Institute of Field Robotics, King Mongkut's University of Technology Thonburi
126 Pracha Uthit Rd, Bang Mot, Thung Khru, Bangkok, Thailand 10140
Senior Thesis FRAB (FIBO Robotics and Automation: Bachelor)
Copyright © 2021 by FIBO

A study of Torque-Control system on UR5 for the drawing task

Jirattanun Leeudomwong, 66340500009
Bharut Aursudkit, 66340500040
Nuttapruet Puttiwarodom, 66340500018

Abstract
Industrial robot manipulators have countless applications nowadays. One of the basic tasks to test the accuracy of a robot manipulator is to perform a drawing task, which can be done with only position control alone. This project focuses on using a UR5 industrial robot manipulator to perform the drawing task. The challenge this project represents is controlling the force between the pen (end-effector) and the canvas using torque control while following a trajectory generated by the image processing method. This project is running on Gazebo Ignition physics simulation and using ROS2 as a middleware.

Keywords: Torque control, Dynamics model, Trajectory generation, Image processing, Edge detection, ROS2, Physic simulation

1 Overview

This topic will explain about what do the contribution of this work and what each topic will cover

2 Software architecture and Simulation

This work performed in Gazebo ignition physic simulation. The simulation simulates the UR5, pen, gravity, collision etc. ROS2 middleware using to interface the simulation and using as the backbone of the robot system. The software architecture can be display as Node and Topic (as shown on Fig.1)

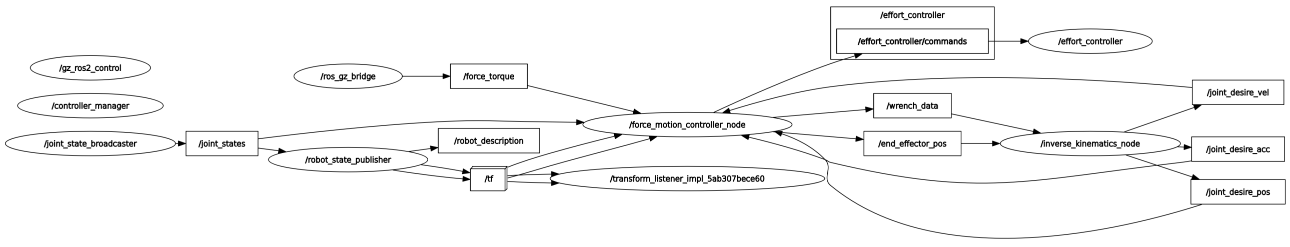


Figure 1: Nodes and topics

The `force_motion_controller_node` and `inverse_kinematic_node` were developed by the project team.

1. `force_motion_controller_node`
This node acquires and filters the wrench from the F/T sensor, reads end-effector positions, computes UR5 dynamics, and generates torques for motion control (can be found in `ur_controller` package).

2. `inverse_kinematic_node`
This node performs inverse kinematics calculations, Z-axis elevation force control and read trajectory from .csv file.

Additionally, There are several Node required to interface with the Gazebo

1. `robot_state_broadcaster`
This node publishes the current state (position, velocity) of individual joints to the `joint_states` topic

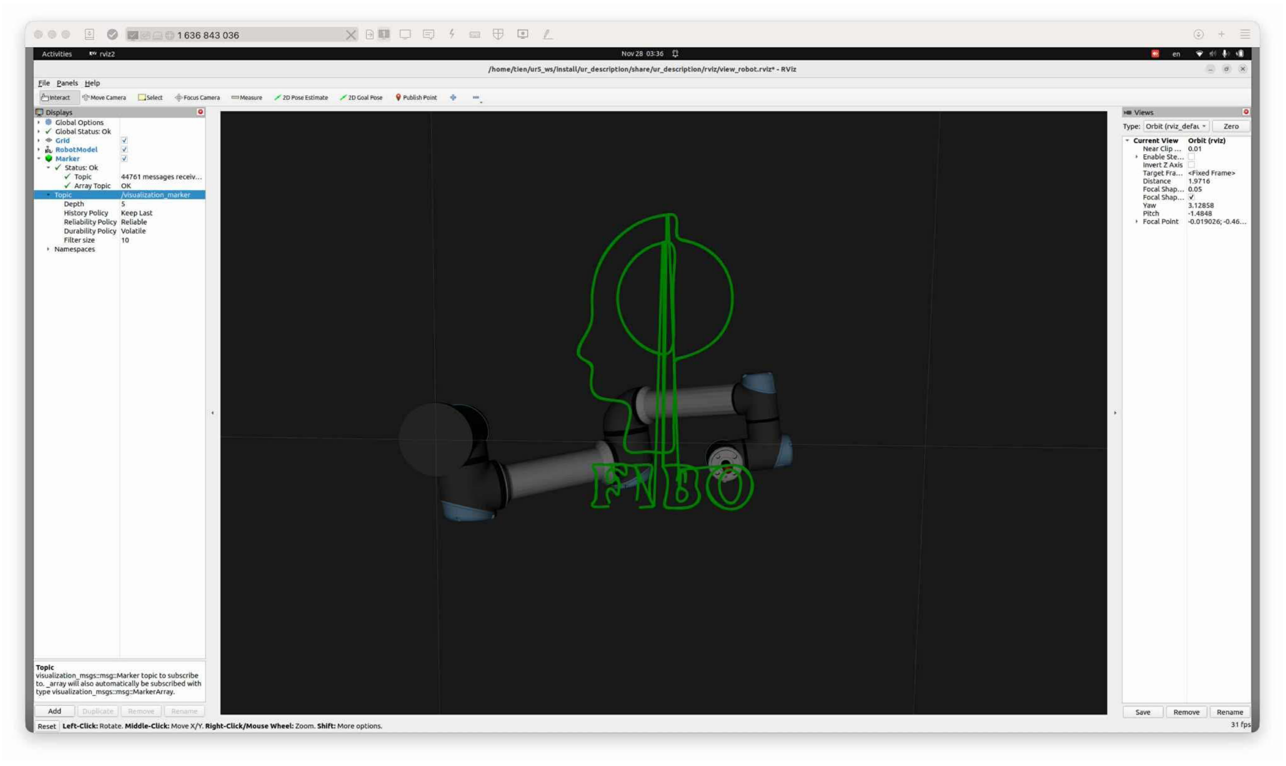
2. `robot_state_publisher`
computes and broadcasts the 3D poses of a robotś links based on its URDF model and incoming `joint_states` messages

