Student ID: 202011724

ACW Report (Mechatronic Systems)

500680

Submission Deadline:

Friday, 22nd Dec. 2021 (2:00 pm)

UNIVERSITY OF HULL

Contents

List of Figures

Abbreviations and Symbols

CAD	Computer-aided design
RR	Revolute Revolute
RRR	Revolute Revolute Revolute
DOF	Degree Of Freedom

θ	Angle
...	...

Introduction and Background

Palletizing is the process of placing goods on top of a pallet to make it easier to transport. It makes handling and transporting the goods easier and more efficient. Palletizing became the most important logistic tool in the 20th century. Before automation workers had to manually organize boxes on top of pallets, but this was very slow. The first mechanical palletizer was designed in 1948 and it was a row-forming palletizer. The in-line palletizer was developed in 1970. Those palletizers are much faster than other alternatives. They work best for a single product style. Robotic palletizers were introduced in the 1980s(Wikipedia). Those palletizers have an arm with a tool at the end to grab and reposition objects from conveyor belts onto pallets. The advantage of a robotic palletizer is that it can receive the object at any height. They can easily operate with a wide variety of goods no matter the size or weight. They can arrange the products in any pattern and operate on multiple packaging lines due to its flexibility.

1.1 Robotic Manipulator

The task is to design a robotic manipulator capable of performing palletizing. It must be able to pick up a box from a conveyor belt and place it somewhere on a palette.

1.2 Kinematic Model

The forward kinematic model of the RR robotic arm. It finds the position of the end of the arm based on the angle of the parts of the arm, where L_1 is the length of the first link and L_2 of the second link. θ_1 and θ_2 are the angles od the first and second link.

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

The inverse kinematic model.

$$c_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$s_2 = \pm\sqrt{1 - c_2^2}$$

$$k_1 = l_1 + l_2c_2$$

$$k_2 = l_2s_2$$

$$\theta_1 = \tan^{-1} 2(\frac{y}{x}) - \tan^{-1} 2(\frac{k_2}{k_1})$$

$$\theta_2 = \tan^{-1} 2(\frac{s_2}{c_2})$$

This model is used to find the angle at which both arm links have to be to touch the x and y coordinates.
In my RRR model the link lengths are L_1 = 2.95m, L_2 = 1.95m. The third link is 0.36m long but is not needed to perform this calculation.

1.3 Workspace

The workspace used to design the palletizer is Simulink and MATLAB. MATLAB is a software used to simulate systems and compute equations. The 3d model is designed using Solidworks, which is a CAD software.

Design of an Autonomous Palletiser

My robotic arm will be a RRR 3-DOF planar manipulator. The third section of the arm will be used to align with the box and grab it from the top. It will grab the box with a scissor-like gripper.

This is the design of the robot. It uses simple joints to connect each of the parts. The order of construction is Base, Arm, Forearm, Wrist and both fingers. The dimensions below are not the lengths of the link in the kinetic model. They are there to show the rough scale. The arm has a simple design. It is a rectangular beam with smoothed out edges. Other parts connect to it via a simple revolute joint. All parts are scaled down from the base.

