Introdução ao Gazebo

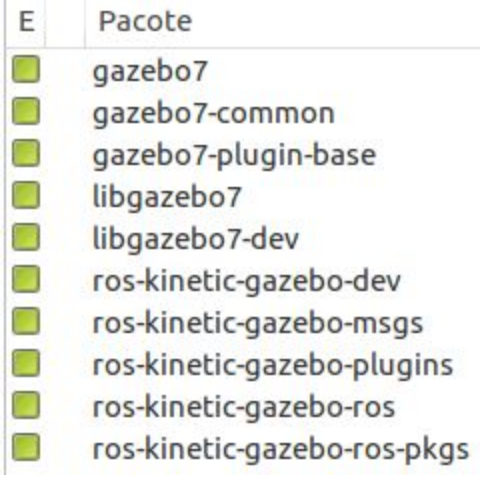
Guilherme Marins Maciel
André Marcato

Instalação

- As instalações completas do ROS acompanham o Gazebo

```
sudo apt-get install ros-<version>-desktop-full
```

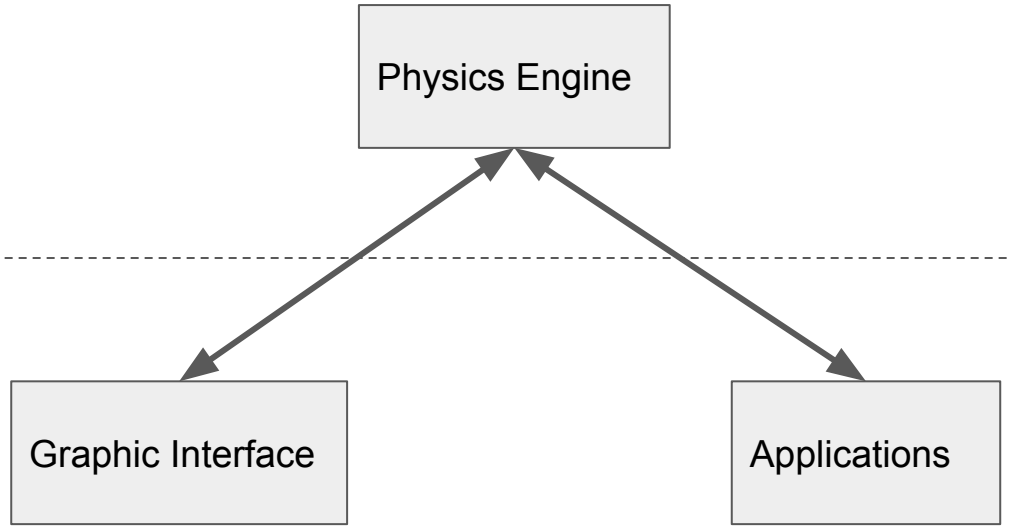
- Opção: Instalar o gerenciador de pacotes synaptic e verificar/instalar os seguintes pacotes:



Para o ubuntu 18.04:
gazebo9 e ros-melodic

Arquitetura

Server



Clients

Elementos da Simulação

World

- Collection of models, lights, plugins and global properties

Models

- Collection of links, joints, sensors, and plugins

Links

- Collection of collision and visual objects

Collision Objects

- Geometry that defines a colliding surface

Visual Objects

- Geometry that defines visual representation

Joints

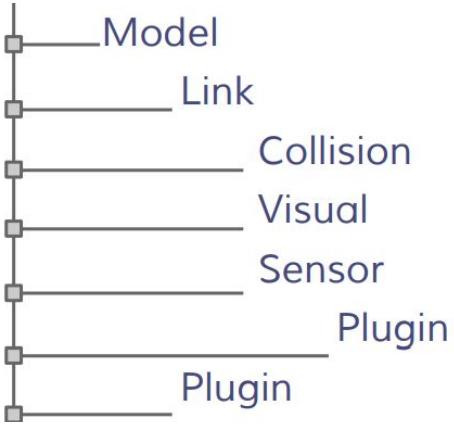
- Constraints between links

Sensors

- Collect, process, and output data

Plugins

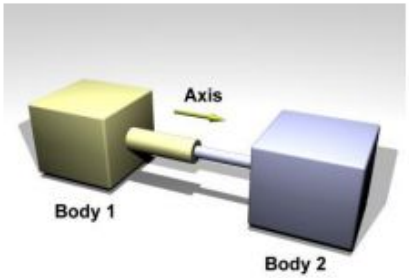
- Code attached to a World, Model, Sensor, or the simulator itself



