DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SPRING 2022

H.I.L ROBOT TEAM
H.A.F.R.A

RESHA ADHIKARI
KATIA LOPEZ
NISHAN PATHAK
CESAR REA
RICHARD TRAN

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	2.20.2022	KL, NP	document creation
0.2	2.20.2022	RA, KL, NP, CR, RT	complete draft
1.0	2.21.2022	RA, KL, NP, CR, RT	official release
2.0	5.11.2022	RA, KL, NP, CR, RT	final release

CONTENTS

1 Introduction **5**

2 System Overview **5**

3 UR5 Robot Layer Subsystems **6**
3.1 Layer Hardware 6
3.2 Layer Operating System 6
3.3 Layer Software Dependencies 6
3.4 Arm Subsystem 6
3.5 Suction Gripper Subsystem 7
3.6 RealSense Camera Subsystem 7

4 Movement Program Layer Subsystems **9**
4.1 Layer Hardware 9
4.2 Layer Operating System 9
4.3 Layer Software Dependencies 9
4.4 Picking Subsystem 9
4.5 Position Instructions 10

5 Vision Program Layer Subsystems **12**
5.1 Layer Hardware 12
5.2 Layer Operating System 12
5.3 Layer Software Dependencies 12
5.4 AruCo Detector 12
5.5 World Coordinates 12
5.6 Human Assistance 13

6 Appendix A **14**

List of Figures

1	System architecture	5
2	Arm Subsystem Description Diagram	6
3	Suction Gripper Subsystem Description Diagram	7
4	RealSense Camera Subsystem Description Diagram	8
5	Picking Subsystem Description Diagram	9
6	Position Instructions description diagram	10
7	AruCo Detector subsystem description diagram	12
8	World Coordinates subsystem description diagram	13
9	World Coordinates subsystem description diagram	14
10	Rviz Design	14

1 INTRODUCTION

The UR5 robot is a lightweight robot that is used to pick and drop of objects up to 11lbs. For this project, we are introducing machine learning concept along with the bin picking technology. The UR5 robot will be able to pick up the envelopes with the Fiducial markers. With the help of the camera installed on the robot, and the assistance of a computer vision system it will be able to identify the envelope with the markers. After identifying the object, the robot's arm will create the movement towards that envelope and grab it with the assistance of suction attached at the robot arm.

2 SYSTEM OVERVIEW

The Human Assistance For Robot Arm (H.A.F.R.A) project has three major architectural layers which are the the UR5 robot, the Movement Program, and the Vision Program. The program will send instructions to the robot to perform its task, which will be picking up envelopes with an ArUco tag label off the table and placing them into a bin. The RealSense camera will be used to detect the fiducial markers. In addition, the vision program will have human assistance, for cases such that if the envelopes have damaged or missing markers. A human will be able to access the robot remotely, will assist the robot, and ultimately fix the problem by inserting the pixel coordinates to guide the robot as where to move. This section contains the high-level block diagram of the layers, as shown in the figure below, as well as detailed descriptions of the functions of each layer.

