Run Raven simulator in Gazebo with demo automove msgs

term1:

source ~/catkin\_ws/devel/setup.bash

roslaunch raven\_ros\_gazebo ravenII\_world.launch

no args = no controller (instant joint updates)

with control:=true uses gazebo pid plugin for controller

term2:

source ~/catkin\_ws/devel/setup.bash

~/catkin\_ws/src/raven\_imposter/scripts/pub\_automove.py

Gazebo physics engine: ODE (default), can be changed with arg in raven\_world.launch

tutorial link: http://gazebosim.org/tutorials/?tut=ros\_control

Create a ros\_gazebo catkin package

contains all content contained in rrbot\_control/description/gazebo packages

create worlds directory and include raven.world basic world file

urdf directory with raven.xacro (only?) file

config directory for raven\_control.yaml

PID gains and controller settings must be saved in a yaml file that gets loaded to

the param server via the roslaunch file.

launch for raven\_world.launch and raven\_control.launch

raven\_world.launch launches raven\_control.launch and gazebo's empty\_world.launch

create raven\_control.launch file

Load joint controller configurations from YAML file to parameter server

create controller types in .yaml

http://wiki.ros.org/ros\_control -> controller type info

only absolute position controller for now?

load the controllers

convert joint states to TF transforms for rviz, etc

create raven\_world.launch file

launches gazebo with empty world

loads URDF into the ROS Parameter Server (from .xacro file)

Run a python script to the send a service call to gazebo\_ros to spawn a URDF robot

launches ros\_control rrbot launch file