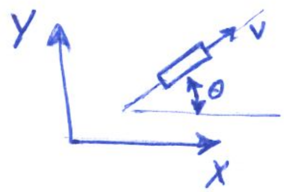
1) $\dot{x} = v\cos\theta$

$\dot{y} = v\sin\theta$

$\dot{\theta} = \omega$

$$\rightarrow \dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix}$$



dynamical wrt deviation

$$e_x = \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \end{bmatrix}, \quad e_u = \begin{bmatrix} v - v_d \\ \omega - \omega_d \end{bmatrix}$$

$v_d = v_r \qquad v_r = 10 \qquad v_d = 10$

$x_d = v_r t \qquad y_r = 2 \qquad x_d = 10t$

$y_d = y_r \qquad\qquad y_d = 2$

$$\left[\begin{matrix} \theta_d = 0 \\ \omega_d = 0 \end{matrix}\right. \qquad \theta_d = 0 \qquad \omega_d = 0$$

$$X_d = \begin{bmatrix} v_r t \\ y_r \\ 0 \end{bmatrix}, \quad X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

$$u_d = \begin{bmatrix} v_r \\ \omega_d \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$\dot{e}_x = \dot{X} - \dot{X}_d = f(e_x + X_d, e_u + u_d) - f(X_d, u_d) \approx f(X_d, u_d) + \frac{\partial f}{\partial X}\Big|_{\substack{X_d \\ u_d}} e_x + \frac{\partial f}{\partial u}\Big|_{\substack{X_d \\ u_d}} e_u - \cdots - f(X_d, u_d)$$

$$\dot{e}_x \approx \underbrace{\frac{\partial f}{\partial X}\Big|_{X_d, u_d}}_{A(t)} e_x + \underbrace{\frac{\partial f}{\partial u}\Big|_{X_d, u_d}}_{B(t)} e_u$$

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} = f(X, u)$$

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v\sin\theta \\ 0 & 0 & +v\cos\theta \\ 0 & 0 & 0 \end{bmatrix}\Bigg|_{X_d, u_d} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 10 \\ 0 & 0 & 0 \end{bmatrix}}_{A(t)}$$

$$\frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}\Bigg|_{u_d, X_d} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{B(t)}$$

$$\dot{e}_X = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 10 \\ 0 & 0 & 0 \end{bmatrix} e_X + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} e_u, \text{ where, } e_X = X - X_d, \; e_u = u - u_d$$

$$e_X = \begin{bmatrix} X - 10t \\ Y - 2 \\ \theta - 0 \end{bmatrix}, \quad e_u = \begin{bmatrix} V - 10 \\ \omega - 0 \end{bmatrix}$$

using Pole placement techique we find the desired K (gain) which places the poles in a stable LHS of the imaginary axis. K is calculated as : K=[5000, 200, 300; 1000, 200, 300]

Also, by using the reachability matrix we see that the system is full rank and Controllable.

$$\text{rank}([B \;\; AB]) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \text{rank} = 3$$

$u = u_d + e_u, \; e_u = -Ke_X$

$$u = -K e_X + u_d \rightarrow u = -\begin{bmatrix} 5000 & 200 & 300 \\ 1000 & 200 & 300 \end{bmatrix} \begin{bmatrix} X - X_d \\ Y - Y_d \\ \theta - \theta_d \end{bmatrix} + \begin{bmatrix} V_d \\ \omega_d \end{bmatrix}$$

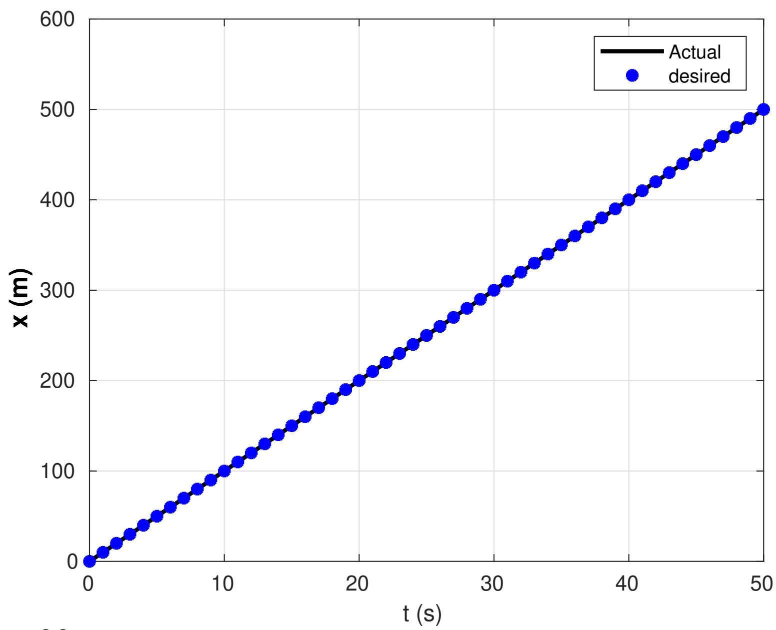
$$u = \begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

