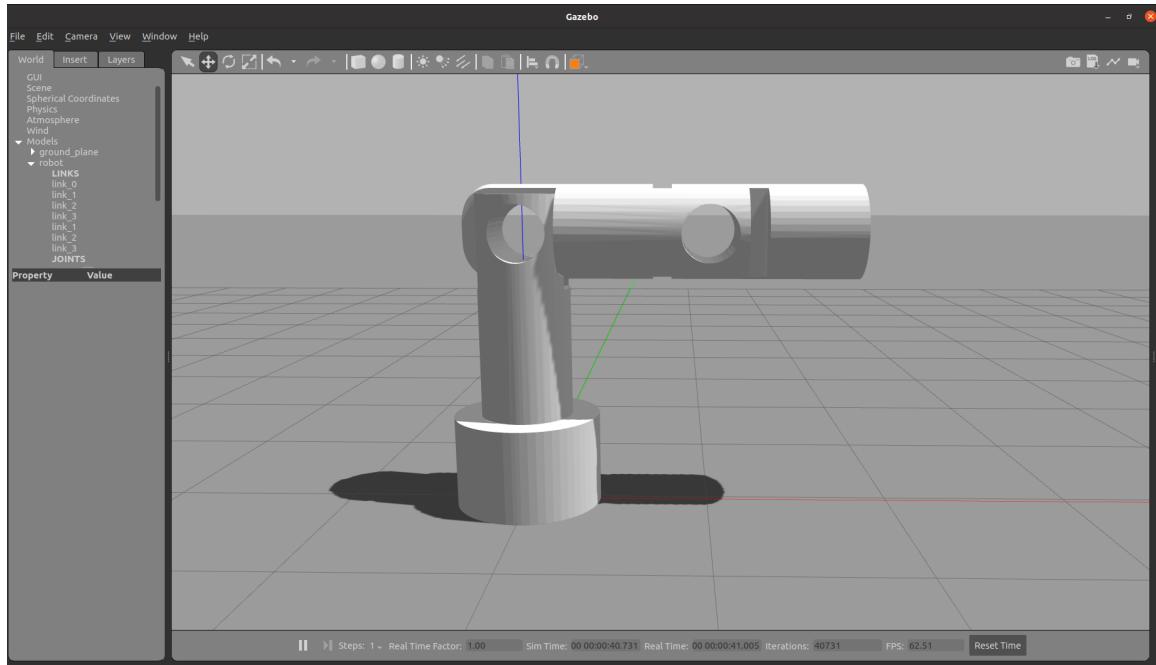
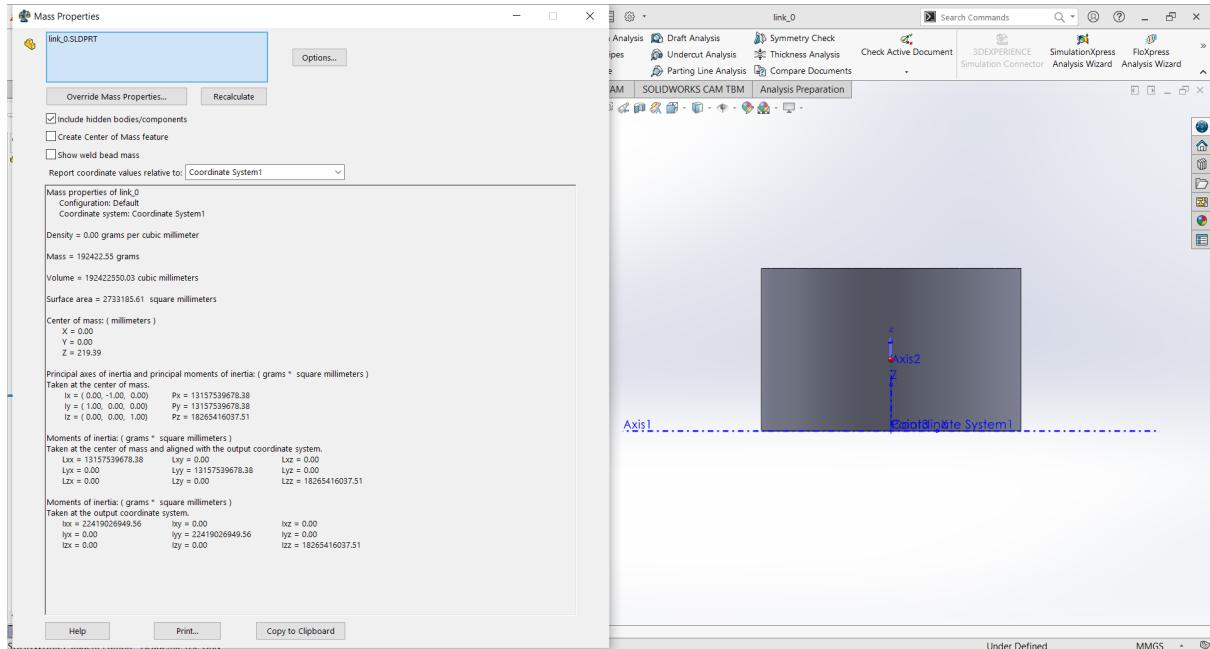


