UNIVERSITY OF TECHNOLOGY, SYDNEY

University of Technology Sydney (UTS)
P.O. Box 123, Broadway, Ultimo, N.S.W 2007
Australia

Report

Assignment 1 - Dual Cobot Assembly

Robotics (41013) – Autumn 2021

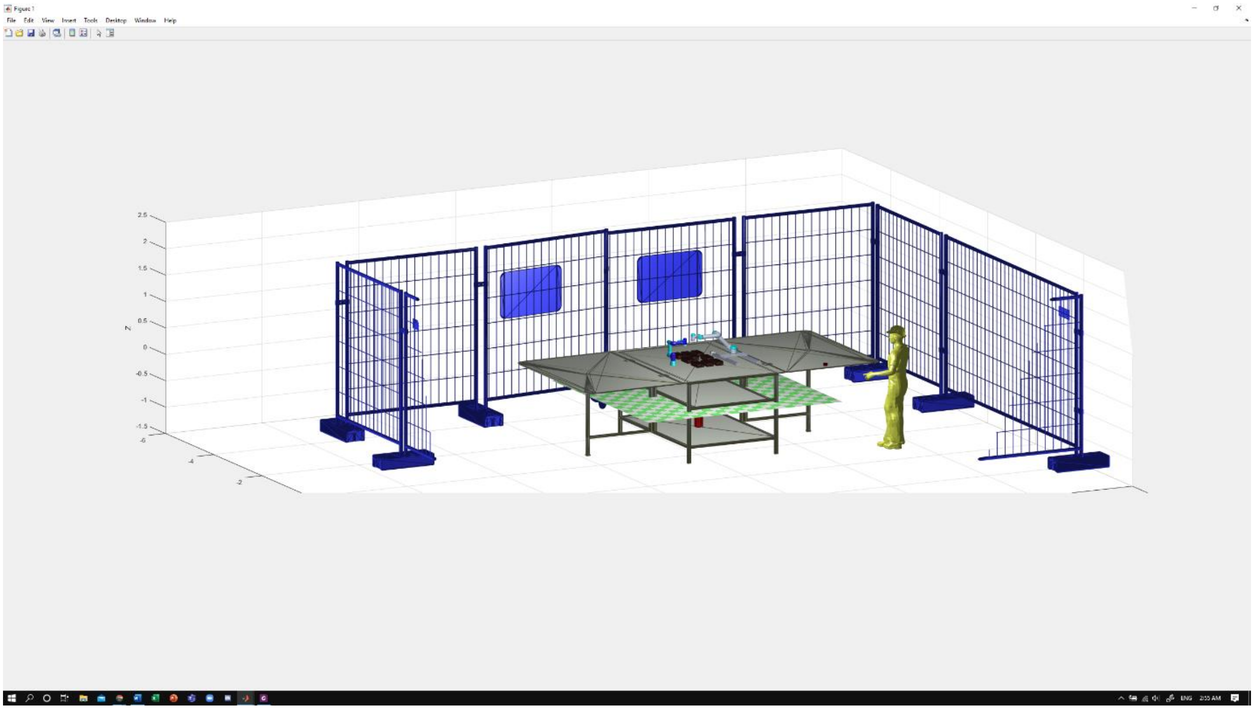
Student Name: Huy Nhat Minh Nguyen

Student ID: 13734569

March 2021

Table of Contents

1. Design the workspace (with the aid of figures) so that someone else can install the two robots including engineering considerations.



The relative location of the environment objects is as follows:

- Fence: `transl(0,0,-0.25)`
- Table: `eye(4)`
- Lab supervisor: `transl(-3,2,0.2)`
- Emergency button: `transl(-2,1.3,0.26)`
- Fire extinguisher: `transl(0,0,-0.5)`
- The UR3: `transl(0.5,0,0.25)`
- The Linear UR5: `transl(-0.75,-0.3,0.25) * trotx(pi/2)`

2. Include considerations of safety in terms of a basic risk assessment that identifies hazards and their likelihood of occurrence, and Safe Work Method Statement.

I have submitted the Risk Assessment form and the Safety Work Method Statement form via Canvas.

3. Reflect upon why a UR3 OR Sawyer is a good choice, whilst also investigating what sort of limitations you expect to encounter with the use of this robot.

<u>a. Suitability and convenience of using UR3 or Sawyer</u>

- The reasons of using UR3 (or the Sawyer) for this assignment are:

+ UR3 has 6 degrees of freedom. It allows its end-effector come to almost anywhere in its workspace, make it easily finish the task.