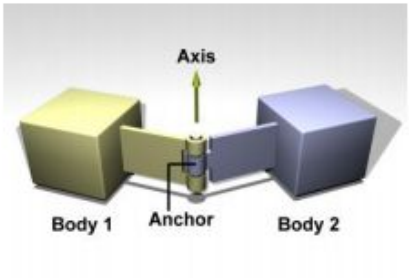
Links vs Joints

- **Collision and Visual Geometries**
  Simple shapes: sphere, cylinder, box, plane
  Complex shapes: heightmaps, meshes

- **Joints**
  Fixed: 0 DOF; Prismatic: 1 DOF translational ; Revolute: 1 DOF rotational ; Revolute2: Two revolute joints in series ; Ball: 3 DOF rotational ; Universal: 2 DOF rotational ; Screw: 1 DOF translational, 1 DOF rotational



Prismatic

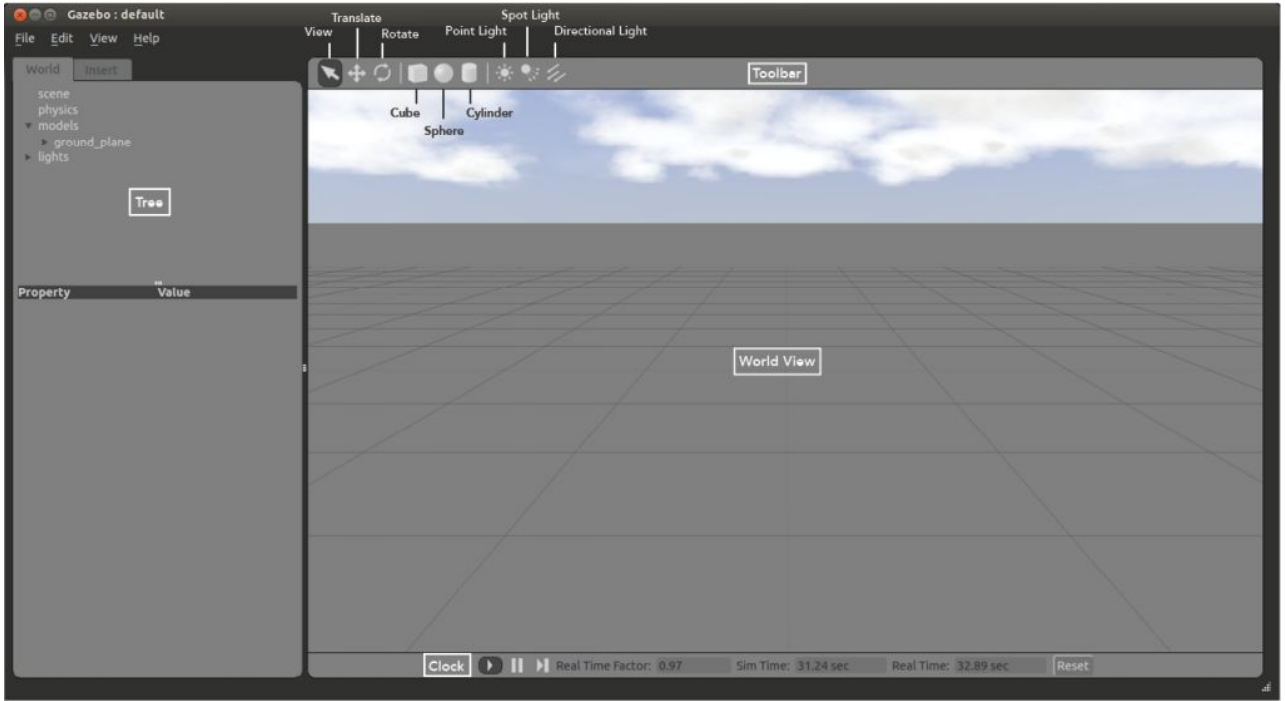


Revolute

Testando a Interface gráfica

- Para testar o gazebo funcionando vinculado ao ROS:

