ROS Packages for Mover4 and Mover6 Robot Arms

Contact: www.cpr-robots.com
crm@cpr-robots.com

1. Summary

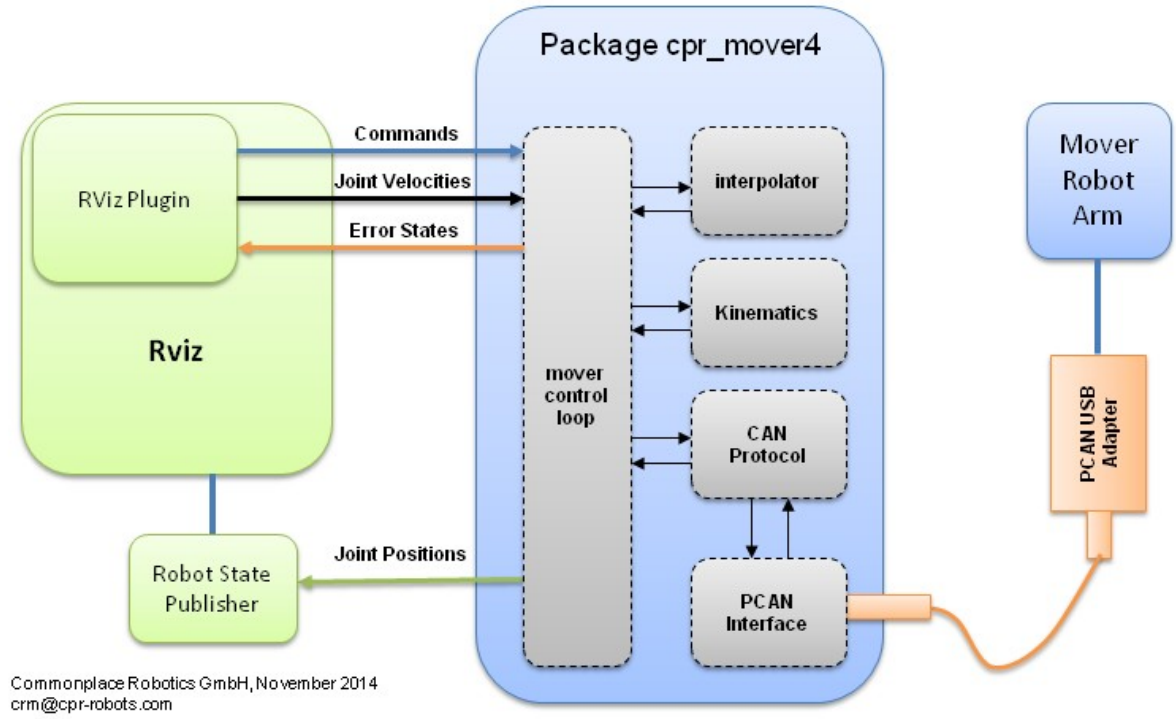
The robot package (cpr_mover) and the RViz plugin (cpr_rviz_plugin) allow to integrate the Mover4 and Mover6 robot arms into ROS environments. An integrated JointTrajectoryAction server allows to perform MoveIt generated motions.