LAB 4 REPORT

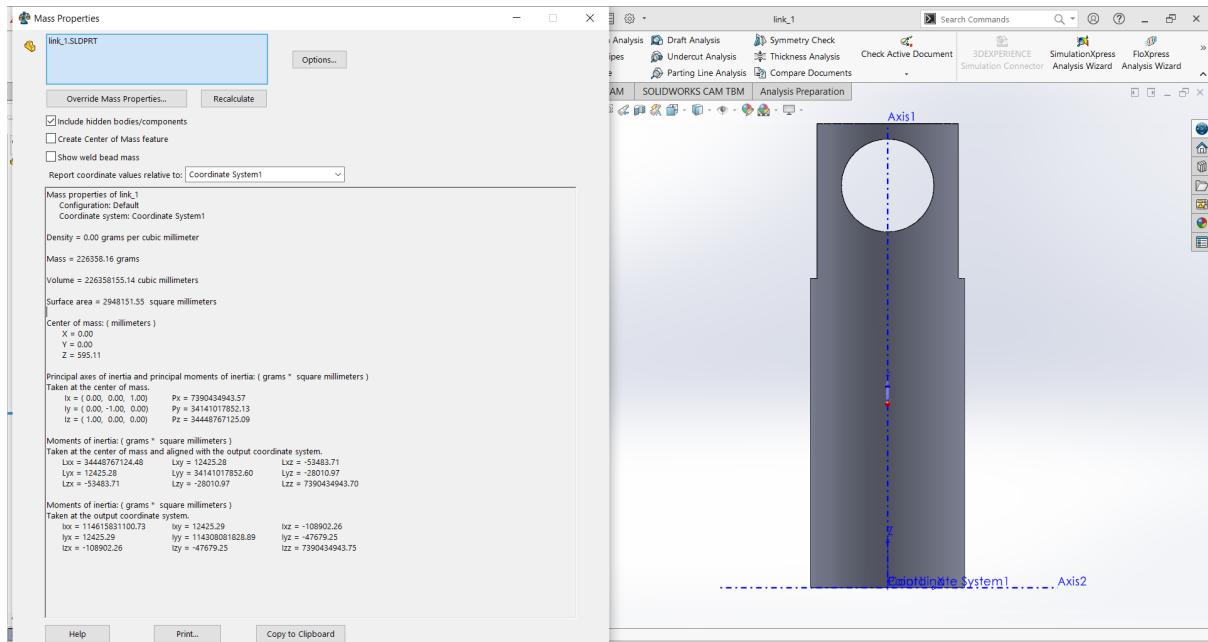
LAB 4 Model



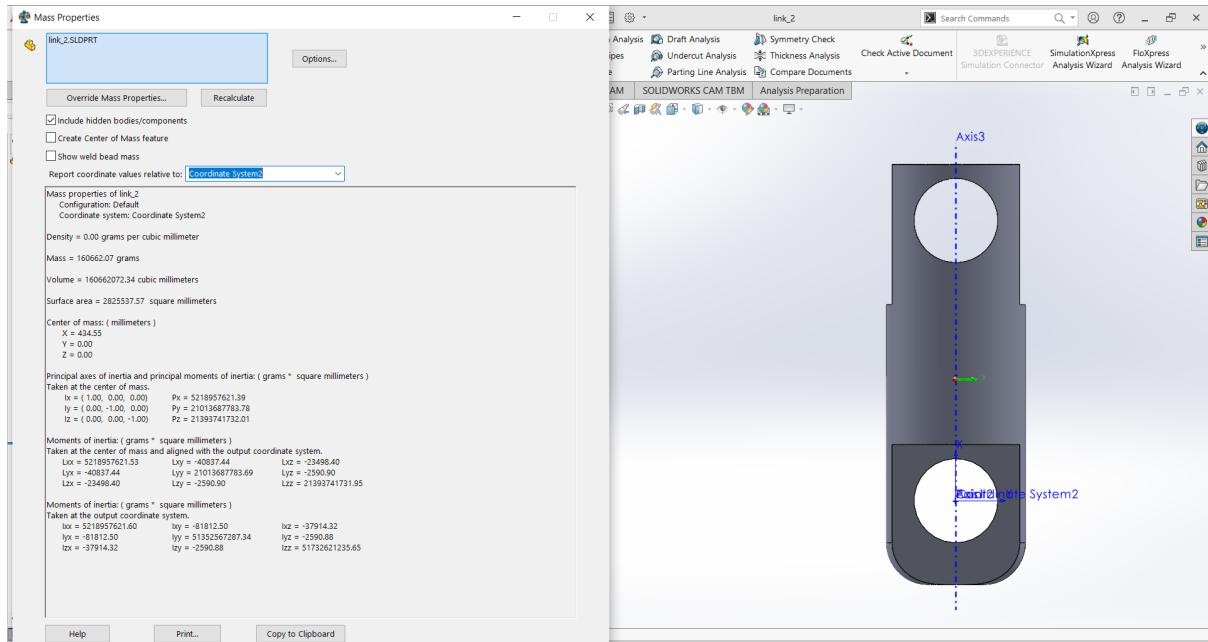
Link 0



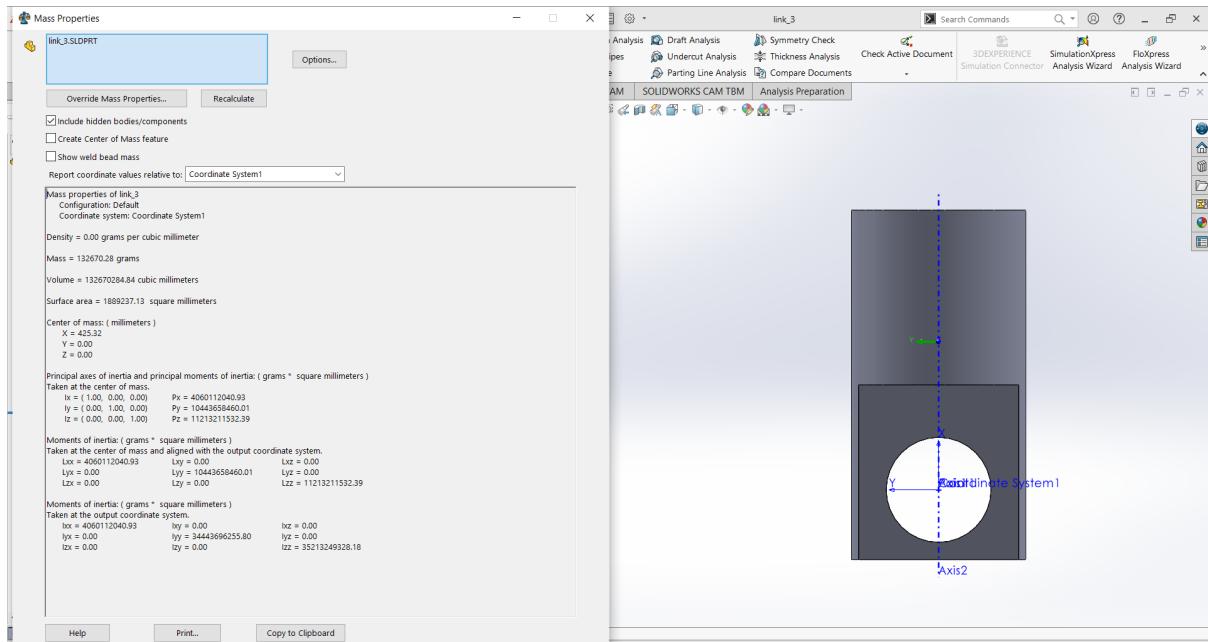
Link 1



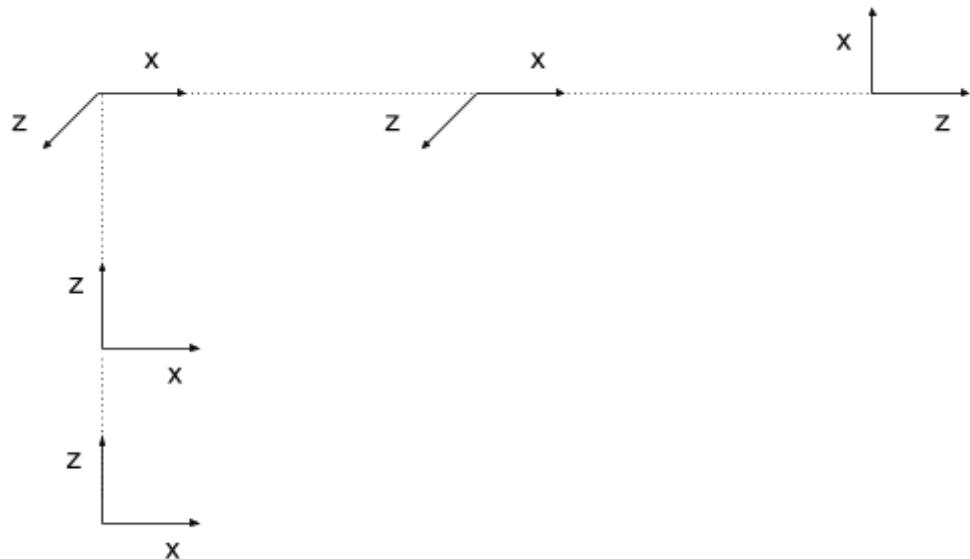
Link 2



Link 3



Calculate FK and IK



Forward Kinematics for Position

DH Parameter

i-1	i	a	alpha	d	theta
0	1	0	0	200	0
1	2	0	90	1300	0
2	3	1000	0	0	0
3	e	800	90	0	90

$$H_e^3 = [0 \ 0 \ 1 \ 800], [1 \ 0 \ 0 \ 0], [0 \ 1 \ 0 \ 0], [0 \ 0 \ 0 \ 1]$$

Inverse Kinematics for Position Control

$$l_1 = 1000 \text{ mm}$$

$$l_2 = 1000 \text{ mm}$$

$$l_3 = 800 \text{ mm}$$

$$z = 1500 \text{ mm}$$

$x, y = \text{Via Point}$

$$r = \text{gamma1} * \sqrt{x^2 + y^2}$$

$$\mathbf{c2} = (r^2 + (z - l_1)^2 - l_2^2 - l_3^2) / 2l_2l_3$$

$$\mathbf{s2} = \text{gamma2} * \sqrt{1 - c_2^2}$$

$$\mathbf{q2} = \text{atan2}(z - l_1, r_2) - \text{atan2}(l_3 s_2, l_2 + (l_3 c_2))$$

