



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**GUSTAVO PENAFORTE BRITO**

**ANÁLISE DE CONTROLE DE UM BRAÇO ROBÓTICO COM 2 GRAUS DE  
LIBERDADE ATRAVÉS DE SIMULAÇÃO COMPUTACIONAL**

**FORTALEZA**

**2022**

GUSTAVO PENAFORTE BRITO

ANÁLISE DE CONTROLE DE UM BRAÇO ROBÓTICO COM 2 GRAUS DE LIBERDADE  
ATRAVÉS DE SIMULAÇÃO COMPUTACIONAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Wilkley Bezerra  
Correia

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

B875a Brito, Gustavo Penaforte.

Análise de controle de um braço robótico com 2 graus de liberdade através de simulação / Gustavo Penaforte Brito. – 2022.  
81 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2022.

Orientação: Prof. Dr. Wilkley Bezerra Correia.

1. Robótica. 2. Controle PD. 3. Manipulador robótico. 4. Incerteza paramétrica. I. Título.

CDD 621.3

---

GUSTAVO PENAFORTE BRITO

ANÁLISE DE CONTROLE DE UM BRAÇO ROBÓTICO COM 2 GRAUS DE LIBERDADE  
ATRAVÉS DE SIMULAÇÃO COMPUTACIONAL

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia Elétrica do  
Centro de Tecnologia da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. Wilkley Bezerra Correia (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Diego de Sousa Madeira  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Dalton de Araújo Honório  
Universidade Federal do Ceará (UFC)

A Deus;

Aos meus pais e toda minha família;

Ao meu professor e orientador Wilkley Bezerra;

A todos que contribuíram com meu desenvolvimento e me fizeram chegar até aqui.

## AGRADECIMENTOS

A Deus, pela minha vida e por me guiar durante toda minha trajetória, sempre me dando saúde, força, disposição e sabedoria para chegar até aqui. Sem Ele, sei que nada teria sido possível.

Aos meus pais, Francisco Carlos e Silce Maria, que me criaram com muito amor, carinho e sempre me incentivaram e apoiaram principalmente nas horas mais difíceis e nunca permitiram que eu desistisse dos meus sonhos. A vocês, todo o meu amor e gratidão, vocês são tudo em minha vida.

À minha avó e segunda mãe, Ceci, que sempre fez de tudo para me ver e fazer feliz e me criou desde pequeno com todo amor e carinho. Agradeço por todos os seus ensinamentos e cuidado.

À minha avó, Olga, sempre lembrando de mim em suas orações e torcendo pela felicidade e meu sucesso.

Aos meus tios e tias, em especial Annibal, Syrleine, Sara, Selene e Ricardo, que sempre me ajudaram e pude contar com todos em vários momentos que foram muito difíceis em minha vida.

À minha prima Cristina, pelos incentivos, apoios e sempre estar com o coração aberto para me ouvir e ajudar. Agradeço muito sua companhia e amizade.

Ao meu professor e orientador, Wilkley Bezerra, pelos ensinamentos que permitiram meu desenvolvimento profissional e pelas orientações que puderam fazer deste trabalho possível.

À equipe de trabalho do braço robótico, João, Iury, Livia e Gabriel, que contribuíram com o desenvolvimento desse projeto e sei que ainda haverá uma longa caminhada pela frente.

Aos meus grandes amigos Leonardo, Lucas e Millena, que só tenho agradecer por todos os nossos encontros no pátio do prédio, saídas, conversas e por nossa amizade de longa data.

À Diretoria Executiva de 2019.2 da Tecsys Jr, Davi, Letícia e Alice, pois vocês foram um grande apoio à mim durante a gestão e não só contribuíram com minha formação profissional, mas também na minha formação pessoal. Sou muito grato a vocês por tudo.

Ao meu amigo Daniel, que sempre me acompanhou nessa caminhada universitária desde que ingressou na universidade. Seu companheirismo foi essencial durante todo esse período.

À minha amiga Rebeca, pela sua amizade, apoio e pelas conversas até de madrugada.  
Muito obrigado por sua presença em minha vida.

Ao meu amigo Victor, por nossa amizade de longa data e por todos os nossos momentos juntos.

“Não ambicione ser considerado uma pessoa importante. Deseje apenas ser admirado como uma pessoa benevolente”

(Meishu-Sama)



## RESUMO

Tendo em vista a acessibilidade de sistemas robóticos em ambientes industriais, acadêmicos e residenciais, surge uma necessidade constante de melhorias de estudo de modelos matemáticos e de controladores para que esses sistemas possam desempenhar sua função de forma adequada. Com isso, esse trabalho consiste na modelagem matemática e computacional de um manipulador robótico, apresentando a Equação de Corpo Rígido e técnicas de simplificações para esse tipo de sistema. O braço robótico escolhido possui 3 graus de liberdade (DOF) e 2 links. Será realizado uma simplificação fixando o grau de liberdade da base e o braço robótico apresentará 2 graus de liberdade. A modelagem matemática é realizada a partir das equações de dinâmicas já conhecidas e a modelagem computacional através de uma *Toolbox* para sistemas robóticos em Python. O trabalho apresenta o projeto implementação na *Toolbox* de um controlador PD que satisfaz às condições de operação do sistema robótico. Ao aplicar incertezas paramétricas na massa da carga para diferentes trajetórias, percebe-se uma mudança de desempenho no sistema controlado. Por fim, comparam-se os resultados obtidos antes e depois de implementar a lei de controle, assim como a aplicação das incertezas paramétricas de massa.

**Palavras-chave:** Robótica. Controle PD. Manipulador Robótico. Incerteza Paramétrica.

## ABSTRACT

Considering the accessibility of robotic systems in industrial, academic and residential environments, there is a constant need for improvements in the study of mathematical models and controllers so that these systems can perform their function properly. Thus, this work consists of the mathematical and computational modeling of a robotic manipulator, presenting the Rigid Body Equation and simplification techniques for this type of system. The robotic arm chosen has 3 degrees of freedom (DOF) and 2 links. A simplification will be carried out by fixing the degree of freedom of the base and the robotic arm will have 2 degrees of freedom. The mathematical modeling will be carried out from the already known dynamics equations and the computational modeling will be carried out through a Toolbox for robotic systems that will be applied in Python. A PD controller will be designed and implemented that is robust and satisfies the operating conditions of the robotic system by applying parametric uncertainties in the load mass for different trajectories. Finally, the results obtained before and after implementing the control law can be compared, as well as the application of parametric mass uncertainties.

**Keywords:** Robotics. Control PD. Robotic Manipulator. Parametric Uncertainties.

## LISTA DE FIGURAS

Figura 1 – Representação de um braço de 2 Links . . . . .	18
Figura 2 – Representação de um braço de 1 Link . . . . .	21
Figura 3 – Representação de um braço de 2 Links . . . . .	22
Figura 4 – Diagrama de Blocos referente ao Sistema Robótico . . . . .	29
Figura 5 – Posição dos polos e resposta do sistema em relação ao tempo . . . . .	31
Figura 6 – Posição dos polos e resposta do sistema em relação ao tempo . . . . .	31
Figura 7 – Posição dos polos e resposta do sistema em relação ao tempo . . . . .	32
Figura 8 – Diagrama de Blocos do Modelo Massa Mola Controlado . . . . .	34
Figura 9 – Diagrama de Blocos do Modelo Massa Mola Controlado Ajustado . . . . .	35
Figura 10 – Braço de Bancada do Laboratório de Sistemas Motrizes (Lamotriz) UFC . . . . .	36
Figura 11 – Braço de Bancada desenhado no AutoCAD . . . . .	37
Figura 12 – Fluxograma de Cálculo da Trajetória seguida pelo Braço Robótico . . . . .	40
Figura 13 – Fluxograma de Cálculo da Trajetória seguida pelo Braço Robótico . . . . .	41
Figura 14 – Trajetória 1 para o braço robótico . . . . .	42
Figura 15 – Trajetória 2 para o braço robótico . . . . .	42
Figura 16 – Torque Calculado pela RNE para o Link 1 . . . . .	43
Figura 17 – Torque Calculado pela RNE para o Link 2 . . . . .	44
Figura 18 – Torque Calculado pela Equação de Corpo Rígido para o Link 1 . . . . .	44
Figura 19 – Torque Calculado pela Equação de Corpo Rígido para o Link 2 . . . . .	45
Figura 20 – Comparação do Torque Calculado pela RNE e Equação de Corpo Rígido no Link 1 . . . . .	45
Figura 21 – Comparação do Torque Calculado pela RNE e Equação de Corpo Rígido no Link 2 . . . . .	46
Figura 22 – Fluxograma Cálculo de Aceleração, Velocidade e Posição do Braço Robótico	47
Figura 23 – Aceleração do Braço para Trajetória 1 . . . . .	48
Figura 24 – Velocidade do Braço para Trajetória 1 . . . . .	49
Figura 25 – Posição do Braço para Trajetória 1 . . . . .	49
Figura 26 – Posição Angular do Braço Robótico em Malha aberta sem carga (Trajetória 1)	52
Figura 27 – Posição Angular do Braço Robótico em Malha aberta com carga de 50g (Trajetória 1) . . . . .	52
Figura 28 – Posição Angular do Braço Robótico em Malha Fechada sem carga (Trajetória 1)	53

Figura 29 – Posição Angular do Braço Robótico em Malha Fechada com carga de 50g (Trajetória 1) . . . . .	54
Figura 30 – Posição Angular do Braço Robótico em Malha Fechada com carga de 100g (Trajetória 1) . . . . .	54
Figura 31 – Posição Angular do Braço Robótico em Malha Fechada com carga de 250g (Trajetória 1) . . . . .	55
Figura 32 – Torque de Feedback para o Braço Robótico sem carga (Trajetória 1) . . . . .	56
Figura 33 – Torque MA vs Torque MF do Braço Robótico sem carga (Trajetória 1) . . . . .	56
Figura 34 – Torque de Feedback para o Braço Robótico com carga 50g (Trajetória 1) . . . . .	57
Figura 35 – Torque MA vs Torque MF do Braço Robótico com carga 50g (Trajetória 1) . . . . .	58
Figura 36 – Torque de Feedback para o Braço Robótico com carga 100g (Trajetória 1) . . . . .	58
Figura 37 – Torque MA vs Torque MF do Braço Robótico com carga 100g (Trajetória 1) . . . . .	59
Figura 38 – Torque de Feedback para o Braço Robótico com carga 250g (Trajetória 1) . . . . .	59
Figura 39 – Torque MA vs Torque MF do Braço Robótico com carga 250g (Trajetória 1) . . . . .	60
Figura 40 – Posição Angular do Braço Robótico em Malha aberta sem carga (Trajetória 2) . . . . .	61
Figura 41 – Posição Angular do Braço Robótico em Malha aberta com carga de 50g (Trajetória 2) . . . . .	61
Figura 42 – Posição Angular do Braço Robótico em Malha Fechada sem carga (Trajetória 2) . . . . .	62
Figura 43 – Posição Angular do Braço Robótico em Malha Fechada com carga de 50g (Trajetória 2) . . . . .	63
Figura 44 – Posição Angular do Braço Robótico em Malha Fechada com carga de 100g (Trajetória 2) . . . . .	63
Figura 45 – Posição Angular do Braço Robótico em Malha Fechada com carga de 250g (Trajetória 2) . . . . .	64
Figura 46 – Torque de Feedback para o Braço Robótico sem carga (Trajetória 2) . . . . .	64
Figura 47 – Torque MA vs Torque MF do Braço Robótico sem carga (Trajetória 2) . . . . .	65
Figura 48 – Torque MA vs Torque MF do Braço Robótico sem carga para o Link 1 (Trajetória 2) . . . . .	65
Figura 49 – Torque MA vs Torque MF do Braço Robótico sem carga para o Link 2 (Trajetória 2) . . . . .	66
Figura 50 – Torque de Feedback para o Braço Robótico com carga de 50g (Trajetória 2) . . . . .	66

Figura 51 – Torque MA vs Torque MF do Braço Robótico com carga de 50g para o Link 1(Trajectoria 2) . . . . .	67
Figura 52 – Torque MA vs Torque MF do Braço Robótico com carga de 50g para o Link 2 (Trajetória 2) . . . . .	67
Figura 53 – Torque de Feedback para o Braço Robótico com carga de 100g (Trajetória 2)	68
Figura 54 – Torque MA vs Torque MF do Braço Robótico com carga de 100g para o Link 1(Trajectoria 2) . . . . .	68
Figura 55 – Torque MA vs Torque MF do Braço Robótico com carga de 100g para o Link 2(Trajectoria 2) . . . . .	69
Figura 56 – Torque de Feedback para o Braço Robótico com carga de 250g (Trajetória 2)	70
Figura 57 – Torque MA vs Torque MF do Braço Robótico com carga de 250g para o Link 1(Trajectoria 2) . . . . .	70
Figura 58 – Torque MA vs Torque MF do Braço Robótico com carga de 250g para o Link 2 (Trajetória 2) . . . . .	71

## LISTA DE TABELAS

Tabela 1 – Medições do Braço de Bancada . . . . .	37
Tabela 2 – Parâmetros Estimados do Braço de Bancada . . . . .	50

## LISTA DE ABREVIATURAS E SIGLAS

Lamotriz	Laboratório de Sistemas Motrizes
DOF	Graus de Liberdade ou <i>Degrees of Freedom</i>
RNE	Equação Recursiva de Newton-Euler
PD	Proporcional Derivativo
PID	Proporcional Integral Derivativo
LPV	Parâmetros Lineares Variantes

## LISTA DE SÍMBOLOS

$\tau$	Torque Mecânico
$m$	Massa de um Corpo
$b$	Coefficiente de Atrito
$k$	Coefficiente Elástico
$K_p$	Constante de Proporcionalidade
$K_v$	Constante de Velocidade
$q$	Posição de uma junta robótica
$\dot{q}$	Velocidade de uma junta robótica
$\ddot{q}$	Aceleração de uma junta robótica
$\tau_a$	Torque de um Atuador
$T_s$	Tempo de Amostragem



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Motivação do Trabalho</b>	<b>18</b>
<b>1.2</b>	<b>Justificativa do Trabalho</b>	<b>19</b>
<b>1.3</b>	<b>Objetivos</b>	<b>19</b>
<b>1.4</b>	<b>Organização do Trabalho</b>	<b>19</b>
<b>2</b>	<b>MODELAGEM MATEMÁTICA DO BRAÇO ROBÓTICO</b>	<b>21</b>
<b>2.1</b>	<b>Modelagem de um sistema robótico</b>	<b>21</b>
<b>2.2</b>	<b>Elementos da Equação de Corpo Rígido</b>	<b>24</b>
<b>2.3</b>	<b>Simplificação do Modelo Matemático para um Braço de 2 Links</b>	<b>25</b>
<b>2.4</b>	<b>Modelo Massa-Mola</b>	<b>27</b>
<b>3</b>	<b>CONTROLE EM SISTEMAS ROBÓTICOS</b>	<b>28</b>
<b>3.1</b>	<b>Controle de um Sistema Robótico</b>	<b>28</b>
<b>3.2</b>	<b>Controle de um Sistema Massa-Mola</b>	<b>30</b>
<b>4</b>	<b>DESCRIÇÃO DO BRAÇO DE BANCADA</b>	<b>36</b>
<b>4.1</b>	<b>Estimativa dos Momentos de Inércia</b>	<b>37</b>
<b>4.2</b>	<b>Definição do Modelo de Bancada em Código Python</b>	<b>39</b>
<b>4.3</b>	<b>Cálculo da Trajetória realizada pelo Braço Robótico</b>	<b>40</b>
<b>4.4</b>	<b>Definição de Trajetórias em Código Python</b>	<b>41</b>
<b>4.5</b>	<b>Calculo da RNE em Código Python</b>	<b>42</b>
<b>4.6</b>	<b>Cálculo dos ângulos a partir do Torque Calculado</b>	<b>46</b>
<b>4.7</b>	<b>Controle PD</b>	<b>50</b>
<b>5</b>	<b>RESULTADOS</b>	<b>51</b>
<b>5.1</b>	<b>Análise da Trajetória 1</b>	<b>51</b>
<b>5.2</b>	<b>Análise da Trajetória 2</b>	<b>60</b>
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>72</b>
	<b>REFERÊNCIAS</b>	<b>73</b>
	<b>APÊNDICES</b>	<b>74</b>
	<b>APÊNDICE A – CÓDIGOS DESENVOLVIDOS</b>	<b>74</b>

## 1 INTRODUÇÃO

O termo "robô" foi utilizado pela primeira vez no ano de 1920 em uma peça de teatro, em que o papel foi feito para representar uma pessoa artificial. O termo foi se popularizando e expandindo, e dessa forma, veio o incentivo de trabalhos para produzirem sistemas similares ao que já vinha sendo conhecido na ficção científica (CORKE, 2017).

Um manipulador robótico é um sistema composto por links, sendo estes conectados por juntas. Cada junta está atrelada a conexão de dois links. Além disso, todo manipulador robótico possui uma quantidade  $n$  de Graus de Liberdade ou *Degrees of Freedom* (DOF). No caso de um manipulador robótico, o seu número de graus de liberdade é sempre determinado pelo número de juntas no sistema (SPONG *et al.*, 2006).

Os manipuladores robóticos vêm sendo largamente utilizados em parques industriais, pois sua utilização garante mais eficiência, produção, velocidade e precisão de diversos processos que antes eram atribuídas ao ser humano, o que resultava elevados custos, carga de trabalho e esforços repetitivos (CRAIG, 2013) (GAIDHANE *et al.*, 2019). Outro motivo que justifica a grande utilização de sistemas robóticos no campo industrial atualmente é sua acessibilidade financeira que vem aumentando a cada ano desde a década de 1990. (CRAIG, 2013).

Há diversas formas de classificar os manipuladores robóticos e uma delas é pela função que o mesmo desempenha. Os "robôs de manufatura" são os robôs utilizados em indústrias e realizam processos relacionados à produção. Já os "robôs de serviços" são robôs que realizam serviços de apoio ao ser humano, sejam de limpeza, reabilitação e etc. Os "robôs de campo" são utilizados em aplicações em campo e relacionadas ao meio ambiente, sendo mapeamento do local, agricultura, florestamento e etc. Por fim, os "robôs humanoides" são os que possuem formato humano e podem estar associado a realizar as funções de serviços também (CORKE, 2017).

Outra forma interessante e bastante utilizada de classificar os manipuladores robóticos é de acordo com a sua geometria. Podem ser do tipo: Articulado (RRR), Esférico (RRP), SCARA (RRP), Cilíndrico (RPP) ou Cartesiano (PPP), em que R e P são os tipos de junta, sendo do tipo Revolução (R) ou Prismática (P) (SPONG *et al.*, 2006).

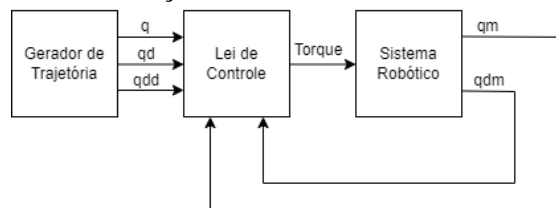
Em todos esses casos mencionados anteriormente, é necessário inserir um controlador ao sistema robótico, de forma que o sistema possa seguir os requisitos de desempenho desejado e rejeitar perturbações externas, visto que nem sempre é uma realidade alcançável sem o uso de uma lei de controle. Na prática, é necessário o uso de sensores para medir a posição e velocidade

de cada junta, para calcular os erros de posição e velocidade do sistema e gerar um torque de controle aplicado ao manipulador de forma a corrigir as incertezas e perturbações do sistema.

Existem diversas formas de implementar uma lei controle em um sistema robótico. Um deles, e que será abordado nesse trabalho, é o controle baseado na equação recursiva de Newton-Euler (RNE, do inglês *Recursive Newton-Euler*). Nesse caso, o controle é construído por duas parcelas: *feedforward* e *feedback*. A parcela de feedforward é responsável pelo cálculo da Equação Recursiva de Newton-Euler (RNE) propriamente dita, no que resulta no torque para cada junta do sistema para a posição desejada. A parcela de feedback é responsável por compensar os erros devidos às incertezas paramétricas do sistema, dinâmicas não-modeladas e perturbações externas (CORKE, 2017).

A lei de controle em um manipulador robótico pode ser aplicado da seguinte maneira:

Figura 1 – Representação de um braço de 2 Links



Fonte: Adaptado de (CRAIG, 2013)

Em que  $q$ ,  $q_d$  e  $q_{dd}$  são, respectivamente, posição, velocidade e aceleração de referência em uma determinada trajetória que o sistema irá percorrer; O torque é a variável calculada a partir da malha de controle;  $q_m$  e  $q_{dm}$  são, nessa ordem, posição e velocidade medidas a partir de sensores no sistema.

De maneira inicial, o uso de controladores em processos industriais se resumiam a aplicação de controladores Proporcional Derivativo (PD) e Proporcional Integral Derivativo (PID) (GAIDHANE *et al.*, 2019). O controle PD é bastante utilizado na robótica na parcela de feedback, mas nesse caso é geralmente necessário a linearização do sistema em um ponto de operação (CORKE, 2017) (SICILIANO *et al.*, 2008).

## 1.1 Motivação do Trabalho

Existem diversos tipos de incertezas paramétricas que podem ser aplicadas. Entre elas, existe a incerteza da massa na carga, que será abordada nesse trabalho. A mesma é destacada, pois há vários tipos de possibilidades em que um sistema pode estar sujeito a uma carga diferente

do que um controlador que tem um determinado ponto de operação para o qual foi projetado. Então, como exemplo, tem-se um sistema que opera em um determinado processo industrial e deve carregar diversos blocos, com massas diferentes, um por vez. Nesse caso, definir apenas um ponto de operação e projetar um controlador não é a melhor opção, pois o sistema é submetido a grandes variações de massa e é necessário garantir que o controlador seja robusto para manter o desempenho adequado para todas essas variações.

## 1.2 Justificativa do Trabalho

A partir da problematização apresentada na Seção 1.1 e sabendo do aumento de acessibilidade de manipuladores robóticos, surge-se a necessidade de aprofundar os estudos em controladores robustos, no caso desse trabalho, o controlador PD, de forma a garantir que um sistema robótico de 3-DOF esteja apto a seguir a referência para uma carga com massa variável.

## 1.3 Objetivos

A partir das problematizações apresentadas, este trabalho tem como principais objetivos:

- Modelar um sistema robótico de 3-DOF em uma plataforma de simulação, nesse caso a *Robotics Toolbox* em Python (CORKE, 2017);
- Validar o modelo simulado através das equações matemáticas teóricas apresentadas em (URREA *et al.*, 2020);
- Projetar um controlador PD que seja robusto e permita o sistema operar, de forma estável sem carga, ou com uma carga de massa igual ao valor da massa total do manipulador robótico.

