

UNIVERSIDADE FEDERAL DA BAHIA

ENGA73 - SISTEMAS ROBÓTICOS

Cinemática e planejamento de trajetória do UR5

Discentes:

Mateus Meneses
Rafael Queiroz
Yuri Oliveira

Docente:
André Scolari

11 de Junho de 2021



Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 4 |
| 2 | Preparação do ambiente de simulação | 4 |
| 3 | Cinemática do UR5 | 6 |
| 3.1 | Cinemática Direta | 6 |
| 3.2 | Cinemática Inversa | 7 |
| 3.3 | Validação | 10 |
| 4 | Planejamento de Trajetória | 18 |

Lista de Figuras

| | | |
|----|--|----|
| 1 | Comparação entre a adoção de links pelo método DH e modelo URDF | 5 |
| 2 | Mundo criado no simulador Gazebo | 6 |
| 3 | Possíveis posições para as juntas θ_1 do UR5 | 8 |
| 4 | Possíveis posições para as juntas θ_5 do UR5 | 9 |
| 5 | Possíveis posições para as juntas θ_3 do UR5 | 9 |
| 6 | Posicionamentos da garra para pegar a lata pela esquerda ou direita | 10 |
| 7 | Posicionamentos da garra para pegar a lata pela frente ou por cima | 11 |
| 8 | Fluxograma da validação do cálculo de cinemática direta e inversa do UR5 | 14 |
| 9 | Comparação dos erros máximos obtidos no cálculo de cinemática direta e inversa para cada conjunto i de valores de juntas | 16 |
| 10 | Erros de translação da garra integrado ao UR5 para cada conjunto i de valores de juntas | 17 |
| 11 | Erros das 6 juntas do UR5 para cada conjunto i de valores de juntas | 18 |
| 12 | Comparação dos erros máximos da cinemática direta e inversa das 6 juntas do UR5 para cada conjunto da Tabela 3 | 19 |
| 13 | Erros das 6 juntas do UR5 para cada conjunto da Tabela 3 | 19 |
| 14 | Planejamento de trajetória pelo método LSPB em cima e polinomial em baixo | 20 |
| 15 | Execução de trajetória pelo método LSPB | 21 |
| 16 | Execução de trajetória pelo método polinomial | 22 |
| 17 | Execução de trajetória sem planejamento | 22 |

Lista de Tabelas

| | | |
|---|--|----|
| 1 | Tabela de Denavit-Hartenberg modificado para o UR5 | 7 |
| 2 | Conjuntos dos valores de juntas utilizados para validação do cálculos de cinemática direta e inversa do UR5 | 12 |
| 3 | Conjuntos dos valores de juntas recortados utilizados para validação do cálculos de cinemática direta e inversa do UR5 . . . | 18 |

1 Introdução

O principal objetivo desse trabalho é simular o manipulador robótico UR5 da Universal Robots em uma missão de *pick and place* de uma lata. Para isso, se utilizou o *framework* de robótica ROS e o simulador Gazebo. O modelo do robô utilizado é o próprio modelo da fabricante, porém a preparação do ambiente do Gazebo (*world*), integração do UR5 com a garra da Robotiq e o desenvolvimento do código de cálculo das cinemáticas direta e inversa, planejamento de trajetória e de missão foram desenvolvidos pela equipe deste trabalho.

O desenvolvimento do projeto pode ser dividido em quatro partes: preparação do ambiente de simulação, cálculo de cinemática direta e inversa, planejamento de trajetória e planejamento de missão. A fase de preparação do ambiente de simulação consistiu na criação de um pacote de software utilizado para executar a simulação do manipulador UR5 utilizando a integração entre ROS e Gazebo. Para o cálculo da cinemática direta e inversa foi utilizado a linguagem de programação Python e estruturada uma classe de software com métodos para os cálculos das respectivas cinemáticas. A validação desses métodos foi realizada comparando os resultados fornecidos pelos métodos da classe com os valores fornecidos pelo ambiente de simulação. Em uma segunda etapa do projeto foi desenvolvido o planejamento de trajetória para cálculo da posição de cada junta em cada instante de tempo da missão, de forma que o robô se movesse de forma suave. Nessa etapa foi utilizada a classe de cálculos cinemáticos do UR5 para fornecer os valores necessários para a computação do planejamento de trajetória. Por fim, o planejamento da missão foi realizado para a integração de todos os sistemas na execução da tarefa.

2 Preparação do ambiente de simulação

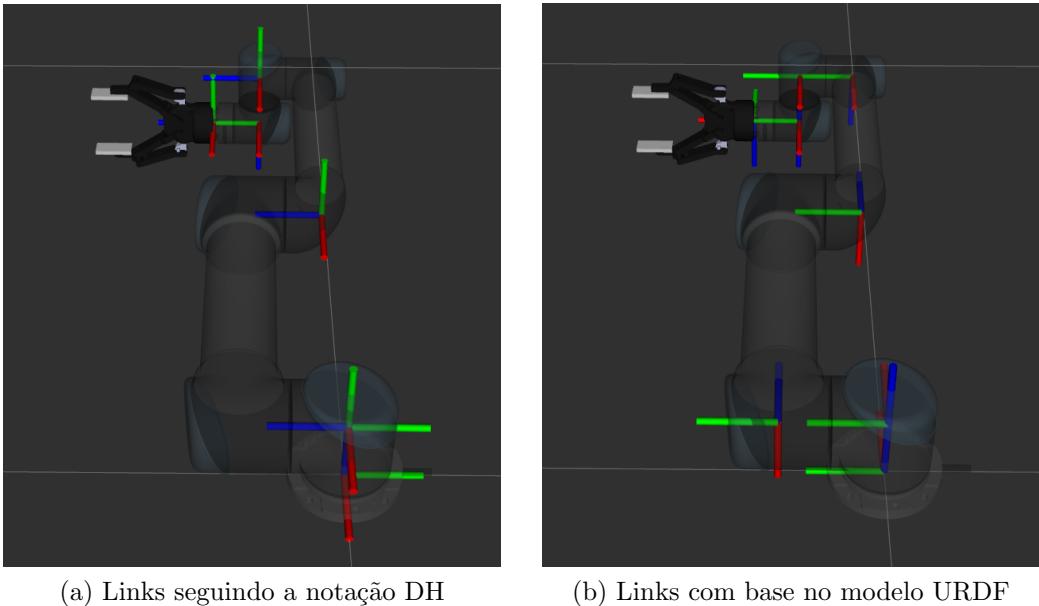
Para o ambiente de simulação, foi utilizado o pacote de software para ROS `universal_robots`¹ que reúne os modelos URDF (*Universal Robot Description Format*) e *drivers* dos manipuladores da série URX da Universal Robots. Vale salientar que este pacote não provém um *end effector* para os manipuladores e, dessa forma, foi necessário fazer a inclusão da garra 2F-140 da

¹https://github.com/ros-industrial/universal_robot

Robotiq, que também possui um pacote ROS² com drivers e modelos URDF.

