
SOLVE_ROBOT

Table of Contents

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

Calcula a cinemática inversa do robô planar RRR tendo como entradas o objetivo do sistema da ferramenta em relação ao sistema da estação no formato do usuário, a posição atual do robô (ângulos das juntas), a descrição do sistema da ferramenta em relação ao sistema do punho, a descrição do sistema da estação em relação ao sistema da base, os comprimentos dos ligamentos e os limites de operação das juntas do robô.

Calling Syntax

[near, far, sol] = solve_robot(goal, current, trelw, srelb, L, thetalim)

I/O Variables

IN Double Array **goal**: User form [x y theta] [meters meters degrees]

IN Double Array **current**: [theta1 theta2 theta3] [degrees degrees degrees]

IN Double Matrix **trelw**: Homogeneous Transformation Matrix 4x4

IN Double Matrix **srelb**: Homogeneous Transformation Matrix 4x4

IN Double Array **L**: [l1 l2] [meters meters]

IN Double Matrix **thetalim**: [lim1n lim1p; lim2n lim2p; lim3n lim3p] [degrees degrees; degrees degrees; degrees degrees]

OU Double Array **near**: [theta1 theta2 theta3] [degrees degrees degrees]

OU Double Array **far**: [theta1 theta2 theta3] [degrees degrees degrees]

OU Double **sol**: Number of solutions

Example

```
goal = [0 0 -90]
current = [0 0 0]
trelw = [0.8660    -0.5000    0    0.1000;
         0.5000    0.8660    0    0.2000;
```

```
0      0      1.0000      0;
0      0      0      1.0000]
srelb = [1.0000      0      0      -0.1000;
0      1.0000      0      0.3000;
0      0      1.0000      0;
0      0      0      1.0000]
L = [0.5 0.3]
thetalim = [ -170 170; -170 170; -170 170]
[near, far, sol] = solve_robot(goal, current, trelw, srelb, L,
    thetalim)

near =

148.1060 -100.2537 -167.8530

far =

148.1060 -100.2537 -167.8530

sol =
```

1

Hypothesis

RRR planar robot.

Version Control

1.0; Leonardo da Cunha Menegon, Michel Kagan, Vinícius Nardelli; 01/05/2023; First issue.

Function

```
function [near, far, sol] = solve_robot(goal, current, trelw, srelb,
    L, thetalim)
```

Validity

```
arguments
goal (1,3) {mustBeNumeric, mustBeReal, mustBeFinite}
current (1,3) {mustBeNumeric, mustBeReal, mustBeFinite}
trelw (4,4) {functions.mustBeHomTransfR}
srelb (4,4) {functions.mustBeHomTransfR}
L (1,2) {mustBeNumeric, mustBeReal, mustBeFinite} = [0.5, 0.3]
thetalim (3,2) {mustBeNumeric, mustBeReal, mustBeFinite} =
repmat([-170, 170], 3, 1)
end
```

Main Calculations

```
wrelt = functions.tinvert(trelw);
```

```
trels = functions.utoi(goal);  
trelb = functions.tmult(srelb, trels);  
wrelb = functions.tmult(trelb, wrelt);  
  
[near, far, sol] = functions.invkin(wrelb, current, L, thetalim);
```

Output Data

```
[near, far, sol];  
  
end
```

Published with MATLAB® R2020a