+ UR3 is a human-robot interaction. The UR3 is designed to receive a response every time it collides with a person or object and will stop working immediately to ensure safety.

b. Limitations

- Honestly, I have some difficulties when using UR3 for this task because of its limited range and limited workspace. It would be more convenience if I would use 2 UR5s for this task. UR5 is the most optimal choice with the suitable range and payload.

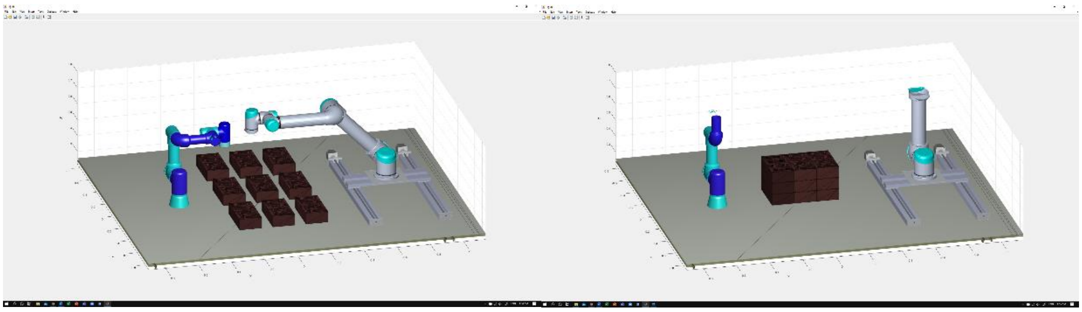
4. Reflect upon the usage of MATLAB and the robotics toolbox

- C++ and MATLAB can be categorized as "Languages" tools. "Performance" is the primary reason why developers consider C++ over the competitors, whereas "Simulink" was stated as the key factor in picking MATLAB, and this is the main reason for using MATLAB for this assignment. Honestly, I prefer to use MATLAB because of its convenience in debugging. When I code with C or C++, I usually get stuck with programming errors, but when I use MATLAB, it clearly shows the errors, for example: *"Too many output arguments", "Unrecognized property for Class"*…, therefore I can easily debug my code. Another good thing is that MATLAB can print the results while running (such as log file, command window…). This helps me keep track of my code process, making it easy for me to detect errors during my calculation.

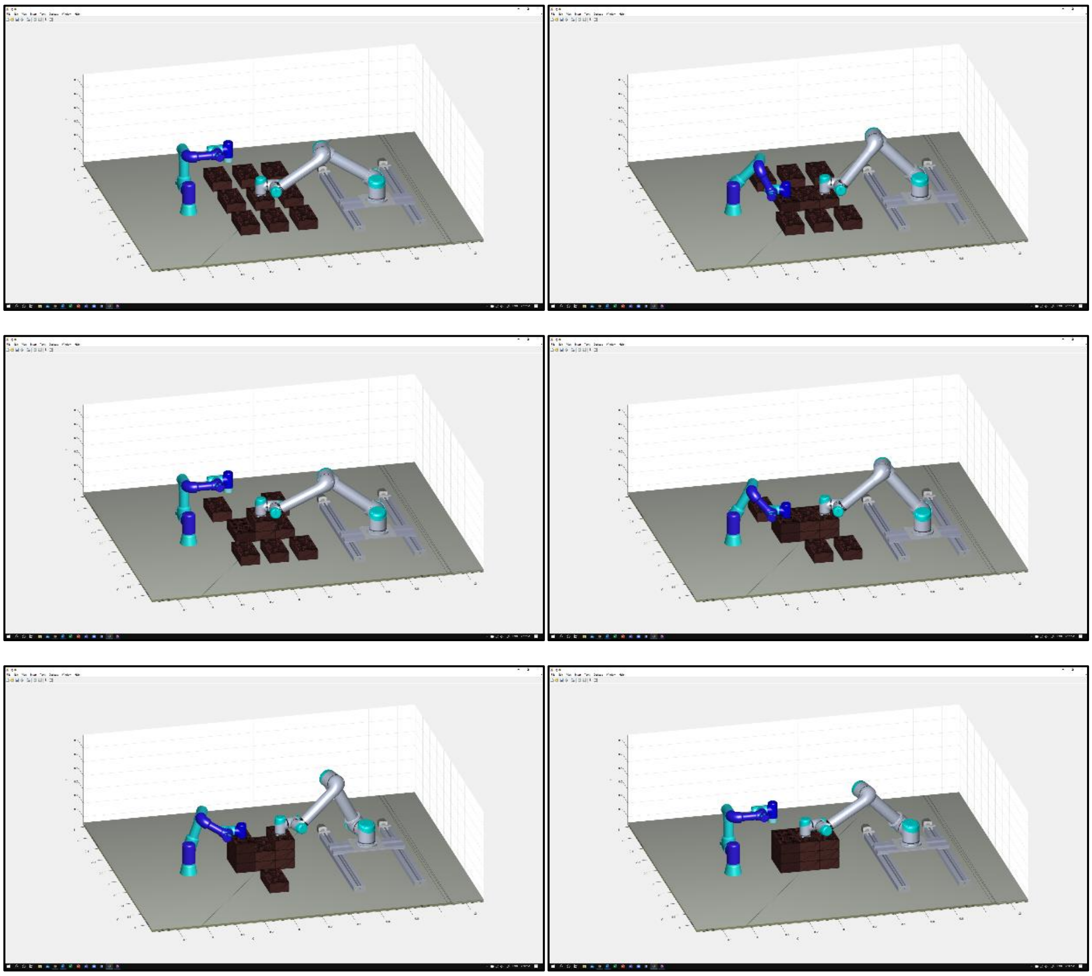
- About the robotics toolbox, using Peter Corke's toolbox makes it simpler to code for the assignment. For example, I can create a 4x4 homogeneous transform by using toolbox function `trotx(angle)` or `transl(x,y,z)` instead of MATLAB function `makehgtform('zrotate', angle)` or `makehgtform('translate', [x,y,z])`. Besides, Peter Corke's toolbox has many helpful functions to convert between many types of transform, for example: function `tr2rpy` converts a homogeneous transform to roll-pitch-yaw angles, function `tr2rt` converts homogeneous transform to rotation and translation, function `r2t` converts rotation matrix to a homogeneous transform,… Furthermore, Peter Corke's toolbox provide the `jtraj` function, which helps to create a trajectory matrix with just only one line of code.

