

OCS2 优化控制工具箱（四）本体例子 汉化

前言

OCS2 包含多个机器人示例。我们在此简要讨论每个示例的主要特点。

System	State Dim.	Input Dim.	Constrained	Caching
Double Integrator	2	1	No	No
Cartpole	4	1	Yes	No
Ballbot	10	3	No	No
Quadrotor	12	4	No	No
Mobile Manipulator	6-13	6-13	Yes	Yes / No
Legged Robot	24	24	Yes	No

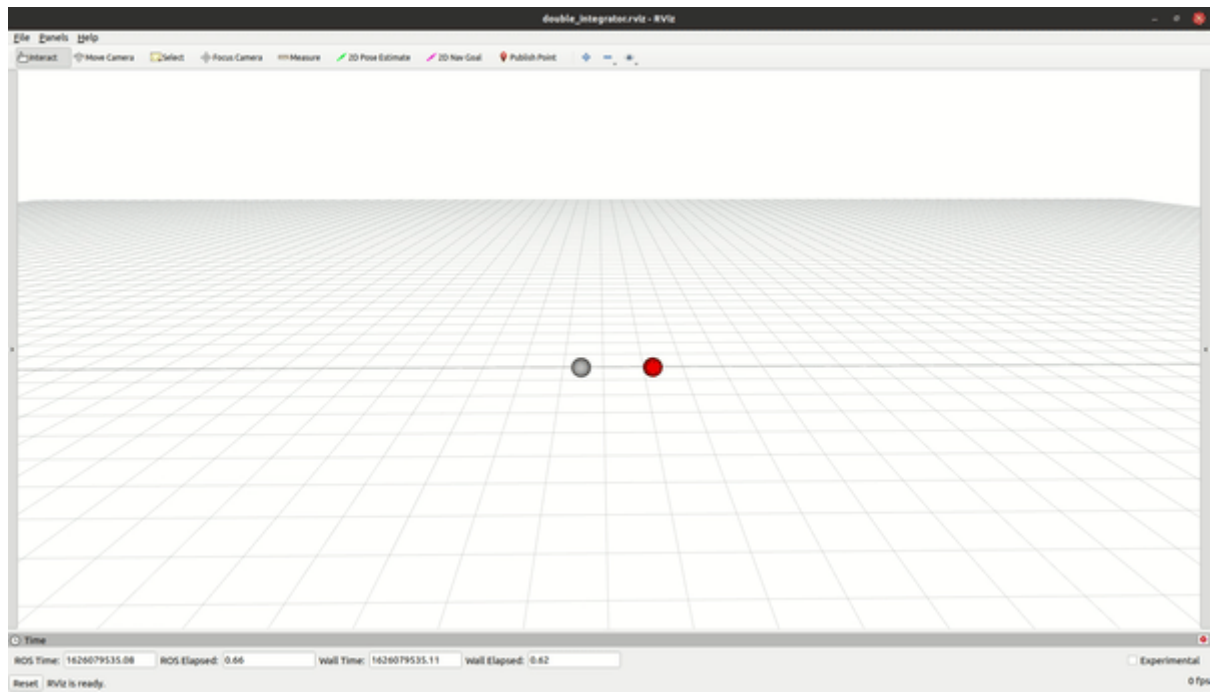
对于所有这些机器人示例，有两个独立的软件包：

- ocs2_<robot>：提供机器人专用 MPC 实现库。
- ocs2_<robot>_ros：用 ROS 封装 MPC 实现，以定义 ROS 节点。

一、Double Integrator

双积分器示例是我们最简单的问题。它模拟了一个沿 x 方向移动的一维点质量。模型是线性的，成本函数是二次函数。目标点通过参考管理器模块设置为二次成本。

