UNIVERSITY OF MINNESOTA
Mechanical Engineering
ME 5286

Flashlight Assembly

Abstract

In this experiment, a UR5 cobot was programmed to assemble a flashlight. This was accomplished by programming the robot to pick and place the flashlight parts from a pallet to a pneumatic clamp and thread both the barrel and the tail cap to an appropriate torque of 3 Nm. We set out to achieve an assembly time of under 120 seconds, and were ultimately able to assemble the flashlight in 106 seconds. Despite meeting the assembly goals for the experiment, several accuracy improvements would be necessary before deploying the code to an assembly line, such as sensors or computer vision to improve the position information of the parts. Additionally, the assembly time could have been shortened by changing the starting position of the clamp or the parts or further optimizing the movements of the cobot. Overall, the experiment had satisfactory results.

Table of Contents

1 Introduction

In this experiment, we set out to demonstrate the capabilities of our UR5 cobot by assembling a flashlight on a pneumatic clamp from a pallet of parts. Cobots are very useful in modern workspaces due to the large amount of tasks that they are able to accomplish, the relative ease in programming them, and the safety features built into the robots to prevent harm to nearby workers. These safety features allow the robots to be used side by side with human workers without the need for safety cages. Because of these benefits, cobots are widely used in modern day industry. By programming our cobot to assemble a flashlight, the we can validate the functionality of the robots and gain experience in using them.

2 Methods

The first thing that needed to be done when assembling the flashlight was finding the X and Y positions of the parts of the assembly. These were determined by transforming the coordinates of the pallet that held the parts into the coordinate frame of the robot base. The height that the gripper needed to be at for each part also needed to be determined. This was accomplished by using free drive to move the robot to the precise locations for picking up and setting down the parts and recording the respective z values in the base coordinate system.

In order to transform the pallet locations to the coordinate frame of the robot, a homogenous transformation was used. To do this, we started with the locations of each of the part (in pallet coordinates starting at 0,0 for the lower left position) and converted them to polar coordinates, with a radius and angle value for each coordinate:

$$r = \sqrt{(x^2 + y^2)} \quad \theta = \mathrm{arctan2}\left(\frac{\mathrm{y}}{\mathrm{x}}\right)$$

Note that the use of arctan2 is important for this algorithm since it allows for negative x and y values for ball locations. Then, the angle of rotation was simply added to each of the theta values before converting the points back into cartesian coordinates:

$$\theta_1 = \theta + \Delta\theta \qquad x_1 = r * \cos(\theta_2) \qquad y_1 = r * \sin(\theta_1)$$

Finally, the x and y offset (or the location of the origin) was applied to each of these rotated cartesian points:

$$x_2 = x_{offset} + x_1 \qquad y_2 = y_{offset} + y_1$$

Our algorithm was consolidated into a function, which was then used to calculate all positions for both pallets.

The first step in actually assembling the flashlight was moving the head of the flashlight into the pneumatic clamp. This was fairly straightforward since the coordinates of the pick and place were already defined. In order to firmly grab the part without damaging it, the gripper was lined with packaging tape and set to the maximum force value of 255. Additionally, the end effector was moved to a height of 50 millimeters above the part before picking and placing the part and, after picking up the part, was moved straight upwards before moving over to the clamp. This ensured that the gripper and the held part did not accidentally hit any of the other parts or the clamp. Once the head was dropped into the clamp, the clamp was activated to hold it in place.