$$\mathbf{q3} = \text{atan2}(s_2, c_2)$$

$$\mathbf{q1} = (y/\text{gamma1}, x/\text{gamma1})$$

Inverse Kinematics for Velocity Control

$\mathbf{J} =$

$$[-l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3),$$

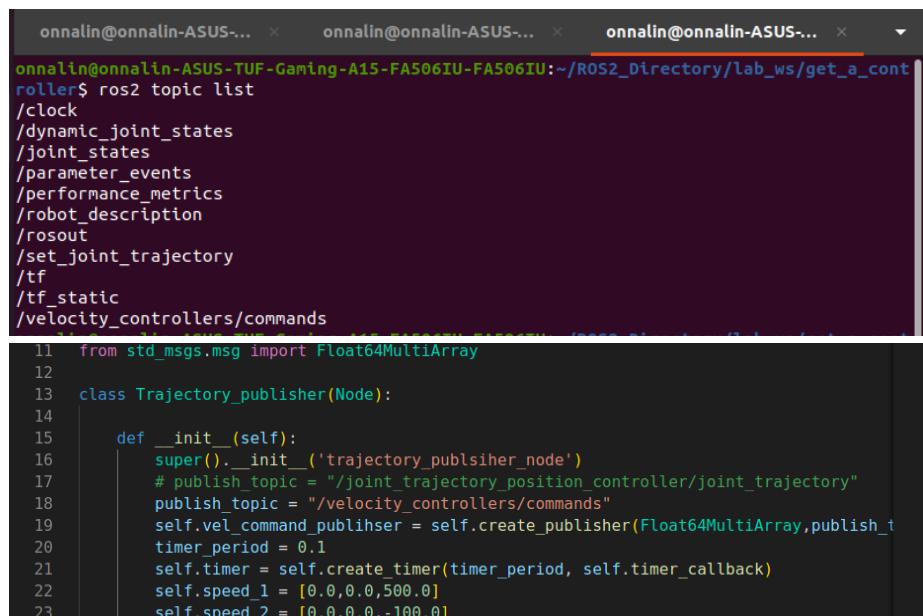
$$, -l_2 \sin(\theta_1 + \theta_2) - l_3 \theta_1 + \theta_2 + \theta_3, -l_3 \sin(\theta_1 + \theta_2 + \theta_3)]$$

$$,[l_2 \cos(\theta_1 + \theta_2) + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2 + \theta_3),$$

$$l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3), l_3 \cos(\theta_1 + \theta_2 + \theta_3)]$$

Step 1: Setting up velocity controller

run trajectory and Follow Topic
`/velocity_controllers/commands`



```
onnalin@onnalin-ASUS-... ~ onnalin@onnalin-ASUS-... ~ onnalin@onnalin-ASUS-...
onnalin@onnalin-ASUS-TUF-Gaming-A15-FA506IU-FA506IU:~/ROS2_Directory/lab_ws/get_a_controller$ ros2 topic list
/clock
/dynamic_joint_states
/joint_states
/parameter_events
/performance_metrics
/robot_description
/rosout
/set_joint_trajectory
/tf
/tf_static
/velocity_controllers/commands

11  from std_msgs.msg import Float64MultiArray
12
13  class Trajectory_publisher(Node):
14
15      def __init__(self):
16          super().__init__('trajectory_publisher_node')
17          # publish_topic = "/joint_trajectory_position_controller/joint_trajectory"
18          publish_topic = "/velocity_controllers/commands"
19          self.vel_command_publisher = self.create_publisher(Float64MultiArray,publish_
20          timer_period = 0.1
21          self.timer = self.create_timer(timer_period, self.timer_callback)
22          self.speed_1 = [0.0,0.0,500.0]
23          self.speed_2 = [0.0,0.0,-100.0]
```

