Coretronic

ROS 滑動控制教學

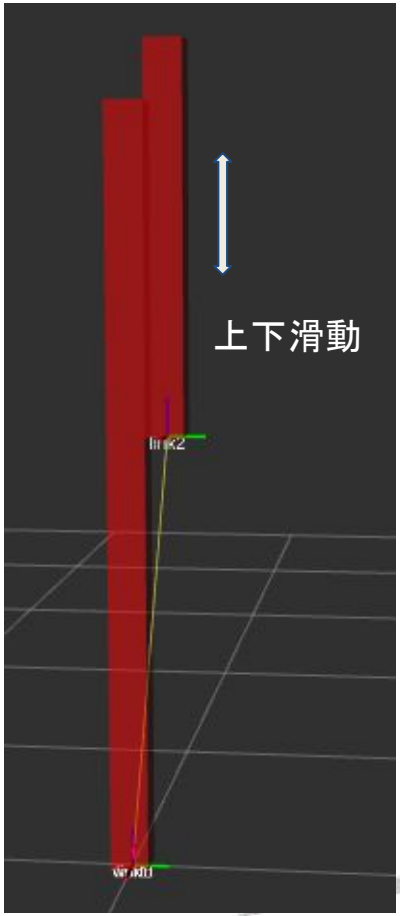
Prepared by : KC.Chang

Date : 2019/08/28

RISE

Responsibility Innovation Superiority Entrepreneurship

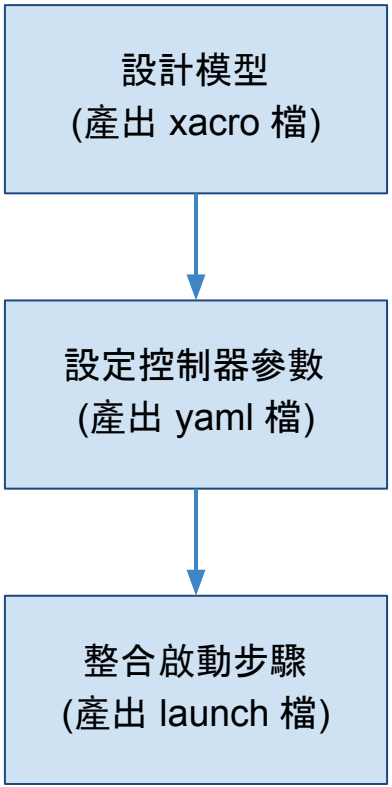
Overview



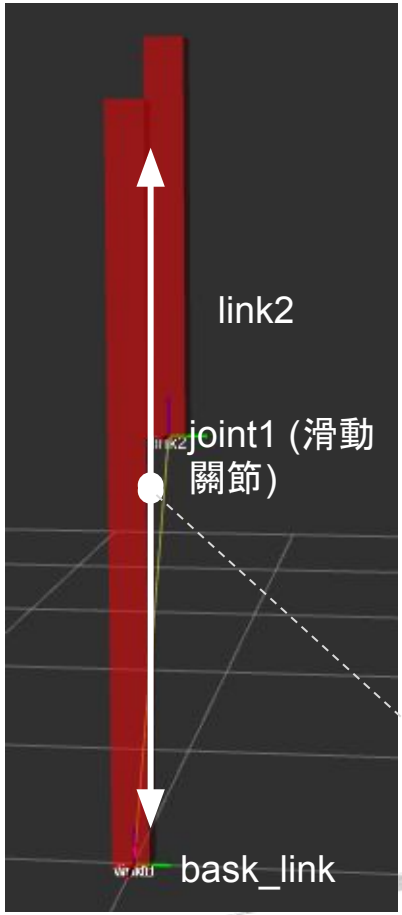
本文件參考自 rrbot 專案，並且加以簡化，做成 my_slider package，可以滑動部件。