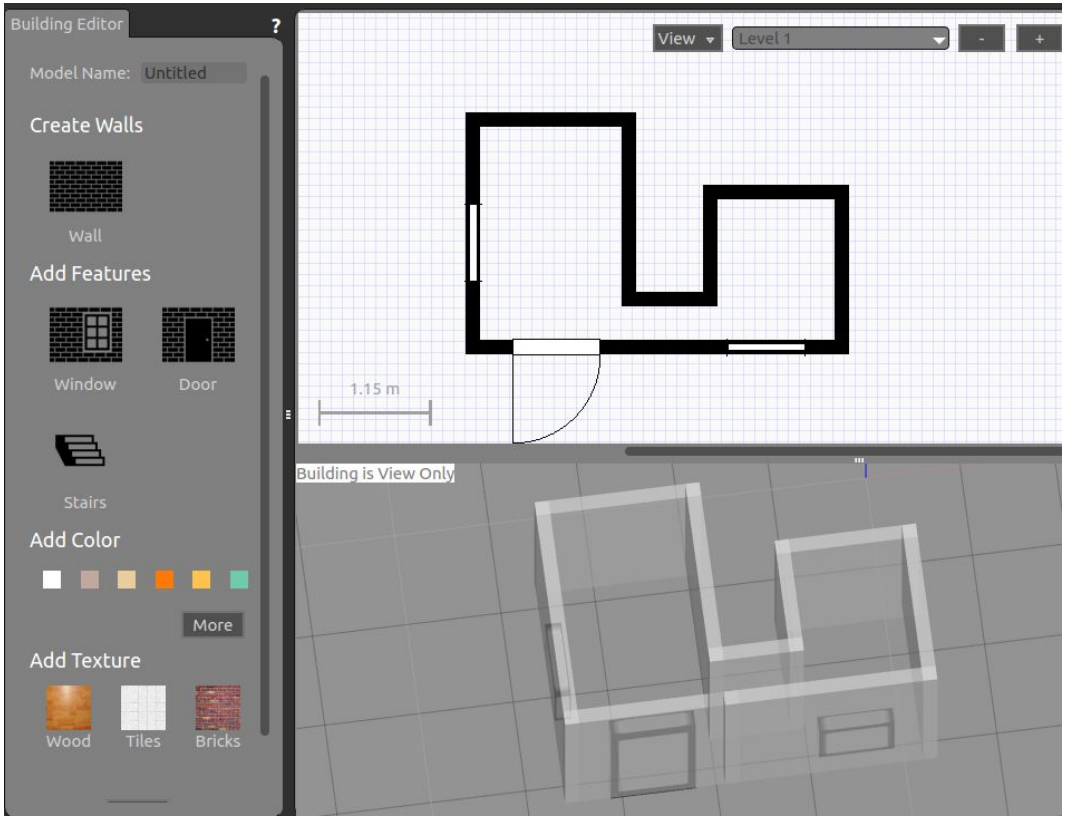
roslaunch gazebo_ros empty_world.launch



Editor de Construções

- Criação de ambientes indoor: paredes, escadas, portas e janelas.

