Assignment #2 – Path Planning – 20 points of your final grade.

Your second assignment follows. You may complete this assignment using any of the following programming languages: MATLAB, Python, C++, C#, C, or Visual Basic (.NET). If you wish to complete this work in another programming language, please contact me by email or WeChat.

Your work should be submitted as an executable file (*.exe), as well as source code. If you choose to program in MATLAB, you may submit an *.m file instead of an executable. If you have trouble creating an executable, you may submit your source code alone, as long as it is clear what program to run. This may result in errors due to different versions of compilers, so I do not suggest this.

Submit your work (executable and source code) as a single compressed file (such as ZIP) via the course website by 19.5.2021, 23:59 (China Standard Time).

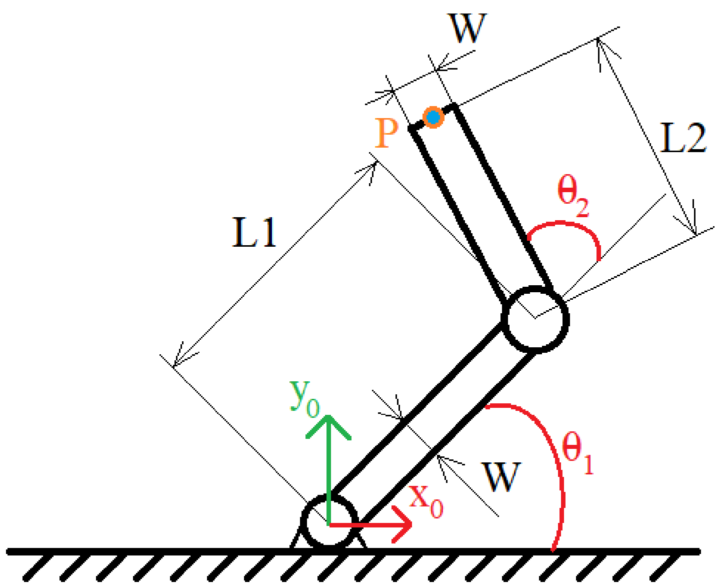
Important note: There are two questions in this assignment. You must only complete one. The second question is significantly more difficult than the first, but if you successfully complete it you may earn up to 10 bonus points (i.e., if you solve the second question correctly you will receive 30 points to your final grade). I suggest that you only attempt the second question if you feel very confident in the course material and in your chosen programming language.

If you have any questions, please contact me by email or WeChat.

The assignment is as follows:

Question 1:

We wish to navigate the following robotic arm from a Start to Target point, without colliding with any objects.



The world frame (x0,y0) is centered in the first joint, as drawn in the figure. The end of the robot P is the tip of the second link. The robot is at the Start position when point P equals the Start point, etc.

The inverse kinematics solution for a two link, two revolute-joint arm is as follows:

$$\theta_1 = \arctan 2(y, x) \pm \arccos\left(\frac{L_2^2 - L_1^2 - x^2 - y^2}{-2L_1\sqrt{x^2 + y^2}}\right)$$

$$\theta_2 = 180 \pm \arccos\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{-2L_1L_2}\right)$$

Where x,y are the goal coordinates.

Note that there are two solutions (not four).

You will need the forward kinematics equations to determine the joint values, given x, y coordinates.

You must create a computer program that accepts a comma separated value file (*.csv) of parameters, and returns a new comma separated value file of joint angles. Example input and output files are provided in the course website.

The input file is called “input.csv”, which I will place in the same folder as your program. When checking your work, I will use my own input file, which is similar to the example input file but with different parameters. Make sure your program works for variable inputs!

The input file includes the following data: Robot dimensions (L1, L2, W), a list of point-obstacles (obstacles that are a single point) in physical space (x,y), A single Start point and a single Target point, in physical space (x,y).

The output file is a list of joint angle pairs, that if sequentially given to the robot, it will navigate from Start to Target without colliding with any obstacles. Use degrees as your units.

To perform the path planning in configuration space, you may use either the RRT algorithm (or any of its variants) or the PRM method (or any of its variants). If you are unfamiliar with graph search algorithms such as Dijkstra or A*, I suggest you implement RRT.

