

Infinium Robotics - Assignment Documentation

System Requirements

i. Install ROS and gazebo in the system

I have used Ubuntu 14.04, ROS-indigo and Gazebo 7. Refer to <http://wiki.ros.org/ROS/Installation> for instructions.

ii. Install libquirc

Download the package **libquirc** → <https://github.com/dlbeer/quirc>

(I will also include the package separately along with the code.)

Installation instructions are available in the README file.

You can test the library by `./qrtest qr_image_name.jpg`

Running the simulation

Make sure that you have run `catkin build` after you have placed the folders in the `src` folder in the `src` folder of your catkin workspace.

World:

The description of the world is given in `wall_qr3.world` file.

Models:

The models for the world file is given in the `QR_Simulation/models` folder. Place the models in the

`~/ .gazebo/models` folder.

The models contains models for several warehouse objects and the marker tags. If you want to change the marker tag, just replace the picture in the `materials/texture` folder in the marker models.

UAV:

Firefly hexrotor is used as the platform for the simulation. I have used the model from the `rotors_simulation` package. ([github_link](#))

The properties of the hexrotor is given in the `rotors_gazebo/resource/firefly_with_camera2.yaml` file.

It is important to account for the mass of additional payloads (such as gimbal and camera) for accurate vertical positioning. Other parameters for the `lee_position_controller` (such as velocity gain and angular rate gain) are given in the `rotors_gazebo/resource/lee_controller_firefly.yaml` file.

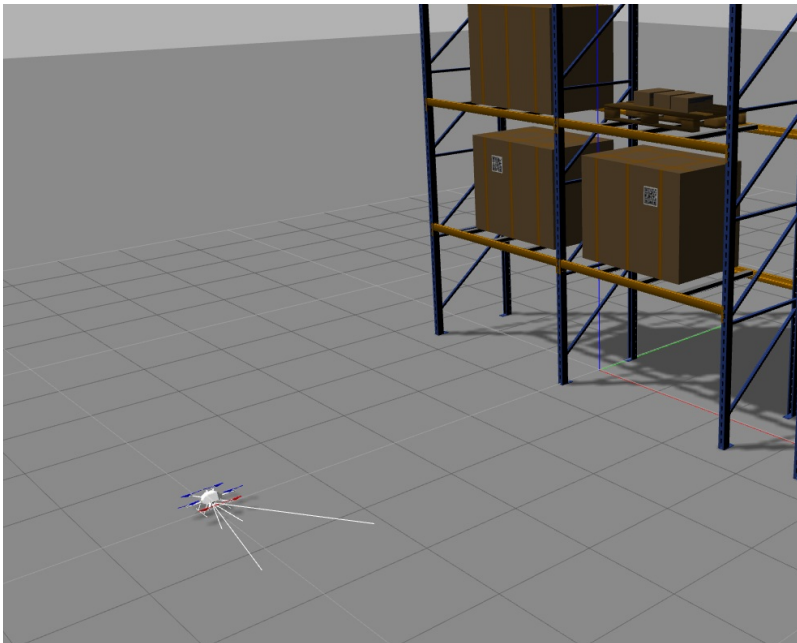
Use this line to open the launch file.

```
roslaunch rotors_gazebo mav_qr_code_scan2.launch .
```

This loads up the world with the hexrotor at (0.0, -5.0, 0.0) as per the lines:

```
<arg name="x" default="0.0"/>
<arg name="y" default="-5.0"/>
<arg name="z" default="0.1"/>
```

The simulation will begin as seen in the picture below.



The simulation is paused when it launches. When you press play, the UAV flies to the position (-2.5,-2.9,2.5) as per the line:

```
<node name="waypoint_publisher" pkg="rotors_gazebo" type="waypoint_publisher" output="screen" args="-
2.5 -2.9 2.5 90 1"/>
```

Lee position controller is utilized to position the UAV at the desired location. This controller is not a path planner. So use this controller only for minor changes in the position.

Navigation of UAV through waypoint publisher

The global path planning can be performed using a waypoint publisher node. Here the `waypoint_publisher_aisle` node is used to generate waypoints along a straight line at specified intervals. For example:

```
roslaunch rotors_gazebo waypoint_publisher_aisle -2.5 -2.4 8.5 -2.4 20 2.5 90 __ns:=/firefly
```

generates 20 equally spaced waypoints between (-2.5,-2.4) and (8.5,-2.4) at an altitude of 1.0m. The UAV will also have a yaw orientation of 90 degrees.

The source code of the waypoint publisher node is in the `rotors_gazebo/src/waypoint_publisher_aisle.cpp` file.

The 2-axis gimbal stabilizer

The gimbal will stabilize the camera to look at the horizontal direction irrespective of the camera roll or pitch.

This node is launched as part of the launch file.

```
<node name="firefly_cam_hold_horizontal2" pkg="venky_controls" type="cam_hold_horizontal2" args="-robo
t firefly" />
```

The source code of the gimbal stabilizer node is in the `venky_controls/src/cam_hold_horizontal2.cpp` file. The description of the gimbal and the camera properties (such as focal-length, resolution) is given in `rotors_description/urdf/firefly_camera_2axis_gimbal.xacro` file. The PID controller parameters are given in the `venky_controls/config/camera_2axis_gimbal_controller_firefly.yaml` file.

Reading the QR Code

The QR code can be read using the `read_qr_code` node:

```
roslaunch venky_controls read_qr_code -robot firefly
```

The node reads the image input from the topic `/firefly/camera1/image_raw`.

Depth Information

The depth information can be visualized using the node `image_view`:

```
roslaunch image_view image_view image:=/camera2/depth/image_raw
```



The parameters of the depth camera can be found in the file:

```
rotors_description/urdf/firefly_camera_2axis_gimbal.xacro
```