## 1.4 Organização do Trabalho

Este trabalho é organizado nos seguintes capítulos:

1. **Introdução:** Neste capítulo é contido a contextualização histórica relacionado aos manipuladores robóticos, indicando a justificativa de utilizar uma lei de controle no mesmo e contendo os objetivos gerais desse trabalho.
2. **Modelagem Matemática do Braço Robótico:** Este capítulo introduz a modelagem matemática de um manipulador robótico a partir da Equação de Corpo Rígido, além de

mostrar uma abordagem de simplificação da mesma, utilizando um sistema massa-mola equivalente.

3. **Controle em Sistemas Robóticos:** Após definir a matemática relacionada à modelagem de sistemas robóticos, é necessário introduzir estratégias de controle que podem ser implementadas nesses sistemas para garantir o comportamento desejado em pontos de operações.
4. **Descrição do braço de bancada:** Aqui é introduzido o processo e os primeiros resultados experimentais da modelagem do braço de bancada na *Robotics Toolbox* em Python, comparando com os resultados teóricos. É mostrado também o projeto de controle PD realizado para o braço.
5. **Resultados:** Este capítulo retrata os resultados do braço operando em malha aberta, para diversos valores de massa na carga, assim como os resultados de malha fechada, com a lei de controle implementada, os gráficos de sinal de controle, de feedback e a comparação dos resultados.
6. **Conclusões e Trabalhos Futuros:** Por fim, este capítulo relaciona os resultados obtidos com os objetivos gerais desse trabalho, além de indicar possíveis trabalhos futuros que podem ser abordados a partir desses resultados obtidos.

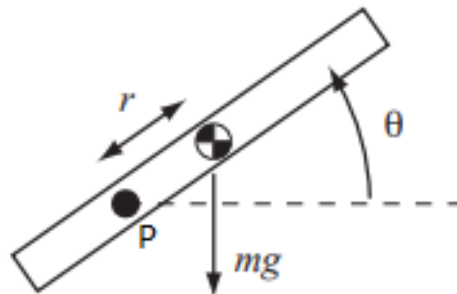
## 2 MODELAGEM MATEMÁTICA DO BRAÇO ROBÓTICO

A importância da modelagem de sistemas robóticos deve-se à necessidade do conhecimento da dinâmica do sistema. Esse sistema é do tipo não-linear e o conhecimento da sua dinâmica permite saber as suas respectivas faixas de operações e a aplicação de controladores que melhorem a sua performance. Com essa ideia, será abordado nesse capítulo a modelagem matemática de sistemas robóticos a partir da Equação de Corpo Rígido, e mais a frente, será abordado ferramentas que simplificam esse tipo de sistema, assim como realizar o controle do mesmo.

### 2.1 Modelagem de um sistema robótico

Para realizar a modelagem matemática de um braço robótico de  $n$  links, iniciamos a análise cinemática com o caso inicial em que o número de links é unitário (CHOI K.; KIM, 2014). Para representar esse caso, utiliza-se um sistema como o da Figura 2.

Figura 2 – Representação de um braço de 1 Link



Fonte: Adaptado de (CHOI K.; KIM, 2014)

Em que nesse caso, o ponto P é o ponto de rotação, CM é o centro de massa do sistema,  $\theta$  é o ângulo de rotação e  $r$  é a distância do ponto de rotação até o centro de massa. Com isso, é possível estimar o torque produzido em torno do ponto P, como é descrito na Equação 2.1:

$$T = M \cdot \ddot{\theta} + m \cdot g \cdot r \cdot \cos\theta \quad (2.1)$$

Entretanto, a equação acima não possui uma parcela que representa a fricção entre a barra e o ponto de rotação P. Para resolver isso, é necessário incluir um termo de atrito viscoso

"b", que seja proporcional a velocidade de rotação, apresentado na Equação 2.2:

$$T = M \cdot \ddot{\theta} + m \cdot g \cdot r \cdot \cos\theta + b \cdot \dot{\theta} \quad (2.2)$$

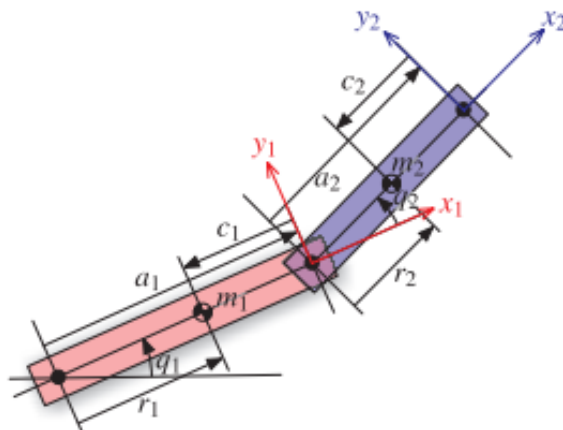
Percebe-se que a Equação 2.2 descreve um sistema rotacional em que  $M$  é uma matriz que contém os termos inerciais do sistema. Além disso, a posição angular descrita por  $\theta$  é comumente escrita na forma generalizada em que se utiliza a variável  $q$ .

Nesse contexto, convém substituir  $\theta$ ,  $\dot{\theta}$  e  $\ddot{\theta}$ , por  $q$ ,  $\dot{q}$  e  $\ddot{q}$ . A matriz  $M$ , que depende dos momentos de inércia dos links, deve ser escrita como  $M(q)$ , porque tem seu valor alterado em função de cada posição dos links, sendo chamada de matriz de inércia. Por fim, considera-se o termo  $G(q)$  que refere-se à força gravitacional do braço e o termo  $F(\dot{q})$ , que representa as forças de fricção, resultado no modelo generalizado:

$$T = M(q) \cdot \ddot{q} + G(q) + F(\dot{q}) \quad (2.3)$$

Na sequência, a análise é expandida para o caso em que o braço possui 2 links, como representado na Figura 3:

Figura 3 – Representação de um braço de 2 Links



Fonte: Adaptado de (CORKE, 2017)

Com a introdução de mais um link no sistema, aparece um novo termo que será acrescentado na Equação 2.3, que é a força de Coriolis, em que essa indica a força causada de um link no outro. Com isso, é possível definir o torque do Link 1 e as seguintes Equações abaixo de forma intuitiva (CORKE, 2017).

$$T_1 = M_{11}(q_2) \cdot \ddot{q}_1 + M_{12}(q_2) \cdot \ddot{q}_2 + C_1(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 + C_2(q_2) \cdot \dot{q}_2^2 + F(\dot{q}) + G(q_1, q_2) \quad (2.4)$$

Em que:

$$M_{11} = m_1 \cdot (a_1^2 + 2 \cdot a_1 \cdot c_1 + c_1^2) + m_2 \cdot (a_1^2 + (a_2 + c_2)^2) + (2 \cdot a_1 \cdot a_2 + 2 \cdot a_1 \cdot c_2) \cdot \cos(q_2) \quad (2.5)$$

$$M_{12} = m_2 \cdot (a_2 + c_2) \cdot (a_2 + c_2 + a_1 \cdot \cos(q_2)) \quad (2.6)$$

$$C_1 = -2 \cdot a_1 \cdot m_2 \cdot (a_2 + c_2) \cdot \sin(q_2) \quad (2.7)$$

$$C_2 = -a_1 \cdot m_2 \cdot (a_2 + c_2) \cdot \sin(q_2) \quad (2.8)$$

$$G = (a_1 \cdot m_1 + a_1 \cdot m_2 + c_1 \cdot m_1) \cdot \cos(q_1) + (a_2 \cdot m_2 + c_2 \cdot m_2) \cdot \cos(q_1 + q_2) \quad (2.9)$$

A partir da Equação 2.4, percebe-se que apenas o termo  $M_{11}(q_2) \cdot \ddot{q}_1$  depende somente de elementos que são próprios do link 1, podendo o termo  $M_{12}(q_2) \cdot \ddot{q}_2 + C_1(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 + C_2(q_2) \cdot \dot{q}_2^2 + F(\dot{q}) + G(q_1, q_2)$  ser visto como uma perturbação para  $T_1$  (CORKE, 2017). Dessa forma, percebe-se que a modelagem de um manipulador não pode ser dissociada em links individuais já que o torque  $T_1$  depende de  $q_2$ , o termo gravitacional depende de  $q_1$  e  $q_2$ , além dos termos de Coriolis. Enfim, o movimento rotacional de um sistema de juntas e links mecânicos é não trivial e o movimento de uma junta é afetado pelo movimento de todas as juntas (CORKE, 2017).

Ainda analisando a Equação 2.4, temos que  $M_{11}$  e  $M_{12}$  são as matrizes de inércia próprias dos links 1 e do link 2 causado no link 1, respectivamente. Pode-se inserir os termos  $C_1$  e  $C_2$ , na Equação 2.3 e generalizar a mesma para um caso de um braço de n-links. Então, é possível obter a Equação de Corpo Rígido para um sistema robótico:

$$\tau = M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F(\dot{q}) + G(q) \quad (2.10)$$



## 2.2 Elementos da Equação de Corpo Rígido

A partir da Equação 2.10, pode ser analisado cada termo:

$G(q)$ : É o termo relacionado a força gravitacional do braço. É geralmente o termo dominante da Equação 2.10. A melhor maneira de controlar esse termo é utilizando molas e amortecedores de forma a reduzir o torque gravitacional (CORKE, 2017). É uma matriz do formato  $N \times 1$ , em que  $N$  é o número de links do sistema robótico.

$$G(q) = \begin{bmatrix} G_{11} \\ \vdots \\ G_{N1} \end{bmatrix} \quad (2.11)$$

$M(q)\ddot{q}$ : É o termo que representa os momentos de inércia próprios e cruzados do braço robótico. É uma matriz simétrica e quadrada,  $N \times N$ , em que  $N$  é o número de links do sistema robótico. Os termos  $M_{ij}$ , em que  $i = j$ , representam os termos dos momentos de inércias próprios, em que  $i$  é respectivo número do link do sistema robótico. Para  $i \neq j$ , nesse caso, são os momentos de inércias cruzados entre os links  $i$  e  $j$ . Como a matriz é simétrica, então  $M_{ij} = M_{ji}$  (CORKE, 2017).

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1N} \\ M_{21} & M_{22} & \dots & M_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{2N} & \dots & M_{NN} \end{bmatrix} \quad (2.12)$$

$C(q, \dot{q})$ : É o termo matricial  $N \times N$ , sendo  $N$  o número de links do sistema robótico em que depende da posição e velocidade angular de cada junta do sistema. A força de Coriolis é proporcional a  $q_i \cdot q_j$  (CORKE, 2017). Pode ser representada por:

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1N} \\ C_{21} & C_{22} & \dots & C_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N1} & C_{2N} & \dots & C_{NN} \end{bmatrix} \quad (2.13)$$

$F(\dot{q})$ : É o segundo termo dominante da Equação 2.10, sendo uma matriz do formato  $N \times 1$ , em que  $N$  é o número de links do sistema robótico. Geralmente utiliza-se constantes

relacionadas à caixa de redução dos motores e a fricção de Coulomb para dimensionar esse termo (CORKE, 2017). Em alguns casos em que os motores ou servomotores são muito pequenos, é possível desprezar esse termo da equação.

$$F(\dot{q}) = \begin{bmatrix} F_{11} \\ \vdots \\ F_{N1} \end{bmatrix} \quad (2.14)$$

Em outras literaturas, é retratado sobre a possibilidade de estimar o torque mecânico de um sistema robótico através do método Recursivo de Newton-Euler (SPONG *et al.*, 2006), em que este é mais eficiente em aplicações utilizando meios computacionais (CORKE, 2017). Apesar dessa vantagem, esse método não calcula de forma explícita os termos M, C, G e F, e portanto, ele não será útil para fins de modelagem matemática, mas será utilizado nos Capítulos 4 e 5 para estimar o torque de cada junta a partir de uma trajetória desejada calculada previamente.

### 2.3 Simplificação do Modelo Matemático para um Braço de 2 Links

No caso de uma simplificação de um modelo de sistema robótico de 2 Links, é possível estimar as matrizes M, C e G pelas seguintes equações (URREA *et al.*, 2020):

$$\tau = M(q) \cdot \ddot{q} + C(q, \dot{q}) + F(\dot{q}) + G(q) \quad (2.15)$$

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (2.16)$$

$$M_{11} = m_1 \cdot l_{c1}^2 + m_2 \cdot [l_1^2 + l_{c2}^2 + 2 \cdot l_1 \cdot l_{c2} \cdot \cos(\theta_2)] + I_1 + I_2 \quad (2.17)$$

$$M_{12} = m_2 \cdot l_1 \cdot l_{c2} \cdot \cos(\theta_2) + m_2 \cdot l_{c2}^2 + I_2 \quad (2.18)$$

$$M_{21} = m_2 \cdot l_1 \cdot l_{c2} \cdot \cos(\theta_2) + m_2 \cdot l_{c2}^2 + I_2 \quad (2.19)$$

$$M_{22} = m_2 \cdot l_{c2}^2 + I_2 \quad (2.20)$$

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} \\ C_{21} \end{bmatrix} \quad (2.21)$$

$$C_{11} = -m_2 \cdot l_1 \cdot l_{c2} \cdot \sin(\theta_2) \cdot \dot{\theta}_2^2 - 2 \cdot m_2 \cdot l_1 \cdot l_{c2} \cdot \sin(\theta_2) \cdot \dot{\theta}_1 \cdot \dot{\theta}_2 \quad (2.22)$$

$$C_{21} = m_2 \cdot l_1 \cdot l_{c2} \cdot \sin(\theta_2) \cdot \dot{\theta}_1^2 \quad (2.23)$$

$$G(q) = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} \quad (2.24)$$

$$G_{11} = m_1 \cdot l_{c1} \cdot \cos(\theta_1) + m_2 \cdot [l_1 \cdot \cos(\theta_1) + l_{c2} \cdot \cos(\theta_1 + \theta_2)] \cdot g \quad (2.25)$$

$$G_{21} = [m_2 \cdot l_{c2} \cdot \cos(\theta_1 + \theta_2)] \cdot g \quad (2.26)$$

Em que:

$m_1$  : Massa do primeiro link, em kg

$m_2$  : Massa do segundo link, em kg

$l_1$  : Comprimento do primeiro link, em metros

$l_2$  : Comprimento do segundo link, em metros

$l_{c1}$  : Comprimento do primeiro link até seu centróide, em metros

$l_{c2}$  : Comprimento do segundo link até seu centróide, em metros

$I_1$  : Momento de inércia do primeiro link com respeito ao eixo Z, em  $kg \cdot m^2$

$I_2$  : Momento de inércia do segundo link com respeito ao eixo Z, em  $kg \cdot m^2$

$g$  : Constante gravitacional, que será utilizada a aproximação de  $9.81 \text{ m/s}^2$

A parcela referente a fricção, a Equação 2.14, será mantida, visto que essa depende de constantes específicas de cada servo ou motor utilizado no sistema.

## 2.4 Modelo Massa-Mola

A equação (2.10) depende de matrizes que variam de acordo com a posição. Entretanto, a equação de corpo rígido pode ser resolvida de forma recursiva e implementada em uma estrutura de controle feedforward. Porém, se faz necessário adicionar uma parcela de controle feedback para correção eventual de perturbações ou incertezas paramétricas de modelo. Nesse caso, convém obter um modelo linear equivalente baseado em um modelo massa-mola, chamado de modelo de impedâncias mecânicas, em que "m" é a massa de um bloco do sistema, "b" é a constante relacionada ao atrito viscoso, "k" é a constante elástica,  $x(t)$  é a posição do bloco em função do tempo e  $f(t)$  é uma força exercida ao sistema, como é representado na Equação 2.27 (CRAIG, 2013):

$$f(t) = m \cdot \ddot{x}(t) + b \cdot \dot{x}(t) + k \cdot x(t) \quad (2.27)$$

Esse modelo representado na Equação 2.27 pode ser associado a um modelo de sistema robótico, pois ao analisar-se cada junta de um manipulador robótico de forma individual, levando em consideração não apenas a Equação 2.10, mas adicionando termos relacionados as características flexíveis do braço, é possível associar cada junta a um sistema massa-mola equivalente (AL-SHUKA *et al.*, 2018) (CHENG *et al.*, 2017). Ao aplicar a transformada de Laplace para a Equação 2.27, com as condições iniciais nulas, temos a seguinte resposta:

$$F(s) = X(s) \cdot (m \cdot s^2 + b \cdot s + k) \quad (2.28)$$

O que resulta na seguinte função de transferência  $G(s)$ :

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{m \cdot s^2 + b \cdot s + k} \quad (2.29)$$

E por fim, ajustando os termos, pode-se obter o seguinte modelo abaixo:

$$G(s) = \frac{X(s)}{F(s)} = \frac{\frac{1}{m}}{s^2 + \frac{b}{m} \cdot s + \frac{k}{m}} \quad (2.30)$$

Vale ressaltar que o modelo obtido na Equação 2.30 é um modelo linear, enquanto o modelo obtido na Equação 2.10 é não-linear. Portanto, é necessário fazer a linearização em torno de um ponto de operação do sistema robótico para manter a equivalência entre os modelos associados.

### 3 CONTROLE EM SISTEMAS ROBÓTICOS

Nessa Seção será abordada estratégias de controle e formas de equivalências entre os modelos abordados na Seção anterior para utilizar em uma aplicação de sistemas robóticos.

#### 3.1 Controle de um Sistema Robótico

Seguindo a abordagem a partir da Equação 2.10, estimando-se  $\tau_a$ , que representa a força de um atuador externo que tem comportamento semelhante a um sistema massa-mola de constante elástica  $K_j$  e atrito viscoso  $B_j$ , ela pode ser representada por (CRAIG, 2013):

$$\tau_a(q, \dot{q}) = K_j \cdot (q_d - q) + B_j \cdot (\dot{q}_d - \dot{q}) \quad (3.1)$$

em que  $q_d$  é a posição desejada de trajetória angular e  $\dot{q}_d$  é sua primeira derivada, a velocidade angular desejada do sistema.

Substituindo em 2.10, resulta:

$$M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F(\dot{q}) + G(q) = K_j \cdot (q_d - q) + B_j \cdot (\dot{q}_d - \dot{q}) \quad (3.2)$$

No caso da implementação de uma lei de controle PD, é necessário existir duas constantes,  $K_p$  e  $K_v$ , que as mesmas sejam proporcionais aos erros de posição e velocidade, respectivamente. Os erros mencionados podem ser calculados da seguinte forma (CRAIG, 2013):

$$E = q_d - q \quad (3.3)$$

$$\dot{E} = \dot{q}_d - \dot{q} \quad (3.4)$$

Em que  $E$  e  $\dot{E}$  são os erros de posição e velocidade, respectivamente. Multiplicando os erros pelas respectivas constantes de proporcionalidades relacionadas ao controlador e somando esses termos, é possível obter o termo referente à parcela feedback do sistema (CORKE, 2017) (MOBERG, 2010), que será representado por  $u$ :

$$u = K_p \cdot (q_d - q) + K_v \cdot (\dot{q}_d - \dot{q}) \quad (3.5)$$

É possível agora comparar as Equações 3.5 e 3.1 e notar a semelhança entre as mesmas e os seus termos. Então, é possível substituir  $K_j$  e  $B_j$ , respectivamente, por  $K_p$  e  $K_v$ , resultando em um caso de controle do tipo PD:

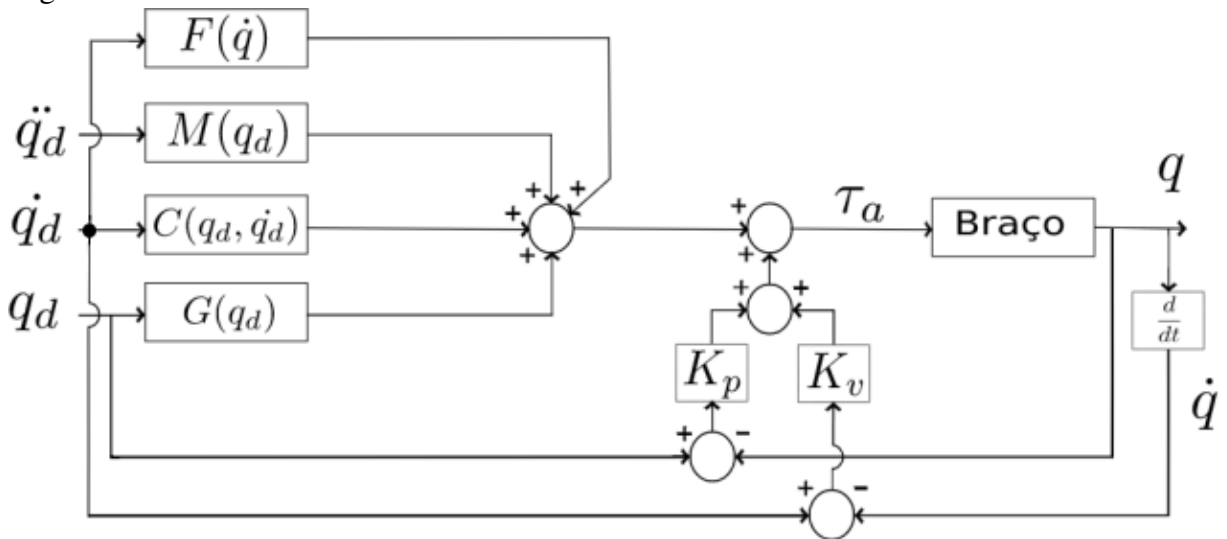
$$M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F(\dot{q}) + G(q) + K_p \cdot (q_d - q) + K_v \cdot (\dot{q}_d - \dot{q}) = 0 \quad (3.6)$$

Entretanto, nota-se que em 3.6, que o lado direito da Equação é nulo, isso significa que o somatório das forças no lado esquerdo também é nulo. Na prática, esse caso é quando o manipulador robótico está parado em uma determinada posição. Se desejarmos realizar uma trajetória, esse lado direito precisa ser uma força não-nula, que será representada por  $\tau$ , para garantir que o braço se movimentará. Assim, é possível obter o seguinte resultado (CORKE, 2017):

$$\tau = M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F(\dot{q}) + G(q) + K_p \cdot (q_d - q) + K_v \cdot (\dot{q}_d - \dot{q}) \quad (3.7)$$

Então, a partir da Equação 3.7, é possível obter o seguinte diagrama de blocos que representa o comportamento do sistema robótico:

Figura 4 – Diagrama de Blocos referente ao Sistema Robótico



Fonte: Autor

A partir da Figura 4, é possível notar duas parcelas. A primeira, é a que não depende dos termos de realimentação, que contém  $M(\ddot{q}_d)$ ,  $C(q_d, \dot{q}_d)$ ,  $F(q_d)$  e  $G(q_d)$ , e essa parcela é chamada de *feedforward*. A outra parcela, que consiste nos termos que dependem da realimentação da posição angular  $q_d$  e da velocidade angular  $\dot{q}_d$  é chamada de *feedback* (CORKE, 2017). A partir do diagrama de blocos mostrado, é possível projetar um controlador do tipo PD, a partir dos valores dimensionados de  $K_p$  e  $K_v$  de forma ao sistema seguir a trajetória de posições desejada de acordo com algum desempenho de projeto especificado.

