Universidade de Aveiro
Dep. de Engenharia Mecânica

Robótica Industrial

Trabalho Prático nº 2 - Novembro 2022

Simulação de um processo de montagem com um robô UR10e

Objetivo e descrição geral

O objetivo principal do trabalho é planear e simular trajetórias de um robô com seis graus de liberdade, similar ao UR10e da Universal Robots, para realizar um processo de montagem de dois componentes diferentes sobre um outro componente principal, resultando num objeto composto. O componente principal (C1) não precisa de ser manipulado, e os outros dois componentes (C2 e C3) chegam em dois tapetes transportadores diferentes. O componente principal surge num tapete que passa no espaço de trabalho do robô e deve parar em dois pontos diferentes (A e B) para otimizar os tempos de ciclo. O objeto composto é escoado no final do tapete principal. Existe uma ordem de montagem dos componentes que deve ser respeitada: o componente C2 deve pousar sobre o componente C1, e o componente C3 deve pousar em cima dos dois. As dimensões do robô e o posicionamento dos tapetes são configuráveis. Os movimentos incluem planeamento no espaço operacional e no espaço das juntas.

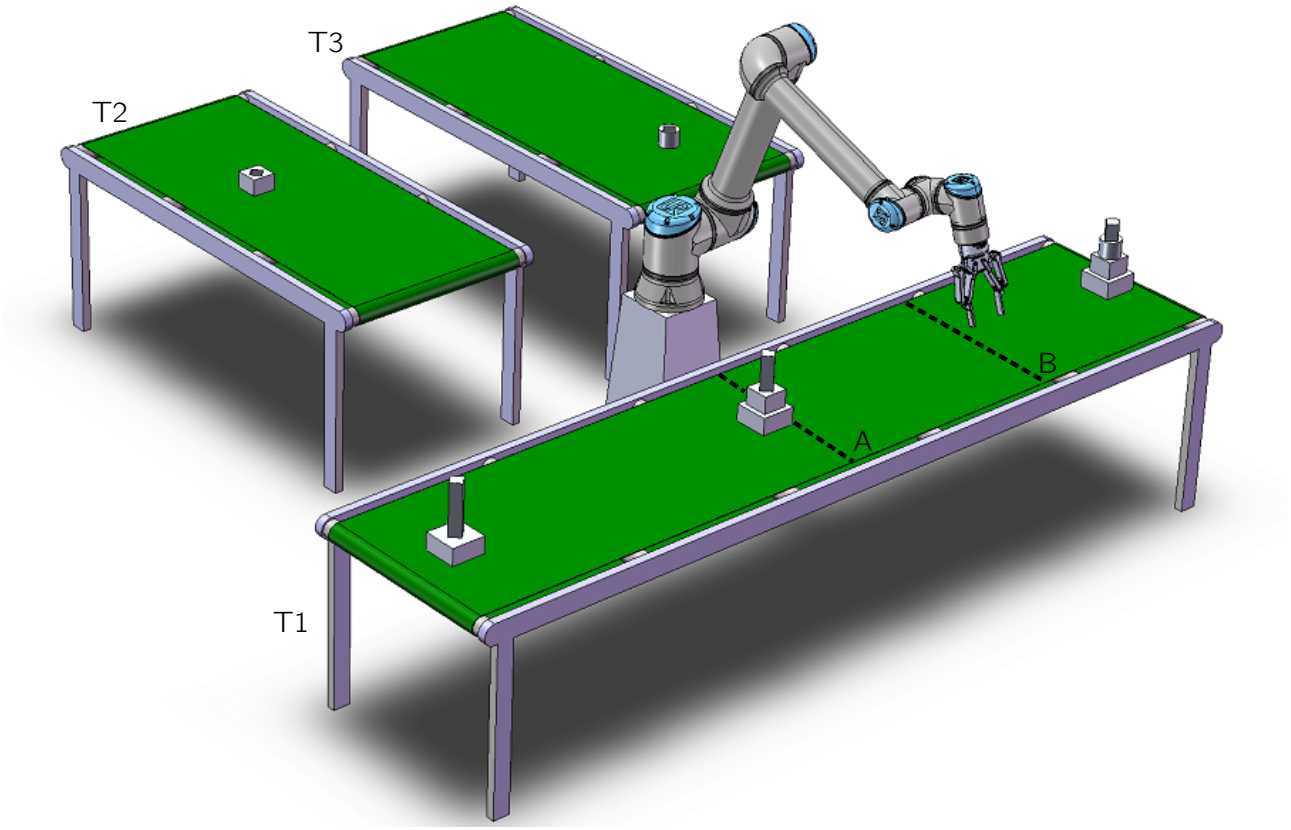

Figura 1: *Ilustração da célula com o robô, tapetes e componentes num estado intermédio do processo. Em primeiro plano está o tapete T1 onde vem o componente principal e sobre o qual se faz a montagem. Para efeitos ilustrativos mostra-se o objeto principal à esquerda, o objeto no primeiro ponto de paragem A já com o componente C2 montado, e à direita do tapete o objeto já todo montado depois do ponto de paragem B.*

O robô está em cima de um pedestal com a mesma altura dos tapetes; parte de uma certa configuração inicial e efetua a montagem na ordem adequada dos dois componentes sobre o componente principal, e quando terminar o processo todo regressa de novo à configuração inicial (figura 2).

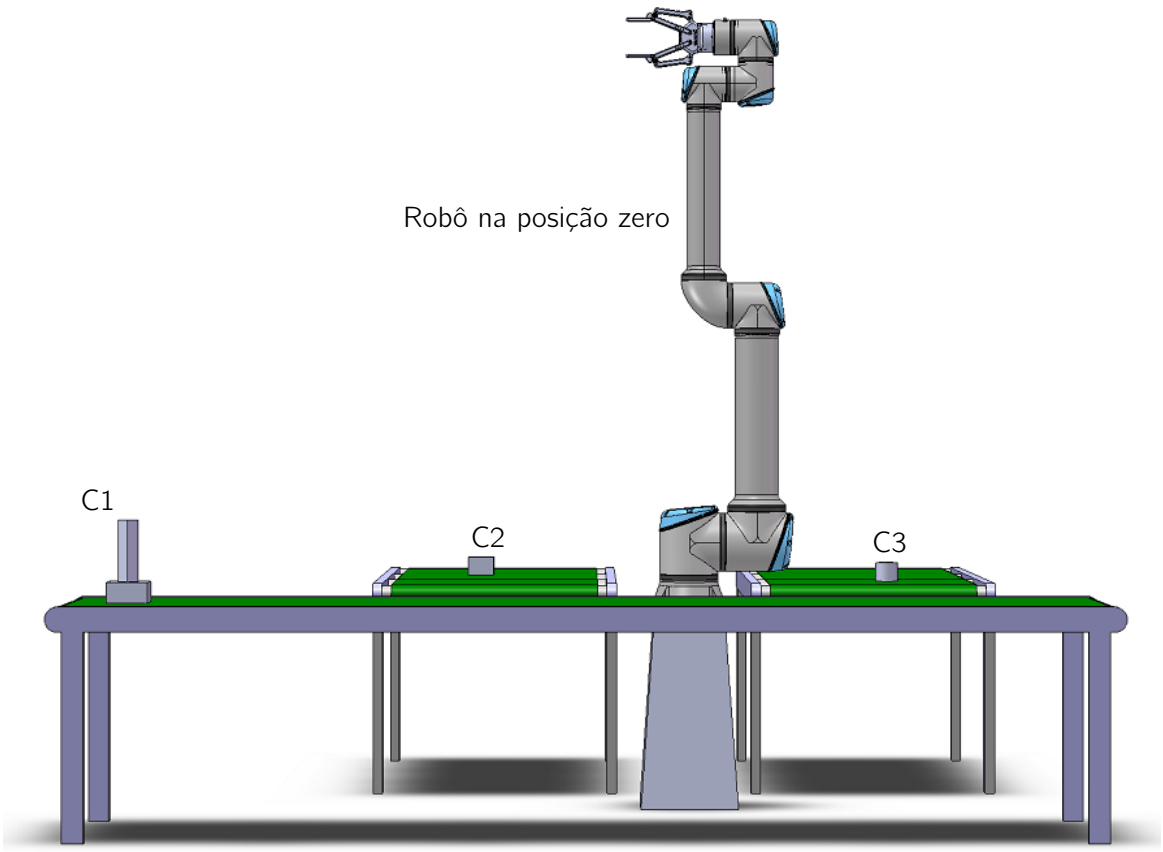

Figura 2: *Configuração inicial do robô que é também a configuração depois do processo terminar.*