Como primeiro passo para utilização do manipulador UR5 no ambiente de simulação, foi necessário tornar os *frames* dos links compatíveis com a notação de Denavit-Hartenberg (DH), já que o URDF não usa tal notação para gerar a árvore de transformação entre os links. Essa compatibilização com a notação DH foi realizada com o auxílio do pacote ROS static_transformer_publisher³. A Figura 1 mostra a distinção entre a adoção dos links pelas duas notações.

Figura 1: Comparaçao entre a adoção de links pelo método DH e modelo URDF



(a) Links seguindo a notação DH

(b) Links com base no modelo URDF

Após a compatibilização dos links, foi realizado a integração com a garra 2F-140 da Robotiq. Esta integração foi realizada unindo os dois modelos através de configuração dos arquivos URDF de ambos os modelos. Para a união foi utilizado arquivos XACRO também fornecidos pelos respectivos fabricantes. Durante a integração ocorreram alguns problemas relacionados a utilização da garra no Gazebo. Mais especificamente devido ao não correto funcionamento de um *plugin* e devido a um problema de renderização das

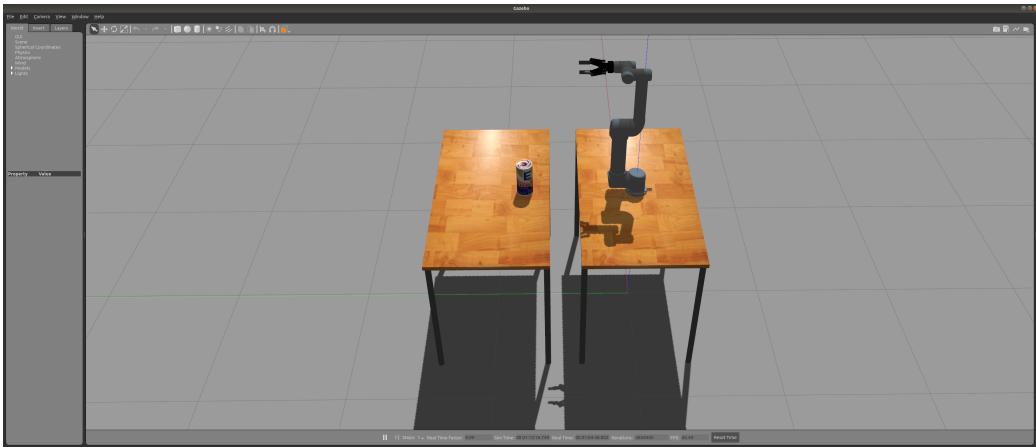
²<https://github.com/mateusmenezes95/robotiq>

³http://wiki.ros.org/tf#static_transform_publisher

meshes dentro do simulador. Esses problemas podem ser melhor entendidos em um *Pull Request*⁴ realizado como solução para os pontos mencionados.

Por fim, foi criado um mundo (*world*) no Gazebo formado por duas mesas, uma lata de cerveja e o próprio manipulador UR5 já com a garra 2F-140 integrada, como está ilustrado na Figura 2.

Figura 2: Mundo criado no simulador Gazebo



O pacote criado chama-se `ur5_simulation`⁵ e encontra-se disponível para utilização da comunidade *open source*. Na documentação do pacote é explicado como utilizá-lo.

3 Cinemática do UR5

Os tópicos a seguir descrevem os métodos de cálculo da cinemática direta e inversa, realizados conforme [Andersen, 2018], bem como foram realizadas as validações destes cálculos usando a integração ROS/Gazebo e por fim há breve discussão dos resultados obtidos.

3.1 Cinemática Direta

Para o cálculo da cinemática direta foram utilizados os parâmetros de Denavit-Hartenberg modificados, de forma a facilitar o cálculo da transformação ho-

⁴<https://github.com/L-eonor/robotiq/pull/1>

⁵https://github.com/mateusmenezes95/ur5_simulation

mogênea entre dois links consecutivos, cuja equação é representada por 1. Os parâmetros mecânicos do UR5 utilizados para o cálculo da cinemática direta foram retirados do pacote ROS da Universal Robots e encontram-se na Tabela 1.

$${}_{i-1}^1 T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \alpha_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Tabela 1: Tabela de Denavit-Hartenberg modificado para o UR5

| i | α_{i-1} | a_{i-1} | d_i | θ_i |
|-----|------------------------|------------------|------------------|------------|
| 1 | 0 | 0 | $d_1 = 0.089159$ | θ_1 |
| 2 | $\alpha_1 = 90^\circ$ | 0 | 0 | θ_2 |
| 3 | 0 | $a_2 = -0.42500$ | 0 | θ_3 |
| 4 | 0 | $a_3 = -0.39225$ | $d_4 = 0.10915$ | θ_4 |
| 5 | $\alpha_4 = 90^\circ$ | 0 | $d_5 = 0.09465$ | θ_5 |
| 6 | $\alpha_5 = -90^\circ$ | 0 | $d_6 = 0.0823$ | θ_6 |

3.2 Cinemática Inversa

No cálculo da cinemática inversa, a seguinte ordem foi seguida para obtenção dos ângulos de cada uma das juntas: θ_1 , θ_5 , θ_6 , θ_3 , θ_2 e θ_4 . As equações para obtenção desses parâmetros estão representadas, respectivamente, em 3, 4, 5, 6, 7 e 8.

$${}^0 P_5 = {}^6_0 T \begin{bmatrix} 0 \\ 0 \\ -d_6 \end{bmatrix} \quad (2)$$

$$\theta_1 = \text{atan2}\left({}^0 P_{5y}^0 P_{5x}\right) \pm \text{acos}\left(\frac{d_4}{\sqrt{{}^0 P_{5x}^2 + {}^0 P_{5y}^2}}\right) + \frac{\pi}{2} \quad (3)$$

$$\theta_5 = \pm \text{acos}\left(\frac{{}^0 P_{6x} \sin(\theta_1) - {}^0 P_{6y} \cos(\theta_1) - d_4}{d_6}\right) \quad (4)$$

$$\theta_6 = \text{atan}2\left(\frac{{}^6X_{0y}\sin(\theta_1) + {}^6Y_{0y}\cos(\theta_1)}{\sin(\theta_5)}, \frac{{}^6X_{0x}\sin(\theta_1) - {}^6Y_{0x}\cos(\theta_1)}{\sin(\theta_5)}\right) \quad (5)$$

