Parth Patel, Noopur Koshta, Cameron Schloer

Group Assignment - Part 1

Create Robot in Gazebo World

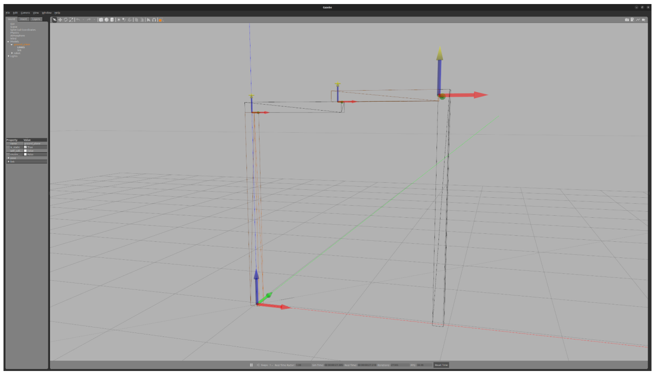


Figure 1: Custom made SCARA robot in Gazebo, frames visible

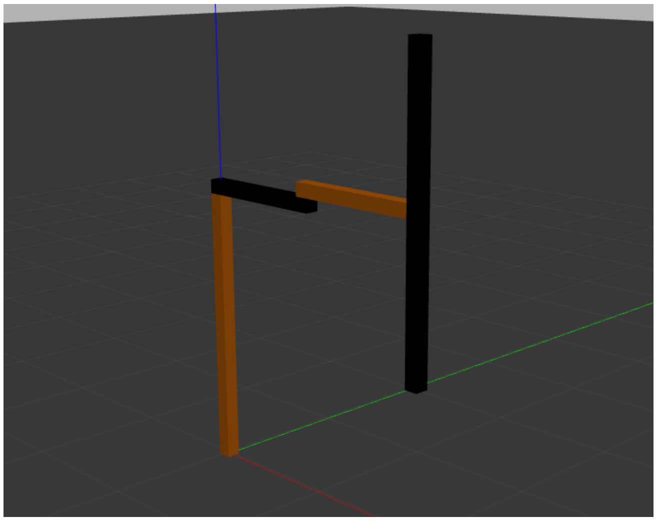
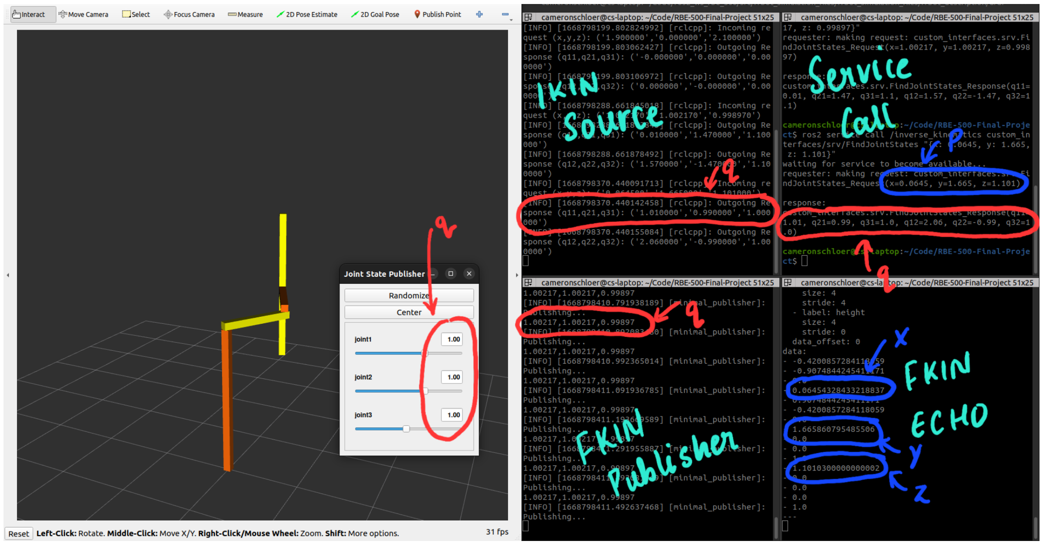


Figure 2: Home configuration of SCARA robot in Gazebo

RBE 500 – Foundation of Robotics - Fall 2022

Testing the Forward and Inverse Kinematics



joint1 1.00
joint2 1.00
joint3 1.00

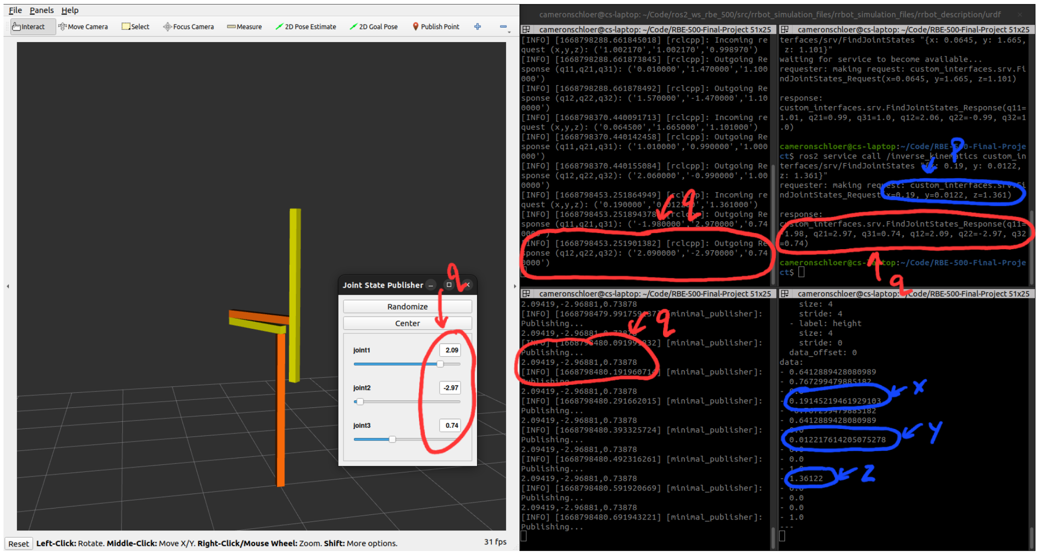
```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
[INFO] [1668798199.802824992] [rclcpp]: Incoming re
quest (x,y,z): ('1.900000','0.000000','2.100000')
[INFO] [1668798199.803062427] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('-0.000000','0.000000','0.00
0000')
[INFO] [1668798199.803106972] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('0.000000','-0.000000','0.00
0000')
[INFO] [1668798288.661845018] [rclcpp]: Incoming re
quest (x,y,z): ('1.002170','1.002170','0.998970')
[INFO] [1668798288.661873845] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('0.010000','1.470000','1.100
000')
[INFO] [1668798288.661878492] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('1.570000','-1.470000','1.10
0000')
[INFO] [1668798370.440091713] [rclcpp]: Incoming re
quest (x,y,z): ('0.064500','1.665000','1.101000')
[INFO] [1668798370.440142458] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('1.010000','0.990000','1.000
000')
[INFO] [1668798370.440155084] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('2.060000','-0.990000','1.00
0000')
```

```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
17, z: 0.99897}"
requester: making request: custom_interfaces.srv.Fi
ndJointStates_Request(x=1.00217, y=1.00217, z=0.998
97)

response:
custom_interfaces.srv.FindJointStates_Response(q11=
0.01, q21=1.47, q31=1.1, q12=1.57, q22=-1.47, q32=1
.1)

cameronschloer@cs-laptop:~/Code/RBE-500-Final-Proje
ct$ ros2 service call /inverse_kinematics custom_in
terfaces/srv/FindJointStates "{x: 0.0645, y: 1.665,
 z: 1.101}"
waiting for service to become available...
requester: making request: custom_interfaces.srv.Fi
ndJointStates_Request(x=0.0645, y=1.665, z=1.101)

response:
custom_interfaces.srv.FindJointStates_Response(q11=
1.01, q21=0.99, q31=1.0, q12=2.06, q22=-0.99, q32=1
.0)

cameronschloer@cs-laptop:~/Code/RBE-500-Final-Proje
ct$
```

```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
1.00217,1.00217,0.99897
[INFO] [1668798410.791938189] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [16687984[illegible]0] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [1668798410.992365014] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [1668798411.091936785] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [1668798411.193609589] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [1668798411.291955887] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [16687984[illegible]] [minimal_publisher]:
Publishing...
1.00217,1.00217,0.99897
[INFO] [1668798411.492637468] [minimal_publisher]:
Publishing...
```

```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
    size: 4
    stride: 4
  - label: height
    size: 4
    stride: 0
  data_offset: 0
data:
- -0.4200857284118059
- -0.9074844245411171
- 0.0
- 0.06454328433218837
- 0.9074844245411171
- -0.4200857284118059
- 0.0
- 1.665860795485506
- 0.0
- 0.0
- 1.0
- 1.1010300000000002
- 0.0
- 0.0
- 0.0
- 1.0
---
```