Descrição do processo completo

1. Os três componentes (C1, C2 e C3) surgem no início de cada tapete transportador (T1, T2 e T3). Os componentes C2 e C3 movem-se até próximo do outro extremo, junto do robô, onde há um sensor que deteta a chegada de um componente e que suspende o movimento do tapete quando esse componente lá chega, e depois retoma o movimento quando o componente é retirado pelo robô. No caso do tapete T1, o componente terá dois pontos de paragem, A e B, cada qual controlado por um sensor para garantir a paragem do objeto em montagem.
2. No arranque, os três componentes surgem nos seus respetivos tapetes e param junto de um sensor: nos tapetes T2 e T3 isso ocorre no seu final, no tapete T1 a paragem ocorre primeiro no ponto A.
3. Quando o componente C2 chegar no fim do seu tapete, o robô deve pegar nele com o *gripper* em posição horizontal ou vertical (escolha livre) e encaixa-lo no objeto C1 que deverá estar parado no ponto A do tapete T1.
4. Depois desse encaixe e da libertação do robô, o tapete T1 arranca de novo e o objeto semi-montado (C1+C2) avança até ao ponto de paragem B onde ficará estacionário até à montagem do componente C3 que o robô foi entretanto buscar ao tapete T3.
5. Quando o componente C3 chegar no fim do seu tapete, o robô deve pegar nele e pousá-lo sobre o conjunto já semi-montado (C1+C2) no tapete T1.
6. Quando os três componentes (C1, C2 e C3) estiverem montados sobre o tapete T1, o objeto composto final avança para escoamento num qualquer posto a jusante (na simulação pode desaparecer no final do tapete). Assume-se também que o arranque do tapete T1 é comandado pelo próprio robô quando termina as montagens nos pontos A e B.
7. Este processo repete-se em ciclo contínuo.

Movimentos e trajetórias

1. Os componentes C1, C2 e C3, bem como o objeto montado, C1+C2 ou C1+C2+C3, devem ter movimentos lineares simples sobre os seus tapetes (não é preciso simular a aceleração e desaceleração no arranque e paragem).
2. Devem ser definidos pontos de aproximação para a preensão e colocação dos componentes por parte do robô.
3. Os movimentos entre os pontos de aproximação e os pontos em que se faz o contacto ou libertação dos objetos devem ser em trajetórias lineares no espaço operacional.
4. Os movimentos do robô entre pontos de aproximação devem ser feitos por juntas, mas com planeamento polinomial dos valores de juntas, garantindo que se passa sempre por um ponto de segurança S equidistante dos pontos de *grasping* e de largada dos objetos, mas a uma altura do chão maior ou igual a 1.5 vezes a altura H dos tapetes.
5. Par efeitos de ilustração nesta simulação, o processo fica completo no fim de **três** objetos terem sido montados. Depois disso, a simulação do processo pode parar.

Componentes e suas medidas

Os três componentes (C1, C2 e C3) terão sempre as mesmas dimensões e geometria, e devem ser montados por essa ordem sobre o tapete T1.

