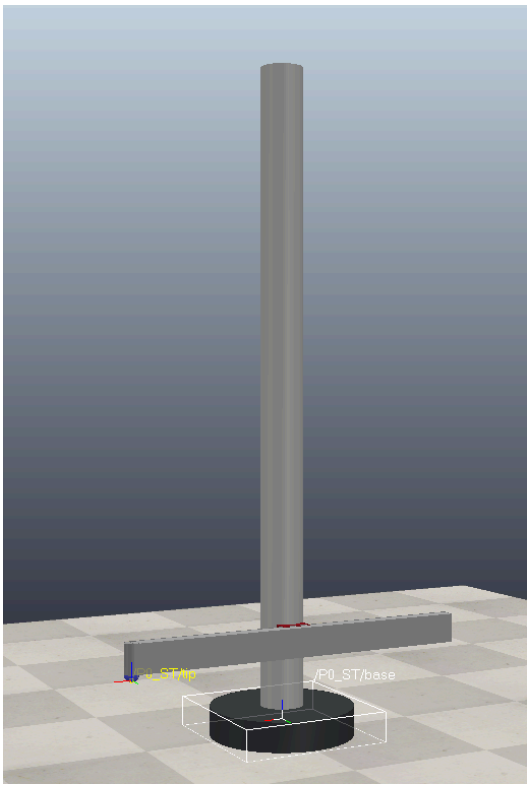
Projeto final - ELT 535 - Manipuladores robóticos

Aluno: Matheus Sousa Soares Matrícula: 122800

Para o trabalho final foi construído um manipulador cilíndrico no simulador CoppeliaSim conforme a imagem abaixo.



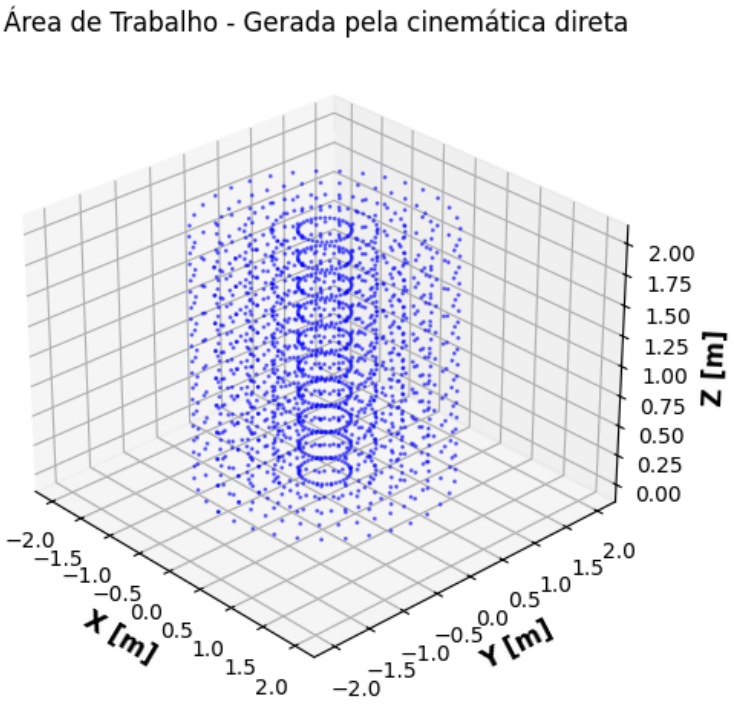
Esse manipulador foi construído com o intuito didático de desenvolver a simulação e de calcular a cinemática inversa. O frame do robô foi alinhado com a origem de coordenadas do CoppeliaSim e foi criado um dummy tip para representar a posição da ferramenta do robô.

Todos os arquivos do projeto podem ser encontrados no repositório: https://github.com/Mat198/cylindrical_robot_arm. A cena com o robô, chamado P0_ST está no arquivo ‘cilyndric_robot_scene.ttt’ na pasta ‘sim’.

