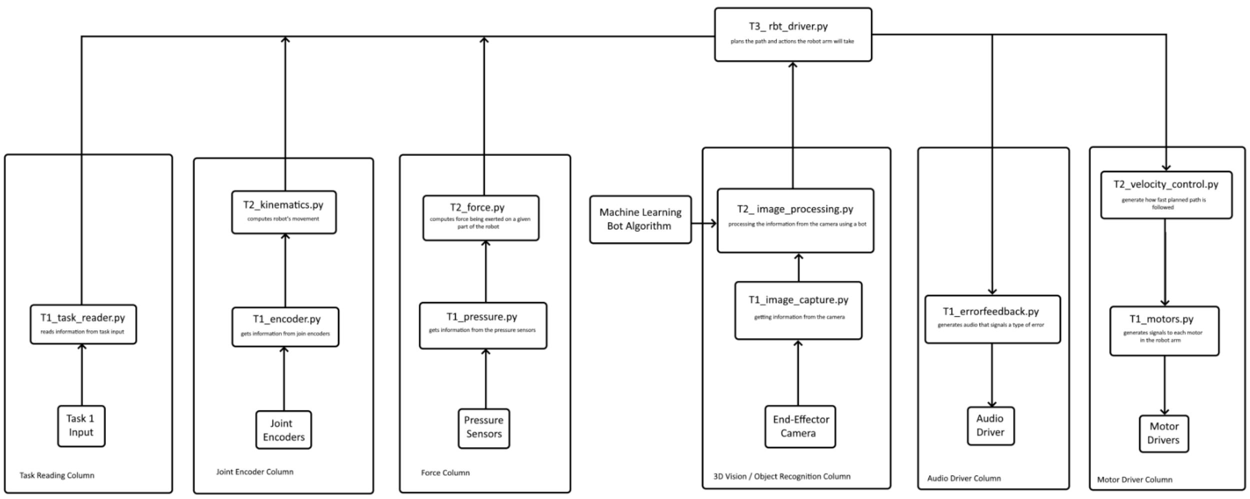
Detailed Device List

Hardware	Reasoning
Robot: UR10e	It has the carrying capacity I need for forks, cups, and plates, it has the reach to set up a table, it is also lightweight enough to be moved around, and it's an advanced version of the UR5e,
Custom Gripper Features: Pneumatic Gripper Vacuum Gripper Camera	I wanted a custom gripper so that it has a wider range of applications. Picking up a plate or something flare is easier to do with vacuum suction than it is with a pneumatic gripper that needs to get under the flat object.
Pneumatic Gripper: AG-105-145 linkage-type adaptive robot gripper	I would want something similar to the AG-105-145 but with the ability to retract its graspers/claws from the workspace and place them at its base and with enough room in between its claws to allow a vacuum gripper. Essentially the pneumatic gripper will release the object from the vacuum gripper.
Vacuum Gripper: Suction cup	I want a suction cup with enough suction to pick up 5lbs or so, that's all that's needed for plates.
Camera: SensoPart - V10-RO-A2-R12 - Visor Robotic	A small camera would be mounted to the side of the end-effector and would always point down. This is so the robot knows where the objects that it wishes to grasp are exactly in the environment so that then the program can calculate the motion to pick up the said object. The robot can have stored in its database objects interests so that it can recognize them in the environment.
Pressure Sensor	I want a pressure sensor incorporated into the grippers(both pneumatic and vacuum) so that the robot knows how much pressure is being applied to an object. This way the robot doesn’t accidentally damage something delicate.
External Camera	Ideally I’d want several external cameras from multiple angles of the robot’s workspace. That way the robot has more information about its environment that can be used to help calculate the trajectory/movement of the arms and grippers.
Contact Sensors	Error planning. Essentially I want the robot to sense if it is bumping into anything it's not supposed to and stop its motion or reverse it. This will prevent it from damaging the object it's bumping into or from damaging itself. For example, if a wall was next to the robot when it was motion planning and didn't detect the obstacle with its cameras, and the robot didn’t have contact sensors placed throughout its links then the robot could topple over by pushing against the wall.
Motion Sensors	Motion sensors would be used to calculate the robot's current position of its links relative to its self/base. This way the program can tell it to move to certain positions.
Force Sensors	If the robot presses onto something, then it’ll be able to measure the force being exerted.
3D vision	Will allow the robot to plan its trajectory toward objects, how to grasp said objects, and how to carry them to the desired location.

