TRAJECTORYGENERATIONRRR

Table of Contents

Função retorna os valores de posição, velocidade e aceleração de cada junta rotacional do robô no decorrer do movimento, recebendo os valores de posição que deseja movimentar o robô, deslocamento da ferramenta ao punho, da base em relação a mesa, além do tempo de cada intervalo e o tempo de discretização.

Calling Syntax

[thpath]=trajectorygenerationrrr(uform_vec, trelw, srelb, T, Ts)

I/O Variables

IN 1 Double Matrix **uform_vec**: Three-dimensional matrix Nx3x3 with position, velocity and acceleration vectors of the 3 joints N = number of trajectory points

IN 2 Double Array **trelw**: User form [x y theta] [meters meters degrees]

IN 3 Double Array **srelb**: User form [x y theta] [meters meters degrees]

IN 4 Double **T**: Time of each segment

IN 5 Double **Ts**: Discretization resolution time for each segment

OUT 1 Double Matrix **thpath**:

Example

```
uform_vec = [0.758 0.173 0.0;
             0.6  -0.3  45.0;
             -0.4 0.3  120.0;
             0.758 0.173 0.0];
```

trelw = [0.1 0.2 30.0]; srelb = [0.0 0.0 0.0];

T = 3; Ts = 0.01;

thpath = trajectorygenerationrrr(uform_vec, trelw, srelb, T, Ts);