Next, the barrel was picked and placed on top of the head. After the barrel was dropped, the gripper was reactivated to clamp the now set barrel part, and a custom threading function was run to complete the first five full turns of tightening the barrel. This function set the joint acceleration of the robot to the maximum value of 360 rad/s^2, completed one full turn of the gripper joint, then released the gripper and undid the turn before reactivating the gripper and completing another full turn. This action was repeated for the number of turns passed to the function before resetting the joint position to its default value and setting the joint speed back to its initial value of 100 rad/s^2. The turning and "unturning" of the gripper were handled by subfunctions to keep the code as clean and understandable as possible. These settings were chosen to maximize the speed of the robot while threading the flashlight and minimizing the shaking of the table, since running the robot at full speed and acceleration while the base joint was moving showed the highest amount of movement in the table. After the custom function was run to do most of the threading, the provided "tighten_torque" function was run. This function finished threading the barrel, stopping once a torque value of 3 Nm was reached. The start and end angles for this function were set to 0 and $\frac{\pi}{2}$ radians, respectively. The joint speed and acceleration and gripper speed and acceleration parameters were set to 2, 2, 100, and 100 respectively. The gripper open parameter was set to 50%, though a higher value could have been used without any issues. It was important throughout the threading of the barrel to ensure that accurate X and Y values were used, since diagonal threading of the barrel could cause damage to the part.

After the barrel was threaded, a simple pick and place of the battery pack was accomplished. Early on in the testing of the assembly, we found that the battery pack was difficult to pick up when it was laying on its side in the pallet. To resolve this, a part was 3D printed to hold the battery in a standing position. This also simplified the placement of the battery pack in the barrel since it didn't need to be rotated before it was placed. This part can be seen in Figure 1, below.

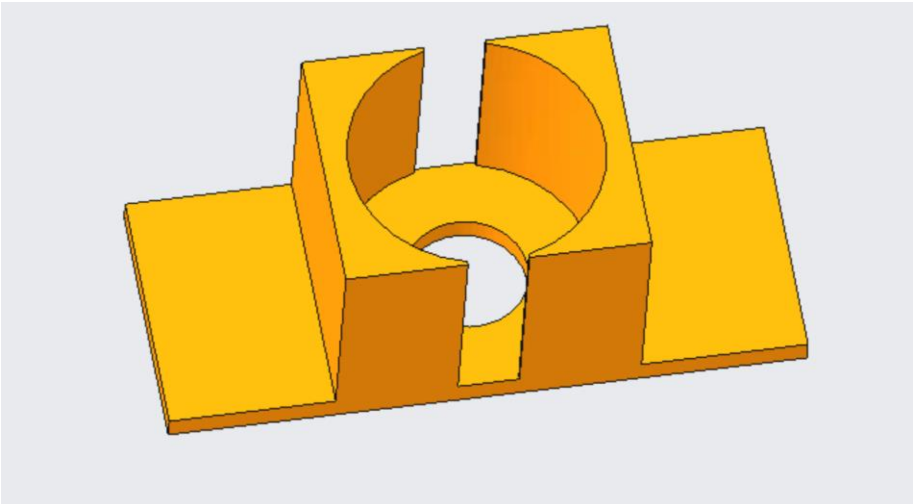

Figure 1: 3D printed battery pack holder

The last part that needed to be assembled was the tail cap of the flashlight. This part needed to be threaded onto the barrel of the flashlight, but had the added difficulty of needing to resist the spring force of the battery pack. To negate this, the threading of the part was started without releasing it when it was placed on the barrel. The same custom function was used for the tail cap as was used for the barrel, with 3 turns being completed before the tighten torque function finished the threading. The same parameters were passed to the function, threading the tail cap to a torque of 3 Nm. This differed from the recommended value of 2 Nm for this step, but we found that 3 Nm provided a more solid construction of the flashlight.

Finally, the assembled flashlight was removed from the gripper and placed in the position of the pallet that the flashlight head was initially placed by grabbing the flashlight, releasing the clamp, and slowly moving it over to the pallet.

