Assignment 2 ROS

Rafal Grasman

Menno Broeders

The assignment is to investigate which method is best for sending the robot to a certain position, facing a certain direction.

There are two options to do this, which we will both provide code for.

The first option is point-shoot, which is also the easier option. This involves the robot turning into the direction of the goal position, then moving to the spot, and then turning into the final direction. The second option uses control laws, such as feedback control. The direction and velocity are proportional to the remaining distance, or angle.

The two files we created are steering\_pointshoot\_node.cpp and steering\_servoing\_node.cpp. The pointshoot node turns, moves and finally turns again, in order. The servoing node uses feedback control to adjust its speed and angle. Both nodes listen to the same format of message, which is x, y, angle. X and Y to define the final position the stagesim must be at, facing the direction defined by angle (in degrees).

To run our code, first extract the zipfile to a ros workspace. From this workspace, run rosmake to compile.

In a second terminal, run ‘roslaunch assignment2 assignment2-pointshoot.launch’ for the pointshoot variant, or ‘roslaunch assignment2 assignment2-servoing.launch’ for the servoing variant.

This will open a window which shows the stagesim in a empty world.

In a third terminal, run ‘rostopic pub /stagesim/steering\_pointshoot/goal geometry\_msgs/Point -- <x> <y> <angle>’. You must substitute x, y, angle for values you would like to see.

In case of servoing: run ‘rostopic pub /stagesim/steering\_servoing/goal geometry\_msgs/Point -- <x> <y> <angle>’ instead.

Differences between both modes:

Servoing makes more turns, where pointshoot is much more direct.

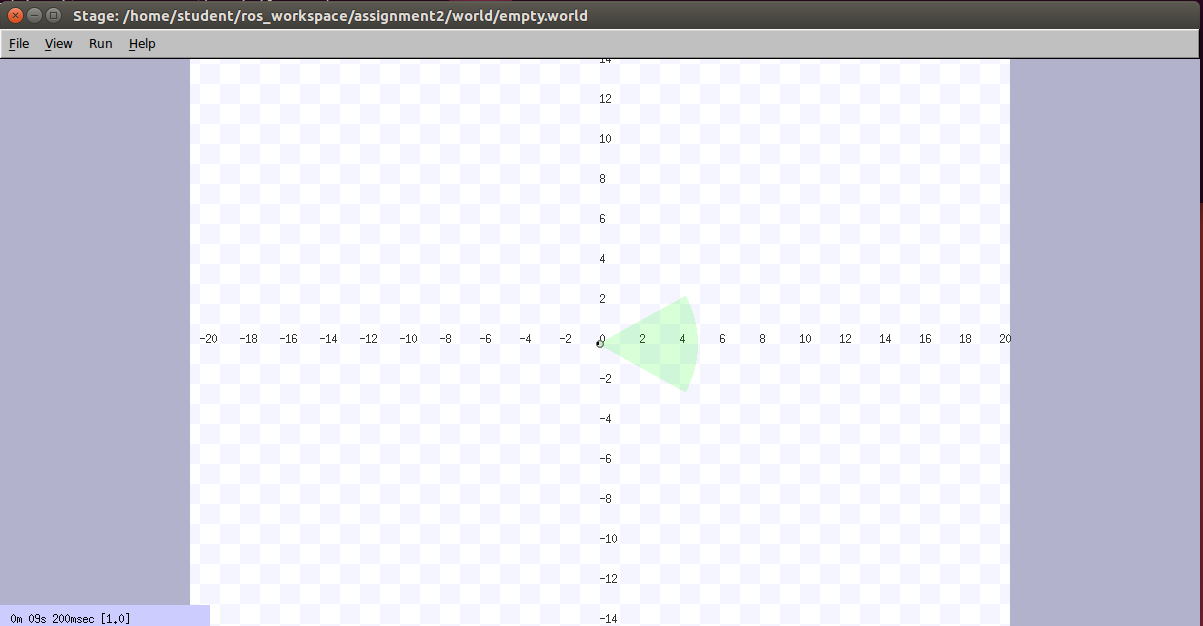
Servoing mode is slower compared to pointshoot.

