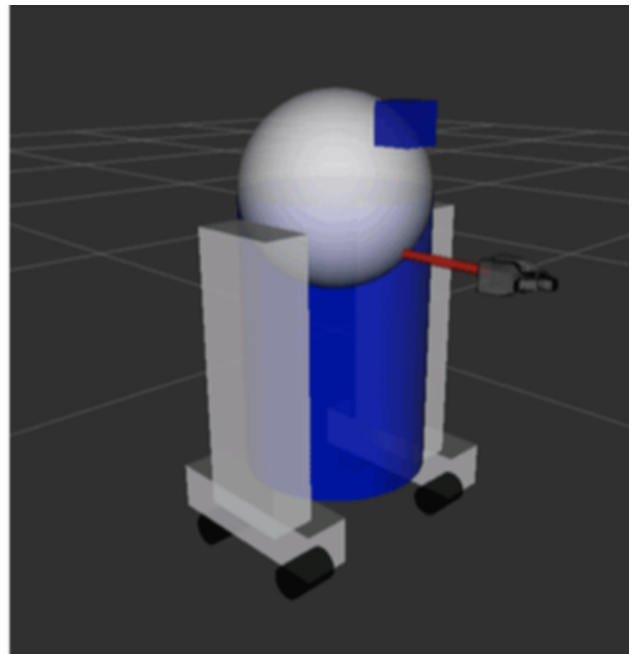


**npaka**

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

+ フォロー

ROS



ROS入門 (28) - URDFとGazeboによるロボッ トモデルのシミュレーション

♡ 7



npaka

2021年11月28日 18:03



「URDF」と「Gazebo」によるロボットモデルのシミュレーションを行う手
順をまとめました。

• Melodic

▼ 目次

前回

1. urdf_sim_tutorial

2. urdf_sim_tutorialのセットアップ
3. Gazeboの起動とURDFのロボットモデルの表示



ROSとGazeboの対話プラグインの追加

npaka

3. コントローラの追加

プログラマー。iPhone / Android / Windows / Linux / macOS / Raspberry Pi / Arduino / Unity / ROS / AI / AR / VR / RasPi / ロボット / ガジェット。年2冊ペースで技術書を執筆。アニソン / カラオケ / ギター / 猫 twitter : @npaka123

+ フォロー

前回

ROS入門 (27) - Xacroによるロボットモデルの作成

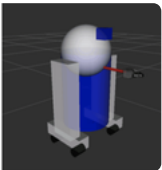
「Xacro」によるロボットモデルの作成の手順をまとめました。・ Melodic 前回 1. Xacro 「Xacro」は、「URDF」を生成するためのマクロ言語です。次の3つの機能を持っています。・ 定数・数式・マクロこれまで、URDFで...

♡ 3



npaka
2021/11/27 07:07

ROS



1. urdf_sim_tutorial

今回は、公式の「urdf_sim_tutorial」を利用します。

GitHub - ros/urdf_sim_tutorial

Contribute to ros/urdf_sim_tutorial development by creating a github.com

2. urdf_sim_tutorialのセットアップ

「urdf_sim_tutorial」のセットアップ手順は、次のとおりです。



npaka

(1) ROS1のセットアップ。

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

2)パッケージの作成と同様です。

ットアップ。

2)パッケージの作成と同様です。

+ フォロー

(3) 「~/catkin_ws/src」に「urdf_sim_tutorial」パッケージをクローン。
マウントしたPC側でクローンしました。

```
$ cd ~/catkin_ws/src
$ git clone git@github.com:ros/urdf_sim_tutorial.git
```

(4) 「~/catkin_ws」に移動してビルド。

```
$ cd ~/catkin_ws
$ catkin build
$ source ~/catkin_ws/devel/setup.bash
```



(5) 「ros-melodic-joint-state-publisher-gui」のインストール。
「rviz」で利用します。

```
$ sudo apt update
$ sudo apt install ros-melodic-joint-state-publisher-gui
```

