
TRAJECTORYGENERATIONRRR

Table of Contents

Calling Syntax	1
I/O Variables	1
Example	1
Hypothesis	1
Limitations	2
Version Control	2
Group Members	2
Function	2
Validity	2
Main Calculations	2
Output Data	3

Plota a posição, a velocidade e a aceleração em termos da forma Cartesiana do usuário, dadas as definições conforme os sistemas $\{T\}$ e $\{S\}$ usadas anteriormente na disciplina

Calling Syntax

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

I/O Variables

IN Double Matrix **uform_vec**: *User form vectors* Posições inicial e intermediárias da ferramenta até o goal

IN Double Array **trelw**: *T relativo a W* Array 1x3 da transformação homogênea

IN Double Array **srelb**: *S relativo a B* Array 1x3 da transformação homogênea

IN Double **T**: tempo de duração de cada segmento (seg)

IN Double **Ts**: é a taxa de atualização do caminho (seg)

OU Double Array **thpath**: plote da trajetória completa no espaço de juntas

Example

```
uform_vec = [.758 .173 0; .6 -.3 45; -.4 .3 120; .758 .173 0]; trelw = [.1 .2 30]; srelb = [0 0 0]; T = 3; Ts = 1/40;
```

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

Hypothesis

RRR planar robot.

Limitations

A matriz de transformação homogênea precisa seguir a sintaxe de classe e não tem validade para qualquer configuração de robô.

Version Control

1.0; Grupo 04; 2025/06/19 ; Second issue.

Group Members

- Guilherme Fortunato Miranda

13683786

- João Pedro Dionizio Calazans

13673086

Function

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

Validity

Not apply

Main Calculations

```
L = [.5 .3];
thetalim = [170 170 170; -170 -170 -170];
segm = length(uform_vec(:,1))-1;
trelw = utoi(trelw); srelb = utoi(srelb);
t = 0:Ts:T*(segm); t = t';

uform_vec = cat(1, [0 0 0], uform_vec);
traji = zeros(segm, 3); % segmento x junta
thpath = zeros(length(t), 3, 3); % tempo x traj. x junta

for i = 1:3
    for k = 1:segm+1
        % 1 -> 2 -> 3 -> 4
        % 3 juntas
        % 3 posições e 3 segmentos
        current = uform_vec(k,:);
        goal = uform_vec(k+1,:);
        [near,~,~]=solve_robot(goal,current,trelw,srelb,L,thetalim);
        traji(k,:) = near;
    end
end
```

```

    thpath(:, :, i) = trajectorygeneration(traji(:, i), T, Ts, false);
end
%transformando em coordenadas cartesianas
%posicoes
x = zeros(length(thpath(:, 1, 1)), 1);
y = zeros(length(thpath(:, 1, 1)), 1);
phi = zeros(length(thpath(:, 1, 1)), 1);
%velocidades
x_dot = zeros(length(thpath(:, 1, 1)), 1);
y_dot = zeros(length(thpath(:, 1, 1)), 1);
phi_dot = zeros(length(thpath(:, 1, 1)), 1);
%acelerações
x_ddot = zeros(length(thpath(:, 1, 1)), 1);
y_ddot = zeros(length(thpath(:, 1, 1)), 1);
phi_ddot = zeros(length(thpath(:, 1, 1)), 1);
L = [.5 .3 0.1155];
for i=1:length(thpath(:, 1, 1))
    thetas_temp = squeeze(thpath(i, 1, :));
    dthetas_temp = squeeze(thpath(i, 2, :));
    ddthetas_temp = squeeze(thpath(i, 3, :));
    xyphi = itou([kin(thpath(i, 1, :), L)*trelw]);
    x(i) = xyphi(1);
    y(i) = xyphi(2);
    phi(i) = xyphi(3);

    dxyphi = jacobian_tool(thpath(i, 1, :), L) * dthetas_temp;
    x_dot(i) = dxyphi(1);
    y_dot(i) = dxyphi(2);
    phi_dot(i) = dxyphi(3);

    ddxphi = jacobian_acc(ddthetas_temp, dthetas_temp, thetas_temp, L);
    x_ddot(i) = ddxphi(1);
    y_ddot(i) = ddxphi(2);
    phi_ddot(i) = ddxphi(3);
end

```

Output Data

```

figure
plot(t, thpath(:, 1, 1), 'b')
hold on
plot(t, thpath(:, 1, 2), 'r')
hold on
plot(t, thpath(:, 1, 3), 'y')
xlabel('Tempo [s]');
ylabel('Posição Angular de Junta [°]');
legend('$\theta_1$', '$\theta_2$', '$\theta_3$', 'Interpreter', 'latex')
axis([-inf inf -inf inf]);
grid on

figure
plot(t, thpath(:, 2, 1), 'b')
hold on