Figure 3: First position that tests Forward and Inverse Kinematics



joint1 2.09
joint2 -2.97
joint3 0.74

```
cameronschloer@cs-laptop: ~/Code/ros2_ws_rbe_500/src/rrbot_simulation_files/rrbot_simulation_files/rrbot_description/urdf
```

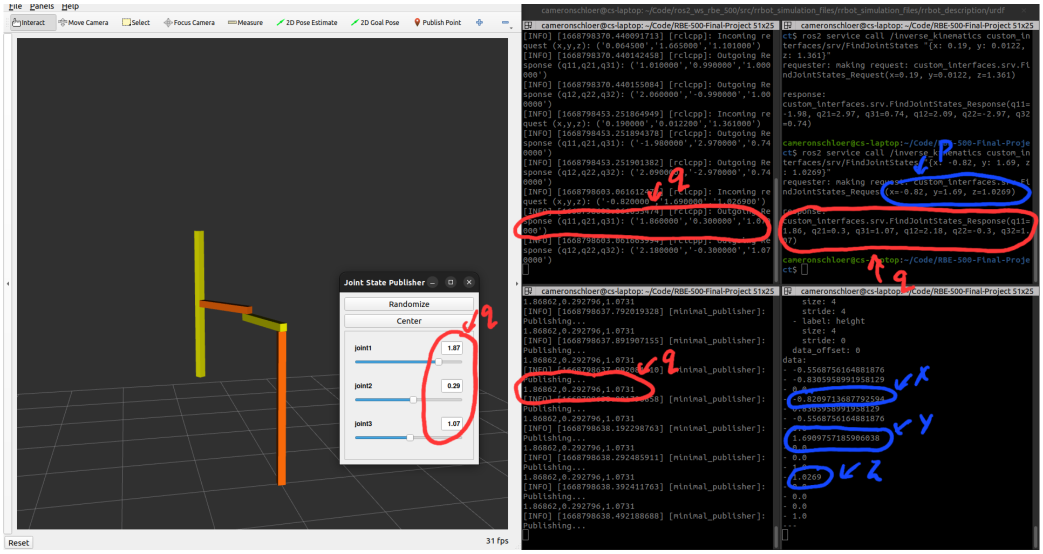
```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
[INFO] [1668798288.661845018] [rclcpp]: Incoming re
quest (x,y,z): ('1.002170','1.002170','0.998970')
[INFO] [1668798288.661873845] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('0.010000','1.470000','1.100
000')
[INFO] [1668798288.661878492] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('1.570000','-1.470000','1.10
0000')
[INFO] [1668798370.440091713] [rclcpp]: Incoming re
quest (x,y,z): ('0.064500','1.665000','1.101000')
[INFO] [1668798370.440142458] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('1.010000','0.990000','1.000
000')
[INFO] [1668798370.440155084] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('2.060000','-0.990000','1.00
0000')
[INFO] [1668798453.251864949] [rclcpp]: Incoming re
quest (x,y,z): ('0.190000','0.012200','1.361000')
[INFO] [1668798453.251894378] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('-1.980000','2.970000','0.74
0000')
[INFO] [1668798453.251901382] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('2.090000','-2.970000','0.74
0000')
```

```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
terfaces/srv/FindJointStates "{x: 0.0645, y: 1.665,
 z: 1.101}"
waiting for service to become available...
requester: making request: custom_interfaces.srv.Fi
ndJointStates_Request(x=0.0645, y=1.665, z=1.101)

response:
custom_interfaces.srv.FindJointStates_Response(q11=
1.01, q21=0.99, q31=1.0, q12=2.06, q22=-0.99, q32=1
.0)

cameronschloer@cs-laptop:~/Code/RBE-500-Final-Proje
ct$ ros2 service call /inverse_kinematics custom_in
terfaces/srv/FindJointStates "{x: 0.19, y: 0.0122,
z: 1.361}"
requester: making request: custom_interfaces.srv.Fi
ndJointStates_Request(x=0.19, y=0.0122, z=1.361)

response:
custom_interfaces.srv.FindJointStates_Response(q11=
-1.98, q21=2.97, q31=0.74, q12=2.09, q22=-2.97, q32
=0.74)

cameronschloer@cs-laptop:~/Code/RBE-500-Final-Proje
ct$
```

```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
2.09419,-2.96881,0.73878
[INFO] [1668798479.991759[illegible]37] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.0919[illegible]832] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.191960714] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.291662015] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.393325724] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.492316261] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.591920669] [minimal_publisher]:
Publishing...
2.09419,-2.96881,0.73878
[INFO] [1668798480.691943221] [minimal_publisher]:
Publishing...
```

```
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25
    size: 4
    stride: 4
  - label: height
    size: 4
    stride: 0
  data_offset: 0
data:
- 0.6412889428080989
- 0.767299479885182
- 0.0
- 0.1914521946192910З
- -0.767299479885182
- 0.6412889428080989
- 0.0
- 0.012217614205075278
- 0.0
- 0.0
- 1.0
- 1.36122
- 0.0
- 0.0
- 0.0
- 1.0
---
```

