

Simulação de um processo de montagem com um robô UR10e

Departamento de Engenharia Mecânica
RI - Universidade de Aveiro
Paulo Guilherme Soares Pereira, 98430
26 de Novembro de 2022

1- Introdução

O presente documento visa explicar o código desenvolvido em MatLab no âmbito do TP2 da cadeira de Robótica Industrial. Era pedido o desenvolvimento de um modelo 3D que visava simular e planear as trajetórias dos seus elos simulando assim um processo de montagem de dois componentes diferentes sobre um outro componente principal, resultando num objeto composto.

Todos os componentes do robô, bem como as componentes do objeto principal foram desenhadas em 3D de acordo com as instruções disponibilizadas.

Foram utilizadas funções definidas nas aulas como AnimateRobot e InitRobot. Foi também utilizada a função a Paralel adaptada do TP1 com o objetivo de criar algumas das componentes.

2- Funções

Além das funções previamente mencionadas foram desenvolvidas 6 funções adicionais

- Cyl: Tal como a função Paralel já mencionada esta função cria um Prisma em que a base é um octógono com intenção de simular um cilindro.
- MakeMat: À imagem da função Paralel esta função consegue devolver ao utilizador um prisma cuja base é um retângulo, sendo usada para simular os tapetes rolantes.
- CreatePieces: Esta função é responsável por criar e mover as componentes do objeto composto para a sua posição inicial
- MovePieces: Esta função anima todo o movimento das componentes.
- ConfigEnv: Esta função é responsável por criar as partes estáticas da animação, ou seja, os tapetes (ou bancadas) e as linhas de paragem dos objetos.
- VfromC: Esta função é capaz de calcular qualquer vértice de um prisma a partir do ponto central da sua face alinhada no eixo xy com valor z mais elevado.

A função TP2 trata de tudo o resto bem como é o motor que chama as outras funções sendo assim a componente “Main” do projeto.

Outras partes a salientar é que devido à falta de tempo no que toca à adaptação do código optei por não desenvolver o ponto que envolvia um ficheiro de texto com

configurações do utilizador, mas apesar disso grande parte do código não é estático e depende dessas configurações iniciais.

3- Ambiente de Execução

1- Estado Inicial

Quando se inicia o programa (pressionando uma tecla após chamar na consola TP2) aparece os componentes do objeto no seu estado inicial numa posição fixa, bem como aparece o Robô que realiza uma pequena animação para se mover para o seu estado inicial.

