

---

# EX3

## Table of Contents

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

Calcula as velocidades linear e angular da ponta da ferramenta com relação à própria ferramenta, isto é,  ${}^T\nu_T$ .  
Calcula todas as possíveis respostas, se existir mais de uma. Caso não, retorna uma matriz vazia (NaN).

## Calling Syntax

ex3

## I/O Variables

OU Double List **vrelt**: *Results* List the two 6x1 result arrays of the possible angles, based on the current e desired positions.

## Example

ex3

## 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

# Validity

Not apply

# Main Calculations

```
trelw = utoi([0.1 0.2 30]); % trelw
srelb = utoi([0. 0. 0.]); % srelb
trels = [0.6 -0.3 45]; % goal
delthetas = [20.; -10.; 12.]; % graus/seg
delthetas = delthetas*pi/180; % rad/seg
L = [0.5 0.3];
thetalim = [170 170 170; -170 -170 -170];
% conforme enunciado |
% do ex 4             |
%                     V
current = [10. 20. 30.];
[near,far,sol] = solve_robot(trels,current,trelw,srelb,L,thetalim);
```

# Output Data

```
vrelt = NaN(6, 1, 2);
if sol == 1
    %jacobian(near, L); returns 6x3
    Jw=jacobian(near, L);
    %delthetas is (3x1)
    vrelw = Jw*delthetas;
    % vrelw should be 6x1
    vrelt(:, :, 1)=veltrans(vrelw,trelw);
    Jw=jacobian(far, L);
    vrelw = Jw*delthetas;
    vrelt(:, :, 2)=veltrans(vrelw,trelw);
    disp(vrelt)
end
```

(:,:,1) =

```
0.1706
0.0579
0
0
```

0  
0.3840

(:,:,2) =

0.1692  
0.1000  
0  
0  
0  
0.3840

*Published with MATLAB® R2024b*