```

```

plot(t, thpath(:,2,2), 'r')
hold on
plot(t, thpath(:,2,3), 'y')
xlabel('Tempo [s]');
ylabel('Velocidade Angular de Junta [°/s]');
legend('$\dot{\theta}_1$', '$\dot{\theta}_2$', '$\dot{\theta}_3$',
'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

figure
plot(t, thpath(:,3,1), 'b')
hold on
plot(t, thpath(:,3,2), 'r')
hold on
plot(t, thpath(:,3,3), 'y')
xlabel('Tempo [s]');
ylabel('Aceleração Angular de Junta [°/s^2]');
legend('$\ddot{\theta}_1$', '$\ddot{\theta}_2$', '$\ddot{\theta}_3$',
'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

figure;
subplot(2,1,1);
plot(t, x, 'b');
hold on
plot(t, y, 'r');
xlabel('Tempo [s]');
title('Posição cartesiana [m]');
legend('$x$', '$y$', 'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

subplot(2,1,2);
plot(t, phi, 'y');
hold on
xlabel('Tempo [s]');
title('Orientação da ferramenta [°]');
legend('$\phi$', 'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

figure;
subplot(2,1,1);
plot(t, x_dot, 'b');
hold on
plot(t, y_dot, 'r');
xlabel('Tempo [s]');
title('velocidade cartesiana [m/s]');
legend('$\dot{x}$', '$\dot{y}$', 'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

