

TRABALHO DE GRADUAÇÃO

**TELEOPERAÇÃO DE ROBÔ MANIPULADOR  
ATRAVÉS DE INTERFACE HÁPTICA  
COM FEEDBACK DE FORÇA**

Rafael Koji Vatanabe Brunello

Brasília, dezembro de 2017



**ENGENHARIA  
MECATRÔNICA**  
UNIVERSIDADE DE BRASÍLIA

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

TRABALHO DE GRADUAÇÃO

**TELEOPERAÇÃO DE ROBÔ MANIPULADOR  
ATRAVÉS DE INTERFACE HÁPTICA  
COM FEEDBACK DE FORÇA**

**Rafael Koji Vatanabe Brunello**

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

Banca Examinadora

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

\_\_\_\_\_

Prof. Mariana Costa Bernardes, FGA/UnB  
*Coorientadora*

\_\_\_\_\_

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

\_\_\_\_\_

Dr. Eng. Roberto de Souza Baptista  
*Examinador*

\_\_\_\_\_

**Brasília, dezembro de 2017**

## FICHA CATALOGRÁFICA

RAFAEL KOJI, VATANABE BRUNELLO

Teleoperação de robô manipulador através de interface háptica com feedback de força,  
[Distrito Federal] 2017.

28, 84p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2017). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1. Teleoperação

2.Interface háptica

3. Manipuladores Complacentes

I. Mecatrônica/FT/UnB II. Teleoperação de robô manipulador através de interface háptica com feedback de força. (n°028)

## REFERÊNCIA BIBLIOGRÁFICA

BRUNELLO, RAFAEL KOJI VATANABE, (2017). Teleoperação de robô manipulador através de interface háptica com feedback de força. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-n°028, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 84p.

## CESSÃO DE DIREITOS

AUTOR: Rafael Koji Vatanabe Brunello

TÍTULO DO TRABALHO DE GRADUAÇÃO: Teleoperação de robô manipulador através de interface háptica com feedback de força.

GRAU: Engenheiro

ANO: 2017

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.

---

Rafael Koji Vatanabe Brunello

Quadra 06, Conjunto H, Casa 20.

73025-068 Sobradinho – DF – Brasil.

## **Dedicatória**

*À minha família , por sua capacidade de acreditar e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.*

*Rafael Koji Vatanabe Brunello*

## Agradecimentos

*Primeiramente agradeço aos meus pais, sem os quais, provavelmente, não teria conseguido nem entrar na universidade que dirá então finalizá-la. Em seguida, agradeço aos meus orientadores Geovany e Mariana por seus conselhos e orientações durante o processo de desenvolvimento deste trabalho. Agradeço também, aos colegas Marcos Pereira e Rafael Lima, os quais auxiliaram imensamente ao trabalharmos em conjunto no manipulador Meka. Ademais, agradeço aos meus amigos, especialmente o João Carlos e o José Gabriel por, além de ajudarem na realização dos testes e diagramação dos gráficos, estarem sempre disponíveis e dispostos a ajudar e muitas vezes distrair do trabalho com nossas maravilhosas resenhas de virar noites conversando. Da mesma forma, agradeço aos meus colegas de laboratório, em especial ao guru do LARA, o grande menino Moisés, sem o qual diversos erros que ocorreram durante o desenvolvimento deste trabalho não teriam sido solucionados. Por último, mas com certeza não menos importante, agradeço a minha namorada Andressa, que de forma especial e carinhosa me deu força e coragem, me apoiando nos momentos de dificuldades, virando noites em claro corrigindo meus milhares de erros gramaticais, me dando broncas quando eu queria desistir e ficando mais nervosa do que eu em minha própria defesa. À todas estas pessoas e a todos aqueles que me apoiaram e auxiliaram por este caminho árduo que foi minha graduação só posso lhes dizer com toda a sinceridade e gratidão: Muito obrigado.*

*Rafael Koji Vatanabe Brunello*

---

## RESUMO

Este trabalho implementa uma teleoperação de um braço manipulador Meka A2 utilizando uma interface háptica Omega7. Tendo como motivação avaliar a possibilidade de solucionar desvantagens comuns da telecirurgia como longos tempos de setup de equipamento, alto custo de equipamento e principalmente a falta de retorno háptico para os médicos cirurgiões. Para isto, são desenvolvidos códigos para servirem de interfaces de comunicação entre o braço manipulador e a interface háptica. A fim de facilitar a comunicação e a simulação de robôs são utilizadas ferramentas como o ROS e o VREP. São então descritos os equipamentos, fundamentos teóricos e a motivação por trás de seus usos. Em seguida, descreve-se o processo de desenvolvimento, a metodologia utilizada e os erros encontrados e solucionados. Após isto, a qualidade do retorno háptico e da movimentação do robô são analisados em relação ao resultado dos testes escolhidos. Por fim, conclui-se com a verificação dos objetivos alcançados e formulação de trabalhos futuros.

Palavras Chave: Teleoperação, Interface háptica, Manipuladores complacentes

---

## ABSTRACT

This work implements a teleoperation of an robot manipulator Meka A2 using an Omega7 haptic interface. Its motivation is to evaluate the possibility of solving common disadvantages of telesurgery, such as long equipment setup times, high equipment costs, and especially the lack of haptic feedback for surgeons. For this, codes have been developed to serve as communication interfaces between the manipulator arm and a haptic interface. In order to facilitate the communication and simulation of robots that are used tools like ROS and VREP. It then describes the equipment and theoretical foundations and a motivation behind its uses. Next, is described the development process, the methodology used and, the problems found and solved. In addition, the quality of the return and the movement of the robot are analyzed in relation to the result of the tests chosen. Finally, we conclude with a verification of the cleared objectives and formulation of future work.

Keywords: Teleoperation, Haptic interface, Compliant manipulators

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	CONTEXTUALIZAÇÃO	1
1.2	TRABALHOS RELACIONADOS	3
1.2.1	TELEOPERAÇÃO/TELECIRURGIA	3
1.2.2	MANIPULADORES COMPLACENTES	4
1.2.3	CONTROLE DE MANIPULADORES COM QUATÉRNIONS DUAIS	5
1.2.4	INTERFACE HÁPTICA/FEEDBACK DE FORÇA	6
1.3	OBJETIVOS	6
<b>2</b>	<b>FERRAMENTAS DE APOIO A APLICAÇÃO</b>	<b>8</b>
2.1	INTERFACE HÁPTICA OMEGA	8
2.2	MEKA A2 COMPLIANT ARM	9
2.2.1	REAL TIME CONTROL PC (RTPC/MEKAMAN)	10
2.2.2	FORCE TORQUE SENSOR (FTS)	11
2.3	ROBOT OPERATING SYSTEM (ROS)	12
2.4	CATKIN	13
2.5	VIRTUAL-ROBOTIC EXPERIMENTATION PLATAFORM (VREP)	13
2.6	FUNDAMENTOS MATEMÁTICOS	14
2.6.1	NÚMEROS DUAIS	14
2.6.2	QUATÉRNIONS	15
2.6.3	QUATÉRNIONS DUAIS (DQ)	16
2.7	CONTROLADOR	18
<b>3</b>	<b>METODOLOGIA</b>	<b>20</b>
3.1	VISÃO GERAL	20
3.2	OMEGA7	20
3.3	VREP	22
3.3.1	MODELO KUKA	22
3.3.2	CONTROLE HÁPTICO	23
3.3.3	MODELO MEKA	24
3.3.4	PROGRAMAÇÃO ROBOT CONTROLLING NODE	27
3.3.5	PROGRAMAÇÃO OMEGA7-ROS INTERFACE	29
3.4	BRAÇO MANIPULADOR MEKA A2	30

3.4.1	SENSOR DE FORÇA .....	31
3.4.2	MODELO CINEMÁTICO .....	32
<b>4</b>	<b>ANÁLISE DE RESULTADOS .....</b>	<b>35</b>
4.1	VALIDAÇÃO .....	35
4.2	ANÁLISE DOS RESULTADOS DA SIMULAÇÃO .....	36
4.3	ANÁLISE DOS RESULTADOS NO MANIPULADOR REAL .....	36
4.4	ANÁLISE DOS RESULTADOS DE POSICIONAMENTO .....	38
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>40</b>
5.1	PERSPECTIVAS FUTURAS .....	40
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>42</b>
	<b>ANEXOS .....</b>	<b>46</b>
<b>A</b>	<b>FLUXOGRAMA VREP-MEKA-INTERFACE .....</b>	<b>47</b>
<b>B</b>	<b>FLUXOGRAMA OMEGA7-ROS-INTERFACE .....</b>	<b>49</b>
<b>C</b>	<b>FLUXOGRAMA DO HAPTIC CONTROL .....</b>	<b>51</b>
<b>D</b>	<b>PESOS DE REFERÊNCIA .....</b>	<b>53</b>
<b>E</b>	<b>GRÁFICOS DE CALIBRAÇÃO .....</b>	<b>57</b>
<b>F</b>	<b>ESPECIFICAÇÕES TÉCNICAS OMEGA7 .....</b>	<b>59</b>
<b>G</b>	<b>ESPECIFICAÇÕES TÉCNICAS MEKA A2 .....</b>	<b>61</b>
<b>H</b>	<b>MODELO CINEMÁTICO MEKA A2 .....</b>	<b>63</b>
<b>I</b>	<b>GRÁFICOS DE FORÇA .....</b>	<b>65</b>
I.1	SIMULAÇÃO .....	65
I.1.1	TESTE COM O ROBÔ PARADO .....	65
I.1.2	TESTE COM O MANIPULADOR EM MOVIMENTO .....	66
I.1.3	TESTE DE COLISÃO .....	67
I.2	TESTES COM O ROBÔ REAL .....	67
I.2.1	TESTE COM O MANIPULADOR PARADO .....	67
I.2.2	TESTE COM O MANIPULADOR EM MOVIMENTO .....	68
I.2.3	TESTE DE COLISÃO .....	69
<b>J</b>	<b>GRÁFICOS DE POSIÇÃO .....</b>	<b>71</b>
J.1	SIMULAÇÃO .....	71
J.2	ROBÔ REAL .....	72



# LISTA DE FIGURAS

1.1	Robô cirúrgico Da Vinci .....	2
1.2	Robô cirúrgico Zeus .....	4
1.3	Ilustração de um SEA .....	5
1.4	Tactical Haptics VR Motion Controller da empresa Tactical Haptics.....	6
2.1	Sistema de coordenadas do Omega .....	8
2.2	Sistema de coordenadas do Omega7 .....	9
2.3	Braço robótico complacente Meka A2 .....	10
2.4	Sensor de força e torque ATI Mini-40 SI-80-4 .....	11
2.5	Modelo do Nível de Arquivos de Sistema .....	12
2.6	Modelo do Nível Grafo Computacional .....	13
2.7	Modos de APIs remotas.....	14
3.1	Sistema proposto.....	21
3.2	Calibração dos eixos de translação da interface háptica .....	21
3.3	Modelo Kuka LBR iiwa 7 R800 no VREP em pose de descanso .....	22
3.4	Simulação com o Kuka na pose de trabalho .....	23
3.5	Modelo Meka em sua pose inicial no VREP .....	25
3.6	Composição de frames rotations do Meka .....	26
3.7	Sistema proposto para uso do Robot Controlling Node.....	27
3.8	Fluxograma simplificado do Robot Controlling Node .....	27
3.9	Gráfico de comunicação dos nós do ROS durante a execução dos programas .....	29
3.10	Sistema proposto para uso do Omega7-ROS Interface .....	29
3.11	Fluxograma simplificado do Omega7-ROS-Interface.....	30
4.1	Gráfico ampliado de força no eixo Z no teste de colisão da simulação .....	37
4.2	Nova posição inicial .....	37
4.3	Gráfico ampliado de força no eixo Z no teste de movimentação livre no robô real .....	38
4.4	Gráfico ampliado de força no eixo Z no teste de colisão do robô real .....	39
4.5	Resultado do teste de escrita .....	39
A.1	Fluxograma do Robot Controlling Node.....	48
B.1	Fluxograma do Omega7 ROS Interface .....	50
C.1	Fluxograma do Haptic Control.....	52

D.1	Peso de referência para 500g .....	54
D.2	Peso de referência para 1000g .....	54
D.3	Peso de referência para 2000g .....	55
D.4	Peso da garra completa .....	55
D.5	Peso da placa de acoplamento .....	56
D.6	Peso da contenção da placa com a garra .....	56
E.1	Primeiro gráfico de saída da calibração <b>cl</b> .....	58
E.2	Segundo gráfico de saída da calibração <b>cl</b> .....	58
H.1	Modelo cinemático do Meka A2 .....	64

# LISTA DE SÍMBOLOS

## Símbolos Gregos

$\varepsilon$	Componente dual	
$\theta$	Ângulo	rad

## Grupos Adimensionais

$\mathbf{i}, \mathbf{j}, \mathbf{k}$	Versores unitários
$\mathbb{R}$	Conjunto dos números reais
$\mathbb{D}$	Conjunto dos números duais
$\mathbb{H}$	Conjunto dos quatérnion
$\mathbb{H} \otimes \mathbb{D}$	Conjunto dos quatérnions duais

## Subscritos

$tr$	translação
$rot$	rotação

## Sobrescritos

$\bullet^\dagger$	Valor conjugado de um número dual
$\bullet^{-1}$	Valor inverso de $\bullet$
$\bullet^*$	Valor conjugado de um quatérnion
$\ \bullet\ $	Norma de $\bullet$
$r_1^0$	Rotação do <i>frame</i> $\mathcal{F}_0$ para o <i>frame</i> $\mathcal{F}_1$
$p_1^0$	Ponto $p$ representado do <i>frame</i> $\mathcal{F}_0$ para o <i>frame</i> $\mathcal{F}_1$

## Siglas

ROS	Robot Operating System
VCS	Version Controlling System
VREP	Virtual Robotics Experimentation Plataform
DOF	Degrees of Freedom ou Graus de Liberdade
SEA	Series Elastic Actuators
RTPC	Real Time Personal Computer
LTS	Long Time Support
GCC	GNU Compiler Collection
API	Application Programming Interface
SI	Sistema Internacional de Unidades
FTS	Force Torque Sensor
SDK	Software Development Kit
DQ	Dual-Quaternions
LBR	Light Built Robot
iiwa	intelligent industrial work assistant
LARA	Laboratório de Automação e Robótica
FKM	Forward Kinematics ou Cinemática direta
SSH	Secure Shell
DH	Denavit-Hartenberg
VPN	Virtual Private Network

# Capítulo 1

## Introdução

### 1.1 Contextualização

A humanidade sempre sofreu de uma escassez de profissionais nas áreas de saúde quando comparado ao número de pessoas que necessitam da prestação deste tipo de serviço. Além disso, o problema se agrava ainda mais quando há a necessidade de tratamento especializado, tendo em vista que para regiões pobres ou de interior a chance dos centros clínicos e hospitais não possuírem tal recurso é alta, sendo preciso que os enfermos se locomovam até locais capacitados. Contudo esta locomoção pode não ser viável para todos os pacientes, seja por motivos econômicos, seja por suas condições de saúde.

Dessa forma, com o intuito de solucionar parte destes problemas surge como uma alternativa a telemedicina. Esta por sua vez, é um método que possibilita examinar, monitorar e tratar pacientes, ainda que o médico possa se encontrar em outro local. Assim, um dos maiores objetivos da telemedicina é diminuir, ou até mesmo eliminar, deslocamentos desnecessários de pacientes e médicos a partir do transporte não de pessoas, mas de expertise ou informações [1].

Além de diminuir o tempo e os custos por evitar um deslocamento de pessoal, a telemedicina viabiliza um acesso mais rápido a especialistas e também que um maior número de pessoas usufruam destes recursos. Outro benefício é a redução do risco de infecção hospitalar, uma vez que haverá um menor contingente de pacientes em cada hospital. Similarmente, a telemedicina proporciona uma maior cooperação e integração entre pesquisadores e médicos, favorecendo o aprendizado daqueles profissionais que se encontram em zonas fora de centros especializados [1].

