Simulação de uma Grua Robotizada para Operações de Inspeção

Departamento de Engenharia Mecânica
RI - Universidade de Aveiro - Vitor Santos

Filipe André Seabra Gonçalves - 98083

Dezembro de 2023

1 Introdução

Este relatório tem por objetivo explicar a abordagem e o código desenvolvido para obter uma aplicação em Matlab que mostre, de forma animada, a simulação de uma grua robotizada para operações de inspeção, dados os tamanhos referentes aos componentes da grua.

Para a construção da árvore de natal, foram usados os valores defeito mostrados no enunciado, uma vez que a árvore não deverá ter mudanças em tamanhos.

Para a grua, os tamanhos dos componentes são lidos do ficheiro **traj.csv**, e se este não existir na pasta quando o projeto for executado, serão usados os valores defeito referidos no enunciado.

Podemos ver a construção defeito de ambos os objetos na Figura 1

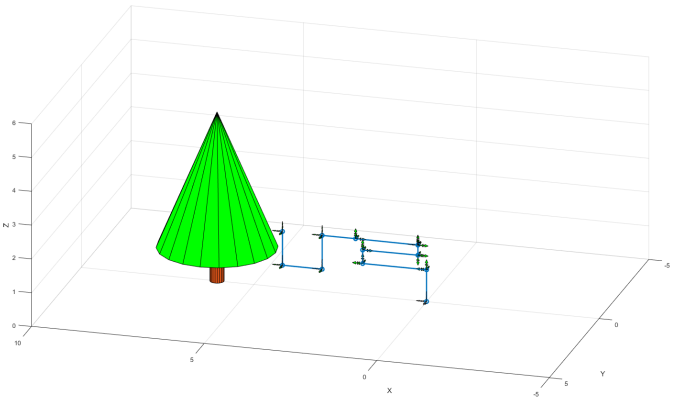

Imagem 1: Grua e Árvore de Natal

2 Funções Usadas e Criadas

No decorrer do projeto foram usadas algumas funções criadas nas aulas práticas, como **jacobianGeom** para calcular o Jacobiano direto, **InitRobot** para iniciar os **handlers** gráficos da grua e dos seus eixos e **AnimateRobot** para animar o movimento do robot, de acordo com os valores de entrada das juntas.

Foi também criada uma função especificamente para este projeto que calcula a cinemática inversa da grua, **invkinPROJETO**.

3 Procedimento

3.1 Ficheiro de Configuração

Os tamanhos de cada componente da grua são retirados de um ficheiro de configuração **tp2.txt** que deve estar presente no mesmo diretório que o projeto. Caso não existe este ficheiro dentro da pasta, o programa escreve uma mensagem no terminal, "Cannot open file.", e desenha a grua de acordo com os valores defeito presentes no enunciado do projeto.

Para uma melhor visualização e para não estar a fazer cálculos com números muito grandes, os tamanhos todos foram reduzidos por 1000, passando a trabalhar em metros ao invés de milímetros, mantendo sempre uma escala de 1:1000, sem influenciar cálculos ou a complexidade destes.

3.2 Árvore de Natal

A árvore de natal é composta por dois cilindros, usando a função **cylinder** do **Matlab**, que dados os raios das bases, cria um cilindro com esses tamanhos. Ãssim, para criar a parte verde de uma árvore de natal, foi criado um cilindro com a base do topo com 0 de raio, simulando um cone, e a base de baixo com o tamanho dado. Para o tronco, foi criado um cilindro com o raio das bases igual ao dado.

O cálculo da distância da grua à origem, **TD**, é de acordo com o comprimento da grua no seu **Zero Hardware** e a uma distância de 200 milímetros entre a grua e a árvore.

3.3 Cinemática Direta

A Cinemática Direta foi cálculada com base no desenho da grua dividida em três partes, Figura 3.3, e juntar as partes numa só no final.

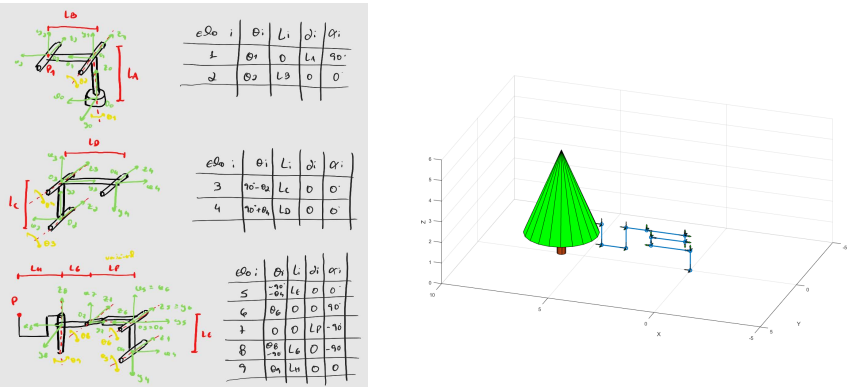

Imagem 2: Desenho da Grua em três partes - Grua no Matlab

A matriz Denavit-Hartenberg deduzida foi a seguinte, estando a grua no seu **Zero Hardware** (sem contar com as juntas virtuais):