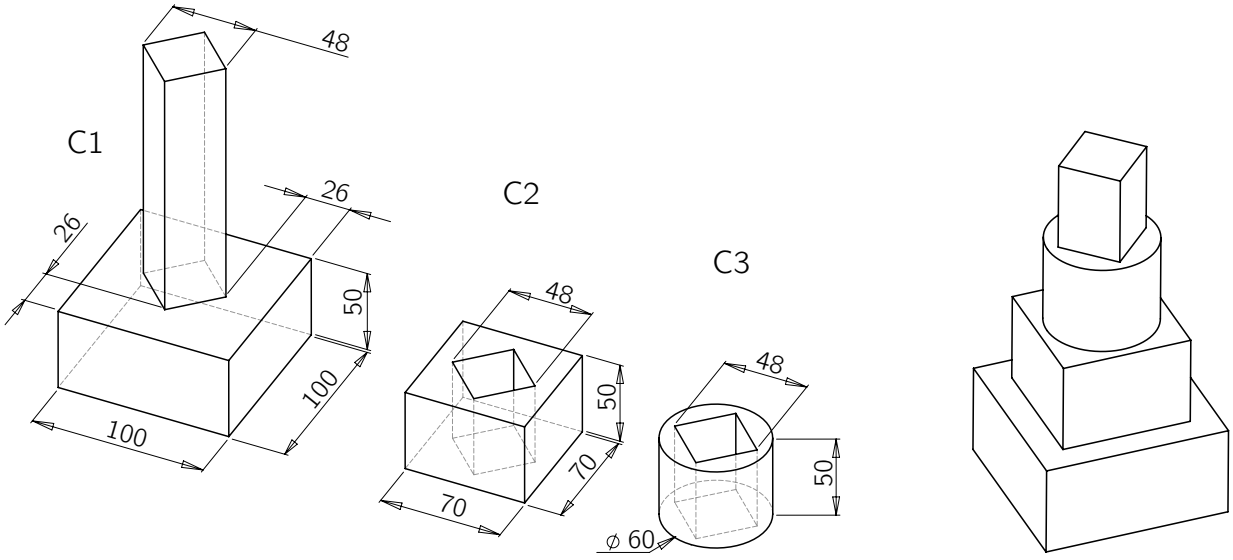

Figura 3: *Geometria e medidas dos três componentes C1, C2 e C3 à esquerda, e a respetiva montagem à direita. As medidas são fixas e a ordem de montagem deve ser sempre esta ilustrada.*

Na simulação em Matlab, o componente C3 (cilíndrico) pode ser aproximado por um poliedro com pelo menos 8 faces laterais.

Medidas do robô, tapetes e posicionamentos

As medidas do manipulador e dos tapetes, bem como a sua posição, serão lidas de um ficheiro de configuração (`tp2.txt`) caso ele exista; se não existir, devem usar-se valores de defeito. A figura 4 indica os nomes das variáveis dessas medidas e os respetivos valores por defeito.

