

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE MELHORIAS EM UM SENSOR  
DE FORÇA PARA FEEDBACK HÁPTICO**

Vitor Hugo Prado Gomes

Brasília, dezembro de 2019



**ENGENHARIA  
MECATRÔNICA**  
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

IMPLEMENTAÇÃO DE MELHORIAS EM UM SENSOR  
DE FORÇA PARA FEEDBACK HÁPTICO

Vitor Hugo Prado Gomes

*Relatório submetido como requisito parcial de obtenção  
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB  
*Orientador*

\_\_\_\_\_

Profa. Mariana Costa Bernardes Matias,  
FGA/UnB  
*Coorientadora*

\_\_\_\_\_

Prof. João Yoshiyuki Ishihara, ENE/UnB  
*Examinador interno*

\_\_\_\_\_

Profa. Carla Maria Chagas e Cavalcante Koike,  
CIC/UnB  
*Examinador Interno*

\_\_\_\_\_

Brasília, dezembro de 2019

## FICHA CATALOGRÁFICA

VITOR HUGO, PRADO GOMES

Implementação de melhorias em um sensor de força para feedback háptico.

[Distrito Federal] 2019.

ix, 43p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2019). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Sensor de Força

2. Interface Háptica

3. Robótica Cooperativa

4. Indústria 4.0

I. Mecatrônica/FT/UnB II. Implementação de melhorias em um sensor de força para feedback háptico.(n°012)

## REFERÊNCIA BIBLIOGRÁFICA

GOMES, VITOR HUGO PRADO, (2019). Implementação de melhorias em um sensor de força para feedback háptico. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-n°012, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 52p.

## CESSÃO DE DIREITOS

AUTOR: Vitor Hugo Prado Gomes

TÍTULO DO TRABALHO DE GRADUAÇÃO: Implementação de melhorias em um sensor de força para feedback háptico.

GRAU: Engenheiro

ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

---

Vitor Hugo Prado Gomes

SHIGS 706 Bloco Q Casa 55.

70350-767 Brasília – DF – Brasil.

---

## RESUMO

O LARA possui um projeto de desenvolvimento de uma célula de indústria 4.0. Um dos robôs utilizados é controlado por uma interface háptica, porém o *feedback* háptico foi pouco trabalhado e não está adequado. Esse trabalho possui como objetivo fazer a melhoria do *feedback* háptico fazendo a calibração do sensor de força e torque, implementação de filtros digitais e a compensação da força peso do efetuator terminal.

Palavras Chave: Indústria 4.0, Robótica cooperativa, Interface háptica, Sensor de Força, Filtros Digitais

---

## ABSTRACT

LARA has an industry 4.0 cell development project. One of the robots used is controlled by a haptic interface, but the force feedback hasn't been well developed. This work aims to improve the force feedback with sensor calibration, digital filters and a weight compensation of the end effector.

Keywords: Industry 4.0, Cooperative robotics, Haptic interface, Force sensor, Digital filters

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO	3
1.4	RESULTADOS OBTIDOS	3
<b>2</b>	<b>Fundamentos</b>	<b>4</b>
2.1	FUNDAMENTAÇÃO TEÓRICA	4
2.1.1	SINAIS	4
2.1.2	FILTROS	6
2.2	FERRAMENTAS UTILIZADAS	11
2.2.1	ROBÓTICA COMPLACENTE E MEKA A2	11
2.2.2	INTERFACE HÁPTICA OMEGA.7	12
2.2.3	SENSOR DE FORÇA E TORQUE (FORCE TORQUE SENSOR - FTS)	13
<b>3</b>	<b>Procedimentos</b>	<b>16</b>
3.1	CALIBRAÇÃO DO SENSOR	16
3.2	FILTROS	18
3.2.1	FILTRO BUTTERWORTH	18
3.2.2	FILTRO DE KALMAN ADAPTATIVO	19
3.3	ROTAÇÃO DAS FORÇAS	19
3.4	COMPENSAÇÃO DO PESO DA GARRA	22
<b>4</b>	<b>Resultados</b>	<b>25</b>
4.1	CALIBRAÇÃO	25
4.2	FILTRO BUTTERWORTH	27
4.3	FILTRO DE KALMAN ADAPTATIVO	30
4.4	ROTAÇÃO DE FORÇAS E COMPENSAÇÃO DA GRAVIDADE	34
<b>5</b>	<b>Conclusões</b>	<b>40</b>
5.1	PERSPECTIVAS FUTURAS	41
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>42</b>

# LISTA DE FIGURAS

2.1	Sinal analógico, discreto no tempo e digital .....	5
2.2	Braço robótico Meka A2. (Fonte: <a href="https://www.dndkm.org">https://www.dndkm.org</a> ).....	12
2.3	Interface Háptica Omega.7. (Fonte: <a href="http://www.forcedimension.com">http://www.forcedimension.com</a> ).....	12
2.4	Sensor de Força ATI Mini-40 (Fonte: <a href="https://www.ati-ia.com">https://www.ati-ia.com</a> ) .....	13
2.5	Esquemático eletrônico do sensor ATI Mini-40 (Fonte: <a href="https://www.ati-ia.com">https://www.ati-ia.com</a> ).....	14
2.6	Esquemático do <i>software</i> do sensor ATI Mini-40 (Fonte: <a href="https://www.ati-ia.com">https://www.ati-ia.com</a> ) ...	15
3.1	Processamento dos dados do sensor realizados para melhoria do <i>feedback</i> háptico. ...	16
3.2	Posição de calibração do robô. ....	17
3.3	Sistema de coordenadas do efetuator terminal e do sensor. ....	20
3.4	Modelo cinemático do braço robótico Meka [1]. ....	21
3.5	Sistema de coordenadas da base do robô e da interface háptica.....	21
3.6	Posição de descanso do robô. ....	23
4.1	Curva de calibração gerada pelo API.....	25
4.2	Dados do sensor na posição de descanso após a montagem da garra no sensor de força	26
4.3	Posição de teste do robô. ....	27
4.4	Resultado do teste dos três filtros de Butterworth. ....	28
4.5	Ampliação da Figura 4.4 em duas regiões disntintas. ....	28
4.6	Filtro Butterworth implementado em C em no MATLAB. ....	30
4.7	Comparação da força no eixo x para os filtros Kalman para $\alpha = 0.1$ e $\alpha = 0.9$ .....	31
4.8	Comparação do torque no eixo z para os filtros de Kalman para $\alpha = 0.1$ e $\alpha = 0.9$ ..	32
4.9	Comparação do FKA e um filtro Butterworth .....	33
4.10	Comparação entre o FKA implementado em MATLAB e em C.....	34
4.11	Etapas realizadas para o <i>feedback</i> Háptico. ....	34
4.12	Rotação e compensação de forças.....	35
4.13	Teste de forças para posição de teste com aplicação de força externa.....	37
4.14	Teste de forças para posição de teste com movimentação do robô. ....	38

# LISTA DE TABELAS

2.1	Características da interface háptica Omega.7. ....	13
2.2	Características do sensor Mini-40 SI-80-4. ....	14
4.1	Comparação do atraso de cada filtro. ....	29



## Siglas

LARA	<i>Laboratório de Automação e Robótica</i>
FIR	<i>Finite Impulse Response</i>
IIR	<i>Infinite Impulse Response</i>
ARMA	<i>Autoregressive Moving Average</i>
ROS	<i>Robot Operating System</i>
SDK	<i>Software Development Kit</i>
LED	<i>Light-Emitting Diode</i>

# Capítulo 1

## Introdução

*Este capítulo fala sobre a motivação da indústria 4.0, bem como as tecnologias que impulsionam essa indústria e a diferença entre alguns paradigmas entre essa nova indústria e a indústria comum.*

### 1.1 Contextualização

No últimos anos, a indústria 4.0 vem atraindo atenção de grandes fabricantes e prestadores de serviços. Porém, ainda não existe uma definição clara de indústria 4.0 e também não existe uma única tecnologia inovadora que está iniciando a transformação da indústria para a indústria 4.0. A indústria 4.0 caracteriza-se pela integração de novas tecnologias e paradigmas da cadeia de suprimento, sistemas de serviços e instalações de produção junto com a utilização de diversas tecnologias para fazer essa integração. [2]

A transformação da indústria para a indústria 4.0 está sendo guiada pelo avanço tecnológico de 9 tecnologias fundamentais: análise de *big data*, robôs autônomos, simulação, integração de sistemas horizontais e verticais, internet das coisas industriais, cibersegurança, armazenamento em nuvem, manufatura aditiva e realidade aumentada. Muitas dessas tecnologias já estão sendo utilizadas para manufatura, mas na indústria 4.0 elas irão transformar a forma de produção. Células isoladas e otimizadas serão totalmente integradas, automatizadas e otimizadas para produção, que será mais eficiente e mudará a relação entre fornecedores, produtores e consumidores, bem como a relação homem-máquina. [3]

Grandes empresas da Europa, Estados Unidos e Ásia estão já adotando os elementos da indústria 4.0, devido ao impacto positivo que eles geram. Esse impacto foi quantizado por Russman [3] para demonstrar a influência da indústria 4.0 na Alemanha:

- **Produtividade:** Nos próximos 5 a 10 anos, a indústria 4.0 será adotada por mais empresas da Alemanha, aumentando a produtividade do setor em 90 a 150 bilhões de euros.
- **Crescimento de Receita:** A demanda de equipamentos melhores e aplicações com *big data* dos fabricantes e a demanda de produtos customizados dos consumidores vai aumentar

a receita em aproximadamente 30 bilhões de euros por ano na Alemanha.

- Emprego: O crescimento estimulado pela indústria 4.0 aumentará o taxa de emprego em 6% nos próximos 10 anos. Trabalhos repetitivos e vão ser substituídos por desenvolvimento de software, tecnologia da informação e especialistas em mecatrônica.
- Investimento: Para conseguir incorporar os elementos da indústria 4.0, os produtores alemães devem investir aproximadamente 250 bilhões de euros nos próximos 10 anos.

Dentre as tecnologias fundamentais da indústria 4.0, os robôs já estão sendo utilizados para fazerem tarefas complexas, mas estão evoluindo ainda mais. Com a indústria 4.0, os robôs estão se tornando mais autônomos, flexíveis e cooperativos. Eventualmente os robôs irão cooperar entre si e com os humanos para executar tarefas, o que dará uma maior flexibilidade para as células de produção. [3]

Em uma indústria onde o homem trabalha ao lado de um manipulador industrial, a segurança do trabalhador no ambiente de trabalho deve ser priorizada. Os manipuladores industriais comuns são construídos para atingirem velocidades e precisão elevadas e por isso não são adequados para trabalhar com um ser humano. Nesse contexto surgem os manipuladores industriais complacentes, que são produzidos para diminuir o dano causado por colisões. Esses robôs devem possuir algum tipo de complacência, que é a capacidade do robô reagir à forças de contato durante o movimento, e devem ser capazes de operar em velocidades menores.

Uma outra forma de interação humano robô é a utilização de uma interface háptica. Essa interface permite uma comunicação bilateral entre o homem e o robô, onde o operário faz o controle do robô por meio dessa interface e recebe de volta o *feedback* das forças e torques que o robô está realizando, que pode ser uma informação importante em algumas aplicações. Esse tipo de dispositivo é muito utilizado para a realização de telecirurgias, onde o médico obtém muitas informações utilizando o tato.

## 1.2 Definição do problema

Em uma situação onde o trabalhador opera o robô a partir de um controle remoto, o *feedback* de forças pode ser muito importante. Para garantir um bom *feedback* háptico, é necessário que os dados utilizados para esse propósito sejam processados. O robô Meka do LARA possui um sensor de força no punho, e seus dados são utilizados para implementar o *feedback* para a interface háptica. Porém, os dados adquiridos por esse sensor são ruidosos e estão inadequados para fazer um bom *feedback* de forças. Este trabalho propõe um método de processamento desses dados para serem utilizados para o *feedback* de forças.

### 1.3 Objetivos do projeto

Aperfeiçoar o *feedback* de força da interface háptica por meio do processamento dos dados obtidos pelo sensor de força, fazendo a calibração do sensor de força, a implementação de filtros digitais para a diminuição do ruído dos dados do sensor e a compensação estática da força de gravidade do efetuador terminal.

### 1.4 Resultados obtidos

A calibração do sensor altera o zero de referência e a curva de calibração e os resultados mostraram que eles foram calibrados corretamente. Porém, a metodologia adota para calibrar gerou erro na medição das forças nos eixos x e y causados pela força medida no eixo z.

Os dois filtros digitais propostos foram implementados e testados no MATLAB antes de serem implementados para rodar no código C que faz a filtragem dos dados do sensor online. O primeiro deles teve bons resultados na implementação em MATLAB e na implementação em C. O outro só obteve resultados bons na implementação em MATLAB.

O modelo de compensação estática da força de gravidade proposto mostrou ser melhor que o antigo, conseguindo identificar corretamente a força peso do efetuador terminal, o erro estático de calibração e as forças externas.

A proposta deste trabalho foi cumprida, mostrando resultados satisfatórios em todas as etapas do projeto.