Com base nas informações previamente apresentadas a telemedicina possui diversos usos latentes. Sendo os principais: promoção da saúde, prevenção e informação sobre tratamentos e doenças, educação a distância para profissionais da saúde, e telecirurgias. Este último se destaca por ser uma das subáreas da medicina que mais obtém benefícios ao se utilizar de tecnologias como braços robóticos e teleoperação por exemplo.

A telecirurgia é o resultado da união entre procedimentos cirúrgicos e teleoperação de mecanismos para realizá-los. Esta é realizada por um cirurgião capacitado utilizando um console remoto



Figura 1.1: Robô cirúrgico Da Vinci

Fonte: <http://macarthurmc.com/the-davinci-robot-redefining-the-art-of-surgery/>

ao paciente. É necessário equipamento em ambos os locais, sendo que em um ficará o controlador remoto e no outro o robô a ser controlado. Além disso, é preciso uma linha de comunicação segura de forma a não ocorrer ou minimizar falhas na transmissão de dados. Deve haver ainda, para segurança do paciente, uma equipe local disponível e capacitada para prosseguir com a operação em caso de emergência. O teleoperador possui controle sobre câmeras endoscópicas e robôs cirúrgicos, e tanto o cirurgião remoto quanto a equipe de cirurgia local usufruem das mesmas perspectivas visuais.

A área cirúrgica encontra diversos entraves em seu desenvolvimento em consequência de limitações humanas, limitações estas que podem ser atenuadas ao se empregar tecnologias como a telecirurgia. Alguns dos ganhos que podem ser adquiridos são: diminuir o esforço físico necessário, aumentar a precisão dos movimentos e até mesmo mitigar os tremores nas mãos dos cirurgiões.

Contudo, mesmo com todos os benefícios citados previamente ainda há muito a ser aprimorado. Um aperfeiçoamento imprescindível é o de incorporar tecnologias hápticas, de forma a se obter um retorno tátil e cinestésico, dado que quase todos os robôs telecirúrgicos ainda empregam somente retorno visual. Considerando que a palavra cirurgia é derivada do grego *kheirourgia*, a qual pode ser traduzida como "ação de trabalhar com as mãos", e que o tato é um sentido do qual cirurgiões obtêm informações proveitosas para o bom andamento do procedimento cirúrgico, a inclusão do retorno háptico se demonstra essencial para a tecnologia telecirúrgica [2].

Outros pontos que devem ser refletidos são as questões do preço e da segurança dos equipamentos utilizados para a telecirurgia. Por serem usualmente dispositivos altamente especializados, como o robô laparoscópico Da Vinci [3] da Figura 1.1, despende-se largas quantias monetárias para adquirir máquinas capazes de realizar somente um tipo de cirurgia. Ademais, estes robôs habitualmente dispõem de juntas rígidas, ou seja, uma junta que prosseguirá com sua trajetória definida independente dos obstáculos em seu caminho, sendo limitada somente por seus ângulos de abertura e a potência de seus atuadores. Esta característica torna a interação humana com o robô perigosa, havendo o risco de choques de alto impacto em caso de colisões ao longo de sua trajetória.

Atualmente, buscando eliminar tais riscos, robôs complacentes vem sendo propostos para aplicações com interações humano-robô. Eles possuem juntas que devido suas características de construção são capazes de ceder em sua trajetória dependendo da resistência percebida durante a movimentação.

De modo a averiguar uma solução plausível para esses pontos apresentados, este trabalho irá discorrer sobre a possível utilização de um robô manipulador complacente de baixo custo em conjunto com uma interface háptica com retorno de força em cirurgias teleoperadas<sup>1</sup>.

## 1.2 Trabalhos relacionados

Nos subtópicos a seguir serão apresentadas referências e discussões a cerca dos principais temas relacionados a este trabalho. Neste sentido, foram selecionados alguns artigos, livros e até mesmo revistas, de modo a destacar que uma diversidade de temas comumente abordados de forma individual podem ser combinados de uma maneira sinérgica para gerar resultados desejados.

### 1.2.1 Teleoperação/Telecirurgia

A primeira cirurgia robótica foi conduzida em 1985 com o auxílio de um braço robótico chamado Puma 560 para realizar biópsias neurocirúrgicas não-laparoscópicas [4, 5]. Em seguida, surgiram em 1987 as cirurgias minimamente invasivas com as colecistectomias laparoscópicas. Este tipo de cirurgia possui diversos tipos de vantagens para os médicos cirurgiões e pacientes. Dentre as vantagens estão: menor risco de infecção, incisões menores e menor tempo de recuperação da cirurgia. Existem, no entanto, diversas limitações, sendo as mais relevantes as de naturezas técnicas e mecânicas dos equipamentos necessários. Os equipamentos inicialmente usados para estas cirurgias incorriam em perda do sentido háptico, da coordenação mão-olho e da destreza dos cirurgiões. Com o intuito de solucionar estas desvantagens e expandir os benefícios das cirurgias minimamente invasivas, surgiu a telecirurgia.

A telecirurgia é definida como um procedimento realizado em um paciente que não se encontra no mesmo local imediato do cirurgião que o está operando. Entretanto, existem outros conceitos comumente associados com esta palavra, como: consultas clínicas com um médico fora do local (teleconsulta), realizar uma operação cirúrgica com o auxílio e orientação de um especialista em outro local (*telementoring*) e o ato de executar um procedimento cirúrgico a longa distância (cirurgia remota). Destes conceitos apresentados, irá ser desenvolvido neste trabalho aquele inicialmente apresentado, em que o cirurgião não se encontra no local imediato da operação e a realiza por meio de interfaces robóticas e computacionais [1].

Desde o surgimento da telecirurgia foram desenvolvidos robôs especializados para realização de cirurgias como o Da Vinci e o Zeus, ilustrados na Figura 1.1 e 1.2 respectivamente. Além

---

<sup>1</sup>Robôs complacentes, também chamados de robôs flexíveis, são aqueles em que seus atuadores permitem desvios de sua posição de equilíbrio quando sofrem forças externas. A posição de equilíbrio de um atuador complacente é definida como a posição onde atuador não gera força ou torque.



Figura 1.2: Robô cirúrgico Zeus

Fonte: <https://twcroboticsurgery.weebly.com/devices-and-systems.html>

disso, já se é pensado em empregar a telecirurgia para a melhora do sistema de saúde de locais mais carentes [6], fora os casos concretos de telecirurgias a distâncias extremas como os casos transatlânticos [7, 8] ou acesso a regiões remotas via VPN no Canadá [9]. Há também um grande número de cirurgias que foram feitas por telecirurgia [10], sendo possível desta forma, reavaliar as vantagens e desvantagens da telecirurgia atual.

As principais vantagens da telecirurgia são o aumento da destreza e a possibilidade de cirurgias remotas. O aumento da destreza está relacionado com a melhora da acurácia, precisão e resistência física[11]. Enquanto que as desvantagens mais comumente citadas são: alto custo para procedimentos simples [10], alta curva de aprendizado dos equipamentos [12] e, por fim, a crítica mais recorrente, falta de retorno de força [4, 10, 7, 2, 12, 11]. Apesar de todas estas desvantagens ainda existe um otimismo de que a telecirurgia irá deixar de ser uma sub-especialização com foco em cirurgias minimamente invasivas e laparoscopias, evoluindo de maneira a ser fundamental em cirurgias gerais [12]. Ademais, é essencial que, assim como será visto a frente neste trabalho, existam diversos estudos e pesquisas sendo realizados para adicionar o retorno háptico a telecirurgia, tornando-a desta forma, mais segura e intuitiva [10].

### 1.2.2 Manipuladores complacentes

Os robôs cirúrgicos são normalmente desenvolvidos tendo como limitações dos atuadores suas velocidades, acelerações e seus limites de juntas, tendo como exemplos disto os modelos Da Vinci e Zeus. No entanto, para aplicações médicas, por envolver contato direto com seres humanos, é desejado uma maior segurança [13]. Um dos métodos para aumentar esta segurança é aplicar uma certa complacência na atuação dos robôs, a qual surge na forma de *series elastic actuators*(SEA) [14, 15]. O SEA faz uso de um elemento elástico entre o trem de engrenagem e a carga. A Figura 1.3 mostra um exemplo de SEA, em que a deflexão da mola sobre efeito da carga é medida e usada como um sinal de retorno de força para o controle. Ademais, como a mola desacopla a inércia do motor dos elos do robô durante impactos, a segurança do atuador é aumentada.

Estes atuadores complacentes possuem dependabilidade e segurança para uso em ambiente



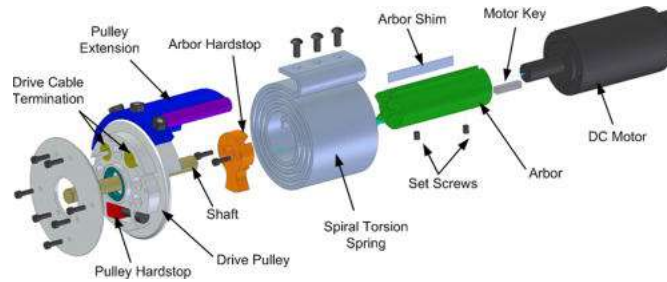


Figura 1.3: Ilustração de um SEA

Fonte: <http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1450219>

populados [16, 17, 18, 19], podendo até mesmo ser utilizado em procedimentos cirúrgicos [20]. Além disso, é possível desenvolver manipuladores complacentes de baixo custo para usos médicos [21, 22], sendo assim, uma solução para um dos problemas da telecirurgia apresentados anteriormente. Essa mudança de paradigma, saindo da precisão, velocidade e performance para a segurança trás consigo novos desafios. tendo como principal desvantagem uma perda na acurácia do posicionamento absoluto do manipulador.

### 1.2.3 Controle de manipuladores com quatérnions duais

O quatérnion dual existe desde 1882 [23], mas não conseguiu tanta atenção da comunidade acadêmica quanto os quatérnions não-duais. Enquanto que a comunidade de computação gráfica ainda não adotou o seu uso por completo, a área da robótica começou a abraçar seus princípios nos anos recentes. Alguns exemplos atuais do uso de quatérnions duais para controle de manipuladores podem ser encontrados nos trabalhos de Pham [24], Adorno [25], Marinho [26], Figueredo [27] e Pereira [28]. Dentre estes, os trabalhos de Adorno, Figueredo e Pereira, são especialmente relevantes pois, além de tratarem de controle de manipuladores com quatérnions duais eles também consideram manipuladores complacentes e espaços de trabalho humano-robô.

Usamos quatérnions duais como uma ferramenta para expressar e analisar as propriedades físicas dos corpos rígidos. Quatérnions duais podem formular um problema de maneira mais concisa, resolvê-lo mais rapidamente e em menos passos, apresentar o resultado mais claramente para outros, ser posto em prática em menos linhas de código e debugado sem esforço. Além disso, não existe perda de eficiência; quatérnions duais podem ser tão eficientes quanto senão mais eficientes do que utilizar o método das matrizes. Em resumo, há diversos motivos para usar quatérnions duais, os quais são resumidos em: livre de singularidades, não ambíguo, interpolação de menor caminho, forma mais eficiente e compacta de representar transformações de corpos rígidos, representação unificada de translação e rotação, pode ser integrado em outros sistema com pequeno esforço de codificação e, informações individuais de translação e rotação podem ser combinadas para produzir um único sistema invariante de coordenadas [29].



Figura 1.4: Tactical Haptics VR Motion Controller da empresa Tactical Haptics

#### 1.2.4 Interface háptica/Feedback de força

Háptico é um termo usado para se referir amplamente às interações de toque que ocorrem para a percepção ou manipulação de objetos. Estas interações podem ser entre uma mão humana e um objeto real; o efetuador terminal de um robô e um objeto real; uma mão humana e um objeto simulado; ou uma variedade de combinações entre humanos e máquinas interagindo com objetos reais, virtuais ou remotos [30]. Deste modo, uma interface háptica é um dispositivo instrumentado, atuado e controlado por computador, capaz de permitir que um usuário toque ou manipule objetos, seja em um ambiente virtual, seja no mundo real através de dispositivos teleoperados [31].

Assim como a interface háptica tem um conceito amplo, amplas também são as suas aplicações, podendo ser aplicada em áreas como, controle remoto de drones [32], telecirurgias [33, 34, 35] e até mesmo na indústria de entretenimento, como é mostrado na Figura 1.4. Dentre estes, o uso de interfaces hápticas vêm sendo muito estudado na área da telecirurgia [36], dado que a capacidade de possuir um feedback háptico é algo muito vantajoso para médicos cirurgiões.

Considerando isto, existem diversas vertentes de aplicação da interface háptica na telecirurgia, como por exemplo: Hamza-Lup [37] analisa a combinação de feedbacks visuais e hápticos, de forma a se obter melhores resultados; enquanto que Kesner [38] emprega a interface de maneira a discriminar a rigidez de tecidos do coração; por fim, King [39] avalia a possibilidade de realizar telecirurgias no espaço, a partir do uso de uma interface háptica.

Maiores informações sobre o funcionamento das interfaces hápticas o trabalho de Diolaiti *et al.* [40] apresenta conceitos como estabilidade, discretização, quantização, delay do tempo e efeito Coulomb nestas interfaces. Enquanto que Samur[31] define as métricas necessárias para avaliar a performance de uma interface háptica.

### 1.3 Objetivos

Este trabalho têm como objetivo avaliar a possibilidade de se unir: a segurança e baixo custo do manipulador complacente Meka A2; um controle de manipulador simples utilizando quatérnions duais e; a multi-utilidade da interface háptica Omega7 para propor soluções para os problemas relatados na secção 1.2.1. Neste trabalho, o foco é na funcionalidade da aplicação sugerida e

não na análise do controle utilizado como em Pereira [28]. Um objetivo secundário é a descrição e desenvolvimento de um setup experimental, de modo a induzir e facilitar a produção de mais trabalhos empregando a interface háptica Omega e o braço manipulador Meka A2.

Desta forma, este trabalho será dividido nos capítulos de Ferramentas de Apoio a Implementação, Metodologia, Resultados e Conclusão. No capítulo de Ferramentas de Apoio a Implementação serão descritos os equipamentos e os motivos destes terem sido escolhidos e usados. Na Metodologia, será descrito o processo no qual o trabalho foi feito, os códigos desenvolvidos e as tentativas e soluções realizadas para problemas encontrados. Ademais, serão listados arquivos e comandos considerados essenciais para o entendimento e continuidade deste trabalho no futuro. Nos Resultados, serão analisados os gráficos e resultados qualitativos adquiridos a partir da realização dos testes descritos no final da Metodologia. Enfim, a Conclusão irá ponderar o cumprimento dos objetivos propostos, classificar os resultados obtidos, conjecturar motivos e soluções para falhas observadas e, considerar os trabalhos futuros.

## Capítulo 2

# Ferramentas de Apoio a Aplicação

Este capítulo contém uma breve explanação sobre cada um dos recursos materiais, computacionais e matemáticos que foram utilizados por todo este trabalho. Esta explicação irá abranger uma sinopse do que são, porque foram utilizados neste projeto e, em alguns casos, informações técnicas mais específicas sobre seus funcionamentos e composições.

### 2.1 Interface háptica Omega

A interface háptica Omega da Force Dimension é uma das mais avançadas para uso em conjunto com desktops comerciais, podendo ser utilizada nos principais sistemas operacionais. Esta apresenta alta precisão em sua capacidade de compreender sua orientação e posição, permite compensação de gravidade, sua estrutura cinemática desacopla movimentos translacionais e rotacionais, permite uma taxa de transmissão de retorno de força de até 4KHz e possui um SDK com todas as funções básicas necessárias já prontas para serem utilizadas [41].

