

Worksheet 1

Solutions:

Due Date: 16.11.2020

Exercise 1.

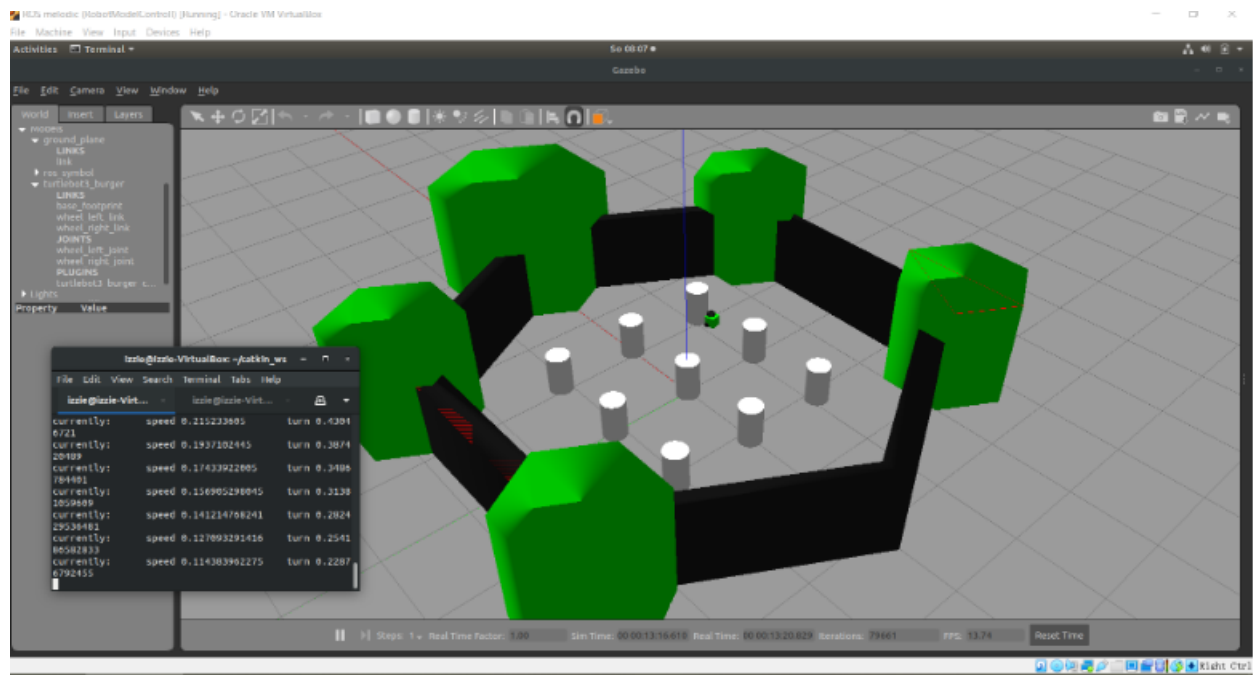
1.3 Steering

A table describing what each of these buttons do.

u	Circle ahead to the left
i	Steer straight ahead
o	Circle ahead to the right
j	Rotation by counter clockwise
k	Stop steering
l	Rotation by clockwise
m	Circle backwards to the left
,	Steer straight back
.	Circle backwards to the right
q	increase speed
z	decrease speed

1.4 Pictures!

Here is a picture that shows the working simulator where I drive the robot close to the top-right pillar of the arena.



Exercise 2.

What is the difference between `roslaunch` and `roslaunch`?

`roslaunch` can only execute a single node from a single package while `roslaunch` could launch two or more nodes from multiple packages at a time. Meanwhile `roslaunch` does put each node into its own process and pipe the output of that node to a log file. Whereas `roslaunch` generates no log file. `roslaunch` like a shortcut execute a node in the terminal. In my opinion `roslaunch` could be tracked but `roslaunch` not.

What dose the command `"rosmmsgshowstdmsgs/String"` do?

This command displays the fields in a ROS message type. Here the type is string. I may omit the package name of the type String, in which case `rosmmsg` will search for matching type String in all packages.

How can you make sure that a node is subscribing to a topic?

Using `rostopic echo`. It shows the data published and subscribed on a topic.

`rostopic echo [topic]`

Using `rostopic list`. It returns a list of all topics currently subscribed to and pulished.

When we press the options -s after print the command, it could return only subscribers.

Exercise 3.

Batteries lose capacity over time. What do you need to do to assure a long lifetime of a battery?

First of all, everyone should use the battery with safety instructions. Safe battery usage and storing batteries in a correct way would assure a long lifetime of a battery.

