ELECOMP CAPSTONE
SUMMIT
FRIDAY, MAY 10, 2019

THE UNIVERSITY OF RHODE ISLAND

THINK BIG WE DO™

Optimization of a Cobot Loading System

HEXAGON

Team Members: Harrison Timperley (CPE), Zhengkun Long(ELE)

Technical Directors: Jonathan O'Hare, Joseph Vanpelt,

Project Motivation

The CMM, coordinate measuring machine, is a Hexagon device that measures parts that are loaded onto it. Typically, these parts are manually loaded with high labor costs. Since the loading process is a low value-added activity for humans, it is a good candidate for automation. Not only will automation be more time efficient, but it will also be able to run on off-work hours and increase utilization of the CMM. However, most industrial robots are not safe for close interaction and need highly skilled humans to program them. Therefore, the introduction of a collaborative robot is best suited for this type of human-robot interaction. This robot will be easily programmable and have safety features to protect their human operators.

Key Accomplishments

Drawer Reliability Testing: It is often a misconception that robots have no limits of movement among their own space. Consistently during our tests, we have encountered what is called a singularity. It is a position in which the movement is not mathematically possible to calculate. As a result, this causes the robot to halt. Also, like our own limbs, there are spaces around the robot that are not possible to reach and alternative movement paths need to be created. Lastly, certain operations, such as drawer manipulation result in forceful oscillations and cause the robot to detect collisions. As a result, we implemented a flexible damper on the robot end effector to absorb lost motion. We then placed a drawer handle approx. 6" from the middle of the drawer so that the payload is closer to the center of mass while still avoiding singularities.

Robot Routines Implementation: We programmed two different types of routines, "open and close drawer" and "pick and place parts". These are the two most fundamental routines for this automation program. This is because most every action the robot performs utilizes one of these subroutines. The open and close drawer routine is now implemented with waypoint moves. We tried to use force moves, however, there was too much complication with all the singularity errors. When we were still working on the pick and place routines, one solution we found was to use the built-in UR pallet function. This made pick and place more automatic and easier to use than regular waypoint routine.

Robot Simulation Program: We have used the robot simulation program from Universal Robots since the beginning of this semester. It has helped us to test some of the functions we wanted to use when we are not working directly with the robot. The simulation allowed us to test some force moves when we needed to do the repeatability tests. Using this, we have found out what positions we should have used and which ones worked the best. We have also used this to test the pallet functions and it has worked very well.

Convert Routines into Coordinate: One of the major steps of this project is to convert the routines into coordinates so we can store it in JSON files for easy integration with our software. This allows us to pull the routine from a file and convert it into coordinates for the robot to operate with. This is done by applying robot variables on a plane, getting the coordinates from the plane and inputting the coordinates into JSON files. Then, we can call the coordinates from our system which will allow our interface to communicate with the robot.

Data Parsing and Architecture Organization: We have decided that we will use JSON as opposed to XML and csv files to store our data. This is because JSON is a faster more appropriate data format for our part templates. We designed a modular data structure in layers so that the more general part information could be reused. For example, for each different type of part, similar information such as gripper location and opening width can be reused and then individual part information such as cell locations and part serial numbers can be stored as a queue. This makes it easier for an operator to schedule jobs by reducing the amount of data entry and preventing entry errors.

Part Detection: It is important for the robot to sense whether it picks up a part or not. Otherwise, it will act like it has the part and it will not alert the operator that the part has been missed. This is why we somehow need to sense whether there is a success in the part pick. Luckily, the robot has force detection. We tell the arm to go towards the part and stop if it passes a certain point or if it hits something unexpected. We then tell it to grip the part and if it detects nothing then it will report to the operator that there is an error and it will respond appropriately.

Data Loading and Operation: One of the main requirements of our project is that it must be easy for an operator to operate. This capstone project allows the data to be created beforehand and allows the operator to simply insert the data into the system. Regardless of the different variety of parts, so long as the data is specified for each part beforehand, the system will be able to place the parts into the measurement machine.

Anticipated Best Outcome

Our anticipated best outcome was to create a software interface that will allow the user to initialize the robot with their specific part and pallet parameters on an intuitive interface. The interface will be operated by a human with minimal training and interaction. After the initialization is complete, human interaction would only be to load data and to exchange parts. Different subroutines for the robots to open and close the drawer and pick and place parts on to CMM would be automatically generated from the pallet and part data.

Project Outcome

The Anticipated Best Outcome was achieved. We were able to deliver what was asked for at the beginning of the year.