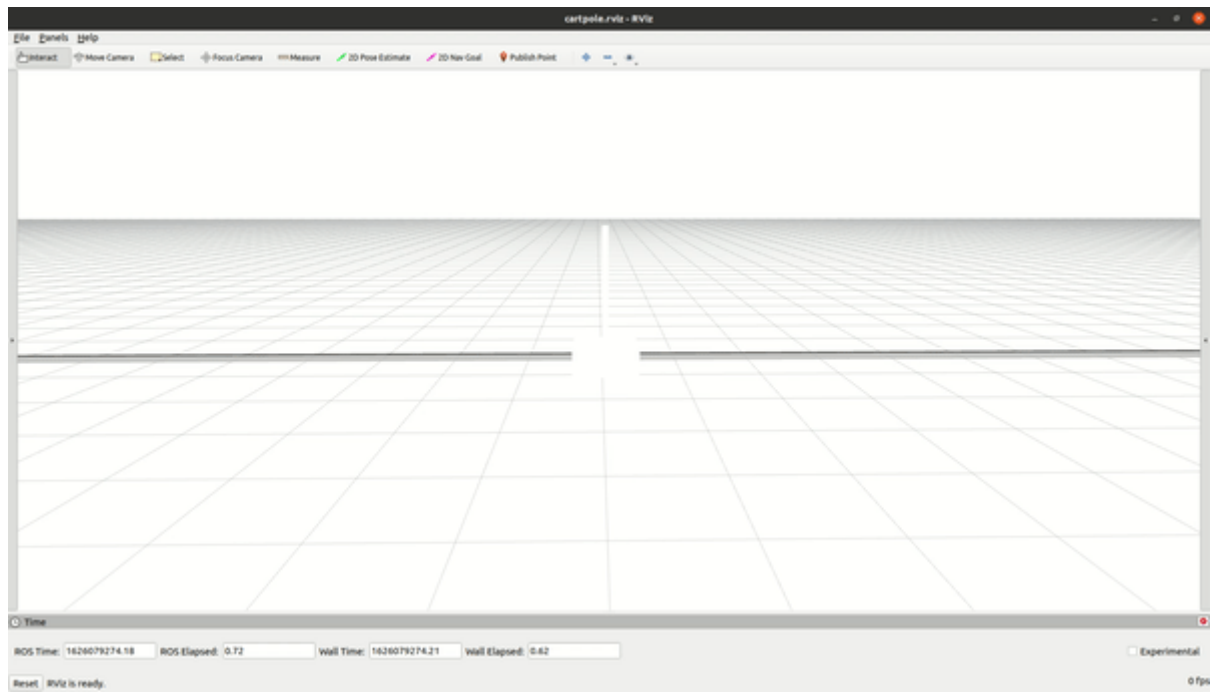
```
1 # Build the example
2 catkin build ocs2_double_integrator_ros
3 # Source workspace
4 # Do not forget to change <...> parts
5 source <directory_to_ws>/<catkin_ws_name>/devel/setup.bash
6
7 # Launch the example
8 roslaunch ocs2_double_integrator_ros double_integrator.launch
```



二、Cartpole

推车杆的例子是一个典型的控制问题，一根杆子通过一个无驱动的头连接到一辆小车上。小车沿着无摩擦轨道移动。目标是在遵守输入限制的前提下，通过沿轨道加速或减速小车，使摆锤从右下方位置开始摆动起来并保持平衡。