3. Gazeboの起動とURDFのロボットモデルの表示



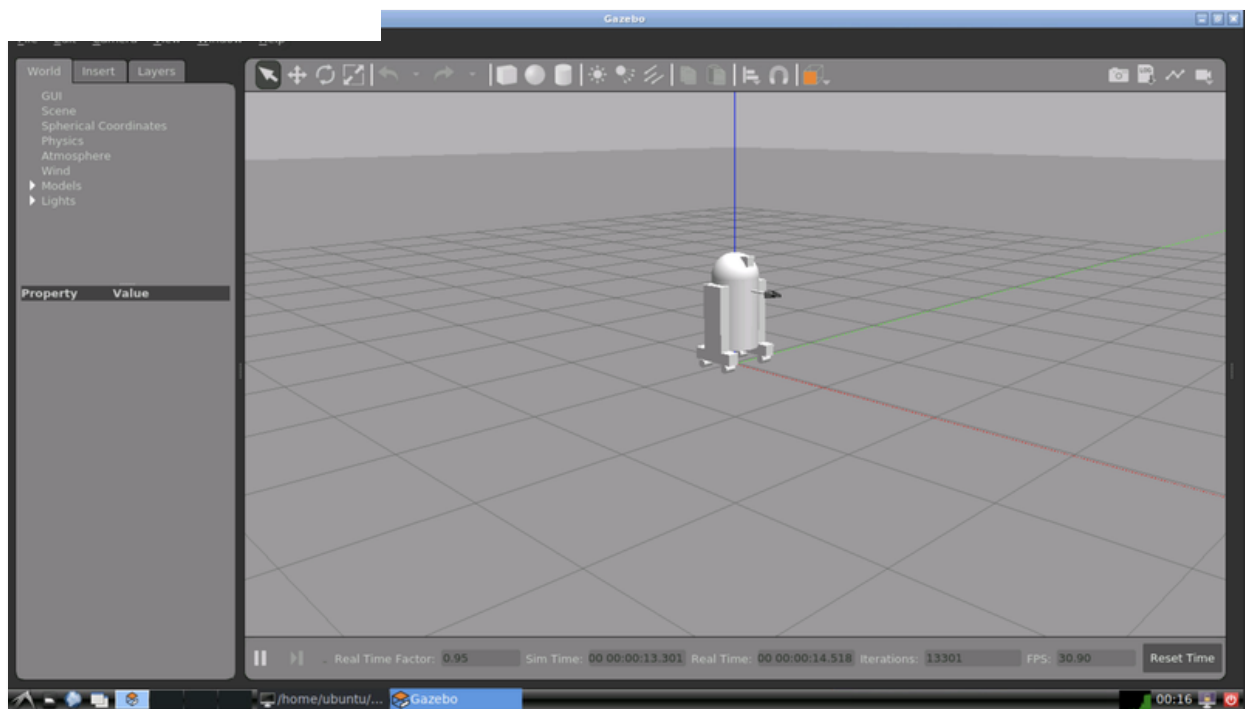
「**gazebo.launch**」を使用して、Gazeboの起動とURDFのロボットモデルの表

示します。

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

tutorial gazebo.launch

+ フォロー



「**gazebo.launch**」の中身は、次のとおりです。

```
<launch>

<!-- 引数 -->
<arg name="paused" default="false"/>
<arg name="use_sim_time" default="true"/>
<arg name="gui" default="true"/>
<arg name="headless" default="false"/>
<arg name="debug" default="false"/>
<arg name="model" default="$(find urdf_tutorial)/urdf/08-macroed.urdf.xacro"/>

<!-- Gazeboの起動 -->
<include file="$(find gazebo_ros)/launch/empty_world.launch">
```

```

<arg name="debug" value="$(arg debug)" />
<arg name="gui" value="$(arg gui)" />
<arg name="paused" value="$(arg paused)"/>
<arg name="use_sim_time" value="$(arg use_sim_time)"/>
<arg name="headless" value="$(arg headless)"/>

```



</include>
npaka

<!-- URDFをパラメータに読み込み -->

```

scription" command="$(find xacro)/xacro $(arg model)" />

```

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

Fを読み取り、Gazeboでモデルを表示 -->

```

ner" pkg="gazebo_ros" type="spawn_model"
ause -urdf -model robot -param robot_description" respawn="false"