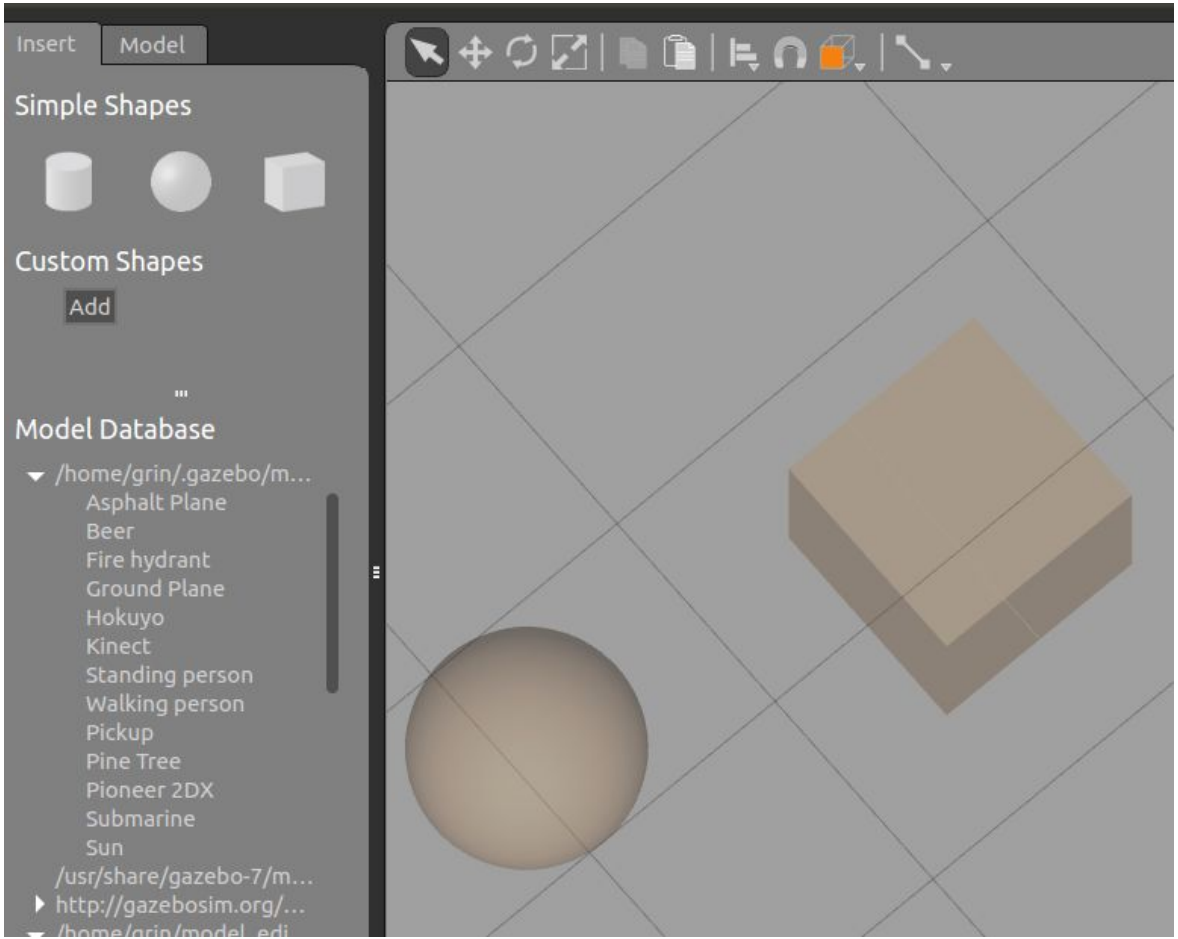
- Edit >> building editor



Editor de modelos

- Edição de modelos simples.

- Baseado em parelelepipedos, cilindros e esferas.



Criando mundos (.world)

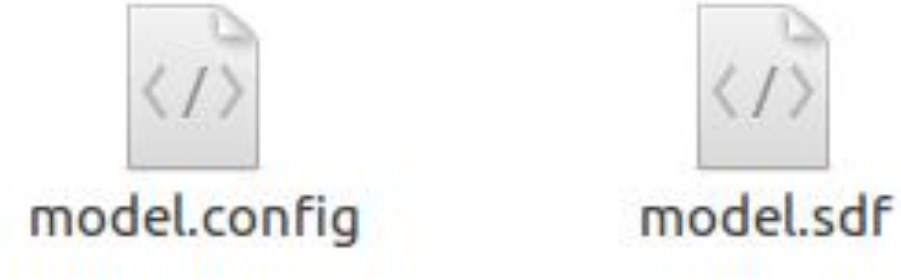
- Após inseridos todos os modelos da cena, incluindo robôs, objetos, construções e plugins:

File >>> save world as

Salva todo o cenário e seu estado físico em um único arquivo.