As posições cartesianas do efetuador terminal e sua orientação podem ser obtidas a partir do controlador da interface. As coordenadas são expressas em metros e o sistema que as representam é observado na Figura 2.1. A verdadeira origem do sistema utilizado encontra-se em um ponto virtual no centro do volume de trabalho.

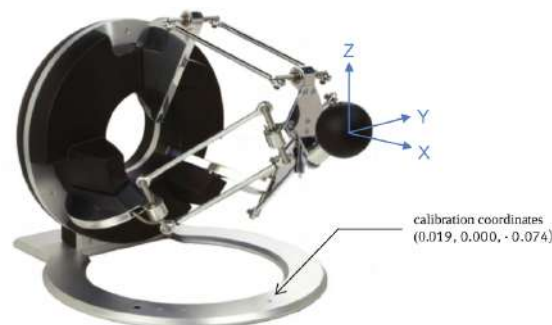


Figura 2.1: Sistema de coordenadas do Omega

O equipamento é desenvolvido de tal forma para que não ocorra deriva na calibração conforme

o tempo passe. Desta forma, é preciso somente que este seja calibrado toda vez que for ligado. Ademais, é importante frisar que a calibração automática do dispositivo o posiciona nas coordenadas observadas na Figura 2.1.

Este dispositivo pode apresentar de 3 até 7 graus de liberdade (DOF), sendo destes, 3 para suas coordenadas cartesianas, 3 para rotação em torno de cada uma destas coordenadas e o último em relação a uma garra utilizada como efetuator terminal, como representado na Figura 2.2.

A interface háptica suporta até 12N de forças continuamente, possui um espaço de trabalho cilíndrico de 160mm de diâmetro e 110mm de largura com resolução linear menor que 0,01mm. Os eixos rotacionais possuem 240, 140 e 180 graus nos respectivos eixos x, y e z com uma resolução de 0,09 graus. Quanto ao gripper, este possui 25mm de abertura, uma feedback de força contínua de até 8N e uma resolução de 0,006mm. Maiores informações técnicas estão disponíveis no Anexo F.



Figura 2.2: Sistema de coordenadas do Omega7

Em virtude de todas suas características, a interface háptica Omega é amplamente empregada em aplicações em que a segurança é um aspecto crítico, como por exemplo aplicações médicas [34, 35] ou até mesmo aeroespaciais [39]. Esta interface háptica foi selecionada para o presente trabalho uma vez que possui precedentes em aplicações médicas e ainda permite um feedback de força para todos os DOFs existentes no sensor de força do braço manipulador utilizado.

## 2.2 Meka A2 Compliant Arm

O Meka A2 da Mekabot é um robô com braços manipuladores complacentes com 7 graus de liberdade (DOF) desconsiderando o efetuator terminal, sendo mais utilizado em ambientes de pesquisa. Este pode dispor de até 2 braços manipuladores ligados em uma base fixa, conforme mostrado na Figura 2.3. Cada braço pesa em torno de 7kg e possui limites de juntas similares aos limites humanos, além disso, sua carga máxima é de 2kg e possui uma precisão menor do que 1cm.<sup>1</sup>

Os atuadores empregados neste braço robótico advêm de uma tecnologia chamada de Series Elastic Actuators - SEA [14, 15], ou seja, atuadores elásticos. Estes, também chamados de atuadores complacentes, estão presentes em todas as juntas do robô e incorporam elementos elásticos

---

<sup>1</sup>Dados retirados de: <<https://github.com/ahorau/mekabot/wiki/Meka-robot-overview>>. Acesso em: em 30 de setembro de 2016.

entre o trem de engrenagem e a carga, como por exemplo uma mola, sendo possível desta forma, medir a deflexão sofrida na mola e utilizá-la como um sinal de feedback de força para controle. Além deste feedback, o Meka também é capaz de medir força, ângulo das juntas, corrente, temperatura e torque dos motores [42]. As informações técnicas do robô e seus atuadores podem ser encontradas no Anexo G.



Figura 2.3: Braço robótico complacente Meka A2

É por possuir esta complacência que o robô apresenta uma maior segurança para humanos, dado que, caso haja um impacto, este será amortecido pelos elementos elásticos. Além de gerar uma maior proteção para pessoas, este também protege o trem de engranagens contra impactos. Por outro lado, esta complacência acarreta em um acréscimo de incerteza na saída da posição das juntas, o braço robótico por sua vez faz uso de encoders absolutos no motor de maneira a tentar mitigar esta incerteza. Apesar de melhorar com juntas complacentes, a segurança fornecida ainda não é o suficiente, por isso, o robô possui mecanimos de segurança por hardware, além de ser possível estabelecer outros destes mecanismos por software.

Visto que o objetivo deste trabalho é vislumbrar a possibilidade de empregar este tipo de robô em cirurgias, o Meka A2 foi selecionado apesar de possuir uma precisão um pouco inferior em relação a outros braços robóticos de uso geral, por dispor de uma maior segurança, e ainda assim, abranger a precisão necessária para algumas funções médicas. Outros benefícios existentes para este trabalho são a fácil comunicação com interfaces em ROS e a arquitetura plug-and-play, a qual permite a fácil utilização e integração de controladores proprietários.

### 2.2.1 Real Time Control PC (RTPC/Mekaman)

Para o uso do manipulador Meka são necessários diversos drivers e bibliotecas específicas, alguns destes podendo ser de difícil instalação e acesso, em contraposição à arquitetura plug-and-play anteriormente citada. Além disto, dado que as funções, scripts e bibliotecas vindos de fábrica do Meka são datados, é aconselhado manter o mesmo versionamento de programas e compiladores de fábrica, evitando desta forma erros deste gênero. Desta forma, para solucionar estes problemas é utilizado um computador exclusivo para se comunicar com o Meka, batizado pela fabricante de Mekaman.

O Mekaman foi adquirido em conjunto com o braço manipulador Meka, já possuindo todo o

conjunto de ferramentas necessárias para a comunicação e uso do braço manipulador. As suas especificações são: sistema operacional Ubuntu 12.04 LTS, ROS Groovy, CMake X, Python 2.7.3 e GCC 4.6.3. Este utiliza uma arquitetura chamada M3RT, que é uma arquitetura de software com programas de tempo real em C++ e scripts de interface em Python. Dentro desta, a principal interface com o braço manipulador é a classe **M3Humanoid**, a qual é uma API que provém o controle de pose e torques do robô a nível de juntas [42].

Ademais, é possível utilizar uma interface gráfica com o Mekaman, dado que este possui uma saída HDMI, no entanto, é necessário utilizar o comando **startx** no terminal após a inicialização do computador, uma vez que este não está configurado para iniciar diretamente para uma interface diferente do terminal. Por fim, empregando o comando **ssh -X meka@ip-adress** no terminal de um computador externo é possível ter acesso ao Mekaman, contudo, para isso é preciso que ambos estejam conectados a mesma rede, além de ter conhecimento do IP e da senha do Mekaman.<sup>2</sup>

### 2.2.2 Force Torque Sensor (FTS)

O sensor de força instalado no braço manipulador Meka A2 é o ATI Mini-40 SI-80-4 que possui 6 graus de liberdade, destes 3 são para as forças e 3 para os torques nos eixos x, y e z. Este sensor é acoplado no fim do braço manipulador antes da montagem final do efetuador terminal como é possível visualizar na Figura 2.4, além disso o Meka A2 já apresenta uma interface para conexão com o sensor em seu antebraço. Com o sistema de controle em tempo real utilizado pelo Meka, os dados obtidos pelo sensor analógico são convertidos para unidades no SI. O alcance de detecção para forças nos eixos x e y é até 80N e para o eixo z até 240N, enquanto que para os torques o alcance é até 4Nm. As resoluções são 0,02N, 0,04N e 0,0005Nm respectivamente [43]. Deve se atentar a possibilidade de se sobrecarregar os sensores com cargas muito altas, resultando em danos permanentes nestes. Os valores de sobrecarga provenientes das especificações são 810N para forças nos eixos x e y, 2400N para forças no eixo z e 19Nm para torques [44].



Figura 2.4: Sensor de força e torque ATI Mini-40 SI-80-4

---

<sup>2</sup>O endereço IP atual do RTPC é 192.168.1.27, é importante lembrar que este endereço está sujeito a mudanças.

## 2.3 Robot Operating System (ROS)

O framework ROS é um conjunto de ferramentas, bibliotecas, e convenções que possibilita simplificar o trabalho, que é desenvolver códigos complexos e robustos através de uma ampla gama de plataformas robóticas. Por possuir um design tão modular e distribuído quanto possível, é factível utilizar somente as partes do framework que serão necessárias, ou úteis, para aplicação desejada. O ROS também dispõe de uma ampla comunidade colaborativa de usuários ativos, a qual coopera de forma a gerar documentações, bibliotecas, tutoriais e ferramentas. Foi fundamentado nesses benefícios que se realizou a escolha de adotar o ROS como framework de comunicação [45].

O ROS tem 3 níveis de conceitos essenciais: o de arquivos de sistema, o de grafo computacional e o de comunidade. Destes, somente os dois primeiros serão explanados, respectivamente, a seguir.

O nível de arquivos de sistema engloba principalmente recursos que são encontrados em disco. Dentre estes, o menor item que se pode construir e lançar neste nível são *packages*. Como verificado na Figura 2.5 *packages* podem conter tudo que seja útil organizar em conjunto, desde bibliotecas a arquivos de configuração. Dentro dos *packages* é possível criar tipos de mensagens e serviços - *msg* e *srv types* - os quais descrevem as estruturas de dados das mensagens e da solicitação e resposta de serviços do ROS respectivamente [45]. Além disso, um grupo de *packages* que possuam o mesmo sistema de versão de controle (VCS) são chamados de repositórios e podem ser construídos ou liberados juntos<sup>3</sup>.

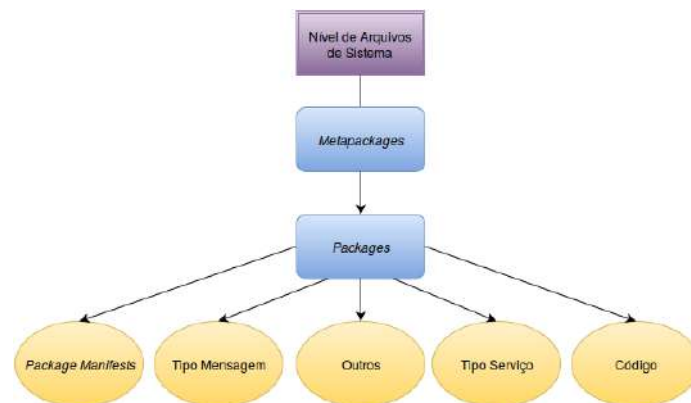


Figura 2.5: Modelo do Nível de Arquivos de Sistema

O nível de grafo computacional é a rede *peer-to-peer*, na qual os processos de ROS tratam os dados e estruturada conforme a Figura 2.6. Este nível tem como base os nós, em que cada um destes controla um processo ou uma ação desejada, por exemplo, um nó realiza o planejamento da rota enquanto outro controla os atuadores. A comunicação da rede é realizada a partir de tópicos nos quais os nós podem publicar ou se inscrever para transmitir ou receber mensagens respectivamente. É importante notar que um tópico pode conter vários nós publicando e inscritos, além disso um nó pode publicar e se inscrever em vários tópicos diferentes ao mesmo tempo. Os nós podem também prestar e requisitar serviços de outros nós, estes serviços são uma comunicação em que um nó requisita determinado serviço e o nó prestador responde com o tipo de mensagem

<sup>3</sup>Texto inspirado no site disponível em: <<http://wiki.ros.org/ROS/Concepts>>. Acesso em: 19 de Julho de 2016.



apropriado. Toda essa comunicação e chamadas são controladas pelo processo chamado *Master*, sem este os nós não seriam capazes nem mesmo de encontrar uns aos outros.

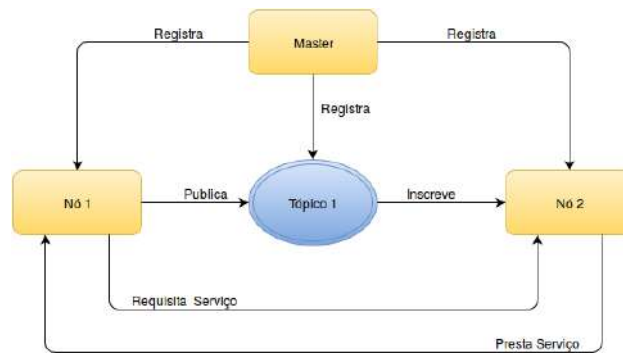


Figura 2.6: Modelo do Nível Grafo Computacional

## 2.4 Catkin

Catkin é o sistema de compilação<sup>4</sup> oficial utilizado pelo ROS. Este combina macros de CMake e scripts em Python para prover mais funcionalidades em cima das já existentes no CMake. O catkin possibilita uma maior facilidade na distribuição de arquivos, melhor suporte na compilação cruzada entre plataformas e portabilidade. E devido a todos estes atributos apresentados e por o catkin poder ser utilizado de forma independente do ROS, isto é, possibilita que seja empregado em outros projetos de forma autônoma, que este sistema foi aplicado no presente projeto.

## 2.5 Virtual-Robotic Experimentation Platform (VREP)

O VREP, assim como seu nome indica, é uma plataforma virtual de experimentação robótica, ou seja, um programa em que se é possível criar, simular e realizar experimentos com diversos robôs ou componentes robóticos. Este simulador robótico é integrado com um ambiente de desenvolvimento que, por possuir uma arquitetura de controle distribuído, permite que cada modelo ou objeto possa ser controlado individualmente. Assim como representado na Figura 2.7 este controle pode ser realizado via script, plugins, nós de ROS, entre outras opções, além disso, estes controles podem ser codificados em diversas linguagens de programação como por exemplo C/C++, Python e Lua.<sup>5</sup>

Devido não só a versatilidade, a simplicidade, a grande quantia de APIs disponíveis, como por ter uma comunicação inerente com o ROS e por possuir uma licença a qual permite o uso gratuito do programa para entidades educacionais como alunos e universidades, é que o VREP foi utilizado neste projeto.

<sup>4</sup>Sistemas de compilação são responsáveis por gerar determinados objetivos, a partir de códigos fonte para o usuário. Estes objetivos podem ser na forma de bibliotecas, executáveis, scripts, interfaces, entre outros.

<sup>5</sup>Texto inspirado no Manual do Usuário do VREP, versão 3.3.1, disponível em: <<http://www.coppeliarobotics.com/helpFiles/>>. Acesso em: 26 de Julho de 2016.

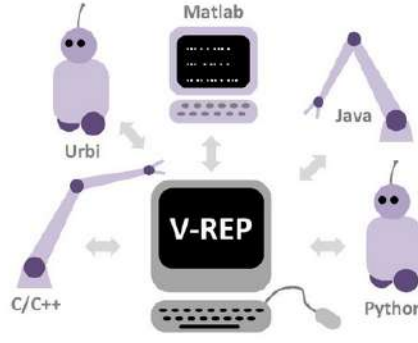


Figura 2.7: Modos de APIs remotas

## 2.6 Fundamentos matemáticos

### 2.6.1 Números duais

Os números duais introduzidos por Clifford [23] são similares aos números complexos que consistem de duas partes conhecidas como real e imaginária. Os números duais consistem em dois componentes que são definidos como a componente primária e a componente dual:

$$z = p + d\varepsilon,$$

onde  $\varepsilon$  é o operador dual,  $p$  é a parte primária e  $d$  a parte dual. O operador dual  $\varepsilon$  é usado da mesma forma que o operador  $i$  dos números complexos, de maneira a distinguir entre a parte primária e dual.

