

1 Objetivo e descrição geral

O objetivo principal do trabalho é estabelecer as cinemáticas e simular trajetórias de uma grua robotizada para operações de inspeção e instalação de componentes decorativos numa árvore. A grua robotizada transporta um operador no cesto que está na extremidade do braço com o propósito de o posicionar e mover de tal forma que o operador vai colocando as decorações enquanto o robô se move.

A cadeia cinemática é redundante e possui um total de 9 juntas, 8 rotacionais e uma prismática. Dessas juntas, 7 são atuadas e duas são passivas. As juntas passivas têm um valor ditado pela posição de outras juntas, ou seja, ajustam-se automaticamente em função de outras juntas, como explicado adiante.

Todo o sistema está instalado em cima de um veículo com rodas, mas que se considera imobilizado durante as operações de movimentação do robô. Portanto, a posição do veículo é definida para uma operação de movimentação do braço e ficará fixa durante toda a operação.

O controlador do robô deve ser capaz de posicionar o cesto em determinadas posições específicas e deve também ser capaz de impor determinados movimentos do cesto para cumprir trajetórias necessárias para o operador colocar elementos de material decorativo (como iluminação ou similar) ao longo da árvore.

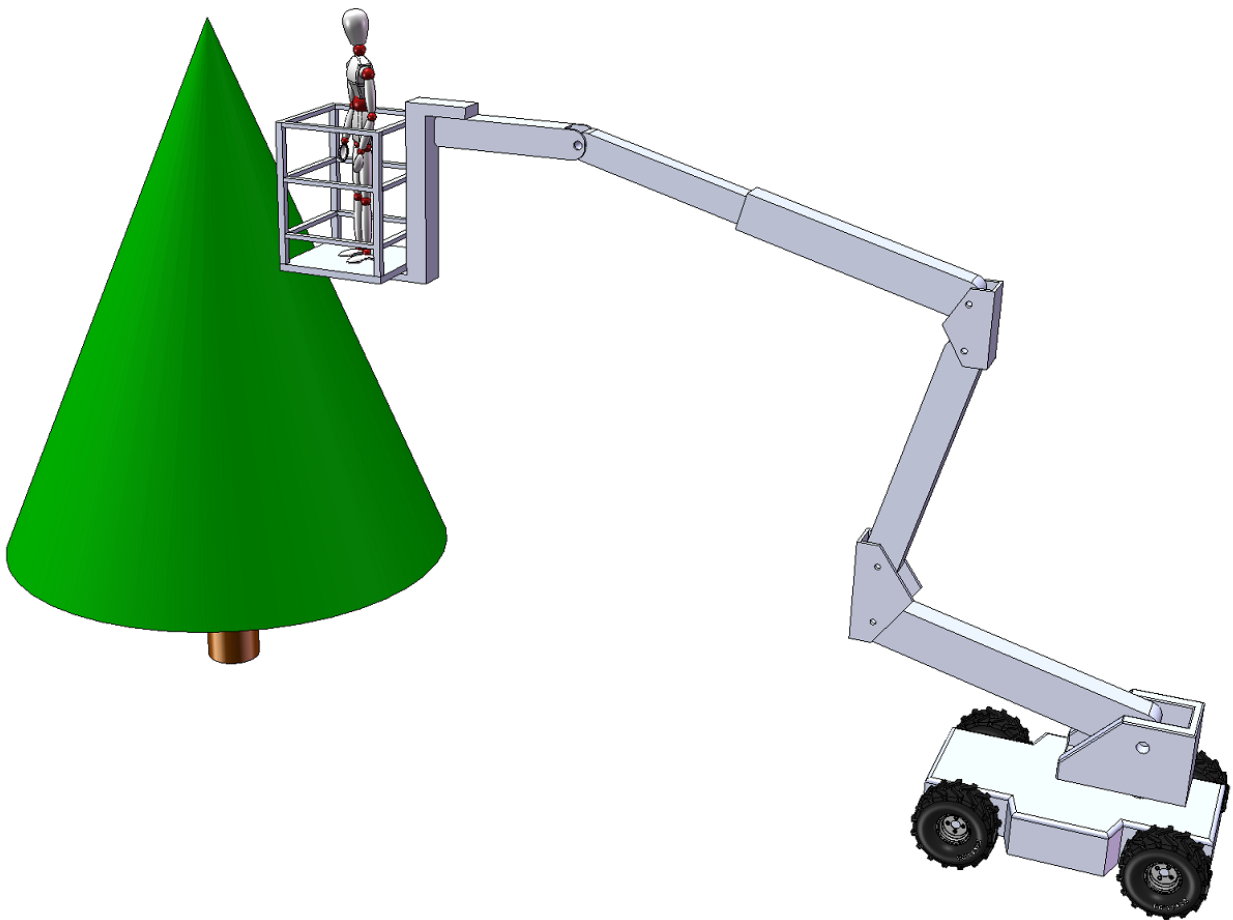


Figura 1: Ilustração de um cenário típico com o operador a ser transportado e posicionado pela grua robotizada em torno da árvore.

2 Medidas do robô e posição zero hardware

O braço robotizado está colocado em cima de um veículo de quatro rodas, e parte de uma certa configuração inicial que é considerada a posição *zero hardware*, conforme ilustrado na figura 2. Essa é também a posição a que o sistema deve regressar no final do procedimento.

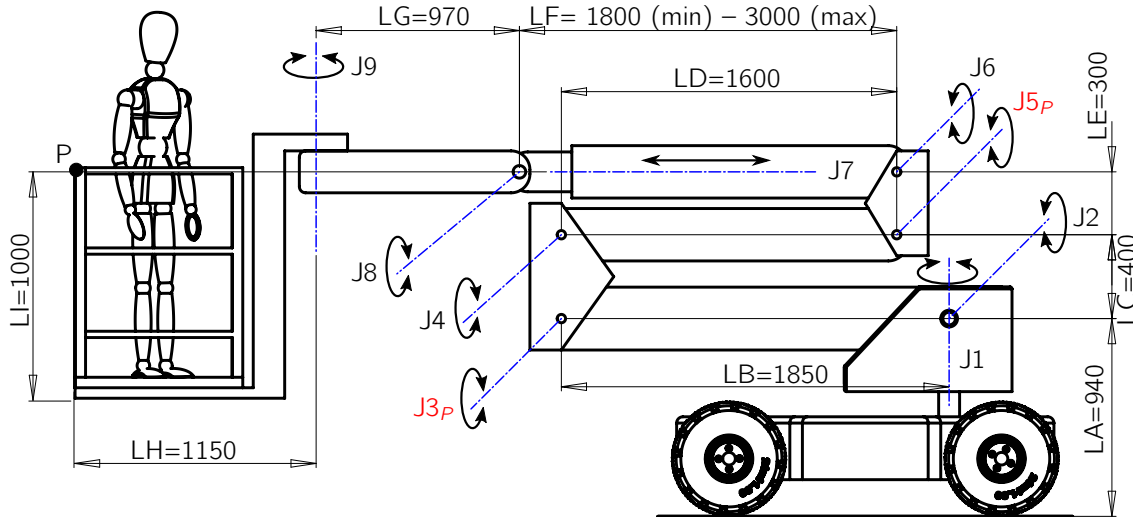


Figura 2: Configuração inicial do robô (Zero Hardware ou Home Position) que é também a configuração que deve ter depois do processo terminar.

Na figura 2, além das medidas de defeito e da representação na posição *zero hardware*, estão também indicadas as juntas do sistema. As juntas J1, J2, J4, J6, J7, J8 e J9 são atuadas independentemente umas das outras com motores individuais. A junta J7 é prismática e tem valores mínimo e máximo. As juntas J3p e J5p são juntas passivas e adaptam-se automaticamente conforme os valores das juntas J2 e J4, respetivamente. Os seus valores são tais que os elos entre as juntas [J3p - J4] e [J5p - J6] se mantêm sempre com a mesma orientação em relação à base do braço. A figura 3 ilustra essa situação.

