In this lecture, we will explore the `static_transform_publisher` command line tool.

Important Note:

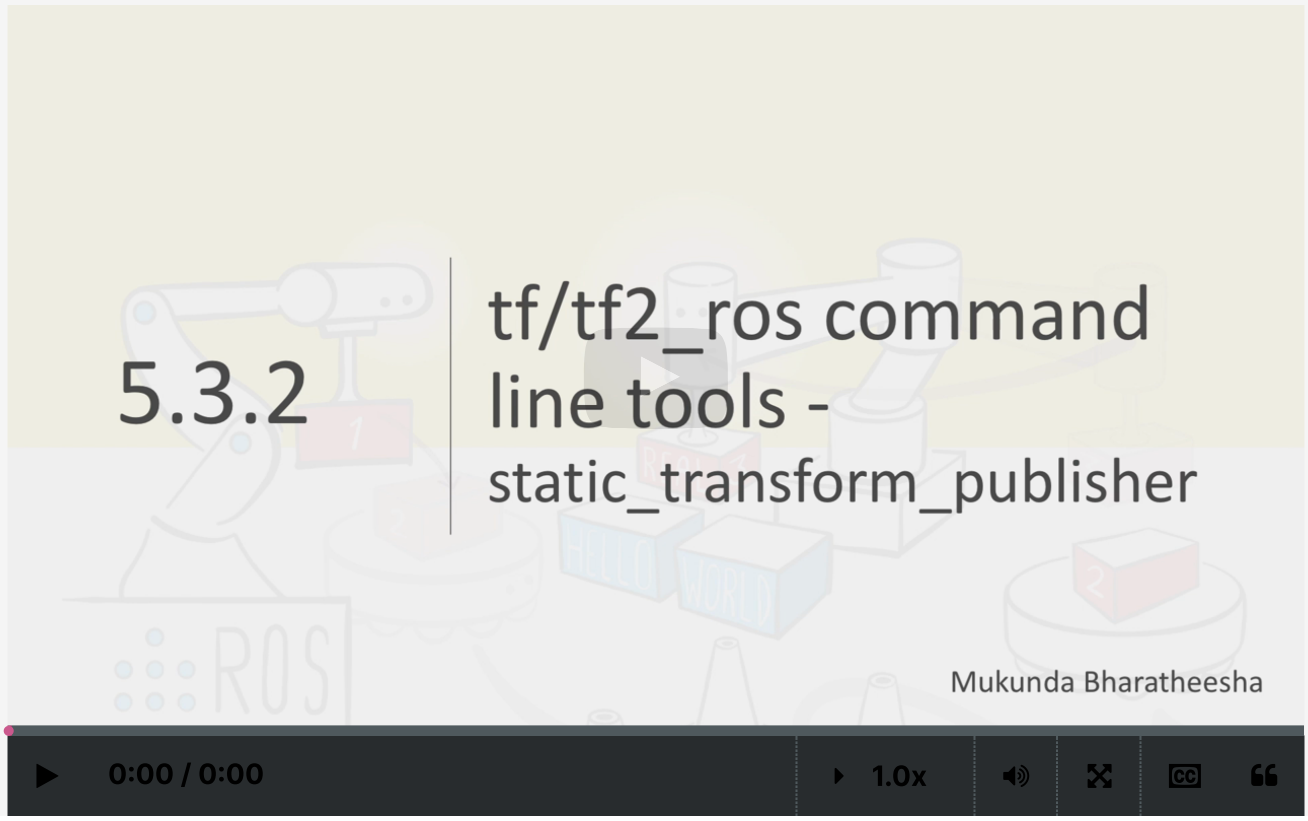
The following command is not available in ROS Noetic:

```
$ rosrun tf view_frames
```

Instead you can use this one:

```
$ rosrun tf2_tools view_frames.py
```

5.3.2 tf/tf2 ros command line tools - static_transform_publisher



Video ▼ Subtitles ▼

Subtitles (captions) in other languages than provided can be viewed at YouTube. Select your language in the CC-button of YouTube.

Let's go back to the terminal of last lecture.

If you took a break in between, or closed it, just open a new one, and restart the factory simulation.

```
$ roslaunch hrwros_gazebo hrwros_environment.launch
```

Then, open a new CCS, source it, and run the following command:

```
$ rosrun tf2_ros static_transform_publisher 0 0 0 0 0 0 1
test_parent test_child
```

Notice, we have called the parent frame `test_parent` and the child frame `test_child`. After running the command above, we see a new ROS node is created, that is publishing a static transform. It will keep running until it is killed.

In another new, and sourced, CCS shell, we can see this nod by looking at `rosnode list`. You should be able to see the node name from the previous command.

Let's look at the TF tree again by generating a new TF tree PDF file. Before doing that, you might want to rename the old file instead of overwriting it.

```
$ rosrun tf2_tools view_frames.py
```

Take a look at the new PDF file:

- We have a new pair of frames.
- We also have two root frames: map, and test_parent.
- This is an example of a disconnected TF tree! This is almost never wanted behaviour with ROS, with it will work fine if you actually want that.
- For example, the tf_echo will now throw an error between disconnected transforms:
  ```
  $ rosrun tf tf_echo test_child base_link
  ```

We can actually kill the transform we created at the start, and make a new one between a new TF, and an already existing transform:

```
$ rosrun tf2_ros static_transform_publisher 0 0 0 0 0 0 1
map test_child
```

Then, if we create the TF tree again, we will see that the tree is completely connected again! If we new ask `tf_echo` to tell us the connections between `test_child` and `base_link`, it will be happy to tell us.

Question 1

1 point possible (ungraded)

Using a static transform publisher always creates a broken TF tree.

○ True

○ False

Submit