# Capítulo 2

## Fundamentos

*Este capítulo fala sobre a fundamentação teórica para o desenvolvimento do trabalho e sobre os equipamentos utilizados.*

### 2.1 Fundamentação Teórica

#### 2.1.1 Sinais

Um sinal é uma função que contém informação do comportamento de um sistema ou atributos de um fenômeno. Os sinais podem ser naturais ou sintetizados. São exemplos de sinais a saída de um termopar, que contém informação de temperatura transduzido para tensão elétrica ou uma onda eletromagnética, que pode ter sido utilizada para transportar a informação de uma imagem. [4]

Sinais não são necessariamente elétricos, mas sinais elétricos são convenientes devido a facilidade de manipulá-los, como por exemplo, fazer a sintetização, transmissão, análise e modificação. Um microfone faz a transdução da pressão do ar gerado pela fala em tensão elétrica, onde ambos os sinais variam no tempo. O sinal elétrico gerado pelo microfone pode ser transportado por condutores elétricos, amplificado e convertido novamente para pressão de ar através de um microfone. Pode-se também utilizar um conversor analógico digital para converter a tensão elétrica em uma sequência de bits e utilizar um computador para fazer manipulações nesse sinal. [4]

Os sinais podem ser descritos como função de uma variável independente, como por exemplo o tempo. A partir de agora vamos utilizar o tempo para falar da variável independente. Os sinais podem ser classificados de diversas maneiras, de acordo com o critério de classificação:

- (i) Sinais contínuos no tempo e discretos no tempo;
- (ii) Sinais periódicos e não periódicos;
- (iii) Sinais de energia e potência;
- (iv) Sinais determinísticos e probabilísticos;
- (v) Sinais pares ou ímpares;

### 2.1.1.1 Sinais Analógicos ou Contínuos no tempo

Esses sinais são definidos continuamente para qualquer valor da variável independente, e compõem a maioria dos sinais encontrados na natureza.

### 2.1.1.2 Sinais Discretos no Tempo

Os sinais discretos no tempo são caracterizados por serem contínuos em amplitude e não contínuos no tempo. Eles são obtidos através da amostragem de um sinal analógico e o intervalo de tempo entre cada amostra é chamada de período de amostragem. Shannon demonstrou que um sinal analógico limitado em banda pode ser completamente recuperado se a frequência de amostragem for maior que duas vezes a frequência máxima do sinal.

### 2.1.1.3 Sinais Digitais

Os sinais digitais são caracterizados por serem discretos tanto em amplitude quanto no tempo e são obtidos através da quantização em amplitude de um sinal discreto. Eles são muito importantes pois são os sinais que podem ser representados por computadores, uma vez que o computador só consegue representar os números com uma certa precisão, que é definida pela quantidade de bits da memória do computador. Diferentemente dos sinais discretos no tempo, os sinais digitais perdem informação quando passam pelo processo de quantização, e essa perda de informação, que está ligada à quantidade de bits do computador, pode ou não ser importante para a aplicação.

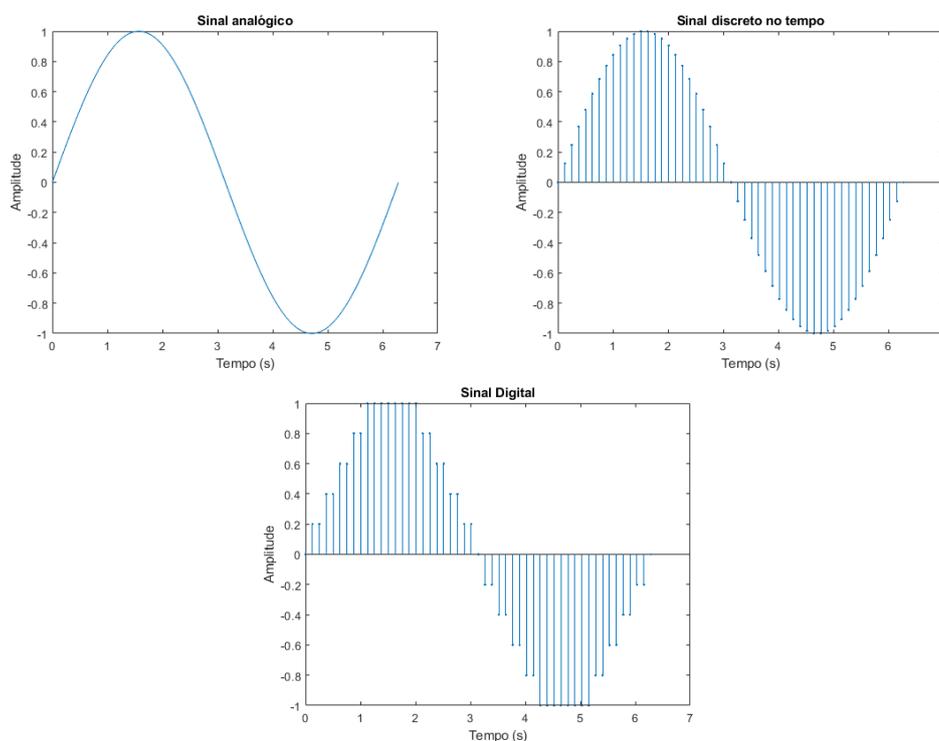


Figura 2.1: Sinal analógico, discreto no tempo e digital

#### 2.1.1.4 Processamento de Sinais

O processamento de sinais consiste na análise e/ou modificação de um sinal para a geração de outro sinal. Isso pode ser desde uma amplificação na amplitude do sinal gerado por um termopar para possibilitar uma melhor leitura até a filtragem de um sinal para diminuir o ruído ou a compressão do sinal para transmissão. O processamento de sinais é uma área que vem crescendo bastante devido às inúmeras aplicações, tanto em sinais analógicos ou sinais digitais, como as citadas anteriormente ou como no caso deste trabalho, que consiste na atenuação dos ruídos do sensor de força e a compensação do peso do efetuador terminal.

#### 2.1.2 Filtros

Existem dois tipos de filtros digitais e lineares invariantes no tempo: FIR (Finite Impulse Response) e IIR (Infinite Impulse Response). Ambos são caracterizados pela sua resposta à um impulso de entrada. Esses tipos de filtro possuem vantagens e desvantagens distintas, o que leva a escolha deles uma questão da aplicação.

##### 2.1.2.1 Filtros FIR

Os filtros FIR são caracterizados pela sua resposta em regime permanente ser 0 para uma entrada impulso ou uma entrada com duração finita. Suas principais características são:

1. Podem atingir um resposta linear em frequência de fase.
2. Nunca são instáveis;
3. São menos sensíveis à aproximações e precisão finita;
4. As metodologias de projeto são lineares;
5. Podem ser realizados em um hardware genérico ou dedicado;

A saída de um filtro FIR é a combinação linear da entrada atual e das  $M$  entradas passadas. Por causa de seus métodos de implementação, esses filtros também são chamados de filtros não recursivos, filtros de convolução ou filtros de média móvel. Pela definição desses filtros, a sua função de transferência é polinomial e a saída desses filtros pode ser escrita como [5]:

$$y(n) = b_0(n) + b_1(n - 1) + \dots + b_m(n - M) \quad (2.1)$$

Essa equação também pode ser escrita na forma compacta utilizando somatório:

$$y(n) = \sum_{r=0}^M x(n - r)b(r) \quad (2.2)$$

A equação do filtro FIR é a mesma da definição de convolução, diferenciado pelo limite finito M. A função de transferência pode ser obtida utilizando a transformada z:

$$H(z) = \sum_{r=0}^M b_r z^{-r} \quad (2.3)$$

### 2.1.2.2 Filtros IIR

A saída de um filtro IIR é uma combinação linear da entrada atual, das M entradas passadas e das N saídas passadas [5]:

$$\sum_{i=0}^N a_i y(n-i) = \sum_{r=0}^M b_r x(n-r) \quad (2.4)$$

Onde n é inteiro, x(n) a sequência de entrada e y(n) a sequência de saída. A diferença de um filtro FIR e IIR é a adição do somatório das saídas anteriores para calcular a saída atual. Esse termo recursivo faz a resposta a um impulso ou entrada finita ter duração infinita. Essa estrutura, que é chamada de equação de diferença, é similar à equação diferencial, que descreve o comportamento da saída e entrada em um sistema contínuo no tempo. O cálculo da saída desse filtro pode ser escrito como:

$$y(n) = \sum_{r=0}^M b_r x(n-r) - \sum_{i=1}^N a_i y(n-i) \quad (2.5)$$

Devido ao *feedback* necessário para a implementação dos filtros IIR, eles também são chamados de filtros recursivos ou filtro de média móvel auto-regressiva (ARMA - autoregressive moving average). Da mesma forma que acontece com o filtro FIR, a função de transferência pode ser obtida utilizando a transformada z:

$$\begin{aligned} \sum_{i=0}^N a_i Y(z) z^{-i} &= \sum_{r=0}^M b_r X(z) z^{-r} \\ H(z) = \frac{Y(z)}{X(z)} &= \frac{\sum_{r=0}^M b_r X z^{-r}}{\sum_{i=0}^N a_i Y z^{-i}} \end{aligned} \quad (2.6)$$

Esses filtros possuem uma função de transferência que é a razão de dois polinômios, que se difere dos filtros FIR, onde a função de transferência é polinomial. Por sua função de transferência ser um razão de dois polinômios, os filtros IIR são mais flexíveis, mas também são mais difíceis de serem implementados.

### 2.1.2.3 Filtro Butterworth

O filtro Butterworth é um tipo de filtro cuja resposta de magnitude é do tipo: [5]

$$|H(j\omega)|^2 = \frac{1}{1 + (\frac{\omega}{\omega_c})^{2N}} \quad (2.7)$$

Os filtros de Butterworth são ótimos em relação a parte plana da banda passante da resposta em frequência. Todas as derivadas da função de magnitude são zero em  $\omega = 0$  e todos os zeros do filtro são em  $\omega = \infty$ .

Para o projeto desse filtro e futuramente a conversão para um filtro digital, é importante determinar a função de transferência  $H(j\omega)$ .

$$H(P)H(-P) = \frac{1}{1 + P^N(-P)^N} \quad (2.8)$$

Assim, a função característica pode ser escrita como:

$$K(P) = P^N \quad (2.9)$$

Os polos da função de transferência podem ser encontrados resolvendo a equação para P:

$$P^{2N} + (-1)^N = 0 \quad (2.10)$$

As  $2N$  raízes dessa equação possuem módulo 1 e estão igualmente espaçadas em volta do círculo unitário. Essas raízes podem ser calculadas analiticamente:

$$P_{\infty v} = j e^{j\pi(2v+1)/2N}, 0 \leq v \leq 2N - 1 \quad (2.11)$$

Metade das raízes vão estar no semiplano esquerdo e a outra metade no semiplano direito. A função de transferência do filtro é composta pelos polos no lado esquerdo do plano.

$$H(P) = \frac{1}{(P - P_{\infty 1})(P - P_{\infty 2}) \dots (P - P_{\infty N})} \quad (2.12)$$

O filtro Butterworth é totalmente especificado por dois parâmetros, a frequência de corte  $\omega_c$  e a ordem do filtro  $N$ . A conversão do filtro Butterworth analógico para digital é normalmente feito utilizando a transformada  $z$  bilateral, que coloca ele na forma descrita pela Equação (2.6).

#### 2.1.2.4 Filtro de Kalman

O filtro de Kalman é um estimador utilizado para estimar o estado de um sistema linear dinâmico que é perturbado por um ruído branco gaussiano, usando medidas que são funções lineares do estado, mas perturbadas pela adição de um ruído branco gaussiano [6].

Um sistema discreto no tempo descrito acima pode ser modelado no espaço de estados como:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (2.13)$$

$$y_k = Cx_k + v_k \quad (2.14)$$

Onde:

1. A = Modelo de transição dos estados;
2. B = Modelo das entradas de controle;
3. C = Modelo de observação;
4.  $w_k$  = Ruído do processo;
5.  $v_k$  = Ruído da observação;
6. P = Matriz de covariância do erro de estimativa;
7. Q = Matriz de covariância do ruído do processo;
8. R = Matriz de covariância do ruído da observação;

Para as próximas equações, o símbolo  $\hat{\phantom{x}}$  significa que é uma estimativa, ou seja,  $\hat{x}$  é uma estimativa de x. Os sobrescritos - e + significam, respectivamente, antes da correção e depois da correção. O algoritmo do filtro de Kalman segue os seguintes passos:

#### **Predição:**

Inicialmente é feito a predição do novo estado  $\hat{x}_k^-$  com base no estado estimado anterior  $\hat{x}_{k-1}^+$  e na entrada de controle  $u_{k-1}$ , com suas respectivas matrizes A e B.

$$\hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1} \quad (2.15)$$

Também é feita a predição da nova matriz de covariância do erro  $P_k^-$  de acordo com a matriz de covariância do erro estimado anterior  $P_{k-1}^+$ , a matriz de covariância do ruído do processo Q e a matriz do modelo de transição dos estados A.

$$P_k^- = AP_{k-1}^+A^T + Q \quad (2.16)$$

#### **Correção:**

Na correção calcula-se a inovação  $d_k$  fazendo a diferença entre a medida real  $y_k$  e a medida estimada, que é calculada utilizando o estado estimado,  $C\hat{x}_k^-$ .

$$d_k = y_k - C\hat{x}_k^- \quad (2.17)$$

Em seguida é calculado o ganho de Kalman  $K_k$

$$K_k = P_k^- C^T (R + C P_k^- C^T)^{-1} \quad (2.18)$$

É feito então a correção dos valores preditos  $\hat{x}_k^-$  e  $P_k^-$  de acordo com a inovação  $d_k$  e o ganho de Kalman  $K_k$ .

$$\hat{x}_k^+ = \hat{x}_k^- + K_k d_k \quad (2.19)$$

$$P_k^+ = (I - K_k C) P_k^- \quad (2.20)$$

### 2.1.2.5 Filtro de Kalman Adaptativo

O filtro de Kalman comum assume que as matrizes Q e R são conhecidas, e elas são utilizadas para calcular o ganho de Kalman  $K_k$  e a matriz de covariância do erro  $P_k^-$ . No entanto, na maioria das situações reais pode-se não ter as matrizes Q e R.