Figure 4: Second position that tests Forward and Inverse Kinematics


File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Goal Pose
Publish Point
Joint State Publisher
Randomize
Center
Reset
31 fps
q
q
q
q
P
x
y
z
cameronschloer@cs-laptop: ~/Code/ros2_ws_rbe_500/src/rrbot_simulation_files/rrbot_simulation_files/rrbot_description/urdf
cameronschloer@cs-laptop: ~/Code/RBE-500-Final-Project 51x25

joint1 1.87

joint2 0.29

joint3 1.07

```
[INFO] [1668798370.440091713] [rclcpp]: Incoming re
quest (x,y,z): ('0.064500','1.665000','1.101000')
[INFO] [1668798370.440142458] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('1.010000','0.990000','1.000
000')
[INFO] [1668798370.440155084] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('2.060000','-0.990000','1.00
0000')
[INFO] [1668798453.251864949] [rclcpp]: Incoming re
quest (x,y,z): ('0.190000','0.012200','1.361000')
[INFO] [1668798453.251894378] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('-1.980000','2.970000','0.74
0000')
[INFO] [1668798453.251901382] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('2.090000','-2.970000','0.74
0000')
[INFO] [1668798603.061612474] [rclcpp]: Incoming re
quest (x,y,z): ('-0.820000','1.690000','1.026900')
[INFO] [1668798603.061633474] [rclcpp]: Outgoing Re
sponse (q11,q21,q31): ('1.860000','0.300000','1.070
000')
[INFO] [1668798603.061663994] [rclcpp]: Outgoing Re
sponse (q12,q22,q32): ('2.180000','-0.300000','1.07
0000')
```

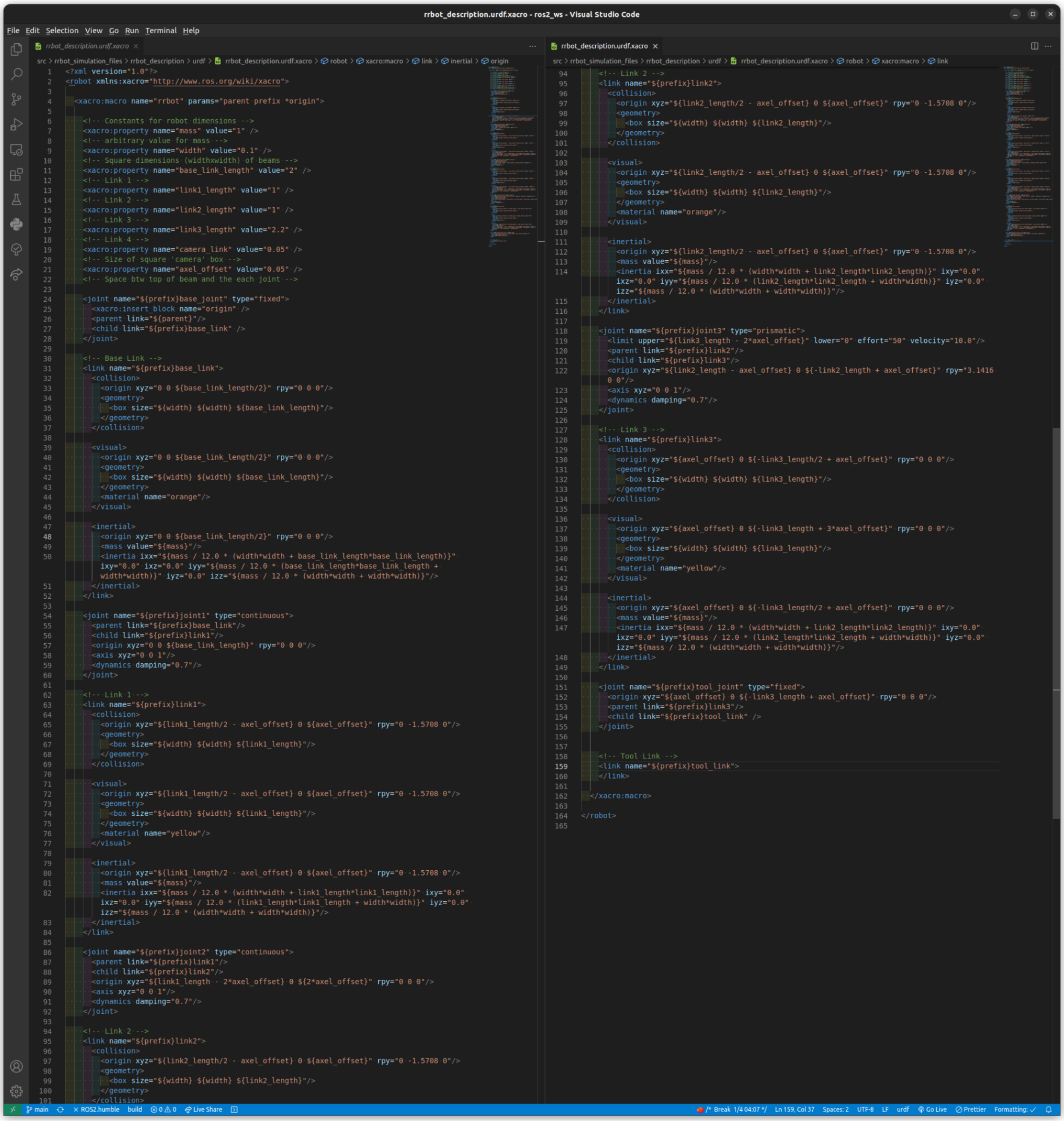
```
ct$ ros2 service call /inverse_kinematics custom_in
terfaces/srv/FindJointStates "{x: 0.19, y: 0.0122,
z: 1.361}"
requester: making request: custom_interfaces.srv.Fi
ndJointStates_Request(x=0.19, y=0.0122, z=1.361)

response:
custom_interfaces.srv.FindJointStates_Response(q11=
-1.98, q21=2.97, q31=0.74, q12=2.09, q22=-2.97, q32
=0.74)

cameronschloer@cs-laptop:~/Code/RBE-500-Final-Proje
ct$ ros2 service call /inverse_kinematics custom_in
terfaces/srv/FindJointStates "{x: -0.82, y: 1.69, z
: 1.0269}"
requester: making request: custom_interfaces.srv.Fi
ndJointStates_Request(x=-0.82, y=1.69, z=1.0269)

response:
custom_interfaces.srv.FindJointStates_Response(q11=
1.86, q21=0.3, q31=1.07, q12=2.18, q22=-0.3, q32=1.
07)

cameronschloer@cs-laptop:~/Code/RBE-500-Final-Proje
ct$
```

```
1.86862,0.292796,1.0731
[INFO] [1668798637.792019328] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798637.891907155] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798637.992081610] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798638.091790858] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798638.192298763] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798638.292485911] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798638.392411763] [minimal_publisher]:
Publishing...
1.86862,0.292796,1.0731
[INFO] [1668798638.492188688] [minimal_publisher]:
Publishing...
```

```
    size: 4
    stride: 4
  - label: height
    size: 4
    stride: 0
  data_offset: 0
data:
- -0.5568756164881876
- -0.8305958991958129
- 0.0
- -0.8209713687792594
- 0.8305958991958129
- -0.5568756164881876
- 0.0
- 1.6909757185906038
- 0.0
- 0.0
- 1.0
- 1.0269
- 0.0
- 0.0
- 0.0
- 1.0
---
```

