# Gazebo and ROS - part 1

Kjartan Halvorsen

February 18, 2022

# Sources

- http://gazebosim.org/
- http://gazebosim.org/tutorials/?tut=ros_comm
- http://sdformat.org/
- http://docs.ros.org/kinetic/api/gazebo_msgs/html/index-msg.html

# What is Gazebo?

A 3D simulation and visualization environment containing a physics engine.

# Why use Gazebo?

It is a playground for your robot where you can test actuators, sensors and control algorithms.

# Let's start with a fun example

Start `roscore`, `gazebo` and `gazebo_ros` in two separate terminals.

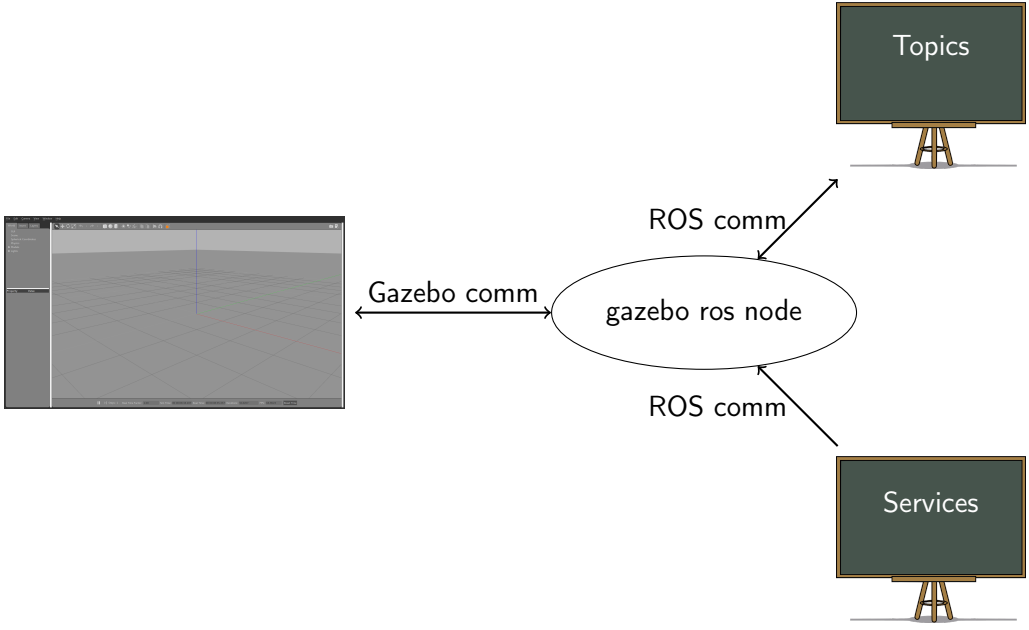
Terminal 1

```
~$ roscore
```

Terminal 2

```
~$ rosrun gazebo_ros gazebo
```

# What just happened



# Add some junk to the world

```
~$ rosrun gazebo_ros spawn_model -database coke_can \
> -sdf -model coke_can1 -y 1 -x 0
~$ rosrun gazebo_ros spawn_model -database coke_can \
> -sdf -model coke_can2 -y 2 -x 0
```

# See the physics engine in action

Pause the gazebo physics engine, then lift the first can up 50cm

```
~$ rosservice call /gazebo/pause_physics
~$ rosservice call /gazebo/set_model_state \
> '{model_state: { model_name: coke_can1, \
> pose: { position: { x: 0., y: 1. ,z: 0.5 } }, \
> reference_frame: world } }'
```

and drop the can (start simulation in the gazebo gui)!

# On your own: Drop the first can on top of the second can from two meters

# Gazebo published topics

Which of the following topics are not available?

`/clock`
`/gazebo/link_states`
`/gazebo/model_states`
`/gazebo/parameter_descriptions`
`/gazebo/parameter_updates`
`/gazebo/set_link_state`
`/gazebo/set_model_state`
`/gazebo/set_world_state`
`/rosout`
`/rosout_agg`

# Gazebo services provided

Which of the following services are not available?

`/gazebo/apply_body_wrench`
`/gazebo/clear_joint_forces`
`/gazebo/get_joint_properties`
`/gazebo/set_joint_force`

Hint: `rosservice`

# Getting the coke can airborn

We can apply a wrench ( a force and torque pair) to any rigid body in the gazebo world

```
~$ rosservice info /gazebo/apply_body_wrench
```

What arguments does the service take?

# Getting the coke can airborn, contd

If the can has a vertical launch velocity of $\mathrm{v_L\ m/s}$, it will reach a height where its potential energy is the same as the kinetic energy at launch. So to reach a height of $h = 3\ \mathrm{m}$ we need a launch velocity of

$$v_L =$$

# Getting the coke can airborn, contd

If the can has a vertical launch velocity of $\mathrm{v_L}\ \mathrm{m/s}$, it will reach a height where its potential energy is the same as the kinetic energy at launch. So, to reach a height of $h = 3\ \mathrm{m}$ we need a launch velocity of

$$v_L = \sqrt{2gh} \approx 7.7\ \mathrm{m/s}$$

# Kicking the coke can, theory

Let's apply a constant, large force $F = F_v + mg$ under a short time $\tau = 10\ \mathrm{ms}$. Newton's second law gives

$$\frac{d}{dt}(mv) = F_v + mg - mg$$

$$\int_0^\tau (\frac{d}{dt}mv)dt = \int_0^\tau F_v dt$$

# Kicking the coke can, theory

Let's apply a constant, large force $F = F_v + mg$ under a short time $\tau = 10\ \mathrm{ms}$. Newton's second law gives

$$\frac{d}{dt}(mv) = F_v + mg - mg$$

$$\int_0^\tau (\frac{d}{dt}mv)dt = \int_0^\tau F_v dt$$

$$mv_L - 0 = F_v\tau$$

The coke can has a mass of $m = 0.39\ \mathrm{kg}$. What force $F_v$ is needed?

# Kicking the coke can, for real (sort of)

```
~$ rosservice call /gazebo/pause_physics
~$ rosservice call /gazebo/apply_body_wrench \
> '{body_name: "coke_can2::link" , \
> wrench: { force: { z: 300 } }, duration: 10000000 }'
```

# URDF - Unified Robotic Description Format

Defining a robot

Video

# Spawning a model defined by a urdf file

```
~$ rosrun gazebo_ros spawn_model -file ./furuta.urd -urdf -\
>    y 2 -model furuta
```

# Making the model move

```
~$ rosservice call /gazebo/apply_joint_effort \
"joint_name: 'furuta::base_to_one_prox' effort: 10 duration: 10000000"
```