م. باهر خير بك

الكلية	الهندسة
القسم	الروبوت و الانظمة الذكية
اسم المقرر	نظم التشغيل
تاريخ الامتحان	7/7/2025

الاختبار العملي / الفصل الثاني / العام 2025	
اسم الطالب	
الرقم الجامعي	
مدة الامتحان	120 minutes

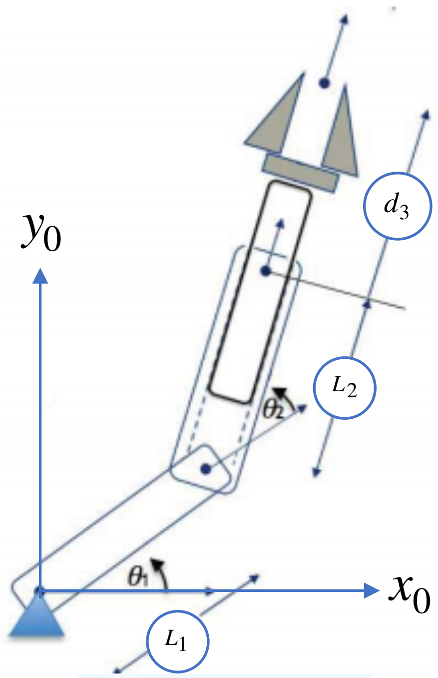
Notes before starting:

Your packages should be created inside a new workspace which should be named “<your_first_name>_ws" and within it add a student.txt file that contains your full name as well as your student_id. Name the package p1 for your project.

```
cd
mkdir -p <FirstName_LastName>_ws/src
cd <FirstName_LastName>_ws
echo “<your full name> <your_student_id>” >> student.txt
catkin_make
cd src
catkin_create_pkg p1 std_msgs rospy roscpp urdf xacro
cd ..
catkin_make
```

Problem (20 points)

Create a urdf model for the following RRP Planar Robot Manipulator using Xacro (ignore the end effector).



Link	Length (cm)	Radius (cm)
$Link_1$	100	10
$Link_2$	100	10
$Link_3$	130	5

Joint	Type	Joint Limit	
		Minimum (rad, cm)	Maximum (rad, cm)
j_1	Revolute	$\frac{-\pi}{2}$	$\frac{\pi}{2}$
j_2	Revolute	$\frac{-\pi}{2}$	$\frac{\pi}{2}$
j_3	Prismatic	35	40

All the Values within these tables should be present in a variables.xacro file and included in the robot.xacro file. All Rotations and Translations should be along the Z axis.

Create an end_effector_pose.py node that subscribes to the /joint_states and computes the end effector position and publishes the position to an /end_pose topic and prints it. Use any message you see fit for the topic (you can use geometry_msgs/Point). Note these equations which are needed to extract the position of the end effector

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} L_1 cos(\theta_1) + r cos(\theta_1 + \theta_2) \\ L_1 sin(\theta_1) + r sin(\theta_1 + \theta_2) \\ 0 \end{pmatrix}$$

r is the length from j_2 to the end of $Link3$

Then, create a marker.py node that visualizes a marker (sphere) at the position of the computed end effector. The node should subscribe and publish to all relevant topics to achieve this.

Note: set the frame_id of your marker to your fixed_frame (base_link)

The color of the end effector will be extracted from a params.yaml file. Set their initial values to whatever you want. The marker will constantly get the R, G, B values from the parameter server before publishing the marker.

RGBA : list

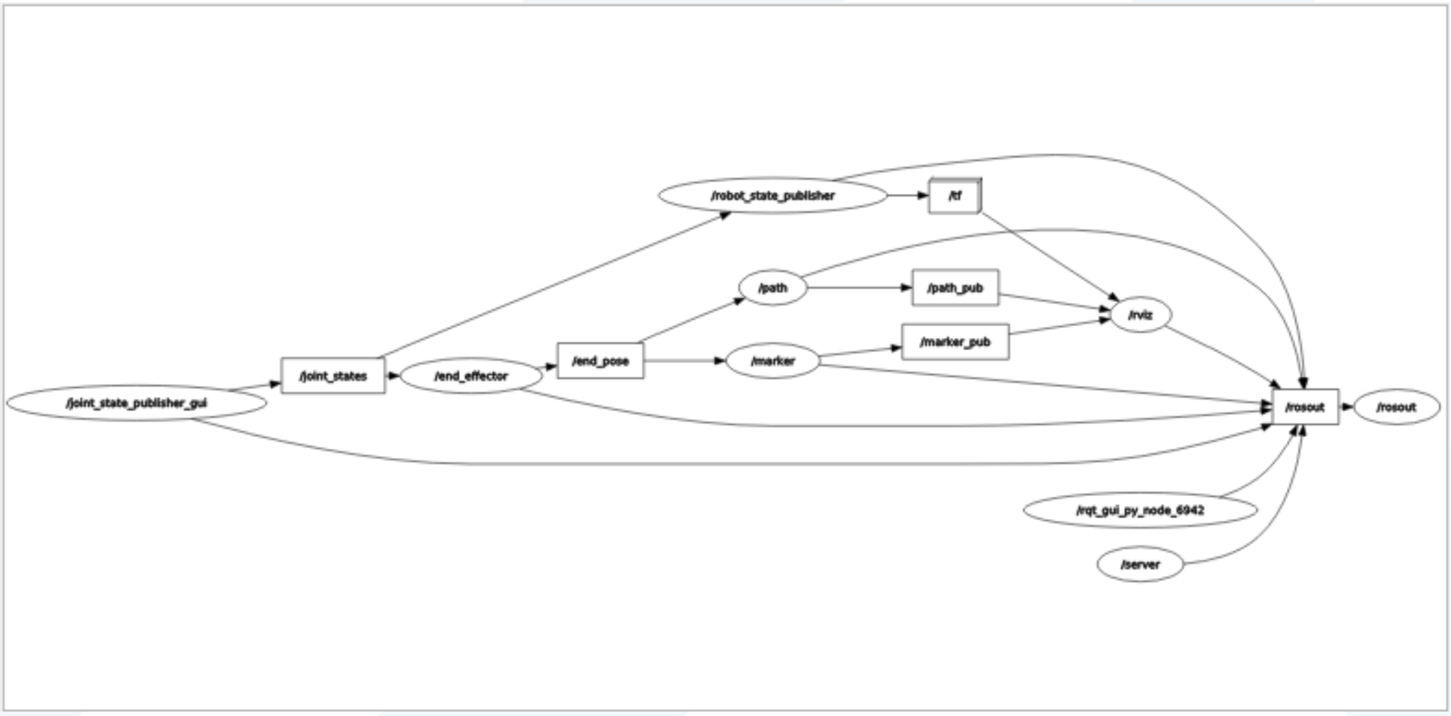
Then, create a path.py node to visualize the task space of the robot by drawing the robot's trajectory (the position that the end effector is traveling along)

Finally, create end_color.srv file that accepts r, g, b parameters as a service request, and a server.py node that updates the parameter server accordingly when the service is called. The marker should also change its color whenever the service is called upon with a specific request for the r, g, b, values.

Create a world.launch file that will bring up and load the entire project (nodes, robot_state and joints_state publishers, rosparams, Rviz config file)

Note: when adding the end_effector_pose.py node to the launch file, add the argument <output=”screen“> in case you want to see the output of the node in the terminal window where the world.launch was called.

This is the rqt_graph for the entire project



Good Luck