Por causa disso surgiu o filtro de Kalman adaptativo, que calcula as matrizes Q e R de acordo com as medições feitas. Ele é utilizado quando se conhece o modelo do processo, mas não se conhece a covariância dos ruídos de entrada (Q) e medida (R), ou elas variam com o tempo [7].

Existem diferentes métodos estatísticos para fazer a adaptação dos parâmetros Q e R. Em [8], Shahrokh Akhlaghi e Ning Zhou propõem o cálculo de um novo termo chamado *residual*  $E_k$ , que é a diferença entre a medida  $y_k$  e o valor estimado  $C\hat{x}_k^+$ . O valor de  $R_k$  é adaptado utilizando esse novo termo *residual*, e o valor de  $Q_k$  é adaptado utilizando o resíduo da medição  $d_k$ . Em [9], é proposta a adaptação de  $Q_k$  e  $R_k$  com base em cálculos estatísticos da variância e do valor esperado. Neste trabalho foi utilizado a proposta de [8]. Assim, são adicionados três novas equações para calcular no algoritmo do filtro de Kalman:

$$E_k = [y_k - C\hat{x}_k^+] \quad (2.21)$$

$$R_k = \alpha R_{k-1} + (1 - \alpha)(E_k E_k^T + C P_k^- C^T) \quad (2.22)$$

$$Q_k = \alpha Q_{k-1} + (1 - \alpha)(K_k d_k d_k^T K_k^T) \quad (2.23)$$

em que  $\alpha$  é o fator de esquecimento e  $E_k$  é chamado de resíduo. O termo  $\alpha$  representa a confiabilidade nos valores anteriores de Q e R e nos valores calculados na iteração. Para  $\alpha$  igual a 1 não existe adaptação nos valores de Q e R e para  $\alpha$  igual 0 a adaptação é máxima.

## 2.2 Ferramentas utilizadas

### 2.2.1 Robótica Complacente e Meka A2

De acordo com a ISO 8373:2012, um robô industrial é um manipulador polivalente, reprogramável, programável em 3 ou mais eixos, que pode ser fixo ou móvel e é utilizado para aplicações de automação industrial.

Na indústria comum, a precisão e a velocidade do manipulador industrial são muito mais importantes que a segurança do ser humano, uma vez que ela é projetada para que não haja interferência humana durante o funcionamento. Em um ambiente onde se pode ter um ser humano e um manipulador industrial trabalhando ao mesmo tempo, como na indústria 4.0, a segurança do ser humano se torna prioridade. Para garantir essa segurança, surgiram vários tipos de atuadores e elos flexíveis e o conceito de robótica complacente. [10]

O termo complacência é utilizado para descrever uma grande variedade de tarefas como: abrir uma porta, colocar um objeto na mesa, segurar um objeto ou manipular ovos. O comum de todas essas tarefas é que a trajetória e a força são modificadas de acordo com as forças de contato durante o movimento. A complacência pode ocorrer devido à reação ao estímulo ou através da complacência passiva do manipulador. [11]

Define-se como complacência em robótica a habilidade do manipulador reagir a forças de contato durante o movimento. Sendo assim, um robô complacente é capaz de detectar colisão com objetos ou seres humanos e pode ter formas para diminuir o impacto causado. Existem duas formas de garantir que o manipulador seja seguro para um ser humano: a complacência passiva e a complacência ativa.

Na complacência ativa, a colisão do manipulador com o ser humano é detectada por diversos sensores e a rigidez do robô é controlado via software. Porém, a complacência ativa possui o problema de ter uma baixa velocidade de resposta, uma vez que envolve a detecção e atuação em uma resposta dinâmica à colisão. Além disso, a instalação de sensores e atuadores para conseguir diminuir o problema citado acaba aumentando os custos de fabricação, aumentando o peso e o tamanho do atuador e aumenta a possibilidade de uma falha nos atuadores. [10]

A complacência passiva é baseada em atuadores com componentes como mola ou amortecedores, que absorvem parte da força do impacto. Esse tipo de complacência não utiliza sensores para detectar colisão e por isso consegue obter respostas rápidas para colisão dinâmica. Porém, a utilização desses componentes com mola ou amortecedores para a complacência passiva gera uma perda de precisão e velocidade de atuação.[10]

O braço robótico utilizado nesse trabalho é o Meka A2 Arm. Ele é um braço robótico redundante com 7 graus de liberdade com atuadores complacentes SEA (Series Elastic Actuators). Ele possui as mesmas juntas e o tamanho parecido a um braço humano e pesa aproximadamente 7kg. Possui o modelo de cinemática inversa e dinâmica inversa (apenas modelo de massa) e possui controle por torque, posição e velocidade. Sua precisão é menor que 1 cm e a carga máxima é menor que 2Kg [12].

A Figura 2.2 mostra uma imagem do braço robótico Meka A2. Diferentemente da imagem, o braço disponível no LARA está equipado com uma garra no efetuador terminal.

### 2.2.2 Interface Háptica Omega.7

Uma interface háptica é um dispositivo que permite um ser humano tocar e manipular objetos em um ambiente real ou virtual a partir de um sistema tele operado. A interface háptica fornece uma interação bilateral entre o usuário e o ambiente em que ele está interagindo, ou seja, além de ser uma interface de entrada para controle de robôs ou manipuladores, ela fornece algum tipo de *feedback* para o usuário, podendo ser a força que o manipulador está realizando. [13]

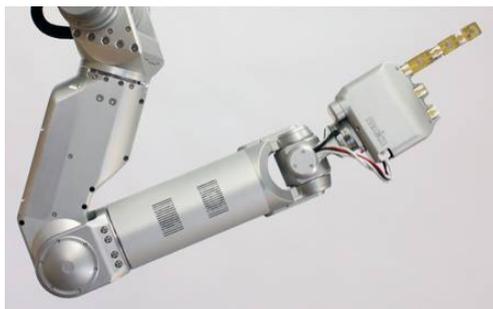


Figura 2.2: Braço robótico Meka A2. (Fonte: <https://www.dndkm.org>)

A interface háptica utilizada nesse trabalho é a Omega.7, da empresa Force Dimension. Ela é possivelmente a interface háptica com 7 graus de liberdade mais avançada para utilização *desktop*. O (Software Development Kit) SDK da interface háptica permite que a leitura das posições e a programação das forças desejadas sejam feitas facilmente. Ela é utilizada para aplicações aeroespaciais e médicas, que demonstra a sua funcionalidade para aplicações em que a segurança é crítica. A Tabela 2.2.2 mostra suas principais características físicas de funcionamento.



Figura 2.3: Interface Háptica Omega.7. (Fonte: <http://www.forcedimension.com>)

A interface háptica é utilizada para controlar o efetuador terminal do Meka A2. Os dados dos sensores dessa interface são obtidos e enviados para o Meka. Esses dados são processados e utilizados para fazer a cinemática inversa para movimentar o efetuador terminal para a posição desejada. Um sensor acoplado entre o braço robótico e o efetuador terminal faz a medição das

forças e torques e envia para a interface háptica para fazer o *feedback* de força. É importante ressaltar que a aquisição de dados desse sensor impacta diretamente o *feedback* de força que a interface háptica provê ao usuário.

Tabela 2.1: Características da interface háptica Omega.7.

Área de trabalho	Translação	Cilíndrica 160 x 110 mm
	Rotação	240 x 240 x 180°
	Garra	25 mm
Força	Translação	12.0 N
	Garra	± 8.0 N
Resolução	Translação	< 0.01 mm
	Rotação	0.09°
	Garra	0.006 mm
Rigidez	Malha Fechada	14.5 N/mm

### 2.2.3 Sensor de Força e Torque (Force Torque Sensor - FTS)

De acordo com a ISO 29182-2:2013, um sensor é um dispositivo que observa e mede uma fenômeno natural ou um processo humano e converte a medida para um sinal, que pode ser elétrico, químico, etc.

O manipulador Meka A2 possui um sensor de força e torque ATI Mini-40 SI-80-4, que é acoplado entre o efetuator terminal e o braço robótico. Ele possui 6 graus de liberdade, que possibilita a detecção de força e torque nos eixos x, y e z.

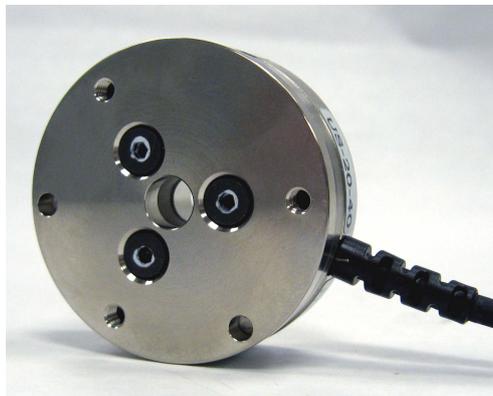


Figura 2.4: Sensor de Força ATI Mini-40 (Fonte: <https://www.ati-ia.com>)

A Figura 2.5 mostra o esquemático do *hardware* eletrônico do sensor. O *hardware* é alimentado por uma fonte de energia externa AC que deve ser chaveada para 110V ou 220V e possui um LED indicando quando está energizado. A medição das forças e dos torques é feita por 6 *strain gauges bridges*. O sinal obtido dos *strain gauges* é processado por um filtro passa baixas e um multiplexador e em seguida é enviado para um conversor analógico-digital, que vai para a CPU. O sistema possui memória RAM e memória não volátil para a matriz de calibração e variável

Tabela 2.2: Características do sensor Mini-40 SI-80-4.

	F <sub>x</sub> ,F <sub>y</sub>	F <sub>z</sub>	T <sub>x</sub> ,T <sub>y</sub>	T <sub>z</sub>
Intervalo de Medição	80 N	240 N	4 Nm	4 Nm
Resolução	1/50 N	1/25 N	1/2000 Nm	1/2000Nm
Carga Máxima	±840 N	±2400 N	±19 Nm	±20 Nm
Rigidez	$1,1 \times 10^7$ N/m	$2 \times 10^7$ N/m	$2,8 \times 10^3$ Nm/rad	$4 \times 10^3$ Nm/rad

de monitoramento. O sistema possui duas entradas de dados e três saídas de dados, sendo uma dessas saídas analógica.

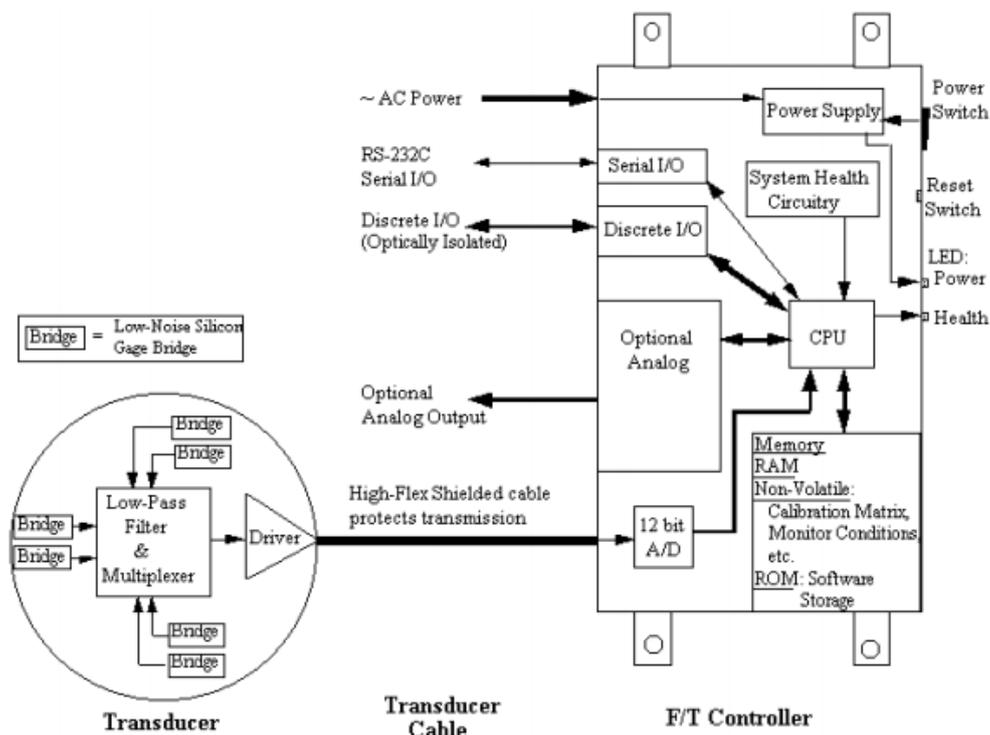


Figura 2.5: Esquemático eletrônico do sensor ATI Mini-40 (Fonte: <https://www.ati-ia.com>)

A Figura 2.6 mostra o *software* da CPU. Inicialmente é feita a leitura dos *strain gauges*. Em seguida é feita a verificação da saturação dos valores lidos. Então é feita uma subtração por um vetor de Bias e em seguida é feita a multiplicação pela matriz de calibração, que gera os dados finais. É feita então a verificação da variável de monitoramento de condição do sistema. Caso ela seja *True*, liga-se a entrada e saída discreta. A saída para o usuário é escolhida para ser em formato ASCII, binário ou analógico.