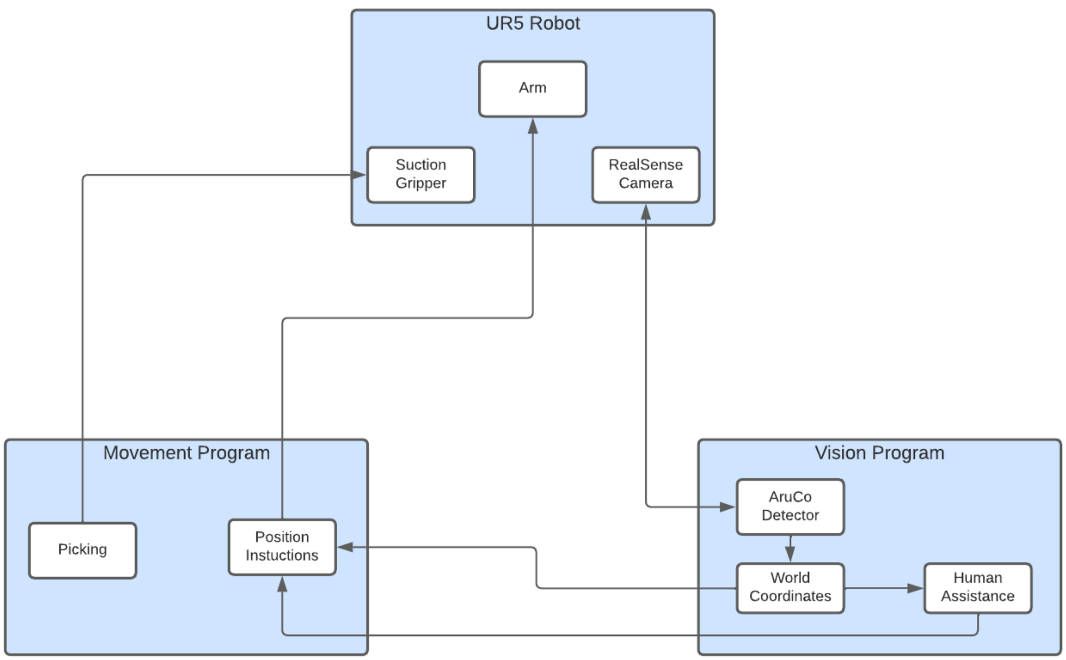

Figure 1: System architecture

3 UR5 Robot Layer Subsystems

This layer contains the hardware of the overall system. This layer responds to the Movement and Vision Program Layers.

3.1 Layer Hardware

N/A

3.2 Layer Operating System

N/A

3.3 Layer Software Dependencies

N/A

3.4 Arm Subsystem

This subsystem is a piece of hardware. It is the robotic arm of the system. This subsystem responds to position instructions inputted by the Movement Program.

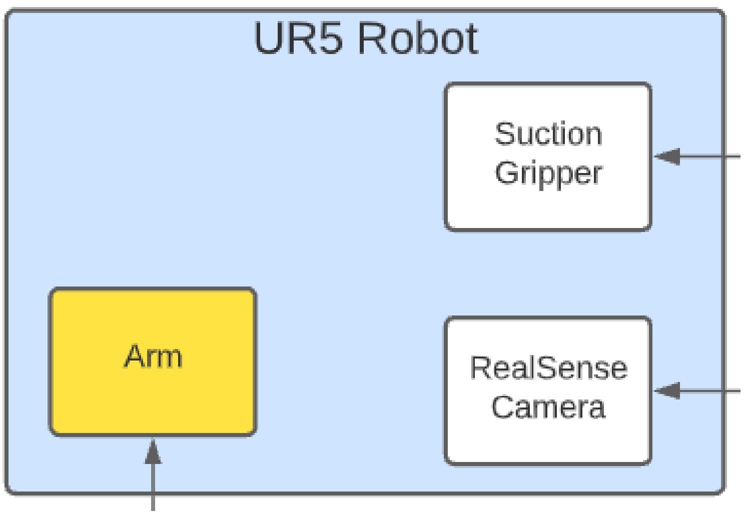

Figure 2: Arm Subsystem Description Diagram

3.4.1 Subsystem Hardware

The hardware component used for this subsystem is the Universal Robot 5.

3.4.2 Subsystem Operating System

N/A

3.4.3 Subsystem Software Dependencies

N/A

3.4.4 Subsystem Programming Languages

N/A

3.4.5 Subsystem Data Structures

N/A

3.4.6 Subsystem Data Processing

N/A

3.5 Suction Gripper Subsystem

The Suction Gripper subsystem will be used to allow the system to pick up the envelopes. This subsystems responds to the picking inputs by the Movement Program layer.