其中可以了解如何使用 ros control 搭配 gazebo plugin 達成滑動控制。

步驟



設計模型



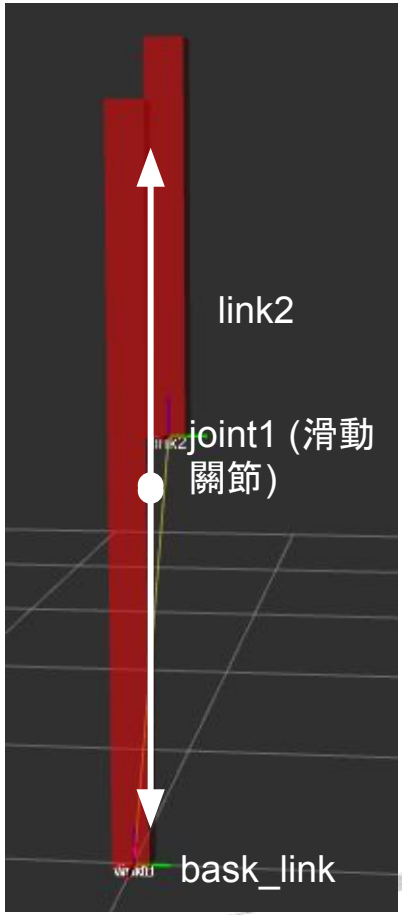

```
<link name="base_link">
  <collision>
    <geometry>
      <box size="${width} ${width} ${height1}" />
    </geometry>
    <origin xyz="0 0 ${height1/2}" rpy="0 0 0" />
  </collision>
…
  </link>
```

```
<link name="link2">
  <collision>
    <geometry>
      <box size="${width} ${width} ${height2}" />
    </geometry>
    <origin xyz="0 0 ${height2/2}" rpy="0 0 0" />
  </collision>
...
  </link>
```

```
<joint name="joint1" type="prismatic">
  <parent link="base_link" />
  <child link="link2" />
  <limit effort="100" velocity="0.1" lower="-1.0"
upper="1.0" />
  <origin xyz="0 ${width} ${height1/2}" rpy="0 0 0" />
  <axis xyz="0 0 1" />
</joint>
```

- 設為滑動關節 (prismatic)
- 最大出力 100 (kg)
- 最大滑動速率 0.1 m/s
- 滑動上、下限 +/- 1.0 m (相對於關節原點)
- 關節原點 base_link 半高處

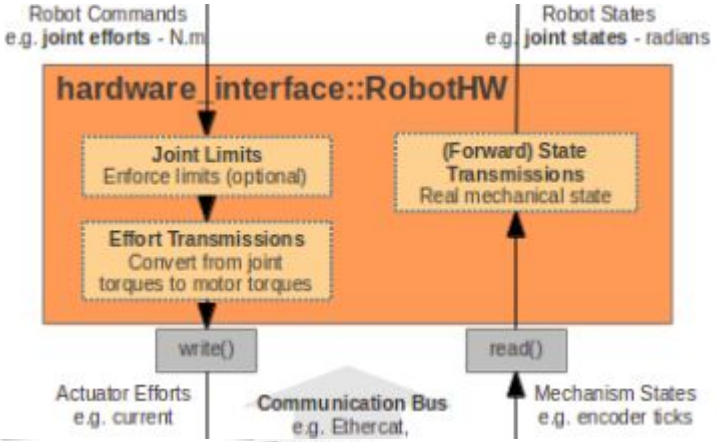
設計模型



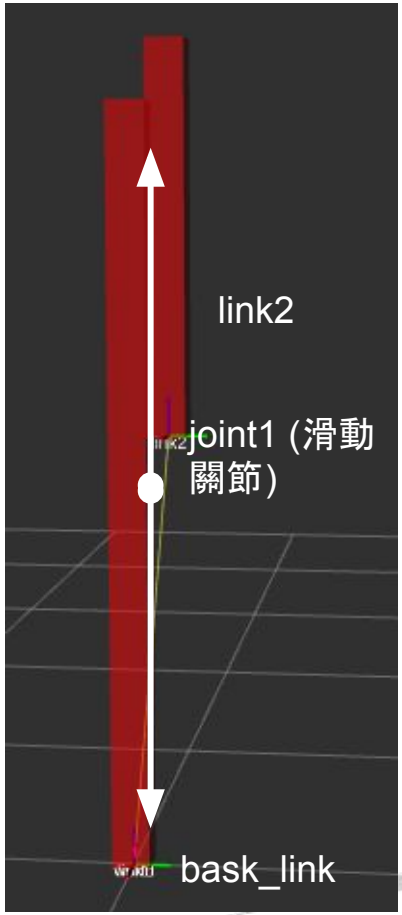

```
<transmission name="tran1">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="joint1">

<hardwareInterface>hardware_interface/EffortJointInterface</hard
wareInterface>
  </joint>
  <actuator name="motor1">

<hardwareInterface>hardware_interface/EffortJointInterface</hardw
areInterface>
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
</transmission>
```

傳動設置

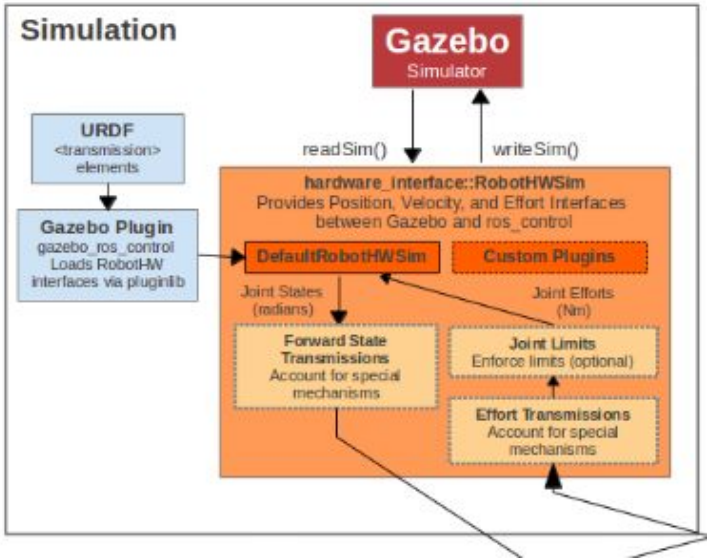
- 目前只有支援 SimpleTransmission 型態
- 發力傳動的關節 (joint1)
- 定義施力控制界面 (如左下圖，將力轉為電流控制指令，並且 write 到硬體)
- 配上驅動器 (motor1)
- 指定機械減速值 1 (非必要數值)
- <hardwareInterface>: Indigo 後版本可略，因 <joint> 已經指定此屬性