Programação dos modelos

- XML
- Formatos: SDF, XACRO, URDF



Arquivos para exemplos e atividades da aula

cd ~/catkin_ws/src

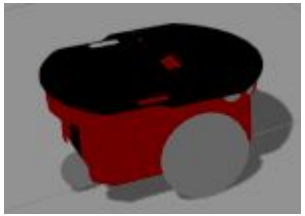
git clone https://github.com/guimarinsjf/aulagazebo.git

catkin_make

abrindo mundo vazio:

roslaunch aulagazebo world.launch

Exemplo: robô diferencial



.DAE

ros plugin
differential
robot

rotational
joint 1

rotational
joint 2

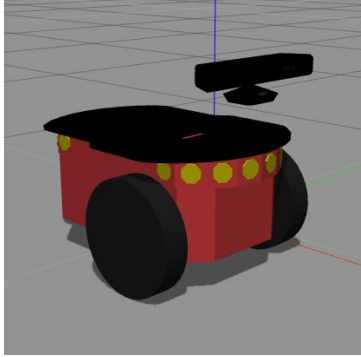
RW

chassi

LW

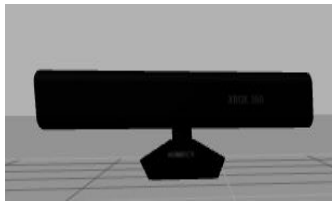
~catkin_ws/src/aulagazebo/models/my_robot/model.sdf

Adicionando sensores: exemplo kinect

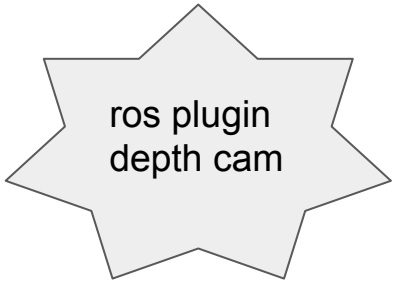


- Adicionar um link com o formato e visual desejado e atribuir o código do sensor contendo o ros pluggin.