Servoing mode is programmed with less code and has less steps to develop, but implements a PID-controller, which is more difficult. Pointshoot has slightly more code and steps to develop.

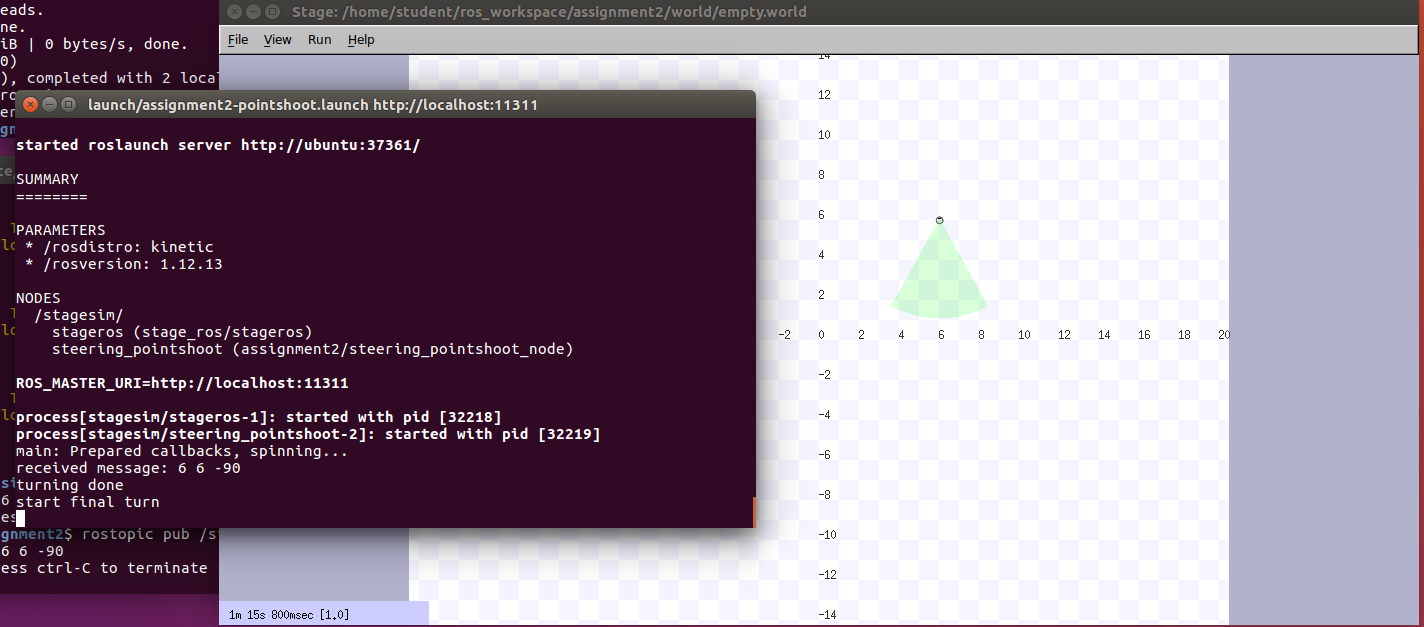
Screenshots:

Pointshoot:

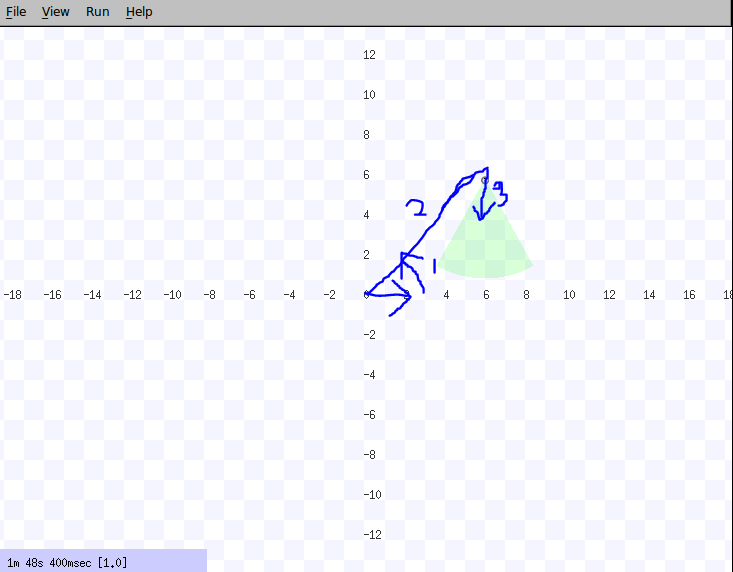
Starting position:



Command using X, Y, Angle = 6, 6, -90:

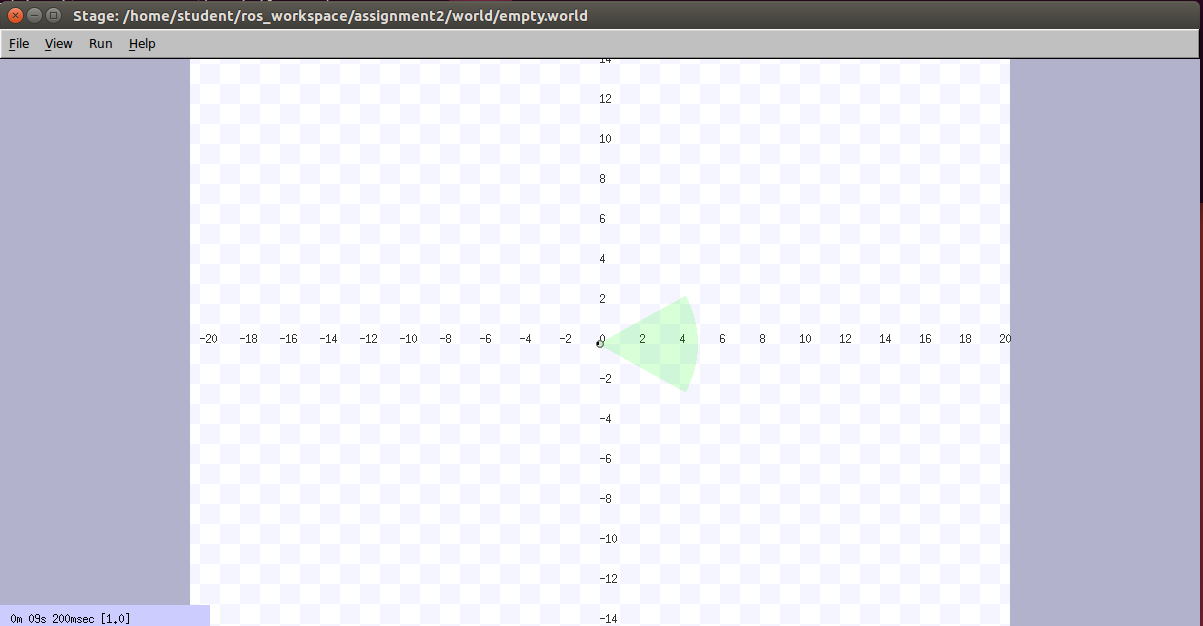


Steps taken: 1: turn, 2: move, 3: final turn

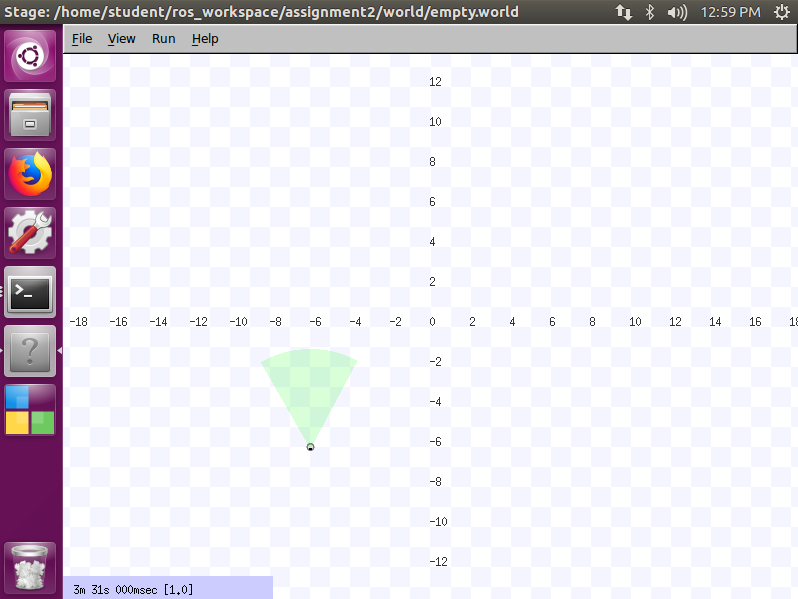


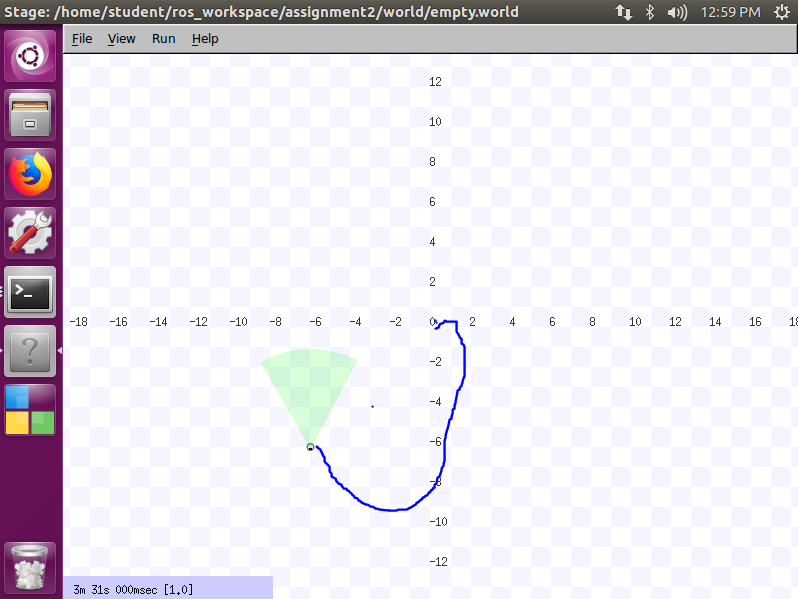
**Servoing:**

**Starting situation:**

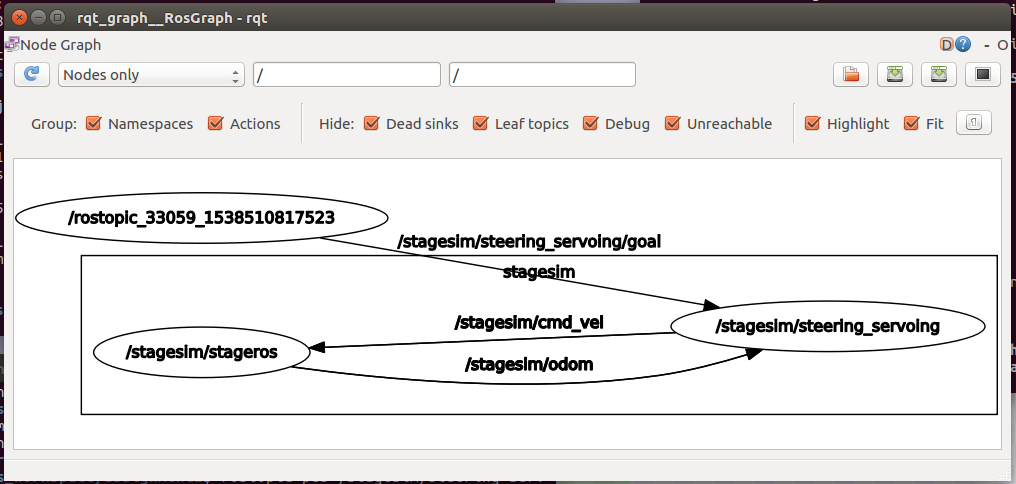


**After x,y,z = -6, -6, 90**



**Path followed (roughly, not too great with paint on a touchpad).**

**Finally, our rqt\_graph:**



**We use command line to publish on the /goal topic for x,y,angle. The servoing/pointshoot node listens to this and calculates what it must publish on stagesim/cmd\_vel for it to move. Finally, stagesim/odom is used to keep track of its current position and direction.**