```

isherの起動 -->

```

e_publisher" type="robot_state_publisher" name="robot_state_publi
h_frequency" type="double" value="30.0" />

```

👤+ フォロー

この状態では、ROSとGazeboはまだ対話できません。

4. ROSとGazeboの対話プラグインの追加

ROSとGazeboを対話させるには、プラグイン「gazebo_ros_control」をURDFに追加します。



(1) URDFの</ robot>タグ直前に、以下のコードを追加。

```

<gazebo>
  <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
    <robotNamespace></robotNamespace>
  </plugin>
</gazebo>

```

「~/catkin_ws/src/urdf_sim_tutorial/urdf/09-publishjoints.urdf.xacro」が追加済みのコードになります。

この状態では、ROSとGazeboはまだ対話してません。



npaka コントローラの追加

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

JSコードを「コントローラ」と呼びます。

節状態をパブリッシュする「**JointStateController**」

+ フォロー

です。

(1) コントローラの設定ファイルの準備。

・ `~/catkin_ws/src/urdf_sim_tutorial/config/joints.yaml`

```
type: "joint_state_controller/JointStateController"
publish_rate: 50
```

(2) Gazeboとrvizの起動し、JointStateControllerを実行し、トピック

「`/joint_states`」をパブリッシュ。

```
$ roslaunch urdf_sim_tutorial 09-joints.launch
```

「**09-joints.launch**」の中身は、次のとおりです。

```
<launch>
  <!-- 引数 -->
  <arg name="model" default="$(find urdf_sim_tutorial)/urdf/09-publishjoints.urdf.xacro"/>
  <arg name="rvizconfig" default="$(find urdf_tutorial)/rviz/urdf.rviz" />
```

```

<!-- Gazeboの起動 -->
<include file="$(find urdf_sim_tutorial)/launch/gazebo.launch">
  <arg name="model" value="$(arg model)" />
</include>
<!-- rvizの起動 -->
<node name="rviz" pkg="rviz" type="rviz" args="-d $(arg rvizconfig)" />

```

**npaka**

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

フォロー

```

  llerをパラメータに読み込み -->
  ad"
  _sim_tutorial)/config/joints.yaml"
  ate_controller" />

  llerの実行 -->
  roller_spawner" pkg="controller_manager" type="spawner"
  state_controller --shutdown-timeout 3"/>

```

(3) トピック「**/joint_states**」が公開されていることを確認。

トピックは公開されていますが、情報 (name、position、velocity、effort)



```

$ rostopic echo /joint_states
header:
  seq: 1809
  stamp:
    secs: 36
    nsecs: 550000000
  frame_id: ''
name: []
position: []
velocity: []
effort: []
---
```

6. トランスミッションの追加

Gazeboの可動関節を制御するには、「トランスミッション」を使います。今回は、頭の関節にトランスミッションを追加します。

追加手順は、次のとおりです。



URDFに以下のコードを追加。

npaka

このコードは基本的に定型文です。<joint>の名前は宣言済みの関節名を指定

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

+ フォロー

```
ead_swivel_trans">
interface/SimpleTransmission</type>
ad_swivel_motor">
ion>1</mechanicalReduction>
```

```
<joint name="head_swivel">
  <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
</joint>
</transmission>
```



「~/catkin_ws/src/urdf_sim_tutorial/urdf/firsttransmission.urdf.xacro」
が追加済みのコードになります。

(2) Gazeboとrvizを実行。

トランスミッションが追加されたため、トピック「/joint_state」の情報
(name、position、velocity、effort) が公開されます。

```
$ roslaunch urdf_sim_tutorial 09-joints.launch model:=urdf/10-firsttransmission.urdf.xacr
```



(3) 「rostopic」でトピック/joint_stateの確認。

```
$ rostopic echo /joint_states
header:
```



```
seq: 1386
stamp:
  secs: 27
  nsecs: 913000000
frame_id: ''
name: [head_swivel]
position: [-4.208654580395432e-08]
velocity: [9.119072519434175e-08]
```



npaka

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

制御

フォロー

頭のリジッドを制御する「**JointPositionController**」を追加します。

追加手順は、次のとおりです。

(1) コントローラの設定ファイルの準備。

- `~/catkin_ws/src/urdf_sim_tutorial/config/head.yaml`

```
type: "position_controllers/JointPositionController"
joint: head_swivel
```