Step 2: Developing a tracker

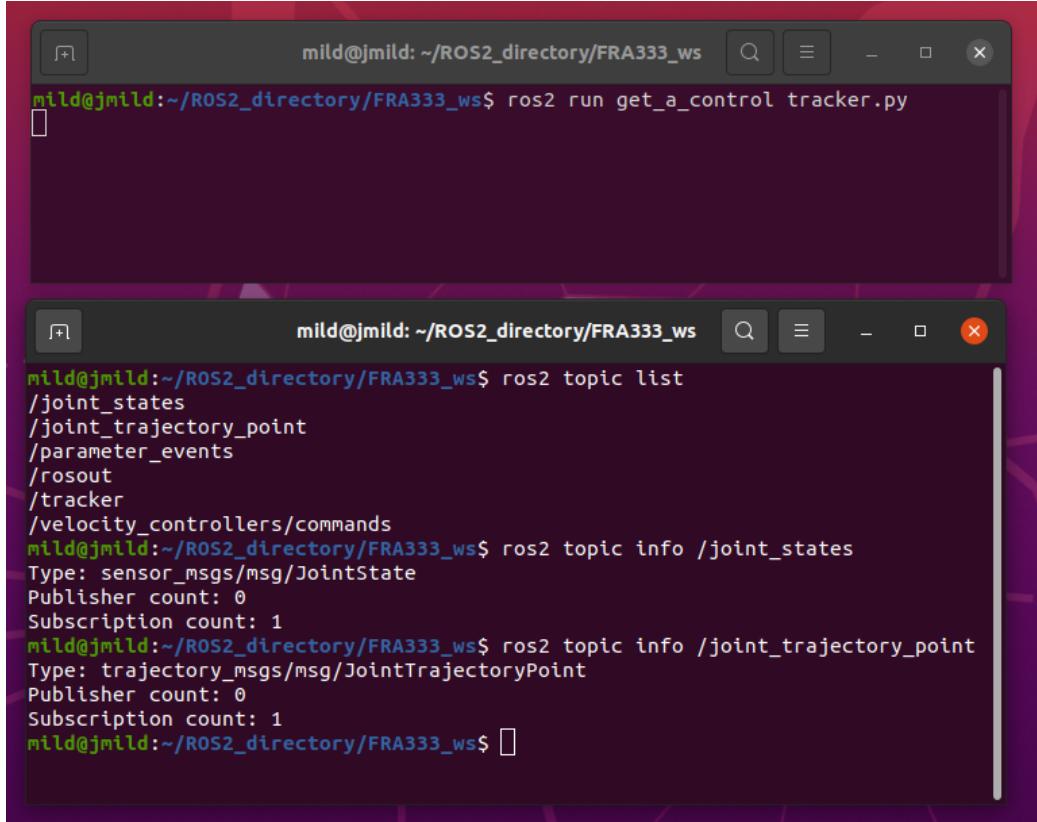
create package get_a_control and create tracker.py in package

```
mild@mild:~/ROS2_directory/FRA333_ws$ colcon build --packages-select get_a_control
Starting >>> get_a_control
Finished <<< get_a_control [0.40s]

Summary: 1 package finished [0.70s]
mild@mild:~/ROS2_directory/FRA333_ws$ 
```

At first, We run ‘ros2 run get_a_trajectory_.py’ for sent Velocity to tracker.py, But tracker.py don’t active until enable is True by ‘ros2 service call /enable get_a_interfaces/srv/EnableTracker “enable: True” ’ and disables by ‘ros2 service call /enable get_a_interfaces/srv/EnableTracker “enable: False” ’

When calling a service “enable” Tracker.py check topic, If enable is true -> Tracker.py publish : ‘sent’ at a fixed rate, If enable is false -> Tracker.py publish : ‘stop’ once time.



The image shows two terminal windows side-by-side. The left terminal window has a title bar "mild@mild: ~/ROS2_directory/FRA333_ws\$". It contains the command "ros2 run get_a_control tracker.py" followed by a blank line. The right terminal window also has a title bar "mild@mild: ~/ROS2_directory/FRA333_ws\$". It contains the command "ros2 topic list", followed by a list of topics: /joint_states, /joint_trajectory_point, /parameter_events, /rosout, /tracker, /velocity_controllers/commands. Below this, it runs "ros2 topic info /joint_states", showing Type: sensor_msgs/msg/JointState, Publisher count: 0, Subscription count: 1. Then it runs "ros2 topic info /joint_trajectory_point", showing Type: trajectory_msgs/msg/JointTrajectoryPoint, Publisher count: 0, Subscription count: 1. Finally, it ends with "mild@mild:~/ROS2_directory/FRA333_ws\$ []".

```
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 run get_a_control tracker.py
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 topic list
/joint_states
/joint_trajectory_point
/parameter_events
/rosout
/tracker
/velocity_controllers/commands
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 topic info /joint_states
Type: sensor_msgs/msg/JointState
Publisher count: 0
Subscription count: 1
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 topic info /joint_trajectory_point
Type: trajectory_msgs/msg/JointTrajectoryPoint
Publisher count: 0
Subscription count: 1
mild@mild:~/ROS2_directory/FRA333_ws$ []
```