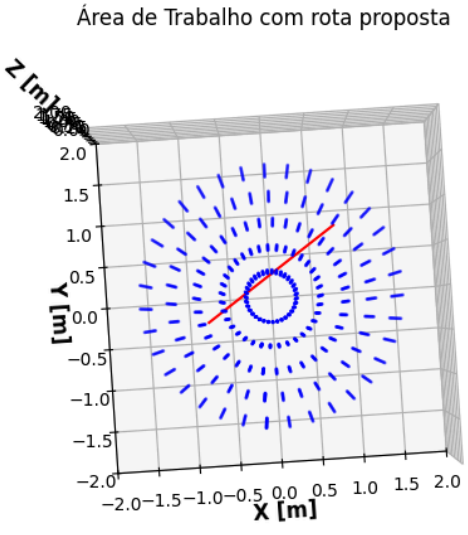
O vídeo com os detalhes do trabalho pode ser visualizado no link: https://youtu.be/5Y4MxMDjVys.

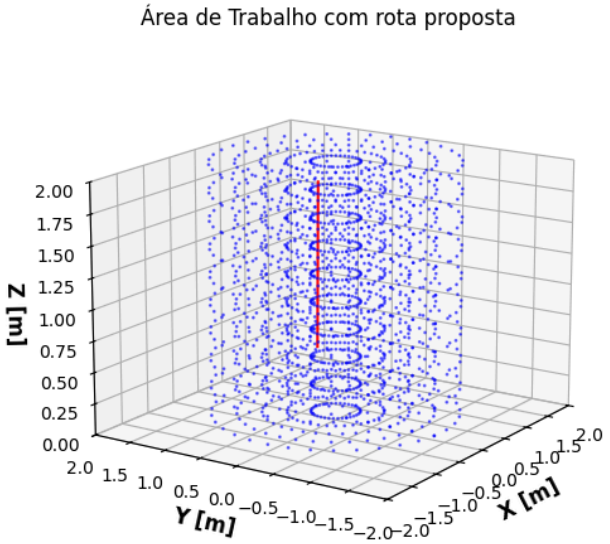
Alcance do manipulador

O alcance do robô foi plotado no gráfico abaixo. Esse gráfico foi gerado por meio do cálculo da cinemática direta para diversas configurações de juntas.

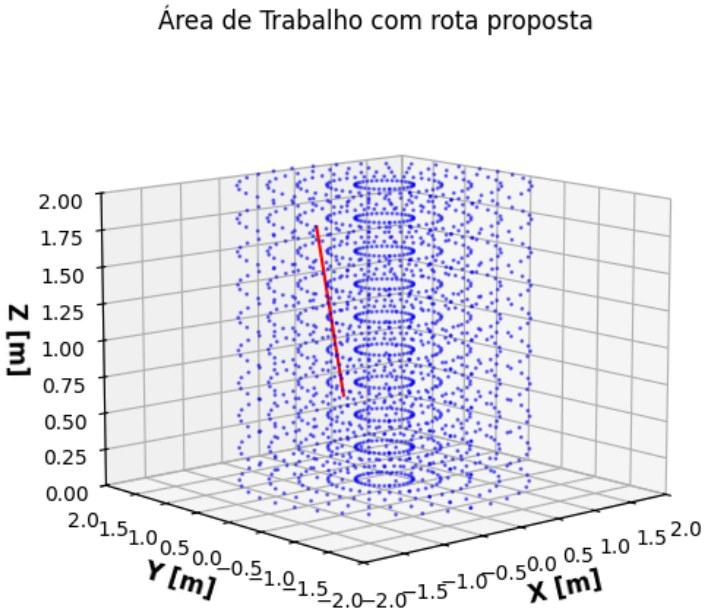


Os pontos planejados para execução da trajetória foram a posição inicial (-0.75, -0.25, 0.75) e a posição final (0.75, 0.75, 1.75). Contudo, foi possível notar que não é possível executar essa rota ao plotar a trajetória esperada no espaço de trabalho do robô. Essa rota resultaria em uma colisão da ferramenta com o eixo de rotação do robô conforme é possível observar nos gráficos abaixo.