Seja  $\mathbb{D} \triangleq \{p + d\varepsilon | p, d \in \mathbb{R}, \varepsilon^2 = 0\}$  o conjunto dos números duais [27]. Os elementos na forma  $z_i = p_i + d_i\varepsilon$  compostos pelos números reais  $p_i$  e  $d_i$  são ditos números duais se as operações - adição/subtração e multiplicação - e igualdade em  $\mathbb{D}$  são definidos da seguinte forma:

1. A operação de adição/subtração  $\mathbb{D} \times \mathbb{D} \mapsto \mathbb{D}$  é definida como

$$z_1 \pm z_2 = (p + d_1\varepsilon) \pm (p_2 + d_2\varepsilon) = (p_1 \pm p_2) + (d_1 \pm d_2)\varepsilon;$$

2. A operação de multiplicação  $\mathbb{D} \times \mathbb{D} \mapsto \mathbb{D}$  é definida como

$$z_1 z_2 = z_2 z_1 = (p_1 + d_1\varepsilon)(p_2 + d_2\varepsilon) = p_1 p_2 + (p_1 d_2 \pm p_2 d_1)\varepsilon;$$

3. A igualdade no conjunto  $\mathbb{D}$ , isso é  $z_1 = z_2$ , é satisfeita se e somente se  $p_1 = p_2$  e  $d_1 = d_2$ .

Após definidas essas operações iniciais, é possível encontrar a operação de divisão baseando-se no elemento conjugado  $z^\dagger = p - d\varepsilon$ . É importante notar que a álgebra dos números duais não é uma álgebra de divisão, uma vez que números puramente duais não são invertíveis.

Seja  $z_1$  e  $z_2$  números duais com  $p_2 \neq 0$ , a operação de divisão entre eles é definida por:

$$\frac{z_1}{z_2} = \frac{z_1}{z_2} \frac{z_2^\dagger}{z_2^\dagger} = \frac{p_1 p_2}{p_2^2} + \frac{p_2 d_1 - p_1 d_2}{p_2^2} \varepsilon.$$

O inverso do número dual  $z$  é então definido como

$$z^{-1} = \frac{z^\dagger}{z z^\dagger} = \frac{[p - d\varepsilon]}{p^2}.$$

A teoria de números duais pode ser estendida a outros conceitos [46], como vetores e números reais, mas aqui esta será utilizada com ênfase em sua aplicabilidade em conjunto com os quatérnions de maneira a representar translações e rotações.

### 2.6.2 Quatérnions

Quatérnions são uma extensão da teoria dos números complexos, introduzidos por Hamilton em 1866 [47], para formular uma variedade matemática de quatro dimensões. Um quatérnion é definido como:

$$q = w + (x\mathbf{i} + y\mathbf{j} + z\mathbf{k}),$$

onde  $w$ ,  $x$ ,  $y$  e  $z$ , são números reais e  $i$ ,  $j$  e  $k$ , são componentes imaginárias [29]. Estas componentes imaginárias possuem as seguintes propriedades:

$$i^2 = j^2 = k^2 = ijk = -1,$$

e

$$ij = k, \quad ji = -k,$$

$$jk = i \quad kj = -i,$$

$$ki = j \quad ik = -j,$$

Também é comum o quatérnion ser representado como um vetor contendo um componente escalar  $w$  e um componente vetorial  $\mathbf{v} = (x, y, z)$  tal que,

$$q = (w, \mathbf{v}).$$

Sendo  $\mathbb{H} \triangleq \{q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} | w, x, y, z \in \mathbb{R}, i^2 = j^2 = k^2 = ijk = -1\}$  o conjunto dos quatérnions [27], as operações de adição/subtração, multiplicação, igualdade, conjugado e magnitude são definidas por:

1. A operação de adição/subtração  $\mathbb{H} \mapsto \mathbb{H} \times \mathbb{H}$  é definida como:

$$\mathbf{q}_1 \pm \mathbf{q}_2 = (w_1 \pm w_2, \mathbf{v}_1 \pm \mathbf{v}_2);$$

2. A operação de multiplicação por escalar, com  $c \in \mathbb{R}$  é definida como:

$$c\mathbf{q} = (cw, c\mathbf{v});$$

3. A operação de multiplicação  $\mathbb{H} \mapsto \mathbb{H} \times \mathbb{H}$  é definida como:

$$\mathbf{q}_1 \mathbf{q}_2 = (w_1 w_2 - v_1 v_2, w_1 \mathbf{v}_2 + w_2 \mathbf{v}_1 + (\mathbf{v}_1 \times \mathbf{v}_2));$$

4. A igualdade no conjunto  $\mathbb{H}$ , isto é  $\mathbf{q}_1 = \mathbf{q}_2$ , é satisfeita se e somente se  $w_1 = w_2$  e  $\mathbf{v}_1 = \mathbf{v}_2$ .

5. O conjugado de um quatérnion  $\mathbf{q}$  é definido como:

$$\mathbf{q}^* = (w, -\mathbf{v});$$

6. A operação de magnitude de um quatérnion é definido como:

$$\|\mathbf{q}\| = \mathbf{q}\mathbf{q}^*;$$

7. Da norma do quatérnion se obtém a operação de inversão definida como:

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}^2\|}.$$

Caso o quatérnion seja unitário o seu módulo é igual a 1. O quatérnion unitário é muito utilizado para representar a rotação de um ângulo  $\theta$  sobre um eixo unitário  $\mathbf{n}$  em um espaço tri-dimensional como na seguinte equação:

$$\mathbf{q} = \left( \cos\left(\frac{\theta}{2}\right), \mathbf{n} \sin\left(\frac{\theta}{2}\right) \right).$$

Para maior detalhamento sobre quatérnions e suas vantagens é recomendado o artigo introdutório de McDonald[48] para estudantes.

### 2.6.3 Quatérnions duais (DQ)

Ao se combinar quatérnions com a teoria de números duais obtém-se os quatérnions duais, que foram inicialmente apresentados por Clifford em 1882 [23]. Enquanto que o quatérnion simples detém a capacidade de representar rotações, o quatérnion dual consegue representar tanto rotação quanto translação juntas. O quatérnion dual é definido como um número dual tendo quatérnions como componentes:

$$\underline{\mathbf{q}} = \mathbf{q}_p + \mathbf{q}_d \varepsilon,$$

onde  $\mathbf{q_p}$  e  $\mathbf{q_d}$  ambos são quatérnions. Fundindo as operações algébricas associadas com os quatérnions e adicionando o componente dual  $\varepsilon$  é formada a aritmética dos quatérnions duais.

Sendo o conjunto dos quatérnions duais definido como o conjunto

$$\mathbb{H} \otimes \mathbb{D} = \{(\mathbf{w_p} + x_p \mathbf{i} + y_p \mathbf{j} + z_p \mathbf{k}) + (\mathbf{w_d} + x_d \mathbf{i} + y_d \mathbf{j} + z_d \mathbf{k}) \varepsilon | \mathbf{w_p}, x_p, y_p, z_p, \mathbf{w_d}, x_d, y_d, z_d \in \mathbb{R}\}$$

sobre  $\mathbb{R}^8$ , as operações aritméticas necessárias para utilizar quatérnions duais são:

1. Multiplicação escalar

$$\underline{c\mathbf{q}} = c\underline{\mathbf{q_p}} + c\underline{\mathbf{q_d}}\varepsilon;$$

2. Adição/Subtração

$$\underline{\mathbf{q_1}} + \underline{\mathbf{q_2}} = \underline{\mathbf{q_{p1}}} + \underline{\mathbf{q_{p2}}} + (\underline{\mathbf{q_{d1}}} + \underline{\mathbf{q_{d2}}})\varepsilon;$$

3. Multiplicação

$$\underline{\mathbf{q_1}} \underline{\mathbf{q_2}} = \underline{\mathbf{q_{p1}}} \underline{\mathbf{q_{p2}}} + (\underline{\mathbf{q_{p1}}} \underline{\mathbf{q_{d2}}} + \underline{\mathbf{q_{d1}}} \underline{\mathbf{q_{p2}}})\varepsilon;$$

4. Conjugado

$$\underline{\mathbf{q}}^* = \underline{\mathbf{q_p}}^* + \underline{\mathbf{q_d}}^*\varepsilon;$$

5. Magnitude

$$\|\underline{\mathbf{q}}\| = \underline{\mathbf{q}} \underline{\mathbf{q}}^*;$$

6. Condição unitária

$$\|\underline{\mathbf{q}}\| = 1, \text{ e}$$

$$\underline{\mathbf{q_p}}^* \underline{\mathbf{q_d}} + \underline{\mathbf{q_d}}^* \underline{\mathbf{q_p}} = 0.$$

O quatérnion dual unitário é o objeto de foco principal neste trabalho, já que ele consegue representar qualquer rotação e translação de corpo rígido. A translação e rotação rígidas representadas por [29]:

$$\underline{\mathbf{q_p}} = \mathbf{r}, \text{ e}$$

$$\underline{\mathbf{q_d}} = \frac{\mathbf{t} \mathbf{r}}{2},$$

onde  $\mathbf{r}$  é um quatérnion unitário representando a rotação e  $\mathbf{t}$  é um quatérnion representando a translação descrito por  $\mathbf{t} = (\mathbf{0}, \tilde{\mathbf{p}})$ . Este quatérnion pode representar uma rotação pura simplesmente usando a parte dual como zero, assim como também é possível obter uma translação pura como mostrado nas equações seguintes:

$$\underline{\mathbf{q_{rot}}} = \left[ \cos\left(\frac{\theta}{2}\right), \mathbf{n_x} \sin\left(\frac{\theta}{2}\right), \mathbf{n_y} \sin\left(\frac{\theta}{2}\right), \mathbf{n_z} \cos\left(\frac{\theta}{2}\right) \right] [0, 0, 0, 0], \text{ e}$$

$$\mathbf{q}_{\text{tr}} = [1, 0, 0, 0] \left[ 0, \frac{t_x}{2}, \frac{t_y}{2}, \frac{t_z}{2} \right].$$

Com estas definições é possível representar um ponto  $\mathbf{p}$  em um *frame* diferente, considerando os quatérnions puros  $\mathbf{p}^0$  e  $\mathbf{p}^1$ , sendo estes quatérnions em que seu quatérnion primário é zero, como a representação do ponto  $\mathbf{p}$  em relação aos *frames*  $\mathcal{F}_0$  e  $\mathcal{F}_1$ , respectivamente. Se o *frame*  $\mathcal{F}_1$  é obtido rotacionando o *frame* anterior por um quatérnion  $\mathbf{r}_1^0$ , então a representação do ponto no *frame*  $\mathcal{F}_1$ , isto é,  $\mathbf{p}^1$  é dado pela transformação

$$\mathbf{p}^1 = \mathbf{r}_1^{0*} \mathbf{p}_0^0 \mathbf{r}_1^0.$$

Esta transformação é conhecida como *frame rotation* [27].

Empregando o mesmo princípio, mas com uma perspectiva diferente, seja  $\mathbf{p}_0^0$  um ponto em relação as coordenadas do *frame*  $\mathcal{F}_0$ . A transformação que corresponde a rotação do mesmo *frame* obtendo-se um novo ponto é chamado de *point rotation*:

$$\mathbf{p}_1^0 = \mathbf{r}_1^0 \mathbf{p}_0^0 \mathbf{r}_1^{0*}.$$

A diferença entre ambas as transformações é uma questão de perspectiva. No primeiro caso, os observadores localizados no ponto ficarão estacionários enquanto o *frame* se desloca. Enquanto que no segundo caso, os observadores localizados no *frame* se manterão estacionários enquanto o ponto se move [27].

## 2.7 Controlador

Dentre os controladores desenvolvidos e testados no trabalho de Pereira[28], foi selecionado para este trabalho um controlador proporcional por sua simplicidade e facilidade de alterar seus parâmetros de controle, fatores importantes para não gerar dificuldades desnecessárias em áreas as quais não são o foco do trabalho. Além disso, considerando a necessidade de reduzir o tempo de resposta entre a interface háptica e o braço manipulador devido a teleoperação, é vantajoso considerar controladores simples de serem calculados, despendendo desta forma, menos tempo e custo computacional para realizar o controle.

Os códigos utilizados estão na pasta **dq\_robotics**, onde se encontra o controlador escolhido **OrdinaryPoseController** e o modelo cinemático do robô **A2Arm**. Para o uso do controlador é preciso ter como entrada o modelo cinemático do manipulador desejado, a matriz de ganho proporcional de tamanho 8x8, pois estes são os ganhos de cada parcela do quatérnion dual que representa o manipulador, e por fim, um parâmetro de controle chamado de  $\lambda$ . Além disso, se obtém e envia as posições de junta do manipulador utilizando os métodos **getNewJointPositions** e **sendPositions**, respectivamente. Também é possível conseguir os mesmos resultados destes métodos se inscrevendo e publicando nos tópicos **/humanoid\_state** e **/humanoid\_command**.

Destes o primeiro é o tópico empregado tanto pelo Meka quanto pelo VREP para comunicar informações do estado do manipulador, alguns exemplos seriam ângulos, velocidades e esforços das juntas. O segundo tópico por sua vez, permite enviar informações para o controle dos mesmos dados adquiridos pelo `/humanoid_state`.

## Capítulo 3

# Metodologia

Neste capítulo serão caracterizados os passos realizados para o desenvolvimento deste trabalho de forma a facilitar a reprodução do mesmo e, além disso, a compreensão e análise dos dados que foram obtidos. Calibração dos sensores, arquivos e programas essenciais também estão inclusos nas informações referentes a este capítulo. Assume-se que o ROS e o VREP já foram instalados e estão prontos para uso. Todo o trabalho foi realizado no sistema operacional Ubuntu 14.04 LTS (computador pessoal) e no Ubuntu 12.04 LTS (computador exclusivo do MEKA), com as versões do ROS, Indigo e Groovy, e Python 2.7.6 e 2.7.3, respectivamente. Caso se deseje utilizar outro sistema operacional poderá ser preciso efetuar adaptações.

### 3.1 Visão Geral

O sistema desenvolvido neste trabalho é apresentado na Figura X, em que são realizadas transferência e controle de dados em duas vias de comunicação. Uma comunicação é em relação ao envio da referência de posição da interface háptica para o robô, podendo ser este real ou simulado. Enquanto que a outra comunicação é referente as medidas de forças obtidas pelo sensor do robô, simulado ou real, para a interface háptica.

### 3.2 Omega7

Preliminarmente ao manuseio da interface háptica, é preciso que a pasta com os arquivos de driver da interface Omega estejam dentro da pasta *source* do *catkin workspace* que está sendo utilizado pelo ROS. Os arquivos de driver necessários são **libdhhd.a** e **libdrd.a**, ambos os quais podem ser encontrados dentro da pasta **forcedimension\_omega7\_driver/lib**. Após isso ainda é necessário adicioná-la como dependência em todo CMakeLists dos códigos que forem utilizar a interface háptica.

Além dos drivers, a interface háptica possui dois headers para utilizar funções predefinidas, o **dhdc.h** e o **drdc.h**. Estes headers facilitam o uso da interface háptica por disponibilizar funções



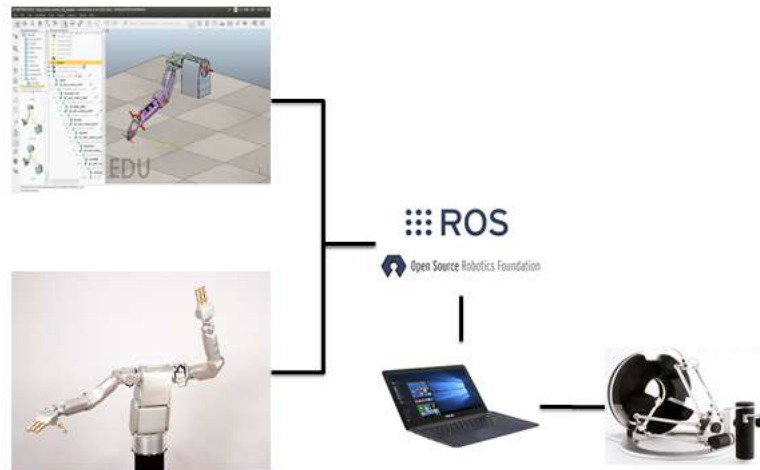


Figura 3.1: Sistema proposto

básicas, como abrir o canal de comunicação, auto iniciar, começar o laço de controle. Ademais possibilita também acesso a dados como posição, orientação e velocidade de todos os graus de liberdade da interface e permite estabelecer as forças e torques utilizadas em cada eixo. Também é factível a movimentação da interface háptica estabelecendo pontos desejados.