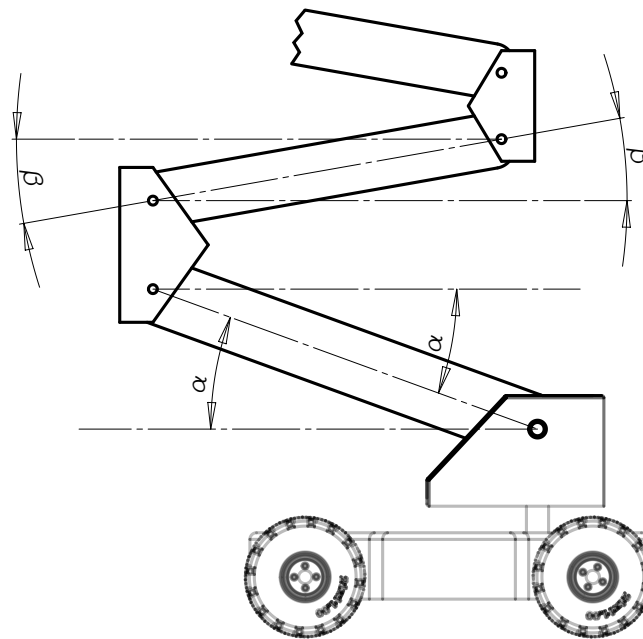


Figura 3: Valores dos ângulos relativos às juntas passivas. Na figura, os ângulos α e β representados traduzem a natureza passiva das juntas J3p e J5p que são impostas pelas juntas J2 e J4.

3 Dimensões e posição da árvore

As dimensões de defeito da árvore são as indicadas na figura 4. As medidas de PH e PD serão sempre fixas independentemente de quaisquer outros dados ou medidas. Por outro lado, a distância entre a árvore e a base do robô TD terá de ser estabelecida na resolução do problema em função das restantes variáveis.

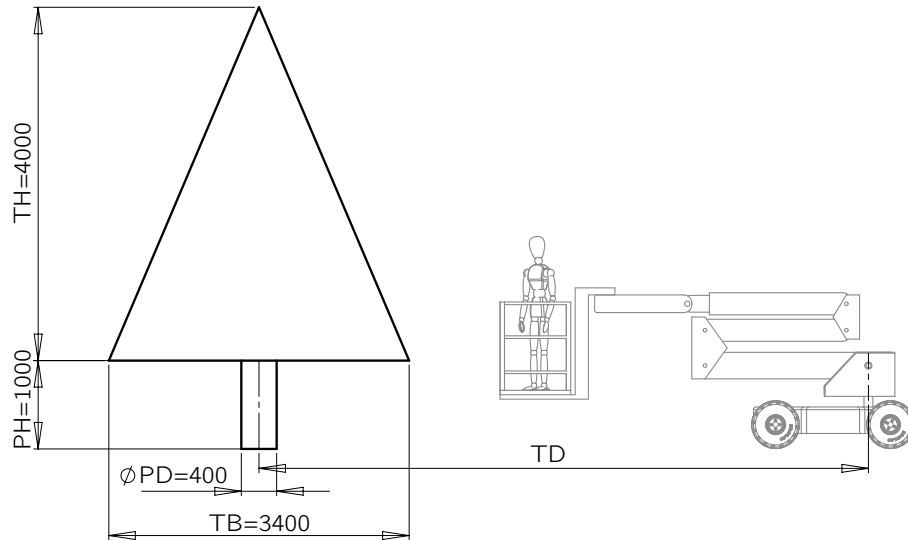
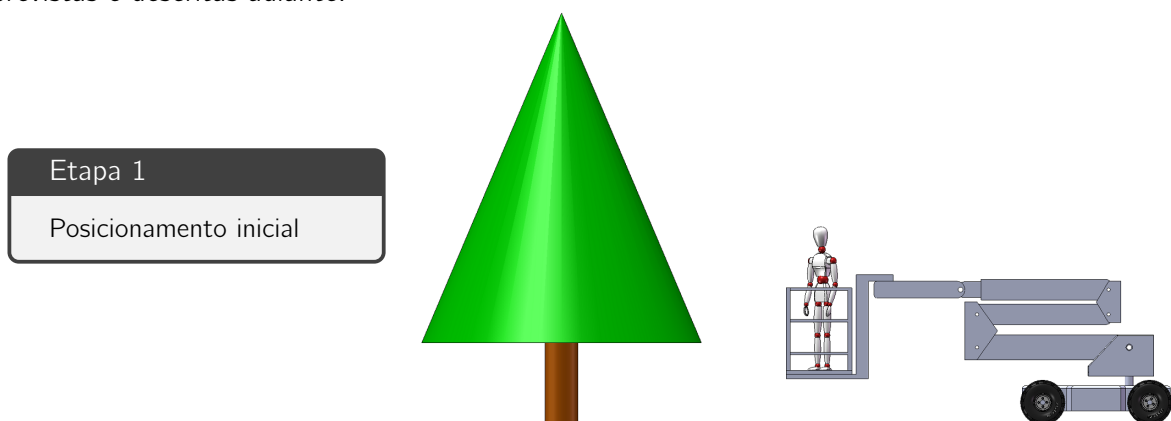