Figure 5: Third position that tests Forward and Inverse Kinematics

Robot URDF

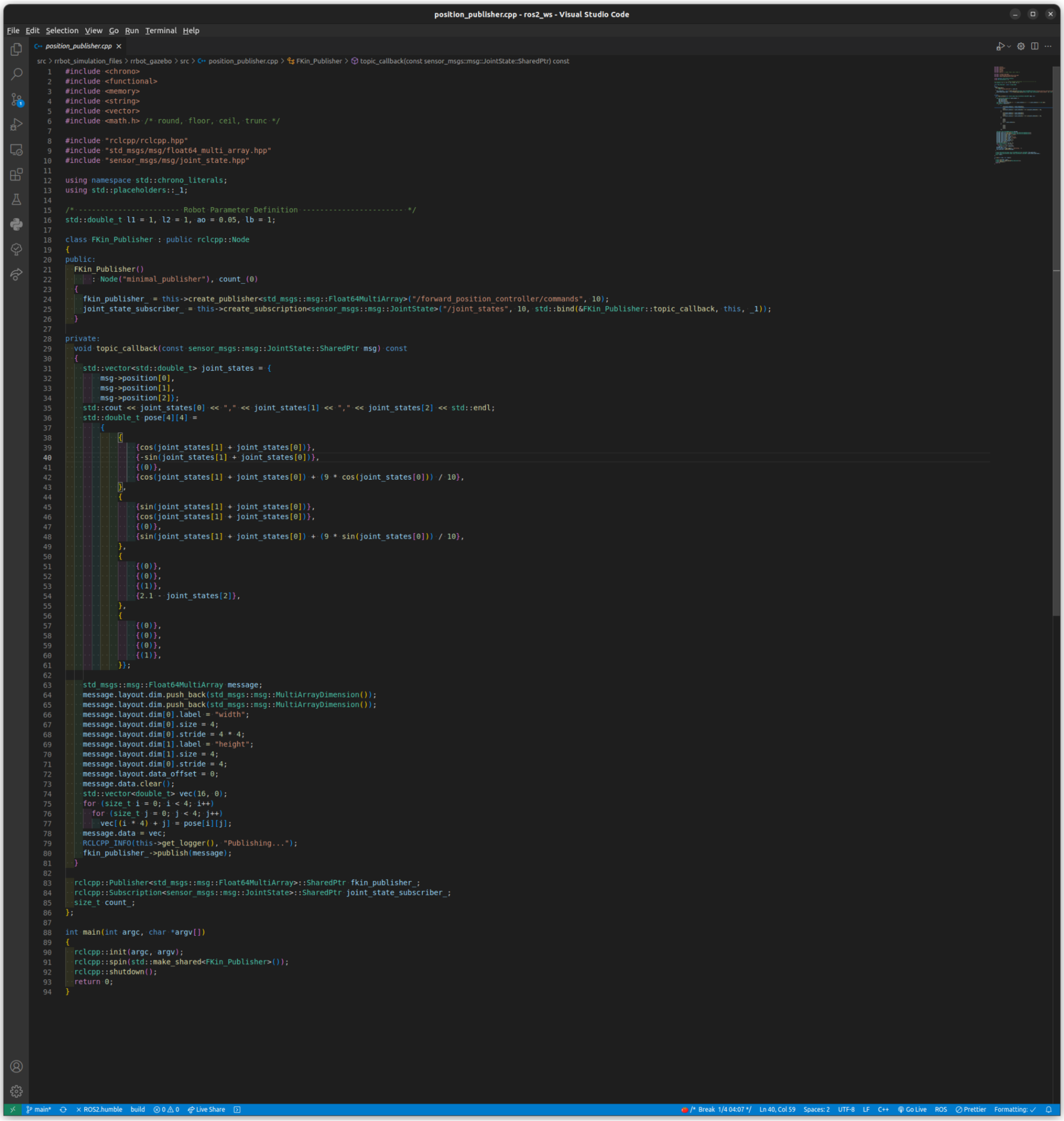

rrbot_description.urdf.xacro - ros2_ws - Visual Studio Code
File Edit Selection View Go Run Terminal Help
rrbot_description.urdf.xacro
src > rrbot_simulation_files > rrbot_description > urdf > rrbot_description.urdf.xacro > robot > xacro:macro > link > inertial > origin
rrbot_description.urdf.xacro
src > rrbot_simulation_files > rrbot_description > urdf > rrbot_description.urdf.xacro > robot > xacro:macro > link
main
ROS2.humble
build
0 0
Live Share
/* Break 1/4 04:07 */
Ln 159, Col 37
Spaces: 2
UTF-8
LF
urdf
Go Live
Prettier
Formatting: ✓

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">

  <xacro:macro name="rrbot" params="parent prefix *origin">

    <!-- Constants for robot dimensions -->
    <xacro:property name="mass" value="1" />
    <!-- arbitrary value for mass -->
    <xacro:property name="width" value="0.1" />
    <!-- Square dimensions (widthxwidth) of beams -->
    <xacro:property name="base_link_length" value="2" />
    <!-- Link 1 -->
    <xacro:property name="link1_length" value="1" />
    <!-- Link 2 -->
    <xacro:property name="link2_length" value="1" />
    <!-- Link 3 -->
    <xacro:property name="link3_length" value="2.2" />
    <!-- Link 4 -->
    <xacro:property name="camera_link" value="0.05" />
    <!-- Size of square 'camera' box -->
    <xacro:property name="axel_offset" value="0.05" />
    <!-- Space btw top of beam and the each joint -->

    <joint name="${prefix}base_joint" type="fixed">
      <xacro:insert_block name="origin" />
      <parent link="${parent}"/>
      <child link="${prefix}base_link" />
    </joint>

    <!-- Base Link -->
    <link name="${prefix}base_link">
      <collision>
        <origin xyz="0 0 ${base_link_length/2}" rpy="0 0 0"/>
        <geometry>
          <box size="${width} ${width} ${base_link_length}"/>
        </geometry>
      </collision>

      <visual>
        <origin xyz="0 0 ${base_link_length/2}" rpy="0 0 0"/>
        <geometry>
          <box size="${width} ${width} ${base_link_length}"/>
        </geometry>
        <material name="orange"/>
      </visual>

      <inertial>
        <origin xyz="0 0 ${base_link_length/2}" rpy="0 0 0"/>
        <mass value="${mass}"/>
        <inertia ixx="${mass / 12.0 * (width*width + base_link_length*base_link_length)}" ixy="0.0" ixz="0.0" iyy="${mass / 12.0 * (base_link_length*base_link_length + width*width)}" iyz="0.0" izz="${mass / 12.0 * (width*width + width*width)}"/>
      </inertial>
    </link>