Finalizada a parte de drivers e dependências, deve-se realizar a calibração da interface a ser utilizada. Ao se ligar o dispositivo haverá uma luz intermitente branca significando a necessidade de realizar a calibração manual. Para realizar essa calibração da interface háptica são necessários apenas dois passos: primeiro, após ligar a interface, é preciso encaixar o pino branco existente na parte inferior do manipulador da interface háptica no furo existente nas coordenadas de calibração apresentadas na Figura 3.2, e uma vez feito isso, a luz que estava piscando irá começar a piscar em uma frequência menor; segundo, após a calibração dos graus de liberdade referentes a translação, é preciso rotacionar os outros eixos até os dois sensores fim de curso de cada um. Feito isso a luz intermitente deve parar de piscar totalmente e o dispositivo irá estar completamente calibrado e pronto para uso.

Posto que todos os passos anteriores foram executados de maneira correta, o efetuator terminal deve estar pronto para ser utilizado. O último aspecto importante a ser ressaltado, é a necessidade da utilização do comando **sudo** antes da execução de programas que se comuniquem diretamente com a interface háptica, isso é feito para que o sistema possa localizar a interface háptica via USB.



Figura 3.2: Calibração dos eixos de translação da interface háptica

### 3.3 VREP

Antes de iniciar o VREP, é fundamental que as pastas **vrep\_common** e **vrep\_plugin** estejam no *catkin workspace* empregado, além disto deve-se iniciar o ROS com o comando **roscore** em um terminal a parte. Em seguida inicializa-se o VREP com o comando **./vrep.sh** em outro terminal, este terminal deve estar na pasta onde se encontra instalado o VREP. Após a tela inicial do VREP surgir, é aconselhável verificar se o plugin do ROS foi carregado com sucesso no terminal previamente operado. Também é possível aferir o sucesso da conexão entre o ROS e o VREP usando o comando **rostopic list**, este lista todos os nós de ROS ativos. Caso o VREP apareça entre os nós, então isto indica que é possível se comunicar com o VREP a partir do ROS.

#### 3.3.1 Modelo Kuka

A princípio foi empregado o modelo base existente no VREP do braço manipulador industrial Kuka apresentado na Figura 3.3. Isso foi feito devido a falta da existência de um modelo pronto do Meka A2 no VREP, sendo assim preciso construir tal modelo para somente então utilizá-lo. O modelo Kuka LBR iiwa 7 R800 foi posto em uso como substituto por possuir o mesmo número de graus de liberdade que o Meka A2, no entanto seus 3 últimos DOFs não alteram somente a orientação de seu efetuator terminal como ocorre no Meka. Considerando isto, é aplicado apenas a movimentação translacional da interface háptica no Kuka.

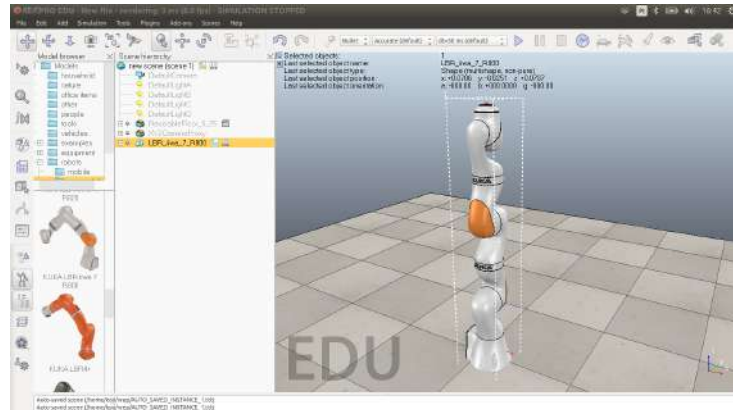


Figura 3.3: Modelo Kuka LBR iiwa 7 R800 no VREP em pose de descanso

Dessa forma já é viável a movimentação do braço manipulador, mas ainda é necessário adicionar outros dois componentes para que se obter o feedback de força, o sensor de força e o gripper. O primeiro componente fundamental é o sensor de força, o qual será utilizado para obter os dados necessários para o feedback de força. Após introduzir o sensor na cena do VREP, este deve ser deslocado até a posição do último link responsivo do manipulador e incorporado a este mesmo link na hierarquia de cena. O sensor de força ainda precisa de outro componente conectado a ele, caso contrário ele não atuará da maneira desejada retornando somente os valores limites em cada eixo. A fim de solucionar isto acopla-se um gripper RG2 igual a Figura 3.4, já disponível no VREP e teve sua escolha como principal fator a semelhança com o efetuator utilizado no manipulador

Meka A2 disponível no LARA.

Os dados obtidos a partir do sensor de força são relativos a orientação do mesmo, ou seja, a força no eixo x observada é na verdade uma força no eixo x da orientação do sensor de força. Esse detalhe pode causar certo desconforto ao se transferir a força obtida do VREP diretamente a interface háptica, dado que não é intuitivo que a orientação de sua mão influencie na direção da reação sentida. Um simples exemplo seria que, ao dar um soco em uma parede, sua mão sentisse uma força para baixo caso fosse o eixo z que estivesse no eixo de reação da força. A partir desta analogia é possível induzir que seres humanos tem a tendência a pensar por uma orientação relativa diferente de suas mãos, esta em muitos casos é a orientação de sua cabeça ou do tronco de seu corpo, sendo que estes podem estar fixos durante a movimentação das mãos.

Considerados estes pontos uma solução simples utilizada para o modelo Kuka foi reposicionar o sensor de força para que sua orientação fosse igual a desejada, já que para este modelo não é realizada mudança da orientação durante a movimentação do robô. Com isso o sensor sempre estará na posição considerada intuitiva para o usuário. Um ponto importante a se atentar é que a orientação do sensor deve estar igual a desejada para a pose em que ele será manuseado e não a pose de descanso, um exemplo de cada seriam as Figuras 3.4 e 3.3 respectivamente.

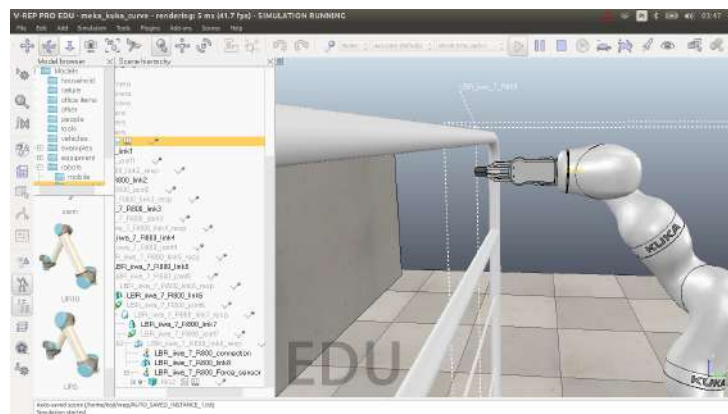


Figura 3.4: Simulação com o Kuka na pose de trabalho

### 3.3.2 Controle háptico

Antes de prosseguir com o uso da interface háptica é recomendado tomar algumas precauções iniciais. A primeira é em relação ao limite da força a ser aplicada na interface háptica, apesar da interface desconsiderar forças acima de seu limite ainda assim é provável que, caso ela seja empregada perto de seu limite durante um longo período de tempo possa vir a ocorrer danos em sua estrutura. É então sugerido que se insira um limiar virtual de forma a delimitar os valores de força a serem enviados para a interface. Neste trabalho, o limite de força determinado é de 8 N, e esse valor foi determinado considerando-o a maior força contínua possível sobre o atuador do gripper em qualquer um dos eixos. Esse valor é suficiente para impedir a movimentação da interface háptica pela mão do usuário em situações normais.

Uma vez que muitas das ações que um braço manipulador pode executar têm a possibilidade

de ultrapassar esse limite, é então aconselhado realizar uma escala das forças. Essa escala pode ser feita por uma constante multiplicando todo valor de força obtido, podendo assim abranger todo o espectro de força que se deseja sentir em até 8N. Neste trabalho, essa constante será chamada de constante de força, deve ser ajustada a depender do braço manipulador a ser utilizado pois é influenciada por fatores como a massa e potência dos atuadores de cada manipulador.

Outro fator relacionado ao feedback de força que, assim como os anteriores, pode tanto interferir na sensação háptica obtida quanto danificar a interface, é uma grande variação de força em um curto período de tempo. Uma alta taxa de variação de força certamente reduzirá a qualidade do retorno háptico e o conforto no uso do dispositivo, além disso um alto ganho de força instantâneo pode acarretar em danos a estrutura do dispositivo. É então conveniente estabelecer um limite no aumento da força a cada iteração do programa para desta forma proteger o usuário e o dispositivo. Também é relevante mencionar que em muitos procedimentos médicos a sensação háptica é um fator mais qualitativo que quantitativo, e consequentemente os valores dessas constantes podem ser alterados de forma a melhor atender a cada usuário e equipamento utilizado.

Além destes controles de segurança existe, do mesmo modo, a possibilidade de se empregar o controle háptico para prestar assistência aos movimentos do usuário. Isso pode ser feito de forma passiva, delimitando um espaço de trabalho no qual a interface pode se locomover, ou de forma ativa, onde a interface possui diretrizes mas se adapta conforme o movimento do usuário. Neste trabalho foi implementado somente a forma passiva de maneira a manter o usuário dentro de um espaço de trabalho desejado, podendo funcionar alternativamente como um limite virtual das juntas do manipulador.

Foram então realizados testes na cena da Figura 3.4 para verificar o funcionamento adequado da translação e do feedback de força, e além disto foram determinadas valores para constante de força e o limite de variação desta, de modo que o uso da interface háptica para o modelo Kuka fosse adequado.

### **3.3.3 Modelo Meka**

Logo que o modelo VREP do manipulador Meka A2 foi disponibilizado pelo LARA, foram realizadas as adaptações necessárias para a utilização deste com a interface háptica. Este modelo referente a Figura 3.5 foi desenvolvido por um aluno de estágio do próprio LARA, aumentando a semelhança entre a simulação e o real. Ao se experimentar no mesmo modelo do robô real se agrega mais valor aos dados adquiridos via simulações, isso propicia também descobrir condições que causem ou facilitem danos no robô podendo assim evitá-las ao utilizar o manipulador real.

Todos os passos para adicionar o sensor de força e o gripper RG2 que foram realizados no modelo Kuka foram refeitos para o modelo Meka. Após isto, foram realizados testes para determinar a constante de força a ser utilizada, como o Meka possui uma massa muito inferior ao Kuka e seus atuadores possuem uma potência muito inferior, a constante de força obtida para o Meka foi muito mais próxima de 1. Devido a isto foi possível aferir a existência de uma força antes oculta, após muitos testes e análises foi constatado que esta força se trata do peso do efetuador terminal no

sensor de força. Além disso, o valor desta força tende a sofrer variações maiores conforme o braço manipulador se movimenta mais rapidamente, sendo deste modo, aconselhado o uso em baixas velocidades de movimentação.

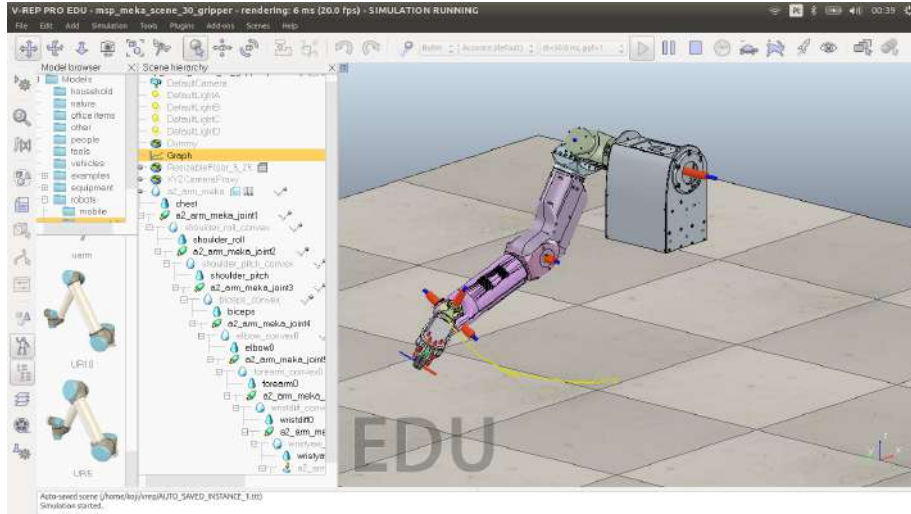


Figura 3.5: Modelo Meka em sua pose inicial no VREP

Na condição de uso apenas da translação e em baixas velocidades, este último não é um problema dado que a natureza da aplicação médica necessita de mais precisão e cuidado do que somente rapidez, este peso pode ser contrabalanceado empregando um offset do valor médio das forças adquiridas pelo sensor de força quando este estiver parado em sua pose inicial.

No entanto, este método não é capaz de solucionar o problema ao se adicionar os 3 últimos graus de liberdade, referentes a rotação da orientação do efetuador terminal. Nessa situação há então uma força que pode variar sua magnitude em cada eixo do sensor de força, inviabilizando desta forma a solução anterior. Outro problema ressurgue ao se incorporar a rotação do efetuador terminal: o da falta de intuitividade entre o movimento realizado e o retorno háptico sentido. Ambos os problemas tem natureza similar e estão relacionados a não serem representados por uma orientação fixa no espaço.

Antes de prosseguir para as soluções dos problemas acima experimentadas é fundamental citar que, todas as soluções partiram do pressuposto que o quatérnion dual utilizado para representar o Meka seria igual ao quatérnion dual do sensor de força, ou seja, que estes estariam na mesma posição e com a mesma orientação. Baseando-se nesse pressuposto como verdade, foram realizadas tentativas de encontrar a solução empregando *frame rotations* com quatérnions da subsecção 2.6.3.

Apesar das bibliotecas utilizadas realizarem operações de quatérnions duais, foi possível utilizá-la para multiplicar quatérnions não-duais já que estes podem ser representados como um subgrupo dos quatérnions duais cuja parte dual é igual a zero. Outro ponto importante na utilização dos quatérnions, é o fato que os quatérnions gerados a partir dos dados obtidos do sensor de força são quatérnions puros, isto é, suas partes reais são iguais a zero. Além disso, uma das características dos quatérnions puros é que, a sua multiplicação tem como resultado um quatérnion puro [27]. Esta é uma característica importante pois, a força em cada eixo está sendo representada por equivalente

imaginário no quatérnion, sendo assim é preciso que o valor real sempre seja zero para que se possa manter a relação entre as forças nos eixos e os valores imaginários no quatérnion.

Para entender as possíveis soluções experimentadas é primeiro preciso ter conhecimento da maneira a qual está sendo realizada o controle de posição do manipulador. Dado uma base fixa no espaço como referência, é então obtido o quatérnion dual, o qual representa a ponta do efetuador terminal na pose inicial desejada. É com base neste quatérnion que foram feitas então composições de translação e rotação para se obter uma pose atual qualquer. Como a translação do efetuador terminal não influencia tanto neste problema é possível ignorá-la, tendo assim que se trabalhar somente com as rotações como observado na Figura 3.6.