Além disso, o sistema pode possuir múltiplas calibrações, cada calibração podendo possuir intervalos de medição diferentes. Essas calibrações são feitas com diferentes valores de carga, o que permite o usuário escolher a calibração que melhor encaixa para sua aplicação. Pode-se também fazer uma calibração manual do sensor, que será apresentada na seção seguinte.

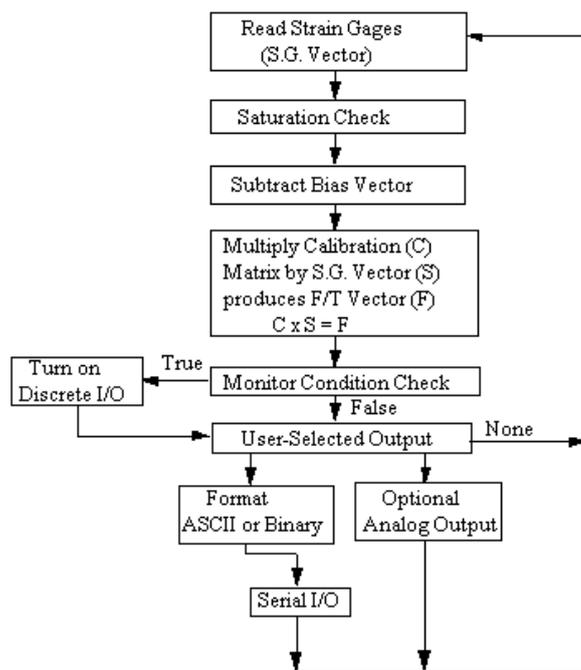


Figura 2.6: Esquemático do *software* do sensor ATI Mini-40 (Fonte: <https://www.ati-ia.com>)

# Capítulo 3

## Procedimentos

Para se fazer a melhoria do *feedback* háptico, inicialmente é feita a calibração do sensor de força para garantir que as forças medidas estão corretas. Depois, é feita uma mudança de escala dos dados do sensor para garantir que as forças passadas para interface háptica não excedam o valor máximo. Os dados então são filtrados por um filtro digital para fazer a atenuação do ruído. É feita a subtração do erros sistemático do sensor, seguida pela rotação das forças para colocá-las nas coordenadas da interface háptica. Por fim, é subtraído o peso da garra das forças medidas pelo sensor para extrair a força externa aplicada ao robô. A Figura 3.1 mostra as etapas realizadas nos processamentos dos dados do sensor.

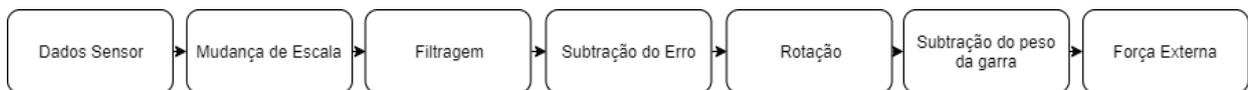


Figura 3.1: Processamento dos dados do sensor realizados para melhoria do *feedback* háptico.

Este capítulo detalha como foi feito cada uma das etapas dos processamentos de sinais citados.

### 3.1 Calibração do Sensor

O primeiro procedimento para melhorar os dados do sensor é fazer sua calibração, uma vez que a última calibração documentada foi feita por Rafael Koji [1], no final de 2017. Para isso, utilizou-se o API de calibração em python disponibilizado pela própria fabricante. É necessário o desacoplamento da garra para fazer a calibração.

Inicialmente, o programa pede para o usuário escolher qual componente do robô deseja-se calibrar, que neste caso é o sensor de força com o nome loadx6. Após isso, o usuário pode escolher entre 5 opções, sendo elas:

1. Display Sensors (ds): Mostra quais sensores estão funcionando.
2. Sensor Analyze (sa): Mostra a leitura dos dados de cada sensor.

3. Zero Load Cell (zl): Faz a calibração do zero do sensor.
4. Calibrate Load Cell (cl): Faz a calibração do zero do sensor e gera uma reta utilizando pesos de 500g, 1000g e 2000g no eixo -z.
5. Zero Load Cell with Payload (zp): Faz a calibração do zero do sensor sendo o peso da carga utilizada.

Fez-se a calibração do sensor utilizando a opção **cl**. Para isso, desmontou-se a garra do robô para poder colocar os pesos em cima do sensor. Além disso, movimentou-se o robô para uma orientação onde o eixo -z do sensor alinhasse com a força peso. Essa posição vai ser chamada de posição de calibração e possui os seguintes ângulos para as juntas do robô: (0,576, 0,0477, -0,0612, 1,726, -0,0155, 0,8383, 0,0564), conforme mostra a Figura 3.2.



Figura 3.2: Posição de calibração do robô.

Após isso, realizou-se os procedimentos conforme especificados pelo código, colocando peso de 0g, 500g 1000g e 2000g, respectivamente. Os pesos utilizados foram pesados em uma balança

digital e possuem  $500g \pm 1g$ ,  $1000g \pm 1g$  e  $2000g \pm 1g$ . O gráfico de calibração é gerado com os pontos coletados e a reta de interpolação desses pontos. Após esse gráfico ser gerado, o usuário deve escolher sobrescrever ou não a calibração atual.

Como o braço robótico Meka é complacente e suporta aproximadamente 2000g, os pesos de 1000g e 2000g mudavam a posição e orientação dele. Foi necessário fazer uma força externa para o robô ficar na posição desejada durante a calibração com os pesos. Por isso, esse procedimento foi feito várias vezes até achar o gráfico com a melhor interpolação entre os pontos.

## 3.2 Filtros

Após a calibração do sensor, é necessário diminuir os seus ruídos de medição. Para isso, utilizou-se filtros digitais. Foi feito um filtro Butterworth de segunda ordem e um filtro de Kalman adaptativo, para analisar qual atende melhor as especificações de filtragem e atraso.

### 3.2.1 Filtro Butterworth

Como os dados de força adquiridos pelo sensor vão ser utilizados para a *feedback* de força para uma interface háptica, esses dados não devem possuir ruídos grandes, mas também não podem possuir um atraso muito alto. Por isso, foram feitos 3 filtros de Butterworth passa baixa de segunda ordem, com frequências de corte diferentes. Uma frequência de corte alta implica em uma baixa filtragem de dados, mas uma resposta mais rápida à variações de força, enquanto uma frequência de corte baixa implica em uma alta filtragem de dados, mais uma resposta mais lenta a variações de força.

Inicialmente fez-se a aquisição dos dados do sensor em um experimento variando a pose do robô. Esses dados foram importados no MATLAB, onde se fez a implementação e os testes para 3 filtros Butterworth com frequências de corte diferentes. A implementação foi feita utilizando a função *butter*, que recebe como parâmetros a ordem do filtro desejado e a frequência de corte, e retorna o numerador e o denominador da função de transferência. Depois, utiliza a função *filter* passando como parâmetros o numerador e o denominador da função de transferência e os dados do sensor, que foram coletados anteriormente [14].

A função de transferência de um filtro Butterworth digital de segunda ordem é:

$$\frac{Y(z)}{U(z)} = \frac{\alpha_1 + \alpha_2 z^{-1} + \alpha_3 z^{-2}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}} \quad (3.1)$$

Onde os parâmetros  $\alpha_n$  e  $\beta_n$  variam de acordo com a frequência de corte dos filtros.

Após os testes feitos em MATLAB, fez-se a implementação do filtro no código da interface háptica, que deve filtrar os dados a medida que eles chegam. Para isso, utiliza a transformada Z inversa para achar a equação de diferença do filtro.

$$\begin{aligned}
Y(z)(1 + \beta_1 z^{-1} + \beta_2 z^{-2}) &= U(z)(\alpha_1 + \alpha_2 z^{-1} + \alpha_3 z^{-2}) \\
y(k) + \beta_1 y(k-1) + \beta_2 y(k-2) &= \alpha_1 u(k) + \alpha_2 u(k-1) + \alpha_3 u(k-2) \\
y(k) &= \alpha_1 u(k) + \alpha_2 u(k-1) + \alpha_3 u(k-2) - \beta_1 y(k-1) - \beta_2 y(k-2)
\end{aligned} \tag{3.2}$$

em que  $y(k)$  é o dado filtrado e  $u(k)$  é o dado do sensor. Para fazer a implementação dessa equação de diferença, deve-se guardar os valores da filtragem e do dado do sensor das últimas 2 iterações.

### 3.2.2 Filtro de Kalman Adaptativo

O filtro de Kalman adaptativo também foi projetado e implementado no MATLAB antes de ser implementado no código da interface háptica. Os testes do filtro de Kalman adaptativo foram feitos utilizando os mesmos dados do sensor dos testes do filtro Butterworth. A implementação no MATLAB utiliza as equações do filtro de Kalman (2.15) a (2.20) e as equações (2.21) a (2.23), referentes a adaptação dos valores de  $Q_k$  e  $R_k$ .

Definiu-se o vetor de estado  $x_k$  como as forças nos eixos x,y,z e os torques nos eixos x,y,z, ou seja, os dados dos sensores. O sensor foi modelado como um sistema onde o novo estado é o estado passado acrescentando um ruído, e a saída é o estado acrescentando um ruído. Assim, tem-se as matrizes do modelo no espaço de estado  $A = I_{6 \times 6}$ ,  $B = 0$ ,  $C = I_{6 \times 6}$ .

A primeira iteração do filtro faz a inicialização do estado  $x_k$  sendo o dado do sensor lido, da matriz de covariância do erro  $P$  sendo a covariância, da matriz de covariância do ruído do processo  $Q$  com valores aleatórios e da matriz de covariância do ruído do processo  $R$  com valores aleatórios.

Após os testes feitos com o filtro de Kalman no MATLAB, fez-se a implementação desse filtro no código da interface háptica. Essa implementação segue os mesmos procedimentos da implementação do MATLAB, e utiliza a biblioteca *Eigen* para fazer os cálculos matriciais.

## 3.3 Rotação das forças

Após a filtragem dos dados do sensor, o ruído já foi atenuado e os dados podem ser utilizados para fazer o *feedback* para interface háptica. Porém, como o eixo de coordenadas do sensor de força não é o mesmo eixo de coordenadas da interface háptica e ele rotaciona de acordo com a movimentação do robô, é necessário fazer a rotação das forças medidas pelo sensor de força.

Existem 4 sistemas de coordenadas importantes para fazer essa rotação completa, sendo eles o sistema de coordenadas do sensor  $S_s$ , o sistema de coordenadas do efetuado terminal  $S_e$ , o sistema de coordenadas da base do robô  $S_r$  e o sistema de coordenadas da interface háptica  $S_h$ . O sistema de coordenadas da interface háptica  $S_h$  é utilizado como referência, uma vez que ele é o final da rotação das forças e não sofre rotação. Além disso, o sistema de coordenadas  $S_r$  também é fixo. Os sistemas de coordenadas  $S_s$  e  $S_e$  rotacionam em relação aos outros dois sistemas de coordenadas

$S_h$  e  $S_r$ , mas eles não rotacionam entre si, uma vez que ambos o sensor e o efetuator terminal estão depois de todas as juntas do robô.

Para facilitar a visualização da transformação de  $S_s$  para  $S_h$ , ela foi dividida em três etapas. A primeira etapa faz a conversão de  $S_s$  para  $S_e$ , a segunda etapa faz a conversão de  $S_e$  para  $S_r$  e a última etapa faz a conversão de  $S_r$  para  $S_h$ .

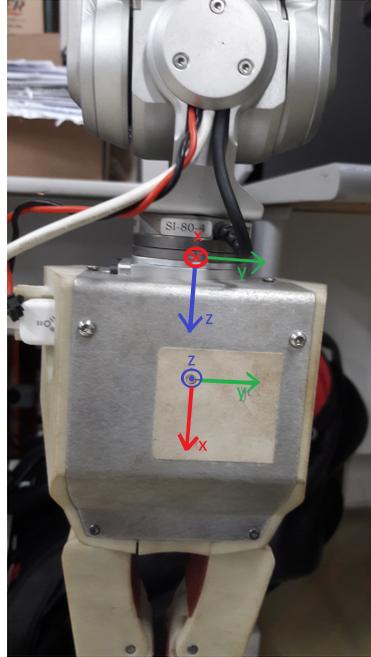


Figura 3.3: Sistema de coordenadas do efetuator terminal e do sensor.

A Figura 3.3 mostra os sistemas de coordenadas do sensor e do efetuator terminal e, como  $S_s$  e  $S_e$  nunca rotacionam entre si, a conversão é feita:

$$\begin{aligned}
 Fx_e &= Fz_s \\
 Fy_e &= Fy_s \\
 Fz_e &= -Fx_s
 \end{aligned}
 \tag{3.3}$$

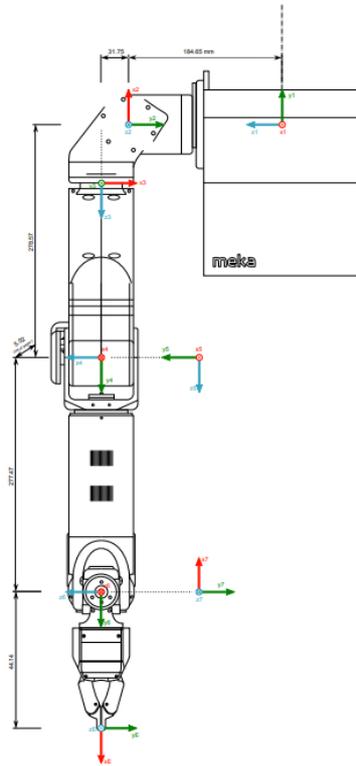


Figura 3.4: Modelo cinemático do braço robótico Meka [1].