設計模型




```
<!-- ros_control plugin -->
 <gazebo>
   <plugin name="gazebo_ros_control"
filename="libgazebo_ros_control.so">
     <robotNamespace>/slider</robotNamespace>

<robotSimType>gazebo_ros_control/DefaultRobotHWSim</robotSi
mType>
     <legacyModeNS>true</legacyModeNS>     </plugin>
 </gazebo>
```

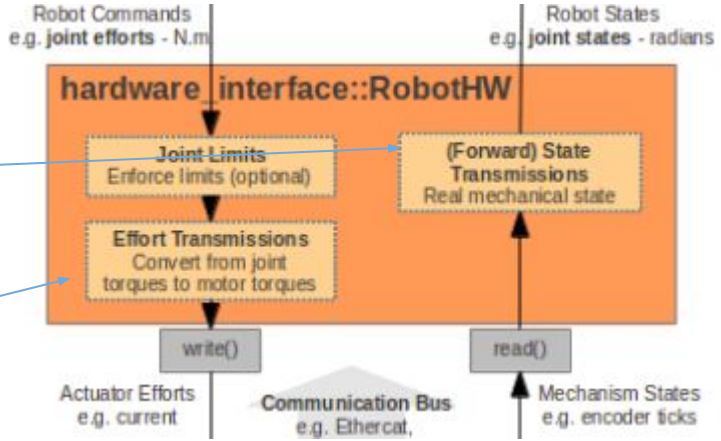


掛載模擬硬體控制界面 [2]

- 負責解析 <transmission> 定義的參數, 並且帶起對應的控制界面 (目前只支援 effort 控制界面)
- 目前 gazebo 只支援 **DefaultRobotHWSim**

設定控制器參數 (yaml)

```
slider:
  # Publish all joint states -----------------------------------
  joint_state_controller:
    type: joint_state_controller/JointStateController
    publish_rate: 50

  # Position Controllers ---------------------------------------
  joint1_position_controller:
    type: effort_controllers/JointPositionController
    joint: joint1
    pid: {p: 100.0, i: 0.01, d: 10.0}
```



整合啟動步驟 (launch)

```
<param name="robot_description"
  command="$(find xacro)/xacro --inorder '$(find my_slider)/urdf/slider.xacro'" />
```

- 以 xacro 指令讀入並且轉成 urdf 文字，指定為變數名稱 robot_description 儲存備用 [3]

```
<node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model"
respawn="false" output="screen"
  args="-urdf -model slider -param robot_description"/>
```

- 使用 spawn_model 這個 package [4]，啟用 gazebo service 來讀入 urdf 內容 (存在 robot_description 變數)，並且產生模型

```
<rosparam file="$(find my_slider)/config/slider_control.yaml" command="load"/>
```

- 將 yaml 讀入 parameter server 備用

```
 <node name="controller_spawner" pkg="controller_manager" type="spawner"
respawn="false"
        output="screen" ns="/slider" args="joint_state_controller
                                           joint1_position_controller" />
```

- 使用 spawner node 載入並且啟動多組控制器 [5]

```
<node name="robot_state_publisher" pkg="robot_state_publisher"
type="robot_state_publisher"
        respawn="false" output="screen">
  <remap from="/joint_states" to="/rrbot/joint_states" />
</node>
```

- robot_state_publisher 根據 robot_description 儲存的關節 (joint) 方位，自動計算方位關係，並且發布到 tf tree.

控制滑動：

```
$ rostopic pub /slider/joint1_position_controller/command std_msgs/Float64 "data: 1.0"
$ rostopic pub /slider/joint1_position_controller/command std_msgs/Float64 "data: 0.0"
$ rostopic pub /slider/joint1_position_controller/command std_msgs/Float64 "data: -1.0"
```

分別可以滑到上、中、下位置。

Reference:

[1]. http://wiki.ros.org/urdf/XML/Transmission

[2]. http://gazebosim.org/tutorials/?tut=ros_control

[3]. http://wiki.ros.org/roslaunch/XML/param

[4]. http://gazebosim.org/tutorials?tut=ros_roslaunch

[5]. http://wiki.ros.org/controller_manager

The end ~ ~