```

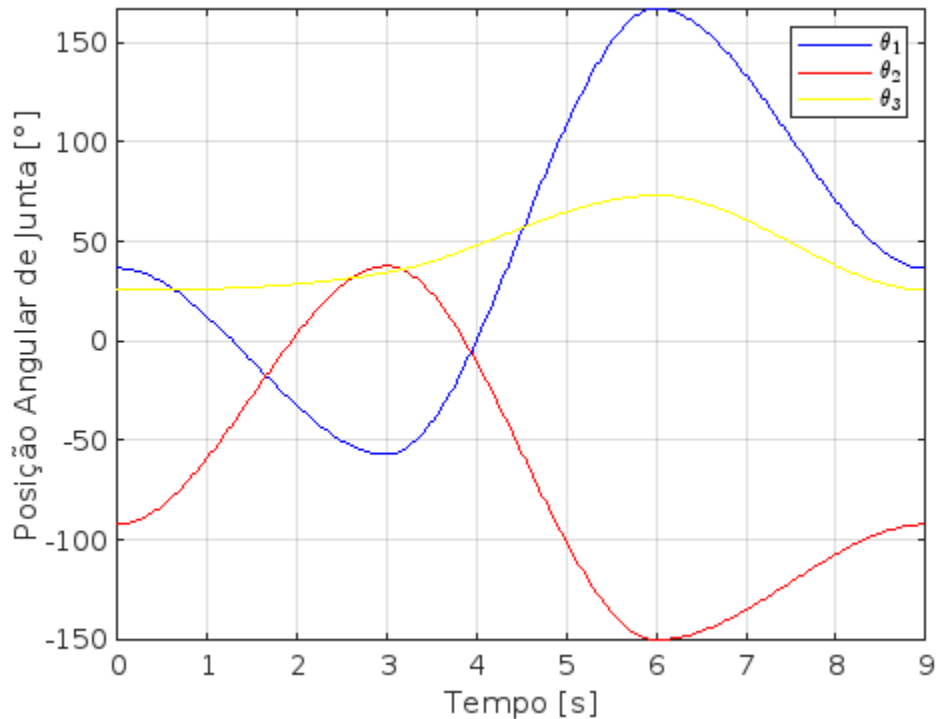
```

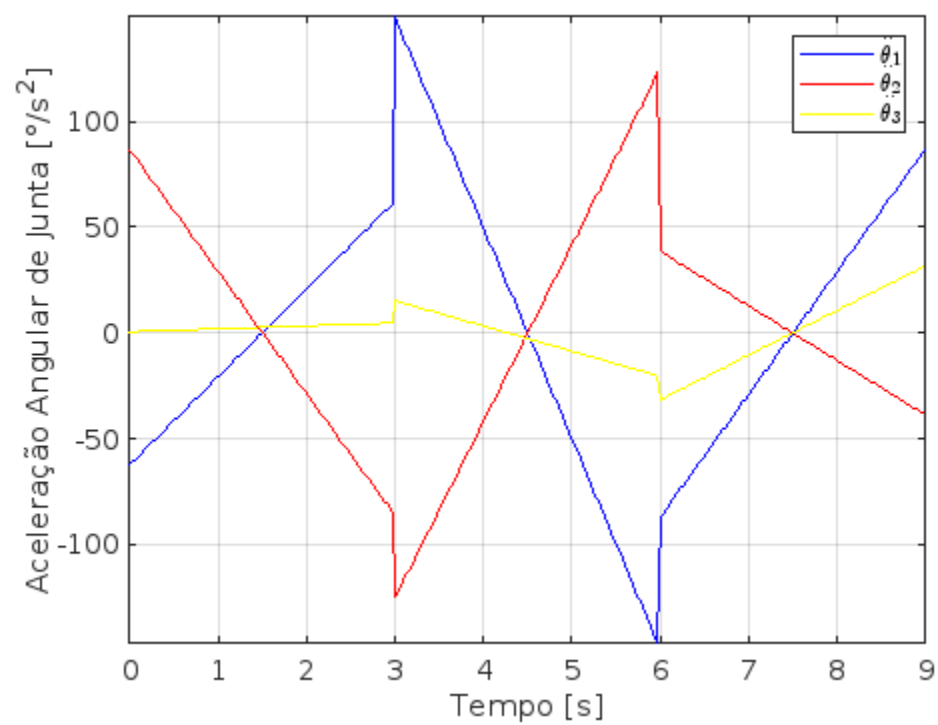
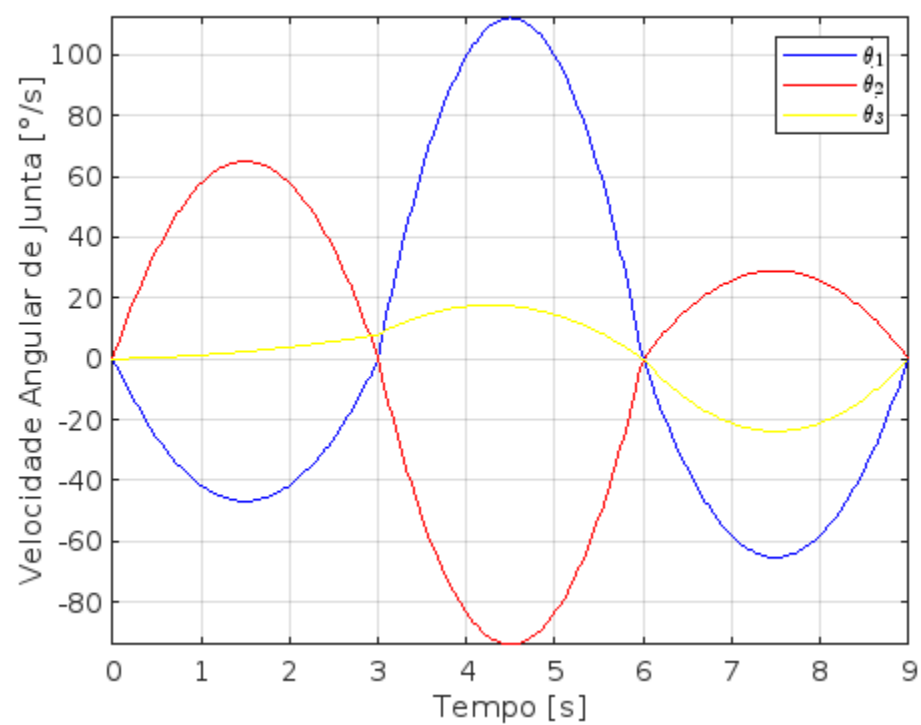
subplot(2,1,2);
plot(t, phi_dot, 'y');
hold on
xlabel('Tempo [s]');
title('Velocidade angular da orientação da ferramenta [°/s]');
legend('$\dot{\phi}$', 'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

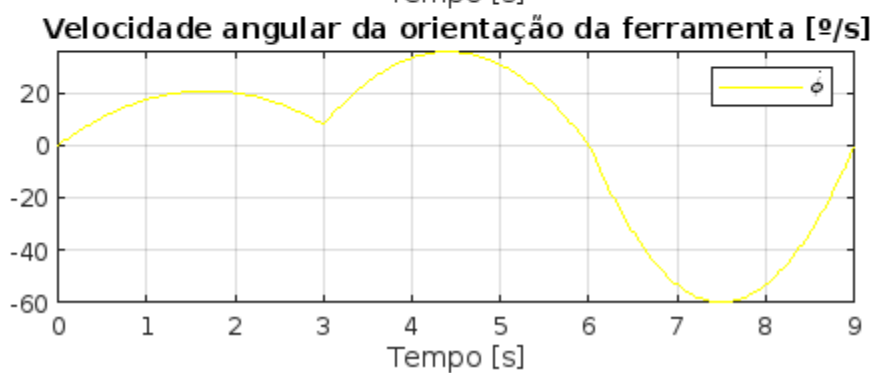
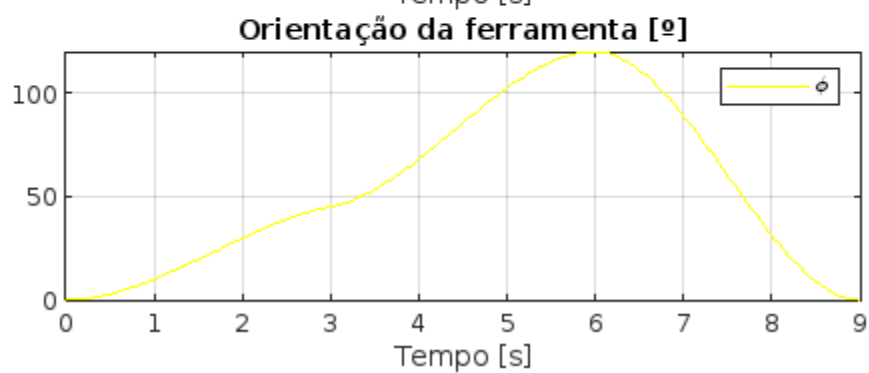
figure;
subplot(2,1,1);
plot(t, x_ddot, 'b');
hold on
