ENEE 322 Signals and Systems

Fall 2016 – Extra Credit Project

1 Background

The purpose of the extra credit project is to provide the class with an opportunity to apply some of the tools covered in the lectures and homework to a realistic problem. In this project we will develop a simulation of a set of communicating, cooperating robots that self-organize and perform a coordinated maneuver. The project parallels an ongoing project in an associated class.

2 A Differential Drive Robot

Students in the course *ENEE 408I Autonomous Robotics* are developing robots that can move in formation along a path. The fundamental formation is a line – the robots find each other in region and line up one after the other and then move along a path following the "leader." There may be obstacles a long the way, and the formation must be preserved during encounters with obstacles, or it must be recreated as soon as possible after encountering an obstacle.

The ENEE 408I robots are "simple cars"[1] that have two motors and wheels on opposite sides of the robot body. The motors that turn the wheels are independently controlled, and so, the robot uses a "differential-drive" mechanism to maneuver. It is able to move in straight or curved lines, or to rotate in place. It cannot move sideways, and so, the robot's motion is "nonholonomic."

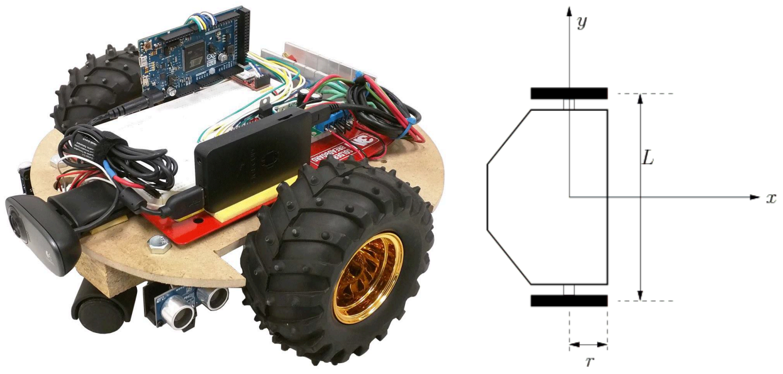

Figure 1 Two-wheel, differential drive robot used in ENEE 408I

[1] See the web site of Steven LaValle (http://planning.cs.uiuc.edu/) for information on planning trajectories and models of vehicles including differential drive cars, airplanes, etc., especially http://planning.cs.uiuc.edu/node659.html

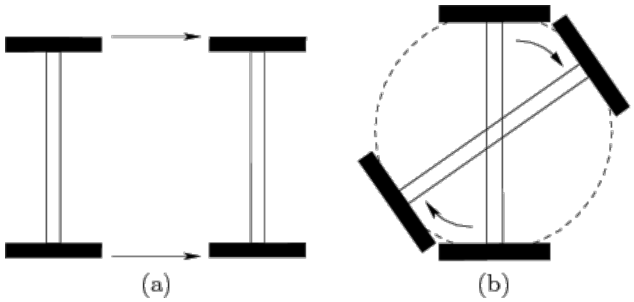

Figure 2 (a) Pure translation occurs when both wheels move at the same angular velocity; (b) pure rotation occurs when the wheels move at opposite velocities.[2]

Moving in the plane, the robot's state is defined by its position (x,y) and its heading angle (q). Taken together, we call the state vector $(x(t),y(t),q(t))$ the robot "pose" at time t.

If we define u_r and u_l to be the wheel speeds of the right and left wheels, respectively, then the equations of motion are

$$\dot{x}=\frac{r}{2}(u_r+u_\ell)\cos q$$

$$\dot{y}=\frac{r}{2}(u_r+u_\ell)\sin q$$

$$\dot{q}=\frac{r}{L}(u_r-u_\ell)$$

Here r is the radius of the wheels and L is the distance between the wheels. The wheel speeds are the "controls" that allow us to maneuver the robot in the plane, including its orientation.

Note that the differential drive car can cause the center of its axle to follow any continuous path in two dimensions. As depicted in Figure 3, it can move between any two configurations by: 1) first rotating itself to point the wheels to the goal position, which causes no translation; 2) translating itself to the goal position; and 3) rotating itself to the desired orientation, which again causes no translation. The total distance traveled by the center of the axle is always the Euclidean distance between the two desired positions. And so, the differential drive robot is *completely controllable*; one can reach any desired pose from any initial pose.

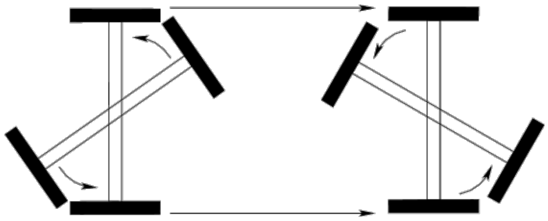

Figure 3 The shortest path traversed by the center of the axle is simply the line segment that connects the initial and goal positions in the plane.

[2] http://planning.cs.uiuc.edu/node659.html

Problem 1: Develop a simulation of the model equations for some open loop controls. Here is some code I used:

```
function dzdt=diffdrive(t,y,rt,RD,lt,LD)
% This defines the right hand side of the differential equation model for
% the simple car with two wheels. RD,rd is the control for the right wheel
% and LD,ld is the control for the left wheel
r=0.1; % Radius of wheel
L=0.5; % Distance between wheels
RD = interp1(rt,RD,t); % Interpolate the data set at time t
LD = interp1(lt,LD,t); % Interpolate the data set at time t
dzdt = [
    (r*(RD+LD)/2)*cos(y(3));
    (r*(RD+LD)/2)*sin(y(3));
    (r*(RD-LD)/L)
    ];

function [t,y] =main
tend=10;
rt = linspace(0,tend,25);
RD = rt.^1/2;
lt = linspace(0,tend,25);
LD = -lt.^2;
tspan = [1 tend];
ic = [0; 0; 0];
opts = odeset('RelTol',1e-2,'AbsTol',1e-4);
[t,y] = ode45(@(t,y) diffdrive(t,y,rt,RD,lt,LD), tspan, ic, opts);
% plot(y(:,1),y(:,2));grid
plot(y(:,1),y(:,2));grid
xlabel('Time (sec)','FontSize',14,'FontWeight','bold','Color','k');
ylabel('Signals','FontSize',14,'FontWeight','bold','Color','k');
title('Robot Pose Versus
Time','FontSize',16,'FontWeight','bold','Color','k');
```

Problem 2: Given a target pose $(x_t y_t ,q_t)$, find a feedback control law that maneuvers the robot from an initial starting pose (e.g., the origin) to the target pose. Implement the control law in your MATLAB simulation.

Problem 3: Given a target pose trajectory $(x_t(t), y_t(t), q_t(t)), 0 \le t \le T$, find a feedback control law that maneuvers the robot to intercept and follow the target pose trajectory. Implement the control law in your MATLAB simulation.

Problem 4: Now assume there are two robots. Simulate the maneuvers of both robots to meet and follow a simple (e.g., straight line) trajectory. The robots should form a formation with one as leader and one as follower.

Problem 5: Design a simulation in which *N* robots assemble into a linear formation and follow a target trajectory. You may find the papers [1] and [2] of interest.

[1] A. Becker, C. Onyuksel, and T. Bretl, “Feedback control of many differential-drive robots with uniform control inputs,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2256–2262, 2012.

[2] D. J. Balkcom and M. T. Mason, “Time optimal trajectories for bounded velocity differential drive vehicles,” *Int. J. Rob. Res.*, vol. 21, no. 3, pp. 199–217, 2002.