Para fazer a conversão de  $S_e$  para  $S_r$ , utiliza-se o modelo da cinemática direta do robô, conforme mostra a Figura 3.4. Utilizou-se a biblioteca de quatérnio dual *dq\_robotics* para transformar a matriz de Denavit-Hatenberg do robô e os ângulos de cada junta em um quatérnio dual  $q$  referente a rotação do efetuador terminal e em seguida utiliza-se a equação de rotação por um quatérnio dual.

$$S_r = qS_eq^* \quad (3.4)$$

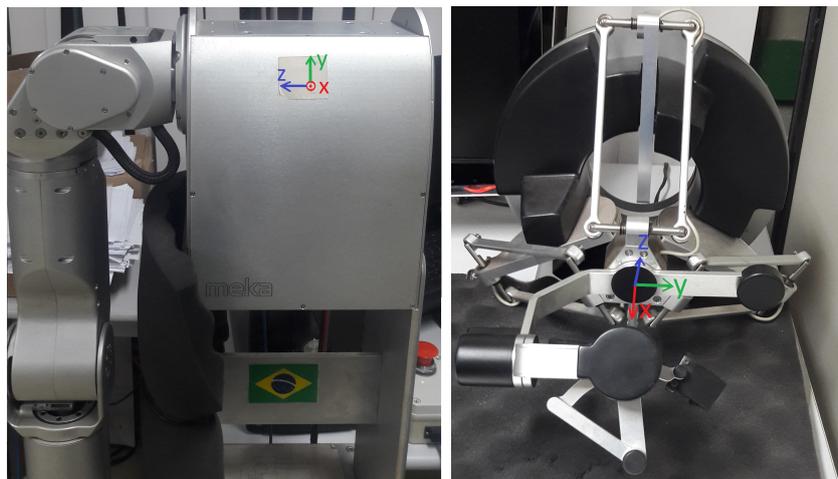


Figura 3.5: Sistema de coordenadas da base do robô e da interface háptica.

A Figura 3.5 mostra os sistemas de coordenadas da base do robô e da interface háptica. No *setup* utilizado, o robô e a interface háptica possuem uma rotação de 180 graus no eixo z, ou seja, ao fazer um movimento na direção y no sistema de coordenadas da interface háptica, o robô se movimenta na direção z no sistema de coordenadas da base do robô e ao fazer um movimento na direção x no sistema de coordenadas da interface háptica, o robô se movimenta na direção -x no sistema de coordenadas do robô. Assim, a conversão é feita:

$$\begin{aligned}Fx_h &= -Fx_r \\Fy_h &= Fz_r \\Fz_h &= Fy_r\end{aligned}\tag{3.5}$$

### 3.4 Compensação do peso da garra

Com a rotação de forças pronta, é necessário extrair a força externa dos dados do sensor. A leitura do sensor de força  $F_s$  em qualquer instante de tempo em que o robô está parado pode ser modelada como a soma da força externa  $F_E$ , força peso  $P$  e do erro  $E(x, y, z)$ .

$$F_s = F_E + P + E(x, y, z).\tag{3.6}$$

Para fazer a compensação estática do peso da garra fez-se o experimento sem a aplicação de nenhuma força externa e por isso considerou-se que  $F_E = 0$ , ou seja, o sensor estava submetido apenas a força  $P$  e o erro  $E(x, y, z)$ .

Porém, quando o robô se move e a orientação do sistema de coordenada do sensor rotaciona, o erro  $E(x, y, z)$  rotaciona junto e faz efeito sobre os mesmos eixos. A força peso  $P$  continua tendo a mesma orientação sobre a referência global, o que faz o sensor medir essa força por um outro eixo ou a combinação dos eixos. É necessário então determinar o erro  $E(x, y, z)$  e a força  $P$  separadamente.

$$F_s = P + E(x, y, z)\tag{3.7}$$

Decidiu-se fazer um procedimento para determinar o  $E(x, y, z)$  sem ter que desacoplar a garra, uma vez que ela pode influenciar em seu valor. Para isso, mediu-se os dados do sensor quando o robô está na posição de descanso, onde os ângulos das juntas são  $(0,0,0,0,0,0)$  radianos, a força peso age somente sobre o eixo z, conforme mostra a Figura 3.6.

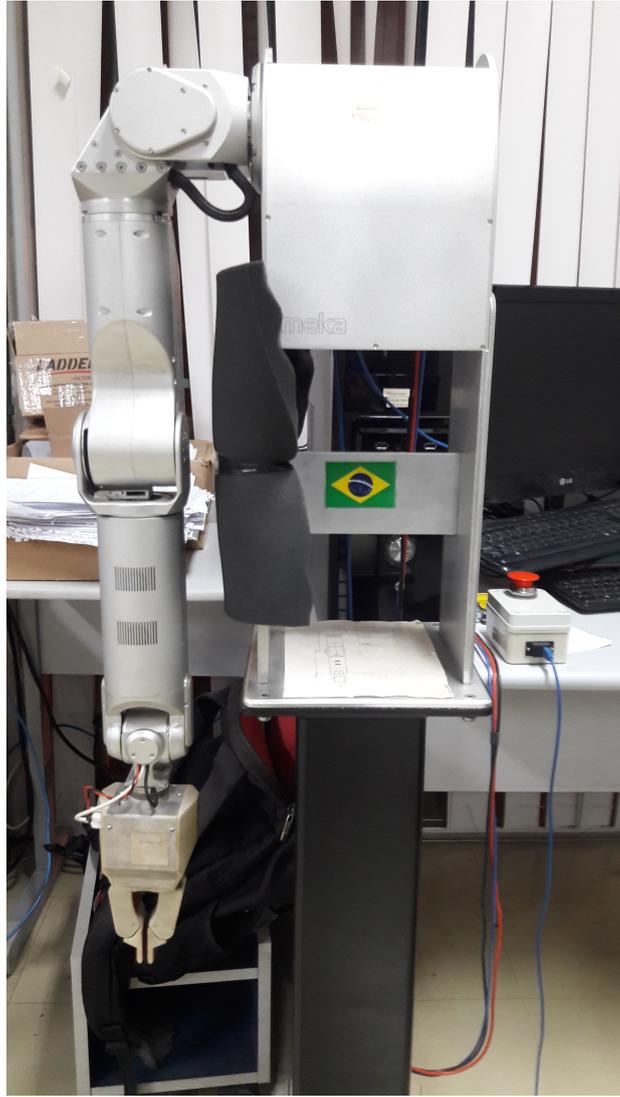


Figura 3.6: Posição de descanso do robô.

Depois, deslocou-se o efetuator terminal para a posição de calibração, onde o eixo  $z$  do sensor rotaciona em 180, ou seja, a força peso age somente sobre o eixo  $-z$ . Fez-se a medida desses valores por um tempo e tirou-se a média. Para achar o  $E(x, y, z)$ , fez-se então a média entre as forças medidas nas duas posições, conforme mostra a equação (3.8):

$$\begin{aligned}
 F_{descanso} + F_{calibração} &= P + E(x, y, z) - P + E(x, y, z), \\
 F_{descanso} + F_{calibração} &= 2E(x, y, z), \\
 \frac{F_{descanso} + F_{calibração}}{2} &= E(x, y, z),
 \end{aligned}
 \tag{3.8}$$

em que  $F_{descanso}$  é a força medida pelo sensor na posição de descanso e  $F_{calibração}$  é a força medida pelo sensor na posição de calibração. Após descoberto o erro associado ao sensor, é necessário subtrair esse erro de todas as medidas feitas no sensor. Assim, a força medida pelo sensor quando o robô não está sob influência de forças externas é apenas a força peso da garra  $P$ .

Antes do programa da interface háptica começar a fazer o controle do robô, faz-se então a medida das forças do sensor de força e tira a média para poder determinar P. Assim, pode-se achar a força externa no robô fazendo a subtração da força  $P$  e do erro  $E(x,y,z)$  da medida do sensor  $F_s$ . Essa força externa é utilizada para fazer o *feedback* de força para a interface háptica.

$$\begin{aligned} F_s &= F_E + E(x, y, z) + P \\ F_E &= F_s - P - E(x, y, z) \end{aligned} \tag{3.9}$$

Porém, é importante lembrar que após a determinar a força P o sensor pode rotacionar, mudando assim a leitura da força peso no sensor de força. Nesse caso, é necessário então fazer a rotação da força  $F_s$  e do erro  $E(x, y, z)$  para achar a força  $F_E$ .

$$\begin{aligned} F_{sR} &= F_E + P + E_R(x, y, z) \\ F_E &= (F_s - E(x, y, z))_R - P \end{aligned} \tag{3.10}$$

# Capítulo 4

## Resultados

### 4.1 Calibração

A calibração do sensor de força gera um gráfico mostrando a reta de calibração dos pontos utilizados, conforme mostra a Figura 4.1. Com esses pontos é feito a calibração do 0 do sensor e do ângulo da reta do sensor e, por isso, o procedimento de calibração foi feito várias vezes até se obter a melhor reta de interpolação entre os pontos.

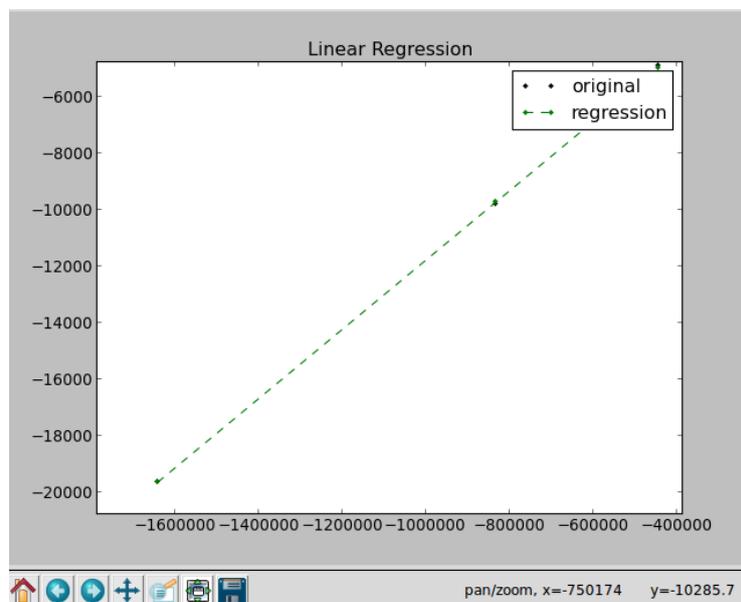


Figura 4.1: Curva de calibração gerada pelo API.

Após a calibração, fez-se um teste colocando o robô na posição de descanso sobre o efeito apenas do peso da garra. O conjunto da garra e acoplamento pesa 710g, os dados do sensor estão em mN e é feita uma mudança de escala multiplicando esses dados por  $0,001 * 0,05$  para atender os forças máximas na interface Háptica. Considerando  $g = 9,82m/s^2$ , a força peso  $P$  esperado pela medição do sensor é 0,3486. Os dados obtidos estão na Fig. 4.2.

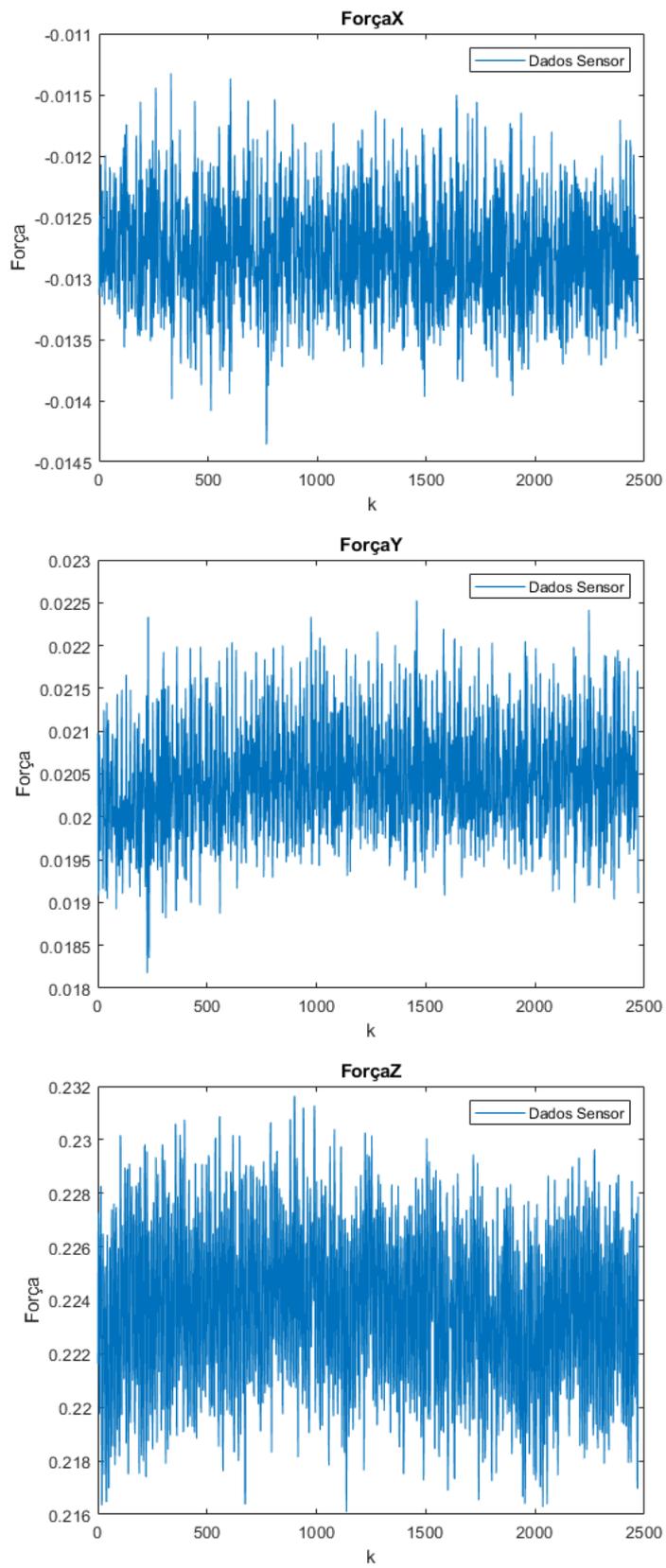


Figura 4.2: Dados do sensor na posição de descanso após a montagem da garra no sensor de força

Observa-se pela Figura 4.2 que todos os eixos possuem um erro associado ao processo de calibração. Para os eixos  $x$  e  $y$  o valor esperado era 0, uma vez que não estão sobre ação de nenhuma força. Para o eixo  $z$  o valor esperado era aproximadamente 0,3486. A grande diferença ocorrida no eixo  $z$  é causada pelo acoplamento da garra no sensor de força utilizando parafusos, que gera uma força contrária a força  $P$ . Para a próxima calibração do sensor, é sugerido desacoplar o sensor do robô, colocá-lo em um superfície nivelada e realizar os procedimentos descritos.