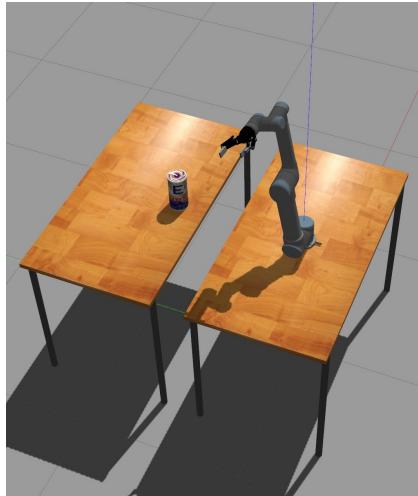
$$\theta_3 = \pm \text{acos}\left(\frac{|{}^1P_{4xz}| - a_2^2 - a_3^2}{2a_2a_3}\right) \quad (6)$$

$$\theta_2 = \text{atan}2(-{}^1P_{4z}, -{}^1P_{4x}) - \text{asin}\left(\frac{-a_3\sin(\theta_3)}{|{}^1P_{4xz}|}\right) \quad (7)$$

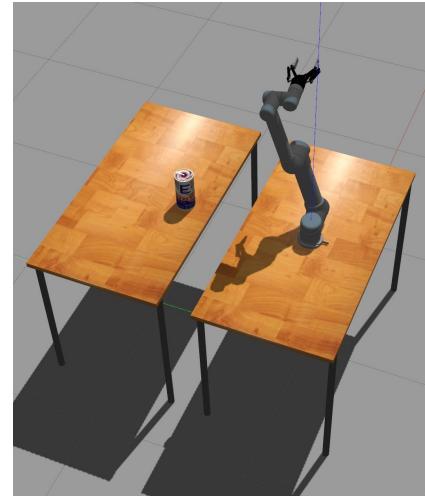
$$\theta_4 = \text{atan}2({}^3X_{4y}, {}^3X_{4x}) \quad (8)$$

Note que θ_1 , θ_5 e θ_3 assumem dois valores possíveis. Para θ_1 , isso corresponde ao ombro estar na posição esquerda ou direita (Figuras 3a e 3b). Para θ_5 , isso implica no pulso do robô estar na posição superior ou inferior (Figuras 4a e 4b). Para θ_3 , os dois valores significam a possibilidade do cotovelo estar na posição superior ou inferior (Figuras 5a e 5b). Dadas essas possibilidades, o robô passa a ter então 8 combinações de ângulos de juntas para alcançar determinada pose: $2\theta_1 \times 2\theta_5 \times 1\theta_6 \times 2\theta_3 \times 1\theta_2 \times 1\theta_4$.

Figura 3: Possíveis posições para as juntas θ_1 do UR5



(a) Ombro na posição esquerda



(b) Ombro na posição direita

Figura 4: Possíveis posições para as juntas θ_5 do UR5

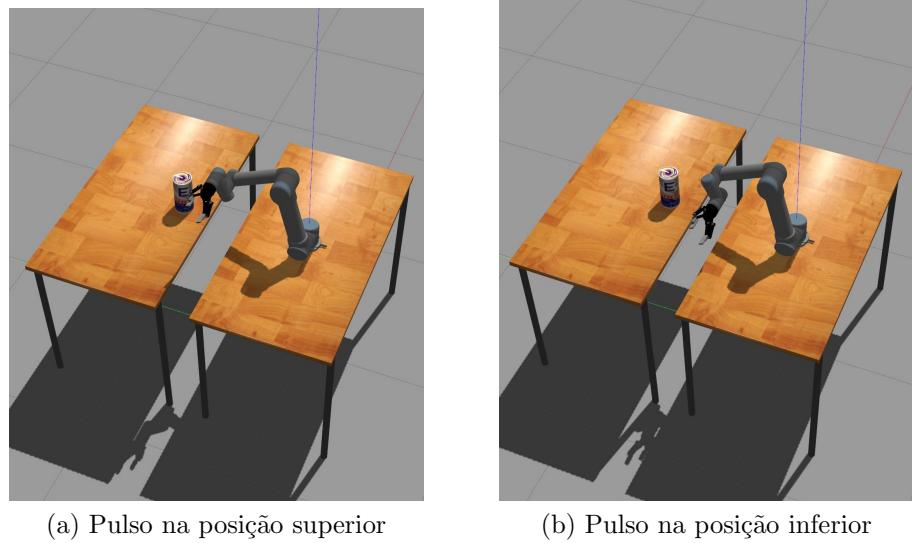
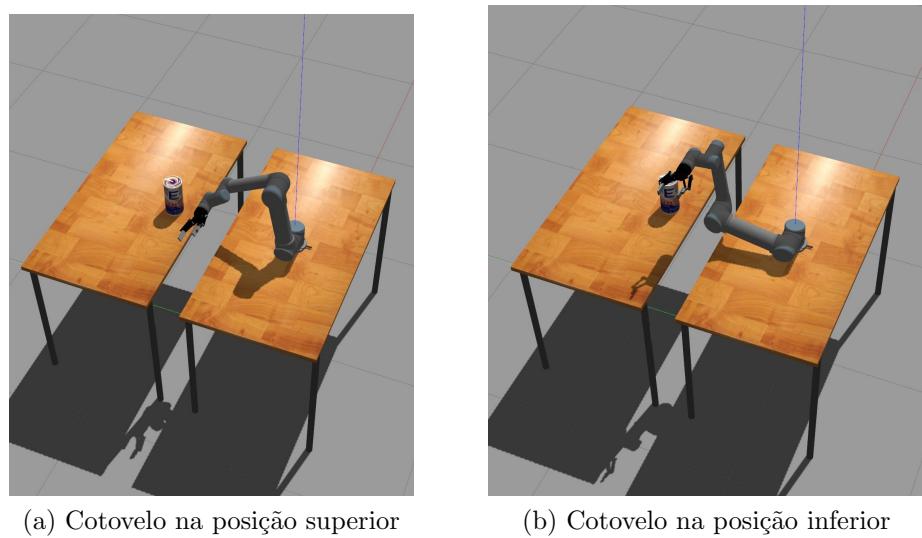


Figura 5: Possíveis posições para as juntas θ_3 do UR5



3.3 Validação

Para validar os métodos que calculam a cinemática direta e inversa do UR5, foram escolhidas 4 posições e orientações do *end effector* do manipulador. As 4 *poses* especificadas simulam 4 posicionamentos possíveis para a garra pegar a lata que se encontra na mesa à frente do manipulador, como pode ser visto nas Figuras 6 e 7. Além disso, foram adicionadas *poses* intermediárias que antecedem o movimento final da garra (*grasp*) e a volta do robô para a *pose* de *home*. Estas *poses* intermediárias foram escolhidas para evitar colisões dos componentes mecânicos do manipulador, principalmente do *end effector*, com a lata de cerveja e/ou com as mesas.

