Robótica Espacial
Aula prática nº 6
Estática de Manipuladores

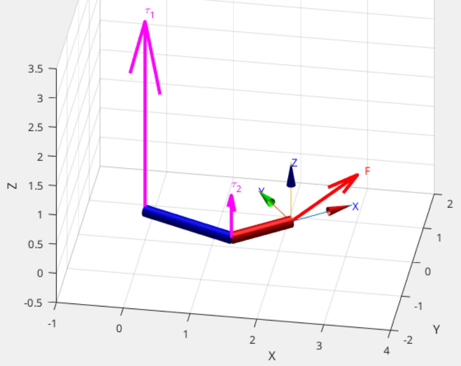
Vitor Santos

Universidade de Aveiro

20 mar 2025

Exercício 1 - Momentos nos atuadores de um RR planar

- Criar um robô RR planar com elos de comprimento L1=1.5, e L1=1
- Para a posição `q=[-pi/8,pi/3]` definir uma força a aplicar na ponta `F=[5;10;0]` N
- Calcular os momentos nos atuadores com $\vec{\tau} = J^{\mathsf{T}}\vec{F}$
- Ilustrar o robô com as opções recomendadas: `'nojoints','nobase','noname'`.
- Adicionar vetores (setas) à representação a mostrar a força aplicada no *end-effector* e vetores alinhados com os eixos das juntas a indicar o valor dos momentos desenvolvidos pelos atuadores.



Neste exemplo usou-se `scale=0.15` para a escala dos vetores

- Sugere-se usar a função quiver3() para representar as setas mas, para facilitar a parametrização, recomenda-se o uso da função `drawFTvector()` fornecida.

Função drawFTvector() para representar forças e momentos

```
function h=drawFTvector(P,v,scale,color,label,prevH)
% Auxiliary function to draw a vector with some parameters
% P and v always needed. Remainder parameters optional.
newPlot=0;
if nargin < 6, newPlot=1; end
if nargin < 5, label='';  end
if nargin < 4, color='r'; end
if nargin < 3, scale=0.5; end
x=P(1); y=P(2); z=P(3);
vx=v(1);vy=v(2);vz=v(3);
if newPlot
  h=quiver3(x,y,z,vx,vy,vz,color,'AutoScaleFactor',scale,'MaxHeadSize',3,'LineWidth',3);
  %You can adjust parameters like the arrow tip size, the line width, etc.
  LLoc=P+v*scale + 0.18*v/norm(v);
  text(LLoc(1), LLoc(2), LLoc(3),label,'Color',color);
  %Optionally add some text to identify the vector
else
  prevH.XData=x;  prevH.YData=y;  prevH.ZData=z;
  prevH.UData=vx; prevH.VData=vy; prevH.WData=vz;
  h=prevH;
end
```

Exercício 1 - Elementos adicionais para o cálculo dos vetores

- Para representar os vetores pedidos é preciso especificar o ponto de aplicação `PX`;
- O da força externa aplicada `PE` é no *end-effector* e essa posição pode ser obtida com a cinemática direta do robô: `T=robot.fkine(q); PE=T.t`
- Para os eixos das juntas é necessária a **transformação geométrica** associada a cada elo:
  - A primeira junta (n=1) está na origem, logo temos a matriz identidade como transformação geométrica (`T0=eye(4)` ou para melhor compatibilidade: `T0=SE3(eye(4))`)
  - Para as seguintes (n=2,3,..) será o produto acumulado de transformações até à junta n:

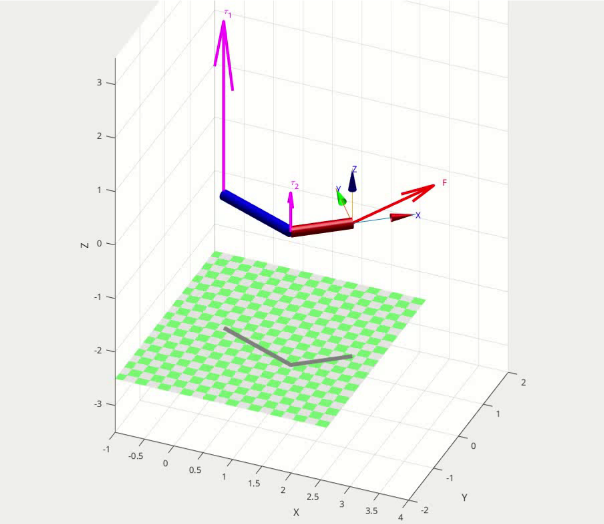
$$T_{n-1} = \prod_{k=1}^{n-1} A_k, \quad \text{para } n = 2, 3, 4, \ldots \quad \text{i.e. } T_1 = A_1, T_2 = A_1 A_2, T_3 = A_1 A_2 A_3, \text{etc.}$$