- Criar uma junta para acoplar esse link

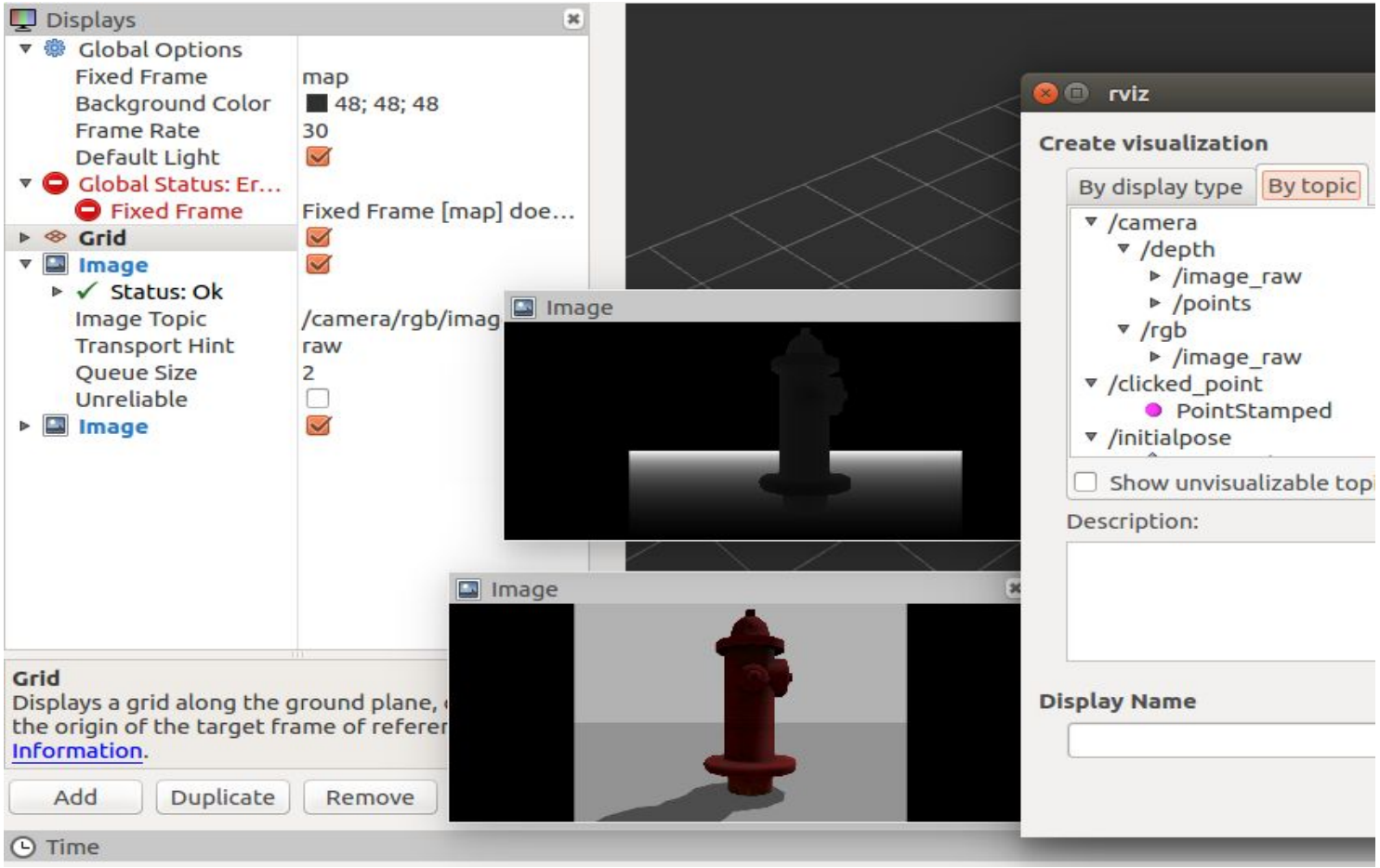


visual .dae



~catkin_ws/src/aulagazebo/models/my_robot_kinect/model.sdf

Ferramenta de visualização rviz



Launch personalizado

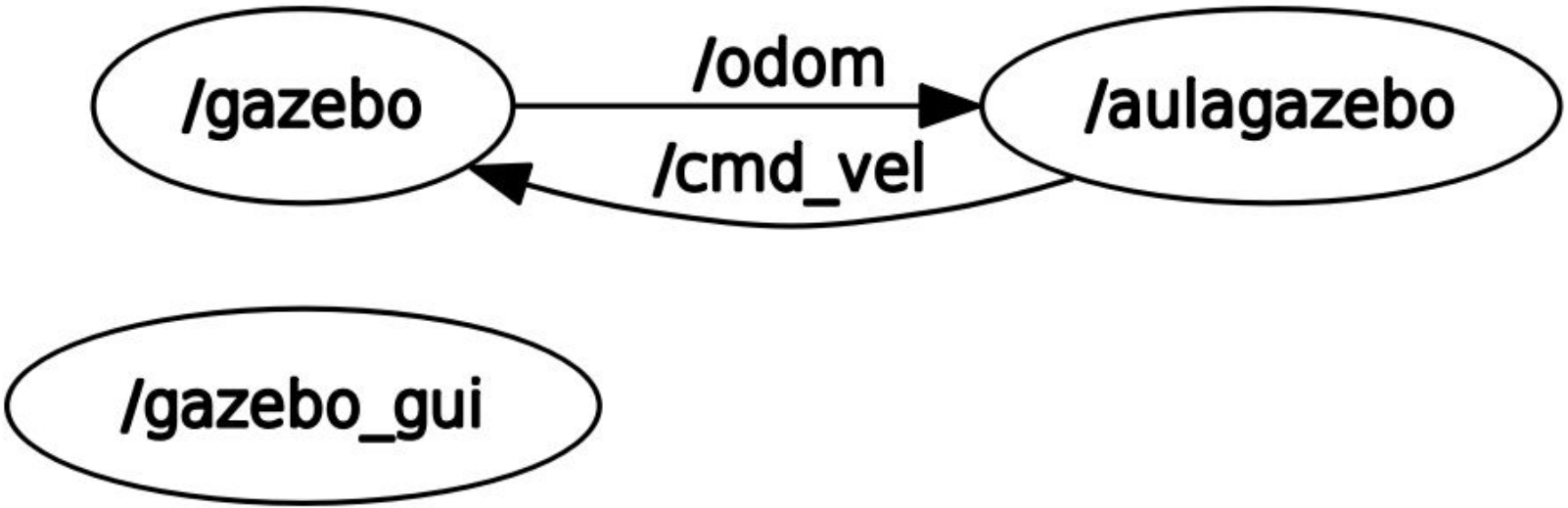
roslaunch aulagazebo aulagazebo.launch

```
<?xml version="1.0"?>

<launch>
<!--  world respawn-->
    <include file="$(find aulagazebo)/launch/world.launch" >
        <arg name="gui" value="true"/>
        <arg name="headless" value="false"/>
        <arg name="world_name" value="$(find aulagazebo)/worlds/empty.world" />
    </include>


<!-- models respawn-->
    <node name="spawn_robot" pkg="gazebo_ros" type="spawn_model"
          args="-file /$(find aulagazebo)/models/my_robot/model.sdf
                -sdf
                -model my_robot
              -x 0 -y 0 -z 0
              -R 0 -P 0 -Y 0"/>

</launch>
```

