RBE 502 Homework 5 Extra Credit

Table of Contents

By Alex Tacescu

```
clc; clear;
% Initial and final condition (x, y, theta)
q0 = [10, 20, pi/4];
qf = [0, 0, pi/3];
vd = 0;
a_xd = 0;
a_yd = 0;
Tf = 15;    % time to reach destination
```

Dynamically Feasible Trajectory

```
syms t
a = sym('a', [1,4]); % the parameters of trajectory for x
b = sym('b', [1,4]); % the parameters of trajectory for y
basis = [1; t; t^2; t^3];
dbasis = [0; 1; 2*t; 3*t^2];
xsym = a*basis;
ysym = b*basis;
dx = a*dbasis;
dy = b*dbasis;

x0 = subs(xsym,t,0);    % EQ 1
y0 = subs(ysym,t,0);    % EQ 2
xf = subs(xsym,t,Tf);   % EQ 3
yf = subs(ysym,t,Tf);   % EQ 4

dx0 = subs(dx,t,0);
dxf = subs(dx,t,Tf);

dy0 = subs(dy,t,0);
dyf = subs(dy,t,Tf);

% Use jacobian linearization to linearize the system to solve for the
% constants. May not work if the change is too big
syms v w theta x y
f= [v*cos(theta); v*sin(theta); w];
dfdx = jacobian(f,  [x;y;theta]);
dfdu = jacobian(f,  [v;w]);

% Solve linear equations for finding the coefficients in the feasible
```

```
% trajectories, initial and terminal condition: with velocity equals
 zero
[matA,matB] = equationsToMatrix([
    x0==q0(1), ...
    y0==q0(2), ...
    dx0*sin(q0(3)) - dy0*cos(q0(3))==0, ...
    dx0*sin(q0(3)) + dy0*cos(q0(3))==0, ...
    xf==qf(1), ...
    yf==qf(2), ...
    dxf*sin(qf(3)) - dyf*cos(qf(3))==0, ...
    dxf*sin(qf(3)) + dyf*cos(qf(3))==0], ...
    [a(1),a(2),a(3),a(4),b(1),b(2),b(3),b(4)]);

param = matA\matB;
avec= double(param(1:4)');
bvec = double(param(5:end)');
```

Dynamically feasible trajectories

```
x_d = subs(xsym, a, avec)
y_d = subs(ysym, b, bvec)

v_xd = diff(x_d)
v_yd = diff(y_d)
```

```
x_d =

(4*t^3)/675 - (2*t^2)/15 + 10


y_d =

(8*t^3)/675 - (4*t^2)/15 + 20


v_xd =

(4*t^2)/225 - (4*t)/15


v_yd =

(8*t^2)/225 - (8*t)/15
```

Dubin's Car Model Dynamic System

```
Ax = [0, 1; 0, 0];
Bx = [0; 1];

kx = place(Ax, Bx, [-3; -5]);

Ay = [0, 1; 0, 0];
```

```
By = [0; 1];

ky = place(Ay, By, [-3; -5]);

syms v_x v_y
```

Linear Accelleration Dynamic System Equations:

```
a_x = -kx * [(x - x_d); (v_x - v_xd)] + a_xd

a_y = -ky * [(y - y_d); (v_y - v_yd)] + a_yd
```

```
a_x =

(4*t^3)/45 - (418*t^2)/225 - (32*t)/15 - 8*v_x - 15*x + 150


a_y =

(8*t^3)/45 - (836*t^2)/225 - (64*t)/15 - 8*v_y - 15*y + 300
```

Angular Acceleration Equations: Equation taken from notes

```
w = (a_y * cos(theta) - a_x * sin(theta)) / v
```

```
w =

-(cos(theta)*(- (8*t^3)/45 + (836*t^2)/225 + (64*t)/15 + 8*v_y + 15*y
 - 300) - sin(theta)*(- (4*t^3)/45 + (418*t^2)/225 + (32*t)/15 + 8*v_x
 + 15*x - 150))/v
```

Published with MATLAB® R2019b