5. Reflect upon the likely precision required, and thoughts about how you would go about controlling the robot.

- With the task assignment, I am going to build a 3x3 wall with 9 bricks (random location) as 2 figures below. I will use both 2 robots, the UR3 and the Linear UR5 to finish this task.



- With a longer arm and a linear rail, the UR5 robot will have a broader range than the UR3 robot. Therefore, the UR3 robot will handle the three bricks of the first column on the left, and the UR5 robot will be tasked with moving the remaining six bricks of the second and the third column (assume that there are small positional changes (less than 200 mm) to the bricks).
- Firstly, the UR5 will move the first brick to the center of the first row of the wall. I put this task in an `if` function, in case the first brick is already in a drop-off position then the robot would not need to do this action. After that, both 2 robots will finish the first row of the wall by moving 2 bricks to the side of the first brick.
- The second and the third row are constructed in the same manner as the first row.



6. Reflect upon the likely challenges of sensing and grasping that you have been allowed to assume are solved problems

- While controlling the robots to do the task, I had to face some challenges of grasping the brick.
- The first problem is when I tried to create a trajectory to pick up the bricks, the robots will move in a trajectory going up from below the table surface. This happened because the z-axis of the brick's transform is pointing up, but the z-axis of the end-effector's transform should be pointing down. Therefore, I had to create an intermediate transform, having the same position as the brick and in the same x, y-direction but in the opposite z-direction. In this way, the z-axis of the end-effector is always pointing down when grabbing the bricks, thus avoiding collisions with the table.
- In building brick wall, when moving the bricks from the table to the second row and the third row of the wall, bricks that are being moved will likely collide with bricks that are in the wall. To solve this problem, I have built a pick-up process for each brick for the robots: default pose → prepare pick-up → pick-up → prepare pick-up → prepare drop-off → drop-off → prepare drop-off → default, in which prepare pick-up and drop-off position is 0.3 meters upwards from the pick-up

