Gazebo에서 모델 생성 및 plugin 작성

참조

World file: http://gazebosim.org/tutorials?cat=guided_i&tut=guided_i1

Plugin, cmakelists file: http://gazebosim.org/tutorials?cat=guided_i&tut=guided_i5

1. 사용할 workspace 생성

   $ cd~

   $ mkdir gazebo_ws

2. 생성된 workspace 안에 world file, CMakeLists file, plugin file을 생성

   $ cd gazebo_ws

   //world file 생성

   $ gedit velodyne.world

```
<?xml version="1.0" ?>
<sdf version="1.5">
  <world name="default">
    <!-- A global light source -->
    <include>
      <uri>model://sun</uri>
    </include>
    <!-- A ground plane -->
    <include>
      <uri>model://ground_plane</uri>
    </include>

    <!-- A testing model that includes the Velodyne sensor model -->
    <model name="my_velodyne">
      <include>
        <uri>model://velodyne_hdl32</uri>
      </include>

      <!-- Attach the plugin to this model -->
      <plugin name="velodyne_control"

<!--filename에서 각자 컴퓨터에 맞는 경로를 설정해줘야함-->

 filename="/home/yelin/gazebo_ws/build/libvelodyne_plugin.so">
        <velocity>1</velocity>
      </plugin>
    </model>
  </world>
</sdf>
```

//plugin file 생성

$ gedit velodyne_plugin.cc

```
#ifndef _VELODYNE_PLUGIN_HH_
#define _VELODYNE_PLUGIN_HH_

#include <gazebo/gazebo.hh>
#include <gazebo/physics/physics.hh>

namespace gazebo
{
  /// \brief A plugin to control a Velodyne sensor.
  class VelodynePlugin : public ModelPlugin
  {
    /// \brief Constructor
    public: VelodynePlugin() {}

    /// \brief The load function is called by Gazebo when the plugin is
    /// inserted into simulation
    /// \param[in] _model A pointer to the model that this plugin is
    /// attached to.
    /// \param[in] _sdf A pointer to the plugin's SDF element.
    public: virtual void Load(physics::ModelPtr _model, sdf::ElementPtr _sdf)
    {
      // Safety check
      if (_model->GetJointCount() == 0)
      {
        std::cerr << "Invalid joint count, Velodyne plugin not loaded\n";
        return;
      }

      // Store the model pointer for convenience.
      this->model = _model;

      // Get the first joint. We are making an assumption about the model
      // having one joint that is the rotational joint.
      this->joint = _model->GetJoints()[0];

      // Setup a P-controller, with a gain of 0.1.
      this->pid = common::PID(0.1, 0, 0);

      // Apply the P-controller to the joint.
      this->model->GetJointController()->SetVelocityPID(
          this->joint->GetScopedName(), this->pid);

      // Set the joint's target velocity. This target velocity is just
      // for demonstration purposes.
      this->model->GetJointController()->SetVelocityTarget(
          this->joint->GetScopedName(), 10.0);

      // Default to zero velocity
      double velocity = 0;

      // Check that the velocity element exists, then read the value
      if (_sdf->HasElement("velocity"))
        velocity = _sdf->Get<double>("velocity");

      // Set the joint's target velocity. This target velocity is just
      // for demonstration purposes.
      this->model->GetJointController()->SetVelocityTarget(
          this->joint->GetScopedName(), velocity);
    }

    /// \brief Pointer to the model.
    private: physics::ModelPtr model;

    /// \brief Pointer to the joint.
    private: physics::JointPtr joint;

    /// \brief A PID controller for the joint.
    private: common::PID pid;
  };

  // Tell Gazebo about this plugin, so that Gazebo can call Load on this plugin.
  GZ_REGISTER_MODEL_PLUGIN(VelodynePlugin)
}
#endif
```

//CMakeLists file 생성

```
$ gedit CMakeLists.txt
```

```
cmake_minimum_required(VERSION 2.8 FATAL_ERROR)

# Find Gazebo
find_package(gazebo REQUIRED)
include_directories(${GAZEBO_INCLUDE_DIRS})
link_directories(${GAZEBO_LIBRARY_DIRS})
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} ${GAZEBO_CXX_FLAGS}")

# Build our plugin
add_library(velodyne_plugin SHARED velodyne_plugin.cc)
target_link_libraries(velodyne_plugin ${GAZEBO_LIBRARIES})
```

3. Workspace 안의 file들을 compile하여 실행파일 만들기

```
$ cd ~/gazebo_ws

$ mkdir build

$ cd build

$ cmake ..

$ make
```

4. 만든 world file 실행

```
$ cd ~/gazebo_ws

$ gazebo velodyne.world
```