Intro to ROS and Square
In class programming exercise
January 30, 2017

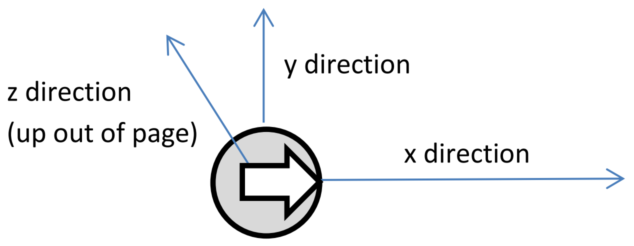
This is what we will work on in class. If you are trying this on your own, after you complete step 3 you should be caught up with the class. If you run into trouble please email Dr. Archibald (ca861@msstate.edu). These instructions assume that you have worked through the ros tutorials (http://wiki.ros.org/ROS/Tutorials) which will have asked you to create a `catkin_ws/src` folder that we will use.

1. First, we will create a new package to house the code we will work on in lecture.
   - Open a new terminal in Ubuntu and change directory (`cd`) to `catkin_ws/src`
   - In `catkin_ws/src` type the following (all on one line) and then press enter
     - `catkin_create_pkg ai_lectures std_msgs rospy roscpp sensor_msgs geometry_msgs`
   - Unzip programming exercise files from blackboard into new `ai_lectures` folder
   - Then make a directory within `ai_lectures` for our python scripts by running
     - `mkdir scripts`
   - Then move the python scripts into the `scripts` folder by running
     - `mv *.py scripts/`
   - We need to make the scripts executable, so now type (and don't forget the *)
     - `chmod +x scripts/*`
   - Now we need to make a directory to hold our world files
     - `mkdir world`
   - And we need to copy over the necessary files into that folder
     - `mv *.world world/`
     - `mv *.png world/`
2. Now we will test that this is all working. Run the following command from the terminal. You should see the simulator window pop up and a robot (circle shape) driving in a circle.
   - `roslaunch ai_lectures square.launch`
3. Now you can modify the code in `square.py` to get the robot to move in square or other interesting shape.

NOTES:

Commands are sent via a `Twist` message:

The following Coordinate system is used, where the grey circle is the robot, facing in the direction of the arrow:



- `Twist` – A `Twist` object has two main fields, which allow for linear and angular velocities to be set. This is super general so that it could work for any robot. We will only use two of the fields (**bolded** below), but it helps to remember which two if you understand what they all mean.
    - `Twist.linear`
        - **`Twist.linear.x`** – this sets the linear velocity in the x direction, which is forward/backwards (for +/- values). This gives the forward speed of the robot
        - `Twist.linear.y` – this sets the linear velocity in the y direction, which is sideways. Our robot can't move like this, so we will not use this field.
        - `Twist.linear.z` – this sets the linear velocity in the z direction, which is up. Our robot can't fly, so we will not use this field either.
    - `Twist.angular`
        - `Twist.angular.x` – This sets the angular velocity/ rotation around the x axis, or the roll of the robot. Since we are on the ground, we cannot rotate around this axis, so we won't use this.
        - `Twist.angular.y` – This sets the angular velocity /rotation around the y axis, or the pitch of the robot (moving front up/down). Since we are on the ground, again, we won't use this.
        - **`Twist.angular.z`** – This sets the angular velocity/rotation around the z axis, which rotates the robot left/right. We will use this command to turn the robot. Units are in radians/sec - positive is a left turn (increasing the angle/heading of robot), while negative is right turn.

Tips:

- To drive the robot forward/backward, set `twist.linear.x` to +/- values. (0.5 seems to work ok)
- To spin the robot, set `twist.angular.z` to +/- values (again, 0.5 seems a good starting point)
- If you have trouble getting ROS to `roscd` to your directory, try running the following command from the `catkin_ws` directory
    - `source devel/setup.bash`
    - Add the line "`source <path to your catkin>/catkin_ws/devel/setup.bash`" to your `.bashrc` file to have this happen automatically when you log into a terminal. (make sure that `<path to your catkin>` is replaced by the path on your system to the `catkin_ws` folder. You can get this by typing `pwd` from the terminal when you are in the directory.