### 3.2 Controle de um Sistema Massa-Mola

Para retratar o desenvolvimento e estudo de um controle PD em um sistema massa-mola, foi utilizado a abordagem de (CRAIG, 2013). A partir da Equação 2.27, realiza-se o estudo das raízes do denominador da Equação. Nesse caso, tem-se que:

$$m \cdot s^2 + b \cdot s + k = 0 \quad (3.8)$$

As raízes dessa Equação são:

$$s = \left\{ \begin{array}{l} \frac{-b}{2m} + \frac{\sqrt{b^2 - 4mk}}{2m} \\ \frac{-b}{2m} - \frac{\sqrt{b^2 - 4mk}}{2m} \end{array} \right. \quad (3.9)$$

As raízes obtidas acima são os polos do sistema, localizados em um plano real-imaginário em que sua posição dita o comportamento do sistema. Para a Equação 3.8, existem 3 possíveis casos em que é obtido os pares de polos  $s_1$  e  $s_2$ :

1. Raízes reais e diferentes: Esse caso é o sobreamortecido, pois  $b^2 > 4mk$ , então o atrito  $b$  é predominante no sistema. Utilizando a transformada inversa de Laplace, pode-se obter posição  $x$ , apresentada na Equação 2.27, em função do tempo  $t$ , em que:

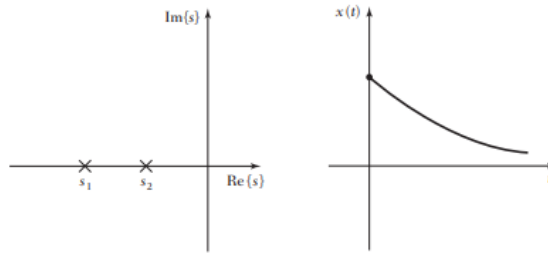
$$x(t) = c_1 \cdot e^{s_1 t} + c_2 \cdot e^{s_2 t} \quad (3.10)$$

Em que  $c_1$  e  $c_2$  são constantes que podem ser determinadas a partir das condições iniciais do sistema. Nesse caso, é normal existir um polo que possui uma magnitude maior do que a do outro. Verificando na Equação 3.10, percebe-se que o polo de maior magnitude terá seu respectivo termo tendendo a zero de forma mais rápida que o polo de menor magnitude. Assim, chamamos o polo de menor magnitude de "polo dominante", pois o comportamento do sistema dependerá diretamente desse polo. Com isso, a localização dos polos  $s_1$  e  $s_2$  e a resposta do sistema é do tipo:

2. Raízes complexas e conjugadas: Esse é o caso subamortecido, em que  $b^2 < 4mk$ , e nesse caso, a resposta do sistema é oscilatória. Nesse caso, haverá dois polos complexos e conjugados, tais quais:

$$s_1 = a + jb \quad (3.11)$$

Figura 5 – Posição dos polos e resposta do sistema em relação ao tempo



Fonte: (CRAIG, 2013)

$$s_2 = a - jb \quad (3.12)$$

E sabendo que a Identidade de Euler é:

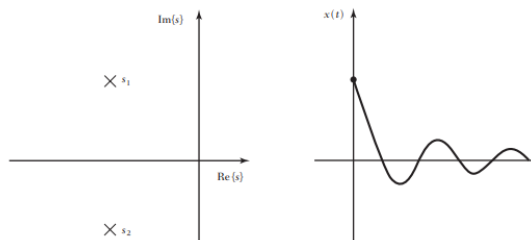
$$e^{jx} = \cos(x) + j \cdot \sin(x) \quad (3.13)$$

Aplicando a Identidade de Euler na Equação 3.10, tem-se que:

$$x(t) = c_1 \cdot e^{at} \cdot \cos(bt) + c_2 \cdot e^{at} \cdot \sin(bt) \quad (3.14)$$

De forma similar ao caso anterior,  $c_1$  e  $c_2$  são constantes determinadas a partir dos valores de condições iniciais de posição e velocidade do sistema. A localização dos polos e a resposta do sistema pra esse caso é do seguinte tipo:

Figura 6 – Posição dos polos e resposta do sistema em relação ao tempo



Fonte: (CRAIG, 2013)

3. Raízes reais e iguais: Nesse caso, tem-se que  $b^2 = 4mk$ , e com isso, é o caso de uma resposta rápida e não-oscilatória. O nome desse tipo de resposta é criticamente amortecida. Para esse caso, nota-se que o segundo termo de 3.9 é zerado, então os polos  $s_1$  e  $s_2$  são dados por:



$$s_1 = s_2 = -\frac{b}{2m} \quad (3.15)$$

E temos que a resposta  $x(t)$  é dada por:

$$x(t) = c_1 \cdot e^{s_1 t} + c_2 \cdot t \cdot e^{s_2 t} \quad (3.16)$$

Em que da mesma forma aos casos anteriores,  $c_1$  e  $c_2$  são constantes determinadas a partir das condições iniciais do sistema. Com isso, substituindo 3.15 em 3.16, obtém-se:

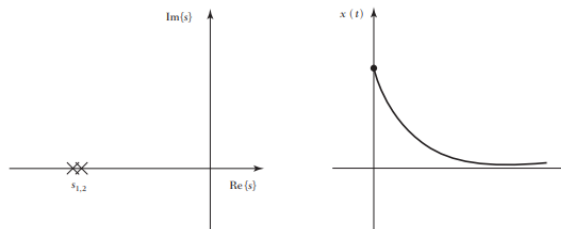
$$x(t) = c_1 \cdot e^{-\frac{b}{2m}t} + c_2 \cdot t \cdot e^{-\frac{b}{2m}t} \quad (3.17)$$

Reajustando 3.17, obtém-se que:

$$x(t) = (c_1 + t \cdot c_2) e^{-\frac{b}{2m}t} \quad (3.18)$$

Sua respectiva localização de polos e sua resposta em função do tempo é do tipo:

Figura 7 – Posição dos polos e resposta do sistema em relação ao tempo



Fonte: (CRAIG, 2013)

Seguindo o raciocínio, para projetar um controlador do tipo PD, de forma a controlar o sistema massa-mola que possui as possíveis respostas mostradas anteriormente, basta projetar um  $K_p$  e um  $K_v$  tal que a resposta do sistema alcance o desempenho desejado. A partir da Equação 2.27, é possível reescrever da seguinte forma (CRAIG, 2013):

$$m \cdot \ddot{x} + b \cdot \dot{x} + k \cdot x = \alpha \cdot f' + \beta \quad (3.19)$$

Sendo:

$$\alpha = m \quad (3.20)$$

$$\beta = b \cdot \dot{x} + k \cdot x \quad (3.21)$$

$$f' = \ddot{x} \quad (3.22)$$

Em que nesse caso  $\alpha$  e  $\beta$  são constantes que são escolhidas que se  $f'$  é uma nova força de entrada do sistema que permite normalizar o sistema, permitindo que a massa fique unitária para o projeto de controle. Entretanto, após a soma do sinal da parcela de feedback, é necessário multiplicar pela massa novamente, como mostrado na Figura 9 (CRAIG, 2013), representada pela seguinte Equação:

$$f' = -K_v \cdot \dot{x} - K_p \cdot x \quad (3.23)$$

A Equação 2.27 agora pode ser reescrita da seguinte forma:

$$\ddot{x} + K_v \cdot \dot{x} + K_p \cdot x = 0 \quad (3.24)$$

Com isso, a nova função de transferência do sistema pode ser representada por:

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + K_v \cdot s + K_p} \quad (3.25)$$

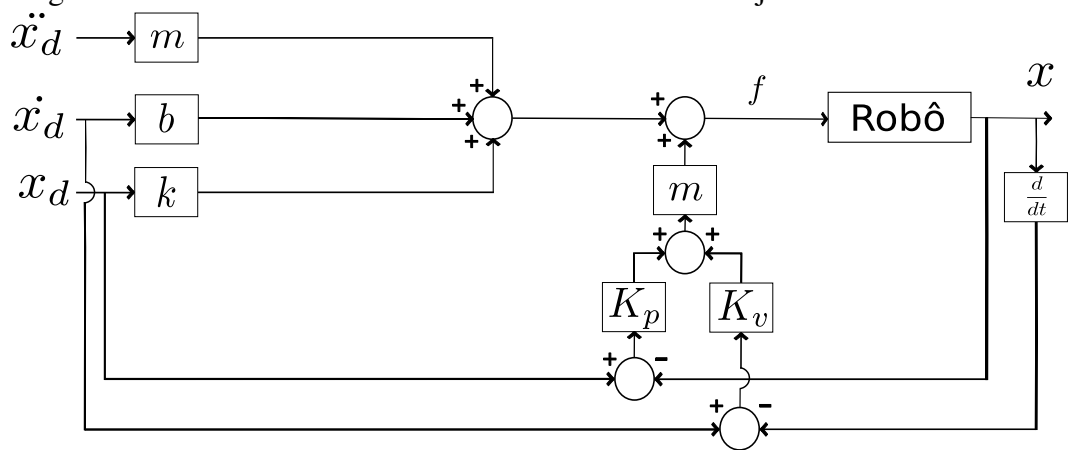
Agora para dimensionar o controlador nesse caso, basta escolher  $K_p$  e  $K_v$  de forma que eles sejam números reais, de forma a não obter uma resposta oscilatória e com o polo dominante localizado de acordo com o critério de desempenho desejado para o sistema. No caso da Equação 3.25,  $X(s)$  é o valor desejado do sistema, que pode ser representado por uma variável  $X_d(s)$  para uma determinada entrada  $F(s)$ . Em caso de aplicações de sistemas robóticos, o ideal é que o braço siga uma trajetória de posições, para cada respectivo link. Então com isso, é necessário modificar  $f'$  de tal forma que o erro de posição e velocidade sejam calculados:

$$f' = \ddot{x}_d - k_v \cdot (\dot{x}_d - \dot{x}) - k_p \cdot (x_d - x) \quad (3.26)$$

O que resulta em uma nova função de transferência do sistema para:



Figura 9 – Diagrama de Blocos do Modelo Massa Mola Controlado Ajustado

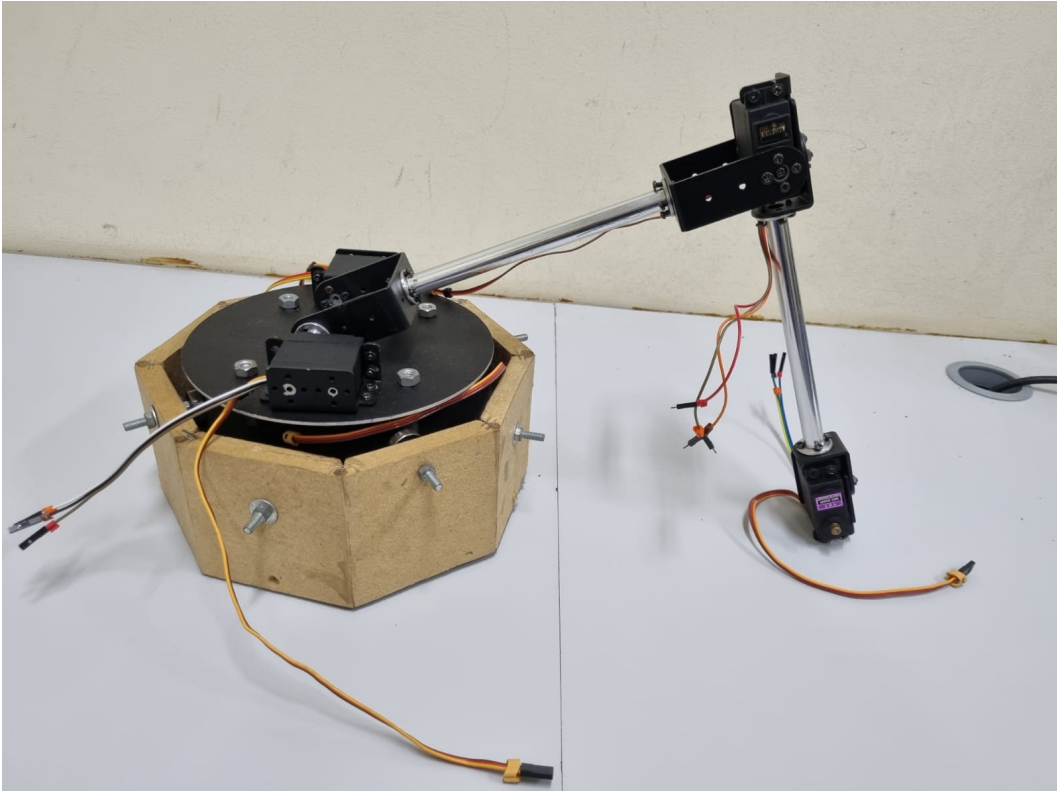


Fonte: Autor

#### 4 DESCRIÇÃO DO BRAÇO DE BANCADA

A partir das Equações definidas no Capítulo 2 e do braço de bancada contido no Lamotriz UFC, foi possível continuar o estudo de forma para o modelo disponível. O modelo do braço robótico de 2-Links encontra-se na Figura 10.

Figura 10 – Braço de Bancada do Lamotriz UFC



Fonte: Autor

Pela Figura 10, é possível notar que o braço tem a configuração de 3-DOF. Entretanto, para essa aplicação, o primeiro grau de liberdade, o da base será considerado como um suporte físico e assim, não será considerado nenhum cálculo ou projeto para o mesmo. Dessa forma, é realizado uma simplificação de forma ao sistema possuir apenas 2-DOF. Partindo da informação que o braço possui apenas dois links, utilizaremos as Equações 2.15 a 2.26. Além disso, como trata-se de um braço leve, com servomotores de baixíssimos coeficientes de atrito por serem de pequeno porte, iniciaremos a análise no modelo desprezando o termo  $F(\dot{q})$  para fins de simplificação do modelo. Além disso, foram medidos todos os parâmetros do braço robótico, mencionados na Seção 2.3, em que estes encontram-se na Tabela 1

Todos os termos foram medidos através de paquímetros e balança de massa de precisão, com exceção dos termos  $I_1$  e  $I_2$ . Para esses termos, o procedimento para estimar será

Tabela 1 – Medições do Braço de Bancada

Link 1	Link 2	Unidade
$m_1 = 0.07576$	$m_2 = 0.1923$	[kg]
$l_1 = 0.255$	$l_2 = 0.199$	[m]
$l_{c1} = 0.122$	$l_{c2} = 0.099$	[m]
$I_1 = 0.00175608$	$I_2 = 0.0174636$	[kg · m <sup>2</sup> ]

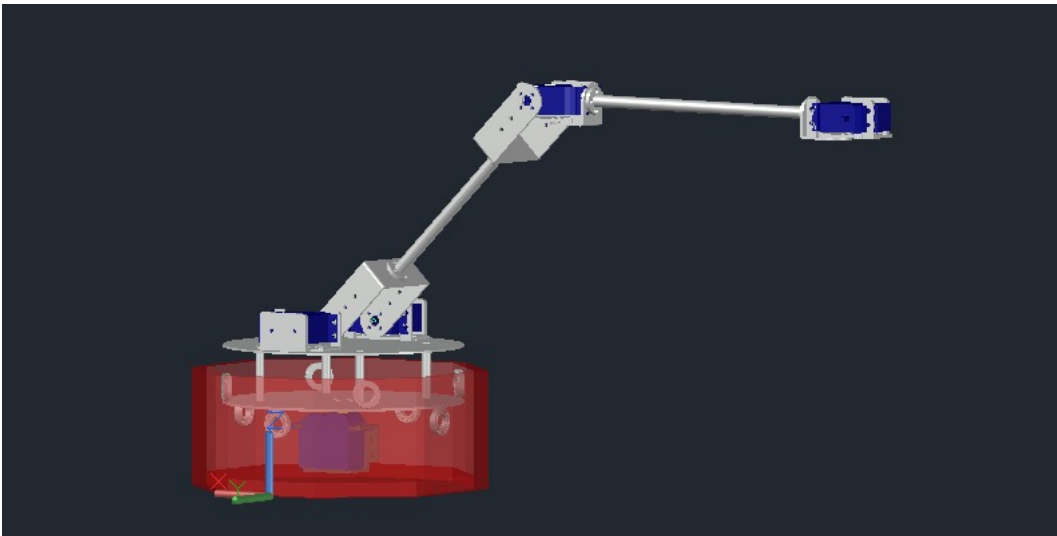
Fonte: Autor (2022).

apresentado na próxima seção.

#### 4.1 Estimativa dos Momentos de Inércia

Inicialmente foi realizado o desenho do modelo do braço de bancada no AutoCAD (AUTODESK, 2018), de forma a utilizar sua função "MASSPROP" para conseguir os dados dos momentos de inércia para uma determinada posição do braço. O braço possui a configuração inicial dos ângulos em 0°, em posição horizontal e o momento de inércia do braço foi calculado para essa posição. Na Figura 11 mostra o braço robótico desenhado no AutoCAD.

Figura 11 – Braço de Bancada desenhado no AutoCAD



Fonte: Autor

Com isso, foi possível utilizar o comando MASSPROP para calcular os momentos de inércia XX, YY, ZZ, XY, XZ e YZ. Apenas o momento de inércia do eixo Z em relação ao próprio eixo será útil para nossa aplicação. Os valores obtidos para o link 1 e o link 2, foram:

$$I'_{1ZZ} = 5.962 \cdot 10^8 \frac{\text{kg}}{\text{mm}^2} \quad (4.1)$$

$$I'_{2ZZ} = 5.967 \cdot 10^9 \frac{kg}{mm^2} \quad (4.2)$$

Como o comando MASSPROP utiliza como parâmetro de densidade um valor unitário (PROPMASS. . . , ) (quantidade de massa igual a quantidade de volume calculado pelo software), é necessário multiplicar o resultado obtido pela densidade do alumínio para ajustar a unidade para  $kg/mm^3$ , utilizando um  $d_1$ , em que este é a densidade do alumínio em  $kg/mm^3$ . Então:

$$d_1 = 2700 \cdot 10^{-9} \frac{kg}{mm^3} \quad (4.3)$$

Além disso, os calculos realizados estão na dimensões de metros, mas o desenho foi projetado para as dimensões de milímetros, e como a fórmula que o software utiliza para cálculo do momento de inércia é:

$$I = m \cdot r^2 \quad (4.4)$$

Em que I é o momento de inércia, m é a massa do objeto e r é o raio do eixo de rotação para a atuação da força. Visto que a unidade é obtida em  $kg \cdot m^2$ , no nosso caso, para ajustar de  $mm^2$  para  $m^2$ , é necessário adicionar mais outro fator multiplicativo,  $d_2$ , tal que:

$$d_2 = 10^{-6} \quad (4.5)$$

Com isso, o fator multiplicativo de ajuste total, d, que resultará em um momento de inércia calculado pelo AutoCAD na unidade de  $kg \cdot m^2$  é a multiplicação de  $d_1$  e  $d_2$ :

$$d = d_1 \cdot d_2 = 2700 \cdot 10^{-9} \cdot 10^{-6} = 2700 \cdot 10^{-15} \quad (4.6)$$

Então, os valores ajustados de momento de inércia, para o link 1 e link 2 são, respectivamente:

$$I_{1ZZ} = d \cdot I'_{1ZZ} = 1.60974 \cdot 10^{-3} \quad (4.7)$$

$$I_{2ZZ} = d \cdot I'_{2ZZ} = 1.61109 \cdot 10^{-2} \quad (4.8)$$

Com a estimativa dos momentos de inércia realizados, agora, é possível completar as Equações 2.17 a 2.20, completando assim, o termo  $M(q)$ . Com os valores mostrados na Tabela 1, também é possível completar os termos  $C(q, \dot{q})$  e  $G(q)$  a partir das Equações 2.22, 2.23, 2.25 e 2.26.

## 4.2 Definição do Modelo de Bancada em Código Python

Com os parâmetros do sistema definidos, será utilizado uma *Toolbox* de robótica desenvolvida em Python, chamada *Robotics Toolbox* (CORKE, 2017)). Existem diversas funções que realizam os cálculos dinâmicos e cinemáticos relacionados aos modelos já existente na mesma ou permitindo até mesmo criar novos modelos. Com isso, foi possível descrever o modelo do braço de bancada a partir de todos os valores medidos e exibidos na Tabela 1. O braço robótico com os parâmetros levantados é do tipo D-H (DENAVID; HARTENBERG, 1955), e por isso, foram utilizados os códigos referentes aos modelos D-H. A partir disso, é possível verificar as matrizes  $M(q)$ ,  $C(q, \dot{q})$  e  $G(q)$  para uma determinada posição do braço. Nesse caso, como mencionado no início deste capítulo,  $F(\dot{q})$  será simplificado a zero. Portanto, para realizar isso, é utilizado a função da *toolbox* "nofriction()". Será definida uma posição inicial em que os ângulos  $\theta_1$  e  $\theta_2$  sejam nulos. Com isso, supondo uma velocidade inicial nula para os dois ângulos, as matrizes  $M(q)$ ,  $C(q, \dot{q})$  e  $G(q)$  calculadas pela *toolbox* são:

$$M(q) = \begin{bmatrix} 0.04329537 & 0.02293755 \\ 0.02293755 & 0.0180339 \end{bmatrix} \quad (4.9)$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.10)$$

$$G(q) = \begin{bmatrix} 0.76854071 \\ 0.1886463 \end{bmatrix} \quad (4.11)$$

A partir desses resultados, podemos verificar os resultados obtidos para validar o modelo a partir das Equações 2.17 a 2.26. A partir do software MATLAB (MATLAB, 2021), foi possível calcular as matrizes  $M(q)$ ,  $C(q, \dot{q})$  e  $G(q)$  e foram obtidos os seguintes resultados:



$$M(q) = \begin{bmatrix} 0.0444 & 0.0242 \\ 0.0242 & 0.0193 \end{bmatrix} \quad (4.12)$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.13)$$

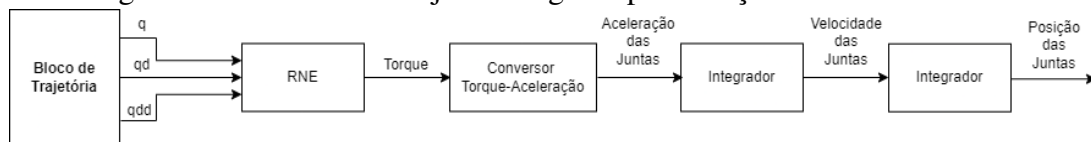
$$G(q) = \begin{bmatrix} 0.7577 \\ 0.1866 \end{bmatrix} \quad (4.14)$$

A partir dos valores obtidos, analisando 4.9 a 4.14, verifica-se que para todas as matrizes,  $M(q)$ ,  $C(q, \dot{q})$  e  $G(q)$ , é possível notar resultados bem próximos entre os que foram obtidos. Então, significa que foi possível modelar o braço de bancada com boa precisão via código em Python.