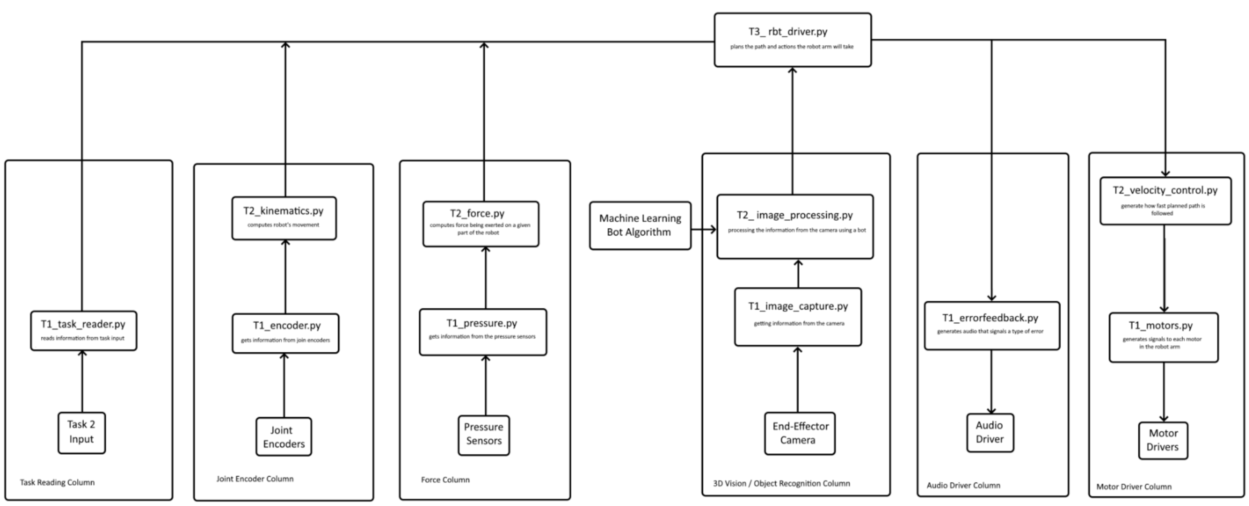
Software Architecture Schematic Drawings

The following are, software architecture schematic drawings for three of the six tasks. Including the detailed components of the software architecture and how they are connected.