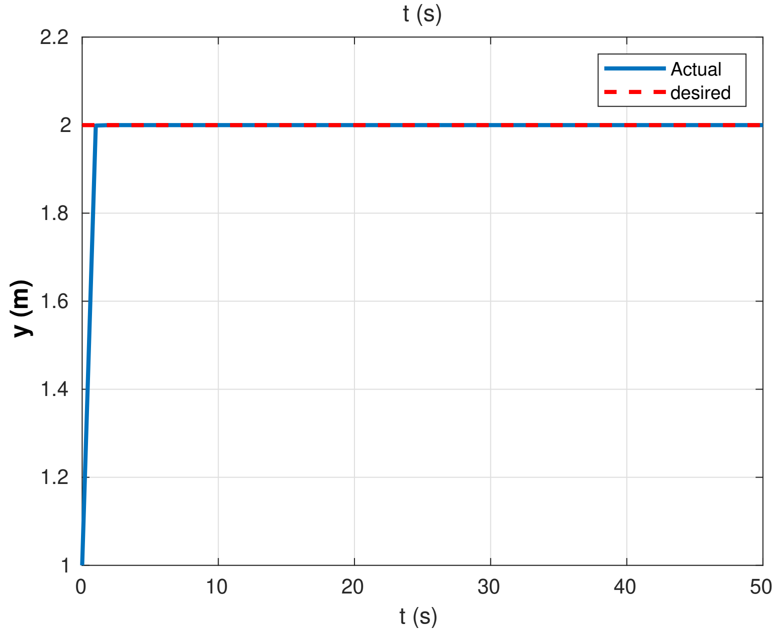
$$\underbrace{u = -\begin{bmatrix} 5000(X - 10t) + 200(Y - 2) + 300(\theta) \\ 1000(X - 10t) + 200(Y - 2) + 300(\theta) \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \end{bmatrix}}_{\text{Feedback Controller}}$$