    <joint name="${prefix}joint1" type="continuous">
      <parent link="${prefix}base_link"/>
      <child link="${prefix}link1"/>
      <origin xyz="0 0 ${base_link_length}" rpy="0 0 0"/>
      <axis xyz="0 0 1"/>
      <dynamics damping="0.7"/>
    </joint>

    <!-- Link 1 -->
    <link name="${prefix}link1">
      <collision>
        <origin xyz="${link1_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <geometry>
          <box size="${width} ${width} ${link1_length}"/>
        </geometry>
      </collision>

      <visual>
        <origin xyz="${link1_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <geometry>
          <box size="${width} ${width} ${link1_length}"/>
        </geometry>
        <material name="yellow"/>
      </visual>

      <inertial>
        <origin xyz="${link1_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <mass value="${mass}"/>
        <inertia ixx="${mass / 12.0 * (width*width + link1_length*link1_length)}" ixy="0.0" ixz="0.0" iyy="${mass / 12.0 * (link1_length*link1_length + width*width)}" iyz="0.0" izz="${mass / 12.0 * (width*width + width*width)}"/>
      </inertial>
    </link>

    <joint name="${prefix}joint2" type="continuous">
      <parent link="${prefix}link1"/>
      <child link="${prefix}link2"/>
      <origin xyz="${link1_length - 2*axel_offset} 0 ${2*axel_offset}" rpy="0 0 0"/>
      <axis xyz="0 0 1"/>
      <dynamics damping="0.7"/>
    </joint>

    <!-- Link 2 -->
    <link name="${prefix}link2">
      <collision>
        <origin xyz="${link2_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <geometry>
          <box size="${width} ${width} ${link2_length}"/>
        </geometry>
      </collision>
```

```
    <!-- Link 2 -->
    <link name="${prefix}link2">
      <collision>
        <origin xyz="${link2_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <geometry>
          <box size="${width} ${width} ${link2_length}"/>
        </geometry>
      </collision>
```

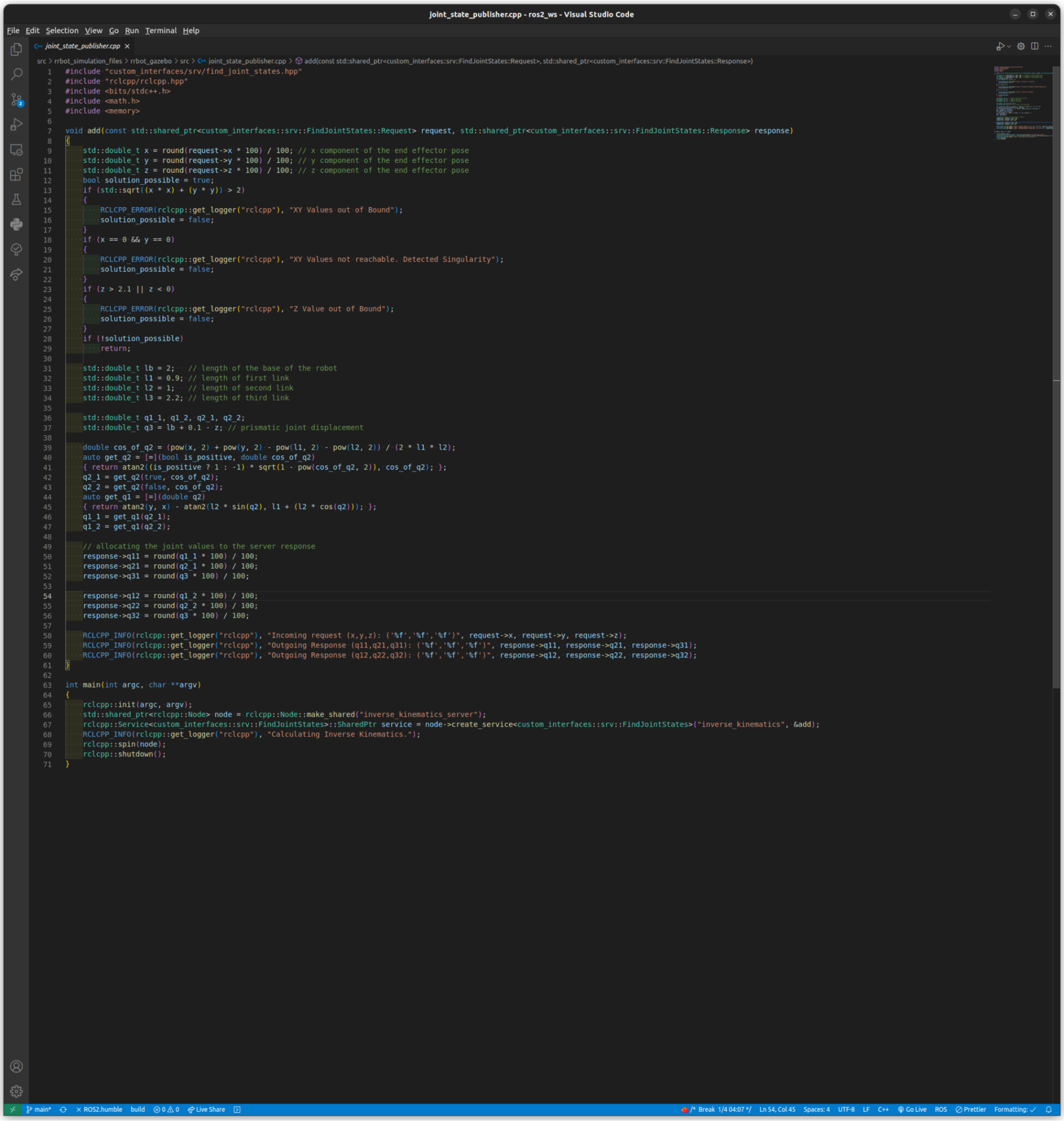
```

      <visual>
        <origin xyz="${link2_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <geometry>
          <box size="${width} ${width} ${link2_length}"/>
        </geometry>
        <material name="orange"/>
      </visual>

      <inertial>
        <origin xyz="${link2_length/2 - axel_offset} 0 ${axel_offset}" rpy="0 -1.5708 0"/>
        <mass value="${mass}"/>
        <inertia ixx="${mass / 12.0 * (width*width + link2_length*link2_length)}" ixy="0.0" ixz="0.0" iyy="${mass / 12.0 * (link2_length*link2_length + width*width)}" iyz="0.0" izz="${mass / 12.0 * (width*width + width*width)}"/>
      </inertial>
    </link>

    <joint name="${prefix}joint3" type="prismatic">
      <limit upper="${link3_length - 2*axel_offset}" lower="0" effort="50" velocity="10.0"/>
      <parent link="${prefix}link2"/>
      <child link="${prefix}link3"/>
      <origin xyz="${link2_length - axel_offset} 0 ${-link2_length + axel_offset}" rpy="3.1416 0 0"/>
      <axis xyz="0 0 1"/>
      <dynamics damping="0.7"/>
    </joint>