plot(t, y_ddot, 'r');
xlabel('Tempo [s]');
title('Aceleração cartesiana [m/s^2]');
legend('$\ddot{x}$', '$\ddot{y}$', 'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

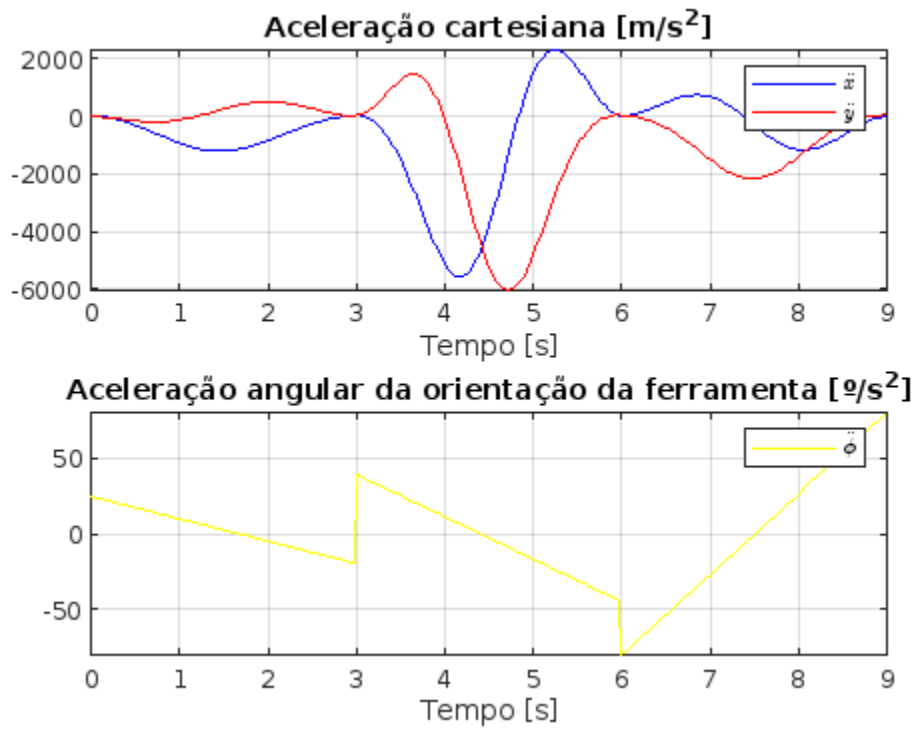
subplot(2,1,2);
plot(t, phi_ddot, 'y');
hold on
xlabel('Tempo [s]');
title('Aceleração angular da orientação da ferramenta [°/s^2]');
legend('$\ddot{\phi}$', 'Interpreter', 'latex')
axis ([-inf inf -inf inf]);
grid on

```









end

Published with MATLAB® R2024b