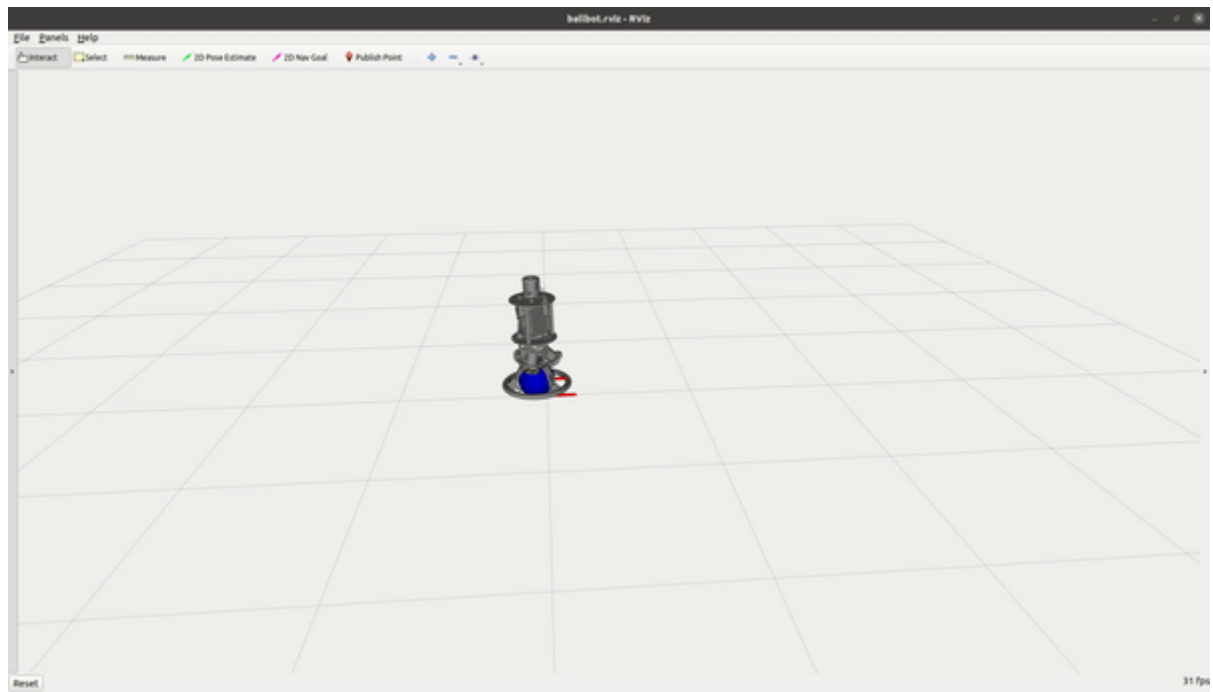
```
1 # Build the example
2 catkin build ocs2_cartpole_ros
3 # Source workspace
4 # Do not forget to change <...> parts
5 source <directory_to_ws>/<catkin_ws_name>/devel/setup.bash
6
7 # Launch the example
8 roslaunch ocs2_cartpole_ros cartpole.launch
```



三、Ballbot

Ballbot 是一个 5DoF 系统。该平台是一个扭矩控制的全向机器人，通过三个全向轮在一个球上保持平衡。该系统具有非线性动力学特性，并表现出非最小相位行为。系统动力学以 Ballbot 的前向动力学为基础，通过自动微分计算流量图的线性近似值。任务目标是根据用户指令控制机器人的 XY 位置和偏航。

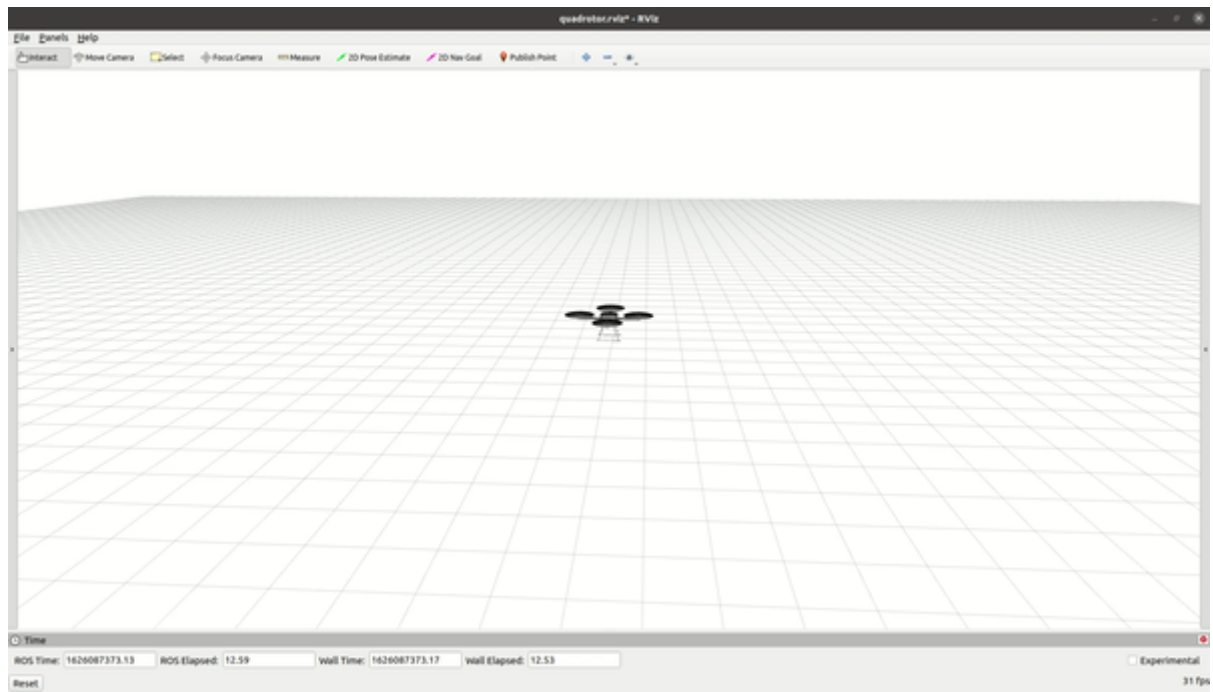
```
1 # Build the example
2 catkin build ocs2_ballbot_ros
3 # Source workspace
4 # Do not forget to change <...> parts
5 source <directory_to_ws>/<catkin_ws_name>/devel/setup.bash
6
7 # Launch the example
8 roslaunch ocs2_ballbot_ros ballbot.launch
```