Figura 4: Dimensões de defeito da árvore.

4 Descrição do processo completo

Para os movimentos pedidos, considera-se que a ponta do *end-effector* é o ponto médio da barra superior frontal do cesto, ou seja, o ponto P assinalado na figura 2. As etapas do processo são as seguintes:

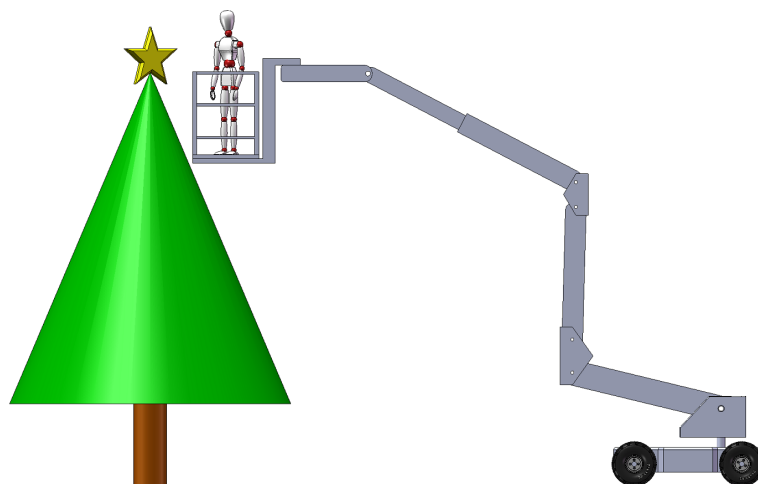
1. O veículo deve ser colocado a uma dada distância da árvore numa posição de alinhamento frontal. Essa distância (TD na figura 4) é arbitrária, mas deve ser tal que seja possível executar as tarefas previstas e descritas adiante.



2. Partindo da posição inicial (*home position*) o braço deve ser posicionado a uma altura tal que o operador possa colocar uma estrela no topo, ou seja, o ponto P deve ficar ao nível da ponta da árvore e a uma distância $D=500\text{ mm}$ medida na horizontal. Este movimento ascendente deve ser feito com planeamento polinomial nas juntas precedido do cálculo da cinemática inversa para atingir esse ponto de destino. Mesmo sendo um movimento com planeamento nas juntas, é necessário garantir que o cesto mantenha sempre a posição vertical, o que é possível de fazer face à redundância cinemática!

Etapa 2

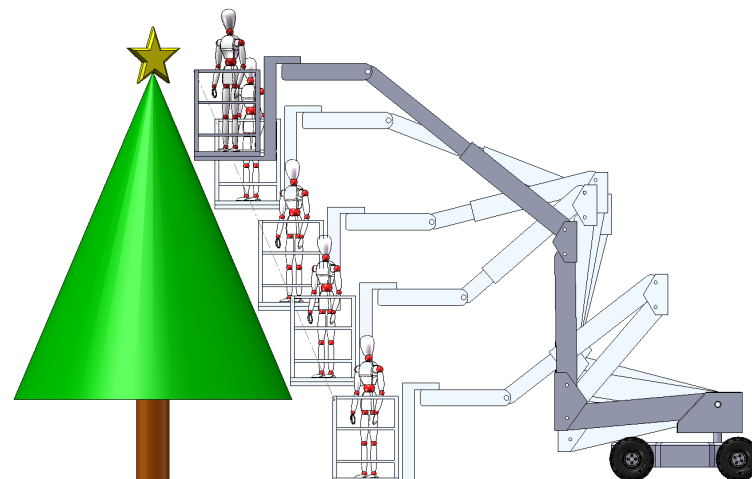
Movimento por juntas para o topo da árvore



3. Partindo da posição anterior, o cesto deve descer de forma paralela ao longo da árvore segundo uma linha reta até à base da sua copa mantendo sempre a distância $D=500\text{ mm}$ já referida. Esta etapa pressupõe o uso de cinemática diferencial.