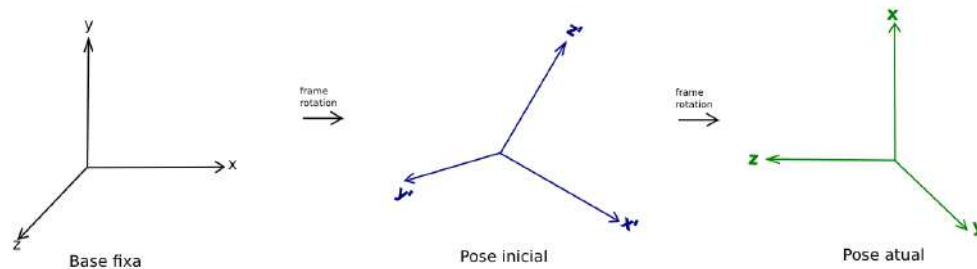


Figura 3.6: Composição de frames rotations do Meka

Dispondo da Figura 3.6 como exemplo, as soluções serão explicadas empregando a mesma nomenclatura para cada etapa.

Primeiramente tentou-se aplicar um offset na orientação da base e utilizando-se das rotações de frames para cancelar o efeito do peso no sensor de força da pose atual. No entanto, não se era conhecido o valor do peso na orientação da base e, mesmo se este fosse conhecido, isto ainda não solucionaria o problema do retorno háptico com orientações variadas.

Após isto foi obtido o valor do peso na pose inicial, para então utilizá-lo como offset a ser rotacionado juntamente com a orientação do efetuador terminal. Infelizmente não se obteve sucesso, mesmo desconsiderando o problema do retorno háptico.

Buscando solucionar ambos os problemas simultaneamente, foi experimentado realizar uma rotação inversa, ou seja, passar da pose inicial para base. Isto pode ser feito somente alternando a posição do quatérnion de rotação com o seu conjugado no momento de realizar a rotação de frames. O motivo por detrás dessa ideia advém do mesmo princípio da primeira solução aplicada, isto é, o peso na orientação da base seria sempre constante. Executando então a rotação inversa da pose inicial para se obter o offset do peso, seguida da rotação inversa da pose atual e por fim subtraindo um do outro, deveria se encontrar um valor próximo de zero caso o manipulador não estivesse sobre o efeito de mais nenhuma força, no entanto, isto não ocorre.

Para se entender o porque das soluções não terem conseguido pelo menos anular o peso do efetuador terminal, é preciso retornar à uma das premissas iniciais. A premissa de que a orientação do quatérnion dual representando o Meka no simulador seria igual ao do sensor, sendo então possível representá-lo pelo mesmo quatérnion dual, contudo essa premissa é falsa. A posição e orientação do efetuador terminal do Meka simulado e do sensor de força não são exatamente as mesmas,

deixando claro então, o motivo pelo qual não foi possível solucionar os problemas.

Descoberto o motivo pelo qual as rotações de frames não funcionavam resolvê-lo foi simples. Para tal era necessário somente obter o quatérnion que representasse o sensor de força, sendo que o ROS já possui tal função. Tendo este quatérnion e aplicando as rotações inversas foi quando os problemas de peso e orientação foram finalmente solucionados.

### 3.3.4 Programação Robot Controlling Node

A fim de se operar o braço manipulador no VREP com a interface háptica foram desenvolvidos dois programas principais a serem executados concomitantemente, o **Robot Controlling Node** e o **Omega7-ROS Interface**. O primeiro destes programas realiza o controle da pose do robô no sistema apresentado na Figura 3.7, em que a posição e orientação espacial obtida na interface háptica é inserida no controle utilizado pelo programa no computador usado, sendo esta parte do processo ilustrado de maneira geral na Figura 3.8 e detalhado no Apêndice A. Em seguida, a saída deste programa é enviado via ROS para o ambiente de simulação VREP ou o manipulador real.

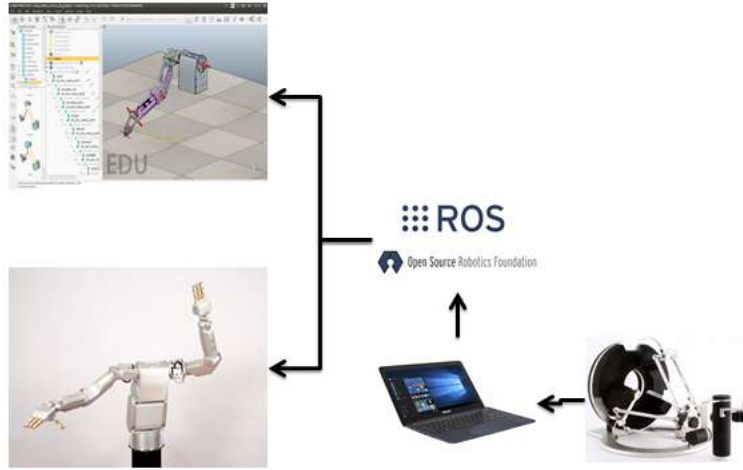


Figura 3.7: Sistema proposto para uso do Robot Controlling Node



Figura 3.8: Fluxograma simplificado do Robot Controlling Node

Detalhadamente, o programa começa com a inicialização de variáveis e objetos, definindo-se o ganho,  $\lambda$  e a cinemática do robô em quatérnions duais, este utilizando as funções **A2ArmKinematics** ou **KukkaKinematics** a depender do robô. Essas variáveis são então enviadas para definir o controlador pelo construtor do objeto tipo **OrdinaryPoseController**, este gera um simples controle proporcional para a posição do efetuador terminal. A partir de um modelo cinemático direto, do

inglês Forward Kinematic Model (FKM), de um vetor com os valores de cada junta da pose inicial é gerado o quatérnio dual desta pose, este é o DQ que será usado como base para compôr o movimento do efetuador terminal. Por fim inicializa-se o nó de ROS a 120Hz.

É definido um loop controlado por um handler de sinais do sistema operacional, sendo necessário usar o comando **CTRL+C** no terminal do programa para finalizá-lo. Obtêm-se as posições e orientação da interface háptica pelo tópico `/omega7/jointstate` empregando o método **getJointPositions**. Este método tem como entrada um vetor do tipo **VectorXd** de tamanho igual ao número de juntas da interface háptica, alterando o valor deste vetor para a posição de cada uma das juntas.

As 3 primeiras posições obtidas pelo tópico são referentes a posição cartesiana do efetuador terminal, as próximas 3 são referentes a orientação, enquanto a última é do espaço de abertura do gripper. Primeiramente é posto em uso somente as posições cartesianas, para tal, é determinado a posição inicial da interface háptica como sendo a mesma da Figura 2.1. Para transformar este ponto em nosso zero virtual emprega-se os valores de x, y e z visualizados na Figura 2.1 como offset. Após obtido o zero virtual pode ser realizado ainda uma escala em relação ao deslocamento real da interface háptica para o deslocamento do braço manipulador, isto torna o sistema robô-interface capaz de alterar o espaço de trabalho do manipulador conforme necessário. Podendo ir de *workspaces* menores, para maior precisão no controle da posição e do feedback de força, até o maior espaço cilíndrico contido dentro do *workspace* do manipulador.<sup>1</sup>

Então estes novos valores de posição são aplicados em conjunto com a equação de translação pura da subsecção 2.6.3 para se obter o quatérnio dual que representa esse movimento, os valores deste quatérnio dual são, 1 na parte real do quatérnio primário e os novos valores x, y e z da translação divididos por dois na parte dual imaginária -i, k e j, respectivamente, sendo o resto dos valores iguais a 0. O valor no eixo x é invertido o sinal e os eixos y e z são alternados para que os movimentos realizados na orientação da interface háptica sejam corretamente efetuados na orientação da base fixa do manipulador do Meka, esse pode ser observado na Figura 3.6. Por fim é feito a composição entre o DQ da pose inicial e o da translação para se obter o quatérnio dual que representa a pose desejada, esta ainda sem rotações em relação a sua orientação inicial.

Este quatérnio dual é enviado em conjunto com o ângulo atual das juntas do manipulador para o controlador, este por sua vez determina o movimento para sair da pose atual até a pose desejada. Finalmente envia-se os novos ângulos de junta para o manipulador.

Outro modo de visualizar o que ocorre é a partir da Figura 3.9, onde o nó `/robot_controlling_node` se inscreve e recebe mensagens do tópico `/omega7/jointstate`, as posições recebidas da interface háptica, e publica no tópico `/humanoid_command` as novas posições de junta desejadas para o robô. O mesmo nó `/robot_controlling_node` está inscrito neste tópico e ao verificar que há novos dados o método de *callback* desse *subscriber* publica essas informações em outro tópico específico para o VREP. Por fim, o VREP lê estes tópicos e altera as juntas do manipulador conforme os dados lidos, o simulador também publica os dados de junta atuais no tópico `/humanoid_state`, o qual é utilizado pelo `/robot_controlling_node` para obter as posições de juntas atuais.

---

<sup>1</sup>Essa limitação se deve ao próprio espaço de trabalho da interface háptica Omega7.



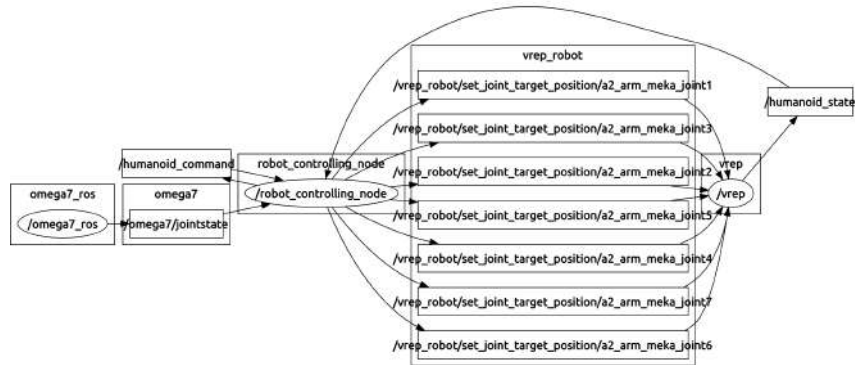


Figura 3.9: Gráfico de comunicação dos nós do ROS durante a execução dos programas

### 3.3.5 Programação Omega7-ROS Interface

O segundo programa desenvolvido para este trabalho funciona como uma interface entre a interface háptica Omega7 e o ROS no sistema apresentado na Figura 3.10. Neste sistema, é inicialmente obtido os valores de força bruto a partir dos sensores de força ou da simulação no VREP ou do robô real. Em seguida, os valores são enviados via ROS para o computador utilizado e então manipulados conforme o fluxograma simplificado da Figura 3.11 ou detalhados nos Apêndices B e C, em que o primeiro representa o programa como um todo e o segundo um destrinchamento da função de controle do feedback de força. Por fim, estes valores são usados como retornos de força na interface háptica.

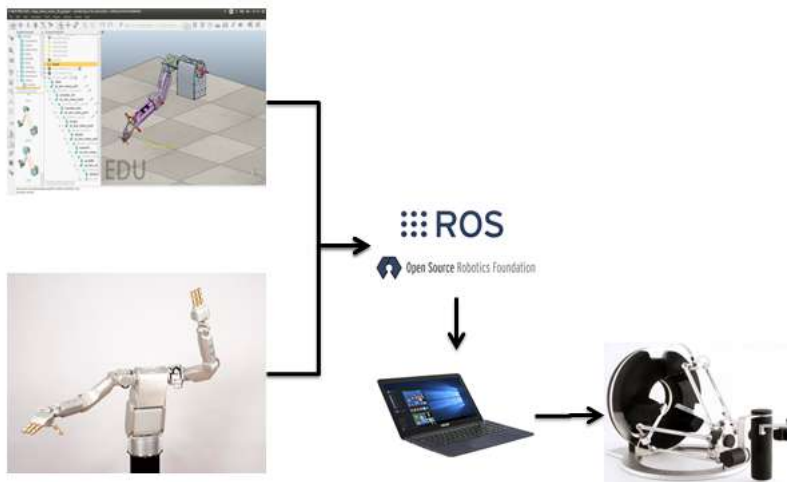


Figura 3.10: Sistema proposto para uso do Omega7-ROS Interface

Detalhadamente, inicialmente é executado um método de auto inicialização da interface háptica, em que este irá se conectar ao primeiro dispositivo Omega disponível, abrindo assim a porta serial USB para a comunicação com tal dispositivo e se posicionando na posição inicial representada na Figura 2.1.

Em seguida é verificado se a execução atual trata-se de uma simulação com o VREP ou uma execução no robô real. Para isso, é utilizado um `#ifdef` pois, o modo de se conectar ao robô em

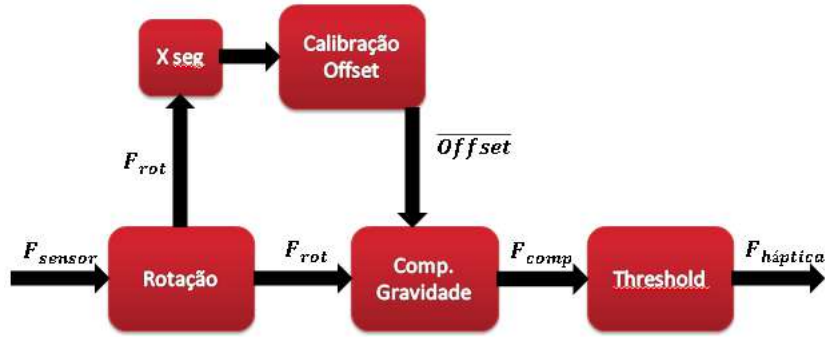


Figura 3.11: Fluxograma simplificado do Omega7-ROS-Interface

cada caso é diferente e certas bibliotecas só podem ser compiladas dentro do computador pessoal utilizado pelo Meka. Caso a execução seja uma simulação, o programa irá ativar clientes para serviços prestados pelo nó do VREP. Estes clientes são referentes aos serviços **simRosGetObjectHandle**, o qual é empregado para se obter o handle do objeto sensor de força dentro do VREP. Esse handle é necessário para o uso de outros serviços como o **simRosReadForceSensor** e **simRosGetObjectPose**. Destes o **simRosReadForceSensor** serve para obter as forças e torques nos eixos x, y e z, enquanto que o **simRosGetObjectPose** retorna a posição x, y, z, e um quatérnio o qual representa a orientação do objeto, esta orientação é constituída de 4 valores, x, y, z e w, em que os 3 primeiros representam a parte imaginária do quatérnio e w a parte real. Após isto, se obtêm a posição, orientação e velocidade da interface háptica, sendo utilizado somente os dois primeiros. É então novamente verificado se é ou não uma simulação, caso o seja é executado o método descrito pelo Apêndice C.

Este controle háptico foi desenvolvido seguindo os passos discutidos na Subsecção 3.3.2, com adição do controle de múltiplas colisões simultâneas em que, caso ocorra do sensor de força captar altos valores de força em todas as direções, o feedback de força é então desativado para possibilitar novamente a movimentação da interface háptica e, desta forma, a movimentação do manipulador. Um ponto importante a ser citado é o fato de que se adquire o valor médio de força de 100 iterações na pose de trabalho antes de ser permitido o retorno de força para a interface háptica, isto serve como o valor de offset para a compensação do peso do efetuador terminal. Por fim, é publicado os valores da posição e orientação no tópico `/omega7/jointstate` como observado na Figura 3.9 e o laço só é finalizado caso haja um sinal de **CTRL+C**.