    <!-- Link 3 -->
    <link name="${prefix}link3">
      <collision>
        <origin xyz="${axel_offset} 0 ${-link3_length/2 + axel_offset}" rpy="0 0 0"/>
        <geometry>
          <box size="${width} ${width} ${link3_length}"/>
        </geometry>
      </collision>

      <visual>
        <origin xyz="${axel_offset} 0 ${-link3_length + 3*axel_offset}" rpy="0 0 0"/>
        <geometry>
          <box size="${width} ${width} ${link3_length}"/>
        </geometry>
        <material name="yellow"/>
      </visual>

      <inertial>
        <origin xyz="${axel_offset} 0 ${-link3_length/2 + axel_offset}" rpy="0 0 0"/>
        <mass value="${mass}"/>
        <inertia ixx="${mass / 12.0 * (width*width + link2_length*link2_length)}" ixy="0.0" ixz="0.0" iyy="${mass / 12.0 * (link2_length*link2_length + width*width)}" iyz="0.0" izz="${mass / 12.0 * (width*width + width*width)}"/>
      </inertial>
    </link>

    <joint name="${prefix}tool_joint" type="fixed">
      <origin xyz="${axel_offset} 0 ${-link3_length + axel_offset}" rpy="0 0 0"/>
      <parent link="${prefix}link3"/>
      <child link="${prefix}tool_link" />
    </joint>


    <!-- Tool Link -->
    <link name="${prefix}tool_link">
    </link>

  </xacro:macro>

</robot>
```

Forward Kinematics Node


position_publisher.cpp - ros2_ws - Visual Studio Code
File Edit Selection View Go Run Terminal Help
position_publisher.cpp
src > rrbot_simulation_files > rrbot_gazebo > src > position_publisher.cpp > FKin_Publisher > topic_callback(const sensor_msgs::msg::JointState::SharedPtr) const
main*
ROS2.humble
build
Live Share
/* Break 1/4 04:07 */
Ln 40, Col 59
Spaces: 2
UTF-8
LF
C++
Go Live
ROS
Prettier
Formatting: ✓

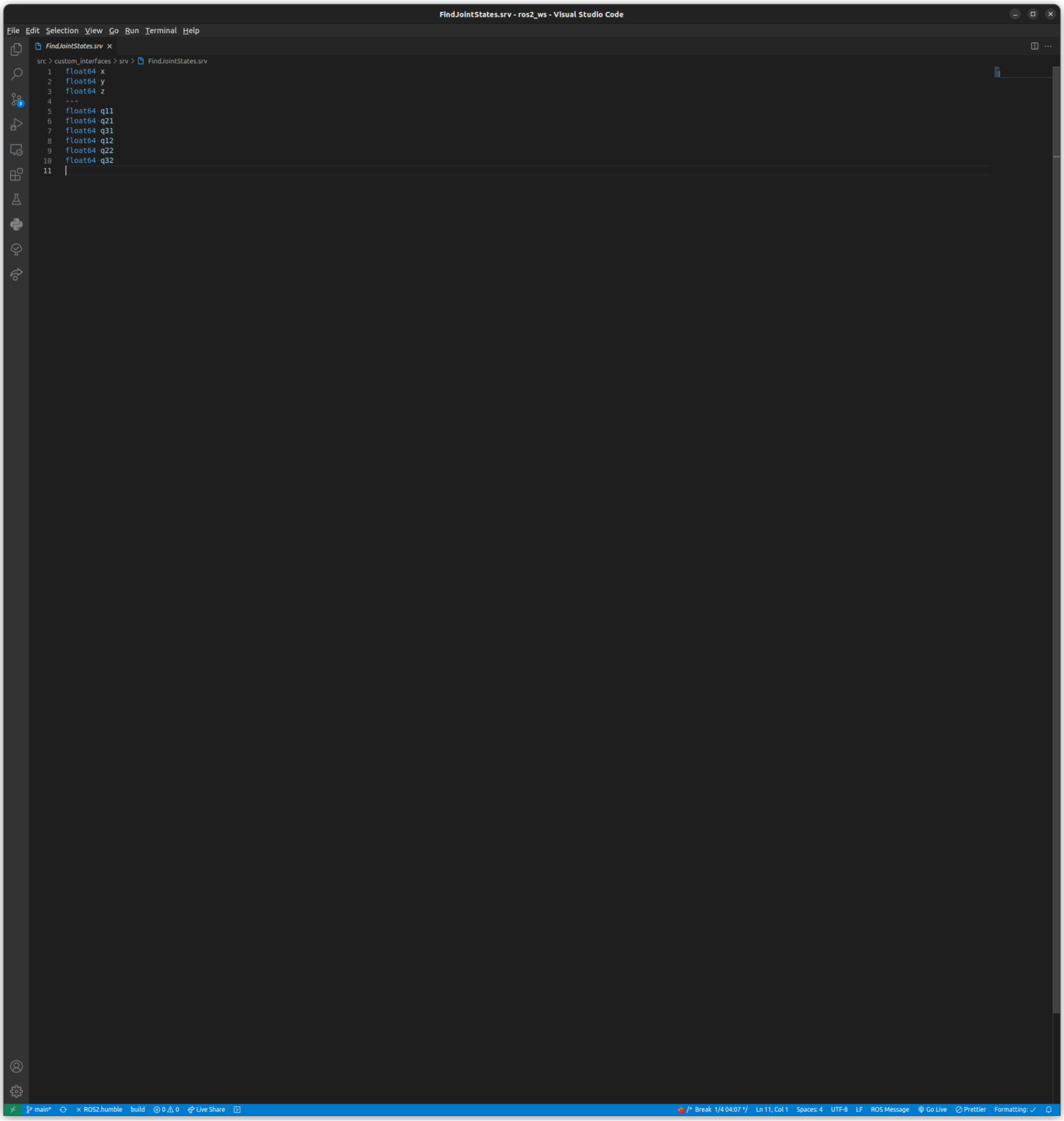
```cpp
#include <chrono>
#include <functional>
#include <memory>
#include <string>
#include <vector>
#include <math.h> /* round, floor, ceil, trunc */

#include "rclcpp/rclcpp.hpp"
#include "std_msgs/msg/float64_multi_array.hpp"
#include "sensor_msgs/msg/joint_state.hpp"

using namespace std::chrono_literals;
using std::placeholders::_1;