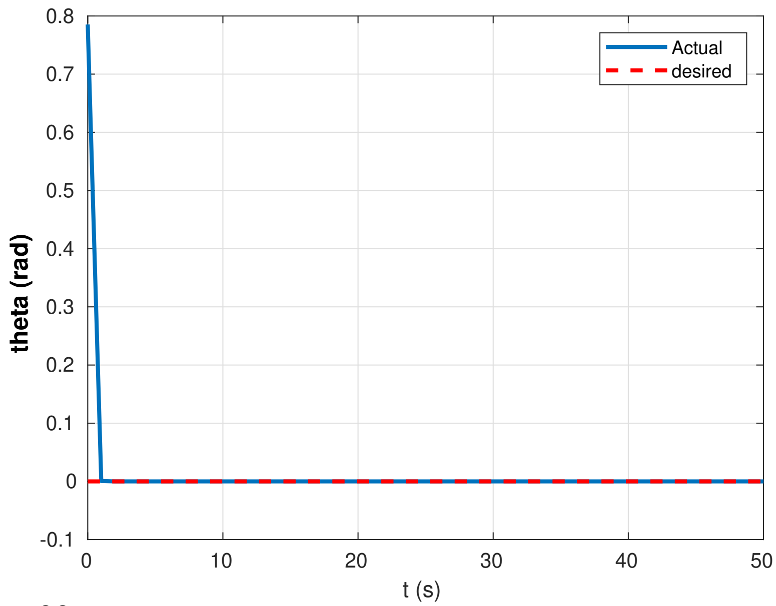
Farid Tavakkolmoghaddam HW4

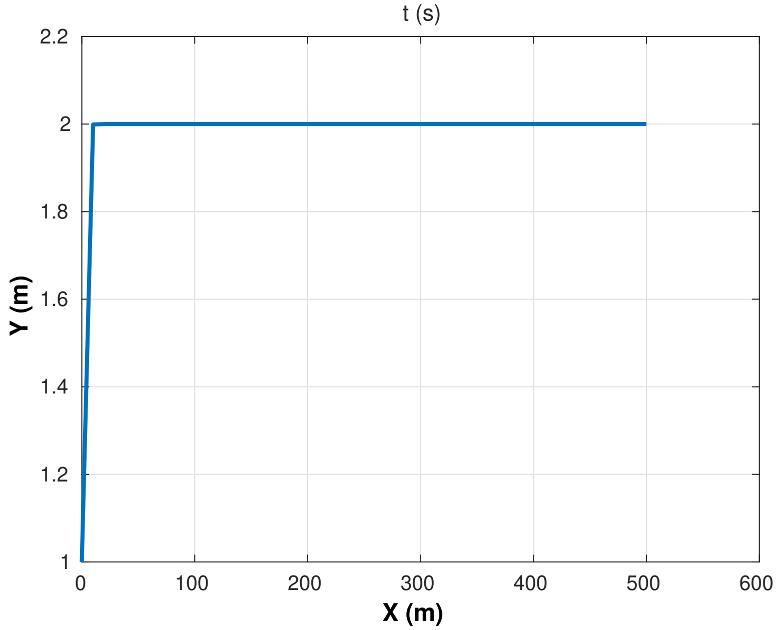
```
%Steering control
clear; close all; clc
T=50;
x0 =[0;1;pi/4]; % Feel free to change the initial state and sampling horizon.



% design steering control to follow a straight lane while maintaining a given velocity
% the lateral position yr= 2;



%TODO: param is the additional parameter to pass to the ode function.
[T,X] = ode45(@(t,x) ode_dubins(t,x), (0:T), x0);
tt=0:1:50;
XX=10*tt; % desired trajectory x
YY=2*ones(1,51); % desired trajectory y
thetta=2*zeros(1,51); % desired trajectory y

% plot your state trajectories for both 1 and 2, using the following code or else.
figure
title(' X position vs. time')
plot(T,X(:,1),'k',tt,XX,'b*','LineWidth',2);
xlabel('t (s)');
ylabel('x (m)','FontSize',12,'FontWeight','bold','Color','k');
legend('Actual','desired')
grid

figure
title(' Y position vs. time')
plot(T,X(:,2),tt,YY,'r--','LineWidth',2);
xlabel('t (s)');
ylabel('y (m)','FontSize',12,'FontWeight','bold','Color','k');
legend('Actual','desired')
grid

figure
title(' Orientation vs. time')
plot(T,X(:,3),tt,thetta,'r--','LineWidth',2);
xlabel('t (s)');
ylabel('theta (rad)','FontSize',12,'FontWeight','bold','Color','k');
legend('Actual','desired')
grid
```

```
figure ('name','X v.s y')
title(' x vs. y')
plot(X(:,1), X(:,2),'LineWidth',2)
xlabel('X (m)','FontSize',12,'FontWeight','bold','Color','k');
ylabel('Y (m)','FontSize',12,'FontWeight','bold','Color','k');
grid
```









1 ODE Dubins

article graphicx color

Contents

- TODO: Here is the code for control input.

- the controller needs to provide control input: linear velocity and
- steering angle: v, delta

```
function dz = ode_dubins(t,z)

% use z for [x,y,theta]
dz =zeros(3,1);
```

TODO: Here is the code for control input.

the controller needs to provide control input: linear velocity and

steering angle: v, delta

```
% step 1: calculate the value of deviation variables given the
% desired state and input and the actual state, exacted from z.
v_d = 10;
w_d = 0;
u_d = [v_d ; w_d];

y_d = 2;
theta_d = 0;
v_r = 10;
x_d = [v_r*t; y_d ;theta_d];
e_x = z - x_d;

% de_x = dx - dx_d;

% step 2: based on the feedback controller, calculate the e_u=
% [e_v, e_w].
k=[5000 200 300;
    1000 200 300];
e_u = -k*e_x;

% step 3: based on the relation between u and e_u, derive the
% desired input v, and w.
% e_u= u - u_d; --> u = e_u + u_d
u= -k*e_x + u_d;
v= u(1);
w= u(2);
theta= z(3);

dz(1) = v*cos(theta);
dz(2) = v*sin(theta);
dz(3) = w;
```

```
Not enough input arguments.

Error in ode_dubins (line 18)
x_d = [v_r*t; y_d ;theta_d];
```

```
end
```