21AIE213

ROBOTIC OPERATING SYSTEMS & ROBOT SIMULATION

TITLE:

Tutorial for Manipulator Control

Name: Ch. Durga Priya

Reg No: BL.EN.U4AIE21045

AIE- D section

COURSE INSTRUCTOR:

Mr. Nippun Kumar A.A., Dept. of CSE, ASEB.

Amrita Vishwa Vidyapeetham
Amrita School of Engineering
Bangalore – 560035.

CONTENTS

PROBLEM STATEMENT

Generate a beginner-level tutorial for using the “Movelt2” tool for controlling a manipulator in ROS2. You can use the “Couger bot” manipulator for controlling.

EXPLANATION

The Couger Bot manipulator is a robotic arm resembling a human arm, capable of performing various tasks. In this Couger Bot is used for controlling, guiding it to write alphabetic letters. It is equipped with joints and links that allow it to move in a manner similar to a human arm. The robot's design makes it suitable for tasks that require dexterous manipulation.

IMPLEMENTATION

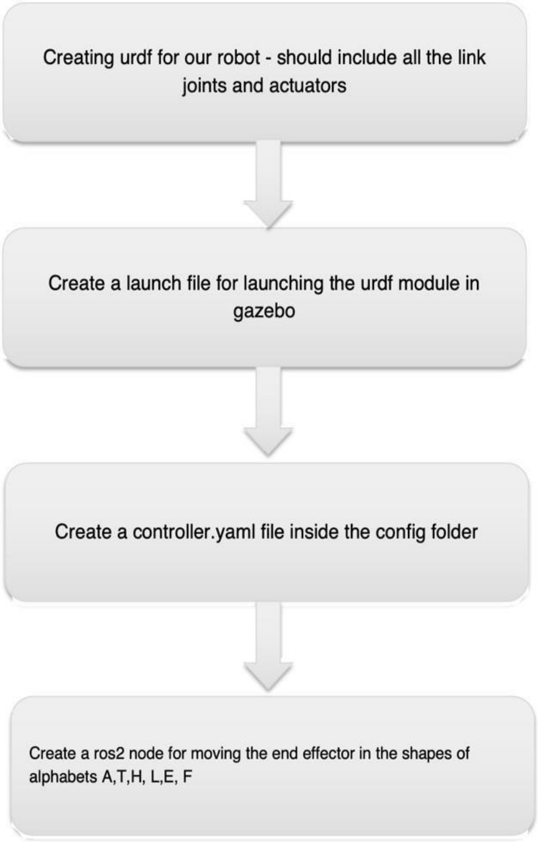
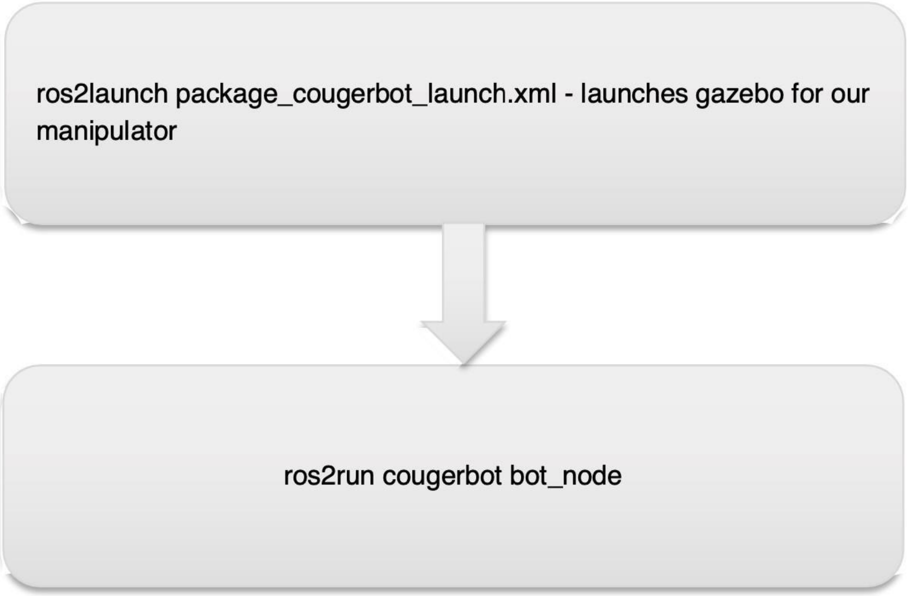

Creating urdf for our robot - should include all the link joints and actuators
Create a launch file for launching the urdf module in gazebo
Create a controller.yaml file inside the config folder
Create a ros2 node for moving the end effector in the shapes of alphabets A,T,H, L,E, F

Figure.1

Implementation flowchart for the project.

