Report

Robotics: Dynamics and Control

Assignment 1

Dhaval Kadia: 101622808

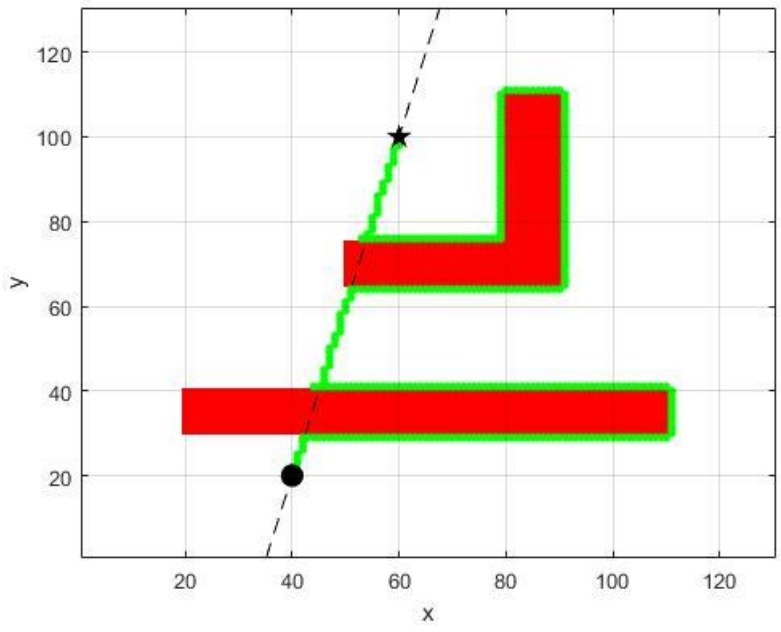
1. **a.** I downloaded and installed Robotics Toolbox from Peter Corke. There has been some difference in different versions of it. The difference is in Bug2.m file. It the size of steps towards the goal. I have used the below module of the latest version of Bug2.m [1] into the Bug2.m that is come after the toolbox installation. This is normalizing the steps, that is useful.

```
% motion on line toward goal
d = bug.goal-robot;
if abs(d(1)) > abs(d(2))
    % line slope less than 45 deg
    dx = sign(d(1));
    L = bug.mline;
    y = -( (robot(1)+dx)*L(1) + L(3) ) / L(2);
    dy = round(y - robot(2));
else
    % line slope greater than 45 deg
    dy = sign(d(2));
    L = bug.mline;
    x = -( (robot(2)+dy)*L(2) + L(3) ) / L(1);
    dx = round(x - robot(1));
end
```

Function Bug2.next(bug, robot): Added this latest module [1]

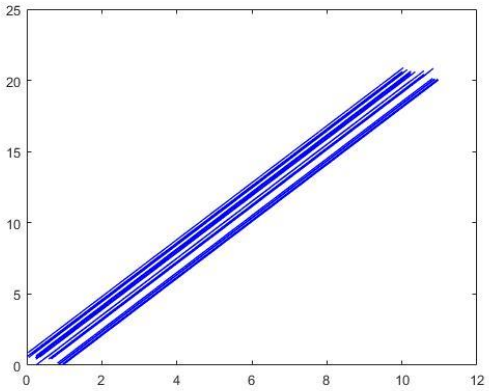
b. I have written a MATLAB file to animate the navigation solution. I have defined the empty map and the obstacles in it. The starting location is denoted by the circle, and the destination location is indicated by the star. The green path is the path on what the robot is navigating.

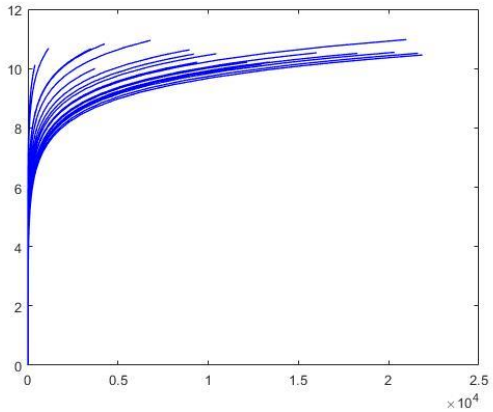
NOTE: If my code shows the error, please temporarily replace bug2.m by the one I provided.



Navigation using Bug2 algorithm

2. I have learned and verified how to solve ODE using ode45. Here are some samples. And then, I solved the given ODE.
    a. Sample: x_dot = 1; y_dot = 2
    b. Sample: x_dot = x; y_dot = 1

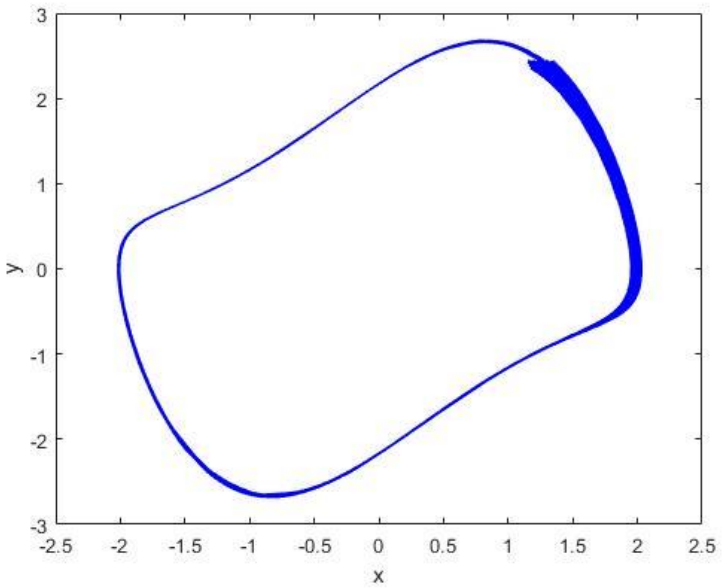




Solution to the given ODE:

X_dot = y

Y_dot = (1 – x * x) * y - x



Solution of x and y of the given ODE

References:

1. https://github.com/petercorke/robotics-toolbox-matlab/blob/master/Bug2.m