四、Quadrotor

四旋翼机器人示例是一个 6DoF 系统。平台建模为浮动基座、刚体动力学、三维力矩和机器人法线方向的一维力控制。系统动力学及其导数均由代码生成。本示例旨在跟踪用户指令，该指令定义为四旋翼机器人的三维位置和偏航。

```
1 # Build the example
2 catkin build ocs2_quadrotor_ros
3 # Source workspace
4 # Do not forget to change <...> parts
5 source <directory_to_ws>/<catkin_ws_name>/devel/setup.bash
6
7 # Launch the example
8 roslaunch ocs2_quadrotor_ros quadrotor.launch
```



五、Mobile Manipulator

移动机械手示例是一个全运动学问题。模型包括一个 6DOF 机械臂和移动底座的 2D 位置和航向。控制输入为机械臂的 6 个关节速度以及底座的前进速度和旋转速度。任务目标是跟踪 6DoF 末端执行器姿势。关节位置和速度限制包含在最优控制问题的约束条件中。避免自碰撞是根据 URDF 模型中的碰撞体和避免碰撞约束来实现的（参见 [ocs2_self_collision](#)）。

注意：本例实现了 MPC 的缓存和非缓存变体，可通过配置文件中的 `usePreComputation` 标志进行选择。

系统模型通过解析 URDF 和任务文件确定。目前支持以下系统模型：

- 默认（值：0）：通过解析 URDF 获得的默认系统模型。
- 驱动假轮距（值：1）：在从 URDF 解析的模型中添加一个假 XY-Yaw 关节，该关节在整体动力学约束（速度控制）下被驱动。这是一个由 SE(2) 控制底座的移动机械手模型。
- Unactuated 假浮动基座（值：2）：在从 URDF 解析出的模型中添加一个未驱动的 XYZ-RPY 假关节。
- 激活的虚拟浮动基点（值：3）：为从 URDF 中解析出的模型添加一个完全致动（速度控制）的 XYZ-RPY 假关节。这是一个用 SE(3) 控制基座的移动机械手模型。

要玩转不同的模型类型，可以更改 [task.info](#) 文件中的模型信息。

```
1 # Build the example
2 catkin build ocs2_mobile_manipulator_ros
3 # Source workspace
4 # Do not forget to change <...> parts
5 source <directory_to_ws>/<catkin_ws_name>/devel/setup.bash
```

对于几种常见的机器人机械手，我们提供了使用 OCS2 运行它们的示例。这里提供了生成机器人模型文件（URDF）的步骤。

在其中一些示例中，我们通过 `task.info` 文件中的 `model_information.removeJoints` 属性固定了 URDF 中的某些关节（如抓手手指关节或车轮关节）。这样就可以简化所构建的 Pinocchio 模型。仅出于 rviz 可视化目的，`task.info` 文件中 `removeJoints` 属性下的关节将通过虚拟仿真节点发布在零关节位置。