- First we create a package in the src of our catkin workspace. Next change directory into the created package and create a new folder ‘urdf’ and add the robot design urdf file into it.
- (‘couger.urdf’ in our case). Next is to create a launch file to open the gazebo environment of the urdf file or the manipulator.
- Create a ‘controller.yaml’ file consisting of all the joints of the manipulator.
- Make necessary changes to set path up for controller.yaml
- Next is to create a python file for moving the manipulator or arm as desired. (Alphabets in our case)



ros2launch package_cougerbot_launch.xml - launches gazebo for our manipulator

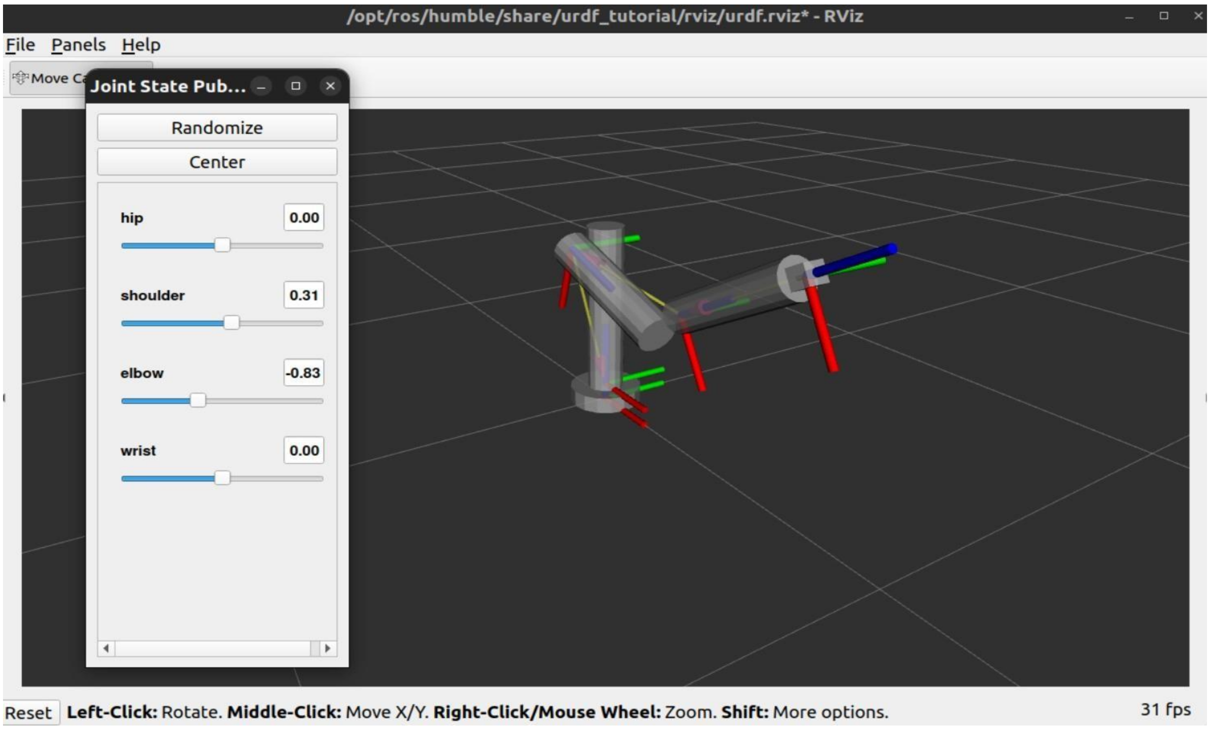
ros2run cougerbot bot_node

Figure.2

List of commands in order to attain the end effector positions.

The above flowchart gives the commands required to run the Alphabet Writing Maniputor.