A coordenada x da posição final foi alterada para -0.25 para que a rota pudesse ser executada. A nova rota no espaço de trabalho pode ser observada na imagem abaixo.



Implementação da cinemática direta e inversa

Para a cinemática direta foram considerados os seguintes parâmetros de Denavit-Hartenberg.

Link	θ_i	d_i	a_i	α_i
1	θ*	d_1	0	0
2	0	d_2*	a_2	-90°
3	90°	$d_f + d_3$*	a_3	0

Com isso foi possível encontrar a matriz de transformação das coordenadas da ponta da ferramenta para as coordenadas da base utilizando a matriz abaixo.

$$\begin{bmatrix} 0 & 0 & 0 & -\text{sen}(\theta)\,(d_3 + d_f) + a_2\cos(\theta) \\ 0 & -\cos(\theta) & -\text{sen}(\theta) & \cos(\theta)\,(d_3 + d_f) + a_2\text{sen}(\theta) \\ 1 & -\text{sen}(\theta) & \cos(\theta) & d_1 + d_2 - a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esses cálculos foram então implementados na função ‘fk’ da classe CylindricRobot.

Aplicando essa transformação para pontos genéricos (x,y,z) foi possível encontrar as equações para a cinemática inversa. Ela foi implementada na função ‘ik’ da classe CylindricRobot. Foram obtidas as equações:

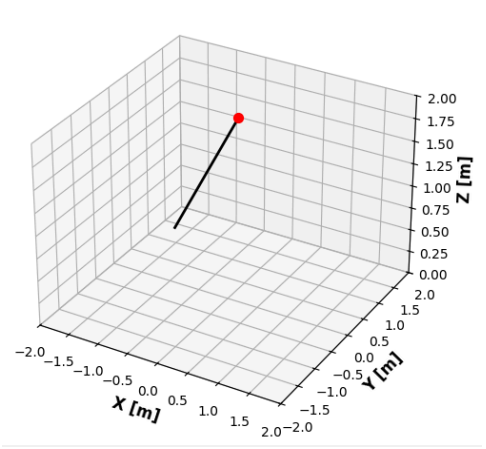
$d_2 = z - d_1 + a_3$
$y * sen(\theta) + x * cos(\theta) - a_2 = 0$
$d_3 = y * cos(\theta) - x * sen(\theta) - d_f$

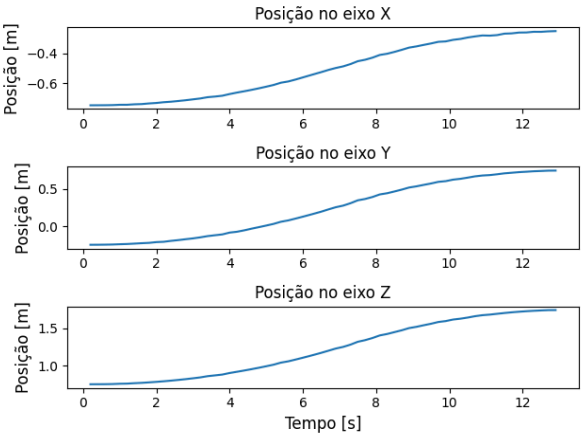
A equação para θ possui uma solução não trivial e foi resolvida utilizando a ferramenta Wolfram Alpha.