Figura 6: Posicionamentos da garra para pegar a lata pela esquerda ou direita

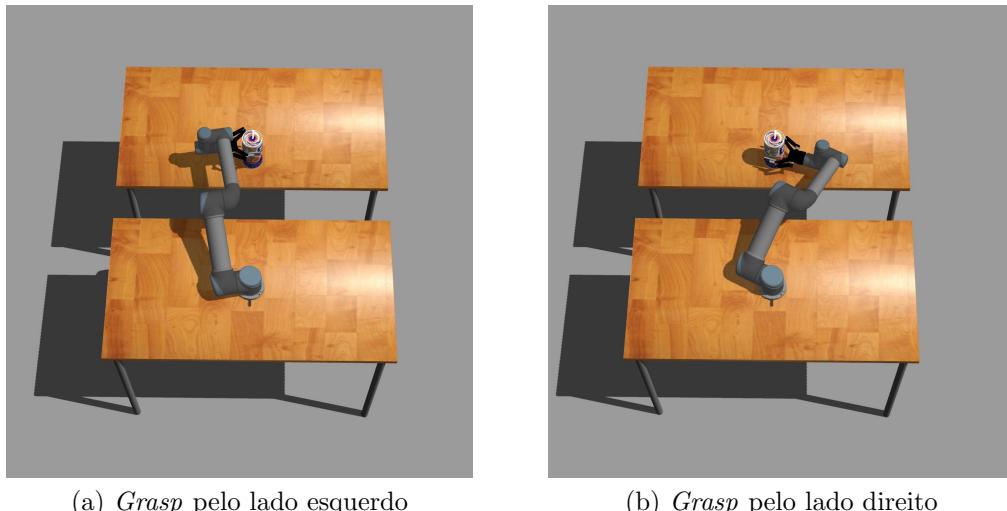
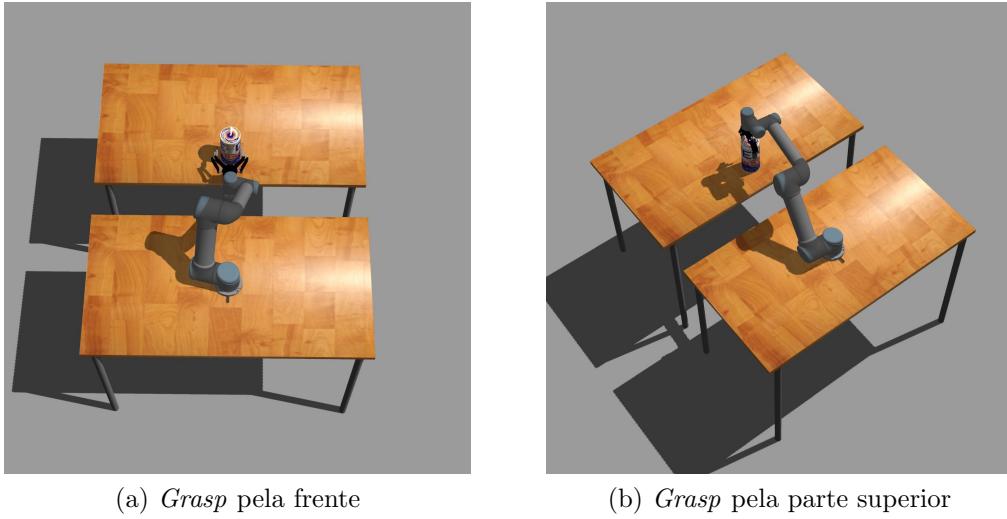


Figura 7: Posicionamentos da garra para pegar a lata pela frente ou por cima



Na Tabela 2 estão listados os valores de juntas usados na validação do cálculo de cinemática. As linhas foram preenchidas em cores para informar se uma determinada linha representa a *pose* de *grasping*, *pre/post grasping* ou *home*.

Tabela 2: Conjuntos dos valores de juntas utilizados para validação do cálculos de cinemática direta e inversa do UR5

| i | θ_1 | θ_2 | θ_3 | θ_4 | θ_5 | θ_6 |
|-----|------------|------------|------------|------------|------------|------------|
| 0 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 0.95 | -0.47 | 0.73 | 0.00 | -0.31 | 1.32 |
| 2 | 1.09 | -0.47 | 0.73 | 0.00 | -0.20 | 1.32 |
| 3 | 0.95 | -0.47 | 0.73 | 0.00 | -0.31 | 1.32 |
| 4 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |
| 5 | 1.94 | -0.45 | 0.00 | 0.00 | 3.27 | 1.57 |
| 6 | 1.94 | -0.45 | 0.45 | 0.00 | 3.27 | 1.57 |
| 7 | 1.76 | -0.45 | 0.45 | 0.00 | 3.22 | 1.57 |
| 8 | 1.94 | -0.45 | 0.45 | 0.00 | 3.27 | 1.57 |
| 9 | 1.94 | -0.45 | 0.00 | 0.00 | 3.27 | 1.57 |
| 10 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |
| 11 | 1.00 | -1.25 | 2.07 | -0.75 | 1.33 | 1.57 |
| 12 | 1.33 | -1.25 | 2.07 | -0.75 | 1.33 | 0.00 |
| 13 | 1.33 | -1.20 | 1.88 | -0.60 | 1.31 | 1.57 |
| 14 | 1.33 | -1.25 | 2.07 | -0.75 | 1.33 | 0.00 |
| 15 | 1.00 | -1.25 | 2.07 | -0.75 | 1.33 | 1.57 |
| 16 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |
| 17 | 1.44 | -0.96 | 0.75 | -1.49 | -1.57 | 0.00 |
| 18 | 1.44 | -0.84 | 0.75 | -1.46 | -1.57 | 0.00 |
| 19 | 1.44 | -0.96 | 0.75 | -1.49 | -1.57 | 0.00 |
| 20 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |

