RBE 3002 Unified Robotics IV: Navigation

D-Term 2015

Lab Assignment #2: Mobile Robot Kinematics and Odometry

Introduction

In this lab, you will develop a kinematics model for the TurtleBot. In addition, you will program the TurtleBot for differential drive motion, simple trajectory generation and odometry reporting. This lab is to be done individually.

Objectives

Upon successful completion of this lab, you will be able to:

1. Perform robot kinematics calculations.
2. Demonstrate ability to create your own ROS package.
3. Demonstrate functionality for TurtleBot differential drive.
4. Instruct the TurtleBot to execute a pre-planned trajectory and track the robot's position.

Pre-Lab

Calculate the forward and inverse velocity kinematics for the TurtleBot.
Fill in the lab 2 template with pseudo code.

Lab Work

In this lab you will create your first ROS package in Python. Use the `lab2.py` skeleton code provided ~~on Piazza~~ to implement the node described below. The final behavior that you will demonstrate, will be the execution of a predefined trajectory when the bumper is pressed. All problems and questions should be posted on the class Q&A. We encourage you to develop and test all your code in Stage before testing on the real robot.

1. Create a new package using `catkin_create_pkg` named *firstinitiallastname_lab2* (e.g. jdoe_lab2). Create a new Python node within this package by copying `lab2.py` from Piazza.

2. Set up your node to publish Twist messages to the `cmd_vel_mux/input/teleop` topic. Refer to code in the `turtlebot_teleop` package for examples or Lab 1 code.

3. Implement `spinWheels(u1, u2, time)`, which takes as input the velocities of the left and right wheels, computes the translational and rotational robot velocities and drives the robot for the specified period of time.

   Test that you are able to drive the (simulated and/or real) robot.

4. Modify your node to ~~subscribe to Odometry messages. Refer to the Kobuki base code for examples: https://github.com/yujinrobot/kobuki/tree/hydro-devel/kobuki_testsuite#~~ utilize the lookupTransform function ~~provided~~, refer to the TF documentation on the ROS wiki at http://wiki.ros.org/tf/Tutorials/Writing%20a%20tf%20listener%20%28Python%29

5. **Implement the following functions:**

   driveStraight(speed, distance)

   rotate(angle)

   Units should be in meters, radians and meters/second.

   ~~`navToPose(goal)`~~

~~This should take a pose or poseStamped as an argument, for implementation in future labs~~

~~7.~~ Write ~~the~~a function called **`executeTrajectory()`** that executes the following sequence:
   - Drive forward 60cm
   - Turn right 90°
   - Drive forward 45cm
   - Turn left 135°

~~8.~~6. Modify your node to subscribe to BumperEvent messages. Refer to the Kobuki base code for examples: https://github.com/yujinrobot/kobuki/tree/hydro-devel/kobuki_testsuite#

~~9.~~7. Identify when the middle of the bumper is pressed. Use this event to initiate **executeTrajectory().**

~~10.~~8. Later in this class you will need to navigate to waypoints that are along a path to a goal. Write the function **`navToPose(goal)`** to receive a nav goal from a mouse click in RVIZ. The function should take a pose or poseStamped as an argument, turn the robot to the proper heading, and drive straight to that point and rotate to the proper pose.

Extra Credit [5~~5~~ points]

When implemented as bang-bang control, the teleoperation code above results in very abrupt and jerky movement. For extra credit, modify your code to include gradual acceleration and deceleration. Be careful not to overshoot the intended distances.

Extra Credit [5 points]

Due to the energy loss and low speed nature of a differential drive, it is common for vehicles to turn in an arc. Implement a **driveArc(radius, speed, angle)** function to drive the robot in an arc.

Deliverables ~~and Deadlines~~

~~This lab is due on Tuesday April 1st at 11:59pm.~~ All submissions must be made through Blackboard, except the sign-off sheets, which can be either scanned and uploaded or placed in the box in AK120. Please submit:

1. Copy of the sign-off sheet
2. Compressed folder containing your package (e.g. jdoe_lab2.zip).
3. A brief report communicating your findings regarding the odometry of the robot. If you were unable to complete the entire lab, you should describe what parts work, which do not and what

Formatted: Strikethrough

Formatted: No underline, Underline color: Auto, Font color: Auto, Strikethrough, Not Expanded by / Condensed by

Formatted: Font color: Auto

Formatted: Font color: Blue

Formatted: Font color: Blue, Strikethrough

Formatted: Font: Calibri, Bold, Not Expanded by / Condensed by

Formatted: Font: Calibri, Bold, Not Expanded by / Condensed by

Formatted: Font: Bold

Formatted: Indent: Left: 0.5"

Formatted: Body Text, Indent: Left: 0", Space Before: 0 pt

Formatted: Font: 11 pt, Font color: Red

Formatted: Body Text, Space Before: 0 pt, No bullets or numbering, Tab stops: Not at 0.57"

Formatted: Font: (Default) Calibri, 11 pt

Formatted: Font: (Default) Calibri, 11 pt

Formatted: Font: (Default) Calibri, 11 pt

Formatted: Font: (Default) Calibri, 11 pt

Formatted: List Paragraph, Indent: Left: 0", Space Before: 0 pt

Formatted: Font: (Default) +Body (Calibri), 11 pt, Condensed by 0.05 pt problems you encountered. The complete document should not exceed 2 typed pages.

4. If you used external sample code other than the packages listed in this lab document, please also submit a document called References.txt that lists all such sources, as well as a very brief description of what you used from each one.

RBE3002 Lab2: Mobile Robot Kinematics and Odometry

Name: ____

Lab Sign-offs

TASK	Simulated*	Real	SIGNED BY	DATE/TIME
Prelab				
Spin Wheels	☐	☐		
Drive Straight	☐	☐		
Rotate	☐	☐		
Bumper Press Trajectory	☐	☐		
Nav To Goal	☐	☐		
Drive Arc *	☐	☐		
Ramped deceleration *	☐	☐		

*Demonstrating each capability on the simulator <u>or</u> on the real robot is sufficient. However, only 80% credit will be given for tasks shown only in simulation.

Grading Rubric

[100 points] All pre-lab and in-lab procedures are completed and demonstrated prior to the deadline. All deliverables and sign-off sheet are submitted on time. The code is well commented and structured.