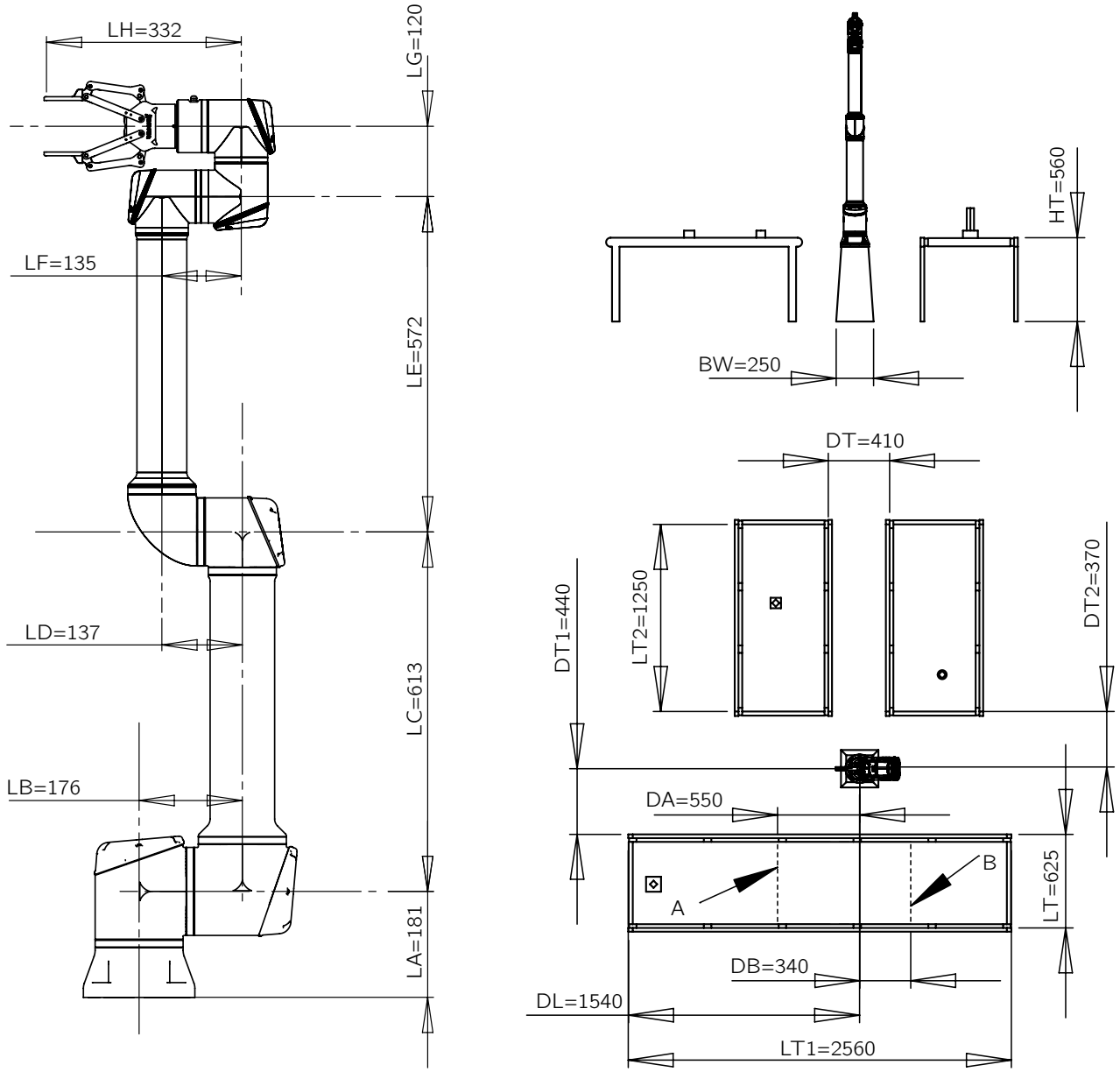

Figura 4: *Medidas do robô na posição inicial e dos tapetes e do seu posicionamento. Os valores indicados são os de defeito uma vez que as medidas reais podem ser carregadas de um ficheiro de configuração, caso ele exista.*

As distâncias das linhas A e B (DA e DB) são medidas a partir de um plano ortogonal ao tapete T1 e que contém o eixo da primeira junta (na base) do robô. Pode-se admitir que a distância entre as linhas A e B é inferior à distância da linha A ao início do tapete, ou seja: $DA + DB < DL - DA$; se assim não fosse, podia dar-se o caso de vir um segundo objeto C1 que pararia em A sem que o anterior C1+C2 tivesse ainda chegado ao ponto B para completar a montagem. O problema também teria solução, mas a sequenciação das ações do robô seriam mais complexas!

Segundo o fabricante do UR10e, as juntas têm todas uma excursão de $\pm 360°$ e, portanto, poderão ser usadas sem restrições relevantes (mesmo havendo riscos de colisão do robô consigo próprio).

Ficheiro de configuração `tp2.txt`

No arranque, o programa a desenvolver deve carregar os parâmetros de configuração do ficheiro `tp2.txt`. Esse ficheiro conterá os parâmetros organizados um por linha indicando-se o nome do parâmetro e o seu valor separados por um sinal de "=" com pelo menos um espaço de cada lado do sinal "=".

No exemplo seguinte ilustram-se as primeiras 5 linhas de um possível ficheiro `tp2.txt` onde se definem algumas variáveis e os seus respetivos valores:

```
LA = 190
LB = 180
LC = 650
DA = 500
DB = 400
```

Em Matlab, para ler um ficheiro com este formato, e criar logo as variáveis com o mesmo nome, pode-se usar um código similar ao seguinte:

```
try
    tfid = fopen('tp2.txt');
    tdata = textscan(tfid,'%s=%s');
    fclose(tfid);
    if( numel(tdata{1}) ~= numel(tdata{2}))
        disp('Error reading file. Missing = !')
        clear tdata tfid
    else
        ndata={ tdata{1} repmat('=', size(tdata{1})) tdata{2}};
        sdata=strcat(ndata{1},ndata{2},ndata{3});
        for i=1:numel(sdata)
            try
                eval(sdata{i});
            catch
                sprintf('Bad format in line %d of data file!',i)
            end
        end
        clear i tfid ndata tdata sdata
    end
catch
    disp('Cannot open file.')
end
```

Note-se que o ficheiro de configuração poderá ter apenas uma parte das variáveis presentes nos diagramas da figura 4. Isso significa que os valores ausentes no ficheiro de configuração terão de ser usados com os valores de defeito.