  - Em matlab, para `robot`, tem-se: `A_k=robot.links(k).A(q(k))` ou mais compacto:
  - `Tn=robot.A(1:n,q)` que devolve o referencial `n` que é o da junta `n+1`.
- Obtida a transformação geométrica `Tn-1` como o referencial da junta `n` virá em matlab:
  - `Pn = Tn-1.t` para o vetor da posição (origem do referencial)
  - `Jnv = Tn-1.a` para o vetor da direção z (direção do eixo da junta no referencial)

Recorde-se que a junta `n` está definida no sistema de coordenadas `{n-1}`, ou seja J1 tem associada a posição T0 (identidade), a junta J2 tem associada a posição T1, e assim sucessivamente, até JN que tem associado T(N-1). A posição TN é a do *end-effector* e não tem nenhuma junta associada.

Exercício 2 - Momentos nas juntas quando a força aplicada varia

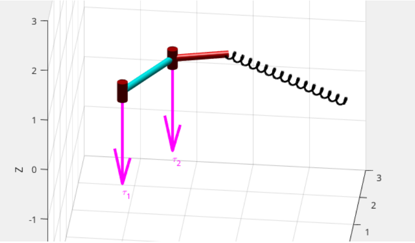
- Estender o exercício 1 para mostrar de forma animada os momentos nas juntas quando a força aplicada variar de direção mas mantendo a intensidade:
  - Intensidade: 10 N
  - Orientação: de 1.25π a -0.75π rad e depois voltar até $\pi/2$ rad.
- Sugestões:
  - Ajustar `axis([ ])` e `view( , )`
  - Ciclo cuja variável é o ângulo da força
  - Obter as suas componentes Fx e Fy
  - Calcular os novos momentos
  - Invocar a função `drawFTvector()` usando os *handles* gráficos do Ex. 1 para atualizar os vetores no ciclo.
- Observe-se os sentidos dos momentos!



External player

Exercício 3 - Momentos por uma mola fixada no end-effector

- Considerar um robô que está preso a uma mola pela sua extremidade
- Simular os momentos desenvolvidos nas diversas juntas quando faz movimentos (que se consideram lentos para ser válida a abordagem da análise estática)
- Assumir o RR planar dos exercícios anteriores.



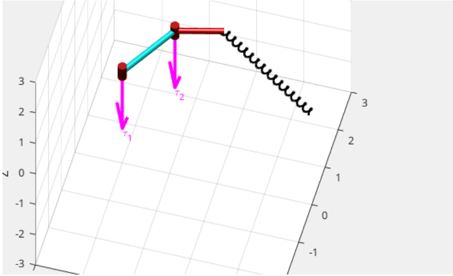
- Foram disponibilizadas as seguintes funções auxiliares para utilizar no código: `drawFTvector.m`, `DrawTorques.m`, `generate_spring.m`
- Para realizar este exercício propõe-se a criação de um ficheiro principal que define o robô, o ambiente (limites dos eixos, ponto de fixação da mola), a trajetória e outras eventuais configurações e depois invoca uma função `animate_robot_with_spring()` onde se faz todo o processo.
- Nos slides seguintes apresenta-se sugestões adicionais

Exercício 3 - Estrutura do programa principal

- Definição do robô (e.g. `LA=1.5; LB=1;`)
- Definição da trajetória: o robô deve começar em `Qi=[pi/3 -pi/4]` e depois evoluir `QA=[pi/2 pi/4]`.
- Definição do ponto de fixação da mola na parede: e.g. `P1 = [4, 0, 0]`
- Definição dos eixos e do ponto de vista
- Representação do robô no ponto de partida (sugerem-se parâmetros como: `'nobase','nowrist','noname','notiles', 'noshadow', 'linkcolor','c','delay', 0.00`, etc.)
- invocar a função onde se faz a simulação do movimento e dos cálculos: `animate_robot_with_spring(robot, q_trajectory, P1, 15, 0.08);`
- Os parâmetros indicados são além do robô e da trajetória proposta, o ponto P1 de fixação da mola e dois parâmetros da mola (numero de esperias e diâmetro das espiras).

Exercício 3 - A função de simulação

- A função de simulação `animate_robot_with_spring()` é fornecida em ficheiro à parte mas que será preciso completar para poder funcionar.
- Os código a preencher ou completar está assinalado com *** e haverá comentários para ajudar no processo.



External player

- As partes principais da função são:
  - Desenhar a mola na figura já criada
  - Definir uma constante da mola para calcular a força
  - Implementar o ciclo de animação com atualização da posição do robô, atualização do estado da mola, cálculo da força da mola, e desenhar ou atualizar os vetores dos momentos.
- Para aplicar a solução noutro robô bastaria mudar a matriz DH e a trajetória. Pode-se evoluir para o RRR planar e depois um RRR antropomórfico, ou outros mais complexos.