/* ---------------------- Robot Parameter Definition ---------------------- */
std::double_t l1 = 1, l2 = 1, ao = 0.05, lb = 1;

class FKin_Publisher : public rclcpp::Node
{
public:
  FKin_Publisher()
      : Node("minimal_publisher"), count_(0)
  {
    fkin_publisher_ = this->create_publisher<std_msgs::msg::Float64MultiArray>("/forward_position_controller/commands", 10);
    joint_state_subscriber_ = this->create_subscription<sensor_msgs::msg::JointState>("/joint_states", 10, std::bind(&FKin_Publisher::topic_callback, this, _1));
  }

private:
  void topic_callback(const sensor_msgs::msg::JointState::SharedPtr msg) const
  {
    std::vector<std::double_t> joint_states = {
        msg->position[0],
        msg->position[1],
        msg->position[2]};
    std::cout << joint_states[0] << "," << joint_states[1] << "," << joint_states[2] << std::endl;
    std::double_t pose[4][4] =
        {
            {
                {cos(joint_states[1] + joint_states[0])},
                {-sin(joint_states[1] + joint_states[0])},
                {(0)},
                {cos(joint_states[1] + joint_states[0]) + (9 * cos(joint_states[0])) / 10},
            },
            {
                {sin(joint_states[1] + joint_states[0])},
                {cos(joint_states[1] + joint_states[0])},
                {(0)},
                {sin(joint_states[1] + joint_states[0]) + (9 * sin(joint_states[0])) / 10},
            },
            {
                {(0)},
                {(0)},
                {(1)},
                {2.1 - joint_states[2]},
            },
            {
                {(0)},
                {(0)},
                {(0)},
                {(1)},
            }};

    std_msgs::msg::Float64MultiArray message;
    message.layout.dim.push_back(std_msgs::msg::MultiArrayDimension());
    message.layout.dim.push_back(std_msgs::msg::MultiArrayDimension());
    message.layout.dim[0].label = "width";
    message.layout.dim[0].size = 4;
    message.layout.dim[0].stride = 4 * 4;
    message.layout.dim[1].label = "height";
    message.layout.dim[1].size = 4;
    message.layout.dim[0].stride = 4;
    message.layout.data_offset = 0;
    message.data.clear();
    std::vector<double_t> vec(16, 0);
    for (size_t i = 0; i < 4; i++)
      for (size_t j = 0; j < 4; j++)
        vec[(i * 4) + j] = pose[i][j];
    message.data = vec;
    RCLCPP_INFO(this->get_logger(), "Publishing...");
    fkin_publisher_->publish(message);
  }

  rclcpp::Publisher<std_msgs::msg::Float64MultiArray>::SharedPtr fkin_publisher_;
  rclcpp::Subscription<sensor_msgs::msg::JointState>::SharedPtr joint_state_subscriber_;
  size_t count_;
};

int main(int argc, char *argv[])
{
  rclcpp::init(argc, argv);
  rclcpp::spin(std::make_shared<FKin_Publisher>());
  rclcpp::shutdown();
  return 0;
}
```

Inverse Kinematics Service


joint_state_publisher.cpp - ros2_ws - Visual Studio Code
File Edit Selection View Go Run Terminal Help
joint_state_publisher.cpp
src > rrbot_simulation_files > rrbot_gazebo > src > joint_state_publisher.cpp > add(const std::shared_ptr<custom_interfaces::srv::FindJointStates::Request>, std::shared_ptr<custom_interfaces::srv::FindJointStates::Response>)
main*
ROS2.humble
build
0 ⚠ 0
Live Share
/* Break 1/4 04:07 */
Ln 54, Col 45
Spaces: 4
UTF-8
LF
C++
Go Live
ROS
Prettier
Formatting: ✓

```cpp
#include "custom_interfaces/srv/find_joint_states.hpp"
#include "rclcpp/rclcpp.hpp"
#include <bits/stdc++.h>
#include <math.h>
#include <memory>

void add(const std::shared_ptr<custom_interfaces::srv::FindJointStates::Request> request, std::shared_ptr<custom_interfaces::srv::FindJointStates::Response> response)
{
    std::double_t x = round(request->x * 100) / 100; // x component of the end effector pose
    std::double_t y = round(request->y * 100) / 100; // y component of the end effector pose
    std::double_t z = round(request->z * 100) / 100; // z component of the end effector pose
    bool solution_possible = true;
    if (std::sqrt((x * x) + (y * y)) > 2)
    {
        RCLCPP_ERROR(rclcpp::get_logger("rclcpp"), "XY Values out of Bound");
        solution_possible = false;
    }
    if (x == 0 && y == 0)
    {
        RCLCPP_ERROR(rclcpp::get_logger("rclcpp"), "XY Values not reachable. Detected Singularity");
        solution_possible = false;
    }
    if (z > 2.1 || z < 0)
    {
        RCLCPP_ERROR(rclcpp::get_logger("rclcpp"), "Z Value out of Bound");
        solution_possible = false;
    }
    if (!solution_possible)
        return;

    std::double_t lb = 2;   // length of the base of the robot
    std::double_t l1 = 0.9; // length of first link
    std::double_t l2 = 1;   // length of second link
    std::double_t l3 = 2.2; // length of third link

    std::double_t q1_1, q1_2, q2_1, q2_2;
    std::double_t q3 = lb + 0.1 - z; // prismatic joint displacement

    double cos_of_q2 = (pow(x, 2) + pow(y, 2) - pow(l1, 2) - pow(l2, 2)) / (2 * l1 * l2);
    auto get_q2 = [=](bool is_positive, double cos_of_q2)
    { return atan2((is_positive ? 1 : -1) * sqrt(1 - pow(cos_of_q2, 2)), cos_of_q2); };
    q2_1 = get_q2(true, cos_of_q2);
    q2_2 = get_q2(false, cos_of_q2);
    auto get_q1 = [=](double q2)
    { return atan2(y, x) - atan2(l2 * sin(q2), l1 + (l2 * cos(q2))); };
    q1_1 = get_q1(q2_1);
    q1_2 = get_q1(q2_2);

    // allocating the joint values to the server response
    response->q11 = round(q1_1 * 100) / 100;
    response->q21 = round(q2_1 * 100) / 100;
    response->q31 = round(q3 * 100) / 100;

    response->q12 = round(q1_2 * 100) / 100;
    response->q22 = round(q2_2 * 100) / 100;
    response->q32 = round(q3 * 100) / 100;

    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Incoming request (x,y,z): ('%f','%f','%f')", request->x, request->y, request->z);
    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Outgoing Response (q11,q21,q31): ('%f','%f','%f')", response->q11, response->q21, response->q31);
    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Outgoing Response (q12,q22,q32): ('%f','%f','%f')", response->q12, response->q22, response->q32);
}

int main(int argc, char **argv)
{
    rclcpp::init(argc, argv);
    std::shared_ptr<rclcpp::Node> node = rclcpp::Node::make_shared("inverse_kinematics_server");
    rclcpp::Service<custom_interfaces::srv::FindJointStates>::SharedPtr service = node->create_service<custom_interfaces::srv::FindJointStates>("inverse_kinematics", &add);
    RCLCPP_INFO(rclcpp::get_logger("rclcpp"), "Calculating Inverse Kinematics.");
    rclcpp::spin(node);
    rclcpp::shutdown();
}
```

Custom Service


FindJointStates.srv - ros2_ws - Visual Studio Code
File Edit Selection View Go Run Terminal Help
FindJointStates.srv
src > custom_interfaces > srv > FindJointStates.srv
main* ROS2.humble build 0 0 Live Share
/* Break 1/4 04:07 */ Ln 11, Col 1 Spaces: 4 UTF-8 LF ROS Message Go Live Prettier Formatting: ✓

```
float64 x
float64 y
float64 z
---
float64 q11
float64 q21
float64 q31
float64 q12
float64 q22
float64 q32

```