### 3.4 Braço Manipulador Meka A2

Preliminarmente ao uso do Meka e a realização dos testes, é fundamental averiguar o funcionamento adequado do robô, seu sensor de força, os códigos e os scripts que serão utilizados. Para isto, é necessário inicializar a interface M3RT de controle do robô, o ROS, executando respectivamente os códigos **m3rt\_server\_run -m** e **roscore**, via SSH ou diretamente no Meka-man. Depois, para se comunicar com o manipulador via ROS é obrigatório a execução do nó **shm\_humanoid\_controller**. Este, por sua vez, é o nó da interface **M3Humanoid** que pu-

blica as informações de controle do robô. Após este setup inicial os scripts e programas que se comunicam via ROS ou diretamente com o Meka já podem ser executados.

### 3.4.1 Sensor de força

Mesmo após a realização deste setup inicial, é essencial explorar como usufruir do sensor de força e, caso seja necessário, recalibrá-lo. Visto que este é imprescindível para o feedback de força da interface háptica, é então feito um estudo sobre as APIs, arquivos de configuração e de comunicação para a obtenção dos dados adquiridos por ele.

Destes, o mais importante são os arquivos de configuração, pois sem estes, a interface M3RT não é capaz de encontrar e se comunicar com o sensor de força. Caso estes arquivos não sejam encontrados, o sensor de força **m3loadx6\_ma26\_10** não irá aparecer entre os componentes carregados após a execução do comando **m3rt\_server\_run -m**. Os arquivos de configuração se encontram no endereço: **sftp://mekaman/home/meka/meka\_bot/m3ene/robot\_config/ma26** do Mekaman e possuem os seguintes nomes **m3loadx6\_ma26\_10.yml** e **m3loadx6\_ec\_ma26\_10.yml**. Destes, o primeiro está relacionado a calibração do sensor de força, enquanto que o segundo relaciona-se a leitura de dados brutos do mesmo.

As APIs utilizadas foram as seguintes: **m3qa\_calibrate** e **shm\_loadx6\_publisher**. Sendo o primeiro um script em python responsável pela calibração dos sensores do robô, enquanto que o último é o nó encarregado de obter os dados do sensor de força e publicá-los no ROS. Além disso, este nó existe e pode ser executado tanto em C++ quanto em Python.

Ao rodar o programa de calibração será solicitado a escolha do sensor que se deseja calibrar, sendo o sensor de força representado pelo número 17 e o nome **loadx6**. Em seguida, é requisitado qual tipo de ação deseja-se realizar, tendo como opções: **display sensors (ds)**, **sensor analyze (sa)**, **zero load cell (zl)**, **calibrate load cell (cl)** e **zero load cell with payload (zp)**. Das quais as duas primeiras opções correspondem respectivamente a mostrar e analisar os dados do sensor de força, já as últimas três opções são referentes a calibração estática do sensor de força.

Ademais, todas as opções relacionadas a calibração do sensor de força alteram os arquivos de configuração do mesmo durante o processo de salvar a nova calibração. De tal forma que, caso ocorra algum erro durante a execução que acarreta na finalização indevida do mesmo, o arquivo de configuração será deletado.

Dentre as opções de calibração possíveis o **zl** e o **zp** alteram somente o valor de referência do zero para o sensor, respectivamente com ou sem um bias devido a uma carga associada. Enquanto que a função **cl** utiliza 4 pesos diferentes para gerar uma reta via regressão linear, alterando assim o ponto inicial, a escala e o ângulo da reta de calibração do sensor. Sendo assim possível de se alterar os parâmetros de calibração conforme desejado.

Tendo em vista todas as informações apresentadas acima e que o arquivo de configuração do sensor de força tinha como data de última atualização o ano de 2013, foi considerado oportuna a possibilidade de realizar uma nova calibração e documentá-la para reproduções futuras. Desta forma, foi empregado o método descrito no guia do usuário do sensor de força.

O guia do usuário pede para que seja retirado o efetuador terminal e a placa de conexão ligada a este. Após isto, executa-se o **m3qa\_calibrate**, em seguida seleciona-se o número do sensor de força e então a função **calibrate load cell**. É posteriormente solicitado que se posicione o sensor em uma configuração de carga igual a 0, 500, 1000 e 2000 gramas respectivamente. Dado que o sensor continua acoplado no braço manipulador, a forma mais simples de se conseguir isto é segurando manualmente o antebraço apontando para cima, alinhado com a gravidade.

Desta forma, fabricou-se então 3 pesos conforme os valores requisitados, os quais foram medidos em uma balança de 1 grama de resolução, conforme é possível observar no Apêndice D. Além disso, alterou-se o script **meka\_demo\_a2r**, o qual faz o braço manipulador ir para uma pose desejada, fazendo o manipulador ir para a posição de calibração sozinho. No entanto, é necessário tomar alguns cuidados extras como dar suporte ao braço caso o mesmo ameace descer, isso pode ocorrer dado que o limite de peso suportado pelo Meka é em torno de 2kg. Apesar de todas as incertezas e erros associados ao processo de calibração, eles possuem exatidão e precisão suficiente para uma aplicação com um enfoque mais qualitativo igual ao deste trabalho. Ao finalizar a calibração, dois gráficos como os do Apêndice E serão apresentados na tela do computador, os quais deverão ser fechados antes de finalizar o programa para evitar mal funcionamentos. Por fim, será então questionado se a nova calibração deverá sobrescrever a antiga ou não.

Posteriormente foram realizados testes para verificar qual seria a combinação de calibrações com maior precisão e exatidão utilizando as funções **zp**, **zl** e **cl**. Inicialmente as combinações utilizadas foram: sem calibração, somente **cl**, somente **zl**, somente **zp**. Após essa primeira bateria de testes verificou-se que o melhor resultado obtido foi com a calibração **cl**. Em seguida, testou-se a possibilidade de utilizar as funções **zl** e **zp** em conjunto com o resultado da **cl**. Ao se comparar os resultados obtidos com o valor conhecido dos pesos da garra, constatou-se que a função **zp** não apresentava alterações visíveis em qualquer resultado, enquanto que a melhor combinação obtida foi realizar a **cl** seguida da **zl**, finalizando assim todo o processo de calibração do sensor de força.

### 3.4.2 Modelo cinemático

Durante a realização de uma bateria de testes para avaliar o resultado obtido a partir do procedimento de calibração, foi observado forças de origem desconhecida em determinadas poses do manipulador. Em seguida a uma inspeção mais meticulosa encontrou-se a causa destas forças como sendo uma auto colisão devido ao atingimento do limite mecânico das juntas, as quais ocorreram especificamente nas últimas juntas de orientação antecedendo o efetuador terminal. Estas juntas, por sua vez, estavam sendo forçadas pelo controle a exceder seus limites físicos de junta, o que acarreta na produção de forças internas ao manipulador que serão propagadas para a interface háptica via o sensor de força.

No que concerne o controle, não é interessante dirigir o manipulador a poses em que as suas juntas são submetidas a esforços não desejáveis. Isto ocorreu porque, como explanado na seção 2.7, o controlador proporcional empregado lida com a movimentação do robô de forma a conservar a orientação inicial fixa. Esta restrição na orientação do braço manipulador faz com que o controlador entre em conflito com a orientação imposta pela interface háptica conforme a seção 3.3.4, tendo

como consequência o controlador considerar a nova orientação como uma perturbação no sistema e agir de forma a compensá-lo, o que ocasiona a alteração de todas as posições de juntas, inclusive as de orientação que estavam sob controle da interface háptica, além de sobrecarregar as demais juntas por tentar distribuir o esforço de correção para elas e não conseguir manter a posição desejada inicialmente.

Desta forma, percebe-se que a descoberta deste erro tornou fundamental o ato de reavaliação do controle e do modelo cinemático aplicado neste trabalho. Inicialmente buscou-se uma possibilidade de solucionar o problema a partir de modificações na matriz de ganhos do controlador proporcional, tentativa a qual não obteve sucesso dado que os valores nesta matriz não estão correlacionados diretamente as juntas, e sim, ao quatérnion dual que representa a pose do manipulador, não existindo desta forma, uma fácil visualização entre as alterações nos ganhos dos componentes do quatérnion e o que isto resulta na locomoção do braço manipulador. Em seguida, tentou-se enviar para o controle a posição e a orientação contidos em um único quatérnion, tendo como resultado instabilidade no controlador do robô.

Após as tentativas de solucionar o problema das forças de auto colisão, via ajustes no controle, não alcançarem os resultados esperados, foi então proposto resolvê-lo modificando o modelo cinemático utilizado pelo controle. O primeiro ajuste a ser implementado foi alterar a posição do efetuador terminal da ponta da garra para o centro do pulso como observado no modelo cinemático do Anexo H, isto foi feito alterando o valor de distância na última junta na matriz de Denavit-Hartenberg para zero. Com esta pequena alteração já se obteve melhoras visíveis no deslocamento do manipulador, isso ocorre devido a rotação das duas últimas juntas do manipulador não estarem mais influenciando na posição do efetuador terminal para o controle, desta forma, sendo necessário compensar menos perturbações. Ainda existe, no entanto, o conflito entre as orientações exigidas pela interface háptica e a orientação fixa que o controlador deseja manter.

Com o objetivo de reduzir este erro, eliminou-se as duas últimas juntas da matriz de DH, as quais representam as juntas do pulso do braço robótico, de sorte que a posição do efetuar terminal se manteve no pulso, mas sua orientação é caracterizada pela antipenúltima junta, a do antebraço. Uma vez que esta junta não apresenta um grande impacto na pose do robô e que sua eliminação faria com que o efetuador terminal fosse localizado no cotovelo, tornando desta forma o controle de posição pela interface háptica não intuitivo, optou-se por manter essa junta no modelo cinemático e retirar a possibilidade de controlá-la pela interface háptica, fazendo com que esta controle somente duas juntas de orientação. Feita esta alteração os movimentos do manipulador tornaram-se mais estáveis e o esforço necessário nas juntas diminuiu, sendo necessário então somente limitar as posições das juntas de orientação que a interface háptica pode enviar para o robô. Este último é solucionado aplicando um limite via software nos ângulos da interface háptica, para evitar um limite duro no software - o que ocasionaria o braço manipulador parar enquanto ainda é possível girar na interface - foi empregado um limite por escala em que o maior ângulo que a interface háptica pode alcançar é menor que os limites de junta do braço manipulador. Por fim, o efeito da remoção das duas juntas no controle faz com que este não seja capaz de atuar tão fortemente na orientação, mas mantendo controle suficiente para translação, obtendo, desta forma, uma diminuição na interferência do controle na orientação e conservando a qualidade do controle

na translação.

## Capítulo 4

# Análise de Resultados

Neste capítulo serão analisados os resultados obtidos a partir dos testes realizados. Estes resultados serão empregados de modo a avaliar a qualidade da movimentação e do retorno háptico realizado. Os gráficos referentes a estes testes se encontram nos Apêndices I e J, sendo o primeiro pertinente aos gráficos das forças captadas pelo sensor de força e transferidas para a interface háptica. Enquanto que o segundo se relaciona com o deslocamento da interface háptica e do efetuator terminal do manipulador.

### 4.1 Validação

Considerando-se concluído os aperfeiçoamentos básicos para o uso da interface háptica em conjunto com o Meka, é relevante avaliar os resultados finais obtidos. Essa análise será feita baseada em dados obtidos a partir dos seguintes experimentos a serem realizados tanto em simulação quanto com o robô real: teste com o braço parado, movimentação livre, teste de colisão e teste de escrita. Em todos os testes, o processo de calibração do offset foi realizado tanto com o robô quanto com a interface háptica parados. Além disso, para os testes realizados no simulador foi utilizado como posição inicial a pose representada na Figura 3.5 e a posição inicial para os testes no robô real é representada na Figura 4.2, em que suas configurações de juntas são respectivamente os vetores em radianos a seguir:

$$\begin{bmatrix} 0.395 & 0.522 & -0.469 & 1.066 & -0.345 & -0.034 & 0.007 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -298 & 0 & -\frac{\pi}{2} & -1.889 & 0 & -0.093 \end{bmatrix}$$

O primeiro teste irá servir para mensurar a capacidade do sensor de força em obter o peso da garra e do programa que deve compensar este peso antes de enviar as forças para interface háptica. O teste de movimentação livre desempenhará o papel de obter os dados para qualificar o controle de posição e orientação da interface háptica para o Meka, além disso, também será possível avaliar a compensação do peso da garra durante translações e em orientações diversas. O teste de colisão

será empregado para aferir a transmissão das forças obtidas pelo sensor de força para a interface háptica.

Por fim, o teste de escrita servirá para examinar o resultado dos controles em uma aplicação real e conjecturar possíveis melhorias a serem propostas. Neste teste deseja-se que o usuário use a interface háptica de modo a formar letras ou palavras a partir do caminho percorrido pelo efetuador terminal do braço manipulador. O resultado deste teste será representado de forma qualitativa a partir de imagens das letras e palavras escritas, é importante citar que este teste será feito somente com o robô real dado as dificuldades de aplicar este teste no ambiente simulado. O teste com o robô real, no entanto, será realizado fixando um pincel atômico em sua garra e desta forma utilizando a interface háptica e o Meka para escrever em um quadro branco.

## 4.2 Análise dos resultados da simulação

Iniciando com os resultados com o robô estacionário, é verificado a eficácia do método empregado em transportar as forças lidas pelo sensor de força para a orientação da interface háptica e, desta forma, compensar o efeito do peso do efetuador terminal em um referencial estático. Com o objetivo de aumentar a confiabilidade deste resultado, o mesmo teste foi realizado em duas posições estacionárias diferentes. A primeira delas sendo a posição de descanso do manipulador, a qual todas suas juntas se encontram em zero. Enquanto que a segunda posição empregada foi a posição inicial da Figura 3.5. Em ambas as poses, o valor da força sentido na interface Omega após o tempo de cálculo do offset é aproximadamente zero. Sendo assim, possível aferir que para posições estacionárias a transformação de coordenadas de referência e a compensação dos efeitos da gravidade são eficientes para o simulador.

Em seguida, os resultados com o manipulador em movimento apontam para as mesmas conclusões. Esta conclusão se mantém mesmo considerando movimentos puramente translacionais, puramente rotacionais ou uma combinação de ambos. Logo após, os resultados qualitativos do teste de colisão indicam a existência do retorno háptico para os usuários durante o contato ou colisão com objetos. Além disso, os gráficos pertencentes a estes testes apresentam variações visíveis no valor da força enviada para a interface Omega durante as colisões como observado na reta vermelha da Figura 4.1, que representa a força sentida pela interface háptica. Finalmente, conclui-se que os resultados obtidos para simulação confirmam a eficácia dos métodos empregados para este meio virtual.

## 4.3 Análise dos resultados no manipulador real

Antes de se iniciar a análise dos resultados do manipulador real, é necessário citar que foram feitas algumas alterações essenciais para a realização dos testes. Entre elas estão a definição de uma nova pose inicial e o acréscimo do número de valores necessários para cálculo do offset. A redefinição da pose inicial decorreu-se do fato que a pose da Figura 3.5 têm as juntas de orientação perto de seus limites físicos, podendo comprometer a movimentação do manipulador. A nova posição de



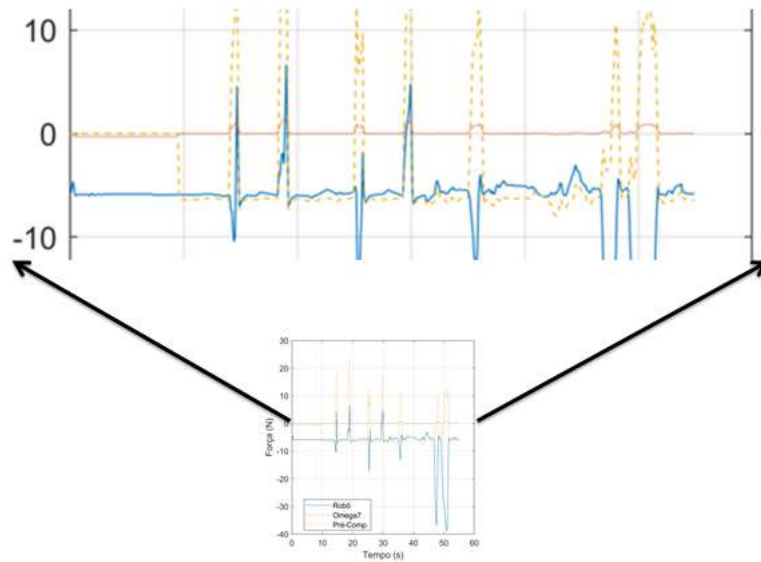


Figura 4.1: Gráfico ampliado de força no eixo Z no teste de colisão da simulação

começo visualizada na Figura 4.2 apresenta, no entanto, diversas desvantagens. As principais destas são: ser uma posição de alto esforço de junta e ser preciso posicionar o manipulador perto da posição desejada manualmente, a fim de evitar que o controle fique instável. Posteriormente, verificou-se a necessidade de se aumentar o número de iterações para 2000 a fim de obter o offset da compensação, isto está relacionado à menor velocidade de iteração da simulação devido a taxa de atualização de quadros do VREP. Após estas considerações foram realizados os mesmos testes a que a simulação foi submetida.



