
JACOBIAN

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

Propagação das velocidades para obtenção do Jacobiano conforme exemplo 5.3 do Craig (até junta 2) e terminada conforme 'jacobian_prop.jpg'. Para o robo planar 3R, conforme vetor de ângulos de juntas e comprimento dos ligamentos informados, calcula o Jacobiano em referência ao sistema do punho, {3}.

Calling Syntax

[jac]=jacobian(theta,L)

I/O Variables

IN Double Array **theta**: *Joint angles* [$\theta_1 \theta_2 \theta_3$] [degrees degrees degrees]

IN Double Array **L**: *Ligaments length* [$L_1 L_2$] [meters meters]

OU Double Matrix **jac**: *sistem {3} jacobian* 6x3 Matrix

Example

```
theta = [0 90 -90]
```

```
L = [0.5 0.3]
```

```
[jac]=jacobian(theta,L)
```

Hypothesis

RRR planar robot.

Limitations

A "Forma do usuário" é específica para o exercício de simulação e não tem validade para qualquer configuração de robô.

Version Control

1.0; Grupo 04; 2025/27/05 ; First issue.

Group Members

- Guilherme Fortunato Miranda

13683786

- João Pedro Dionizio Calazans

13673086

Function

```
function [jac]=jacobian(theta,L)
```

Validity

Not apply

Main Calculations

```
jac = zeros(6,3);
```

Output Data

```
referenciando o J(3) como precisamos para o ex 4 (3 juntas)

jac(1,1:3) = [L(1)*sind(sum(theta(2:3)))+L(2)*sind(theta(3))
+L(2)*sind(theta(3)) 0];
jac(2,1:3) = [L(1)*cosd(sum(theta(2:3)))+L(2)*cosd(theta(3))
L(2)*cosd(theta(3)) 0];
jac(6,1:3) = [1 1 1];

%      referenciando o J(3) como no exercicio 5.11 original
%jac(1,1:3) = [L(1)*sind(theta(2)) 0 0];
%jac(2,1:3) = [L(1)*cosd(theta(2))+L(2) L(2) 0];
%jac(6,1:3) = [1 1 1];

end
```

```
jac =
```

-0.3000	-0.3000	0
0.5000	0	0
0	0	0
0	0	0
0	0	0
1.0000	1.0000	1.0000

Published with MATLAB® R2024b