### 4.3 Cálculo da Trajetória realizada pelo Braço Robótico

Para realizar a simulação do braço robótico via *Robotics Toolbox* no Python, é necessário que seja elaborado um código que conste todas as etapas mostradas na Figura 12:

Figura 12 – Fluxograma de Cálculo da Trajetória seguida pelo Braço Robótico



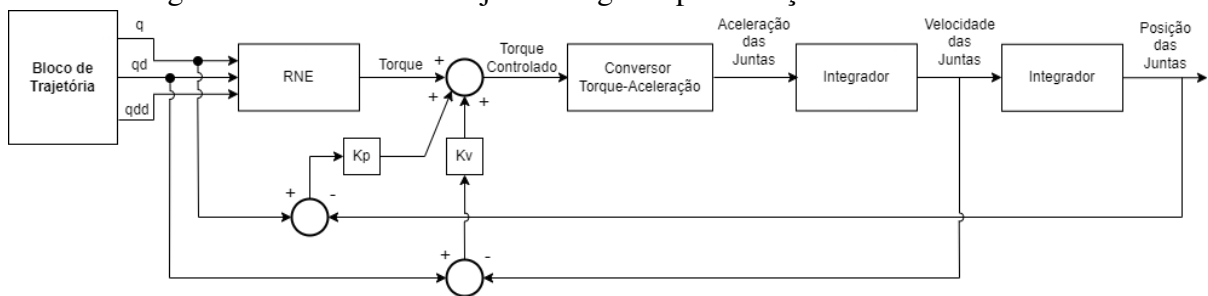
Fonte: Autor

A primeira etapa é calcular a trajetória, representada pelo "Bloco de Trajetória", em que nesse momento é escolhido os ângulos iniciais e finais para cada link do braço e a função respectiva da *toolbox* calcula a trajetória, velocidade e aceleração de referência a partir da dinâmica inversa do sistema robótico (FILGUEIRAS, 2019). Com a trajetória definida, a próxima etapa é o cálculo da RNE, representado pelo bloco "RNE". Nessa etapa, utiliza-se os valores de posição, velocidade e aceleração definidas na etapa anterior para estimar o torque para a respectiva trajetória. Vale ressaltar que é calculado o torque dinâmico, e não o de referência, pois ao inserir os valores de referências nesse bloco, é calculado o valor do torque através dos parâmetros dinâmicos do sistema. A próxima etapa é representada pelos blocos restantes, em

que nesta figura é de forma simplificada. O processo é de forma iterativa e será abordado com mais detalhe na Seção 4.6. Nessa etapa é calculado a aceleração a partir do torque, velocidade e posição anteriores e assim, a partir da integração dupla, é possível resultar o valor de posição para toda a trajetória dada como referência.

Para a Figura 12, foi mostrado o fluxograma da trajetória realizada pelo braço robótico através da *Robotics Toolbox*. Entretanto, essa Figura ilustra apenas a trajetória do sistema em malha aberta. Caso deseja-se adicionar uma lei de controle ao sistema, é necessário adaptar o fluxograma de acordo com a Figura 13:

Figura 13 – Fluxograma de Cálculo da Trajetória seguida pelo Braço Robótico



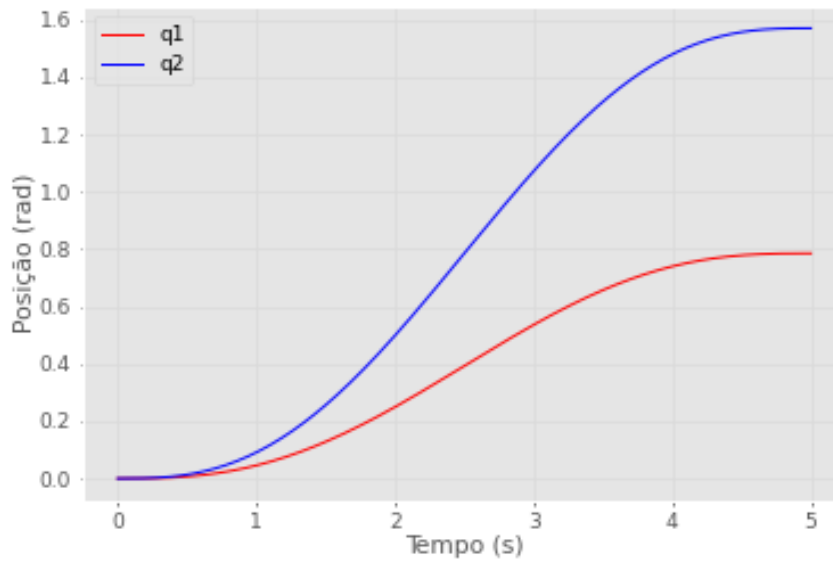
Fonte: Autor

#### 4.4 Definição de Trajetórias em Código Python

Com o modelo definido, é possível criar uma trajetória pela *Robotics Toolbox*, passando os parâmetros de tempo de simulação, tempo de amostragem  $T_s$ , posição inicial e final para cada junta do braço. Para esse trabalho, foi pensado em 2 trajetórias que o braço deveria seguir. A primeira trajetória,  $T_1$  é dada por  $T_1 = [0, 0] \rightarrow [\frac{\pi}{4}, \frac{\pi}{2}]$ , e foi escolhida para analisar o impacto da aplicação de uma carga durante a trajetória em ângulos grandes e pequenos. A segunda, será a  $T_2$ , dada por  $T_2 = [0, 0] \rightarrow [\frac{\pi}{3}, -\frac{\pi}{3}]$ , a escolha dessa trajetória dá-se à uma característica observada em ensaios com ângulos simétricos, em que a posição do *end-effector* permanece inalterada durante toda a trajetória, e poderá verificar o comportamento da RNE nessa aplicação. Em ambas trajetórias será considerado um tempo de simulação de 5 segundos e um tempo de amostra  $T_s = 0.02$  s. A função na *toolbox* para definir uma trajetória é `jtrat()`. A partir dessa função, foi possível gerar os seguintes gráficos de trajetória para o braço:

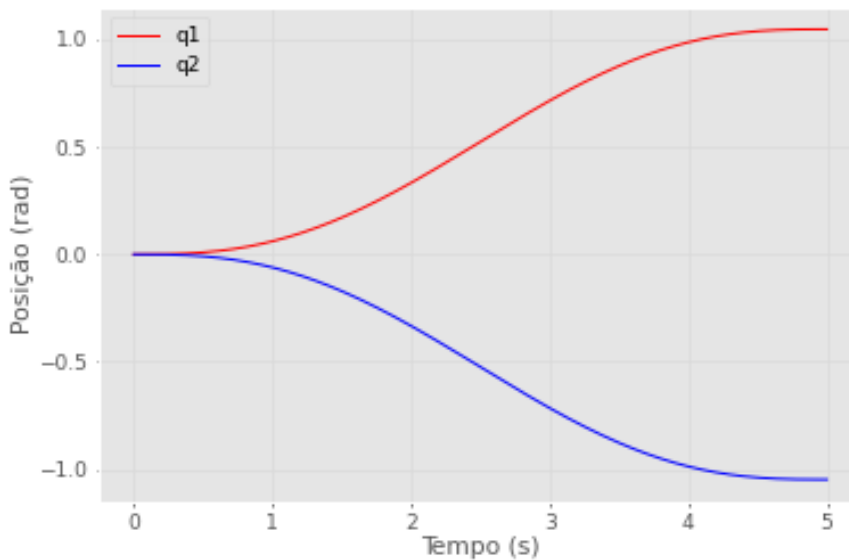
Em que  $q_1$  é a trajetória de posição referente ao link 1 do braço e  $q_2$  é a trajetória de posição referente ao link 2 do braço, sendo ambos valores de referência.

Figura 14 – Trajetória 1 para o braço robótico



Fonte: Autor

Figura 15 – Trajetória 2 para o braço robótico



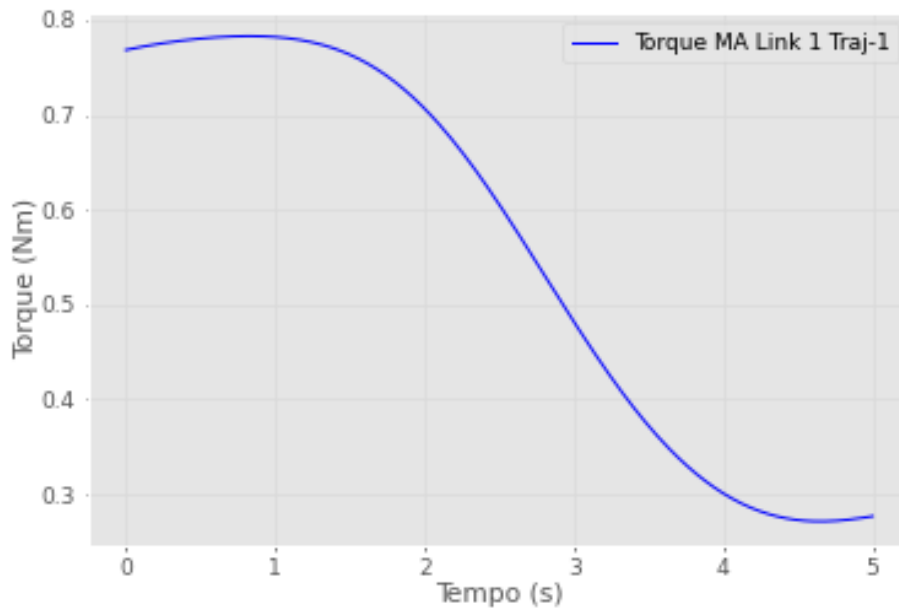
Fonte: Autor

#### 4.5 Cálculo da RNE em Código Python

Como mencionado no Capítulo 2, existe outra fórmula de calcular o valor do torque em uma trajetória do braço, que é usando a RNE, em que esse método apresenta vantagens computacionais comparado a Equação 2.10. A *Robotics Toolbox* utiliza a RNE para calcular o torque em uma determinada trajetória fornecida, através da função "rne()". As Figuras 16 e 17

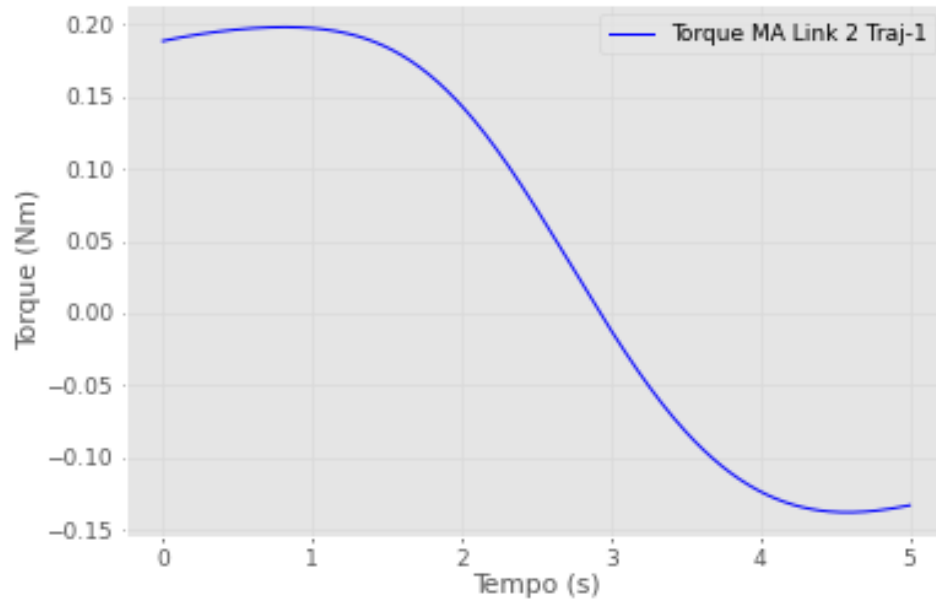
retratam o torque calculado através dessa função utilizando a Trajetória 1. Para validação desse resultado, será utilizada a Equação 2.10, com os resultados ilustrados nas Figuras 18 e 17. Por fim, é comparado os resultados obtidos, sendo estes ilustrados nas Figuras 20 e 21.

Figura 16 – Torque Calculado pela RNE para o Link 1



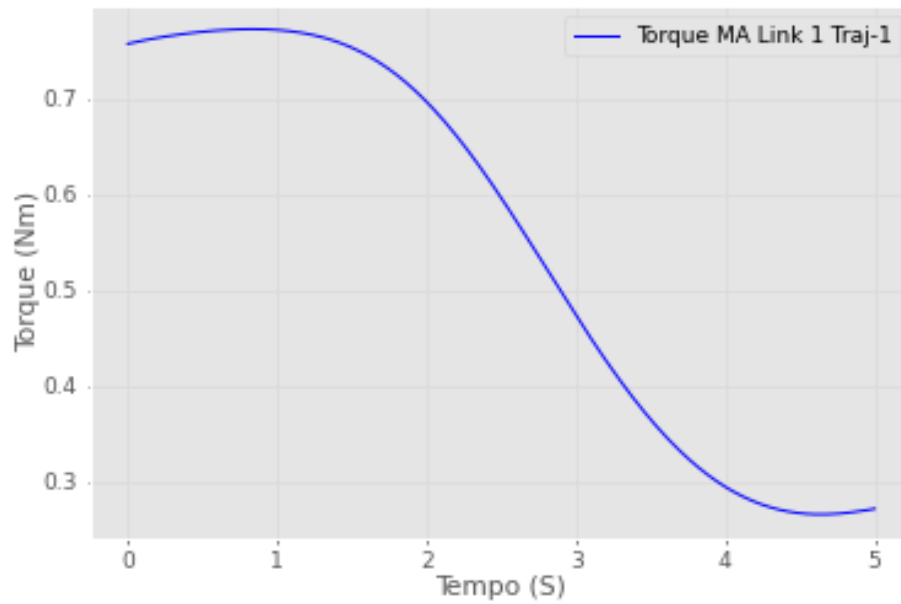
Fonte: Autor

Figura 17 – Torque Calculado pela RNE para o Link 2



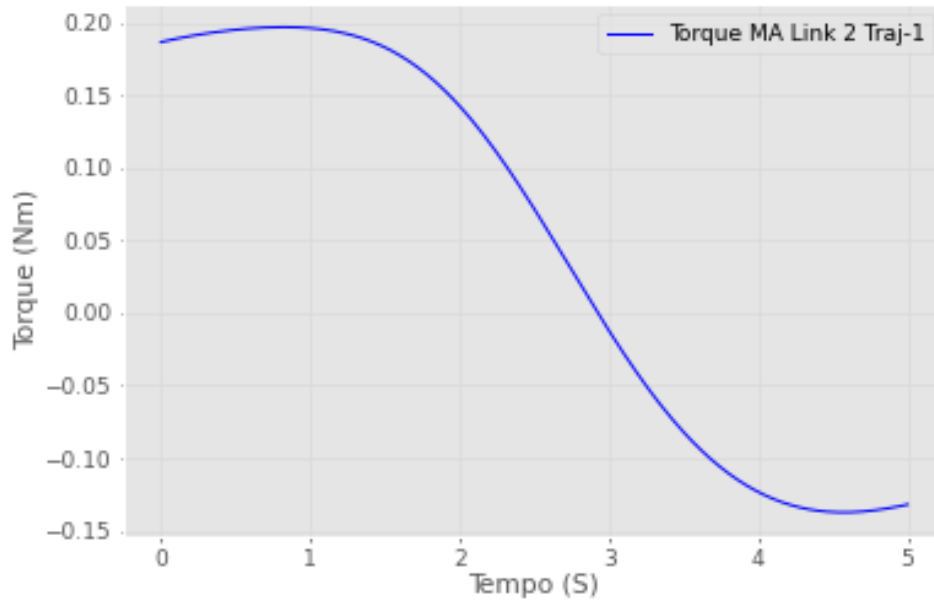
Fonte: Autor

Figura 18 – Torque Calculado pela Equação de Corpo Rígido para o Link 1



Fonte: Autor

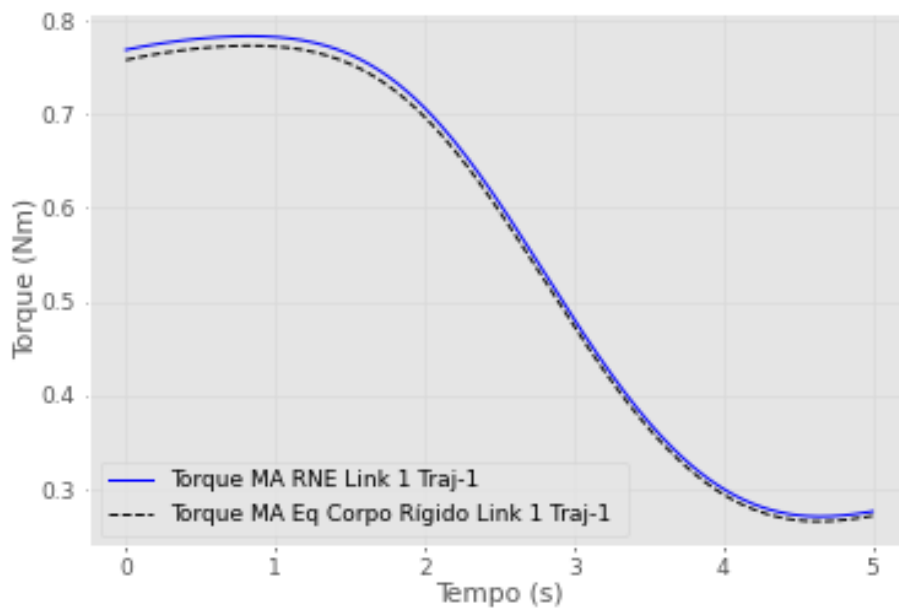
Figura 19 – Torque Calculado pela Equação de Corpo Rígido para o Link 2



Fonte: Autor

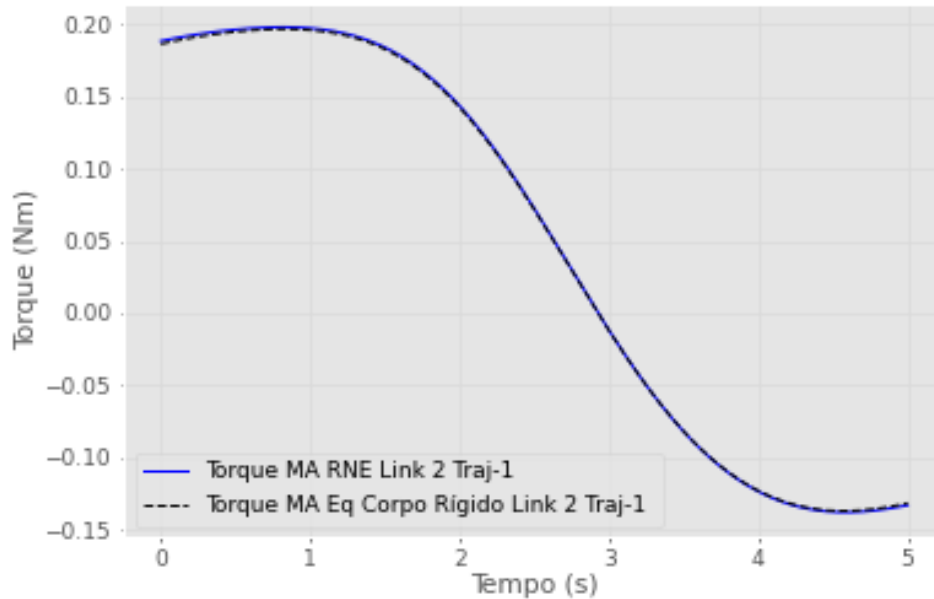
Agora, é possível fazer a comparação entre os resultados de torques estimados obtidos nas Figuras 16 a 19:

Figura 20 – Comparação do Torque Calculado pela RNE e Equação de Corpo Rígido no Link 1



Fonte: Autor

Figura 21 – Comparação do Torque Calculado pela RNE e Equação de Corpo Rígido no Link 2



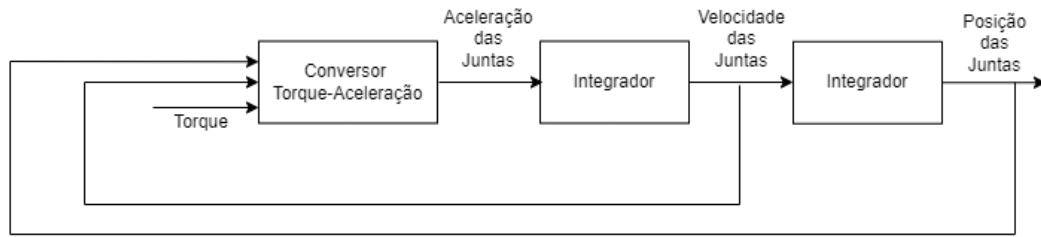
Fonte: Autor

Analisando as Figuras 20 e 21, conclui-se que os resultados de torques obtidos para cada link foram bem próximos e assim, os resultados obtidos pela função "rne()" puderam ser validados e serão utilizados para continuar os estudos do braço robótico via simulação.