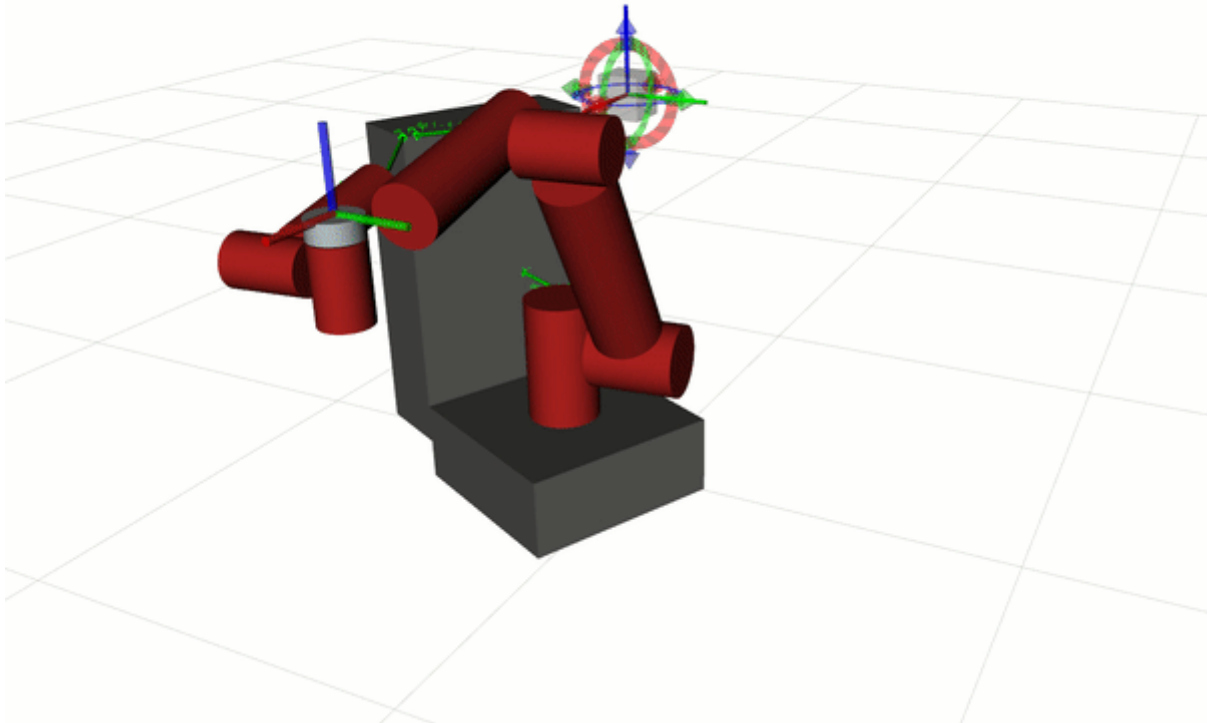


警告

对于这些示例，可能需要对 MPC 进行额外调整才能在硬件上运行。此外，对于某些示例，需要将碰撞网格简化为原始形状（如圆柱和方框），以便避免碰撞。

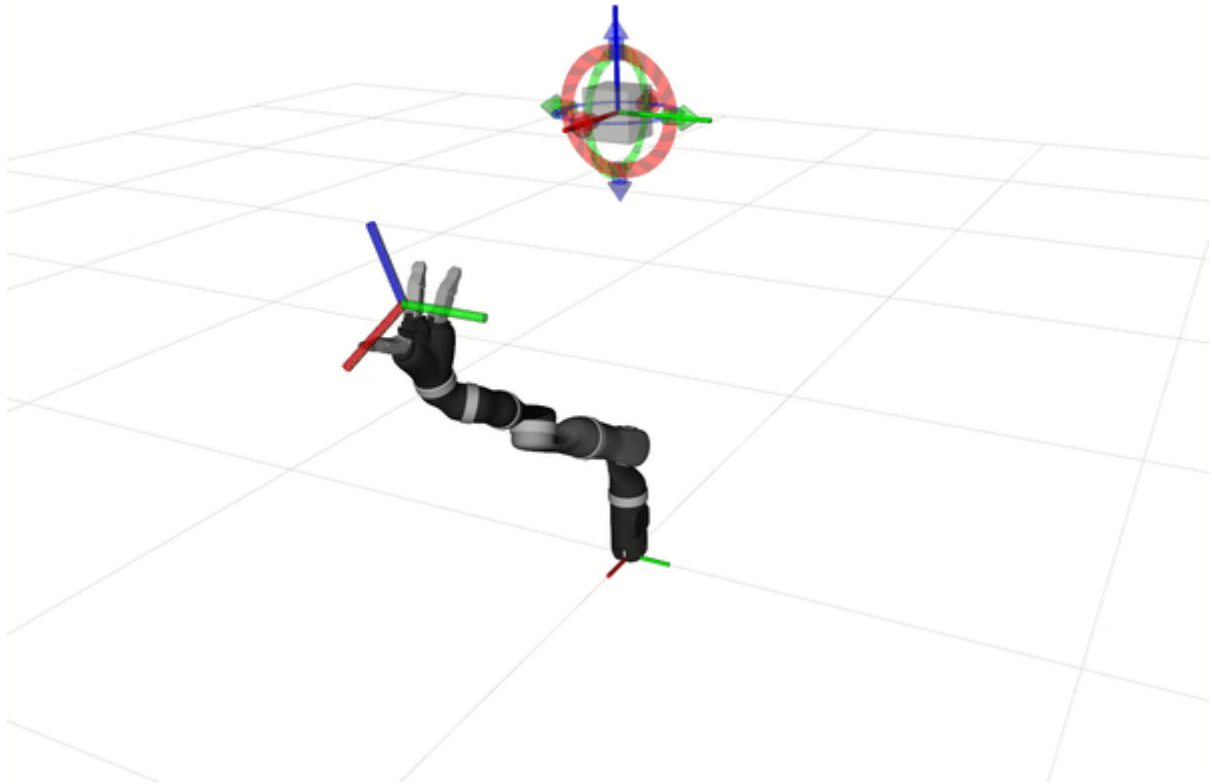
5.1 Mabi-Mobile

```
1 roslaunch ocs2_mobile_manipulator_ros manipulator_mabi_mobile.launch