## 4.2 Filtro Butterworth

Conforme dito na seção 3.2, é necessário verificar a atenuação do ruído bem como o atraso do sinal. Para isso, os testes foram feitos deixando o robô na posição de descanso e depois movendo para uma nova posição, que será chamada de posição de teste e possui ângulos nas juntas  $(0, 0,4, 0, 1,56, 0, 0, -0,093)$  radianos, conforme mostra a Figura 4.3, a fim de variar a força lida em cada eixo. A frequência de corte do filtro Butterworth 1 < Butterworth 2 < Butterworth 3.



Figura 4.3: Posição de teste do robô.

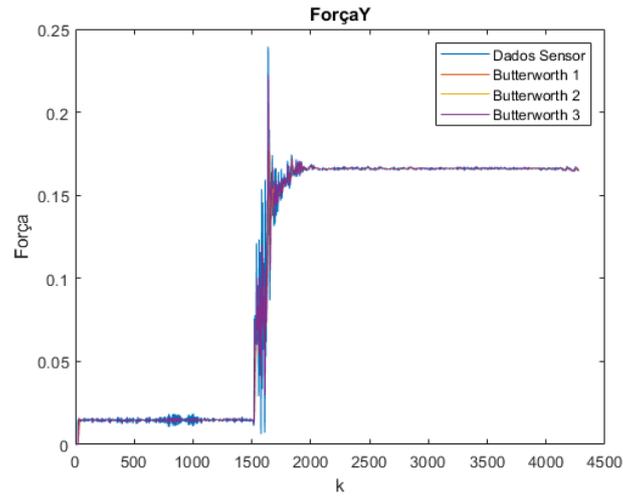


Figura 4.4: Resultado do teste dos três filtros de Butterworth.

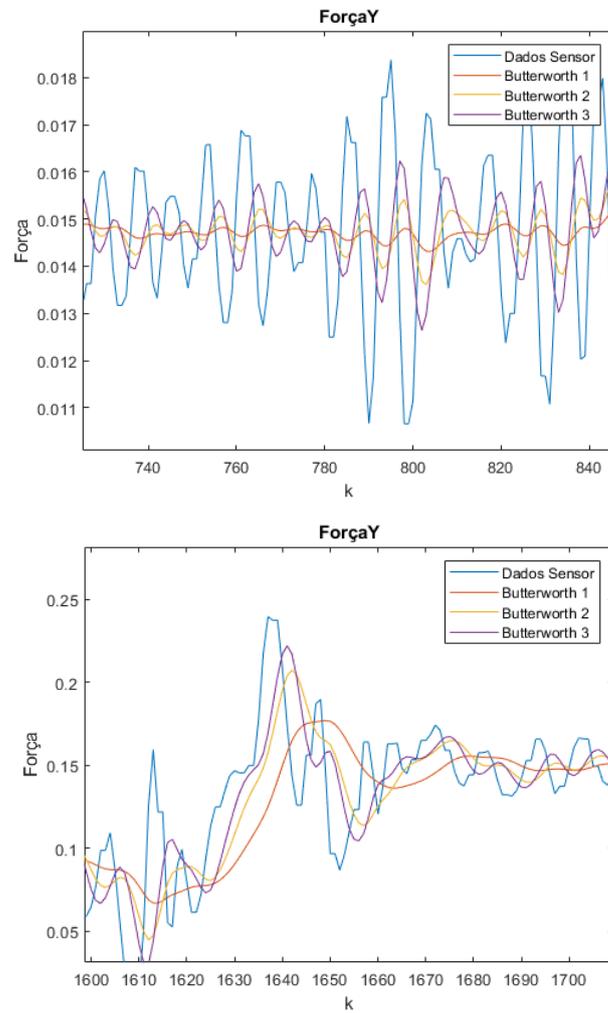


Figura 4.5: Ampliação da Figura 4.4 em duas regiões distintas.

Conforme o esperado, quanto maior a frequência de corte do filtro, menor vai ser a atenuação do ruído mas mais rápido o sinal filtrado acompanha o sinal original. Utilizou-se o MATLAB para encontrar o valor que acontece o pico de cada um dos sinais e a respectiva amostra, para poder comparar o atraso dos filtros. Utilizando a função *rostime* do ROS é possível salvar o tempo de cada iteração e verificar o atraso em segundos. O período de amostragem do sinal é 0,008 segundos, e foi considerado  $t = 0$  para a amostra 1637.

Tabela 4.1: Comparação do atraso de cada filtro.

Sinal	Valor Pico (N)	Amostra Pico (k)	Tempo Pico (s)
Dados Sensor	0,239422	1637	0
Butterworth 1	0,176886	1649	0,096
Butterworth 2	0,207239	1642	0,04
Butterworth 3	0,221975	1641	0,032

É importante conhecer a aplicação para escolher qual o melhor filtro. Para o *feedback* de forças para a interface Háptica, considerou que o atraso de 100 ms era aceitável pela atenuação do ruído do filtro. O filtro implementado para rodar em tempo real foi o Butterworth 1, que possui função de transferência e equação de diferença:

$$\frac{Y(z)}{U(z)} = \frac{0.0055427172 + 0.011085434z^{-1} + 0.0055427172z^{-2}}{1 - 1.7786317778z^{-1} + 0.8008026467z^{-2}}$$

$$y(k) = 0.0055...u(k) + 0.011...u(k - 1) + 0.0055...u(k - 2) - 1.77...y(k - 1) - 0.80...y(k - 2) \quad (4.1)$$

Os dados da filtragem em tempo real foram salvos e comparados com uma filtragem feito no MATLAB após a finalização do experimento. É importante lembrar que o filtro implementado pode ser facilmente alterado modificando os coeficientes da equação de transferência.

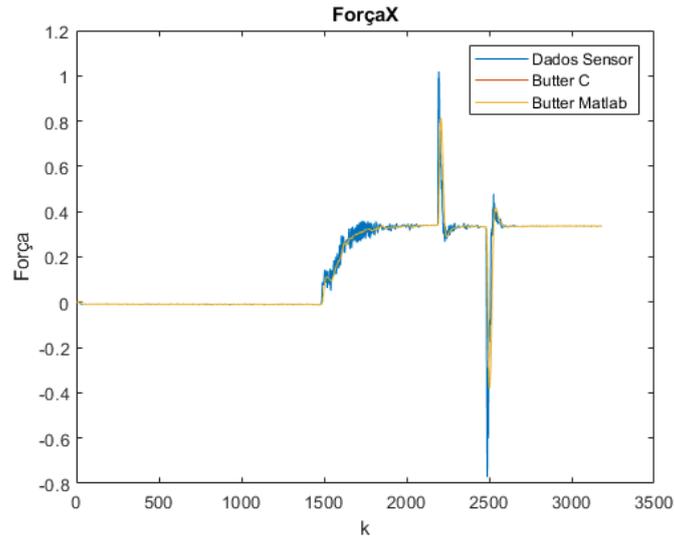


Figura 4.6: Filtro Butterworth implementado em C e no MATLAB.

A Figura 4.6 mostra que não existe diferença significativa entre a implementação do filtro em C e em MATLAB.

### 4.3 Filtro de Kalman Adaptativo

Para cada iteração do filtro, faz-se o cálculo dos valores de  $Q_k$  e  $R_k$  com base nos valores de  $Q_{k-1}$ ,  $R_{k-1}$  e um fator relacionado ao erro do processo e da medição. O fator  $\alpha$  das equações (2.22) e (2.23) mostram a confiabilidade no valores de  $R_{k-1}$  e  $Q_{k-1}$  e nos erros de processo e de medição atual. Para um valor de  $\alpha$  próximo de 1, os pesos dos fatores  $Q_{k-1}$  e  $R_{k-1}$  são maiores, o que implica em uma lenta mudança dos valores  $Q_k$  e  $R_k$  e uma lenta adaptação às mudanças do processo. Para um valor de  $\alpha$  próximo de 0, os pesos dos erros de processo e medição atual são maiores, o que implica em uma rápida mudança dos valores de  $Q_k$  e  $R_k$  e uma rápida adaptação às mudanças do processo. As Figuras 4.7 e 4.8 mostram o efeito da escolha do  $\alpha$  no filtro.

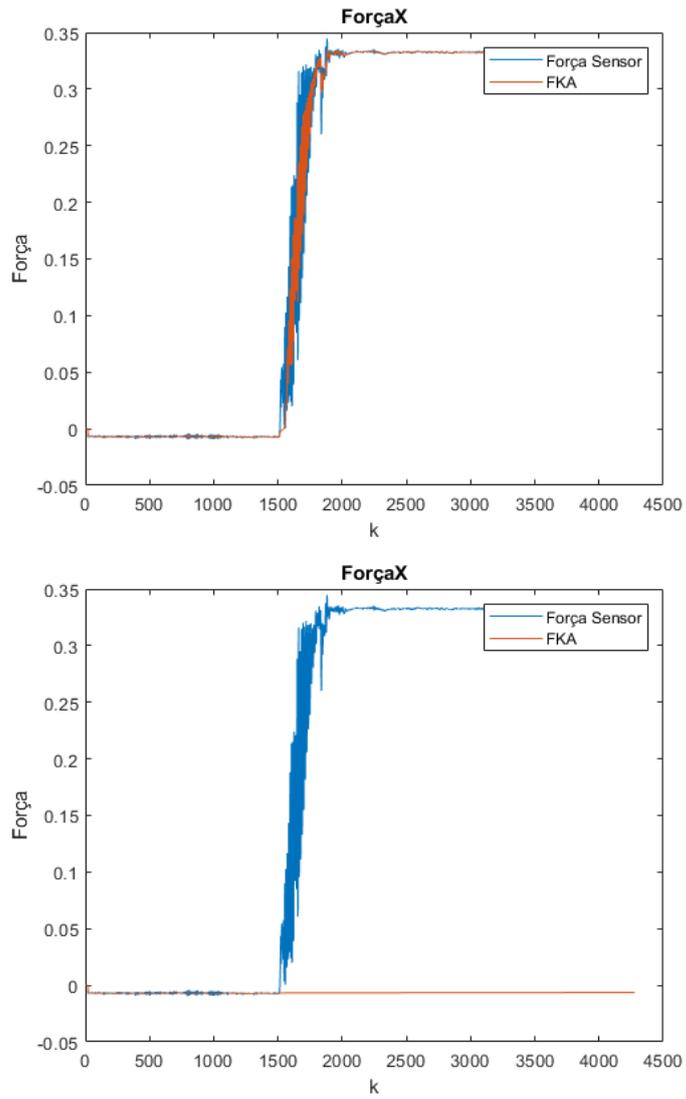


Figura 4.7: Comparação da força no eixo x para os filtros Kalman para  $\alpha = 0.1$  e  $\alpha = 0.9$

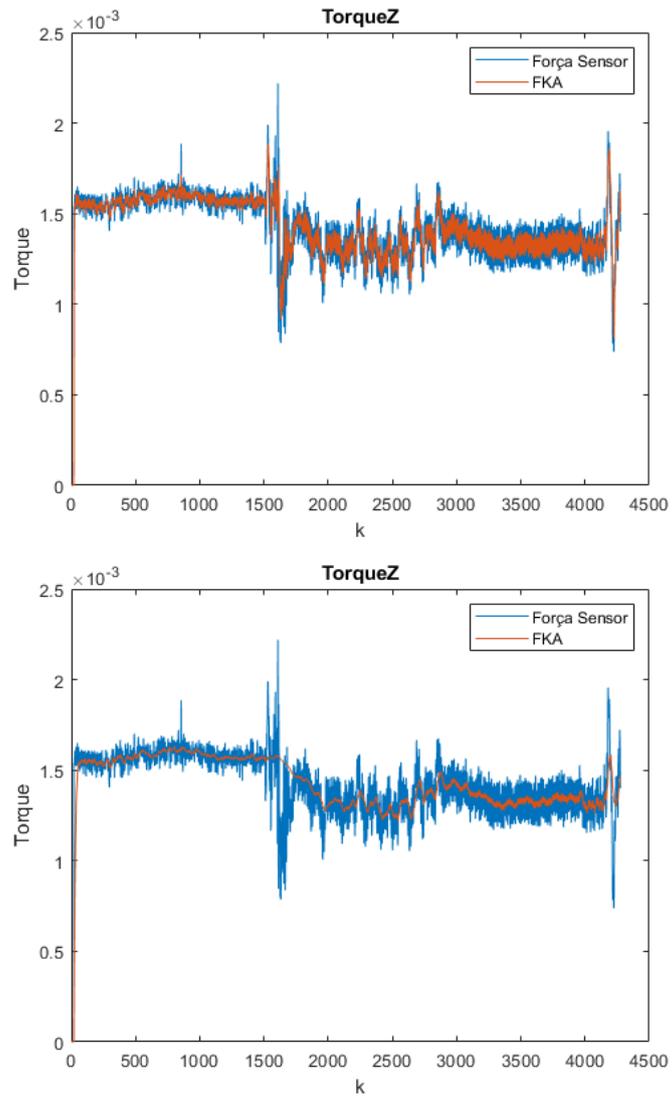


Figura 4.8: Comparação do torque no eixo z para os filtros de Kalman para  $\alpha = 0.1$  e  $\alpha = 0.9$

Conforme o esperado, para um valor de  $\alpha$  próximo de 0, o filtro consegue acompanhar variações no processo de forma rápida, mas em compensação ele filtra menos os dados. Para  $\alpha$  próximo de 1, o filtro não consegue acompanhar variações rápidas no processo, mas a atenuação do ruído é maior.