#### 4.6 Cálculo dos ângulos a partir do Torque Calculado

Essa é a última etapa para o cálculo dos ângulos de trajetória do braço a partir de uma trajetória de referência dada. Essa etapa é feita de forma iterativa, utilizando a função "accel()" da *Robotics Toolbox* no Python. Ela é de forma iterativa, pois para calcular a aceleração em um determinado instante de tempo, é preciso utilizar como parâmetro da função o valor de torque no instante de tempo atual e a posição e velocidade anteriores. E para descobrir a velocidade e posição no mesmo instante de tempo da aceleração calculada, é necessário realizar uma integração (duas, no caso da posição) para obter os seguintes valores. O melhor fluxograma que descreve essa etapa é mostrado na Figura 22:

Figura 22 – Fluxograma Cálculo de Aceleração, Velocidade e Posição do Braço Robótico



Fonte: Autor

O torque é fornecido através da RNE e para começar o processo iterativo, foi definido a posição, velocidade e aceleração inicial como nula. Assim, é possível calcular a aceleração no seguinte instante de tempo, e em seguida, é possível aplicar uma integral para obter a velocidade e aplicando a integral mais uma vez, é possível obter a posição nesse mesmo instante de tempo. Para o cálculo da próxima iteração, utiliza-se os valores obtidos de velocidade e posição no instante anterior e o torque no novo instante de tempo desejado. Esse processo é repetido para todos os valores dentro do vetor de trajetória. O método de integração realizado para integrar a aceleração e obter a velocidade em um instante de tempo qualquer foi utilizar a seguinte Equação:

$$v = v_0 + at \quad (4.15)$$

Em que  $v$  é a velocidade no instante atual,  $v_0$  é a velocidade no instante anterior,  $a$  é a aceleração no instante atual e  $t$  é o tempo de intervalo, que nesse caso  $t = T_s = 0.02s$ .

De forma análoga, para calcular a posição angular em um instante de tempo qualquer, foi utilizado a seguinte Equação:

$$q = q_0 + vt \quad (4.16)$$

Em que  $q$  é a posição angular no instante de tempo atual,  $q_0$  é a posição angular no instante de tempo anterior,  $v$  é a velocidade no instante de tempo atual e  $t$  é o tempo de intervalo, que nesse caso  $t = T_s = 0.02s$ .

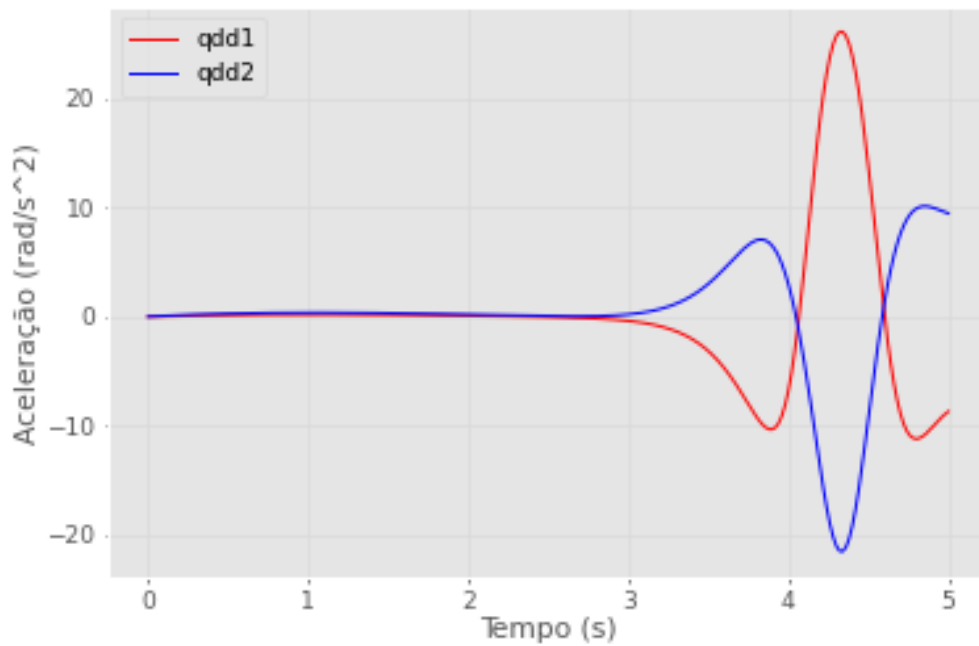
Vale ressaltar que esse método é referente ao sistema operando em malha aberta, e caso seja de preferência do usuário no simulador colocar uma lei de controle, de forma ao sistema operar em malha fechada, é durante o laço iterativo dessa etapa, que a cada atualização



dos valores de posição e velocidade, pode-se estimar um erro de posição e erro de velocidade e através desses valores obtidos, é possível implementar diversas técnicas e leis de controles.

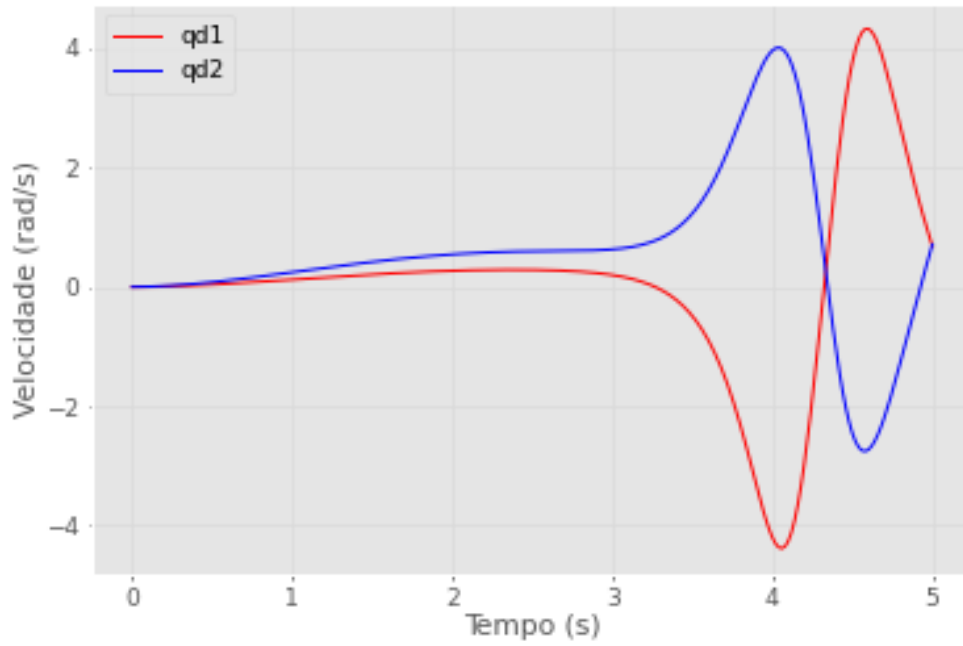
A partir do torque obtido pela RNE na Seção anterior e seguindo essa metodologia abordada nessa seção, é possível obter os seguintes gráfico de aceleração, velocidade e posição angular do braço robótico em malha aberta:

Figura 23 – Aceleração do Braço para Trajetória 1



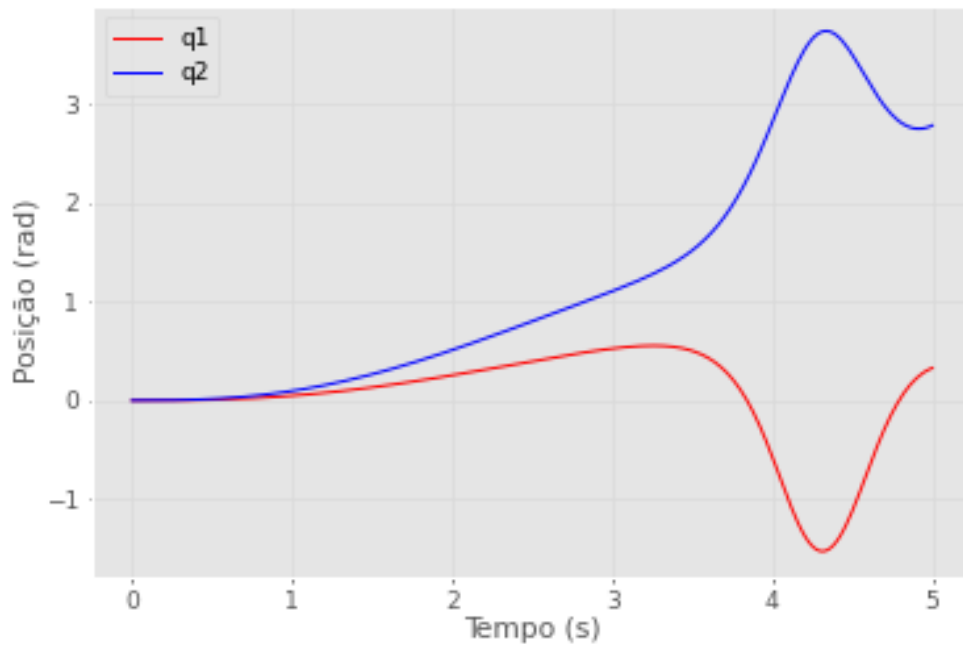
Fonte: Autor

Figura 24 – Velocidade do Braço para Trajetória 1



Fonte: Autor

Figura 25 – Posição do Braço para Trajetória 1



Fonte: Autor

Vale ressaltar que nesse caso da Figura 25, é notório que o sistema não seguiu a referência desejada, como ilustrado na Figura 14 e possui também altos valores de pico para aceleração e velocidade, como mostrado nas Figuras 23 e 24. A solução encontrada para evitar esses problemas apontados é implementar uma lei de controle, de forma que o sistema siga a referência de trajetória desejada.

#### 4.7 Controle PD

Para projetar o controle PD para o sistema robótico que foi modelado de forma computacional via Python, necessitou-se dos parâmetros do sistema massa-mola equivalente. Para obter esses parâmetros, foi realizado ensaios experimentais de identificação a partir do braço de bancada e os resultados obtidos para cada link são aproximadamente (VASCONCELOS *et al.*, 2020):

Tabela 2 – Parâmetros Estimados do Braço de Bancada

Parâmetro	Link 1	Link 2
$m(kg)$	0.072	0.239
$b(Nm/(rad/s))$	2.4919	2.3477
$k(Nm/rad)$	114.0636	41.9973

Fonte: Autor (2022).

Através dos parâmetros obtidos, foi possível projetar um controlador PD, tal que a equação do controlador é do tipo:

$$C = K_p + K_v s \quad (4.17)$$

Analisando o *root-locus* dos sistemas identificados via software MATLAB (MATLAB, 2021), foi possível posicionar um zero para garantir a estabilidade de ambos os sistemas, e também, foi possível encontrar um  $K_p$  e  $K_v$  que controlassem ambos sistemas, sendo  $K_p = 2$  e  $K_v = 0.4$ , de forma que ambos sistemas seguissem a referência fornecida. Com isso, é possível agora seguir com a análise do controlador projetado e seu respectivo desempenho diante de uma trajetória para o braço robótico.

## 5 RESULTADOS

Neste momento, será analisado o desempenho do braço robótico em malha aberta e malha fechada, sendo nesse último caso com a lei de controle projetada na Seção 4.7, variando a carga do braço robótico em passos de 50g até uma carga de 250g. Para a aplicação da carga na *Robotics Toolbox*, foi inserido um fator multiplicativo "d" na massa do link 2, em que esse fator será aumentado de acordo com a variação da massa na carga. Em outras palavras, a massa resultante no link 2 será a soma da massa do link 2 com a massa da carga.

Para um caso inicial em que a massa na carga é nula, foi adotado  $d = 1$ . A cada aumento da massa na carga em 50g, foi adicionado a d um valor de 0.26 (que representa 50g da massa do link 2). Ou seja, no fim do experimento, foram utilizados os valores de d tais quais:

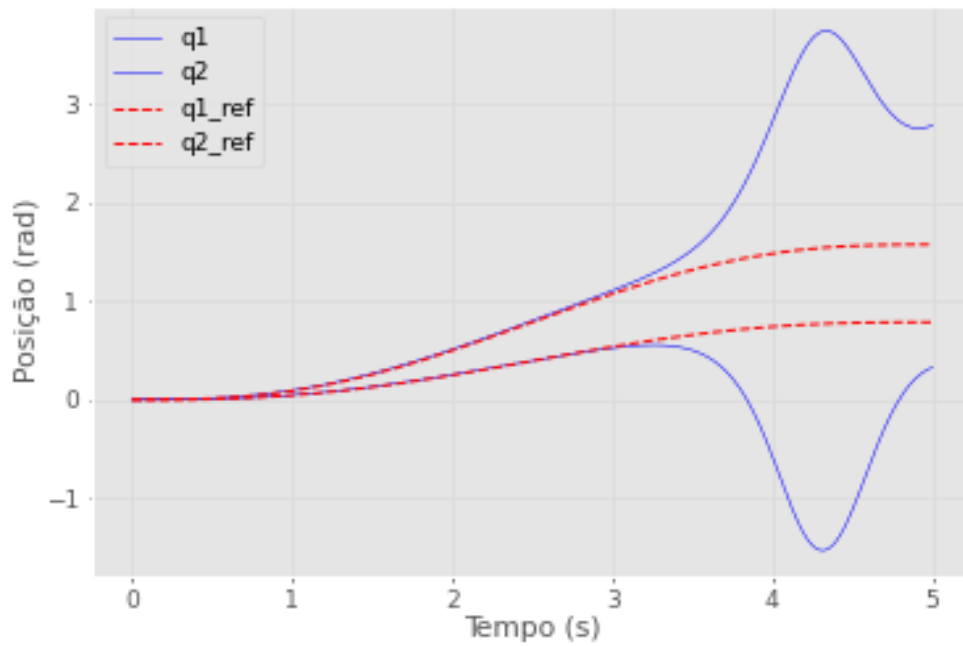
$$d = [1.0, 1.26, 1.52, 1.78, 2.04, 2.30] \quad (5.1)$$

Em que estes valores representam, respectivamente, uma massa de 0g, 50g, 100g, 150g, 200g e 250g adicionados ao link 2. Com isso, será observado o estudo das trajetórias 1 e 2, definidas no Capítulo 4 e será analisado os resultados, realizando as comparações entre malha aberta e malha fechada (aplicando a lei de controle). De forma a simplificar a quantidade de gráficos nesse trabalho, não será exibido todos os casos, mas será apurado a quantidade necessária para validar os resultados.

### 5.1 Análise da Trajetória 1

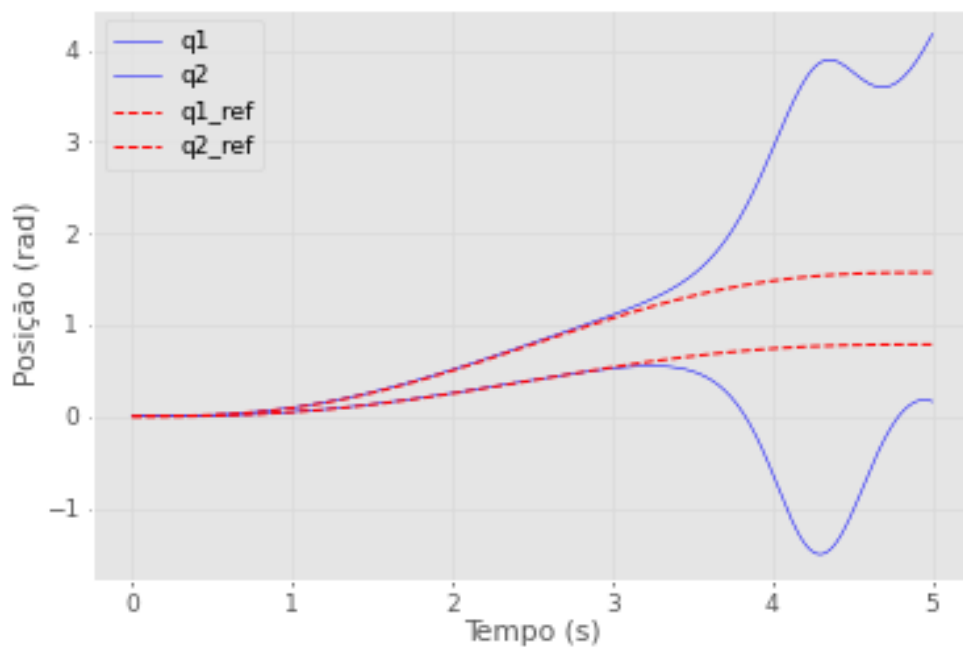
Com base na trajetória definida na Figura 14, foram aplicados todos os valores de d para o sistema em malha aberta e foi possível obter os seguintes resultados de posição angular:

Figura 26 – Posição Angular do Braço Robótico em Malha aberta sem carga (Trajetória 1)



Fonte: Autor

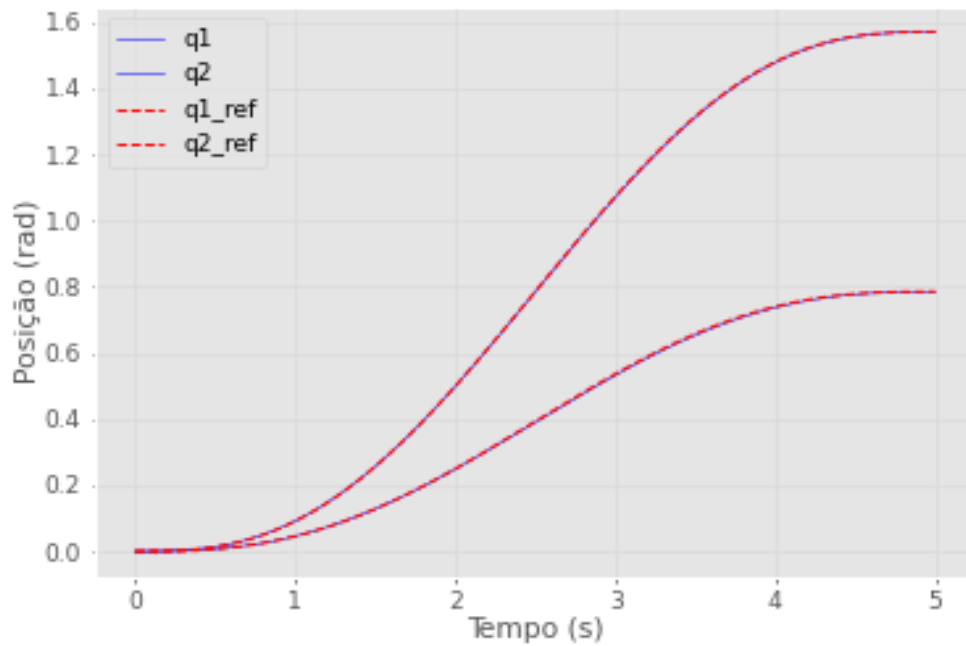
Figura 27 – Posição Angular do Braço Robótico em Malha aberta com carga de 50g (Trajetória 1)



Fonte: Autor

A partir do observado nas Figuras 26 e 27, é possível notar que tanto no caso sem carga, quanto a partir do caso em que a carga de 50g é inserida no braço, o sistema se demonstrou instável e não seguiu a referência para nenhum dos casos, e com isso, reforça-se a utilização do controlador para solucionar esse problema. Ao aplicar a lei de controle obtida em 4.7 nesse sistema, é possível ter os seguintes resultados:

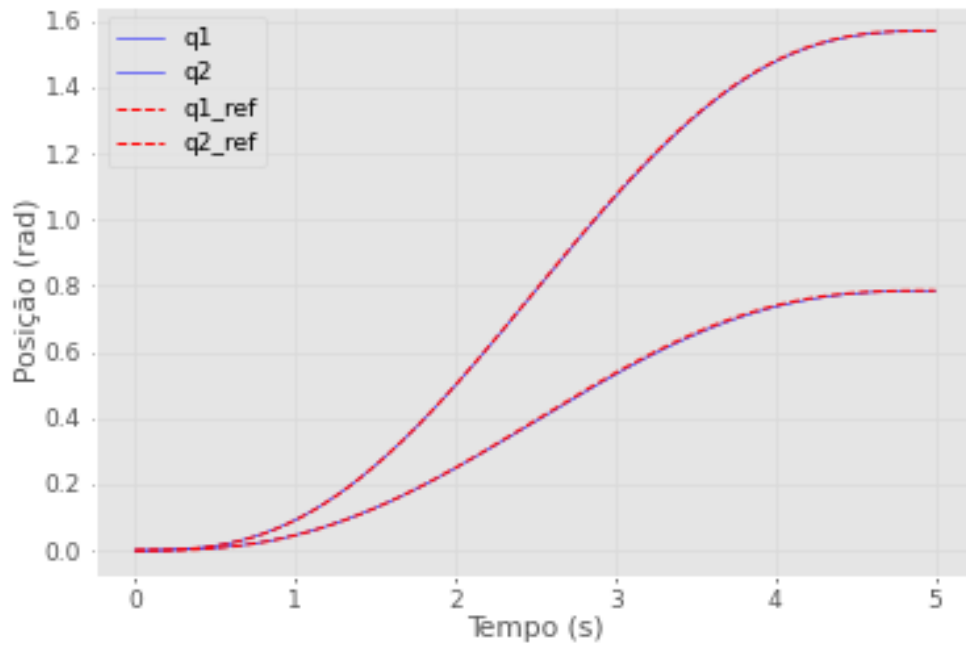
Figura 28 – Posição Angular do Braço Robótico em Malha Fechada sem carga (Trajetória 1)



Fonte: Autor

Percebe-se inicialmente que já para o primeiro caso, em que o braço robótico está sem carga, a utilização do controle já permitiu que o sistema seguisse a referência de posição desejada. Agora, é necessário a validação dos seguintes casos, em que é aplicado diferentes cargas ao sistema.

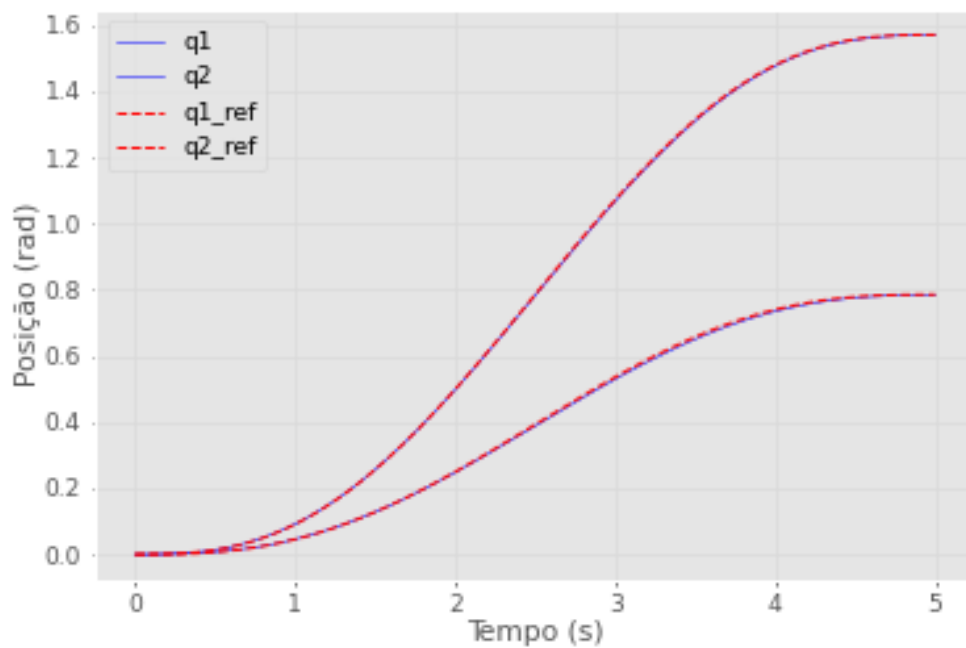
Figura 29 – Posição Angular do Braço Robótico em Malha Fechada com carga de 50g (Trajetória 1)



Fonte: Autor

Para a primeira aplicação de carga, com o valor de 50 gramas, o controlador mostrou bom desempenho e o sistema seguiu a referência desejada.