Geração de trajetória

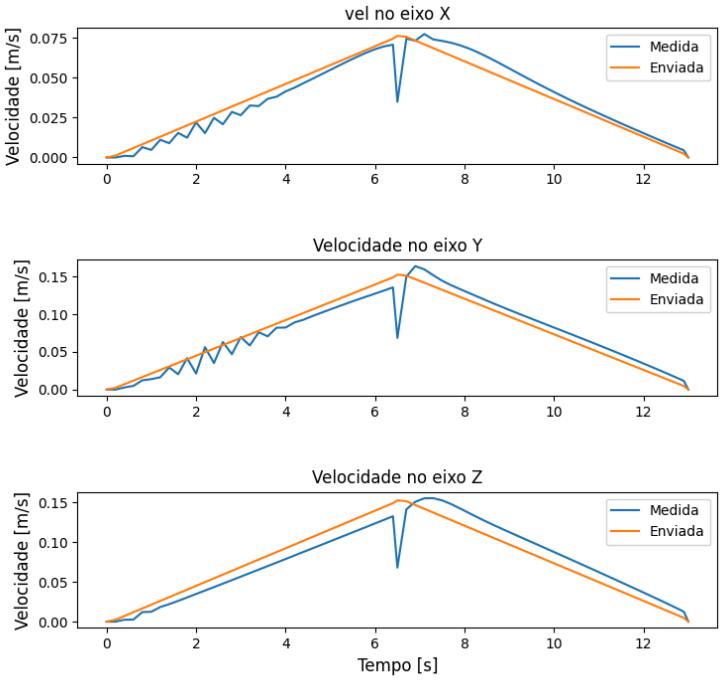
Foi gerada uma referência de trajetória utilizando a estratégia bang-bang. Para antes do switch time, $t < t_f/2$, foi utilizada a equação $q(t) = q_0 + 2(q_f - q_0)\, t^2 / (t_f^2)$ e para depois a equação $q(t) = 2\, q_0 - q_f + 4(q_f - q_0)\, t / t_f - 2(q_f - q_0)\, t^2 / (t_f^2)$, sendo q_0 a posição inicial, q_f a posição final e t_f a duração da trajetória. Para encontrar o valor do alpha, o valor de q_0 foi definido para 0 e q_f para 1.

Com isso foi possível gerar a referência de trajetória conforme as imagens abaixo. É possível notar que, conforme esperado, a trajetória em cada um dos eixos se dá por duas parábolas unidas indicando que houve uma aceleração e em seguida uma frenagem.

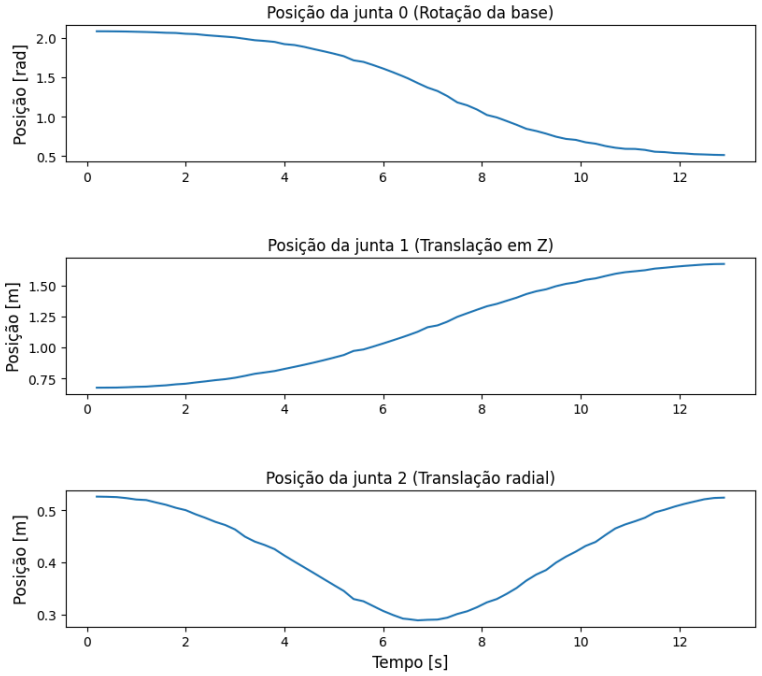




O perfil de velocidade gerado pode ser observado no gráfico abaixo.



A trajetória de cada uma das juntas pode ser observada no gráfico abaixo.



É possível notar certo nível de ruídos nos gráficos, isso se deve ao fato de não terem sido empregadas técnicas de controle ao enviar os dados de execução para as juntas. Caso as juntas sejam teletransportadas usando a função

setJointPosition do CoppeliaSim, a referência de trajetória se comporta exatamente como o esperado com posição e velocidade sem ruídos.