3. `ros_gz_bridge`
The F/T sensor is instant function from Gazebo ignition. To acquire wrench from F/T sensor, only need to bridge the `gz_topic` to the ROS2 topic using `ros_gz_bridge`

4. `effort_controller`
This node use to control the UR5 joint torque. This node is part of `ROS2_control` framework

Lastly, There is addition tools helps this project display drawing line and debugging

1. rviz2
This program show the 3D model of the UR5, force and drawing line as show in Fig.2



Global Options
Global Status: Ok
Grid ✓
RobotModel ✓
Marker ✓
Status: Ok
Topic 44761 messages receiv...
Array Topic OK
Topic /visualization_marker
Depth 5
History Policy Keep Last
Reliability Policy Reliable
Durability Policy Volatile
Filter size 10
Namespaces

Topic
visualization_msgs::msg::Marker topic to subscribe to. _array will also automatically be subscribed with type visualization_msgs::msg::MarkerArray.

Type: Orbit (rviz_defau
Current View Orbit (rviz)
Near Clip ... 0.01
Enable Ste... ☐
Invert Z Axis ☐
Target Fra... <Fixed Frame>
Distance 1.9716
Focal Shap... 0.05
Focal Shap... ✓
Yaw 3.12858
Pitch -1.4848
Focal Point -0.019026; -0.46...

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

Figure 2: rviz2 interface

3 Trajectory generation

This topic will explain about the methodology of trajectory generation and result

4 Inverse Kinematic

5 Dynamics and Control system

There are various type of control method available for UR5 with torque control. This work presents a motion control strategy based on the dynamics of the UR5. To generate the required torque for motion control, The inverse dynamics equation in joint space (equation 1) is used to compute required torque respect to the Joint state[1]. To control position, velocity and acceleration of the UR5 using PD motion controller (equation 2)

$$\tau = M(q)\ddot{q} + C(q,\dot{q}) + g(q) \tag{1}$$

$$\ddot{q} = \ddot{q}_d + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) \tag{2}$$

where:

- τ is the joint torque Vector
- $M(q)\ddot{q}$ is the Inertia Matrix
- $C(q,\dot{q})$ is the Coriolis and Centrifugal Matrix
- $g(q)$ is the Gravity Vector
- $\ddot{q}$ is the computed acceleration (command)
- K_p and K_d are the proportional and derivative gain matrices
- q_d and q are the desired and actual joint positions
- $\dot{q}_d$ and $\dot{q}$ are the desired and actual joint velocities
- $\ddot{q}_d$ is the desired joint acceleration

However, In the dynamics model in joint space (equation 1) and PD controller (equation 2) alone are not able to control the force at the end-effector. The method for control motion and force at end-effector simultaneously is to apply the Hybrid force-motion control. Even so the hybrid force-motion control is using the dynamics model in task space. Make the dynamics calculation more complicated to compute. Even if using dynamics model for joint space for motion controller and force controller together. Controller will be fighting over position. This project presents the Z-axis elevation PI control (equation 3) method to control the force occurs at the end-effector. F/T sensor is noisy sensor even apply the low-pass filter. This cause the force elevation controller is only PI controller because the D term is noist sensor. This method will not affect the control system to compensate the position error with stronger force respect to the time because of the motion controller is only PD controller.

$$Z = -1 \cdot (K_p(W_d - W) + K_i \int (W_d - W)\frac{d}{dt}) \tag{3}$$

where:

- Z is the elevation in Z axis (task space)
- K_p and K_i are the proportional and integral gain matrices
- W_d and W are the desired and actual force at end-effector Z axis

After apply PD motion controller and PI elevation controller, UR5 are able to control motion and force simultaneously without fighting between 2 controllers in joint space. The control diagram is show in Fig.3

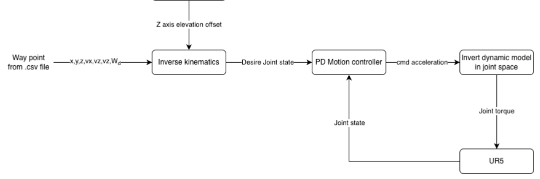


Figure 3: Control system diagram

[1]Joint state: joint position, joint velocity, joint acceleration

Lastly the inverse dynamics model calculation done by pinocchio framework on python. And Controlled by the custom code by this project team. (code implimentation here Github:)

6 Test and Evaluation

The requirement of this project is to perform drawing task within position error 10% range and able to do force control. RMSE statistic formular can be used to find the average position error occurs on the system.

7 Conclusion