Etapa 3

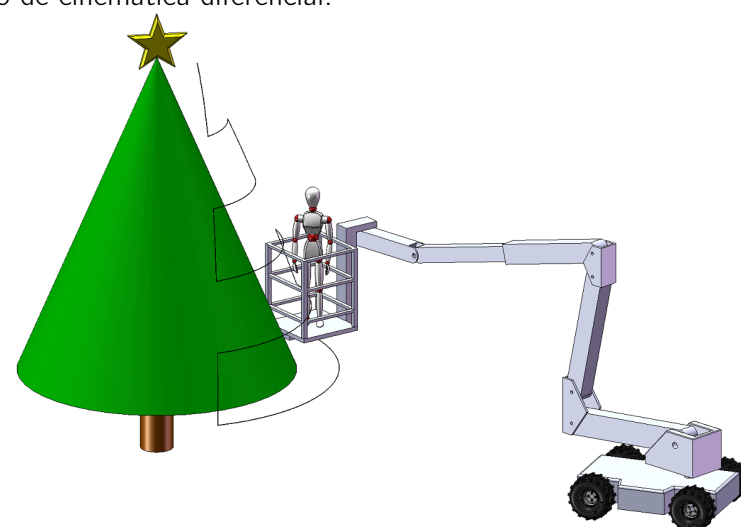
Movimento linear do topo para a base da árvore



4. Depois de voltar a subir pela mesma trajetória linear, o cesto deve descer em zigue-zague com uma composição de movimento circular alternada em torno da árvore e com movimento "linear" descendente. Esta etapa pressupõe o uso de cinemática diferencial.

Etapa 4

Movimento linear da base para o topo e descida em zigue-zague



5. Quando termina o percurso de zigue-zague, o robô finaliza o procedimento retomando a posição *zero hardware* em movimento por juntas, ou seja, retoma a posição da etapa 1.

5 Descrição da trajetória em zigue-zague

A trajetória em zigue-zague é composta por 5 segmentos "lineares" e 5 segmentos circulares. Representam o percurso que o ponto P do cesto deve fazer começando junto ao topo da árvore. Este percurso deve ser executado com velocidade constante do *end-effector* (ponto P). A trajetória deve estar toda definida à distância anteriormente indicada de 500 mm da árvore. As suas dimensões devem ser ajustadas à dimensão da árvore, mas será sempre constituída pelos segmentos referidos (figura 5).

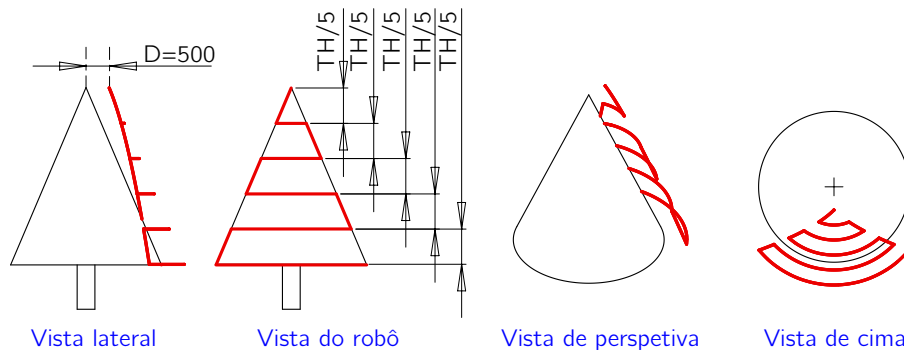


Figura 5: Descrição da trajetória em zigue-zague de diversos pontos de vista.