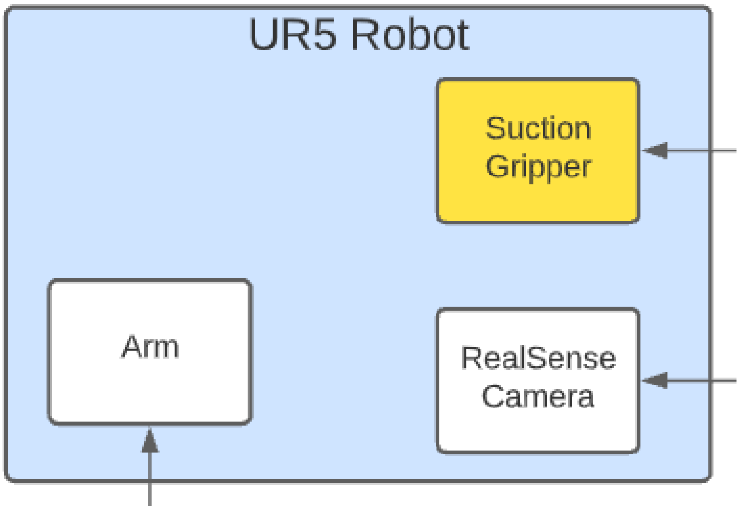

Figure 3: Suction Gripper Subsystem Description Diagram

3.5.1 Subsystem Hardware

This piece of hardware will have the main responsibility of grabbing envelopes. As the name implies, the hardware is a suction gripper.

3.5.2 Subsystem Operating System

N/A

3.5.3 Subsystem Software Dependencies

N/A

3.5.4 Subsystem Programming Languages

N/A

3.5.5 Subsystem Data Structures

N/A

3.5.6 Subsystem Data Processing

N/A

3.6 RealSense Camera Subsystem

The RealSense Camera subsystem will be used to scan the fiducial markers on the envelopes. This subsystem interacts with the AruCo detector subsystem in the Vision Program layer.

3.6.1 Subsystem Hardware

Intel RealSense camera

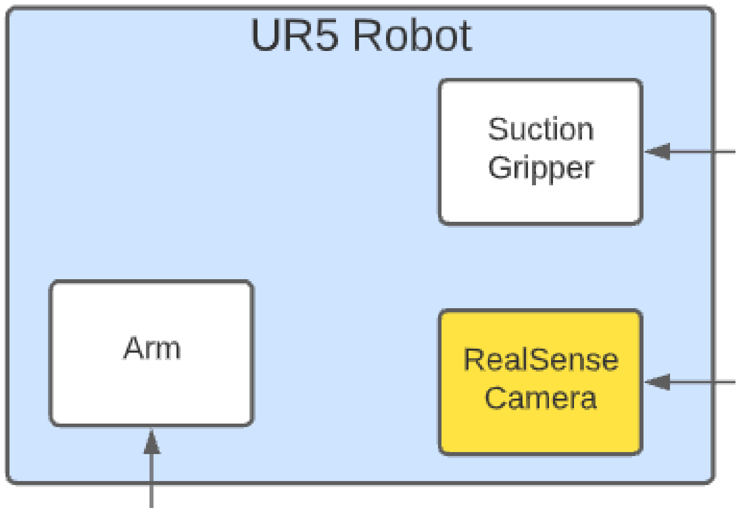

Figure 4: RealSense Camera Subsystem Description Diagram

3.6.2 Subsystem Operating System

N/A

3.6.3 Subsystem Software Dependencies

N/A

3.6.4 Subsystem Programming Languages

N/A

3.6.5 Subsystem Data Structures

N/A

3.6.6 Subsystem Data Processing

N/A

4 Movement Program Layer Subsystems

This layer contains information of the movement program, which contains picking and position instructions.

4.1 Layer Hardware

The hardware components involved are the router and the computer. A local router needs to be implemented with the UR5 and computer assigned as attached devices to allow data exchange on a local network.

4.2 Layer Operating System

The movement program layer requires Ubuntu 18.04 as the operating system.