Assim, a sugestão é que se definam primeiramente todas as variáveis com os valores de defeito, e depois se leia o ficheiro de configuração `tp2.txt` para carregar os valores das variáveis que estiverem no ficheiro.

Para efeitos de teste e avaliação dos trabalhos não serão feitas variar as medidas dos tapetes (LT, LT1, LT2) mas apenas as medidas da sua altura (HT) e da sua colocação (DT, DL, DT1,DT2) e a posição dos detetores de paragem dos objetos no tapete 1 (DA, DB).

Pontos e tarefas relevantes do trabalho

1. Estabelecer a cinemática direta do sistema completo (metodologia de Denavit-Hartenberg).
2. Estabelecer a cinemática diferencial (Jacobiano) do sistema completo.
3. Desenvolver as funções para a cinemática inversa do sistema.
4. Obter as configurações da célula a partir do ficheiro `tp2.cfg`. Se o ficheiro não existir, devem usar-se os valores de defeito indicados no enunciado.
5. Representar graficamente o sistema na sua configuração inicial. Devem ser visíveis as estruturas geométricas dos tapetes (retângulos sobre 4 apoios, ou similar) e os elos do manipulador num grafismo similar ao usado nas aulas (`DrawLinks`), mas podem ser enriquecidos, como usar objetos sólidos em vez de simples linhas para representar os elos e tapetes. O gripper também deve ser representado para se perceber a sua orientação e pode ser um simples grafismo de linhas retas a ilustrar os dedos. Devem ser também visíveis os poliedros (*patch*) que representam os diversos componentes a montar.
6. Deve-se verificar se os pontos de destino estão no espaço de trabalho (soluções reais da cinemática). Se houver uma situação de um destino fora dos limites atingíveis, a simulação deve terminar com informação ao utilizador do ponto onde ocorreu o problema.
7. Estabelecer e calcular pontos de aproximação para efetuar o contacto com os componentes; recomenda-se que estes pontos fiquem a uma distância de pelo menos 20 unidades dos componentes. O movimento da ponta do robô entre estes pontos e os pontos de contacto com os componentes é linear.
8. Fazer os diversos planeamentos de trajetória de acordo com as recomendações anteriores (planeamento nas juntas e no espaço operacional, conforme o caso).
9. Gerar gráficos com os valores das três primeiras juntas para um primeiro ciclo do processo.
10. Antes de executar a animação da simulação, representar previamente o caminho do *end-effector* em todos os pontos correspondentes ao primeiro ciclo completo de montagem. Esta representação deve poder ser suprimida por opção do operador.
11. Com base nos cálculos efetuados, animar o movimento do robô para cumprir a tarefa completa e gerar um filme demonstrativo. Durante a animação deve-se movimentar todos os componentes.

Material a entregar

1. Código Matlab. Um *script* e todas as funções necessárias. Ficheiro empacotado.
2. Um relatório em PDF (6 páginas max.) a explicar os cálculos e as funções desenvolvidas e os procedimentos principais da abordagem com eventuais ilustrações e tabelas. É altamente recomendado usar o LaTeX com o *documentclass* `report`, ou similar. O trabalho deve ter uma capa com pelo menos o título e identificação do autor (O LaTeX também pode gerar essa capa).
3. Vídeo colocado *on-line* no YouTube com a simulação do processo. O relatório deve incluir o *link*.
4. [Opcional] Ficheiro PDF (ou HTML) gerado automaticamente pelo Matlab com base nos comentários devidamente inseridos no código. Consultar as referências ao `publish` do Matlab.

N.B. Todo o material de outros autores usado no trabalho deve ser referenciado: no código, preservar as notas de copyright, e no relatório citar as fontes usadas.

Avaliação

Para além dos pontos das duas secções anteriores, a avaliação também levará em conta o seguinte:

1. Movimento do robô e objetos síncrono (mais do que um em simultâneo) ou assíncrono (objetos/robô movem-se um de cada vez) [Obrigatório — **desconto de até 8%** se não for síncrono].
2. Posicionamento inicial dos objetos C1, C2 e C3 aleatório ao longo da largura dos respetivos tapetes. [Opcional — **bónus de 2%** se ocorrer com sucesso].
3. Inexistência de colisões [Obrigatório — **2% de desconto** se observadas durante a execução.]