$$\begin{pmatrix} \theta_1 & 0 & \text{LA} & \frac{\pi}{2} \\ \theta_2 & \text{LB} & 0 & 0 \\ \frac{\pi}{2} - \theta_2 & \text{LC} & 0 & 0 \\ \frac{\pi}{2} - \theta_4 & \text{LD} & 0 & 0 \\ \theta_4 - \frac{\pi}{2} & \text{LE} & 0 & 0 \\ \theta_6 - \frac{\pi}{2} & 0 & 0 & 0 \\ 0 & \text{LF} & 0 & 0 \\ -\theta_6 & \text{LG} & 0 & 0 \\ \theta_9 & \text{LH} & 0 & 0 \end{pmatrix}$$

Ao invés dos esperados nove valores para as nove juntas, podemos discreterizar dois valores de juntas, Junta 3 e Junta 5, pois sendo juntas passivas, os seus valores serão o simétrico dos calculados para a junta exatamente anterior, Junta 2 e Junta 4, rerspetivamente. Adicionalmente, de modo ao operador da grua não cair, as Juntas 6 e 8 têm de ter os valores simétricos também.

3.4 Cinemática Inversa

De modo a inverter a Cinemática Direta anteriormente calculada, foi usada a mesma estratégia de dividir o problema em partes de modo a tornar o problema mais fácil, resultando nos seguintes cálculos:

- θ_1 - Calculado de acordo com os valores de **x** e **y** - $\arctan(\frac{y}{x})$
- θ_2 - Valor fixado a 0 até ser necessário que a grua vá para valores que exijam que a junta seja movimentada, mudando o seu valor para $\frac{-\pi}{10}$
- θ_4 - Valor fixado a 0 até ser necessário que a grua vá para valores que exijam que a junta seja movimentada, mudando o seu valor para $\frac{-\pi}{2}$
- θ_6 - Calculado de acordo com os valores de **x** e **z** pertencentes ao ponto **Pw**, ponto origem da Junta 6 - $\arctan(\frac{z6}{x6})$
- LF - Calculado de acordo com os valores de **x6** e **z6** - $\sqrt{z6^2 + x6^2}$
- θ_9 - Valor fixado a 0

Para o movimento desde o **Zero Hardware** e o ponto no topo da árvore a uma distância **D**, apenas foi preciso calcular a Cinemática Inversa do ponto final e usar a função **LinspaceVect**, criada nas aulas, de modo a ter um movimento mais suave entres os pontos. A Cinemática Inversa do ponto no **Zero Hardware** é uma matriz de 9 valores iguais a zero, por isso não é necessário calcular.

3.5 Cinemática Diferencial

O cálculo da Cinemática Diferencial foi feito através do cálculo sequencial da diferenciação entre os pontos usando a função **jacobianGeom**, criada nas aulas.

Para os primeiros moviemntos lineares, o seu cálculo não contém grande dificuldade, apenas calcula-se a Cinemática Inversa do primeiro ponto e diferenciasse aos seguintes pontos, sequencialmente.

Para os restantes movimentos, em Zig-Zag ao longo da árvore, primeiramente teve de se calcular os ângulos, que são diferentes para cada segmento, onde os pontos vão acentar na circunferência de centro no centro da árvore, de raio igual ao raio do cone nessa altura mais a distância **D** e paralelo ao plano **xOy**.

Para esta árvore, foi observado que os ângulos para o primeiro segmento curvo é de $\frac{\pi}{7} + \pi$ e $-\frac{\pi}{7} + \pi$, até chegar ao último segmento circular, onde os ângulos são de $\frac{2*\pi}{7} + \pi$ e $-\frac{2*\pi}{7} + \pi$. Assim que os ângulos dos pontos estão calculados, é fácil calcular os pontos por onde a grua tem de passar.

Deste modo, para diferenciar os trajetos lineares podemos apenas calcular o Jacobiano através da função **jacobianGeom**, e calcular a diferença entre pontos **dr**. Porém, para os trajetos circulares, temos primeiro de calcular o centro da circunferência onde os pontos vão assentar, sempre a uma distância de **D** da árvore.