```



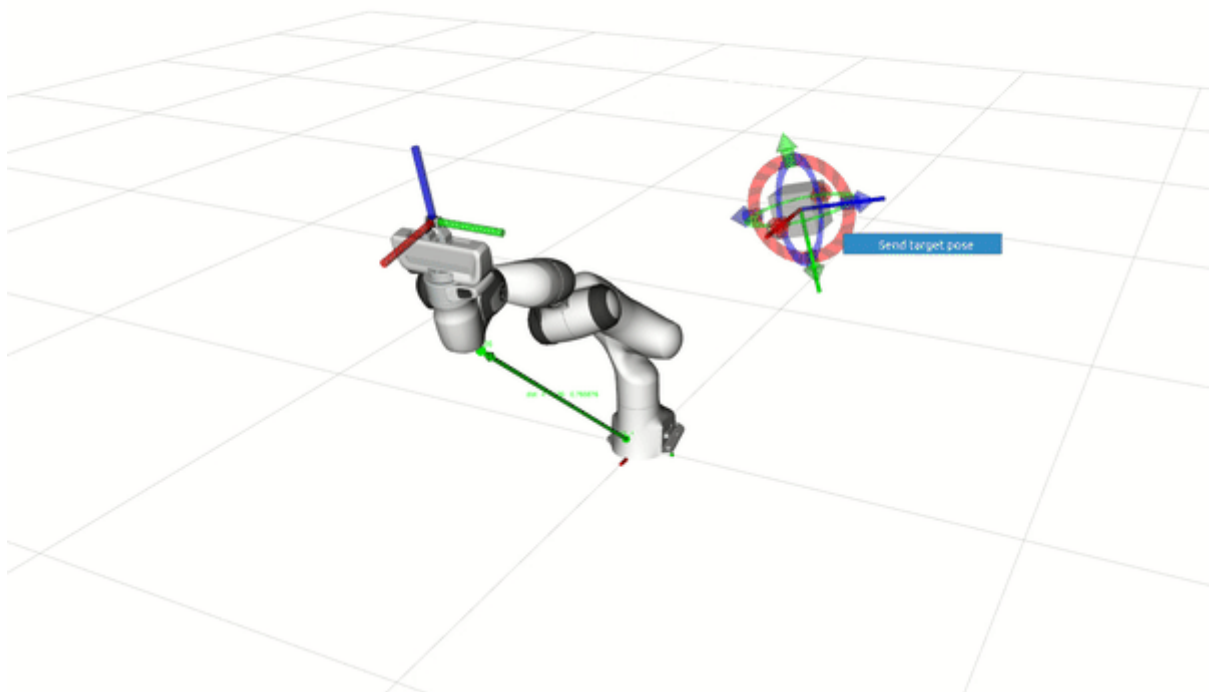
5.2 Kinova Jaco2

```
1 # For 6-Dof
2 roslaunch ocs2_mobile_manipulator_ros manipulator_kinova_j2n6.launch
3 # For 7-Dof
4 roslaunch ocs2_mobile_manipulator_ros manipulator_kinova_j2n7.launch
```



