Table of Contents


```
function [ dx ] = DublinTrajectoryTracking_nonlinear(t,x,param)
```

Implementation

```
        %Extracting the coefficients of the trajectory
        a1=param(1,:);
        a2=param(2,:);

        %Create the actual trajectory
        vec_t = [1; t; t^2; t^3];
        X_d= [a1*vec_t;a2*vec_t];  %position


        % compute the velocity and acceleration in both theta 1 and
 theta2.
        x_vel = [a1(2), 2*a1(3), 3*a1(4), 0];
        x_acc = [2*a1(3), 6*a1(4),0,0 ];
        y_vel = [a2(2), 2*a2(3), 3*a2(4), 0];
        y_acc = [2*a2(3), 6*a2(4),0,0 ];

        % compute the desired trajectory (assuming 3rd order
 polynomials for trajectories)
        dX_d =[x_vel*vec_t; y_vel* vec_t] ; %Velocity
        ddX_d =[x_acc*vec_t; y_acc* vec_t];  %Acceleration
        X= x(1:2,1);

        % Calculating the state variables
        th=x(3);
        v=x(4);
        dX=[v*cos(th);v*sin(th)];
```

```
Not enough input arguments.

Error in DublinTrajectoryTracking_nonlinear (line 5)
        a1=param(1,:);
```

PD controller

```
        KP=5;
        KD=10;
        K=[KP*eye(2), KD*eye(2)];
```

```
U=-K*[X-X_d;dX-dX_d]+ddX_d;


a=U(1)*cos(th)+ U(2)*sin(th);
w=(U(2)*cos(th)-U(1)*sin(th))/v;

A=[0 0 1 0;0 0 0 1;0 0 0 0;0 0 0 0];
B=[0 0;0 0;1 0;0 1];
z=[X;dX];
```

Calculate dx for the dubins car

```
dx=[dX;w;a];
```

```
end
```

Published with MATLAB® R2018b