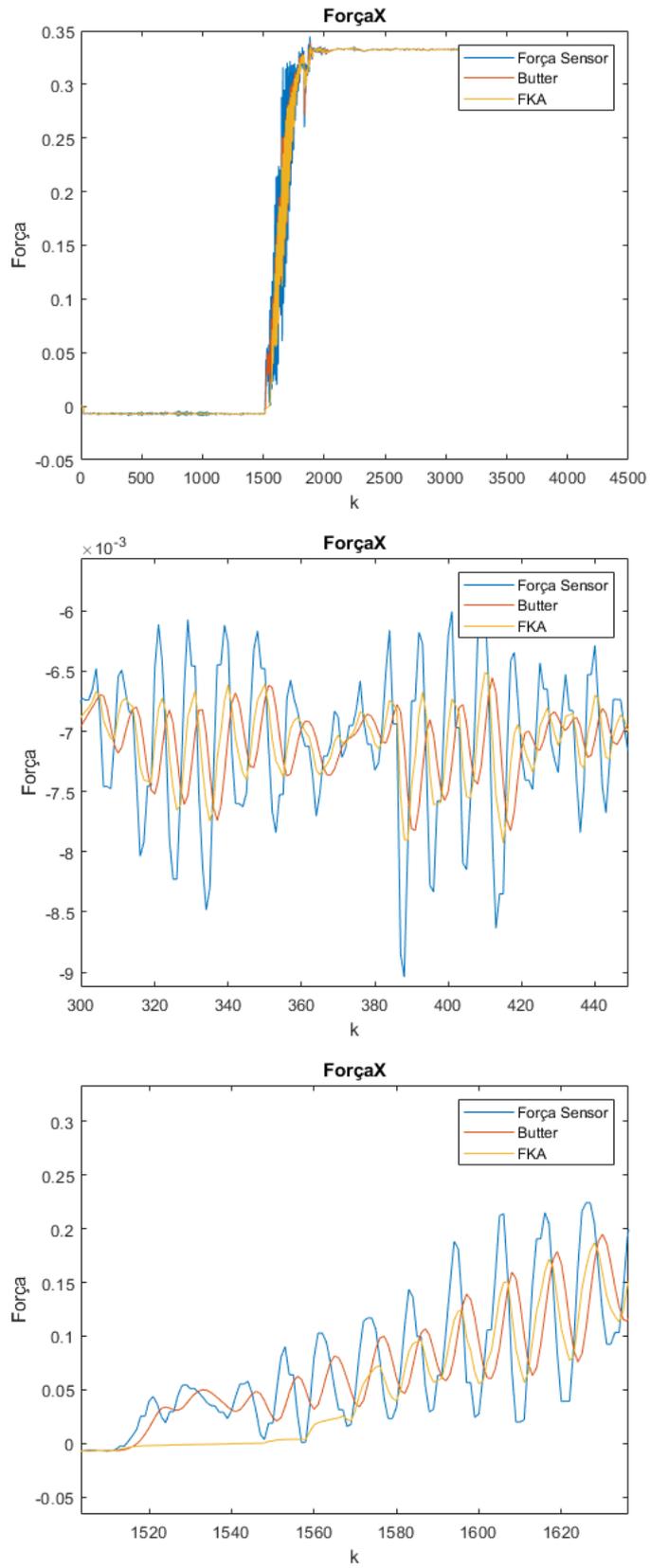


Figura 4.9: Comparação do FKA e um filtro Butterworth

A Figura 4.9 mostra uma comparação de um filtro de Kalman adaptativo com  $\alpha = 0.1$  e um filtro Butterworth projetado para ter uma atenuação de ruído parecido com o FKA. Pode-se observar que o filtro de Kalman adaptativo possui menos atraso que o filtro Butterworth, mas ele precisa de mais tempo para se adaptar a variações no processo. Além disso, quanto mais rápida a mudança do processo, menor tem que ser o valor de  $\alpha$  para conseguir acompanhar a mudança e, conseqüentemente, menor vai ser a atenuação de ruído.

O filtro de Kalman adaptativo também foi implementado em C utilizando a mesma metodologia do MATLAB. A Figura 4.10 mostra que a implementação desse filtro em C está não está exatamente igual ao MATLAB. A fonte dessa diferença pode ser causada pela biblioteca *eigen* e as aproximações feitas para fazer os cálculos matriciais.

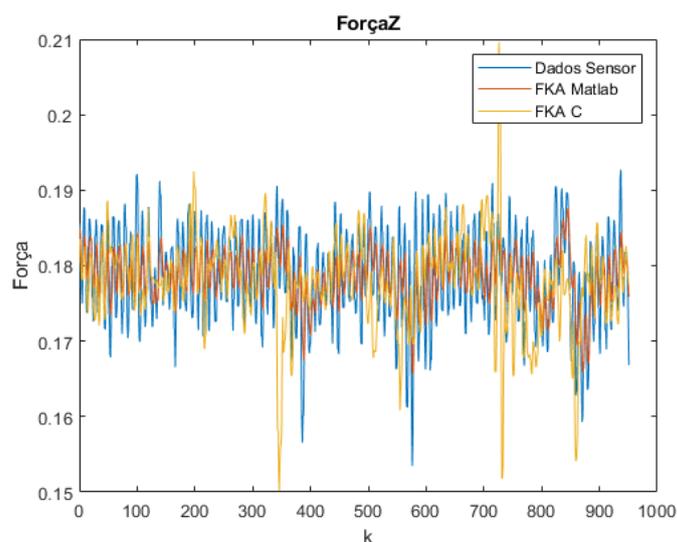


Figura 4.10: Comparação entre o FKA implementado em MATLAB e em C.

#### 4.4 Rotação de forças e compensação da gravidade

A Figura 4.11 mostra um diagrama de blocos das etapas e onde as forças foram medidas.

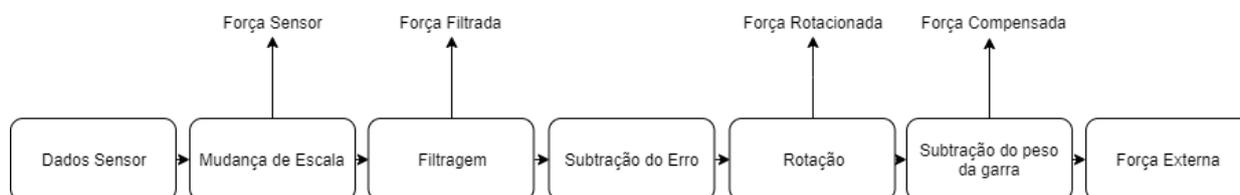


Figura 4.11: Etapas realizadas para o *feedback* Háptico.

A Figura 4.12 mostra o resultado depois da realização de todos os procedimentos descritos. A Força Sensor, reta azul, mostra a leitura dos dados do sensor nos eixos x,y e z apenas passando por uma mudança de escala. A Força Filtrada, reta vermelha, mostra o resultado da filtragem a

Força Sensor. A Força Rotacionada, reta amarela, é a Força Filtrada após a rotação e a subtração do erro  $E(x,y,z)$ . Por fim, a Força Compensada é a Força Rotacionada após a subtração do peso da garra  $P$ . A força utilizada para fazer o *feedback* para interface Háptica é a Força Rotacionada.

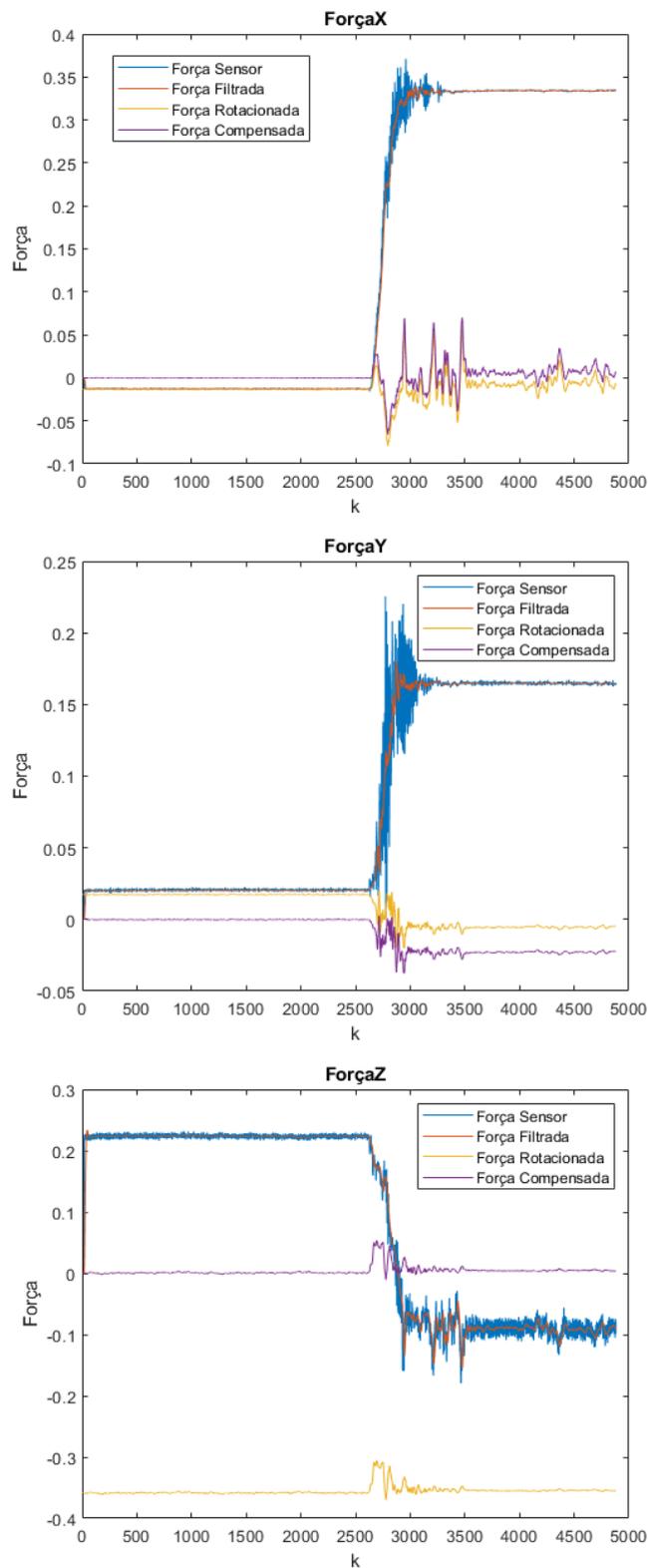


Figura 4.12: Rotação e compensação de forças.

O experimento foi feito deixando o robô na posição de descanso por um tempo e depois movendo ele para a posição de teste. Nesse caso, a única força que o robô estava submetido era a força  $P$ . Inicialmente pode-se observar a diferença causada pela filtragem dos dados e como ela afeta no *feedback* de força para o usuário.

Comparando a Força Filtrada com a Força Rotacionada enquanto o robô estava na posição de descanso, observa-se que a força medida pelo sensor de força não representa a força peso causada pela garra. Comparando a Força Rotacionada com o Força Compensada, observa-se que a força  $P$  foi identificada e compensada corretamente, mesmo quando o robô mudou de orientação. Durante a transição entre as posições, surge uma nova força relacionada ao movimento do robô, que é causada pela inércia. Essa força pode ser compensada utilizando o modelo dinâmico do robô. Além disso, o ruído do sensor aumenta significativamente, e esse ruído pode ser causado por uma interferência devido ao aumento da corrente nas juntas do robô durante o movimento.