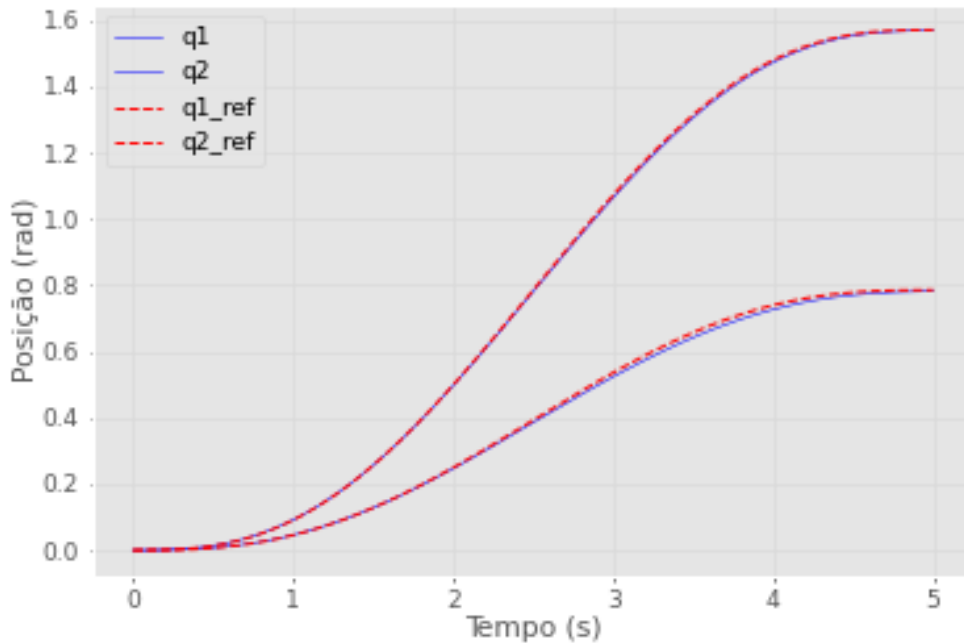
Figura 30 – Posição Angular do Braço Robótico em Malha Fechada com carga de 100g (Trajetória 1)



Fonte: Autor

Com o aumento da carga para 100g, ainda sim foi possível notar um bom desempenho do controlador e foi possível verificar que o sistema seguiu a referência desejada.

Figura 31 – Posição Angular do Braço Robótico em Malha Fechada com carga de 250g (Trajetória 1)



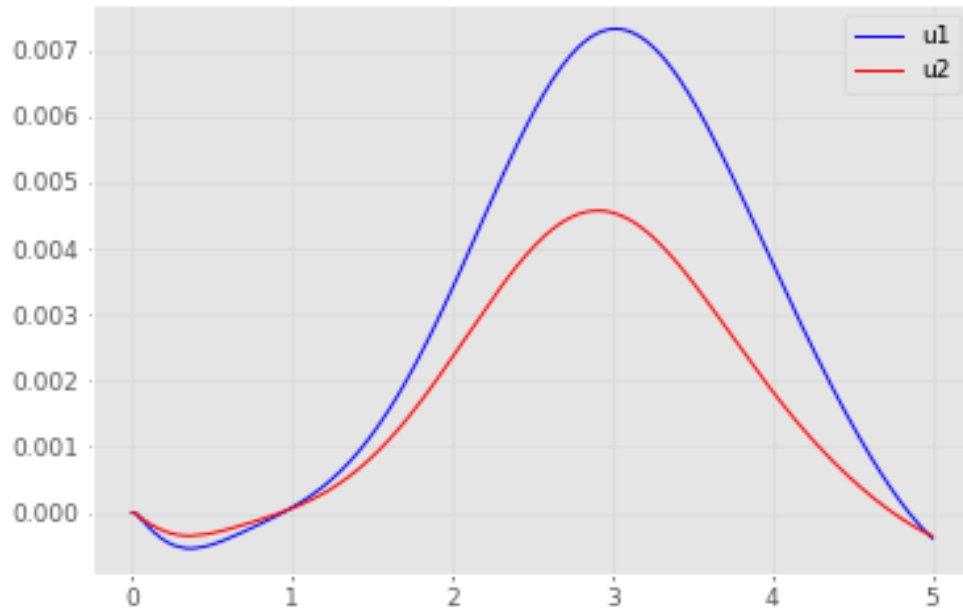
Fonte: Autor

Para uma carga de massa de 250g, aproximadamente a massa do manipulador robótico, foi possível notar novamente que o controlador garantiu que o sistema seguisse a referência desejada para a trajetória definida. Portanto, observa-se que a aplicação da lei de controle foi suficiente para projetar um controle que é robusto para essa faixa de operação e que garante que o braço siga a trajetória definida para todos esses casos.

Agora será analisado os valores de torque da parcela de feedback (calculado através da lei de controle), o torque de malha aberta, sendo representado pelo torque de feedforward (calculado pela RNE da *Robotics Toolbox*), e o torque de malha fechada, sendo definido pela soma do torque referente à parcela de feedforward com a parcela de feedback:

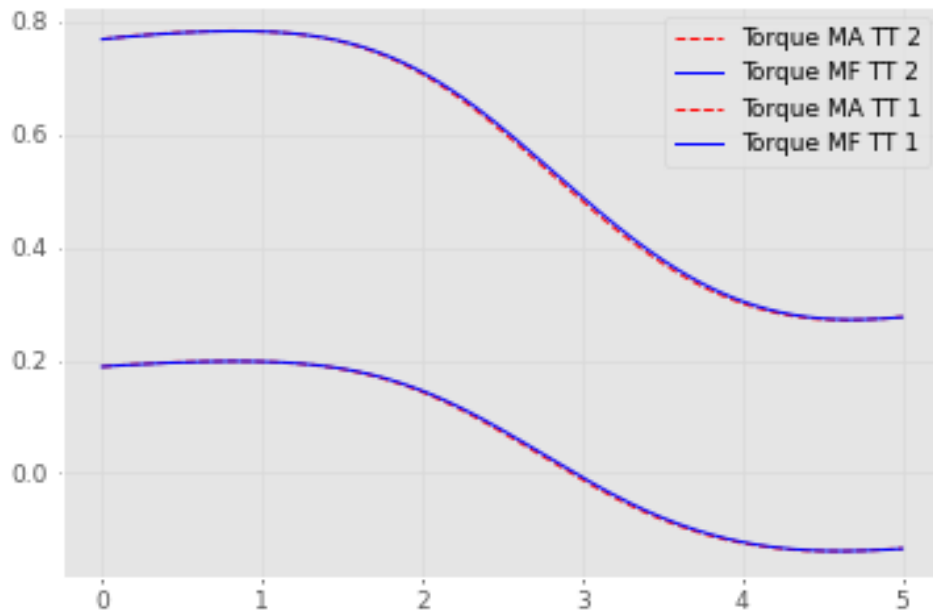


Figura 32 – Torque de Feedback para o Braço Robótico sem carga (Trajetória 1)



Fonte: Autor

Figura 33 – Torque MA vs Torque MF do Braço Robótico sem carga (Trajetória 1)

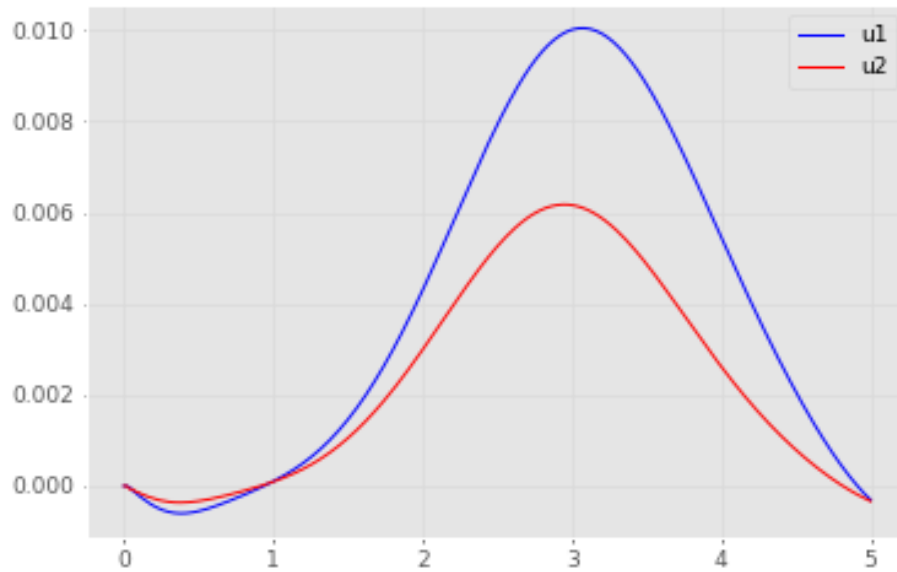


Fonte: Autor

É possível verificar a partir das Figuras 34 e 35, que por menor que seja a influência do torque de feedback, ele é suficiente para garantir que o sistema opere de forma adequada.

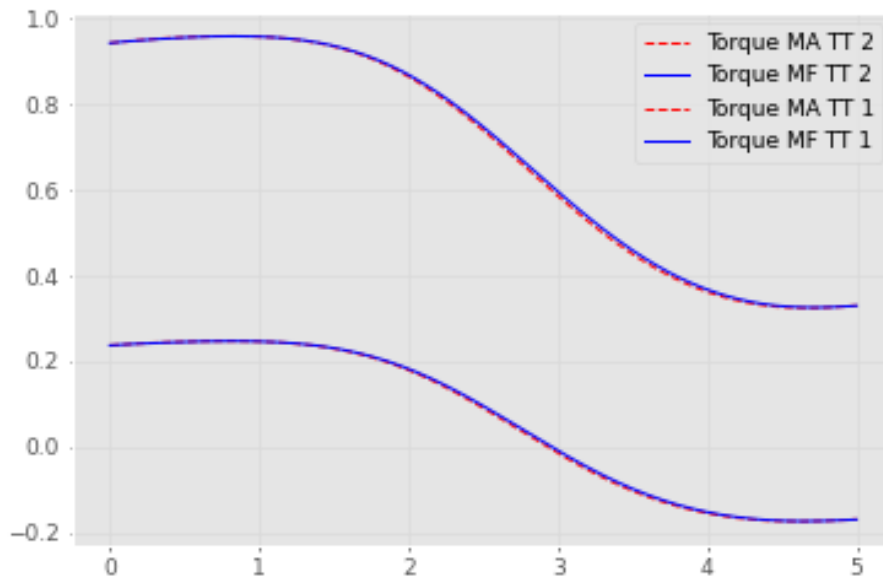
Agora, segue-se a análise para os casos em que é inserido uma carga ao sistema robótico, com o objetivo de analisar o impacto da variação de torque de acordo com o valor da massa associado a carga acrescentada:

Figura 34 – Torque de Feedback para o Braço Robótico com carga 50g (Trajetória 1)



Fonte: Autor

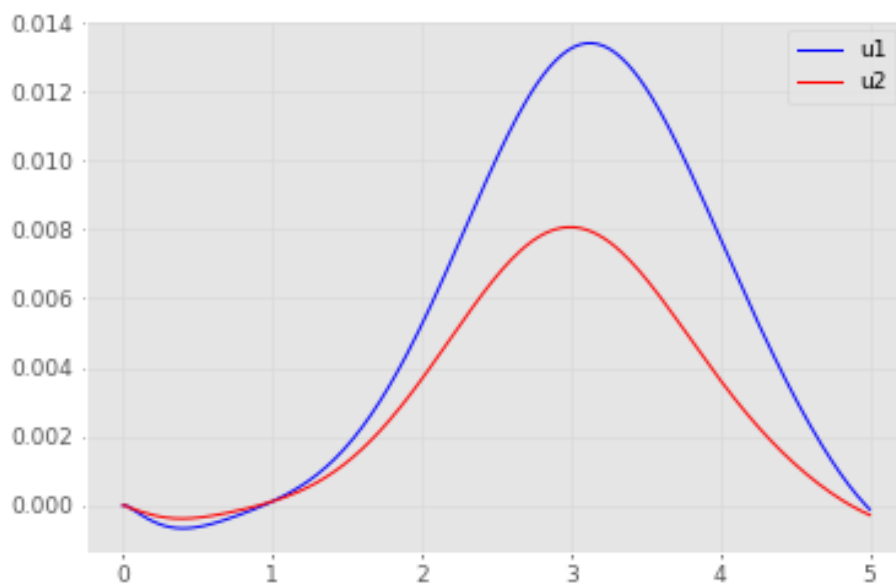
Figura 35 – Torque MA vs Torque MF do Braço Robótico com carga 50g (Trajetória 1)



Fonte: Autor

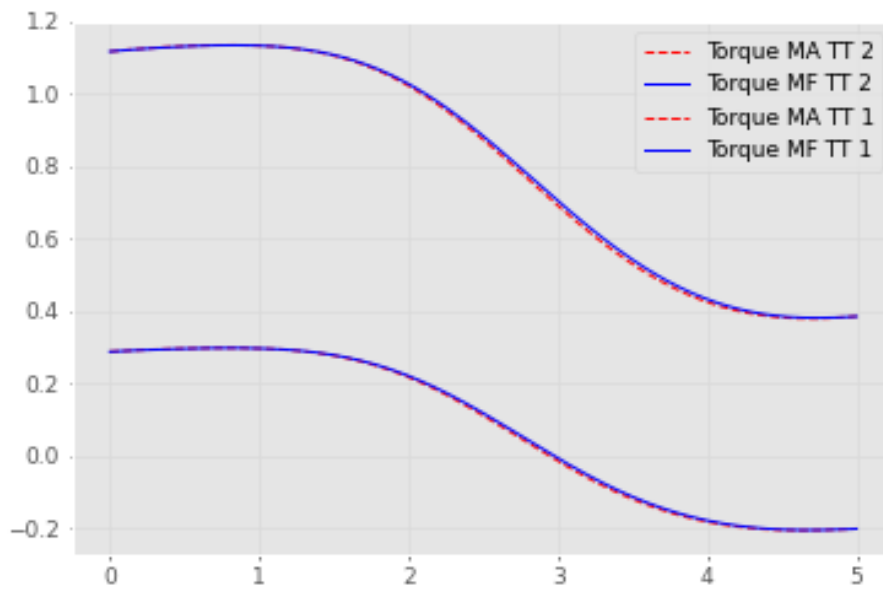
Ao inserir a primeira carga, com a massa de 50g, é possível notar que tanto o torque de feedback quanto o torque total aumentaram, o que era esperado, pois o sistema necessita de uma força maior para realizar o movimento.

Figura 36 – Torque de Feedback para o Braço Robótico com carga 100g (Trajetória 1)



Fonte: Autor

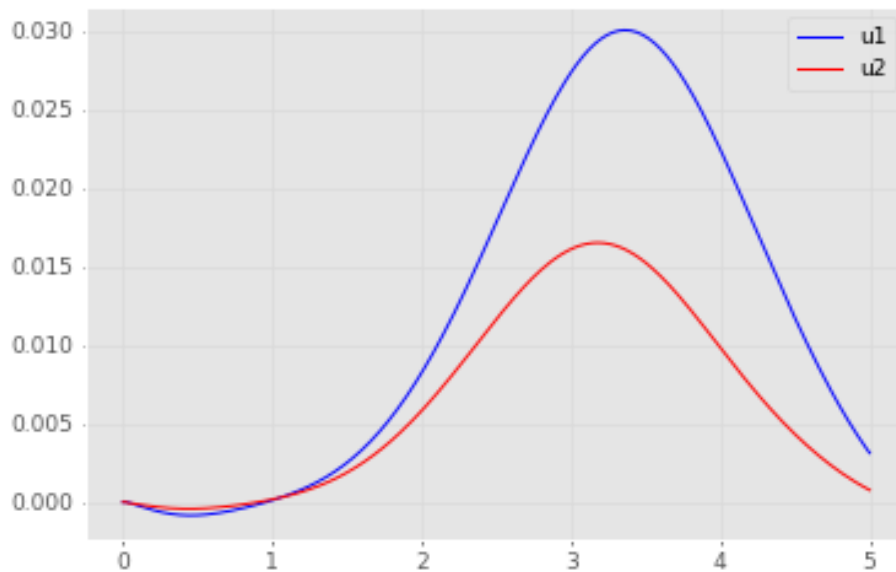
Figura 37 – Torque MA vs Torque MF do Braço Robótico com carga 100g (Trajetória 1)



Fonte: Autor

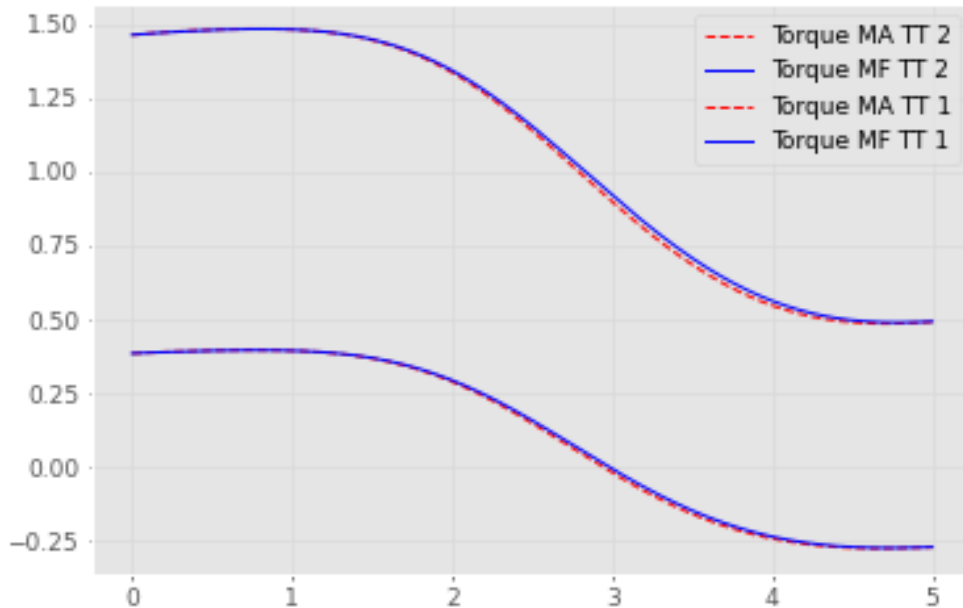
Ao aumentar o valor da massa na carga, verifica-se novamente o aumento do torque de feedback e do torque total.

Figura 38 – Torque de Feedback para o Braço Robótico com carga 250g (Trajetória 1)



Fonte: Autor

Figura 39 – Torque MA vs Torque MF do Braço Robótico com carga 250g (Trajetória 1)



Fonte: Autor

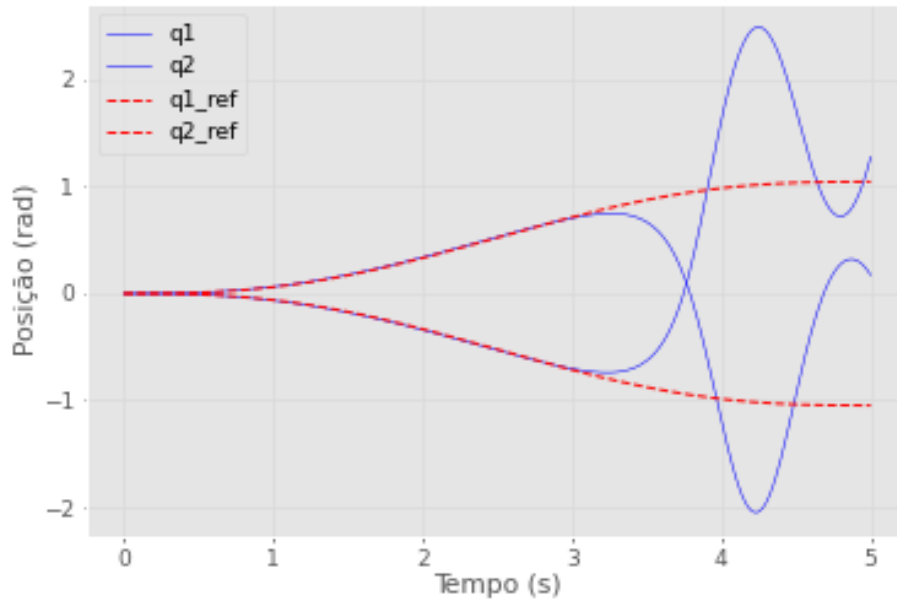
Ao inserir uma carga de massa semelhante a massa do manipulador robótico, nota-se que o torque total praticamente dobrou, comparado ao caso sem carga. A partir da Figura 39, é possível ver uma influência maior do torque de feedback comparado aos casos anteriores.

Com base no observado nas Figuras 32 a 39, verifica-se que por menor que seja a influência do sinal de torque da parcela de feedback que é somado ao torque de feedforward, para todos os casos acima, a parcela de feedback é suficiente para garantir que a posição angular siga a trajetória de referência e garanta a estabilidade do sistema, como foi possível verificar nas Figuras 28 a 31.