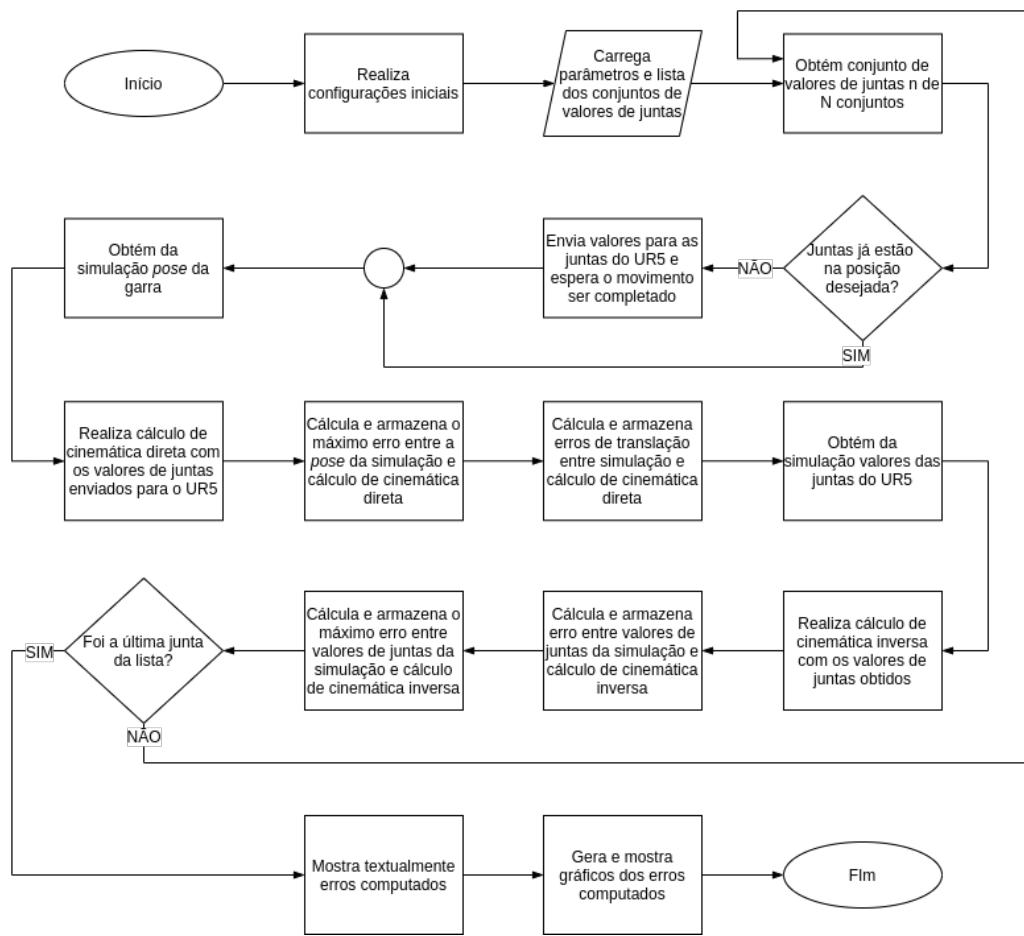
■ *home* ■ *pre-post grasping* ■ *grasping*

Para tornar este cálculo automático, foi criado uma rotina de programação⁶ usando a linguagem Python, e APIs (*Application Programming Interface*) e ferramentas fornecidas pelo *framework* ROS. Os conjuntos de valores de juntas listados na Tabela 2 foram inseridos em um arquivo de configuração⁷ com formato YAML (acrônimo recursivo para *YAML Ain't Markup Language*). Então para cada conjunto de valores de juntas lidos do arquivo de configuração, foi feito a comparação entre os valores calculados pelos métodos criados e os valores fornecidos pelo ambiente de simulação que foram obtidos por intermédio do *framework* ROS. Ao final da rotina são gerados gráficos cujo objetivo é ter uma interpretação gráfica dos erros obtidos entre as duas formas de aquisição dos valores da cinemática direta e inversa do UR5. A Figura 8 ilustra o fluxo desta rotina.

⁶https://github.com/mateusmenezes95/ur5_simulation/blob/main/scripts/joints_controller_node.py

⁷https://github.com/mateusmenezes95/ur5_simulation/blob/main/config/arm_kinematics.yaml

Figura 8: Fluxograma da validação do cálculo de cinemática direta e inversa do UR5



Para o conjunto de juntas da Tabela 2, foram gerados os gráficos de comparação ilustrados nas Figuras 11, 10 e 9. O cálculo do erro máximo e de translação para cinemática direta foram realizados conforme as Equações 9 e 10, respectivamente. Já as Equações 11 e 12 mostram, respectivamente, como foi computado o erro dos valores de juntas e erro máximo nos cálculos da cinemática inversa do UR5. Como métrica para o cálculo do erro entre os valores previstos de ângulos das juntas e os calculados pela cinemática inversa, foi utilizado a função cosseno, uma vez que, ângulos defasados entre si de 2π – o que foi notado no cálculo de θ_5 – irão sempre apresentar o mesmo valor de cosseno.

$$\begin{aligned} max_erro &= \max(|T_{s0}^6| - |T_{c0}^6|) \\ &= \max \left(\left| \begin{bmatrix} r_{s11} & r_{s12} & r_{s13} & o_{s_x} \\ r_{s21} & r_{s22} & r_{s23} & o_{s_y} \\ r_{s31} & r_{s32} & r_{s33} & o_{s_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \right| - \left| \begin{bmatrix} r_{c11} & r_{c12} & r_{c13} & o_{c_x} \\ r_{c21} & r_{c22} & r_{c23} & o_{c_y} \\ r_{c31} & r_{c32} & r_{c33} & o_{c_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \right| \right) \end{aligned} \quad (9)$$

$$\begin{aligned} o_{e0}^6 &= o_{s0}^6 - o_{c0}^6 \\ &= \begin{bmatrix} o_{s_x} \\ o_{s_y} \\ o_{s_z} \end{bmatrix} - \begin{bmatrix} o_{c_x} \\ o_{c_y} \\ o_{c_z} \end{bmatrix} \end{aligned} \quad (10)$$

