
TRAJECTORYGENERATION

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	2

Soluciona o problema de determinação dos coeficientes do polinômio cúbico utilizando a abordagem heurística para a atribuição das velocidades da junta nos pontos intermediários conforme a opção 2 na Seção 7.3 do Craig

Calling Syntax

```
[thpathi]=trajectorygeneration(traj_points, T, Ts, plt)
```

I/O Variables

IN Double Array **traj_points**: é um vetor contendo o ponto inicial, os intermediários e o final da trajetória desejada.

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

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

IN Bool **plt**: indica se é solicitado o plot da trajetória para a junta i

OU Double Array **thpathi**: θ path é um vetor de valores de posição, velocidade e aceleração de junta do manipulador (neste caso, ainda é uma junta desconhecida i qualquer)

Example

```
traj_points = [5 10 30 15];  
T = 3;  
Ts = 1/40;  
[thpathi]=trajectorygeneration(traj_points, T, Ts, true);
```

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; 2025/06/06 ; First issue.

Group Members

- Guilherme Fortunato Miranda

13683786

- João Pedro Dionizio Calazans

13673086

Function

```
function [thpathi]=trajectorygeneration(traj_points, T, Ts, plt)
```

Validity

Not apply

Main Calculations

```
cctot = trajectoryplanning(traj_points,T);
segm = length(traj_points) - 1; % # of segments
N = T/Ts; % # of points per segment
t = 0:Ts:T*segm; t = t';
thddotd = zeros(length(t),1);
thdotd = thddotd; thd = thddotd;

for k = 1:segm
    % tk = t(N*(k-1)+1:N*k+1);
    tk = t(1:N+1); % travado nos N primeiros valores
    % para a traj., concatena por cada segmento:
    thddotd(N*(k-1)+1:N*k+1) = 2*cctot(3,k) + 6*cctot(4,k).*tk;
    thdotd(N*(k-1)+1:N*k+1) = cctot(2,k) + 2*cctot(3,k).*tk +
3*cctot(4,k).*tk.^2;
    thd(N*(k-1)+1:N*k+1) = cctot(1,k) + cctot(2,k).*tk +
cctot(3,k).*tk.^2 + cctot(4,k).*tk.^3;
end
```

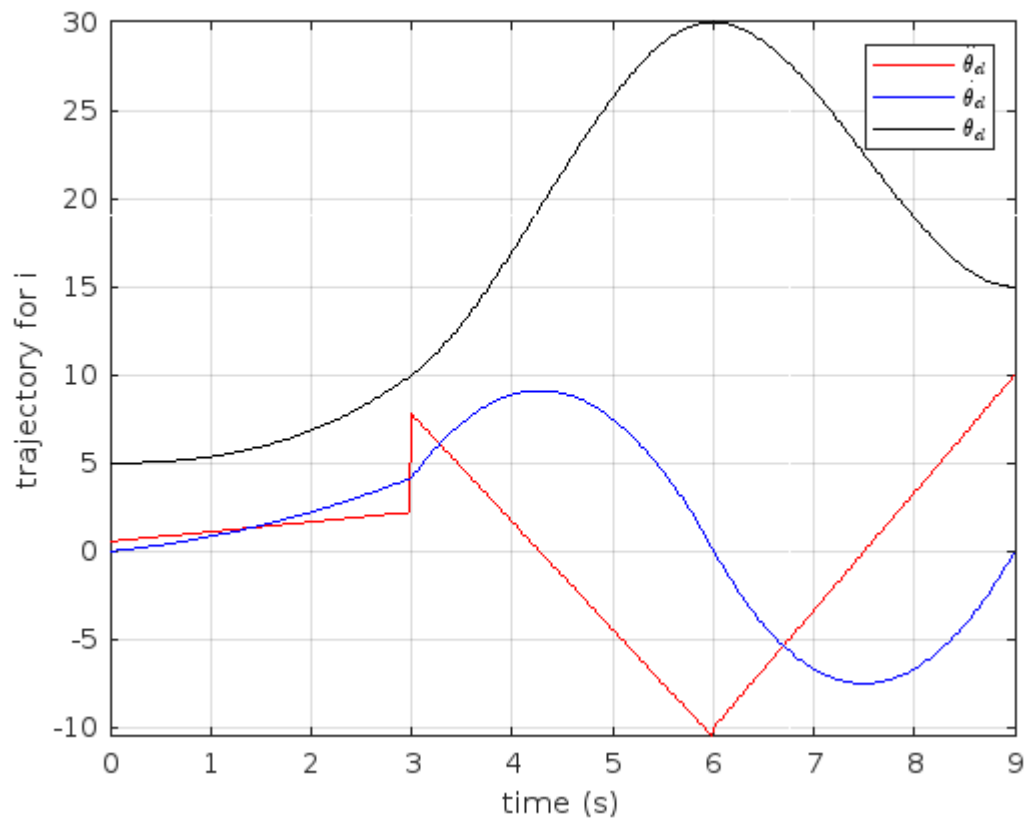
Output Data

```
thpathi = [thd, thdotd, thddotd];
```

```

if plt
    figure
    plot(t, thpathi(:,3), 'r')
    hold on
    plot(t, thpathi(:,2), 'b')
    hold on
    plot(t, thpathi(:,1), 'k')
    xlabel('time (s)');
    ylabel('trajectory for i');
    axis ([-inf inf -inf inf]);
    legend('$\ddot{\theta}_d$', '$\dot{\theta}_d$', '$\theta_d$',
'Interpreter', 'latex')
    grid on
end

```



end

Published with MATLAB® R2024b