As Figuras 4.13 e 4.14 mostram mais testes realizados para validar a compensação da gravidade. O testes foram realizados deixando o robô na posição de descanso durante um tempo e mudando para a posição de teste. Foram aplicadas forças externas no efetuador terminal após ele mudar para a posição de teste no teste referente a Fig. 4.13 e a interface Háptica foi utilizada para controlar o robô no teste referente a Fig. 4.14.

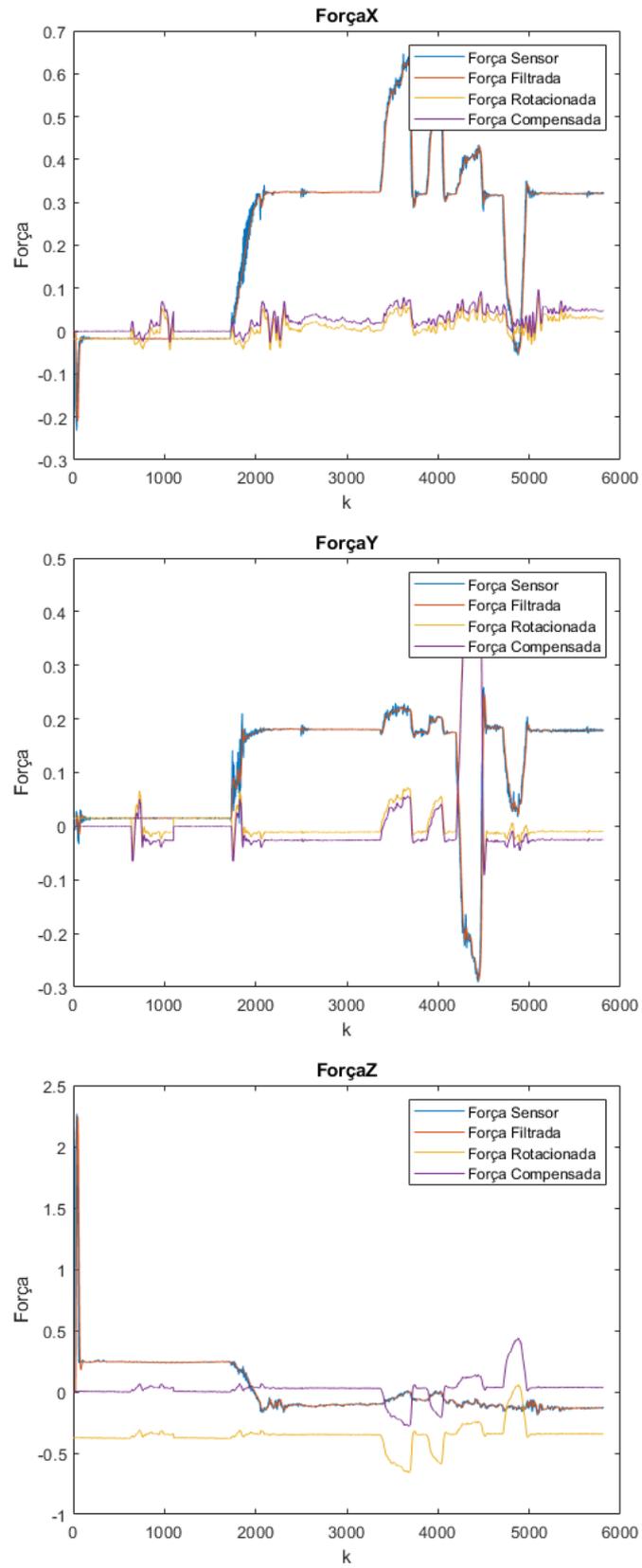


Figura 4.13: Teste de forças para posição de teste com aplicação de força externa.

Observa-se na Figura 4.13 um pequeno erro na compensação das forças nos eixos x e y, e a efetividade da compensação do peso da garra. A força externa aplicada ao robô é extraída dos dados do sensor e utilizada para o *feedback* Háptico.

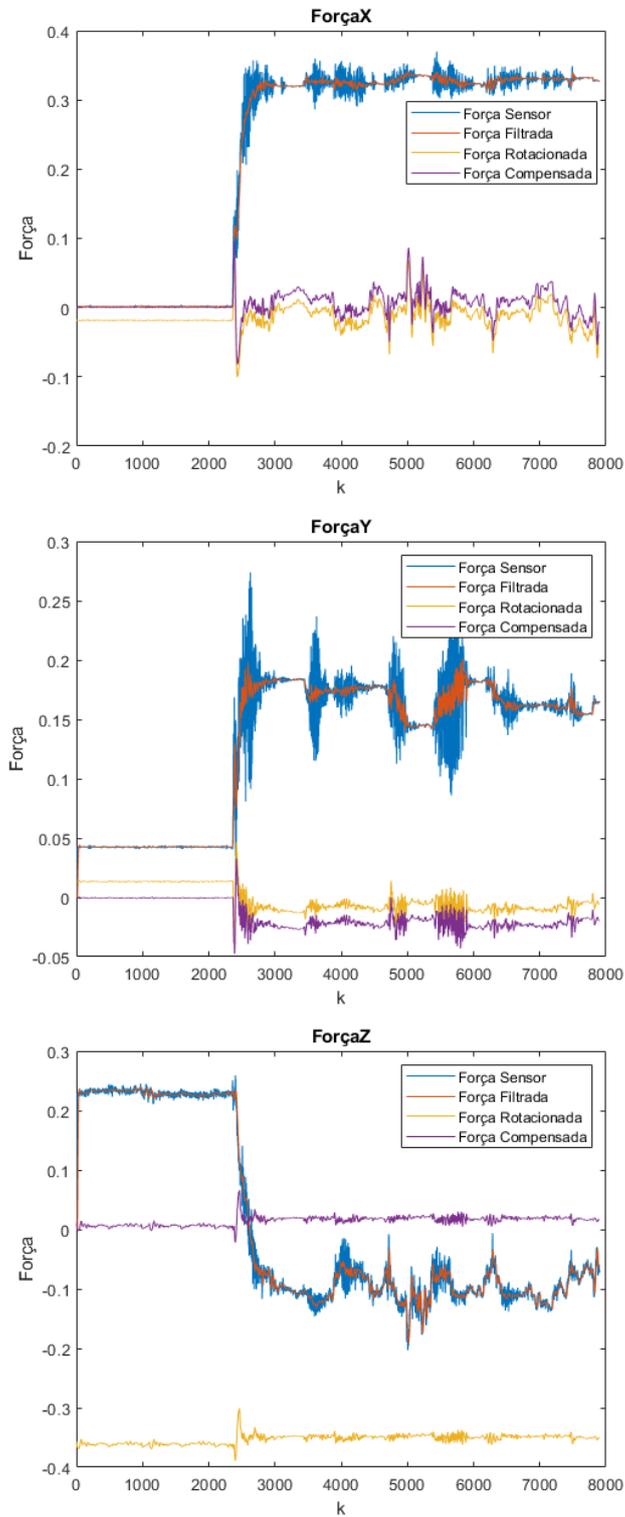


Figura 4.14: Teste de forças para posição de teste com movimentação do robô.

Observa-se na Figura 4.14 que a movimentação do robô gera muito ruído na medição do sensor. Conforme dito anteriormente, esse ruído pode ser causado por uma interferência gerada pelo aumento da corrente nas juntas do robô durante o movimento. O filtro Butterworth implementado consegue fazer uma atenuação nesse ruído, mas ainda pode comprometer o *feedback* Háptico. Para uma melhor atenuação desse ruído, é necessário a mudança do controlador utilizado

# Capítulo 5

## Conclusões

A indústria 4.0 vem atraindo a atenção de grandes empresas nos últimos anos. Ela é impulsionada pela integração de novas tecnologias e pelos avanços tecnológicos em diversas áreas. Dentre esses avanços tecnológicos, os robôs industriais estão se tornando mais autônomos, flexíveis e cooperativos e eventualmente irão trabalhar junto com os humanos para um aumento de produtividade e flexibilidade na célula de produção. Uma forma de cooperação humano-robô e a utilização de uma interface háptica para fazer o controle de um robô. Essa interface permite uma comunicação bilateral onde o humano controla o robô e consegue sentir as forças que o robô está exercendo. Esse *feedback* de forças é importante para o humano conseguir informações do processo além da informação visual.

O LARA possui um robô complacente que é controlado por uma interface háptica. Esse robô possui um sensor de força e torque no punho e seus dados são utilizados para fazer o *feedback* de força. O controle e um *feedback* das forças inicial foram implementados por Rafael Koji [1], que focou em fazer a estrutura que permite controlar o robô utilizando a interface háptica. O trabalho tem como objetivo melhorar o *feedback* fazendo a implementação de filtros digitais e a compensação da força peso do efetuador terminal.

Inicialmente foi refeita a calibração do sensor de força, cuja última calibração tinha sido feita e documentada por Rafael Koji [1] em 2017. Foi utilizado um *script* em python disponibilizado pelo próprio fabricante. Depois da calibração, foram projetados e implementados dois filtros digitais para a atenuação do ruído do sensor, um filtro Butterworth e um filtro de Kalman adaptativo. Por fim, foi feito o modelo da leitura dos dados do sensor e como a força peso poderia ser extraída para sobrar apenas a força externa aplicada ao efetuador terminal. Esse força externa é utilizada para fazer o *feedback* háptico.

A calibração do sensor consiste em aplicar massas de 0g, 500g, 1000g e 2000g no eixo -z, que são utilizadas para gerar a curva de calibração e o eixo zero. Para a realização da calibração, o robô foi colocado em uma posição onde a força peso está alinhada com o eixo -z do sensor e foram colocados objetos com as massas necessárias. Foi utilizado o MATLAB para fazer o projeto de três filtros Butterworth com frequências de corte diferentes para fazer a atenuação do ruído, que foram comparados em relação à atenuação e ao atraso. O filtro escolhido foi implementado no

código em C referente a interface háptica. O filtro de Kalma adaptativo também foi projetado utilizando MATLAB e implementado em C. Os dados do sensor de força foram modelados como a soma da força externa, da força peso e de um erro associado ao processo de calibração. Foram realizados experimentos para extrair o erro e a força peso, que são subtraídos da força do sensor para descobrir a força externa.

A análise dos resultados mostra que os objetivos do projeto foram alcançados de maneira satisfatória. A calibração feita possui um pequeno erro associado a metodologia utilizada, mas esse erro é compensado. A implementação em C do filtro de Butterworth faz a atenuação do ruído em tempo real e melhora o *feedback* háptico. A implementação em C do filtro de Kalman adaptativo possui diferenças do projetado em MATLAB e não consegue fazer a atenuação do ruído. A compensação da força peso e do erro do sensor foram satisfatórios.

## 5.1 Perspectivas Futuras

Ainda na área de processamento de sinais, trabalhos futuros podem envolver a união dos *encoders* nas juntas do robô e os dados do sensor de força utilizando um filtro de Kalman ou fazer a modelagem dinâmica do robô para a compensação dinâmica da gravidade.

Fora da área de processamento de sinais, pode-se considerar que este trabalho foi o suficiente e continuar os trabalhos com a interface háptica e o robô como a melhora do controlador utilizado, o aumento da área de trabalho e a implementação de um efetuator terminal atuado.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BRUNELLO, R. K. V. Teleoperação de robô manipulador através de interface háptica com feedback de força. 2017.
- [2] USTUNDAG, A.; CEVIKCAN, E. *Industry 4.0: Managing The Digital Transformation*: Springer International Publishing, 2017. (Springer Series in Advanced Manufacturing). ISBN 9783319578705.
- [3] RÜSSMANN, M. et al. Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, v. 9, n. 1, p. 54–89, 2015.
- [4] PRIEMER, R. *Introductory Signal Processing*: World Scientific, 1991. (Advanced series in electrical and computer engineering). ISBN 9789971509194.
- [5] SCHLICHTHÄRLE, D. *Digital Filters: Basics and Design*: Springer Berlin Heidelberg, 2011. ISBN 9783642143250.
- [6] GREWAL, M.; ANDREWS, A. *Kalman Filtering: Theory and Practice with MATLAB*: Wiley, 2015. (Wiley - IEEE). ISBN 9781118984963.
- [7] ANDERSON, B.; MOORE, J. *Optimal Filtering*: Dover Publications, 2012. (Dover Books on Electrical Engineering). ISBN 9780486136899.
- [8] AKHLAGHI, S.; ZHOU, N.; HUANG, Z. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. In: IEEE. *2017 IEEE Power & Energy Society General Meeting*, 2017. p. 1–5.
- [9] HUANG, D. et al. Application of adaptive kalman filter in online monitoring of mine wind speed. Preprints, 2019.
- [10] LEE, S.; SUH, I.; KIM, M. *Recent Progress in Robotics: Viable Robotic Service to Human: An Edition of the Selected Papers from the 13th International Conference on Advanced Robotics*: Springer, 2008. (Lecture Notes in Control and Information Sciences). ISBN 9783540767282.
- [11] BRADY, M. et al. *Robot Motion: Planning and Control*: MIT Press, 1982. (Artificial Intelligence Series). ISBN 9780262021821.
- [12] HOARAU, A. *Mekabot*. 2017. Disponível em: <(https://github.com/ahorau/mekabot/wiki/Mekabot-overview)>.

- [13] SAMUR, E. *Performance Metrics for Haptic Interfaces*: Springer London, 2012. (Springer Series on Touch and Haptic Systems). ISBN 9781447142256.
- [14] MATHWORKS. *Butterworth filter design*. 2019. <https://www.mathworks.com/help/signal/ref/butter.html>. Acessado em 30 de novembro de 2019.