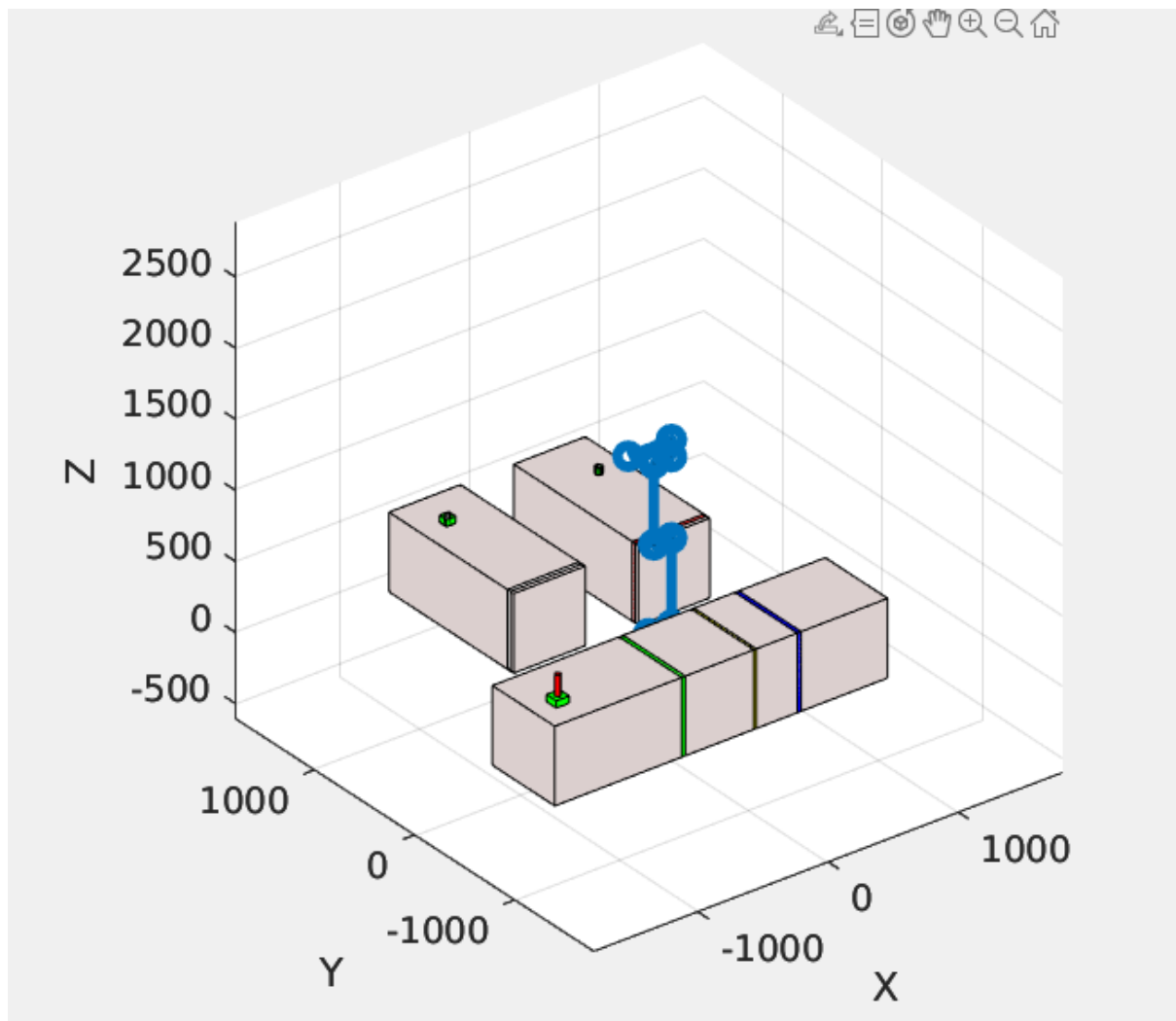


Figura 1- Posição inicial

2- Desenho das figuras

Após isto as peças iniciam o seu movimento ao longo do tapete parando nas linhas designadas, enquanto esperam pelo contacto do robô (C2 e C3), a componente C1 fica à espera que uma das outras componentes seja colocada em cima de si antes de proceder na linha de montagem. Os movimentos do Robot foram calculados usando os seguintes vetores QQ, que incorporam todos os valores dos ângulos dos seus elos (reais e imaginários).

```
QQ = {};  
% Init  
QQ{end+1} = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]'; % 1  
% Get C2  
QQ{end+1} = [0 deg2rad(-150) deg2rad(20) 0 0 deg2rad(60) 0 deg2rad(30) 0 deg2rad(50) 0 deg2rad(80) 0 0]'; % 2  
% Place C2 on C1  
QQ{end+1} = [0 -0.8871 deg2rad(30) 0 0 deg2rad(30) 0 pi/4 0 deg2rad(10) 0 deg2rad(75) 0 0]'; % 3  
% Get C3  
QQ{end+1} = [0 deg2rad(-255) deg2rad(20) 0 0 deg2rad(60) 0 deg2rad(30) 0 deg2rad(50) 0 deg2rad(80) 0 0]'; % 4  
% Place C3 on C1  
QQ{end+1} = [0 deg2rad(0) deg2rad(30) 0 0 deg2rad(33) 0 deg2rad(55) 0 deg2rad(25) 0 deg2rad(75) 0 0]'; % 5  
%reset  
QQ{end+1} = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]'; % 6
```

Figura 2- Definição de QQ

Usando a diferença entre os vários vetores de QQ é possível definir os movimentos do Robot. Nesta zona de código (linhas 95-175) define-se todo o comportamento do robô e das peças durante o movimento.

3- Estado Final

Quando se atinge o final do tapete a peça desaparece e o ciclo retoma o seu início usando um ciclo while que apenas inicia as variáveis estáticas/robô/componentes, uma vez numa tentativa de otimização da velocidade.

Todos os componentes e valores calculados são apagados no fim de cada ciclo para libertar memória.

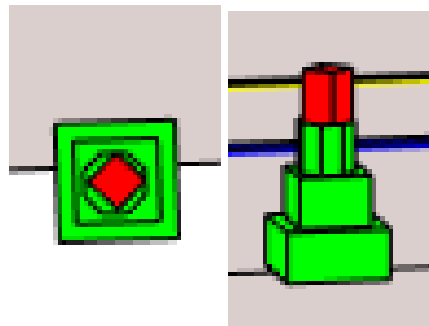


Figura 3 e 4- vista aérea e final da peça completa

4- Conclusão

Alguns requisitos não foram cumpridos devido à falta de tempo, mas no geral considero que o projeto realiza a animação da forma correta e consegue montar um objeto final coerente e sem falhas durante um tempo indefinido.

É possível encontrar o vídeo de demonstração no seguinte link: <https://youtu.be/1Qa8R03O5xc>