3 Results

Figure 2: Placement of the flashlight head into the clamp

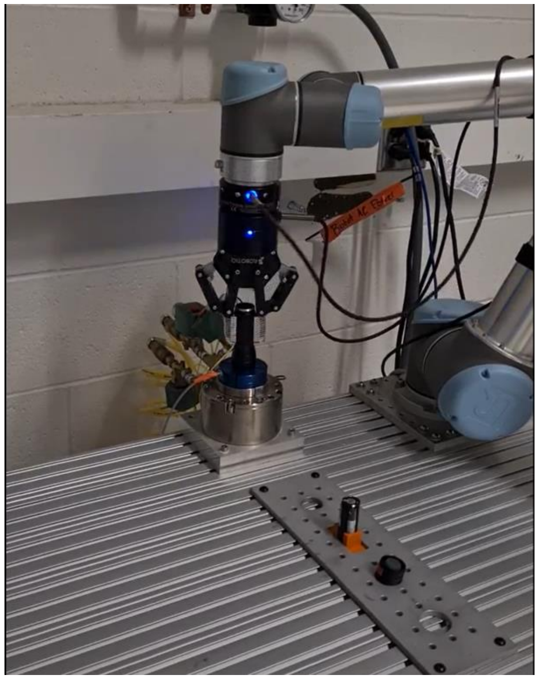

Figure 3: The gripper in between turns while threading the flashlight barrel

Figure 4: The battery pack being picked up from the custom holder

Figure 5: The robot threading the tail cap onto the end of the flashlight

Figure 6: The assembled flashlight after being placed on the head position of the pallet

Target Assembly Time	120 seconds
Actual Assembly Time	106 seconds

Table 1: Target assembly time compared to actual assembly time of the flashlight

4 Discussion

Our programming of the robot provided a more than adequate performance in assembling the flashlight, since the assembly took just 106 seconds, 14 seconds less than the target time of 120 seconds, and was consistently successful in operation after the addition of grip tape. However, I do not think that this would be adequate for industrial flashlight assembly for this flashlight for a few reasons. The first of these reasons is that the positions of the parts on the pallet were programmed directly into the code, something that would not likely work consistently if the parts were being brought in on a conveyer system of some kind (there would likely be some amount of error in the position of the belt). To negate this, some type of corrective positioning system would be required. This system could utilize proximity sensors or computer vision to accomplish this.

There are several ways that the speed of the flashlight assembly could be improved without getting a faster robot. The first way that we could do this is by changing the location of the pallet to be closer to where the clamp is. This, perhaps combined with a tighter layout of the flashlight parts, would create less overall movement for the robot. Another improvement that could be made to the assembly process would be starting with the battery inside of the barrel. This would eliminate an entire pick and place operation and save a significant amount of time. Lastly, budget permitting, we could implement a clamping mechanism into each of the pallets on the position of the head piece. This would allow the assembly to happen directly on the pallet, reduce the amount of movement that is required by the robot, and eliminate the need to pick and place the final flashlight assembly back onto the pallet. This, however, may prove difficult since pneumatic systems need to be connected to an air line to operate, meaning that we may need to connect and disconnect the pneumatic line or create moving pneumatic lines on a larger, fixed pallet. It also may be possible to create a clamp that could rotate continuously to improve the speed of threading, but this would be a fairly advanced system and may require some research and development work to implement. With a combination of some or all of these improvements, the assembly time of the flashlight could be significantly reduced.

5 Conclusion

Overall, the assembly of the flashlight using the UR5 robot went very well. We were able to assemble the flashlight without damaging it in just 106 seconds, well below the target time of 120 seconds. Despite this, the assembly process could be optimized further for even lower assembly times. One improvement that could have saved some time would be tuning our custom turning function to do more than one turn at a time before resetting the jaws. The gripper cables limit this somewhat, but with some trial and error it’s likely that more than a single turn could be completed without tightening the cables too much. This would improve the turning speed since we wouldn’t have to wait as much for the jaws to open and close when turning. Another optimization we could’ve made to our flashlight assembly was fine tuning the tighten_torque function to end its turn at the right point, since it had to spin back into the right position before picking up the next part. We also could have spent more time optimizing the movements of the robot to take the shortest viable path to the clamp, which would reduce the overall movement time of the robot. Regardless of optimizations that could have been made, we were overall very happy with the results of the robot code, especially since it was written in such a short amount of time. The experiment demonstrated the versatility and capabilities of the UR5 cobots that we used in the lab, indicating the multitude of different tasks that they can accomplish with a relatively short amount of preparation time.