(2) Gazeboとrvizを起動し、JointPositionControllerを実行し、トピック「`/r2d2_head_controller/command`」の値をサブスクライブ。

```
$ roslaunch urdf_sim_tutorial 10-head.launch
```

(3) トピック「`/r2d2_head_controller/command`」に値をパブリッシュ。
パブリッシュをするとすぐに頭の向きが変更されます。

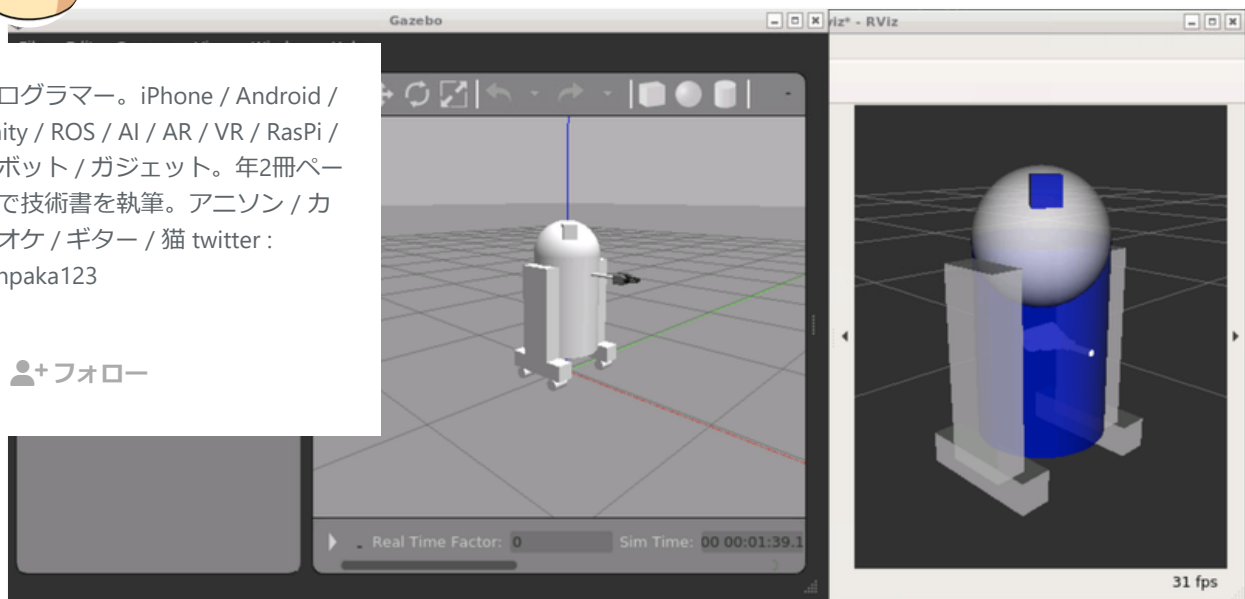
```
$ rostopic pub /r2d2_head_controller/command std_msgs/Float64 "data: -0.707"
```



npaka

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

+ フォロー



(4) URDFを以下のように変更。

パブリッシュをすると徐々に頭の向きが変更されるようになります。

```
<joint name="head_swivel" type="continuous">
  <parent link="base_link"/>
  <child link="head"/>
  <axis xyz="0 0 1"/>
  <origin xyz="0 0 ${bodylen/2}"/>
  <limit effort="30" velocity="1.0"/>
</joint>
```

「~/catkin_ws/src/urdf_sim_tutorial/urdf/11-
limittransmission.urdf.xacro」が変更済みのコードになります。