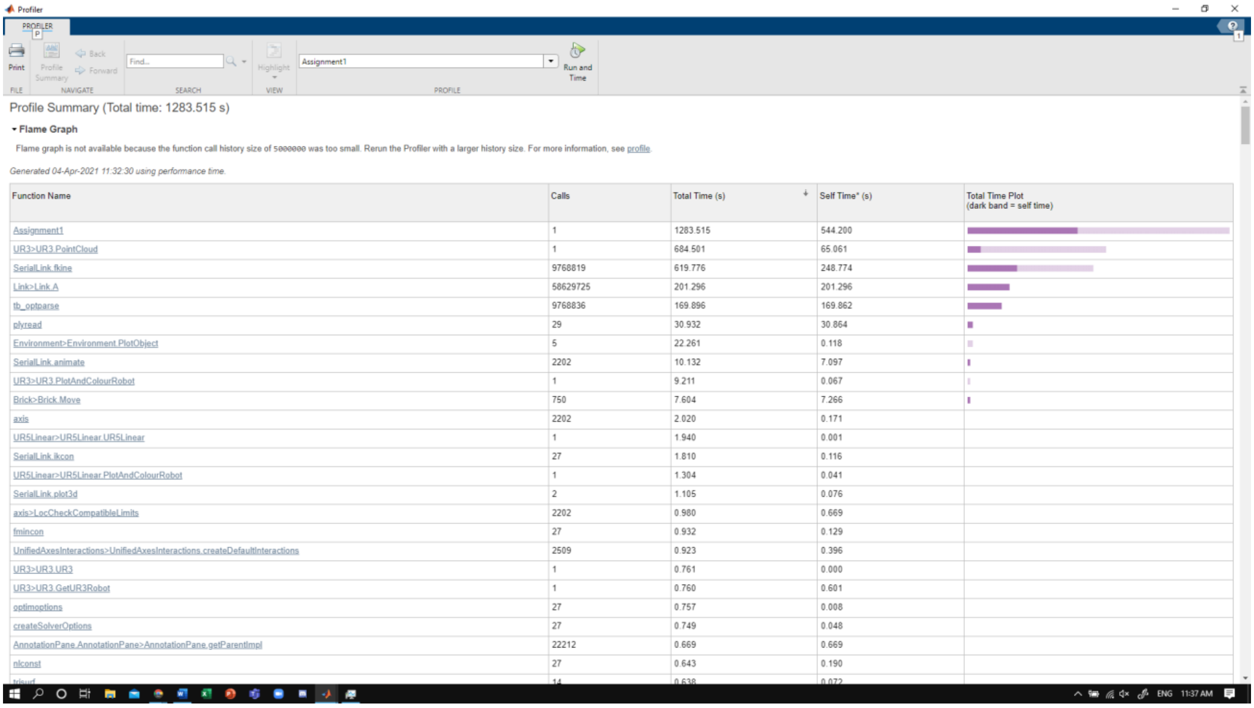
and drop-off position. This will prevent the robots or bricks being moved from colliding with the building wall.

7. Reflect upon the safety of the movement of the robots, such as considering if they may collide with each other or objects in the environment.

- There are several problems that the robots might collide with objects in the environment, such as the table or even the wall bricks.
- The most common problem would be that there will be times when the robot will collide with the table during the task. This happens when the robots are performing the action of picking the brick and are in a state known as "elbow down." This reason of the problem is when I use the ikcon function to solve the state q of each joint of the robots, there will be many solutions q that satisfy the required end-effector, but the solution is not as expected. To figure out this problem, I have to optimize the ikcon function by using the q0 as an initial joint for minimization. This will help the ikcon function solve the solution following my expectations, keep the robots in an "elbow up' state, and avoid collision with the table.

8. Use MATLAB’s “profile” tool to investigate which functions consume most of the time. Reflect briefly on alternative methods to improve computational efficiency.

- After using MATLAB’s profiler tool, the results are shown below.


Profile Summary (Total time: 1283.515 s)

Flame Graph

Flame graph is not available because the function call history size of 5000000 was too small. Rerun the Profiler with a larger history size. For more information, see profile.

Generated 04-Apr-2021 11:32:30 using performance time.

