

plot_manipulator

Arquivo complementar. plot_manipulator gera o desenho do manipulador. Caso não seja fornecida a matriz de transformação da ferramenta em relação ao punho será considerado que a ferramenta é igual a um ligamento ($tTw=[0.5 \ 0 \ 0]$)

Calling Syntax	1
I/O Variables.....	1
Example	1
Version Control	2
Function.....	2
Main Calculations	2
Plot	3

Calling Syntax

plot_manipulator(theta,L)

plot_manipulator(theta,L,tTw)

plot_manipulator(theta,L,tTw, bTs)

I/O Variables

IN Double Array **theta**: Ângulos das juntas [theta1,theta2,theta3]

IN Double Array **L**: comprimento dos ligamentos [L1,L2]

IN Double Matrix **tTw**: Matriz de transformação da ferramenta para punho na forma do usuário; Caso não seja fornecido esse item, o desenho irá conter um ligamento a mais para facilitar a visualização do theta3

IN Double Matrix **bTs**: Matriz de transformação da estação para a base na forma do usuário; Caso não seja fornecido esse item, o desenho considerará a base como o sistema de coordenadas

Example

```
theta=[30 45 -60];
```

```
L=[0.5 0.5];
```

```
tTw=[0.1 0.2 30];
```

```
bTs=[0.3 0.3 45];
```

```
plot_manipulator(theta,L,tTw,bTs)
```

Version Control

1. Gabriel Pereira das Neves; 2016/08/19 ; First issue.
2. Fabio Fialho; 2019/10/09; Minor modifications.
3. Gustavo Gransotto & Lucas Grob; 2020/09/20; bTs argument inclusion
4. Gustavo Gransotto & Lucas Grob; 2020/01/10; {S} and {T} system plot. Fixed some bugs.

Function

```
function plot_manipulator(theta,L,tTw,bTs)
```

Main Calculations

```
if nargin==2
    tTw=[0.5 0 0];
    bTs=[0 0 0];
elseif nargin==3
    bTs=[0 0 0];
end

% Variaveis teoricas para simplificar o plot da ferramenta
length_tool = sqrt(tTw(1)^2 + tTw(2)^2);
theta_tool = atan2d(tTw(2),tTw(1));

T01=[cosd(theta(1)) -sind(theta(1)) 0 0;
     sind(theta(1))  cosd(theta(1)) 0 0;
     0                0                1 0;
     0                0                0 1];

T12=[cosd(theta(2)) -sind(theta(2)) 0 0;
     sind(theta(2))  cosd(theta(2)) 0 0;
     0                0                1 0;
     0                0                0 1];

T23=[cosd(theta(3)) -sind(theta(3)) 0 0;
     sind(theta(3))  cosd(theta(3)) 0 0;
     0                0                1 0;
     0                0                0 1];

T3T=[cosd(theta_tool) -sind(theta_tool) 0 0;
     sind(theta_tool)  cosd(theta_tool) 0 0;
     0                0                1 0;
     0                0                0 1];

T02=T12*T01;
T03=T23*T02;
T0T=T3T*T03;

L1=[L(1);0;0;1];
L2=[L(2);0;0;1];
PT=[length_tool;0;0;1];
```

```

P1=T01*L1;
P2=T02*L2;
PT=T0T*PT;

```

Plot

```

plot([0 P1(1,1)], [0 P1(2,1)], 'k');
title('Representação do robô no plano cartesiano da base')
hold on
plot([P1(1,1) P2(1,1)+P1(1,1)], [P1(2,1) P2(2,1)+P1(2,1)], 'k');
plot([P2(1,1)+P1(1,1) PT(1,1)+P2(1,1)+P1(1,1)], [P2(2,1)+P1(2,1)
PT(2,1)+P2(2,1)+P1(2,1)], 'k');

plot(0,0, 'bo');
plot(P1(1,1), P1(2,1), 'bo');
plot(P2(1,1)+P1(1,1), P2(2,1)+P1(2,1), 'bo');

grid

trelw = utoi(tTw);
wrelb = kin(theta, L);
trelb = tmult(wrelb, trelw);

plot_uform_transformation(bTs, 'r', 0.15, 1);
plot_uform_transformation(itou(trelb), 'g', 0.15, 1);

ax_lims = axis();
x_length = ax_lims(2) - ax_lims(1);
y_length = ax_lims(4) - ax_lims(3);
text(ax_lims(2)- 0.1*x_length, ax_lims(3)+ 0.2*y_length,
'\{S\}', 'Color', 'r', 'FontSize', 12)
text(ax_lims(2)- 0.1*x_length, ax_lims(3)+ 0.1*y_length,
'\{T\}', 'Color', 'g', 'FontSize', 12)

plot(0,0, 'x', 'Linewidth', 2)
xlabel('x (m)')
ylabel('y (m)')
hold off

end

```