Figures

Figure 1. Robot Cell Setup with Coordinate Measuring Machine

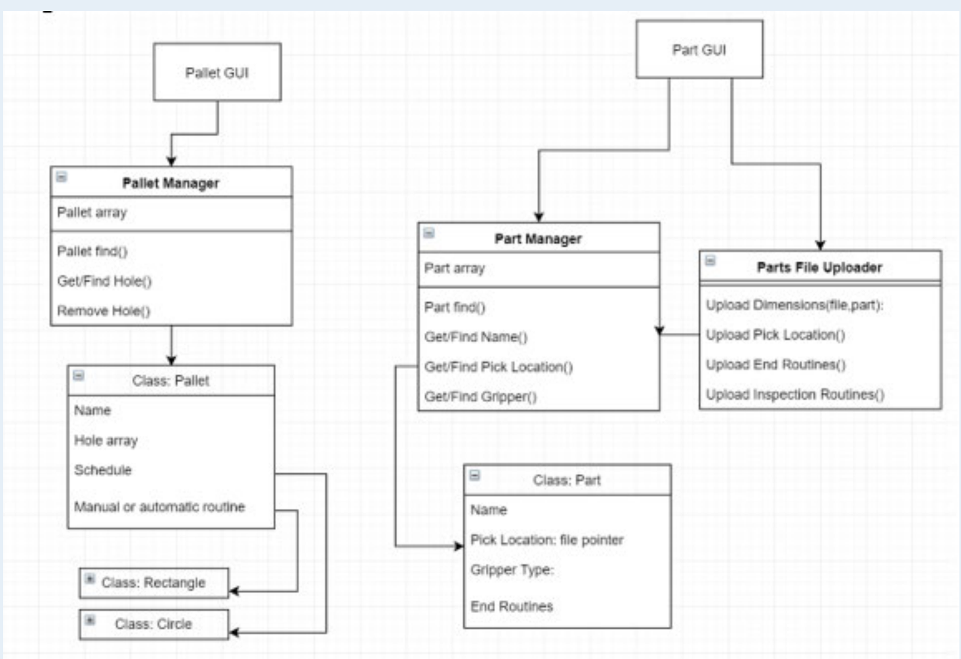

Figure 2. GUI Layout Block Diagram

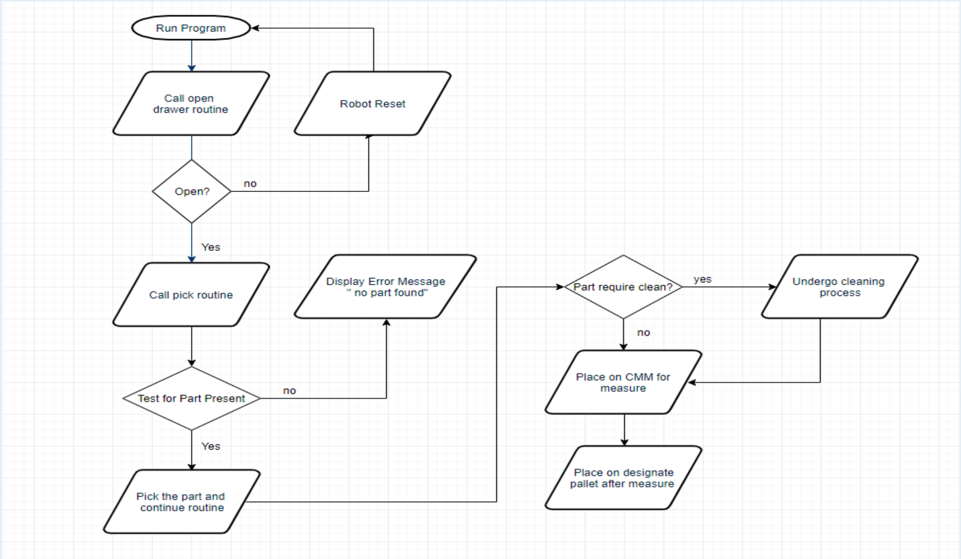

Figure 3. Robot Workflow Block Diagram

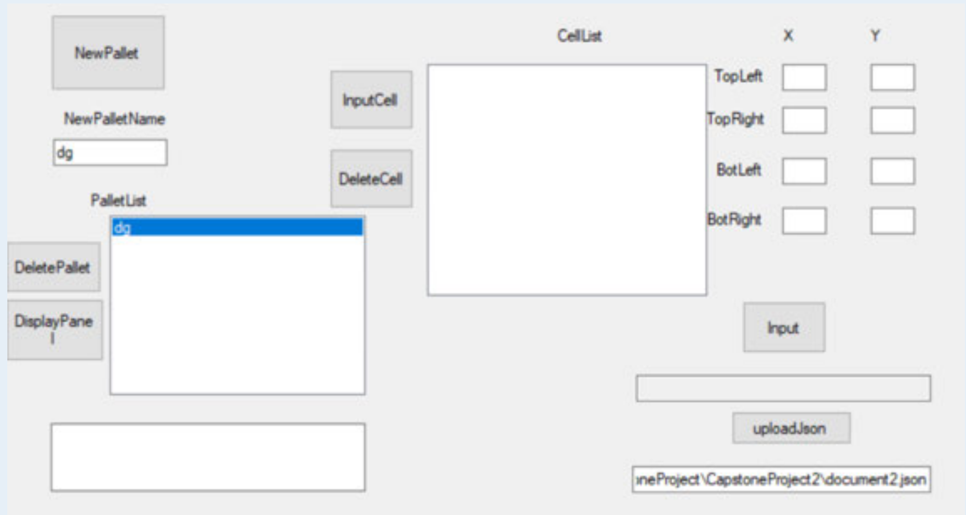

Figure 4. Pallet Layout GUI Design