

---

# TRAJECTORYPLANNING

## Table of Contents

Calling Syntax .....	1
I/O Variables .....	1
Example .....	1
Hypothesis .....	1
Limitations .....	1
Function .....	2

Calcula os coeficientes de equações cúbicas para todos segmentos da trajetória de uma junta a partir dos valores de ângulo entre cada intervalo e de sua duração.

## Calling Syntax

```
cctot = trajectoryplanning(traj_points,T);
```

## I/O Variables

```
|IN 1 Double Array| *traj_points*: trajectory points of a joint  
[theta_t0 theta_t1 ...] [deg deg ...]  
  
|IN 2 Double| *T*: trajectory segment duration [seconds]  
  
|OUT 1 Double Matrix| *cctot*: cubic equation coefficients for each  
segment  
[a0_t0 a1_t0 a2_t0 a3_t0;  
 a0_t1 a1_t1 a2_t1 a3_t1;  
 ...]
```

## Example

```
traj_points = [0 30 15 0];  
T = 3;  
  
cctot = trajectoryplanning(traj_points,T);
```

## Hypothesis

Calcula polinômios da trajetória de juntas rotacionais. Instantes iniciais e finais com velocidade nula.

## Limitations

Somente para intervalos T constantes. Velocidades entre segmentos de direção de movimento diferentes são nulas e para segmentos de mesma

direção de movimentos são médias aritméticas das velocidades adjacentes.

## Function

```
function [cctot]=trajectoryplanning(traj_points,T)

traj_vel = zeros([length(traj_points),1]);
cctot = zeros([length(traj_points)-1,4]);

for a = 2:length(traj_points)-1
    dif1 = traj_points(a)-traj_points(a-1);
    dif2 = traj_points(a+1)-traj_points(a);
    if dif1*(-dif2) >= 0
        traj_vel(a) = 0;
    else
        traj_vel(a) = (dif1+dif2)/(2*T);
    end
    cctot(a-1,:) =
        cubcoef(traj_points(a-1),traj_vel(a-1),traj_points(a),traj_vel(a),T);
end
cctot(end,:) =
    cubcoef(traj_points(end-1),traj_vel(end-1),traj_points(end),traj_vel(end),T);

end

ans =

         0         0    10.0000    -2.2222
    30.0000         0    -3.3333     0.5556
    15.0000    -5.0000    -1.6667     0.5556
```

*Published with MATLAB® R2019b*