OUTPUT



/opt/ros/humble/share/urdf_tutorial/rviz/urdf.rviz* - RViz
File
Panels
Help
Move Ca
Joint State Pub...
Randomize
Center
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.
31 fps

hip 0.00

shoulder 0.31

elbow -0.83

wrist 0.00

Figure.3

Snapshot of rviz window for main launch file.

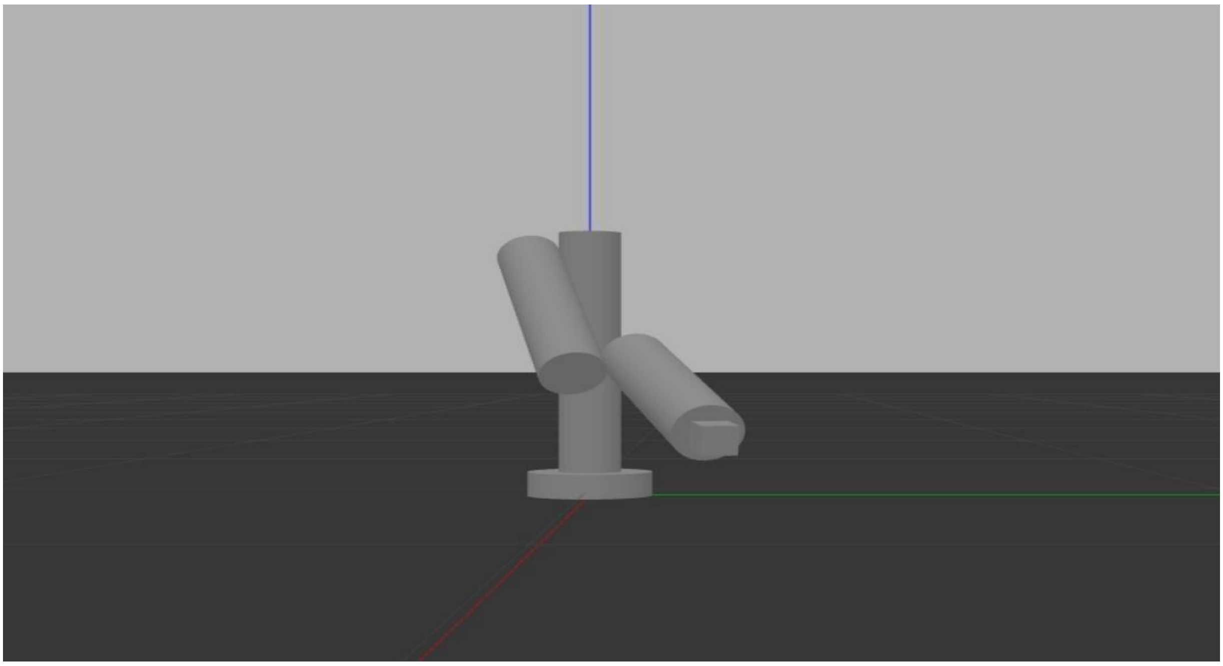
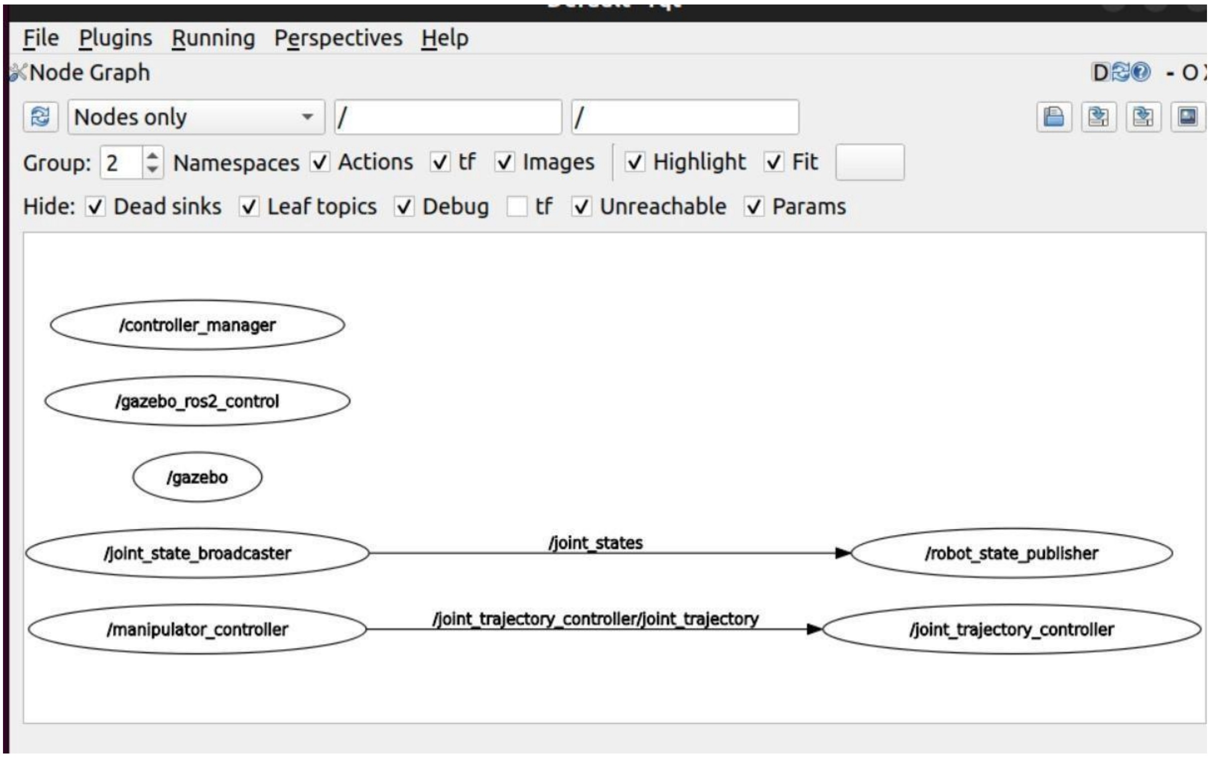