~/catkin_ws/src/aulagazebo/launch/aulagazebo.launch

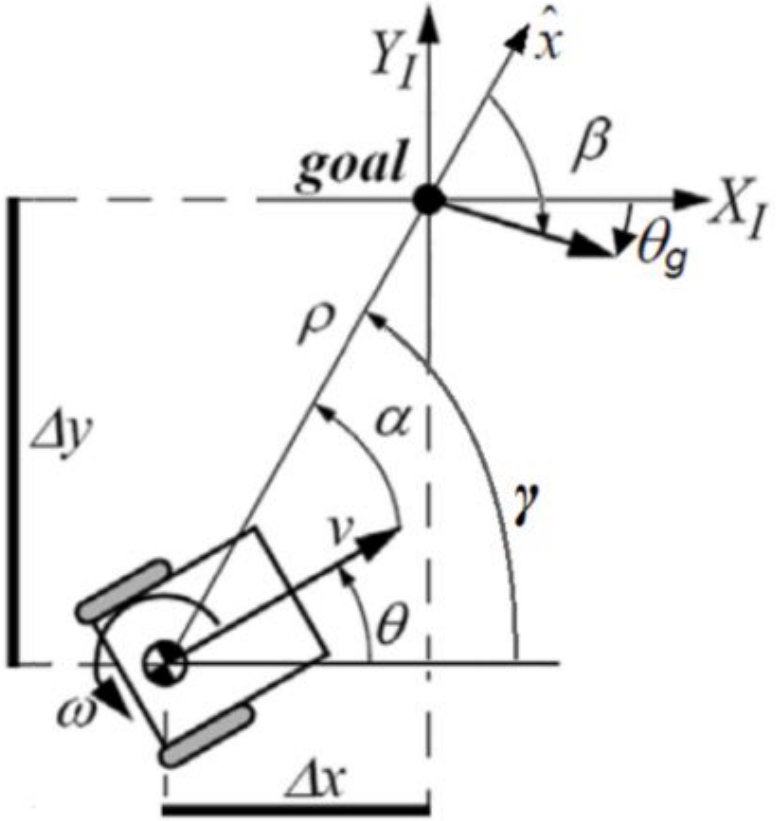
Exemplo de aplicação do rospy: Controle de Posição

rosrun aulagazebo aulagazebo.py



~/catkin_ws/src/aulagazebo/scripts/aulagazebo.py

Controlador



$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\gamma = \arctan 2(\Delta y, \Delta x)$$

$$\alpha = \gamma - \theta_R$$

$$\beta = \theta_G - \gamma$$

$$v = k_\rho \, \rho$$

$$\omega = k_\alpha \, \alpha + k_\beta \, \beta$$

Atividade

- Edite a textura do modelo My Ground Plane com uma foto sua.
- Crie e salve um mundo com My Ground Plane na posição [x=0, y=0, z=0] e aulagazebo_parede na posição [x=0, y=0, z=0].
- Edite o modelo my_robot, adicionando um laser scan. Configure o frame para [x =0 , y = 0, z = 0.4, R= 0, P=0, Y= 0]. Configure o sensor para leituras de -3.14 a 3.14, 360 medidas de um em um grau, e alcance 10 metros.
- Crie um arquivo launch que abra o seu mundo e inicie seu robô na posição [x =-0.75 , y = -0.85, z = 0, R= 0, P=0, Y= 0].
- Edite o programa aulagazebo.py para: reconhecer a passagem através do laser, e fazer um controle de posição para o centro com orientação perpendicular a passagem.

Ilustração do controle