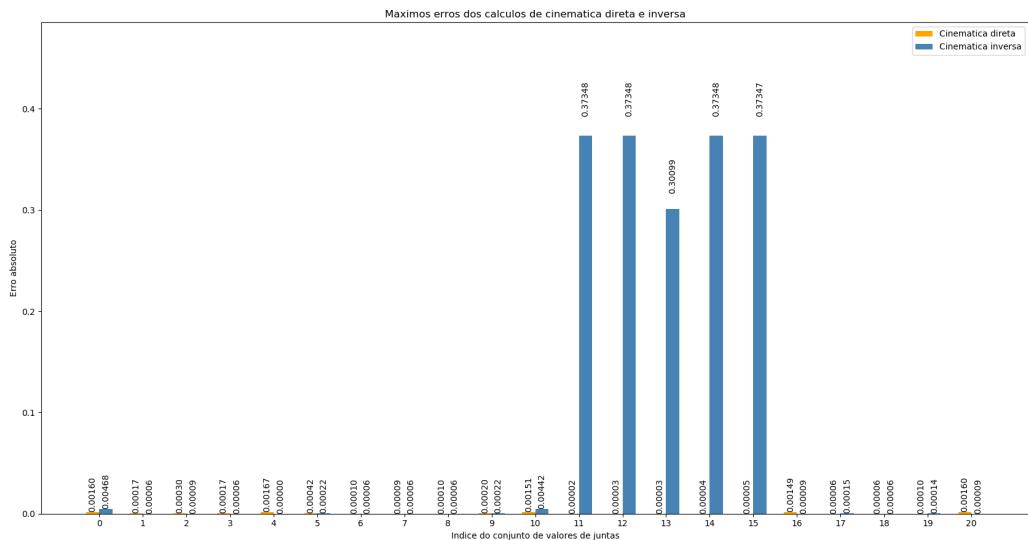
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix} = \begin{bmatrix} \cos(\theta_{s_1}) \\ \cos(\theta_{s_2}) \\ \cos(\theta_{s_3}) \\ \cos(\theta_{s_4}) \\ \cos(\theta_{s_5}) \\ \cos(\theta_{s_6}) \end{bmatrix} - \begin{bmatrix} \cos(\theta_{c_1}) \\ \cos(\theta_{c_2}) \\ \cos(\theta_{c_3}) \\ \cos(\theta_{c_4}) \\ \cos(\theta_{c_5}) \\ \cos(\theta_{c_6}) \end{bmatrix} \quad (11)$$

$$max_theta_e = \max \left(\left| \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix} \right| \right) \quad (12)$$

onde os sobrescritos s e c representam os valores obtidos da simulação e do cálculo próprio, respectivamente.

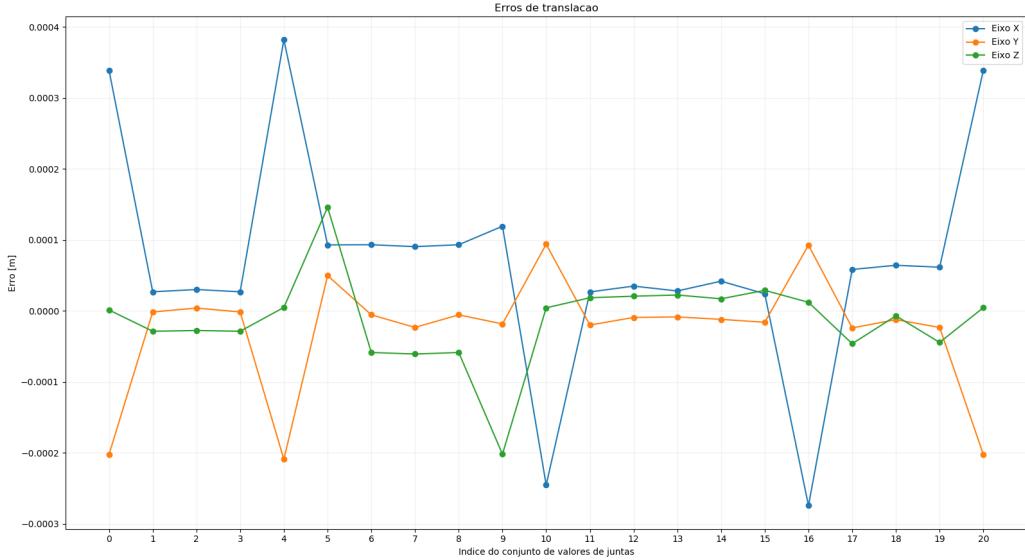
Observa-se na Figura 9 que o maior erro absoluto obtido para o cálculo da cinemática direta foi de 0.00167 para o conjunto 4. Observa-se que os conjuntos que mais apresentaram erros foram os de *home* números 0, 4, 10, 16 e 20. Uma hipótese para estes erros é o ruído que a simulação fornece quando as juntas estão na posição zero e, portanto, seria embutido no cálculo de erro um *bias* com um valor milesimal.

Figura 9: Comparação dos erros máximos obtidos no cálculo de cinemática direta e inversa para cada conjunto i de valores de juntas



O erro de translação do cálculo da Cinemática direta do UR5 pode ser visualizado na Figura 10. Nota-se que o erro máximo foi de aproximadamente 0,4 mm para o conjunto 4. Observa-se novamente também que os maiores erros estão nos conjuntos de *home* (0, 4, 10, 16 e 20). Entretanto, vale salientar que em uma aplicação real o erro na posição do *end effector* pode ser desconsiderado, já que nessa posição o manipulador não executa nenhuma tarefa de precisão.

Figura 10: Erros de translação da garra integrado ao UR5 para cada conjunto i de valores de juntas



Na Figura 11 verifica-se que do conjunto 11 ao 15 há um erro quase que constante para a junta θ_5 , modificando apenas no conjunto 13 (conjunto de *grasp* conforme Tabela 2). Este erro é explicado devido principalmente a consideração da posição do UR5. Conforme explicado em 3.2 o valor de θ_5 é influenciado pela posição do pulso do robô, que para o resultado da Figura 10 foi considerada a posição superior. Fazendo um recorte da Tabela 2, gerando a nova Tabela 3 e considerando agora o pulso na posição inferior, obtemos os gráficos mostrados nas Figuras 12 e 13. Verifica-se agora que os erros se aproximam de zero para a maioria das juntas, com exceção para os conjuntos de *home*. A diferença dos resultados entre as Figuras 11 e 13 pode ser explicada devido as múltiplas soluções encontradas no cálculo da cinemática inversa para o mesmo valor de junta, conforme explicado no tópico 3.2.

Figura 11: Erros das 6 juntas do UR5 para cada conjunto i de valores de juntas

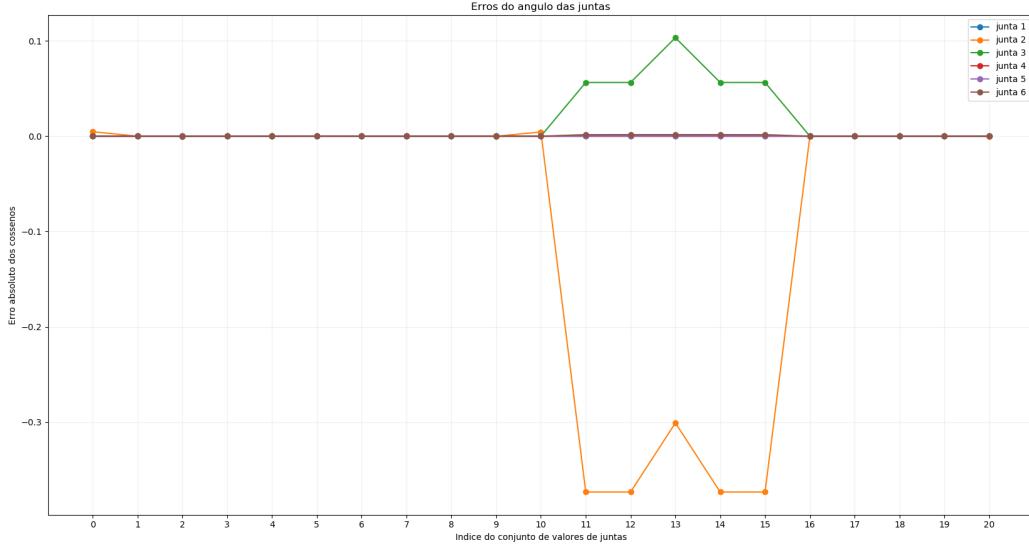


Tabela 3: Conjuntos dos valores de juntas recortados utilizados para validação do cálculos de cinemática direta e inversa do UR5