Figure 1: CAD model components

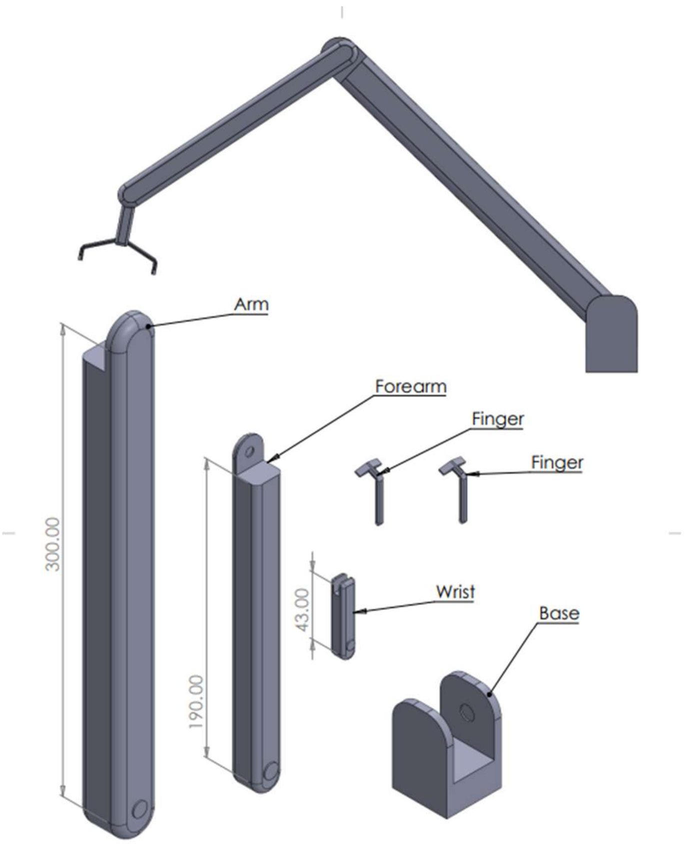
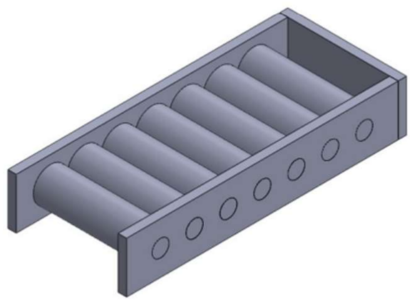


Figure 2: Conveyor belt model



This is the simplified conveyor belt design. It does not have any moving parts and is purely visual.

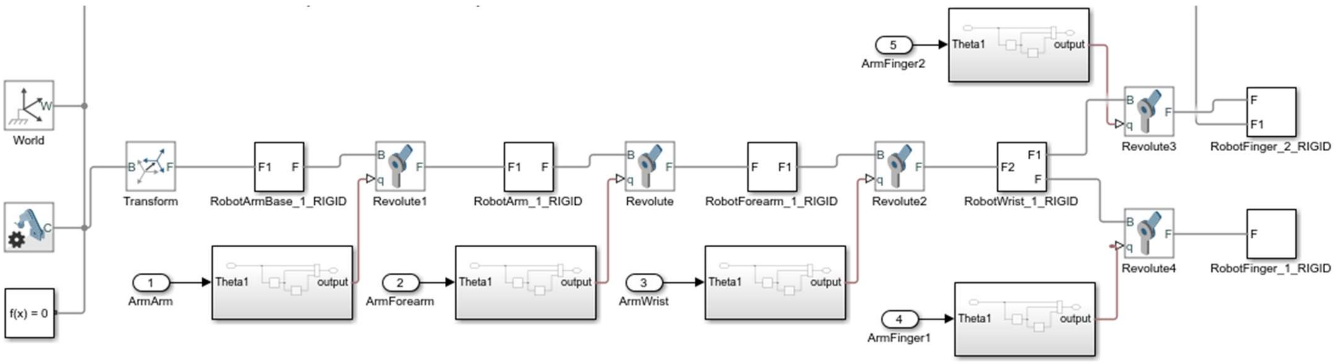


Figure 3: Robotic Arm model in Simulink

This is the Robotic arm in Simulink. It was imported from Solidworks using the Simscape multibody link plugin. Each joint has a control block which has an input. The input value is an angle at which I want the joint to be. The input value is provided from a MATLAB function file. The function file is responsible for the entire movement of the robotic arm. I used it to program a path for the arm.

```
if time < 5 %first 5 seconds from default state to grab box
Arm = deg2rad(-90+time*8.738864);
Forearm = deg2rad(16.365354*time);
Wrist = (-(-pi/2+(-Arm)+(pi-Forearm))/5)*time;
Finger1 = deg2rad(-10);
Finger2 = -deg2rad(-10);
end

if time >= 5 && time < 7  % 5-7 sec grab box
Arm = deg2rad(-46.30568);
Forearm = deg2rad(81.82677);
Wrist = -(-pi/2+(-Arm)+(pi-Forearm));
Finger1 = deg2rad(-10+5*(time-5));
Finger2 = -deg2rad(-10+5*(time-5));
end
```

This is a snippet from the MATLAB function responsible for the arm movement, it uses different time periods at which it changes its position. The first five seconds are spent moving to the state at which the arm grabs the box. The Arm and Wrist variables go from -90° to a specific angle. The angle was found using inverse kinematics knowing the desired position.