Com o cálculo do Jacobiano e do **dr**, podemos inverter o Jacobiano para obter o Jacobiano Inverso, usando a estratégia da pseudóinversa de uma matriz, usando a função **pinv** do Matlab e multiplicar com o **dr**, para obtermos a diferença entre juntas e pudermos adicionar ao **dq**.

Por fim a trajetória ficou com a forma como é mostrado na Figura 3.5.

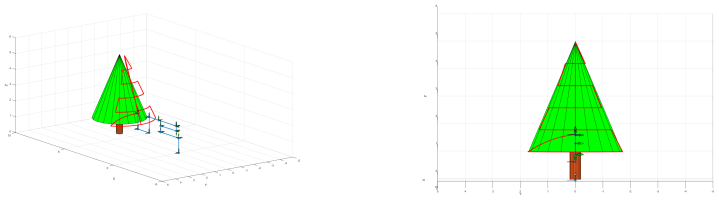

Imagem 3: Trajetória da Grua - Vista Normal e Vista da Grua

3.6 Caminho e Extras

Depois de cálculados todos os pontos, apenas temos de calcular, usando Cinemática Inversa, a trajetória de volta ao ponto inicial, que terá todos os valores a 0.

Adicionando todas as diferenciações de valores nas juntas do caminho todo, pudemos usar a função **InitRobot** para desenhar a grua.

O gráfico com a diferenciação de **LF** é criado de acordo com a matriz criada de acordo com as diferenciações de todos os pontos, apenas usando o valor de **LF**.

Este gráfico, como aparece na Figura 3.6, é mostrado no início do movimento, num **subplot**, juntamente com o movimento da grua.

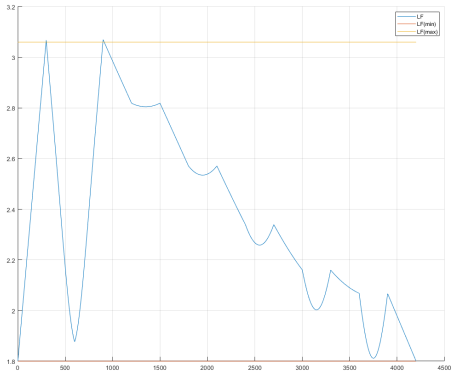

Imagem 4: Valores mínimo possível, máximo possível e usados para LF

Para desenhar o caminho antes da grua se movimentar, usamos a função **LinkOrigins** para calcular o ponto resultante da atuação das juntas com os valores dados.

É importante denotar que a grua espera um segundo antes de começar o seu movimento para ser possível ver todo o moviemnto sem problemas.

O resultado final é mostrado num subplot da seguinta forma, Figura 3.6:

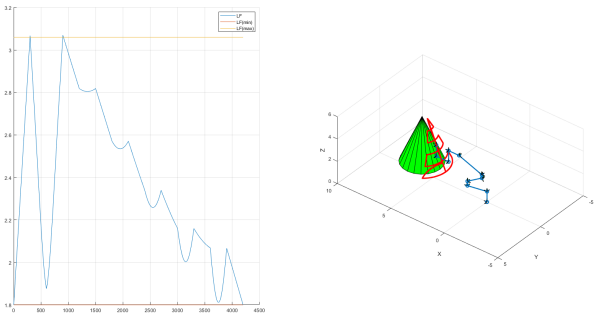

Imagem 5: Output Final

Por último, em relação à orientação da ponta com a árvore, o cálculo do θ_9 pode ser feito a partir do ângulo que o vetor Ponta-Árvore faz em relação ao vetor Origem-Árvore. No entanto, também temos de ter em atenção o valor do θ_1, formando a seguinte equação:

$$-\theta_1 - \arccos(\frac{P1xP2}{||P1|| * ||P2||})$$

Sendo que P1 é o vetor Ponta-Árvore e P2 é o vetor Origem-Árvore. Estes vetores são de duas dimensões, pois, de modo a manter a cabine do operador sempre paralelo ao chão, significa que o θ_9 pode ser calculado no plano xOy.

Contudo, e por muito que esteja a ser bem calculado o ângulo para a Junta 9, não foi possível implementar na grua, por erro de calculo de pontos.

4 Conclusão

Concluindo, foram cumpridos todos os requisitos importantes do projeto, faltando apenas colocar a Junta 9 com valores que se mantém de acordo com o vetor Posição do Operador-Árvore.

Por fim, segue-se o vídeo demonstrativo: `https://youtu.be/DdmD4mIyB9w`.

Qualquer problema que tenha com a submissão de código, tem o link para o github aqui: `https://github.com/FlipGoncalves/Inspection_Operations`.