4.3 Layer Software Dependencies

This layer uses the Robot Operating System (ROS) which is a set of software libraries to help program robot applications. We used ROS melodic because this version had most of the libraries we required than the other versions. ROS Drivers were required for communication between UR5 and the hostâs computer. Rviz was required to simulate the robot and its environment for testing.

4.4 Picking Subsystem

This subsystem contains the functionality of when and how our suction will activate to pick up the envelopes and deactivate to drop them into a bin.

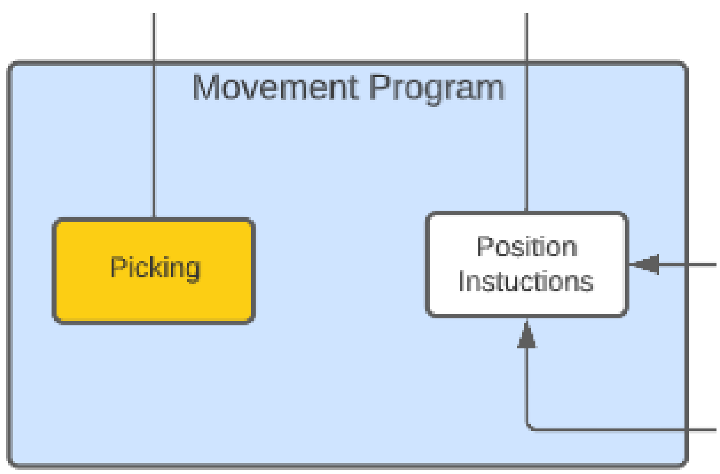

Figure 5: Picking Subsystem Description Diagram

4.4.1 Subsystem Hardware

The involved hardware component for this subsystem is a vacuum connected to a power source with switches that can be activated via GPIO through the UR5 control box.

4.4.2 Subsystem Operating System

N/A

4.4.3 Subsystem Software Dependencies

The picking subsystem uses services from rospy, a Python client library for ROS, to configure GPIO pin states through script and therefore be able to turn the vacuum on and off when picking and dropping the envelopes.

4.4.4 Subsystem Programming Languages

This subsystem uses Python 2.

4.4.5 Subsystem Data Structures

N/A

4.4.6 Subsystem Data Processing

This subsystem uses rospy to request services.

4.5 Position Instructions

These instructions move the robot to the given positions. The vision program calculates the center pixel coordinate of the Aruco tag on the envelope and converts it to an (x,y) real-world coordinate. The robot arm moves towards the tag and picks it up. After that, the robot is instructed to move towards the bin and drop the envelope. For the cases where the tags are not detectable, the human assistance portion of the vision program is used. A user enters the pixel coordinates which guides the robot arm on what location the envelope is. This process continues until the end code is entered by the user and robot comes to the rest position.

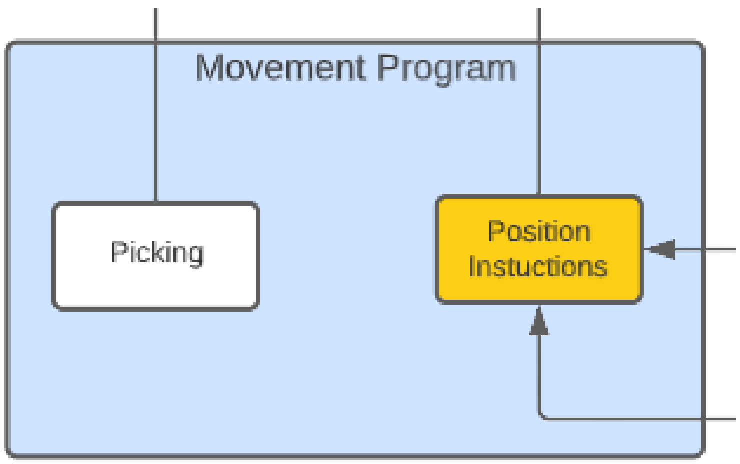

Figure 6: Position Instructions description diagram

4.5.1 Subsystem Hardware

N/A

4.5.2 Subsystem Operating System

We used python programming in ROS to provide movement instructions to the robot arm.