| i | θ_1 | θ_2 | θ_3 | θ_4 | θ_5 | θ_6 |
|-----|------------|------------|------------|------------|------------|------------|
| 0 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 1.00 | -1.25 | 2.07 | -0.75 | 1.33 | 1.57 |
| 2 | 1.33 | -1.25 | 2.07 | -0.75 | 1.33 | 0.00 |
| 3 | 1.33 | -1.20 | 1.88 | -0.60 | 1.31 | 1.57 |
| 4 | 1.33 | -1.25 | 2.07 | -0.75 | 1.33 | 0.00 |
| 5 | 1.00 | -1.25 | 2.07 | -0.75 | 1.33 | 1.57 |
| 6 | 0.00 | -1.57 | 0.00 | 0.00 | 0.00 | 0.00 |

home
 pre-post grasping
 grasping

4 Planejamento de Trajetória

O planjedor de trajetória tem a função de mapear a posição de cada junta durante o movimento. Esse componente deve ser utilizado afim de se suavizar o movimento dos motores para se evitar colisões e *stress* desnecessário das

Figura 12: Comparação dos erros máximos da cinemática direta e inversa das 6 juntas do UR5 para cada conjunto da Tabela 3

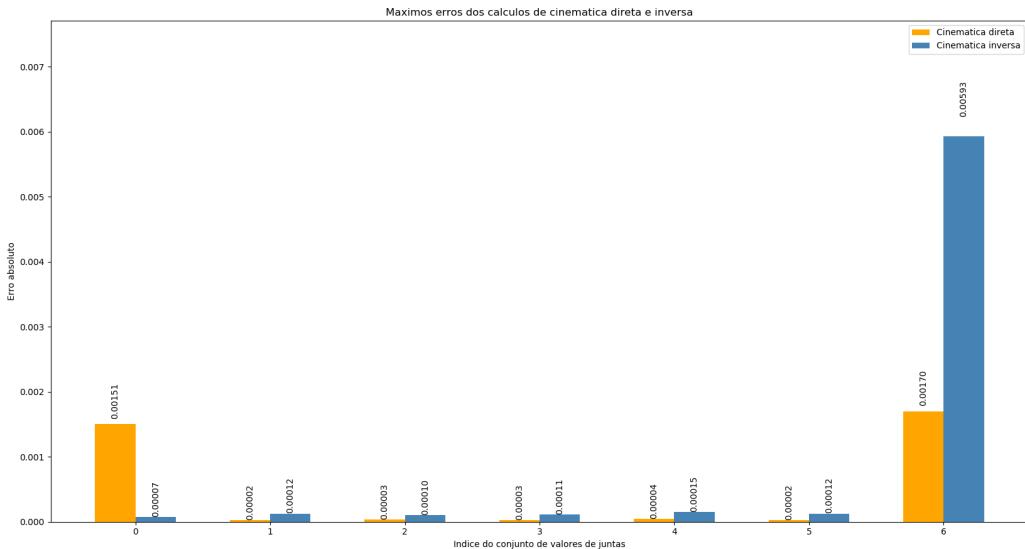
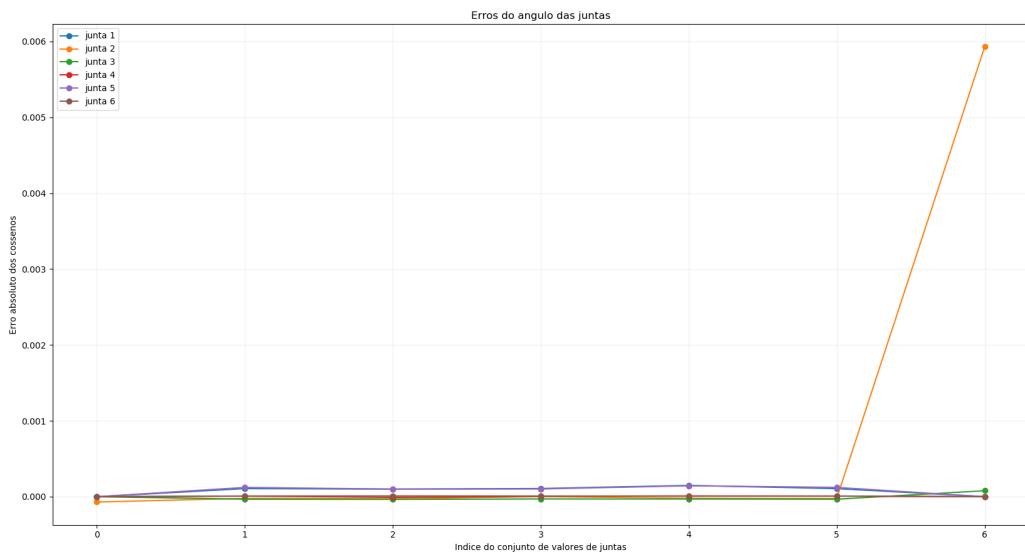