## 5.2 Análise da Trajetória 2

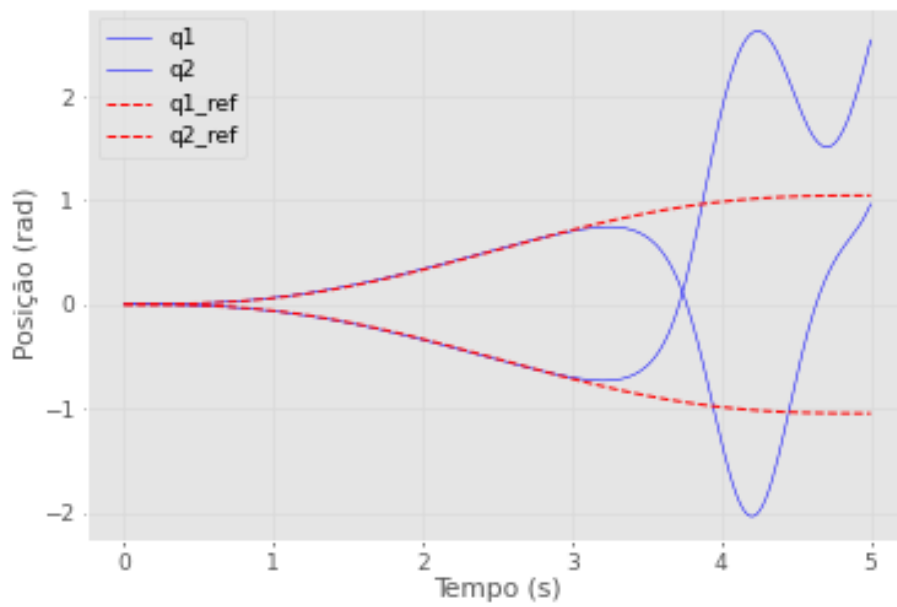
De forma análoga à análise da Trajetória 1, será primeiro analisado os resultados de malha aberta para o braço robótico com a trajetória de referência definida na Figura 15. Após a análise via *Robotics Toolbox*, foi possível obter os seguintes resultados de posição angular:

Figura 40 – Posição Angular do Braço Robótico em Malha aberta sem carga (Trajetória 2)



Fonte: Autor

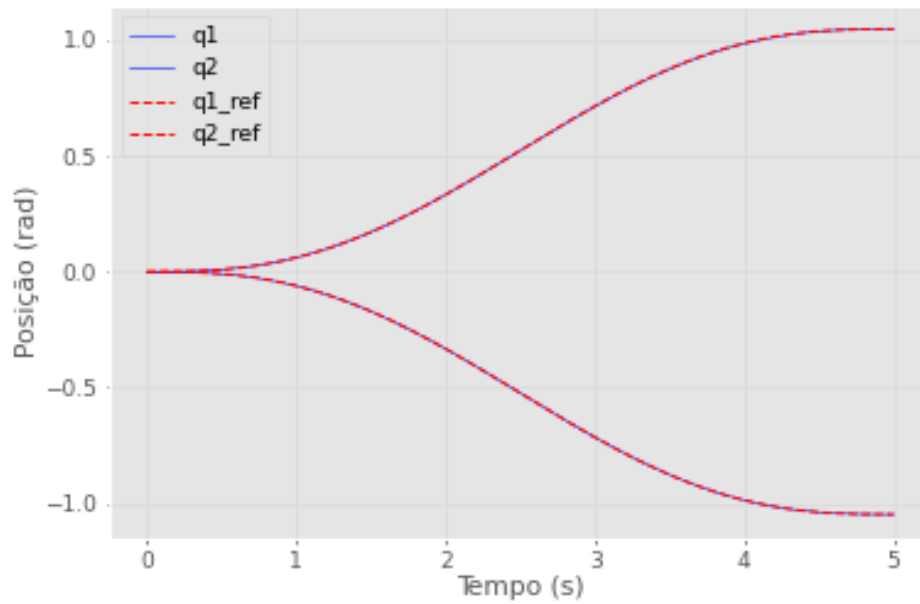
Figura 41 – Posição Angular do Braço Robótico em Malha aberta com carga de 50g (Trajetória 2)



Fonte: Autor

Para essa trajetória, também foi observado que o sistema se demonstrou instável e sem seguir a trajetória de referência tanto para o caso sem carga, quanto para o caso em que uma carga foi adicionada. Para resolver esse problema, foi adicionada a lei de controle como no caso anterior. Com a adição da parcela feedback no sistema, foi possível obter os seguintes resultados:

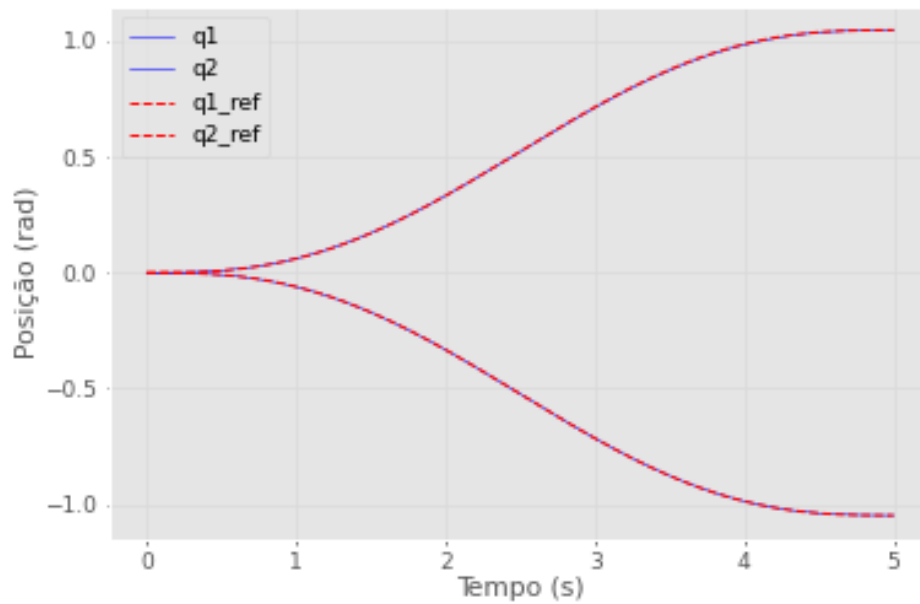
Figura 42 – Posição Angular do Braço Robótico em Malha Fechada sem carga (Trajetória 2)



Fonte: Autor

De forma similar à primeira trajetória, percebe-se que logo no primeiro caso, em que o braço robótico está sem carga, a utilização do controle permitiu que o sistema seguisse a referência de posição desejada. Agora, será validado os seguintes casos, em que é aplicado diferentes cargas ao sistema.

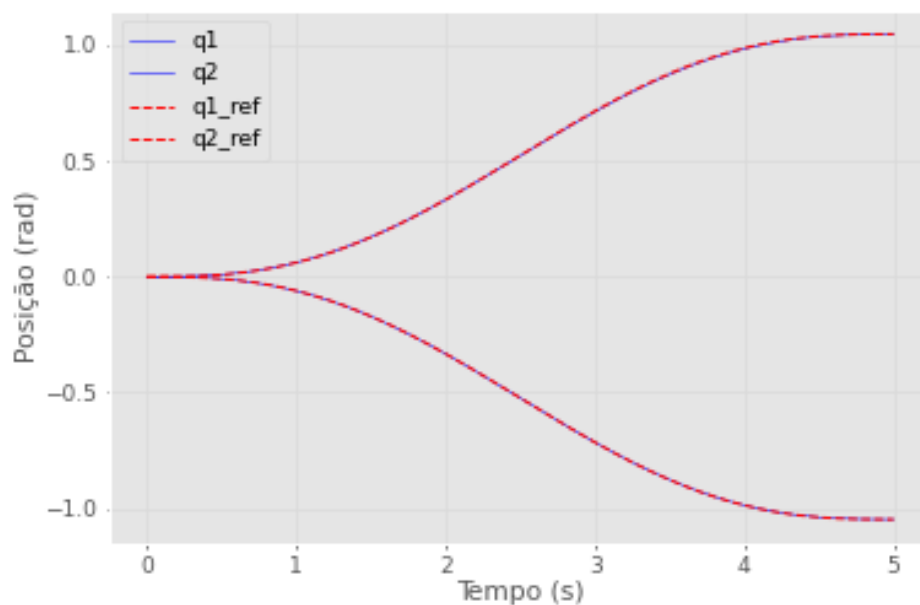
Figura 43 – Posição Angular do Braço Robótico em Malha Fechada com carga de 50g (Trajetória 2)



Fonte: Autor

Também de forma similar à primeira trajetória, ao aplicar a primeira carga, com o valor de 50 gramas, o controlador mostrou bom desempenho e o sistema seguiu a referência desejada.

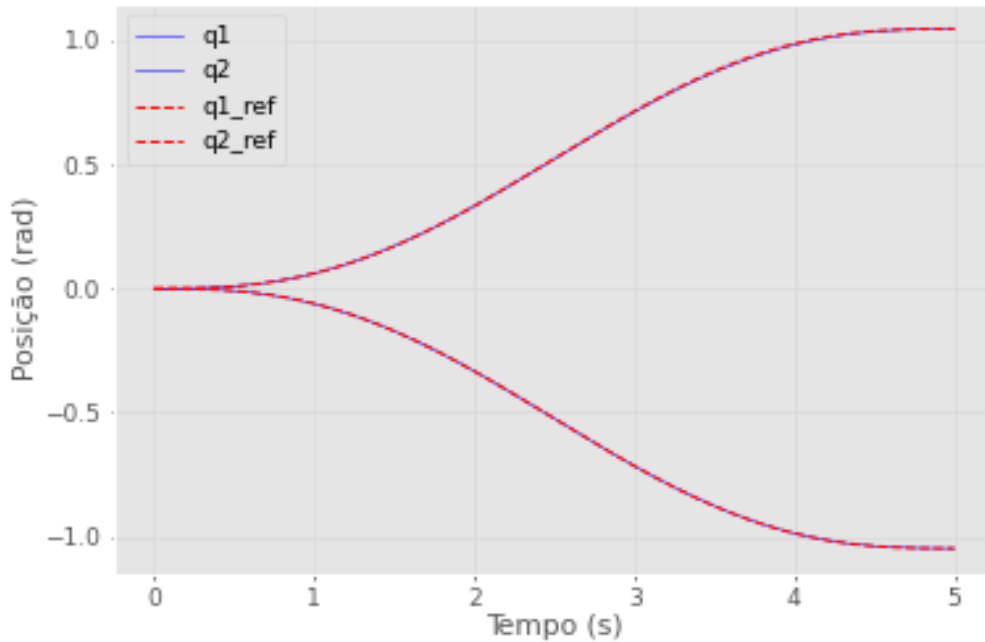
Figura 44 – Posição Angular do Braço Robótico em Malha Fechada com carga de 100g (Trajetória 2)



Fonte: Autor



Figura 45 – Posição Angular do Braço Robótico em Malha Fechada com carga de 250g (Trajetória 2)

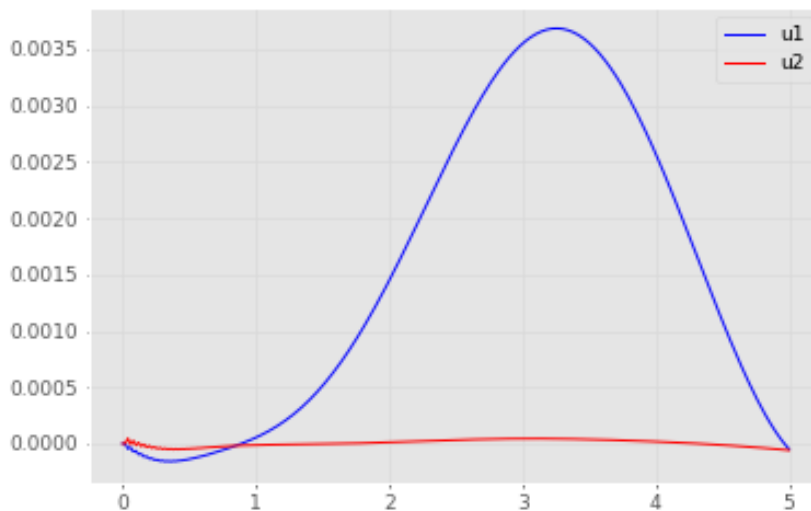


Fonte: Autor

Ao ser aplicado as cargas de 100g e 250g, como representado nas Figuras 44 e 45, verifica-se que mesmo com o aumento de carga, até uma carga de aproximadamente a massa do braço, o sistema demonstrou bom desempenho e seguiu a referência em todos os casos.

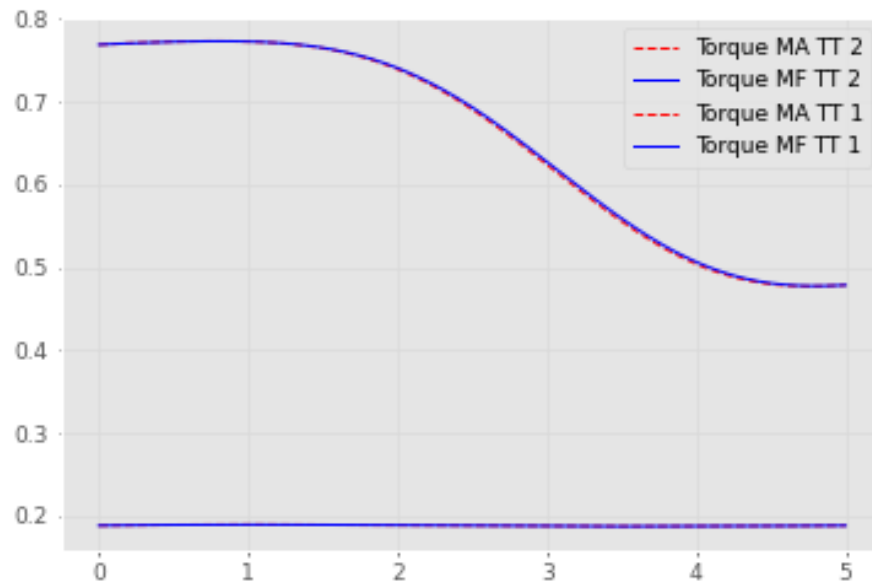
Agora analisa-se os torques de feedback, feedforward e total:

Figura 46 – Torque de Feedback para o Braço Robótico sem carga (Trajetória 2)



Fonte: Autor

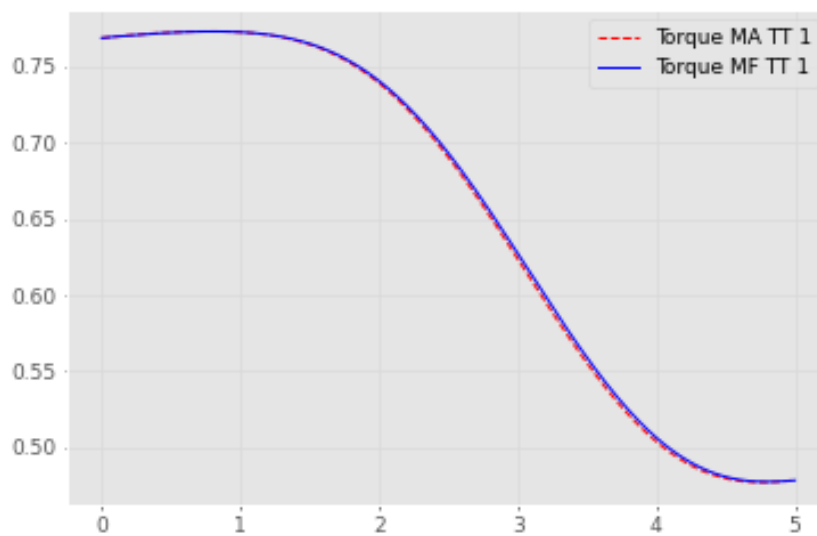
Figura 47 – Torque MA vs Torque MF do Braço Robótico sem carga (Trajetória 2)



Fonte: Autor

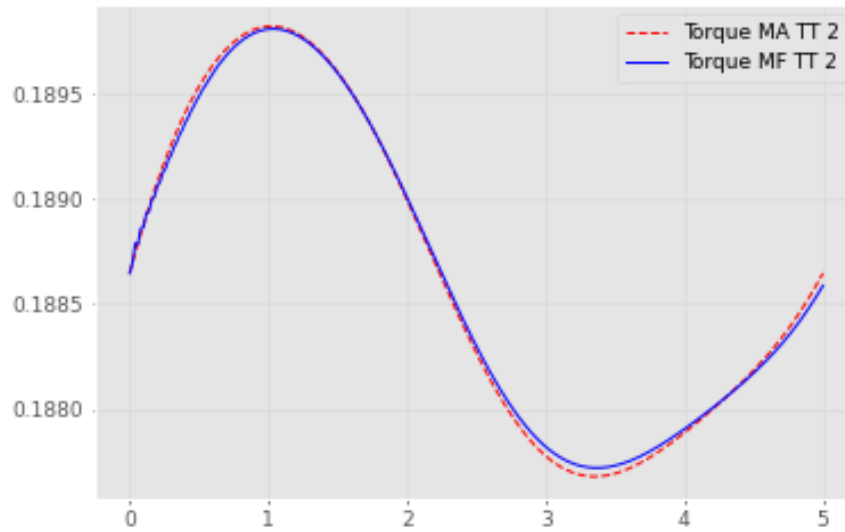
Nesse caso, o torque do segundo link é praticamente constante, devido a característica mencionada no Capítulo 4 e devido a dificuldade de visualizar a sua variação na Figura 47, os torque para cada link serão analisados e gráficos diferentes:

Figura 48 – Torque MA vs Torque MF do Braço Robótico sem carga para o Link 1 (Trajetória 2)



Fonte: Autor

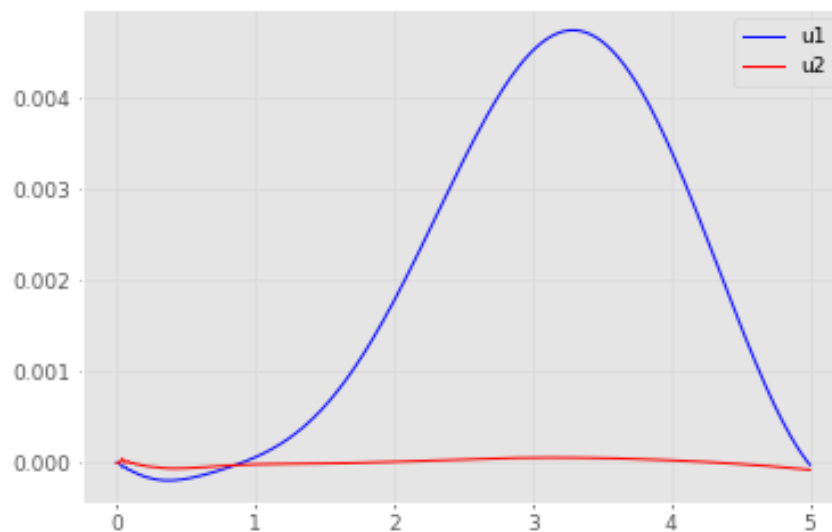
Figura 49 – Torque MA vs Torque MF do Braço Robótico sem carga para o Link 2 (Trajetória 2)



Fonte: Autor

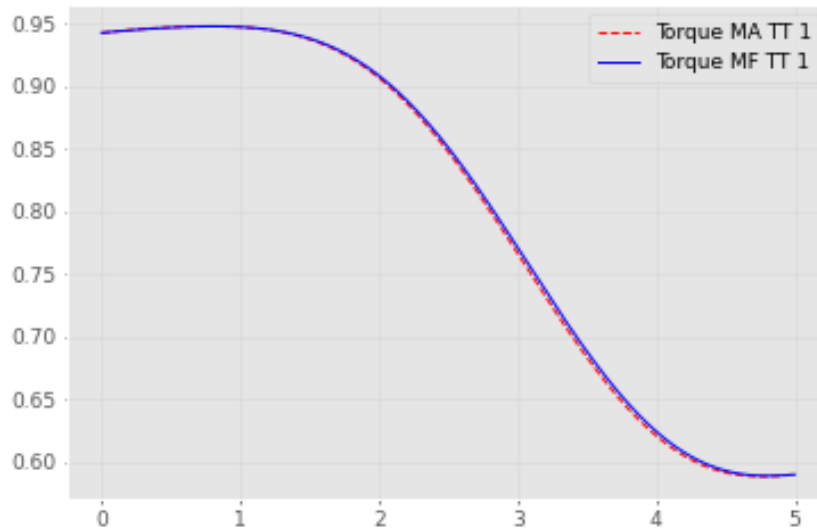
Com base nas Figuras 46 a 49, verifica-se que a influência do torque de feedback é menor do que ao caso da trajetória 1, principalmente no link 2, em que este é praticamente nulo. Apesar disso, a contribuição da parcela de feedback é suficiente para garantir a operação do sistema de forma desejada. Agora, seguindo para os casos em que a carga é acrescentada:

Figura 50 – Torque de Feedback para o Braço Robótico com carga de 50g (Trajetória 2)



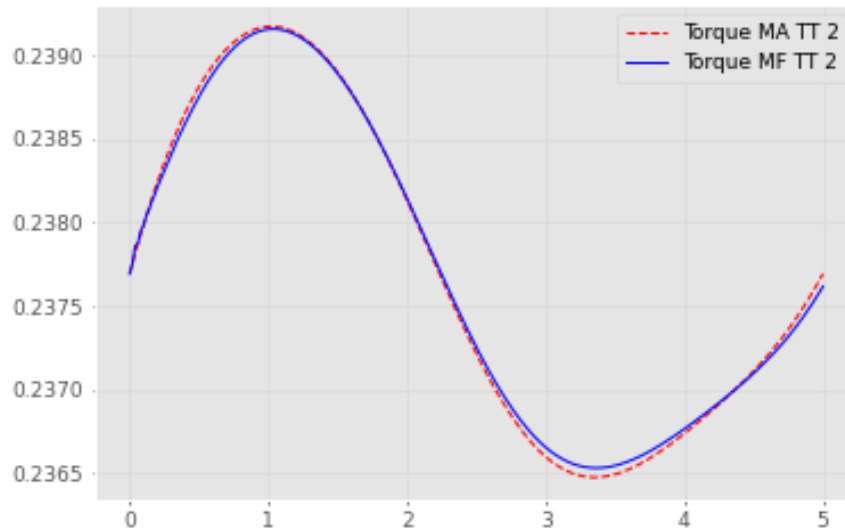
Fonte: Autor

Figura 51 – Torque MA vs Torque MF do Braço Robótico com carga de 50g para o Link 1(Trajectoria 2)



Fonte: Autor

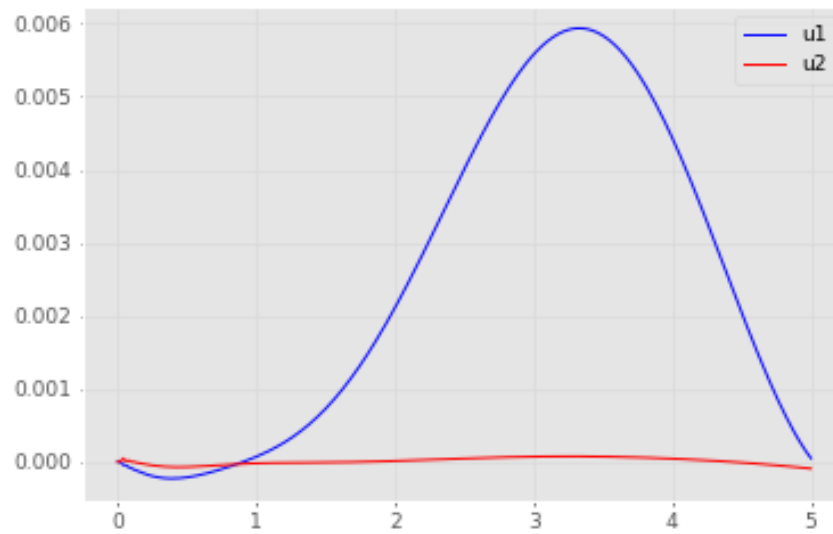
Figura 52 – Torque MA vs Torque MF do Braço Robótico com carga de 50g para o Link 2 (Trajetória 2)



Fonte: Autor

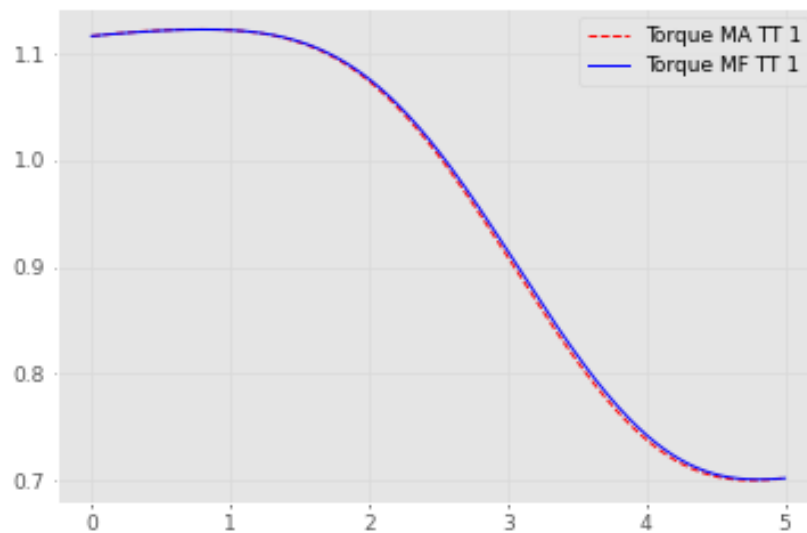
Para esse caso em que a carga possui uma massa de 50g, analisando as Figuras 50 a 52 é possível notar que o torque de feedforward aumenta novamente com o acréscimo da carga, assim como a parcela de feedback, de forma a garantir que o sistema siga a trajetória de referência desejada.