4.5.3 Subsystem Software Dependencies

The software dependencies that this subsystem uses are the software libraries from ROS, Intel RealSense SDK, an open-source library designed for processing 3D data, and the Python library OpenCV. Moveit Libraries were needed for real-time joint readings as well as basic movement functionalities.

4.5.4 Subsystem Programming Languages

This subsystem uses Python 2.

4.5.5 Subsystem Data Structures

Joint values are kept inside a 6-sized array, values are given in radians to represent each jointâs angle for movement instructions to the robot. Positions based on an x,y,z coordinate are also used.

4.5.6 Subsystem Data Processing

This subsystem implements an algorithm to provide appropriate movement directions to the robot as required for the bin picking.

5 Vision Program Layer Subsystems

In this layer, the center pixel coordinate of the AruCo tag is converted into a real-world coordinate, so that the robot can move to that location and pick up the envelope.

5.1 Layer Hardware

The only other hardware except the ones included in the system level is the computer.

5.2 Layer Operating System

This layer will use Ubuntu 18.04 so the operating system is Linux. We use ROS operating system to work on the coordinates calculation.

5.3 Layer Software Dependencies

This layer uses Python 2.

5.4 AruCo Detector

The robot will be able to detect the AruCo tags via the RealSense camera and the OpenCV libraries used in Python.

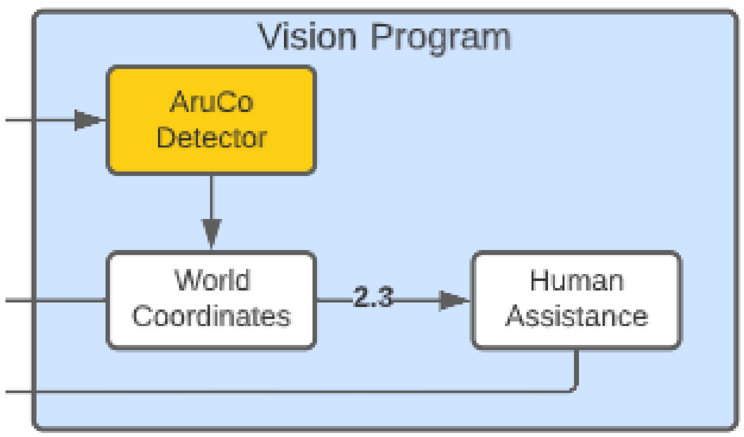

Figure 7: AruCo Detector subsystem description diagram

5.4.1 Subsystem Hardware

Realsense D400 camera is used to detect the AruCo fiducial markers.

5.4.2 Subsystem Operating System

Ubuntu is used for the entire vision program.

5.4.3 Subsystem Software Dependencies

This subsystem will use OpenCV and pyrealsense to detect the pixel coordinates of the AruCo markers.

5.4.4 Subsystem Programming Languages

This subsystem will use Python 2.

5.5 World Coordinates

This section involves the conversion of pixel coordinates of the fiducial markers into the real world x,y coordinates.

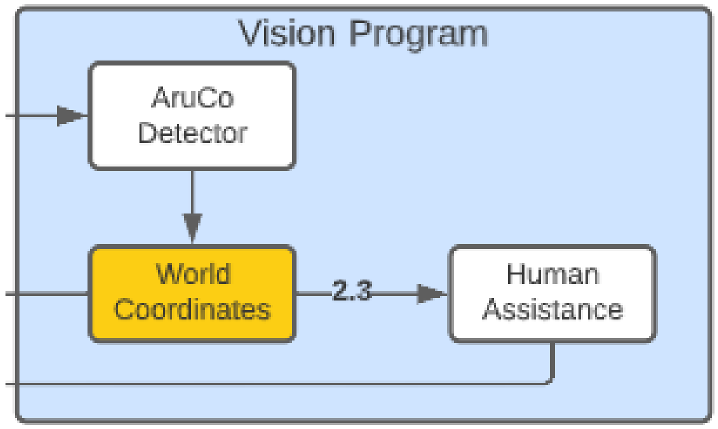

Figure 8: World Coordinates subsystem description diagram

subsubsectionSubsystem Hardware Realsense D400 camera is used to detect the AruCo fiducial markers.