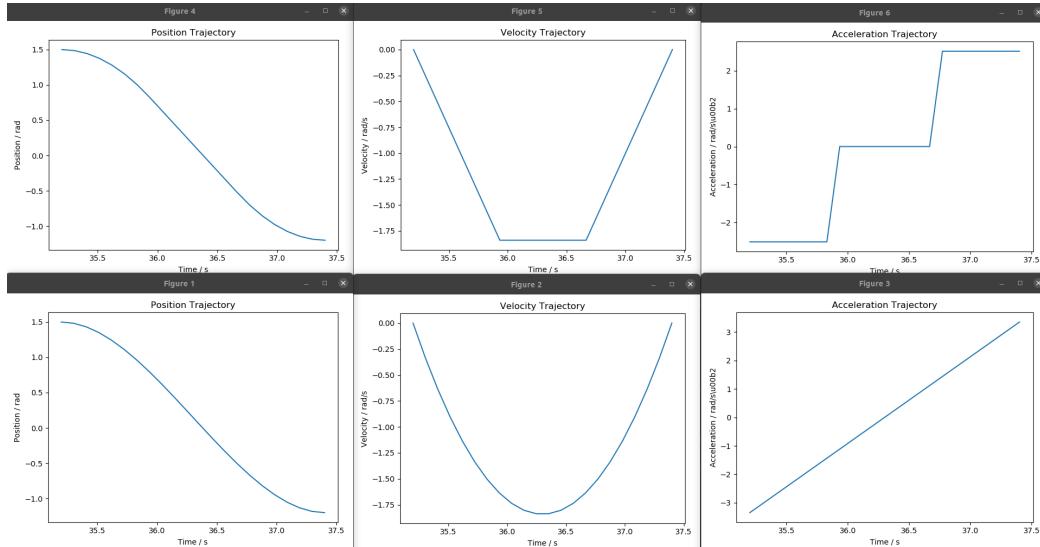
Figura 13: Erros das 6 juntas do UR5 para cada conjunto da Tabela 3



juntas. Nesse trabalho, dois planejadores foram desenvolvidos e as trajetórias planejadas e executadas foram comparadas. Os planejadores escolhidos foram o *Linear Segment Parabolic Blend* (LSPB), o qual se gera uma trajetória trapezoidal com velocidade e aceleração constantes, e o *Cubic Polynomial* (polinomial), o qual se gera uma função polinomial cúbica de posição para ser seguida pelas juntas.

A implementação foi feita com base no livro do Mark Spong, de forma que ao se instanciar o planejador, as curvas de posição, velocidade e aceleração são calculadas. Na imagem abaixo, pode ser visto a comparação das trajetórias calculadas com o lspb em cima e o polinomial em baixo. Se definiu a velocidade inicial e final do planejador polinomial como zero, visto que o LSPB já tem essa restrição, então não seria uma comparação muito válida caso as condições iniciais e finais sejam diferentes. Para o LSPB, se definiu uma velocidade arbitrária de -184 rad por segundo. Por fim, para ambos os planjadores, se determinou a posição inicial e final como 1.5 rad e -1.2 rad, e o tempo inicial e final como 35.2 e 37.4 segundos.

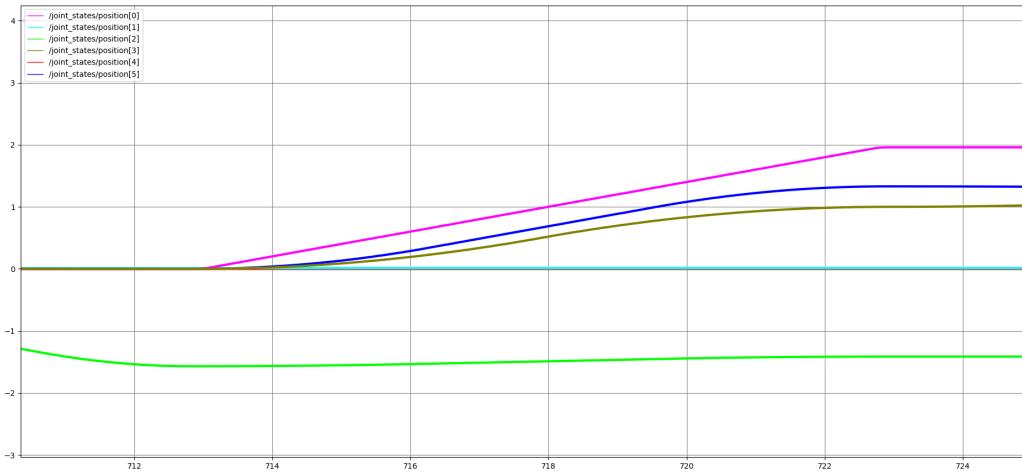
Figura 14: Planejamento de trajetória pelo método LSPB em cima e polinomial em baixo



Após a validação dos algoritmos de planejamento, estes foram testados em simulação para o deslocamento do UR5. Abaixo, é possível visualizar os gráficos do deslocamento de cada uma das 6 juntas utilizando diferentes

estratégias. A Figura 15 representa o movimento das juntas utilizando LSPB, Figura 16 representa o movimento das juntas utilizando o planejamento polinomial, por fim, a Figura 17 representa o movimento do braço sem nenhum planejador, somente com o controle de posição das juntas.

Figura 15: Execução de trajetória pelo método LSPB



Dessa forma, se conclue esse trabalho, no qual foi possível realizar a missão de se aproximar de uma lata, utilizando cinemática inversa e planejador de trajetórias próprios.

Referências

- [Andersen, 2018] Andersen, R. S. (2018). Kinematics of a UR5. Technical report, Aalborg University, May 2018.

Figura 16: Execução de trajetória pelo método polinomial

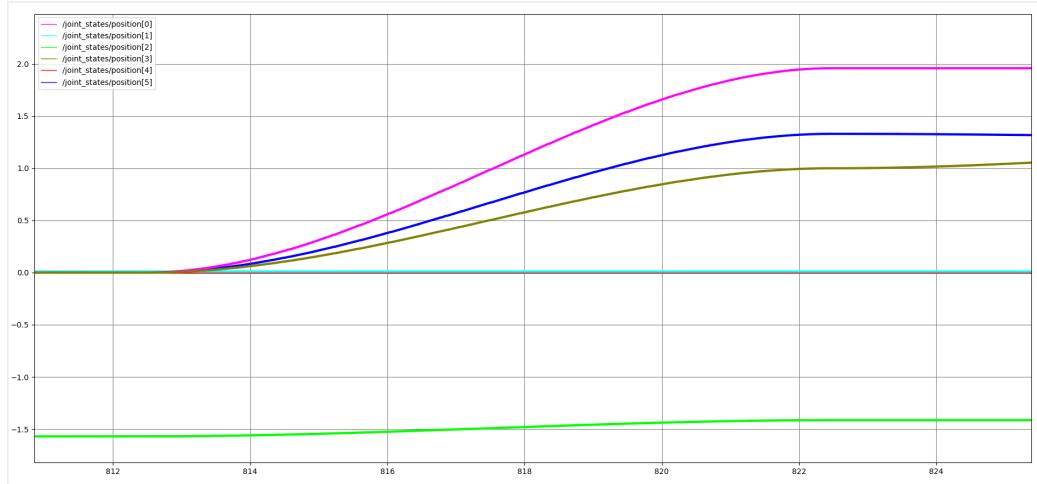


Figura 17: Execução de trajetória sem planejamento