2 Tested Environments:

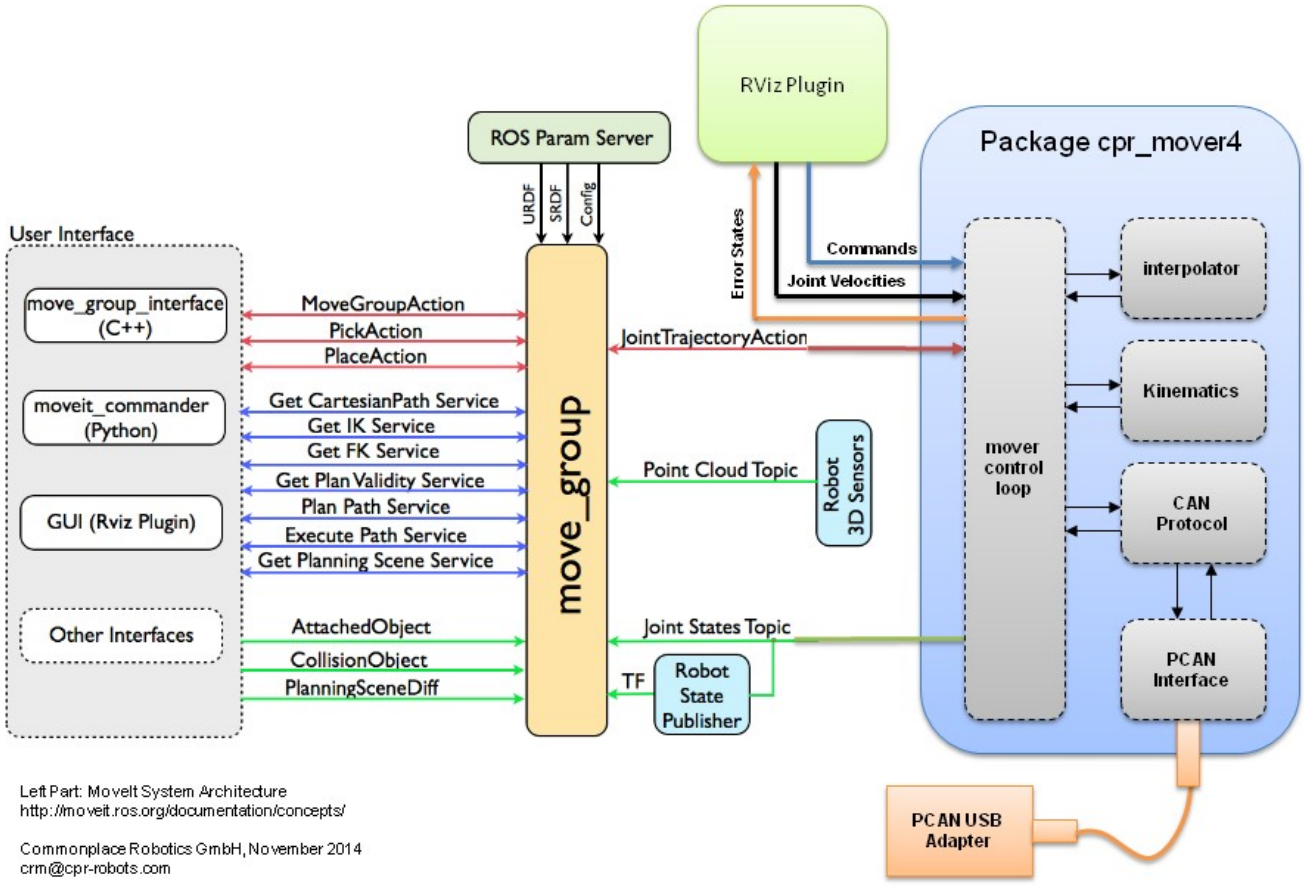
- ROS Indigo, Ubuntu 14.04 LTS

3 Architecture

The standard architecture consists of the core, RViz and the cpr_mover node.



The interaction with MoveIt is shown in the following slide.



2. Download & Installation

2 Package cpr_mover

Download at http://www.github.com/CPR-Robots/cpr_mover
Save in your catkin-src directory, e.g. ~/catkin_ws/src/cpr_mover
Compile with catkin_make

3 Plugin cpr_rviz_plugin

Download at http://www.github.com/CPR-Robots/cpr_rviz_plugin
Save in your ROS workspace, e.g. ~/ros-workspace/cpr_rviz_plugin
Compile with rosmake

4 Peak PCAN USB-Adapter

The robot arm is connected by a Peak Systeme PCAN-USB adapter. To use the adapter the according driver needs to be installed:

- Download the current driver package from Peak

- Extract driver package
- change into the directory, e.g. cd peak-linux-driver-7.10
- make clean
- We want to install the chardev version of the driver: make NET=NO_NETDEV_SUPPORT
- sudo make install
- reboot
- the command cat \proc\pcan should show:

```
*------------- PEAK-System CAN interfaces (www.peak-system.com) -------------
*------------- Release_20140121_n (7.10.0) Mar  3 2014 14:13:14 --------------
*---------------- [mod] [isa] [pci] [dng] [par] [usb] [pcc] -----------------
*--------------------- 1 interfaces @ major 249 found ----------------------
*n -type- ndev --base-- irq --btr- --read-- --write- --irqs-- -errors- status
32    usb -NA- ffffffff 255 0x001c 00000000 00000000 0000000c 00000000 0x0000
```