5.5.1 Subsystem Operating System

Ubuntu is used for the entire vision program.

5.5.2 Subsystem Software Dependencies

This subsystem will use Python programming to convert the pixel coordinates into the real world coordinates.

5.5.3 Subsystem Programming Languages

This subsystem will use Python 2.

5.6 Human Assistance

This section involves the cases where AruCo tags are either damaged or missing and the robot is unable to identify them. In such cases, the camera captures an image of the envelope and the user determines the pixel coordinates of the envelope by using their mouse cursor, then manually enters it on the computer. This pixel coordinate is converted to the real world coordinate as described in subsection 5.5. Then the robot is able to pick up the envelopes with the human assistance.

5.6.1 Subsystem Operating System

Ubuntu is used for the entire vision program.

5.6.2 Subsystem Software Dependencies

N/A

5.6.3 Subsystem Programming Languages

This subsystem will use Python 2.

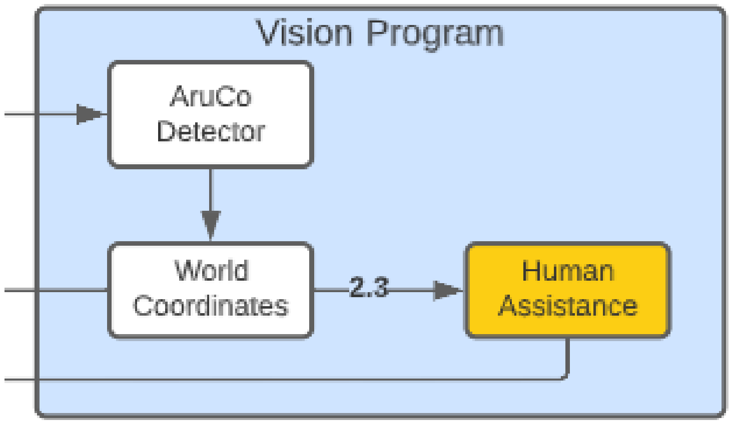

Figure 9: World Coordinates subsystem description diagram

6 Appendix A

Virtual objects were created using ROS visualization (Rviz) to set boundaries for the robot.

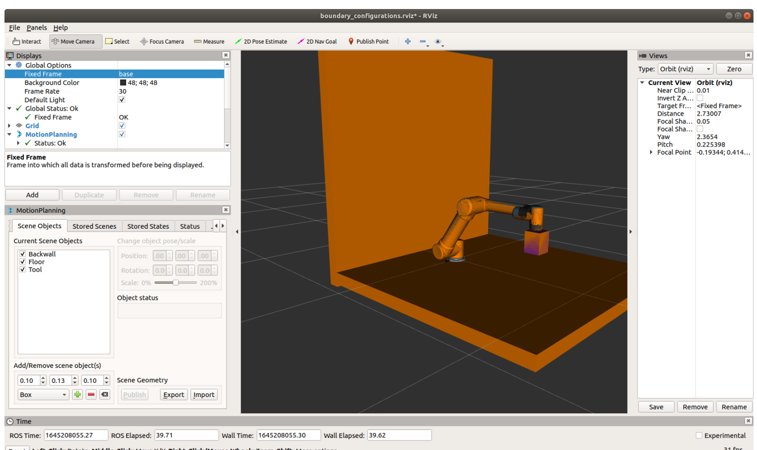

Figure 10: Rviz Design