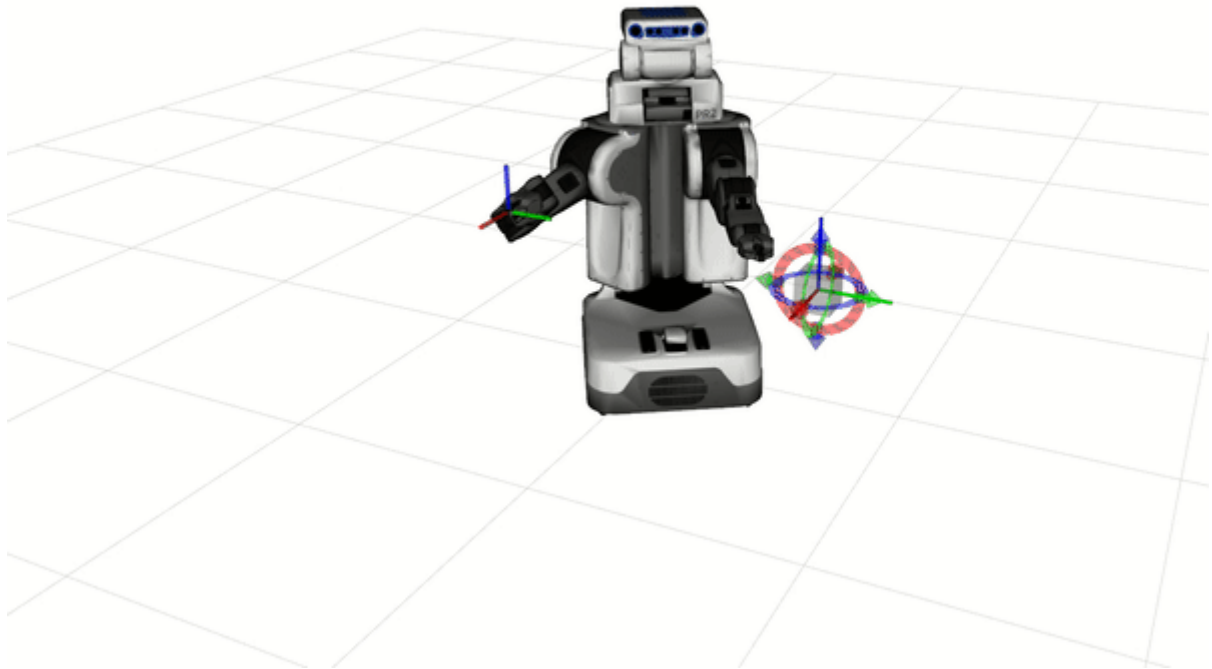
5.3 Franka Panda

```
1 roslaunch ocs2_mobile_manipulator_ros manipulator_franka.launch
```