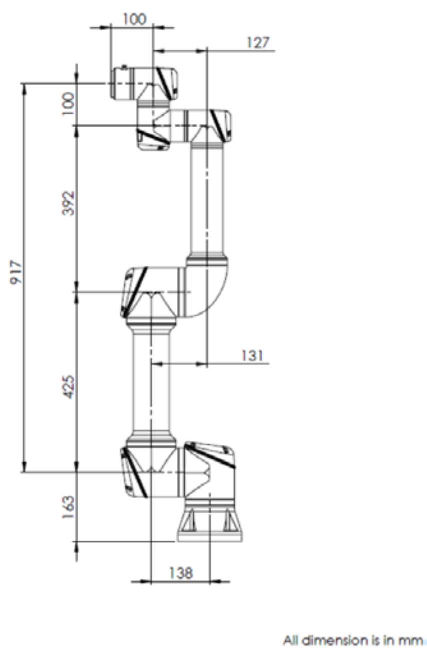
The output file should be named “output.csv”, and be in the exact format of the example file on the course website. The file should be saved either in the same folder as the executable, or at a user-defined location.

If your program does not work, points will be deducted from your grade, and I will ask you to resubmit a working program. Please make sure that your program works on a computer other than the one you used to write the program.

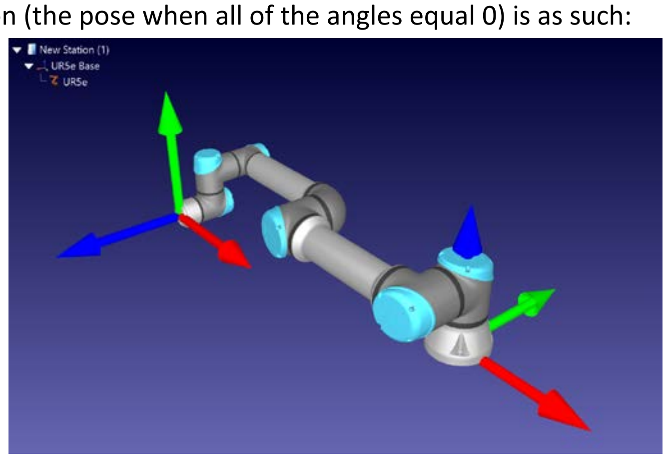
OPTIONAL Question 2:

Reminder- If you solved Question 1, you are not obligated to solve this question as well. Only attempt to solve this question if you feel comfortable with the course material and your programming language. This question is significantly harder than the previous question, but has a maximum score of 30 points (instead of 20).

We wish to navigate the following Universal Robotics UR5e robotic arm from a Start to Target point, without colliding with any objects.



The arm's zero-position (the pose when all of the angles equal 0) is as such:



The world frame (x0,y0,z0) is centered in the robot base, as drawn in the figure. The end of the robot P is the tip of the last link, as drawn.

You must create a computer program that accepts a comma separated value file (*.csv) of parameters, and returns a new comma separated value file of joint angles. Example input and output files are provided in the course website.

The input file is called "input_6dof.csv", which I will place in the same folder as your program. When checking your work, I will use my own input file, which is similar to the example input file but with different parameters. Make sure your program works for variable inputs!

The input file includes the following data: a list of point-obstacles (obstacles that are a single point) in **physical** space (x,y,z), A single Start point and Target point, in **configuration** space ($\theta_1, \theta_2, \ldots, \theta_6$).

The output file is a list of joint angles, that if sequentially given to the robot, it will navigate from Start to Target without colliding with any obstacles.

To check collisions between the robot and the obstacles, I suggest either approximating the robot as a set of cylinders, or as a set of spheres. This is a conservative assumption, but can be considered reasonable.

To perform the path planning in configuration space, you may use either the RRT algorithm (or any of its variants) or the PRM method (or any of its variants).

The output file should be named "output_6dof.csv", and be in the exact format of the example file on the course website. The file should be saved either in the same folder as the executable, or at a user-defined location.

If your program does not work, points will be deducted from your grade, and I will ask you to resubmit a working program. Please make sure that your program works on a computer other than the one you used to write the program.

Good luck!