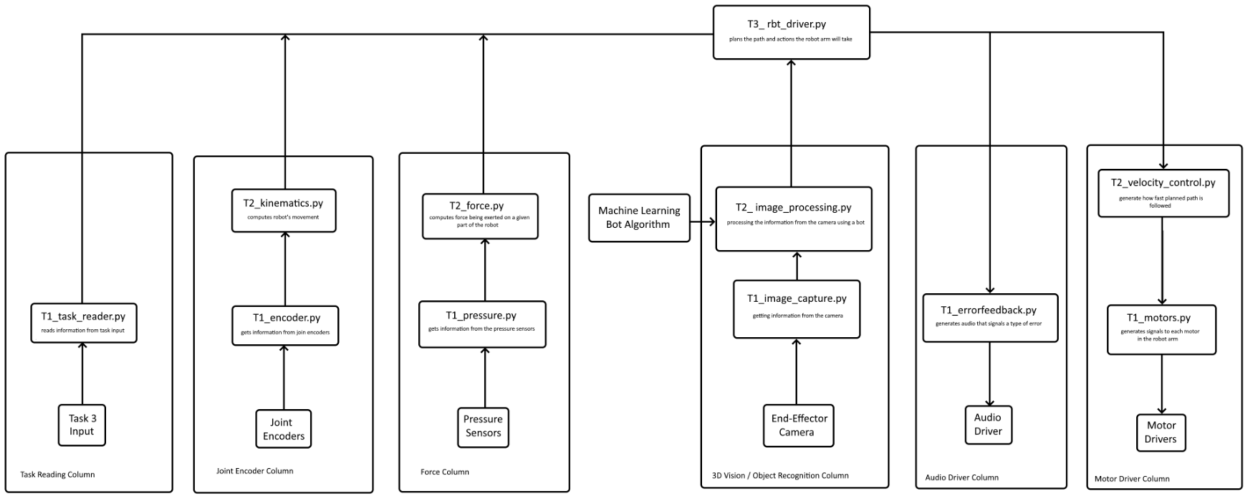
Task 1



Task 2



Task 3

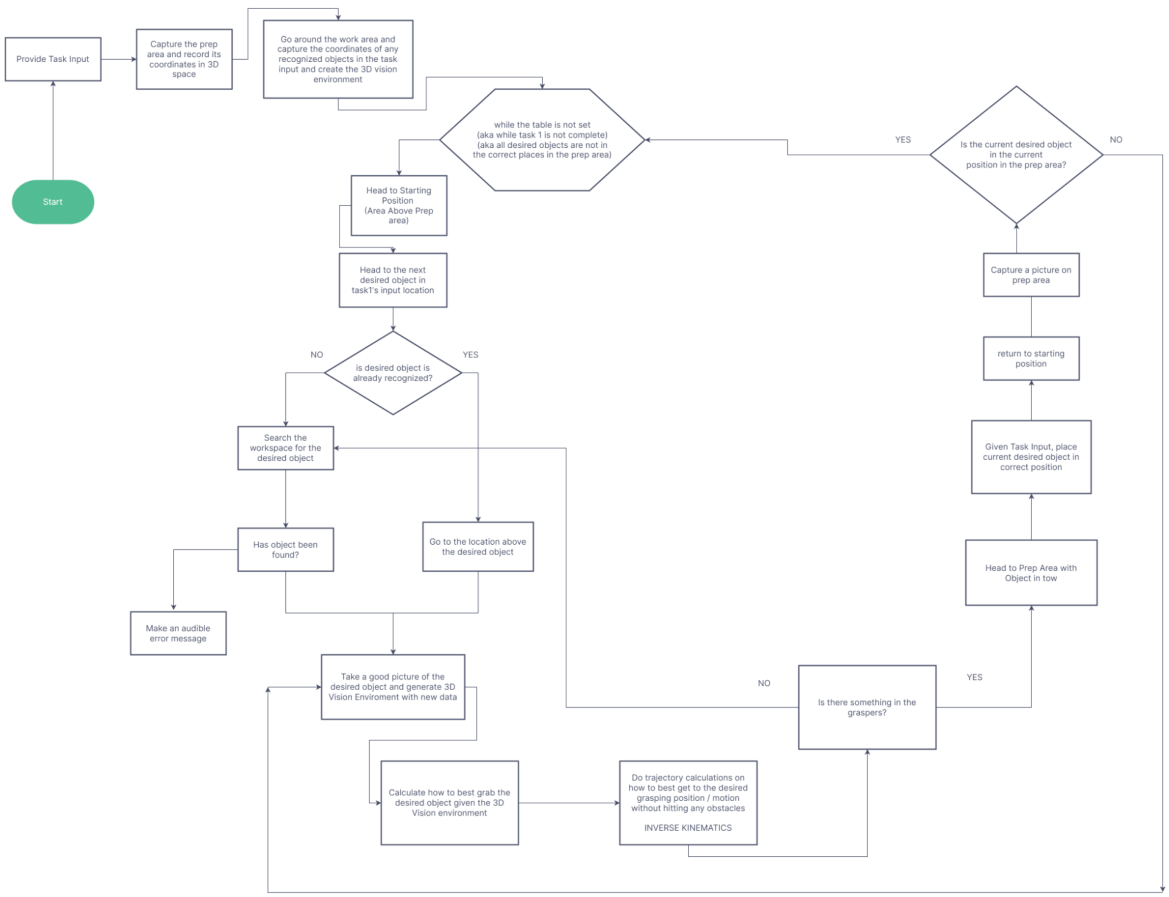

T3_ rbt_driver.py
plans the path and actions the robot arm will take
T1_task_reader.py
reads information from task input
Task 3 Input
Task Reading Column
T2_kinematics.py
computes robot's movement
T1_encoder.py
gets information from join encoders
Joint Encoders
Joint Encoder Column
T2_force.py
computes force being exerted on a given part of the robot
T1_pressure.py
gets information from the pressure sensors
Pressure Sensors
Force Column
Machine Learning Bot Algorithm
T2_ image_processing.py
processing the information from the camera using a bot
T1_image_capture.py
getting information from the camera
End-Effector Camera
3D Vision / Object Recognition Column
T1_errorfeedback.py
generates audio that signals a type of error
Audio Driver
Audio Driver Column
T2_velocity_control.py
generate how fast planned path is followed
T1_motors.py
generates signals to each motor in the robot arm
Motor Drivers
Motor Driver Column

Running Flowcharts

Running Flowcharts of the software for three of the six tasks. Includes all steps such as perception, motion planning, trajectory generation, and error handling.