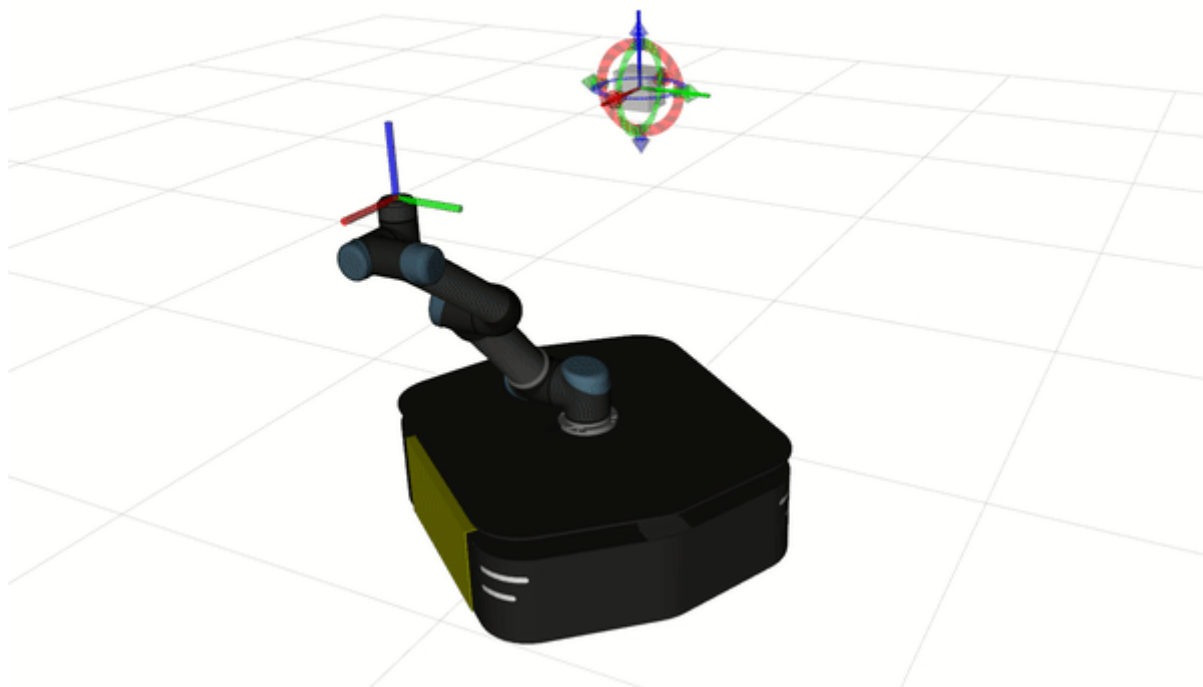
5.4 Willow Garage PR2

```
1 roslaunch ocs2_mobile_manipulator_ros manipulator_pr2.launch
```



5.5 Clearpath Ridgeback with UR-5

```
1 roslaunch ocs2_mobile_manipulator_ros manipulator_ridgeback_ur5.launch
```

六、Legged Robot

腿部机器人示例是一个开关系统问题。它采用 MPC 方法对四足机器人 Anymal 进行运动控制。机器人的步态由用户定义，并可在执行过程中通过解算器同步模块（步态接收器）进行修改。模式序列和目标轨迹通过参考管理器模块（[SwitchedModelReferenceManager](#)）进行定义。成本函数为二次罚则，用于跟踪指令的基座位置和偏航，并将机器人的重量平均分配到站立脚上。该问题有几个取决于模式的约束条件，例如摆动脚的力为零，站立脚的速度为零。在接触力上强制执行摩擦锥，为避免脚擦伤，摆动脚在 Z 方向上跟踪预定义的运动。

系统动力学有两种建模方式，可从配置文件中选择：

单刚体动力学（SRBD）：该模型假定系统具有恒定惯性，与关节位置无关。

全中心动力学（FCD）：该模型使用中心动力学，包含机器人四肢的运动。与 SRBD 相似，它考虑了机器人的全运动学。

```
1 # Build the example
2 catkin build ocs2_legged_robot_ros
3 # Source workspace
4 # Do not forget to change <...> parts
5 source <directory_to_ws>/<catkin_ws_name>/devel/setup.bash
6
7 # Launch the example for DDP
8 roslaunch ocs2_legged_robot_ros legged_robot_ddp.launch
9 # OR launch the example for SQP
10 roslaunch ocs2_legged_robot_ros legged_robot_sqp.launch
```

