Open-loop and Closed-loop control 4

(Max score: 40)

16-311: Introduction to Robotics (Fall 2017)

OUT: October 22, 2017, at 1:00am

DUE: November 1, 2017 at 16:00pm - Available late days: 1

Instructions

Homework Policy

Homework is due on autolab by the posted deadline. As a general rule, you have a total of 8 late days. For this homework you cannot use more than 1 late day. No credit will be given for homework submitted after the late day. After your 8 late days have been used you will receive 20% off for each additional day late.

You can discuss the exercises with your classmates, but you should write up your own solutions. If you find a solution in any source other than the material provided in the course, you must mention the source.

Submission

Create a tar archive of the folder with your ROS packages/nodes and submit it to Homework 4 on autolab. You should also have one PDF file in your archive, with an explaination of your results regarding the experiments with the robot, and the instructions for running the nodes, if any.

Contents

1 Follow a given path in open-loop (14 points)

Implement in ROS/Gazebo an open-loop controller that lets the robot following a given path. The path is given in a parametric form that defines a sequence of Cartesian coordinates $(x(s), y(s))$. Use differential flatness to derive the kinematic controls that let the robot following the target path.

The robot always starts on path: it immediately moves according to the kinematic controls associated to the path (i.e., it doesn't need to reach a path point, the path is implicitly defined relative to the robot).

Consider two families of parametric paths: ellipsis and Archimedean spirals. Make the code parametric, in the sense that a generic ellipsis and a generic spiral (both centered in $(0, 0)$) can be defined based on the input parameters (the x and y semi-axis for the ellipsis, and the growth factor for the spiral).

In particular, report your result for an ellipsis with y semi-axis equal to 3 and x semi-axis equal to 1, and a spiral with growth factor equal to 0.1. Use $ds = 0.01$ as step size in the geometric space.

1. Describe how you have defined the time derivative in the geometric space, as well as the final form for v and ω used to drive the robot along the path.
2. Discuss the effect of different choices for ds and ds/dt.

3. Quantify the error over multiple path revolutions.

1.1 Test on turtlebot (4 points)

Port and test your code on the turtlebot. Quantify the error over multiple path revolutions.

2 Follow a given path in closed-loop (16 points)

Design and implement in ROS/Gazebo a closed-loop PI controller that lets the robot following a given path. The path is given in a parametric form that defines a sequence of Cartesian coordinates $(x(s), y(s))$. The robot always starts significantly off the path.

Consider two families of parametric paths: ellipsis and Archimedean spirals. Make the code parametric, in the sense that a generic ellipsis and a generic spiral (both centered in $(0, 0)$) can be defined based on the input parameters (the x and y semi-axis for the ellipsis, and the growth factor for the spiral). Moreover, robot's initial pose can be given as an optional input.

In particular, report your result for an ellipsis with y semi-axis equal to 3 and x semi-axis equal to 1, and a spiral with growth factor equal to 0.1. Use $ds = 0.1$ as step size in the geometric space for the ellipsis, and $d = 0.25$ for the spiral. Both the ellipsis and the spiral are centered in $(0, 0)$, while the robot starts with pose $\boldsymbol{q} = \begin{bmatrix}1, 2, 0\end{bmatrix}^T$.

Note: Be careful when dealing with the angles. In particular, it is suggested to use $\text{atan2}(y, x)$ for sum or subtraction of angles:

$$\text{atan2}(y_1, x_1) \pm \text{atan2}(y_2, x_2) = \text{atan2}(y_1 x_2 \ \pm \ y_2 x_1, \ x_1 x_2 \ \mp \ y_1 y_2)$$

1. Precisely describe your PI controller and the values of all gains and parameters.
2. Describe how have you selected / tuned controller's gains.
3. Discuss the effects of increasing/decreasing the proportional and integral gains. Show a few sample trajectories corresponding to different choices for the gains.
4. Quantify the error over multiple path revolutions when using either the ground truth from Gazebo or the estimates for odometry for the robot's pose at each time step.

2.1 Test on turtlebot (6 points)

Port and test your code on the turtlebot. Quantify the error over multiple path revolutions.