Especially below the ndev word there should be -NA-

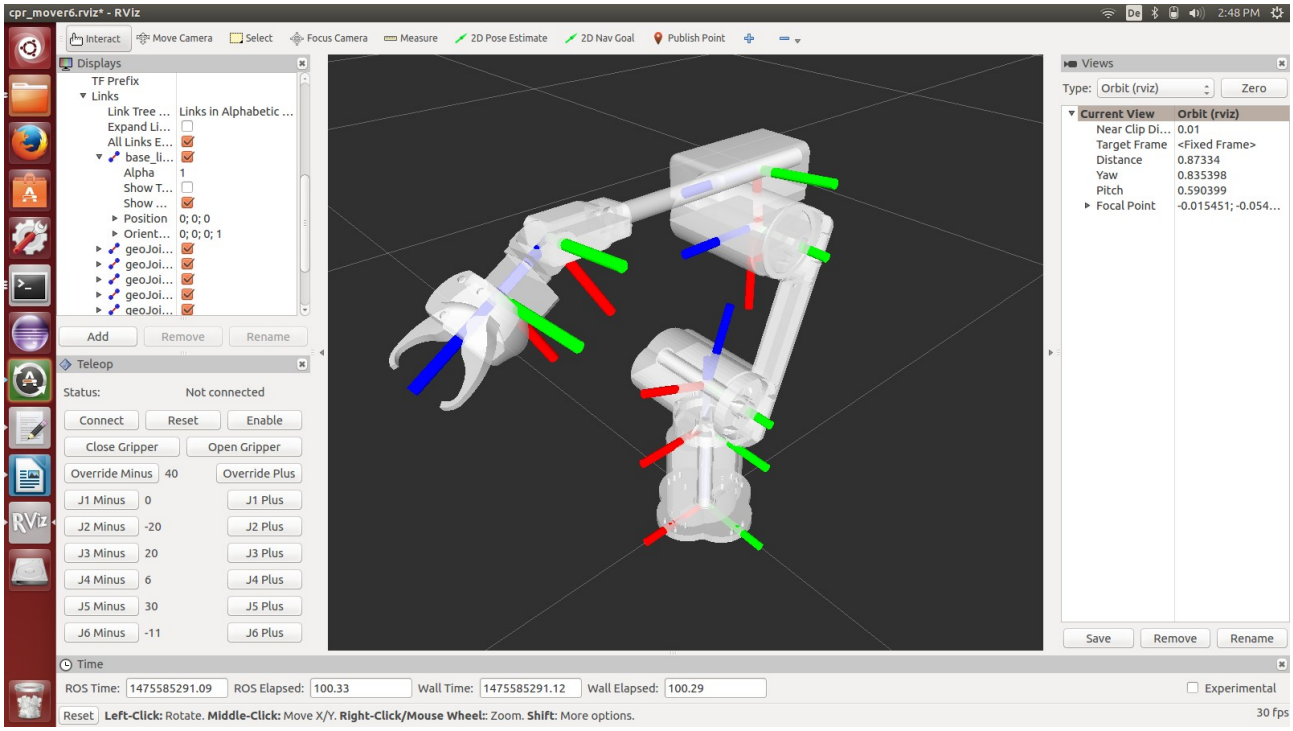
When the adapter is recognized by the OS the red LED on the adapter is on. When the adapter is connected by the software, it blinks slowly. When data is transmitted it blinks faster.

3. Start

- Start roscore
- Open a terminal and cd into the plugin directory: cd ~/ros_workspace/cpr_rviz_plugin/
- Start Rviz with the according launch file: roslaunch cpr_mover4.launch
  or roslaunch cpr_mover6.launch
- The CPR plugin should be loaded, otherwise load with the menu entry.
- Open a second terminal and cd into the cpr_mover directory: cd ~/catkin_ws/src/cpr_mover
- Start the node: rosrun cpr_mover cpr_mover
  It is important to first start Rviz, because the launch file sets the robot_type parameter to choose between Mover4 and Mover6, cpr_mover uses this parameter. If RViz is not used it can be set in another way to the parameter server.
- If the package does not start: source ./devel/setup.bash

Now move the simulated robot. If this on moves connect the real robot:

- Press “Connect” in the RViz plugin. The status changes from “not connected” to another value
- Press “Reset”: This button loads the hardware joint status, the 3D graphics adapts. Also the status changes
- Press “Enable”: Not the status changes to “0x00” or “No Error” and the robot can be moved now.



4. Robot Geometries

For the Mover6 detailed CAD files in .obj format are provided. We also provide simplified .obj files for collision testing. These files are convex and with a low polygon count to allow fast collision computation. But of course the precision is not as in the original files. If you require precise collision testing please have a look at the files and in case use the graphics files also for collision testing.
The Mover4 files are currently based on geometry primities, the update to CAD files will be done.

5. Interface Specification

Publishing Information takes place in cpr_mover::CommunicationROS() within the robot cycle time (standard: 20 Hz).

1 JointState Publisher

Type: sensor_msgs::JointState
Name: /joint_states

Provides 6 joint values + 2 gripper joint values of the robot in radian. The values are setpoint values, not the hardware values. Joint names are Joint0 to Joint5 and Gripper1, Gripper2.

The gripper is not commanded as servo joint, but using digital out by commands, see below.

2 Error Code Publisher

Type: std_msgs::String
Name: /CPRMoverErrorCodes

Provides a string with the current status of the robot arms hardware joints, the error codes.

3 Joint Velocity Subscriber

Type: sensor_msgs::JointState
Name: /CPRMoverJointVel

When there are no points to replay from the actionServer, the robot reacts to the jog values in these messages. The values in msg->velocities[] are percent values with respect to the maxJointVelocity defined in the source code. The allowed range is [-100.0 .. 100.0].

4 Commands Subscriber

Type: std_msgs::String
Name: /CPRMoverCommands

The Mover reacts to commands received with these messages. Commands are:

- Connect
- Reset
- Enable
- GripperOpen
- GripperClose
- Override ppp where ppp is the percent value of the override (integer)

5 TrajectoryActionServer

Type: actionlib::SimpleActionServer<control_msgs::FollowJointTrajectoryAction>
Name: cpr_mover/follow_joint_trajectory

6 GripperCommandActionServer

Type: actionlib::SimpleActionServer<control_msgs::GripperCommandAction>
Name: cpr_mover/gripper_command

7 Trouble Shooting

- Slow, interrupted motions of the robot: Check if there are two TeleopPanels active in RViz. If this is the case delte both and load one again.