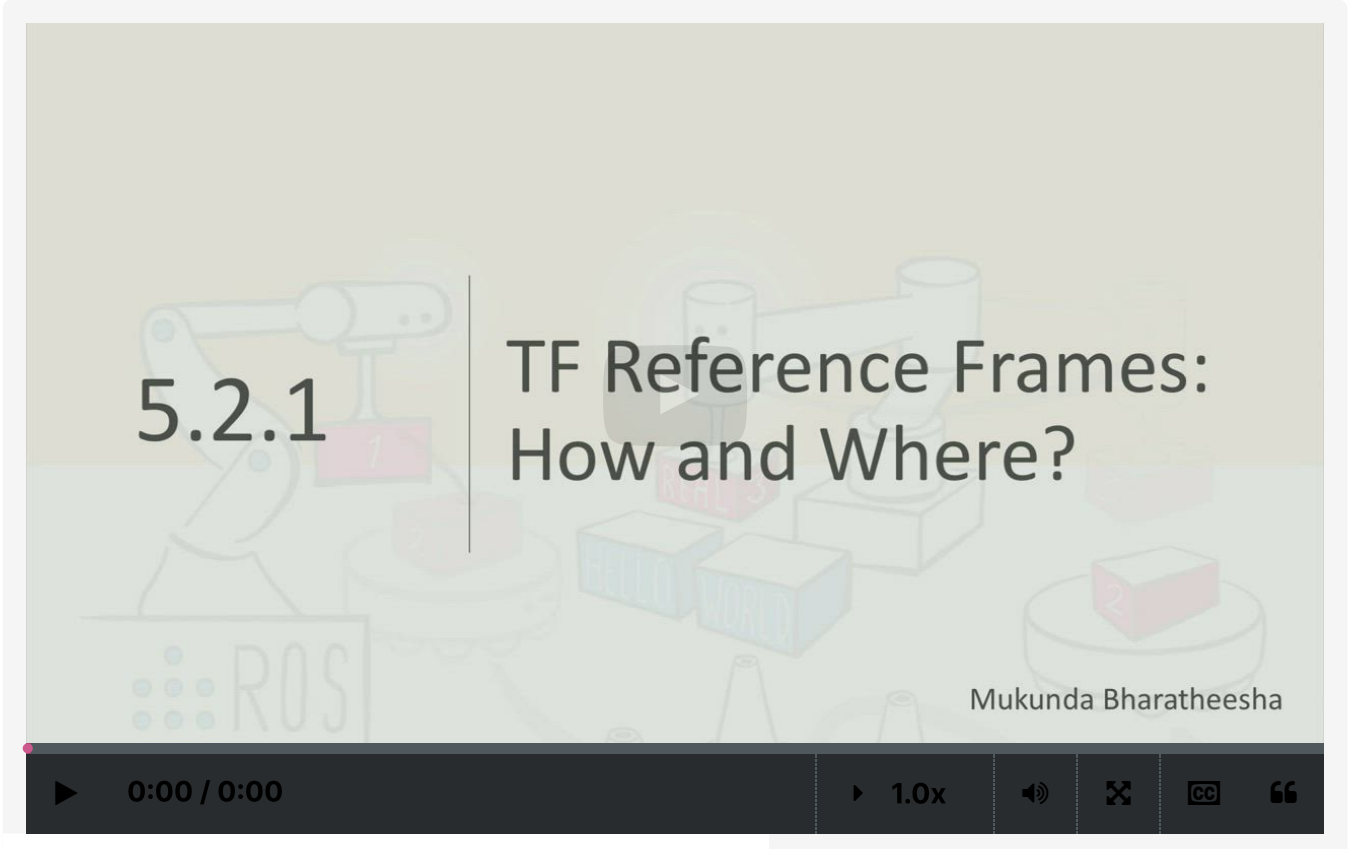
In this lecture, we will learn how the TF reference frames are generated and where they are located.

Important Note:
In the version of the files you received, the combined joint states of the robot arms are in a different topic named ***combined_joint_states*** whereas the turtlebot joint states remain on the ***joint_states*** topic.
This facilitates control and execution of the full factory simulation.

This change of topic names is achieved via the `<remap>` tag, that you already used for the MoveIt Setup Assistant configuration in Week 4.

TF Reference Frames



Video ▼ Subtitles ▼

Other ▼

TF reference frames are generated and update by the ROS package `robot_state_publisher`. This package uses two source information to generate and keep track of the reference frames:

- The joint state information is published to the `/joints_states topics`:
  - For the turtlebot is published on the `/joint_states` topic
  - For the manipulators is published on the `/combined_joint_states` topic.
- The robot_description parameter is populated from the URDF/XACRO fo;es (`hrwros.xacro`)

In addition, the `robot_state_publisher` also uses the root link of the XACRO file as the fixed frame of reference. This fix reference is typically referred to as the "world".

Joint States

The `joint_state_publisher` is used to combine the joint state information from the two robot arms, which is published to the `combined_joint_state` topic. This is achieved via the `source_list` parameter.

Note that another `joint_state_publisher` publishes information to the `joint_state` topic, only for the turtlebot. Recall from week 1 that it is possible for multiple nodes to publish to the same topic!

By using the rostopic command, you can check that there are indeed two publisher publishing to the topic:

```
$ rostopic info joint_states
```

Include URDF

The URDF should be loaded on to the parameter server (in the `hrwors_environment.launch file)` by including the `load_hrwros.lauch` file:

```
<include file="$(find
hrwros_support)/launch/load_hrwros.launch"/>
```

This file contains the `robot_description` parameter. This gets updates with the URDF elements. Recall from week 2 that the URDF defines your environment geometry, in other words it defines where different objects are.

Location Reference Frames

Now that we know that the reference frames are created within ROS, we need to find where they are. The origin tag defines where the TF frames are place. In the example in the video the origin tag is as following:
`<origin xyz="-7.8 -1.5 0"/>`

Objects are defined via **links** which are connected via **joints**. That is why the TF frame are defined at joint **origins** so that we know how two links are located relative to each other.

Question 1

1 point possible (ungraded)
Which of the following statements is true?

- ◯ The ROS package responsible for generating TF frames is the joint_states_publisher.
- ◯ The ROS package responsible for generating TF frames is the robot_state_publisher.
- ◯ The ROS package responsible for generating TF frames is Gazebo.

- ◯ The ROS package responsible for generating TF frames is MoveIt

Submit

Question 2

1 point possible (ungraded)

Where are TF frames located once they are generated?

- ◯ Link origin.
- ◯ Joint origin.
- ◯ Both link origin and joint origin.
- ◯ None of the above.

Submit