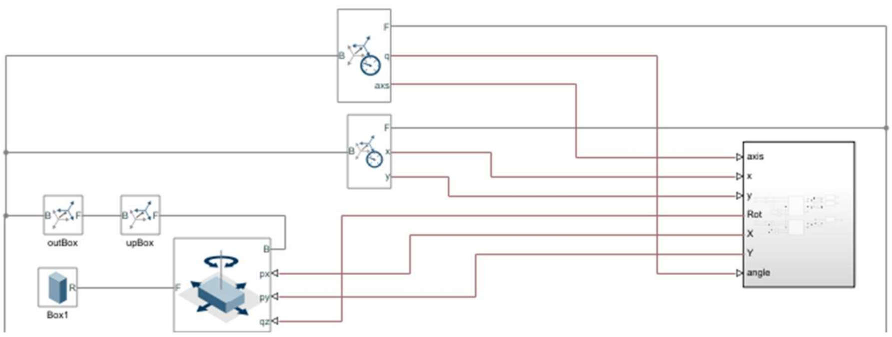


Figure 4: Box model

This is a part of the system responsible for moving the box. It uses two sensors that measure the translation of the end (one of the fingers) and apply the same translation to the box. In the subsystem is function that calculates the relative translation after the box is grabbed.

Summary

The robotic arm works very well. It moves very smoothly and precisely according to the given angle. It can reach the box and place it anywhere on the pallet. I could easily change its path by just changing the input function.
The conveyor belt does not work. This is because it has no joints. Modelling it in Solidworks was not hard. The difficult part was importing it to Simulink. The first problem was combining it with the arm model in Simulink. After importing it from Solidworks the Simulink model uses variables from a file. Each joint and transform uses the file values, so moving it to a different model would require manually inputting those values from the file to the model for each part. Another method would be to combine the assemblies in Solidworks and importing them afterwards, but this caused just as many issues with the Simulink model not importing correctly.
I also had a lot of issues with modelling the box movement. It sounds simple but it took a lot of ingenuity. As to every problem there were many solutions. One of them was giving the box, belt and robot fingers a "spatial contact force" which would allow them to interact with each other. This solution however had many issues, some of them being the inaccuracy of the simulation, extreme computing times or inaccurate mesh boxes. Another approach I found online was using an external library. The Multibody contact forces library offers a range of contact force models and laws for multibody modelling in Simscape Multibody (Steve Miller). This however also used the inaccurate mesh of the gripper fingers and had insane computation time. The solution that I went with was sensing the movement of one of the gripper fingers and moving the box along with the fingers from the moment the box is grabbed. I successfully copied the translation, but I did not manage to make the box rotate with the gripper. I think this is due to the way the rotational translation is stored.
I think the robotic arm picking up the box and the conveyor belt can be greatly improved by including objects from the Multibody contact forces library. It includes a conveyor belt system which would allow for a very realistic conveyor belt. Furthermore, the 3d models can be improved by including more detail and being more realistic The Arm design lacks detail and looks very plane.
The project can be vastly improved and fine-tuned.

References

Wikipedia (2021) Palletizer. Available online: [Accessed 10/12/2021] https://en.wikipedia.org/wiki/Palletizer

Steve Miller (2021). Simscape Multibody Contact Forces. Available online: https://github.com/mathworks/Simscape-Multibody-Contact-Forces-Library/releases/tag/21.2.5.0 [Accessed 10/12/2021]