For example about safe battery usage:

1. Don't let the battery get empty.
2. Assure good operation temperature. When the battery's temperature goes up, it should stop to be using until the temperature is close to normal.
3. Fix the battery in a correct position and couldn't be moved anywhere, when the battery is used.
4. Do not apply physical stress on the battery like huge pressure on the surface.
5. Balance all cells from time to time so they discharge equally during use.

For example about safe storing batteries:

1. Charge to 70 percent.
2. Use the storing program of the charger.
3. Check regularly, at least every 3 months.
4. Store in special fire-proof boxes with integrated extinguishing agent. Meanwhile assure the right temperature. Inside the batteries chemical materials would be reacted in fire or in a high temperature.

What can you do to prevent your battery from exploding?

1. Store the batteries away from fire and high temperature.
2. Avoid the overuse of batteries. Because overuse will lead to a higher temperature inside the batteries.
3. Do not use the batteries during the charging.
4. Stop using the batteries as soon as possible when it has been damaged or some index was unusual after the check.

Exercise 4 Moin, ROS!

4.1 Some Code

Explain what happens when we call the function `schnacker` in the following code:

```
1 #!/user/bin/env python # The script is executed as a Python script.
2 import rospy # Import rospy if we need to write a ROS Node.
3 from std_msgs.msg import String # After that we could reuse the std_msgs/String message
  type for publishing.
4
5 def schnacker(): # We define a method named schnacker.
6     x = rospy.Publisher('chatter', String, queue_size=8) # The node is publishing to the
      chatter topic using the message type String. The queue_size method limits the
      amount of queued messages.
7     rospy.init_node('talker', anonymous=True) # Tells rospy the name "talker" of the node
      . After rospy has had this information, it can start communicating with the ROS
      Master. And "anonymous=True" means that it was ensured that the node has a unique
      name by adding random numbers to the end of NAME.
8     repetitions = rospy.Rate(20) # With the help of its method sleep(), it offers a way
      for looping as the desired rate. Every second the loop will be executed 20 times.
9     while not rospy.is_shutdown(): # Check the method rospy.is_shutdown() and go through
      the loop. That can ensure if the program should exit.
10         foo = "Moin ROS! The timestamp is: %s" % rospy.get_time() # The string named
            foo is a sentence following by a current time. It was executed by a method
            rospy.get_time()
11         x.publish(foo) # That published a string to out chatter topic
12         repetitions.sleep() # The method sleeps enough to maintain the desired rate
            through the loop
13
14 if __name__ == '__main__': # In the standard Python __main__ check, this will catch a
      rospy.ROSInterruptException exception, which can throw by rospy.sleep() and rospy.Rate.
      sleep() when the node is shutdown by pressing Ctrl-c.
15     try:
16         schnacker()
17     except rospy.ROSInterruptException:
18         pass
```

4.2 Better documentation

```
1 #!/user/bin/env python
2 import rospy
3
4 def schnacker():
5     pub = rospy.Publisher('chatter', String, queue_size=8) \# I have changed x to pub, pub
      is abbreviation of publishment. I think that is easier to understand.
6     rospy.init_node('talker', anonymous=True)
7     rate = rospy.Rate(20) \# I have changed repetitions to rate. This name is short and
      represents the meaning of this methods.
8     while not rospy.is_shutdown():
9         hello_str = "Moin ROS! The timestamp is: %s" % rospy.get_time() \# I have
            change foo to hello_str. That string refers to the function of string while
            foo is just a name.
10         pub.publish(hello_str)
11         rate.sleep()
12
13 if __name__ == '__main__':
14     try:
15         schnacker()
16     except rospy.ROSInterruptException:
17         pass
```

In my opinion a good documentation could lead readers to think about and understand it. For example the origin name `foo`. When I first time looked at this variable, I must to

think about the meaning even though I could also by reading the assignment of foo get the meaning. But hello_str is so easier to read. Therefore a name of variable is not only a symbol but also a real name that could transmit information to the readers.

Feedback

How much time did you spend on doing this sheet per person

I have spent 8 hours.

Was it too easy, easy, ok, hard, too hard?

Nothing is too easy

Installation followed the video: easy

Understanding the command: ok

Compare some details like roslaunch: hard

Using the ROS without video and python program: too hard

What additional resources (blogs, papers, books, tutorials, etc) did you use?

<https://www.theconstructsim.com/ros-5-mins-008-difference-roslaunch/>

<http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29> [http:](http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics)

[//wiki.ros.org/ROS/Tutorials/UnderstandingTopics](http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics)

Any other issue?