Do ponto de vista prático, a trajetória pode ser vista como uma projeção sobre uma superfície cônica auxiliar com mais 500 mm de raio em cada ponto do que a superfície cônica da própria árvore. Efetivamente, uma forma possível de obter a trajetória em zigue-zague é definir uma trajetória de troços lineares num plano vertical, afastado da superfície cônica, e fazer a projeção ortogonal sobre a superfície cônica auxiliar, como se ilustra na figura 6.

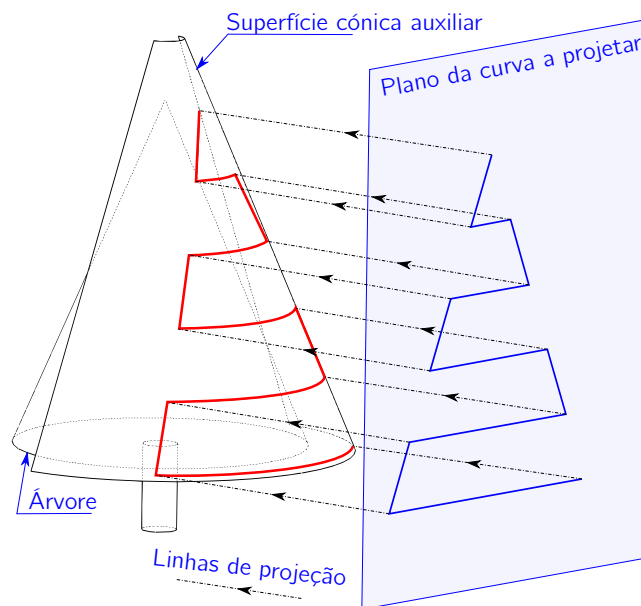


Figura 6: Exemplo de construção da trajetória para o troço em zigue-zague.

A trajetória em zigue-zague a implementar pode apresentar flutuações face à descrita, mas o importante é começar no topo e terminar na base e ter os troços indicados. A amplitude dos troços circulares pode ser diferente da solução indicada. **Nota:** em rigor, nesta abordagem da projeção, os troços ditos lineares não são rigorosamente "lineares" (os cortes num cone geram sempre linhas curvas – as cónicas), mas neste caso podem ser considerados uma boa aproximação a um segmento de reta e, para efeitos da resolução do problema, isso é irrelevante.

6 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 6 linhas de um possível ficheiro **tp2.txt** onde se definem algumas variáveis e os seus respetivos valores:

```
LA = 940
LB = 1850
LC = 400
LD = 1600
LE = 300
LF = 1800
```

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 das figuras 2 e 4. Isso significa que as variáveis ausentes no ficheiro de configuração terão de ser usadas 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.

A medida da altura do cesto (LI), a altura do tronco da árvore até à copa (PH) bem como o seu diâmetro (PD) não serão alterados, e o seu valor é o indicado nas figuras do enunciado. Para traduzir algum realismo físico, o comprimento máximo do elo afetado pela junta J7 (LF_{max}) deve ser limitado a cerca de 1.7 vezes o comprimento mínimo.

7 Pontos e tarefas relevantes do trabalho

1. Estabelecer a cinemática direta do sistema completo (metodologia de Denavit-Hartenberg). As juntas passivas também devem ser contempladas, embora se definam à custa de variáveis de outras juntas.
2. Estabelecer a cinemática diferencial (Jacobiano) do sistema completo.
3. Desenvolver as funções para a cinemática inversa do sistema. Como o sistema é redundante (figura 7), é necessário arbitrar (fixar manualmente) valores de algumas juntas.

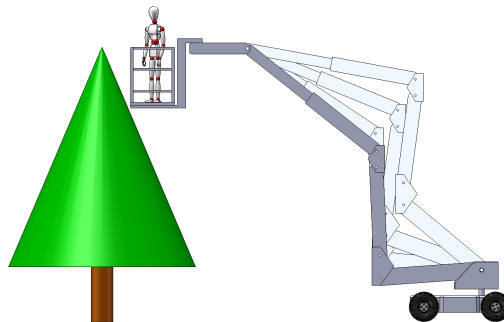


Figura 7: Ilustração da redundância por excesso de juntas para os graus de liberdade envolvidos.