```
clc;
clear all;
close all;
```

Defining DH Parameters

```
syms l1 l2 lb theta1 theta2 d3 ao
% Theta D A Alpha
l1_dh = [0 lb 0 0]
```

l1_dh = $\begin{bmatrix} 0 & \mathrm{lb} & 0 & 0 \end{bmatrix}$

```
l2_dh = [theta1 2*ao l1-2*ao 0]
```

l2_dh = $\begin{bmatrix} \theta_1 & 2\,\mathrm{ao} & l_1 - 2\,\mathrm{ao} & 0 \end{bmatrix}$

```
l3_dh = [theta2 ao l2-ao 0]
```

l3_dh = $\begin{bmatrix} \theta_2 & \mathrm{ao} & l_2 - \mathrm{ao} & 0 \end{bmatrix}$

```
l4_dh = [0 d3-ao ao 0]
```

l4_dh = $\begin{bmatrix} 0 & d_3 - \mathrm{ao} & \mathrm{ao} & 0 \end{bmatrix}$

Making Symbolic A Matrix

```
syms theta d a alpha
A = [cos(theta) -sin(theta)*cos(alpha) sin(theta)*sin(alpha) a*cos(theta);
    sin(theta) cos(theta)*cos(alpha) -cos(theta)*sin(alpha) a*sin(theta);
    0 sin(alpha) cos(alpha) d;
    0 0 0 1]
```

A =

$$\begin{bmatrix} \cos(\theta) & -\cos(\alpha)\sin(\theta) & \sin(\alpha)\sin(\theta) & a\cos(\theta) \\ \sin(\theta) & \cos(\alpha)\cos(\theta) & -\sin(\alpha)\cos(\theta) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

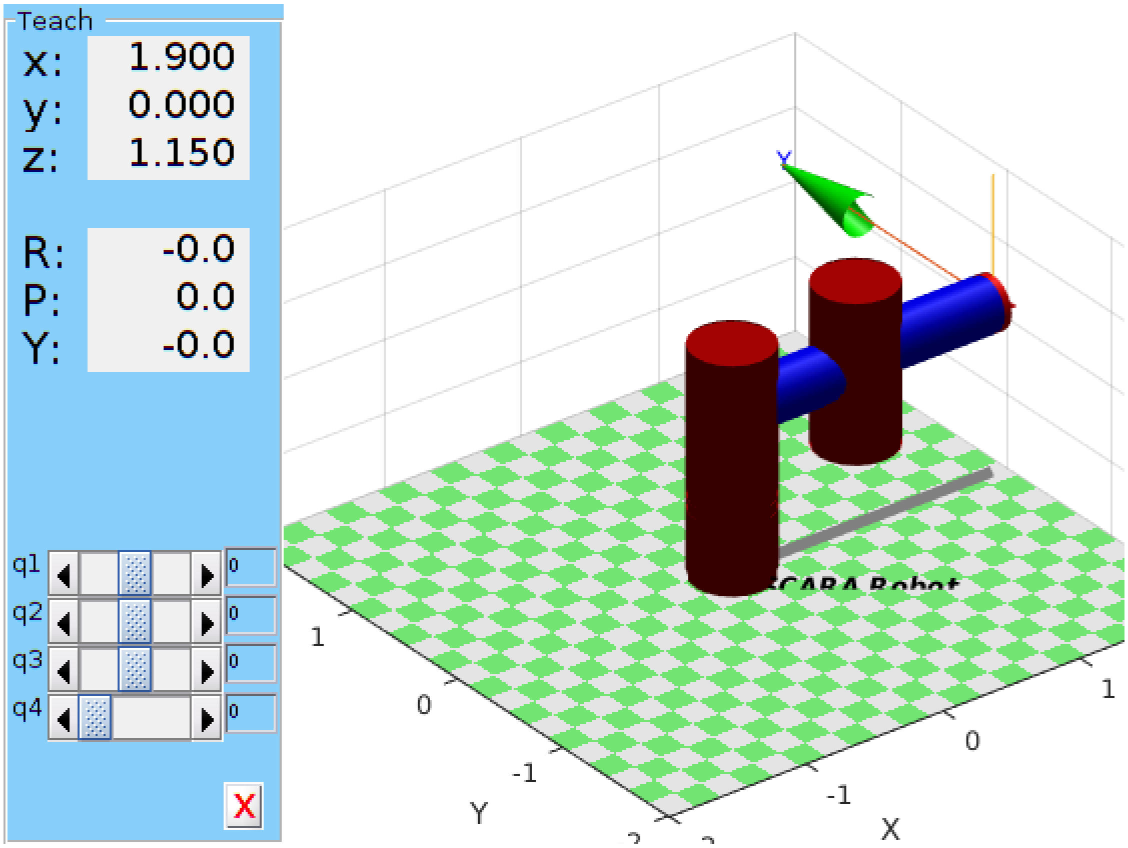
```
A1 = simplify(subs(A,[theta,d,a,alpha],l1_dh));
A2 = simplify(subs(A,[theta,d,a,alpha],l2_dh));
A3 = simplify(subs(A,[theta,d,a,alpha],l3_dh));
A4 = simplify(subs(A,[theta,d,a,alpha],l4_dh));
T = A1*A2*A3*A4;
T = simplify(subs(T,[l1,l2,lb,ao],[1,1,2,0.05]))
```

T =

$$\begin{bmatrix} \cos(\theta_2+\theta_1) & -\sin(\theta_2+\theta_1) & 0 & \cos(\theta_2+\theta_1)+\dfrac{9\cos(\theta_1)}{10} \\ \sin(\theta_2+\theta_1) & \cos(\theta_2+\theta_1) & 0 & \sin(\theta_2+\theta_1)+\dfrac{9\sin(\theta_1)}{10} \\ 0 & 0 & 1 & \dfrac{21}{10}+d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Modelling the Robot from DH Parameters

```
L(1) =  Link(double(subs(l1_dh,[l1,l2,lb,theta1,theta2,d3,ao],[1,1,1,0,0,1.2,0.05])),'s
L(1).qlim = pi/180 * [-90 90];
L(2) =  Link(double(subs(l2_dh,[l1,l2,lb,theta1,theta2,d3,ao],[1,1,1,0,0,1.2,0.05])),'s
L(2).qlim = pi/180 * [-90 90];
L(3) =  Link(double(subs(l3_dh,[l1,l2,lb,theta1,theta2,d3,ao],[1,1,1,0,0,1.2,0.05])),'s
L(3).qlim = pi/180 * [-90 90];
L(4) =  Link([double(subs(l4_dh,[l1,l2,lb,theta1,theta2,d3,ao],[1,1,1,0,0,1.2,0.05])),1
L(4).qlim = [0 2];
scara_robot = SerialLink(L);
scara_robot.name = 'SCARA Robot';
scara_robot.plot([0 0 0 0],'workspace',[-2 2 -2 2 0 2])
scara_robot.teach
```



Teach

x: 1.900
y: 0.000
z: 1.150

R: -0.0
P: 0.0
Y: -0.0

q1 0
q2 0
q3 0
q4 0