Figure.4

Snapshot of gazebo window for motion planning.

ROS NODE GRAPH



Nodes only / /

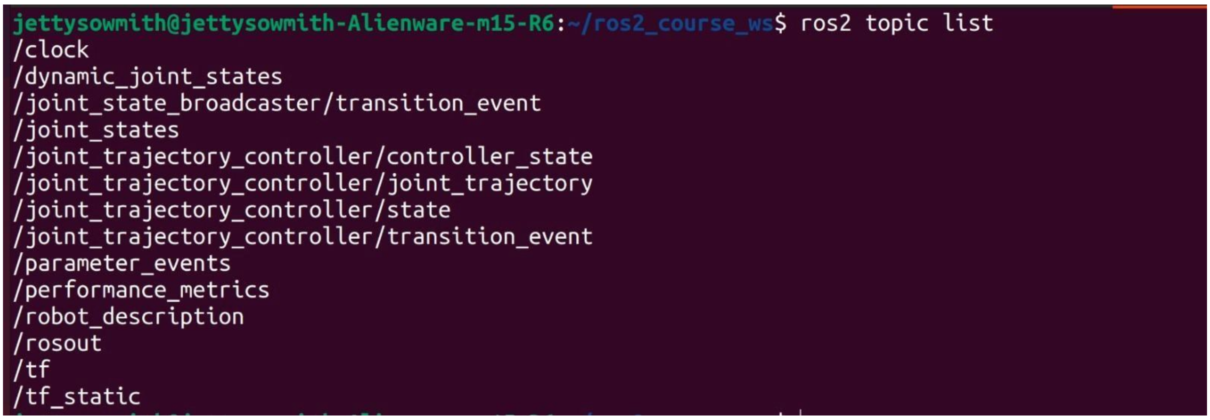
Group: 2 Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params

Figure.5

Snapshot of Node Graph

ROS TOPIC LIST



```
jettysowmith@jettysowmith-Alienware-m15-R6:~/ros2_course_ws$ ros2 topic list
/clock
/dynamic_joint_states
/joint_state_broadcaster/transition_event
/joint_states
/joint_trajectory_controller/controller_state
/joint_trajectory_controller/joint_trajectory
/joint_trajectory_controller/state
/joint_trajectory_controller/transition_event
/parameter_events
/performance_metrics
/robot_description
/rosout
/tf
/tf_static
```

Figure.6

Snapshot of terminal window showing all the active ros topics.

CONCLUSION

This project showcases the integration of simulation, motion planning, and real-time control using ROS2 tools. Users can explore further by experimenting with different trajectories and configurations, demonstrating the versatility and flexibility of ROS2 for robotic applications.