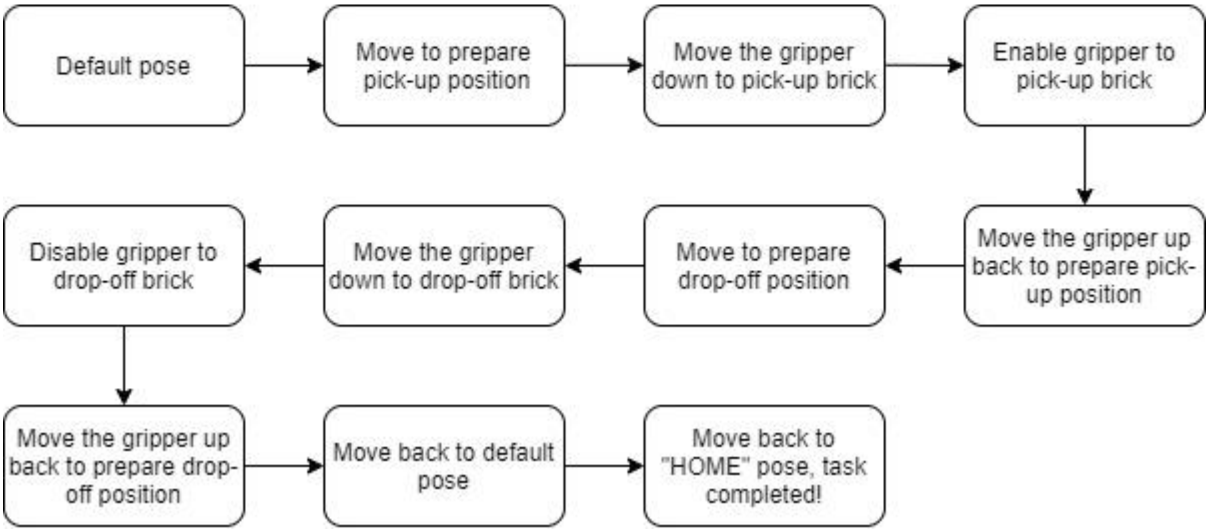
Function Name	Calls	Total Time (s)	Self Time* (s)	Total Time Plot (dark band = self time)
Assignment1	1	1283.515	544.200	
UR3>UR3.PointCloud	1	684.501	65.061	
SerialLink.fkine	9768819	619.776	248.774	
Link>Link.A	58629725	201.296	201.296	
tb_optparse	9768836	169.896	169.862	
plyread	29	30.932	30.864	
Environment>Environment.PlotObject	5	22.261	0.118	
SerialLink.animate	2202	10.132	7.097	
UR3>UR3.PlotAndColourRobot	1	9.211	0.067	
Brick>Brick.Move	750	7.604	7.266	
axis	2202	2.020	0.171	
UR5Linear>UR5Linear.UR5Linear	1	1.940	0.001	
SerialLink.ikcon	27	1.810	0.116	
UR5Linear>UR5Linear.PlotAndColourRobot	1	1.304	0.041	
SerialLink.plot3d	2	1.105	0.076	
axis>LocCheckCompatibleLimits	2202	0.980	0.669	
fmincon	27	0.932	0.129	
UnifiedAxesInteractions>UnifiedAxesInteractions.createDefaultInteractions	2509	0.923	0.396	
UR3>UR3.UR3	1	0.761	0.000	
UR3>UR3.GetUR3Robot	1	0.760	0.601	
optimoptions	27	0.757	0.008	
createSolverOptions	27	0.749	0.048	
AnnotationPane.AnnotationPane>AnnotationPane.getParentImpl	22212	0.669	0.669	
nlconst	27	0.643	0.190	
trisurf	14	0.638	0.072	

- As the results above, the two parts consume most of the time in my code are `UR3.PointCloud` to plot the 3D model showing where the end effector can be over all these samples, and `plyread` to plot environment objects to the workspace. To improve the computational efficiency, when plotting the environment objects to MATLAB, I have to use *blender decimate* to reduce the faces and vertices of the model. This will greatly improve the timing of plotting environment objects to MATLAB. With the `UR3.PointCloud`, the reason this function takes so long is because of the `for` loop. There are 6 joints to be calculated, I even used 30 deg step instead of 1, that means there are $\frac{360}{30} = 12$ (steps) each joint, and overall $12^5 = 248{,}832$ steps for the robot. Even so, I do not think step should be increased to 45 instead of 30, since this would lead to error in calculating volume using `convhull` function, and the sample would be more coarsely. Therefore, the best way to improve this problem is we should assume that joint 3 and 4 should be fixed at 0 degree, then we just have $12^3 = 1{,}728$ steps for the calculation.

9. Given the brick models, select an appropriate gripper and discuss with the use of diagrams the ideal way to grasp the objects.

- Since the brick is rectangular in shape, it is impossible to use the 2-claw gripper because if the center of the brick is not grasped correctly, there is a high chance of it being dropped during moving. Therefore, I propose choosing a "carbon vacuum gripper", which can exert force on the entire brick, which will make the moving process more accessible and safer.

- Diagrams the ideal way to grasp the objects:



- In which prepare pick-up and drop-off position is 0.3 meters upwards from the pick-up and drop-off position. This will prevent the robots or bricks being moved from colliding with the building wall.

10. Include an adequate reference list

https://www.universal-robots.com/plus/products/#/0101111111111111111?

https://www.universal-robots.com/plus/urplus-components/handling-grippers/carbon-vacuum-gripper-cvgc-series/