Hypothesis

A trajetória deve conter pelomenos 2 posições

Ts deve ser algumas vezes menor que T para que ocorra uma discretização

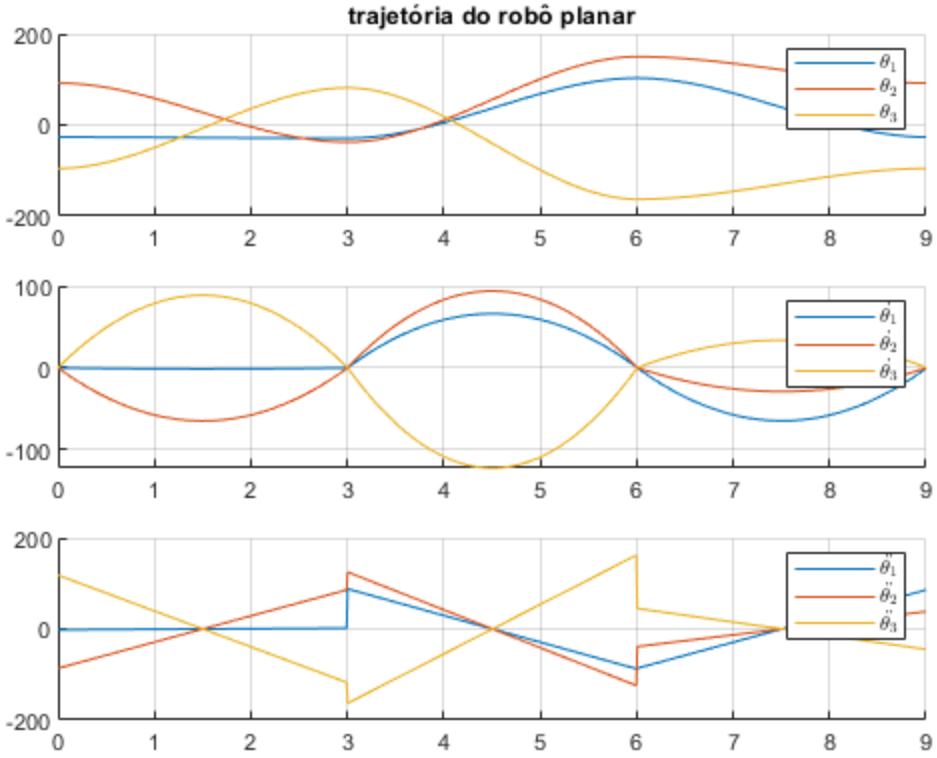
Limitations

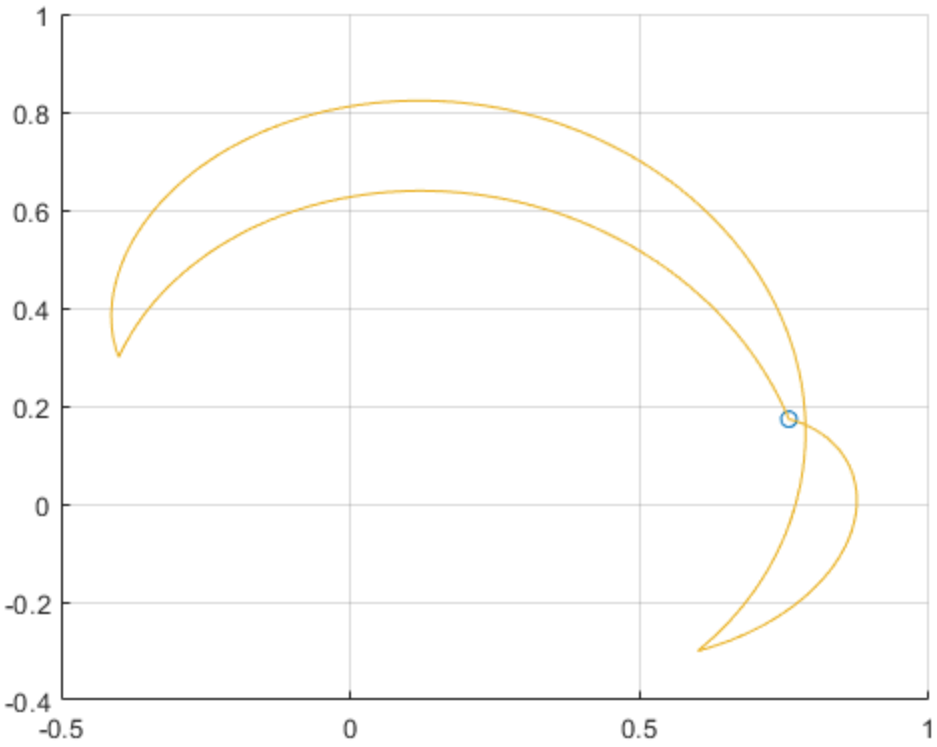
T deve ser divisível por Ts

Function

```
function [thpath]=trajectorygenerationrrr(uform_vec, trelw, srelb, T, Ts)

L = [0.5 0.3];
thetalim = [-170 170;-170 170;-170 170];
srelb = utoi(srelb);
trelw = utoi(trelw);
start_point = kin([0 0 0],L);
goal = utoi(uform_vec(1,:));
[near,far,sol] = solve_robot(goal,start_point,trelw,srelb,L,thetalim);
traj_points = near;

for a = 1:length(uform_vec(:,1))-1
    current = utoi(uform_vec(a,:));
    goal = utoi(uform_vec(a+1,:));
    [near,far,sol] = solve_robot(goal,current,trelw,srelb,L,thetalim);
    traj_points = [traj_points;near];
end

thpath = zeros((T/Ts)*(length(traj_points(:,1))-1),3,3);

for b = 1:3

    thpath(:,:,b)=trajectorygeneration(traj_points(:,b), T, Ts);
end

x = linspace(0,(length(traj_points(:,1))-1)*T, (length(traj_points(:,1))-1)*T/
Ts);


subplot(3,1,1);
title("trajetória do robô planar");
hold on
plot(x,thpath(:,1,1))
plot(x,thpath(:,1,2))
plot(x,thpath(:,1,3))
hold off
legend(["$\theta_{1}$","$\theta_{2}$","$\theta_{3}$"],'Interpreter','latex');
grid on

subplot(3,1,2);
hold on
plot(x,thpath(:,2,1))
plot(x,thpath(:,2,2))
plot(x,thpath(:,2,3))
hold off
```

```
legend(["$\dot{\theta_{1}}$","$\dot{\theta_{2}}$","$
\dot{\theta_{3}}$"],'Interpreter','latex');
grid on

subplot(3,1,3);
hold on
plot(x,thpath(:,3,1))
plot(x,thpath(:,3,2))
plot(x,thpath(:,3,3))
hold off
legend(["$\ddot{\theta_{1}}$","$\ddot{\theta_{2}}$","$
\ddot{\theta_{3}}$"],'Interpreter','latex');
grid on

pontapath = zeros(length(thpath(:,1,1)),3);
for c=1:length(thpath(:,1,1))
    ponta = where_robot(thpath(c,1,:),trelw,srelb,L);
    pontapath(c,:) = itou(ponta);
end

figure
hold on
grid on
comet(pontapath(:,1),pontapath(:,2))

end
```





Published with MATLAB® R2022b