(5) Gazeboとrvizを起動し、JointPositionControllerを実行。

```
$ roslaunch urdf_sim_tutorial 10-head.launch model:=urdf/11-limittransmission.urdf.xacro
```



npaka

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

ad_controller/command」に値をパブリッシュ。
されることを確認します。

フォロー

```
ad_controller/command std_msgs/Float64 "data: -0.707"
```

8. グリッパーの制御

グリッパーの各関節をグループ化して制御する

「**JointGroupPositionController**」を追加します。

追加手順は、次のとおりです。

(1) コントローラの設定ファイルの準備。

- `~/catkin_ws/src/urdf_sim_tutorial/config/gripper.yaml`

```
type: "position_controllers/JointGroupPositionController"
joints:
  - gripper_extension
  - left_gripper_joint
  - right_gripper_joint
```

(2) Gazeboとrvizの起動し、JointGroupPositionControllerを実行し、トピック
「/r2d2_gripper_controller/command」の値をサブスクライブ。

```
$ roslaunch urdf_sim_tutorial 12-gripper.launch
```



npaka

ロボットピック「r2d2_gripper_controller/command」に値をパブリッシュ。

プログラマー。iPhone / Android / Unity / ROS / AI / AR / VR / RasPi / ロボット / ガジェット。年2冊ペースで技術書を執筆。アニソン / カラオケ / ギター / 猫 twitter : @npaka123

```
gripper_controller/command std_msgs/Float64MultiArray "layout:
```

 フォロー

```
    stride: 1
    data_offset: 0
    data: [0, 0.5, 0.5]"
```

```
$ rostopic pub /r2d2_gripper_controller/command std_msgs/Float64MultiArray "layout:
dim:
- label: ''
  size: 3
  stride: 1
  data_offset: 0
  data: [-0.4, 0, 0]"
```

9. ホイールの制御

ロボットのホイールを制御する「DiffDriveController」を追加します。

追加手順は、次のとおりです。

(1) ホイールのトランスミッションを追加。

**npaka**

```
<transmission name="${prefix}_${suffix}_wheel_trans">
  <type>transmission_interface/SimpleTransmission</type>
  <actuator name="${prefix}_${suffix}_wheel_motor">
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
  <joint name="${prefix}_${suffix}_wheel_joint">
    <hardwareInterface/VelosityJointInterface</hardwareInterface>
```

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

フォロー

ルを追加。

```
<gazebo reference="${prefix}_${suffix}_wheel">
  <mu1 value="200.0"/>
  <mu2 value="100.0"/>
  <kp value="10000000.0" />
  <kd value="1.0" />
  <material>Gazebo/Grey</material>
</gazebo>
```

(3) コントローラの設定ファイルの準備。

- ~/catkin_ws/src/urdf_sim_tutorial/config/diffdrive.yaml

```
type: "diff_drive_controller/DiffDriveController"
publish_rate: 50

left_wheel: ['left_front_wheel_joint', 'left_back_wheel_joint']
right_wheel: ['right_front_wheel_joint', 'right_back_wheel_joint']

wheel_separation: 0.44

# ロボットのエンコーダ出力のオドメトリ共分散
pose_covariance_diagonal: [0.001, 0.001, 0.001, 0.001, 0.001, 0.03]
twist_covariance_diagonal: [0.001, 0.001, 0.001, 0.001, 0.001, 0.03]

# robot_descriptionのトップレベルフレーム(リンク)
base_frame_id: base_link

# ロボットの速度と加速度の制限
linear:
```

```
x:
  has_velocity_limits : true
  max_velocity       : 0.2 # m/s
  has_acceleration_limits: true
  max_acceleration   : 0.6 # m/s^2
```