Next, run tracker.py for check count of Subscription.
Tracker should subscription ‘/joint_states’ for required a position feedback and ‘/joint_trajectory_point’ for subscribe ref_position and ref_velocity

```
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 run get_a_control tracker.py
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 topic list
/joint_states
/joint_trajectory_point
/parameter_events
/rosout
/tracker
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 topic echo /tracker
```

```
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 service list
/enableTracker
/tracker/describe_parameters
/tracker/get_parameter_types
/tracker/get_parameters
/tracker/list_parameters
/tracker/set_parameters
/tracker/set_parameters_atomically
mild@mild: ~/ROS2_directory/FRA333_ws$ 
```



```
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 run get_a_control tracker.py
[INFO] [1671547492.178492044] [tracker]: sent
[INFO] [1671547492.664534538] [tracker]: sent
[INFO] [1671547493.164551018] [tracker]: sent
[INFO] [1671547493.664096007] [tracker]: sent
[INFO] [1671547494.164741568] [tracker]: sent
```



```
/tracker
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 topic echo /tracker
layout:
  dim: []
  data_offset: 0
data:
- 0.1
- 0.1
- 0.1
...
layout:
  dim: []
```



```
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 service list
/enableTracker
/tracker/describe_parameters
/tracker/get_parameter_types
/tracker/get_parameters
/tracker/list_parameters
mild@mild: ~/ROS2_directory/FRA333_ws$ ros2 service call /enableTracker get_a_interfaces/srv/EnableTracker "{enable: True}"
requester: making request: get_a_interfaces.srv.EnableTracker_Request(enable=True)

response:
get_a_interfaces.srv.EnableTracker_Response()
```

The screenshot shows three terminal windows on a Linux desktop. The top window displays ROS2 INFO messages from a tracker node. The middle window shows the structure of a message with fields like data_offset, data, and layout. The bottom window shows the execution of service calls to enable and disable a tracker.

```
[INFO] [1671547570.084403324] [tracker]: sent
[INFO] [1671547570.584476794] [tracker]: sent
[INFO] [1671547571.084448282] [tracker]: sent
[INFO] [1671547571.583001311] [tracker]: sent
[INFO] [1671547572.084453467] [tracker]: stop
```

```
data_offset: 0
data:
- 0.1
- 0.1
- 0.1
...
layout:
  dim: []
  data_offset: 0
data: []
```

```
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 service call /enableTracker get_a_interfaces.srv.EnableTracker_Request(enable=True)
requester: making request: get_a_interfaces.srv.EnableTracker_Request()
response: get_a_interfaces.srv.EnableTracker_Response()

mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 service call /enableTracker get_a_interfaces.srv.EnableTracker_Request(enable=False)
requester: making request: get_a_interfaces.srv.EnableTracker_Request()
response: get_a_interfaces.srv.EnableTracker_Response()
```

check subscription

The screenshot shows two terminal windows. The top window runs a Python script named 'get_a_control_tracker.py'. The bottom window lists topics and provides detailed information about the '/joint_states' topic.

```
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 run get_a_control tracker.py
```

```
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic list
/joint_states
/joint_states_estimator
/parameter_events
/rosout
/tracker
```

```
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic info /joint_states
Type: trajectory_msgs/msg/JointTrajectoryPoint
Publisher count: 0
Subscription count: 1
```

```
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic info /joint_states_estimator
Type: sensor_msgs/msg/JointState
Publisher count: 0
Subscription count: 1
```

Kinematic server

```
mild@jmild:~/ROS2_directory/FRA333_ws$ source install/setup.bash
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 run get_a_control kinematics_server.py FK
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic list
/end_effector_state
/joint_states
/parameter_events
/rosout
/topic
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic info /joint_states
Type: sensor_msgs/msg/JointState
Publisher count: 0
Subscription count: 1
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic info /end_effector_state
Type: sensor_msgs/msg/JointState
Publisher count: 1
Subscription count: 0
mild@jmild:~/ROS2_directory/FRA333_ws$ 
```

```
_rclpy.rclpy_wait(wait_set, timeout_nsec)
KeyboardInterrupt
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 run get_a_control kinematics_server.py IK
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic list
/end_effector_state
/joint_states
/parameter_events
/rosout
/topic
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic info /end_effector_state
Type: sensor_msgs/msg/JointState
Publisher count: 0
Subscription count: 1
mild@jmild:~/ROS2_directory/FRA333_ws$ ros2 topic info /joint_states
Type: sensor_msgs/msg/JointState
Publisher count: 1
Subscription count: 0
mild@jmild:~/ROS2_directory/FRA333_ws$ 
```