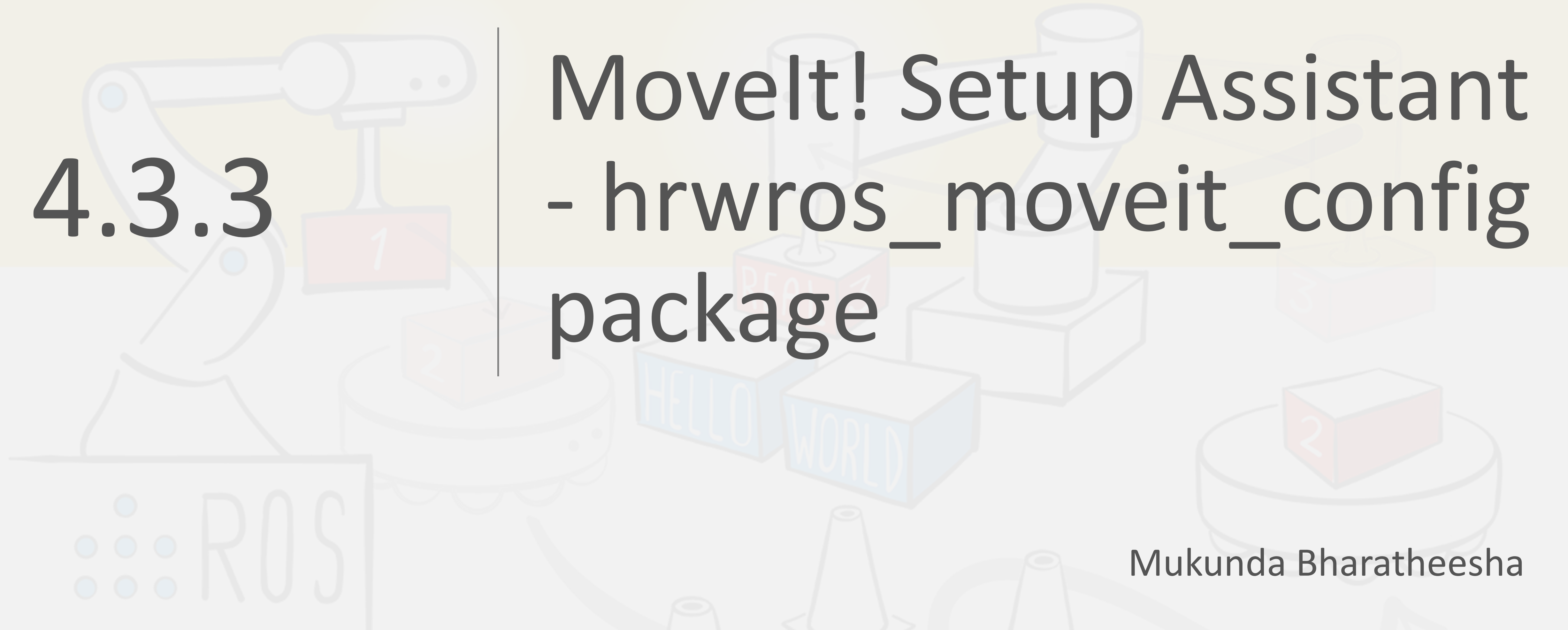

4.3.3

MoveIt! Setup Assistant - hrwros_moveit_config package

Mukunda Bharatheesha

Build configuration package

After creating a new package we build it and source our updated setup files.

```
$ cd $HOME/hrwros_ws
$ catkin build
$ source $HOME/hrwros_ws/devel/setup.bash
```

Navigate and look into the package.xml file.

```
$ roscd hrwros_moveit_config/
$ more package.xml
```

There you can see the author details and a set of dependencies.

Contents of the package

Open the *hrwros_moveit_config* package, in an external editor, by looking at the launch files we can see how it works.

Let's begin with the **launch/move_group.launch** file.

At the beginning of the file there are two group of settings related to code debugging and logging settings. These will be untouched in this course.

Then we have a few parameters:

- **pipeline**: investigated soon
- **allow_trajectory_execution**: investigated soon
- **fake_execution**: investigated soon
- **max_safe_path_cost associated**: integrating sensory information (not necessary in this course)
- **jiggle_fraction**: integrating sensory information (not necessary in this course)
- **publish_monitored_planning_scene**: a parameter used by the PlanningSceneMonitor (not necessary in this course)

Contents of the package

In the Line 43 we find the planning_context.launch file.

Go to **launch/planning_context.launch**: Make sure that *robot_description* file is available. This is usually loaded and thus the default is false.

Go to **config/hrwros.srdf**: The *robot_description_semantic* parameter is located in the config folder and contains the following:

- The semantic information (robot1_base_link, vacuum_gripper1_suction_cup...)
- The robot configurations or robot poses (R1Up, R2Up)
- Virtual joints
- The entire list of link pairs that won't be checked for collision

Go to **config/joint_limits.yaml**: Joint-specific limits for each of the robot arms are updated here.

Go to **config/kinematics.yaml**: Here we have the specifications of the plugin or library we use.

Contents of the package

Go back to **launch/move_group.launch**

The next config file mentioned in the video is the planning_pipeline.launch.xml found in line 51.

It uses the pipeline argument from the begging go the file, remember that it was set by default to ompl.

Go to **launch/ompl_planning_pipeline.launch.xml**: You will find a lot of internal settings for using the ompl planning library.

Go to **config/ompl_planning.yaml**: You will find a whole list of available planners.

Go back to **launch/move_group.launch**

The next interesting file is on line 73, the trajectory_execution.launch.xml.

It uses the *moveit_manage_controllers* parameter, which takes care of trajectory execution and manages controllers, and the aforementioned *fake_execution* parameter, although we're interested on the real one.

Go to **launch/hrwros_moveit_controller_manager.launch.xml**:The video says it's empty, but in this new version, the *moveit_setup_assistant* has already filled it for us!