Figura 53 – Torque de Feedback para o Braço Robótico com carga de 100g (Trajetória 2)



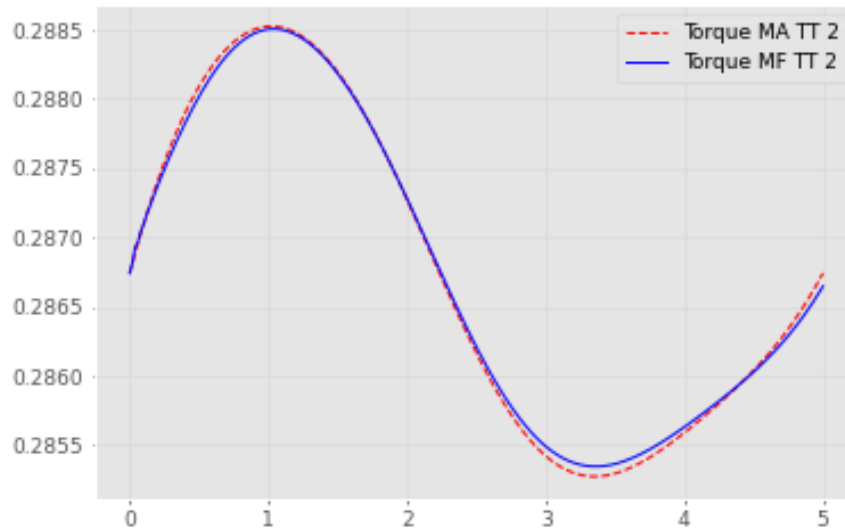
Fonte: Autor

Figura 54 – Torque MA vs Torque MF do Braço Robótico com carga de 100g para o Link 1(Trajetoária 2)



Fonte: Autor

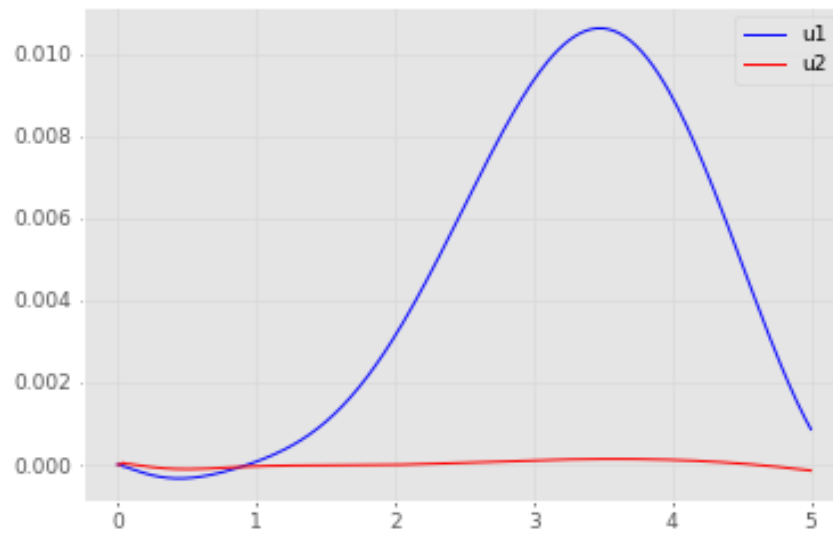
Figura 55 – Torque MA vs Torque MF do Braço Robótico com carga de 100g para o Link 2(Trajétoria 2)



Fonte: Autor

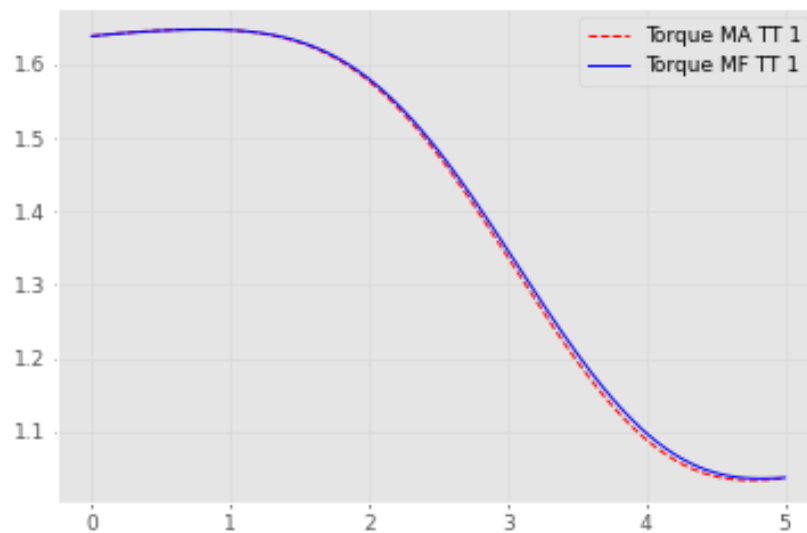
Seguindo a análise para o caso em que a massa da carga é de 100g, verifica-se mais uma vez através das Figuras 53 a 55, o aumento do torque total, de feedforward e de feedback, como esperado. No caso do torque de feedback, o torque do link 1 aumenta de forma bem mais notória que o do link 2, em que esse ainda permanece praticamente nulo.

Figura 56 – Torque de Feedback para o Braço Robótico com carga de 250g (Trajetória 2)



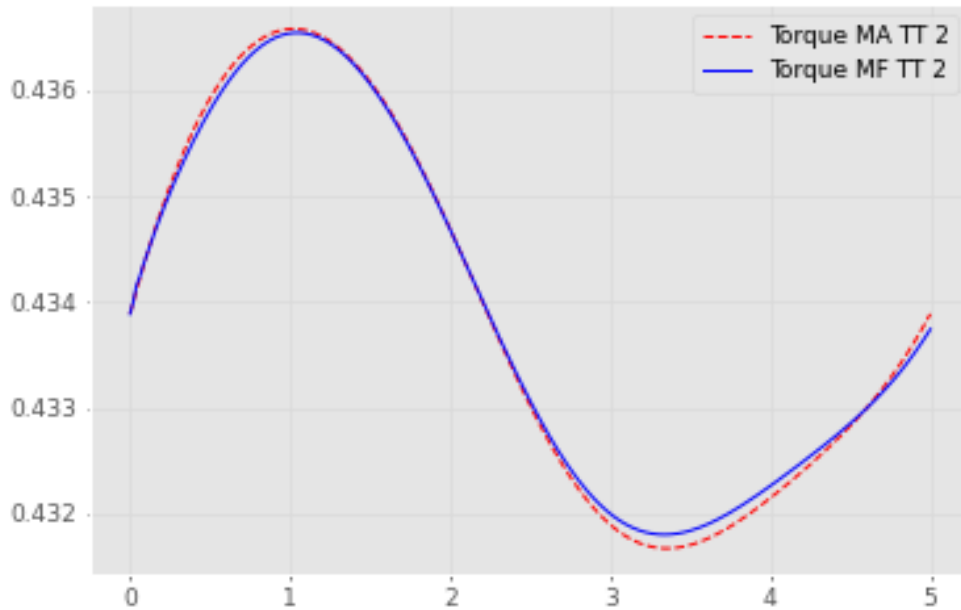
Fonte: Autor

Figura 57 – Torque MA vs Torque MF do Braço Robótico com carga de 250g para o Link 1(Trajétória 2)



Fonte: Autor

Figura 58 – Torque MA vs Torque MF do Braço Robótico com carga de 250g para o Link 2 (Trajetória 2)



Fonte: Autor

Por fim, para o caso em que a massa da carga é de 250g, o torque total e torque de feedforward aproximadamente dobraram para ambos links, comparado ao caso sem carga. Além disso, foi possível notar também o aumento dos torques de feedback, principalmente para o link 1, em que ao analisar a Figura 57, é possível notar uma diferença visual no gráfico entre o torque total e o torque de feedforward

Analisando as Figuras 42 a 45, foi possível verificar que o projeto de controle garantiu o seguimento de referência em todos os casos, inclusive nos casos que houve uma carga associada ao link 2. Verificando também as Figuras 46 a 58, foi possível notar que por menor que seja a influência da parcela de feedback, ela ajuda a corrigir o erro de posição do sistema de forma a garantir o seguimento de referência do sistema robótico.



## 6 CONCLUSÕES E TRABALHOS FUTUROS

Primeiramente, é válido ressaltar que foi possível realizar a modelagem computacional de um sistema robótico real via simulação em Python através da *Robotics Toolbox*. Esse resultado foi validado através dos cálculos teóricos e é de grande importância, pois, a validação da *Toolbox* permite o estudo de outros modelos robóticos que muitas vezes não são acessíveis aos alunos, professores e profissionais, sendo esses modelos industriais, ou até mesmo construídos pelo próprio usuário.

Além disso, foi possível projetar um controlador, do tipo PD, que garantisse o seguimento de referência para o braço de bancada escolhido e esse resultado foi validado por duas trajetórias e com uma massa atuando como carga que variava. O controle foi robusto para a faixa de operação que o braço operou e garantiu o seguimento de referência para todos os casos estudados.

A partir dos resultados obtidos, há diversos trabalhos futuros que podem ser gerados a partir deste. O primeiro, é o estudo de novos modelos de sistemas robóticos, visto que a *Toolbox* possui os próprios modelos e permite a criação de novos modelos, como foi o caso desse trabalho. Outra possibilidade é o estudo de novos controladores que substituam o PD e aplicar outros tipos de controladores que englobem as não-linearidades do sistema, tais como controle adaptativo, Controle com Parâmetros Lineares Variantes (LPV) ou  $H_\infty$ . Também é possível a implementação prática do braço com o controle projetado, de forma a validar os resultados desse trabalho com os resultados experimentais.

Também é válido como trabalhos futuros a redução das simplificações feitas nesse trabalho, como por exemplo, a redução do número de links, o que possibilitaria a utilização do braço em aplicações industriais, como uma paletizadora.

Por fim, pode-se afirmar que os objetivos desse trabalho foram alcançados de forma satisfatória.

## REFERÊNCIAS

- AL-SHUKA, H. F.; LEONHARDT, S.; ZHU, W.-H.; SONG, R.; DING, C.; LI, Y. Active impedance control of bioinspired motion robotic manipulators: An overview. **Applied bionics and biomechanics**, Hindawi, v. 2018, 2018.
- AUTODESK. (2018). San Rafael, California: Autodesk Inc., 2018.
- CHENG, Y.; YIN, C.; XUE, J.; ZHANG, B.; HUANG, X.; HU, B.; SHI, X. Design of trajectory tracking control for adaptive intelligent detection based on 6-dof manipulator. In: IEEE. **2017 Chinese Automation Congress (CAC)**. [S. l.], 2017. p. 7255–7260.
- CHOI K.; KIM, S. **Robot Mechanics and Control, Part II**. 2014. Acesso em: 08 Jan 2022. Disponível em: <https://courses.edx.org/courses/SNUx/SNU446.345.2x/2T2014/course/>.
- CORKE, P. I. **Robotics, Vision & Control: Fundamental Algorithms in MATLAB**. Second. [S. l.]: Springer, 2017. ISBN 978-3-319-54413-7.
- CRAIG, J. J. **Robótica**. 3a. ed. São Paulo: Pearson Education do Brasil, 2013. ISBN 978-8581431284.
- DENAVIT, J.; HARTENBERG, R. S. A kinematic notation for lower-pair mechanisms based on matrices. American Society of Mechanical Engineers, 1955.
- FILGUEIRAS, I. A. G. **SIMULAÇÃO E CONTROLE DE UM MANIPULADOR ROBÓTICO**. Monografia (TCC) – Universidade Federal do Ceará, 2019.
- GAIDHANE, P. J.; NIGAM, M. J.; KUMAR, A.; PRADHAN, P. M. Design of interval type-2 fuzzy precompensated pid controller applied to two-dof robotic manipulator with variable payload. **ISA transactions**, Elsevier, v. 89, p. 169–185, 2019.
- MATLAB. (R2021a). Natick, Massachusetts: The MathWorks Inc., 2021.
- MOBERG, S. **Modeling and Control of Flexible Manipulators**. 2010. ISBN 978-8581431284.
- PROPMASS (MASSPROP) (comando). <https://knowledge.autodesk.com/pt-br/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2015/PTB/AutoCAD-Core/files/GUID-CAA51229-293E-4A0C-BFF3-93226252CF13-htm.html>. Acesso em: 06 de Nov. 2021.
- SICILIANO, B.; KHATIB, O.; KRÖGER, T. **Springer handbook of robotics**. [S. l.]: Springer, 2008. v. 200.
- SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *et al.* **Robot modeling and control**. [S. l.]: Wiley New York, 2006. v. 3.
- URREA, C.; KERN, J.; ALVARADO, J. Design and evaluation of a new fuzzy control algorithm applied to a manipulator robot. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 10, n. 21, p. 7482, 2020.
- VASCONCELOS, F. J. de S.; FILGUEIRAS, I. de A. G.; PAIVA, D. A.; CORREIA, W. B. A simple procedure for modeling and identification of a test bench 4-dof manipulator. **CBA.**, 2020.

## APÊNDICE A – CÓDIGOS DESENVOLVIDOS

```
1 from roboticstoolbox import DHRobot, RevoluteDH, ERobot,
   ELink, ETS
2 from math import pi, sin, cos
3 import numpy as np
4 import pandas as pd
5 np.set_printoptions(linewidth=100, suppress=True)
6 import matplotlib.pyplot as plt
7 from matplotlib import cm
8 import roboticstoolbox as rtb
9
10 d = 1.00
11 # link 1
12 m1 = 0.07576
13 l1 = 0.255
14 lc1 = 0.122
15 I_1 = [0.00116775, 0.00175608, 0.00160974, -0.00058644,
16 0.00056025, 0.00068607]
17
18 # link 2
19 m2 = d*0.1923
20 l2 = 0.199
21 lc2 = 0.099
22 I_2 = [0.0061722, 0.01746360,
23 0.0161109, -0.0057024, -0.00037395, -0.00096714]
24 #Altura da base ate o servo do link 1
25 altura = (11.75)*(10^-2)
26
27 # global parameters
28
```

```
29 g = 9.81
30
31 L1 = RevoluteDH(a=l1, m=m1, r=[-lc1, 0, altura], I=I_1)
32 L2 = RevoluteDH(a=l2, m=m2, r=[-lc2, 0, altura], I=I_2)
33 robot = DHRobot([L1, L2], gravity=[0, -g, 0])
34
35 print(robot)
```

Código-fonte 1 – Definição do Braço de Bancada

```
1 t = np.arange(0.0,5.02,0.02) #Definicao do tempo de
   simulacao
2 qt = rtb.tools.trajectory.jtraj([0,0], [pi/3,-pi/3], t)
3 #Definicao da trajetoria a partir dos angulos finais e
   iniciais
4 rtb.tools.trajectory.qplot(qt.q, block=False) #Plot grafico
   de trajetoria
```

Código-fonte 2 – Definição da Trajetória do Braço

```
1 tau = robot.nofriction().rne(qt.q,qt.qd,qt.qdd) #Calculo da
   RNE
2
3 plt.figure()
4
5 plt.plot(t, tau[:,0], 'b')#Plot dos graficos da RNE
6 plt.plot(t, tau[:,1], 'b')#Plot dos graficos da RNE
```

Código-fonte 3 – Cálculo da RNE para a Trajetória definida

```

1 q = np.array([[0,0]])
2 qd = np.array([[0,0]])
3
4 qdd_0 = np.array([0,0])
5 qd_0 = np.array([0,0])
6 q_0 = np.array([0,0])
7
8 for x in range(251):
9     if x == 0:
10         qdd = robot.nofriction().accel(q[x],qd[x],tau[x])
11         torque_MA = tau[x]
12     else:
13         torque_MA = np.vstack((torque_MA,tau[x]))
14         qdd = np.vstack((qdd,robot.nofriction().accel(q[x
-1],qd[x-1],tau[x])))
15         v = qd_0 + qdd[x]*0.02
16         qd = np.vstack((qd, v))
17         theta = q_0 + qd[x]*0.02
18         q = np.vstack((q, theta))
19         qdd_0 = qdd[x]
20         qd_0 = qd[x]
21         q_0 = q[x]
22
23 rtb.tools.trajectory.qplot(q, block=False)

```

Código-fonte 4 – Cálculo da Posição do Braço a partir da RNE em Malha Aberta

```

1 q = np.array([[0,0]])
2 qd = np.array([[0,0]])
3
4 qdd_0 = np.array([0,0])
5 qd_0 = np.array([0,0])
6 q_0 = np.array([0,0])
7
8
9 for x in range(251):
10     if x == 0:
11         tau_D = 0.4*(qt.qd[x]-qd[x])
12         tau_P = 2*(qt.q[x]-q[x])
13         tau_MF = tau[x] + tau_D[x]+tau_P[x]
14         qdd = robot.nofriction().accel(q[x],qd[x],tau_MF)
15         torque_MF = tau_MF
16         u = [0,0]
17     else:
18         tau_MF = tau[x]+tau_P+tau_D
19         sinal_controle = tau_P + tau_D
20         u = np.vstack((u,sinal_controle))
21         torque_MF = np.vstack((torque_MF,tau_MF))
22         qdd = np.vstack((qdd,robot.nofriction().accel(q[x
-1],qd[x-1],tau_MF)))
23         v = qd_0 + qdd[x]*0.02
24         tau_D = 0.4*(qt.qd[x]-v)
25         qd = np.vstack((qd, v))
26
27         theta = q_0 + qd[x]*0.02
28         tau_P = 2*(qt.q[x]-theta)
29         q = np.vstack((q, theta))
30
31         qdd_0 = qdd[x]

```



```
32     qd_0 = qd[x]
33     q_0 = q[x]
34
35
36
37 rtb.tools.trajectory.qplot(q, block=False)
```

Código-fonte 5 – Cálculo da Posição do Braço a partir da RNE em Malha Fechada

```

1 l1 = 0.255;
2 lc1 = 0.122;
3
4 l2 = 0.199;
5 lc2 = 0.099;
6
7 m1 = 0.07576;
8 m2 = 0.1923;
9
10 i1 = 0.00175608;
11 i2 = 0.0174636;
12 g = 9.8;
13
14 m11 = m1*lc1^2 +m2*(l1^2 +lc2^2 + 2*l1*lc2*cos(theta2)) +
      i1 + i2;
15 m12 = m2*l1*lc2*cos(theta2)+m2*lc2^2+i2;
16 m21 = m12;
17 for i= 1:1:251
18 m22(i,1) = m2*lc2^2 + i2;
19 end
20
21 G11 = (m1*lc1*cos(theta1)+m2*(l1*cos(theta1)+lc2*cos(theta1
      +theta2)))*g;
22 G21 = (m2*lc2*cos(theta1+theta2))*g;
23 for i = 1:1:251
24 C11(i,1) = -m2*l1*lc2*sin(theta2(i))*v2(i)^2 - 2*m2*l1*lc2*
      sin(theta2(i))*v1(i)*v2(i);
25 end
26
27 for i = 1:1:251
28 C21(i,1) = m2*l1*lc2*sin(theta2(i))*v1(i)^2;
29 end

```

```
30
31 for i=1:1:251
32     M11(i,1) = m11(i)*accel1(i) + m12(i)*accel2(i);
33 end
34
35 for i=1:1:251
36     M12(i,1) = m21(i)*accel1(i) + m22(i)*accel2(i);
37 end
38
39 for i=1:1:251
40     tau11(i,1) = M11(i) + C11(i) + G11(i);
41 end
42
43 for i=1:1:251
44     tau12(i,1) = M12(i) + C21(i) + G21(i);
45 end
46
47 plot(tau11)
48 figure(2)
49 plot(tau12)
```

Código-fonte 6 – Cálculo da Posição do Braço a partir das Equações Dinâmicas em Malha Aberta