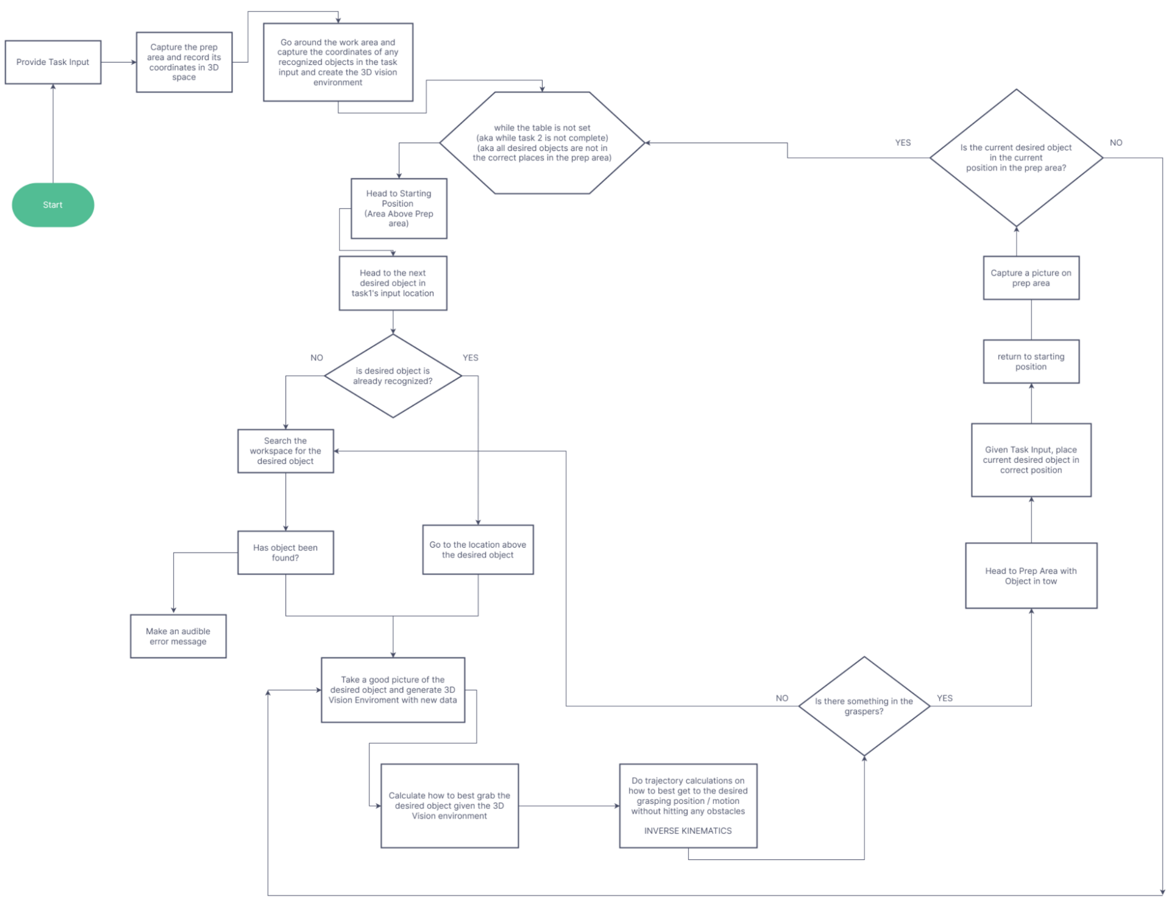
Task 1 Flowchart

Set down a dinner plate at the center (left/right) of the defined dinner set area and one inch from the table edge. Place a salad plate inside the middle of the dinner plate, then place a soup bowl inside the middle of the salad plate.