O número de juntas a fixar depende do número de graus de liberdade da solução a implementar. Assim, o processo pode ser feito em abordagens de diferente complexidade:

- (a) Fazer a cinemática inversa apenas num plano: duas coordenadas de translação e uma coordenada de orientação nesse plano (3 graus de liberdade)
 - (b) Fazer a cinemática inversa no espaço a 3D com uma orientação medida num plano vertical (4 graus de liberdade).
 - (c) Fazer a cinemática inversa no espaço a 3D com duas orientações: uma medida num plano vertical e a outra medida em torno de um eixo nesse plano vertical (5 graus de liberdade).
4. Obter as configurações do sistema 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 essenciais: a árvore (cone com um tronco em cilindro ou paralelepípedo), a plataforma do veículo com as rodas (paralelepípedo e 4 cilindros) e os elos do manipulador num grafismo similar ao usado nas aulas (`DrawLinks`), mas podem ser enriquecidos, como usar objetos polidédricos em vez de simples linhas para representar os elos. O cesto deve ser representado para se perceber a sua orientação, e pode ser um simples grafismo de faces retangulares aberto a ilustrar a região do cesto onde estaria colocado o operador.
 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. Planear os diversos movimentos das etapas indicadas: planeamento nas juntas e no espaço operacional, conforme os casos.
 8. Assegurar que em todos os movimentos o cesto está sempre orientado para o cone da árvore, ou seja, além de ter de estar sempre na vertical (para segurança do operador), o seu plano de simetria (definido pelo eixo de J9 e pelo ponto P) deve sempre conter o eixo do cone da árvore.
 9. As juntas têm limites físicos, mas neste exercício só é necessário levar em conta os limites do eixo de J7 que não podem ser excedidos. Isso deve ser demonstrado com a exibição de um gráfico (antes ou depois da conclusão da tarefa) que mostra a variação dessa junta durante todo o movimento.
 10. Antes de executar a animação da simulação, representar previamente o caminho do *end-effector* em todos os pontos correspondentes ao processo descrito. Esta representação deve poder ser suprimida por opção do utilizador do programa.

11. Com base nos cálculos efetuados, animar o movimento do robô para cumprir a tarefa completa e gerar um filme demonstrativo.

8 Material a entregar

1. Código Matlab. Um *script* e todas as funções necessárias, sejam elas desenvolvidas nas aulas ou especificamente para este trabalho. Ficheiro empacotado (.zip, .7z, etc.)
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 L^AT_EX com o *documentclass report*, ou similar. O trabalho deve ter uma capa com pelo menos o título e identificação do autor (O L^AT_EX também pode gerar essa capa).
3. Vídeo colocado *on-line* no YouTube com a simulação do processo; recomenda-se um filme de 1.5 minutos no máximo e que deve ter um cabeçalho inicial de cerca de 5 segundos com o nome do trabalho, do autor, do ano letivo, da disciplina e da Universidade. Não usar resoluções muito baixas (usar 720 linhas ou mais). O relatório deve incluir o *link* para o filme.
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.

9 Avaliação

Para a avaliação serão levados em conta os materiais entregues (relatório e filme no youtube – com cerca de 20% de peso) e o resultado da execução do trabalho (cerca de 80 % de peso) — um trabalho que resulte em erro ou falha de execução pode comprometer seriamente a avaliação neste projeto. O resultado da execução pressupõe também a verificação dos pontos enumerados na secção 7; desses, são pontos muito importantes os 3 primeiros, e na cinemática inversa a realização da variante (c) permite obter a classificação total nesse item, e as variantes (a) ou (b) terão classificações mais reduzidas. O relatório deve também refletir claramente as opções e soluções implementadas.

Alguns itens da secção 7 não são críticos para a funcionalidade geral do trabalho, mas a sua não verificação tem efeitos na classificação. São exemplos os itens relativos à não verificação dos limites do espaço de trabalho (item 6), à não demonstração dos limites das juntas pedidas (item 9), ou ao não cumprimento das orientações do cesto nos movimentos (item 8).

Os programas entregues serão alvo de escrutínio por plágio através do sistema MOSS (<https://theory.stanford.edu/~aiken/moss/>). Trabalhos que ultrapassem os limites de similaridade serão analisados manualmente para confirmação. São excluídos da comparação as funções criadas nas aulas.