angular:
npaka

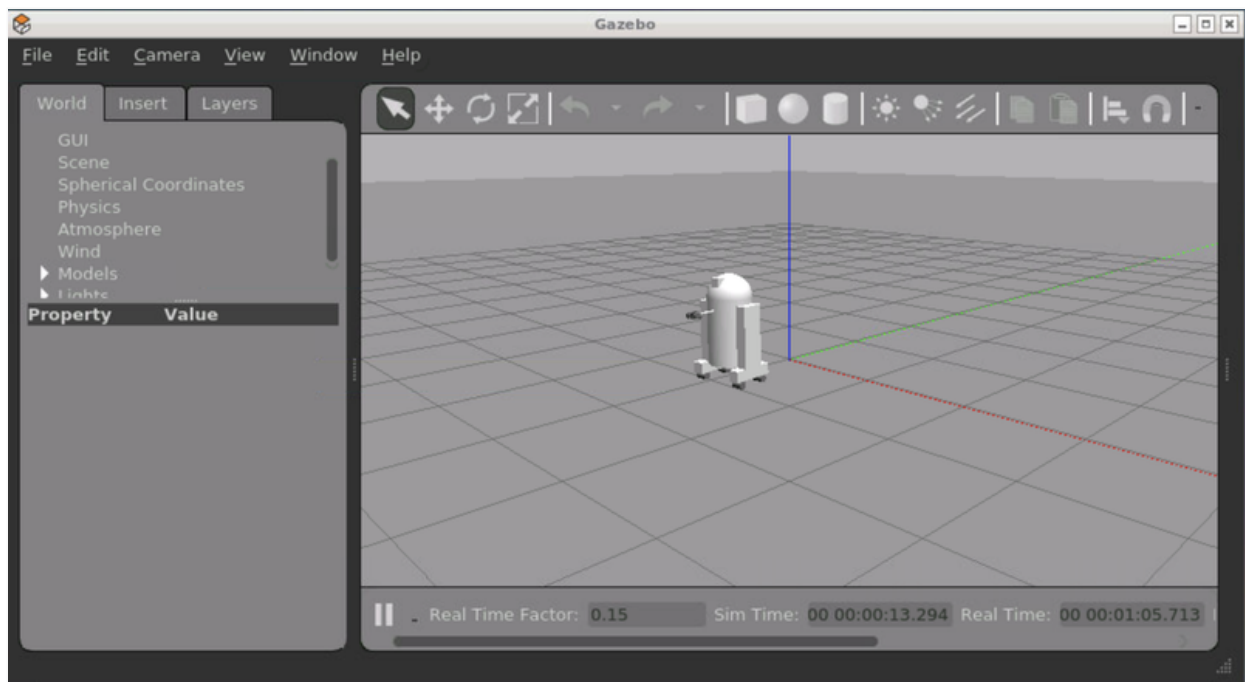
```
z:
  has_velocity_limits : true
  max_velocity       : 2.0 # rad/s
  has_acceleration_limits: true
  max_acceleration   : 6.0 # rad/s^2
```

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

+ フォロー

o


```
$ roslaunch urdf_sim_tutorial 13-diffdrive.launch
```



(5) rqt_robot_steeringの起動。

このGUIでR2D2のホイールを制御できます。

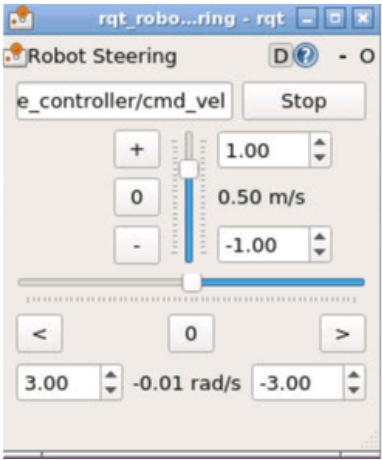
```
$ rosrn rqt_robot_steering rqt_robot_steering
```



npaka

プログラマー。iPhone / Android / Unity / ROS / AI / AR / VR / RasPi / ロボット / ガジェット。年2冊ペースで技術書を執筆。アニソン / カラオケ / ギター / 猫 twitter : @npaka123

+ フォロー



10. 参考

<div><div>urdf/Tutorials/Using a URDF in Gazebo - ROS Wiki</div><div>wiki.ros.org</div></div>	
<div><div>Unityではじめる ROS・人工知能 ロボットプログラミング実践入門</div><div>www.amazon.co.jp</div><div>4,950円 (2023年05月26日 14:09時点 詳しくはこちら).</div><div>Amazon.co.jpで購入する</div></div>	

**npaka**

プログラマー。iPhone / Android /
Unity / ROS / AI / AR / VR / RasPi /
ロボット / ガジェット。年2冊ペー
スで技術書を執筆。アニソン / カ
ラオケ / ギター / 猫 twitter :
@npaka123

 + フォロー