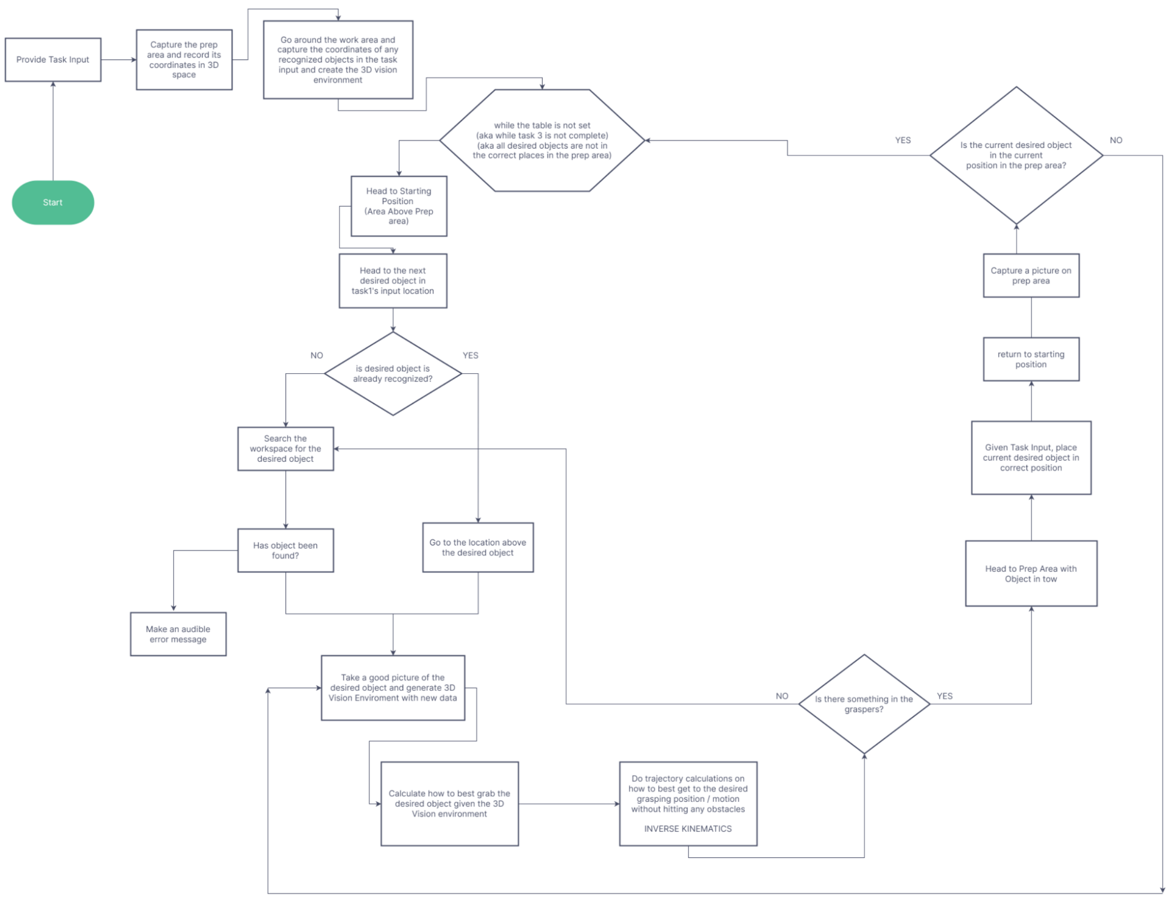
Task 2 Flowchart

Place the dinner fork on the left of the dinner plate. Place the dinner knife on the right side of the dinner plate and have its blade facing the plate. Place the salad fork on the left of the diner fork and the tablespoon on the right side of the diner knife. Place the napkin in a rectangle on the left-hand side of the plates and outside of the forks. Place a teaspoon right above the dinner plate with the handle pointing to the right. The forks and knives should be also vertical to the table edge, and one inch from the table edge. Keep an even distance between the two forks as there is between the dinner fork and the dinner plate, not too much and not too little, but there is not a specific amount. Just make sure you can see the whole utensil and that none of it is hiding under the dinner plate.



Task 3 Flowchart

Place a water glass straight above the table knife. Place a wine glass up and to the left of the water glass. Place the cup and saucer to the right of the spoon directly out from the middle of the dinner plate. The base of the cup will sit in the inner ring of the saucer. Place the cup of salt/pepper/sugar above the teaspoon.



List of Anticipated Challenges

This is a list of five challenges anticipated that the robotics system will not be able to overcome.

1. **Certainly Recognise Objects**. To certainly recognize an object in an image, the bot has to be advance enough that it can recognize objects 100% of the time. This is not possible with the current technology. There will always be a small change the bot won't be able to recognize the object. There may also be external factors that might interfere with object recognition such as light intensity being too bright or too dark, the surface of the object being too shiny, the desired object being hidden by another object, and the recognized objects being too close to one another, interfering with each other.
2. **Grasping Unique Objects**. Some objects might be too slippery or too small to grasp with the given end-effectors. It might even be difficult to grasp something such as paper off a table. The robot would need to leverage the paper's edges to grasp underneath it and try again and again with slightly different approaches.
3. **Finding Objects Outside Its Workspace**. The robot first records its workspace in a 3D environment using its camera, hence if a desired object is outside the initial recording then the robot won't be able to find it and grasp it.
4. **Grasp Objects Outside its Workspace**. The robot has no means of traversing outside of its given workspace. The length of its extended joints in all directions is the maximum distance its workspace can encompass.
5. **Manipulate Delicate Objects**. Manipulating a delicate object requires extreme position this current version of our robot is unable to perform. To pick up a delicate object or something else that is much smaller and more delicate than the robot itself means that the robot will need to execute a precise amount of coordination and precision. It needs to apply force to the object it's picking up, but not too much that it damages it, and it must be accurate enough that it can correctly grasp the object and take it where it needs to go without having the object slip through its graspers.

Conclusion

In conclusion, the robot designed uses two different grippers, has a camera attached to its end-effector that enables 3D vision, it uses machine learning to correctly identify objects in the tasks such as forks, plates, water cups, etc. This report shows “software schematic drawings” and “flowcharts” for tasks 1, 2, and 3. Tasks 1, 2, and 3 are similar in nature; they all require grabbing certain objects and placing those objects in certain positions. That is why their software schematics drawings and flow charts are similar.

Additionally, I wish to tour the Robot Perception and Action Lab and run my HW4 code on the UR5e robot.