Figura 4.2: Nova posição inicial

Os resultados dos primeiros experimentos se assemelham aos da simulação, em que os valores enviados à interface Omega7 se aproximam de zero. No entanto, existem outros fatores preocupantes como: o alto ruído observado nos dados do manipulador e os altos valores de força decorrentes do peso da garra. Este ruído deve-se principalmente ao uso do controlador proporcional, o qual utiliza muito esforço nas juntas para se manter em uma posição. Isto se confirma ao ver que os dados para posição de descanso do manipulador apresentam um ruído muito menor. Ao mesmo tempo, o alto valor da força peso da garra pode ocasionar que, mesmo as pequenas variações observadas na simulação se tornem valores grandes o suficiente para influenciar no retorno de força da interface háptica.

Estas suspeitas se comprovam com os resultados dos testes posteriores apresentados nos gráficos ampliados das Figuras 4.3 e 4.4. Em ambos, é verificado pela força sentida pela interface háptica, representada pela curva vermelha, que a compensação dos efeitos da gravidade se torna incapaz de exercer sua função de maneira efetiva. Sendo assim, possível sentir feedback de força que deveria ser inexistente durante os testes de movimentação livre e de não o sentir durante impactos no teste de colisão. Por fim, conclui-se que o método testado, apesar dos bons resultados em simulação, foi incapaz de obter a eficácia desejada e atender os parâmetros mínimos para o uso do retorno háptico em uma aplicação sobre o robô real.

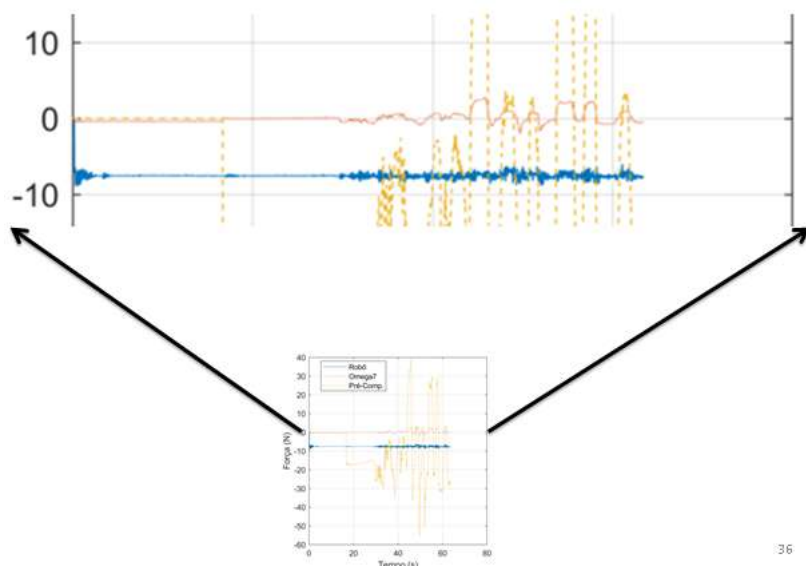


Figura 4.3: Gráfico ampliado de força no eixo Z no teste de movimentação livre no robô real

#### 4.4 Análise dos resultados de posicionamento

Além dos dados correlacionados ao retorno háptico, também foram coletados dados da posição do efetuador da interface háptica e do efetuador terminal do braço manipulador, com a finalidade de avaliar a capacidade do manipulador de executar a mesma trajetória descrita pela interface háptica. Neste teste deseja-se obter curvas no espaço tridimensional idênticas entre o primeiro gráfico representando a posição na interface háptica, e o segundo representando a posição do

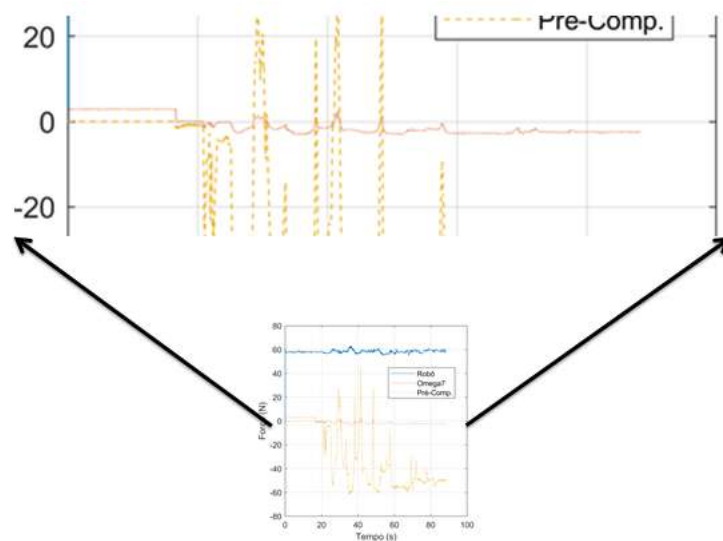


Figura 4.4: Gráfico ampliado de força no eixo Z no teste de colisão do robô real

efetuator terminal do manipulador. No entanto, conforme é possível constatar a partir da forma das trajetórias descritas no Apêndice J, o controle proporcional utilizado não foi capaz de transportar a trajetória do workspace da interface háptica, para o workspace do manipulador, com a qualidade e precisão inicialmente desejadas nem no ambiente virtual e nem no real. Finalmente, ao se realizar o teste de escrita com resultados não ideais na movimentação, obteve-se o resultado apresentado na Figura 4.5, em que foi possível escrever as letras I e D com dificuldade após um período de adaptação na interface háptica. Este visualizado nos rabiscos no centro do quadro branco.

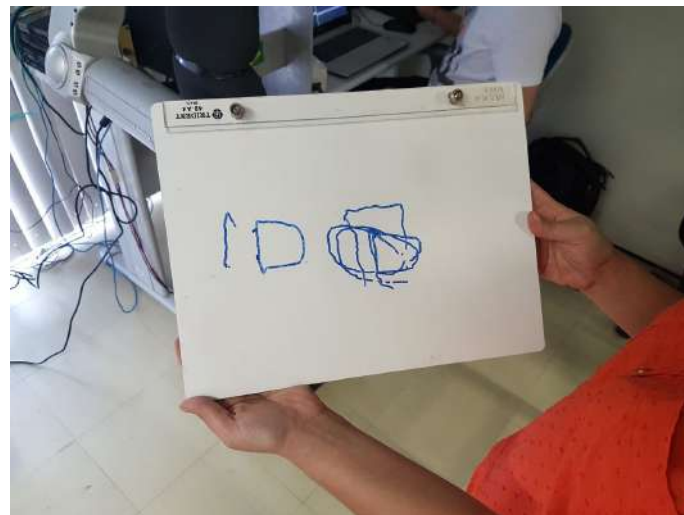


Figura 4.5: Resultado do teste de escrita

## Capítulo 5

# Conclusões

A partir da análise e dos resultados obtidos, conclui-se que, dos objetivos iniciais, não se alcançou uma qualidade suficiente para solucionar os problemas relatados na seção 1.2.1. No entanto, é observado que, apesar deste trabalho não apresentar grande rigor nos dados estatísticos e teóricos na calibração e controle, já que os objetivos iniciais diziam respeito a qualidade do uso de um braço manipulador com um âmbito qualitativo e de aplicação, ele possibilita o desenvolvimento de diversos outros projetos, pesquisas e testes com o rigor necessário de maneira mais fácil, uma vez que as interfaces para o uso do Omega com o Meka ou o VREP estão prontos e acessíveis. Além disso, os erros e dificuldades de modelagem, metodologia e calibração encontrados foram destrinchados e suas possíveis soluções discutidas extensivamente no corpo deste trabalho. Desta forma, podendo servir como um passo inicial, um guia de usuário, um manual ou até mesmo uma coletânea para o uso destes equipamentos e tópicos ainda pouco utilizados em conjunto, sendo estes o manipulador complacente, a interface háptica e o controle via quatérnios duais.

Além disso, apesar da ideia deste trabalho de utilizar um controlador proporcional devido a sua simplicidade e sua pequena influência no tempo de resposta entre o manipulador e a interface háptica, constatou-se a necessidade do uso e estudo de outros controladores possíveis de serem empregados e os efeitos destes. Ademais, foram obtidos bons resultados para simulação, mas resultados negativos para aplicação em conjunto com o manipulador Meka A2. Conjectura-se que estes resultados negativos são consequências dos ruídos causados pelos esforços de juntas, do controlador inadequado e do alto valor de peso da garra (677 gramas) em relação ao peso máximo de 2kg suportado pelo manipulador .

### 5.1 Perspectivas Futuras

O principal objetivo deste trabalho foi realizar a união de diversas áreas de estudos com um foco na aplicação e avaliação qualitativa de resultados. Todos os tópicos aqui abordados podem ser aprofundados de maneira a tentar solucionar os problemas encontrados. Algumas abordagens interessantes a se considerar seriam: analisar qual dos controladores de Pereira [28] melhor se adequa para o uso com a interface háptica, considerar redes de comunicação para teleoperação

remota, desenvolvimento de um modelo dinâmico do Meka A2 para realizar uma compensação mais precisa da gravidade, o uso da interface háptica para delimitar trajetórias como ferramenta de auxílio ao usuário [49], uso da interface háptica para aumentar o espaço de trabalho [50].

Outros exemplos de melhorias simples, mas essenciais a serem realizadas são: realizar uma calibração e filtragem mais rigorosa do sensor de força, acrescentar o uso do DOF de abertura e fechamento da pinça tanto no Meka quanto na interface Omega e, por fim, refinar o retorno háptico utilizando os valores de torque, viscosidades e velocidades, e não apenas de força como atualmente.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MARESCAUX, J.; RUBINO, F. Telesurgery, telementoring, virtual surgery, and telerobotics. *Current Urology Reports*, 2003.
- [2] HASHIZUME, M. et al. A new era of robotic surgery assisted by a computer-enhanced surgical system. 2002.
- [3] BALLANTYNE, G. H.; MOLL, F. The da vinci telerobotic surgical system: the virtual operative field and telepresence surgery. *Surgical Clinics of North America*, 2003.
- [4] LANFRANCO, A. R. et al. Robotic surgery, a current perspective. 2004.
- [5] SURGICAL robots, a brief history. 2014. Disponível em: <http://allaboutroboticsurgery.com/surgicalrobots.html>.
- [6] CAZAC, C.; RADU, G. Telesurgery: an efficient interdisciplinary approach used to improve the health care system. *Journal of Medicine and Life*, v. 7, 2014.
- [7] MARESCAUX, J. et al. Transcontinental robot-assisted remote telesurgery: Feasibility and potential applications. 2002.
- [8] GHODOUSSI, M.; BUTNER, S. E.; WANG, Y. Robotic surgery - the transatlantic case. *International Conference on Robotics & Automation*, 2002.
- [9] ANVARI, M.; MCKINLEY, C.; STEIN, H. Establishment of the worlds first telerobotic remote surgical service for provision of advanced laparoscopic surgery in a rural community. 2005.
- [10] KONSTANTINIDIS, K. M. et al. Current status of robotics in general surgery. *Hellenic Journal of Surgery*, 2010.
- [11] BALLANTYNE, G. H. Robotic surgery, telerobotic surgery, telepresence, and telementoring. 2002.
- [12] WILSON, E. B. The evolution of robotic general surgery. *Scandinavian Journal of Surgery*, 2009.
- [13] ANDRIST, S.; MUTLU, B.; TAPUS, A. Look like me: Matching robot personality via gaze to increase motivation. 2015.

- [14] WILLIAMSON, M. M. *Series Elastic Actuators*. Dissertação (Mestrado) — MASSACHUSETTS INSTITUTE OF TECHNOLOGY, January 1995.
- [15] PRATT, G. A.; WILLIAMSON, M. M. *Series elastic actuators*. 1995.
- [16] DAMME P. BEYL, B. V. R. V. H. I. V. A. M. P. C. D. L. M. V. The role of compliance in robot safety. 2010.
- [17] BICCHI, A.; TONIETTI, G. (Ed.). *Dealing with the Safety-Performance Tradeoff in Robot Arms Design and Control: Fast and Soft-Arm Tactics*. June 2004. (Robotics and Automation Magazine).
- [18] ZINN, M. et al. (Ed.). *A New Actuation Concept for Human-Friendly Robot Design: Playing It Safe*. June 2004. (Robotics and Automation Magazine).
- [19] ZINN, M. et al. A new actuation approach for human friendly robot design. *The International Journal of Robotics Research*, April 2004.
- [20] KUMAR, R. et al. Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation. *International Conference on Robotics and Automation*, 2000.
- [21] BERKELMAN, P. et al. A compact, compliant laparoscopic endoscope manipulator. *International Conference on Robotics and Automation*, 2002.
- [22] QUIGLEY, M.; ASBECK, A.; NG, A. A low-cost compliant 7-dof robotic manipulator. *IEEE International Conference on Robotics and Automation*, 2011.
- [23] CLIFFORD, W. *Mathematical papers*. 1882.
- [24] PHAM, H. et al. Position and orientation control of robot manipulators using dual quaternion feedback. October 2010.
- [25] ADORNO, B. V. *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration*. Tese (Doutorado) — Université de Montpellier, October 2011.
- [26] MARINHO, M. M. *ROBOT-AIDED ENDOSCOPE CONTROL UNDER LAPAROSCOPIC SURGERY CONSTRAINTS USING DUAL QUATERNIONS*. Dissertação (Mestrado) — Universidade de Brasília, Abril 2014.
- [27] FIGUEREDO, L. F. d. C. *KINEMATIC CONTROL BASED ON DUAL QUATERNION ALGEBRA AND ITS APPLICATION TO ROBOT MANIPULATORS*. Tese (Doutorado) — Universidade de Brasília, Julho 2016.
- [28] PEREIRA, M. S. *Trajectory Control of Anthropomorphic Compliant Manipulator with Dual Quaternion Based Kinematic Controllers*. Dissertação (Mestrado) — Universidade de Brasília, Novembro 2016.
- [29] KENWRIGHT, B. A beginners guide to dual-quaternions - what they are, how they work, and how to use them for 3d character hierarchies. 2012.

- [30] SALISBURY, K.; CONTI, F.; BARBAGLI, F. Haptic rendering: Introductory concepts. *IEEE Computer Society*, 2004.
- [31] SAMUR, E. *Performance Metrics for Haptic Interfaces*. [S.l.]: Springer, 2012. (Springer Series on Touch and Haptic Systems, v. 5).
- [32] STEGAGNO, P. et al. A semi-autonomous uav platform for indoor remote operation with visual and haptic feedback. *IEEE International Conference on Robotics & Automation (ICRA)*, June 2014.
- [33] BETHEA, B. T. et al. Application of haptic feedback to robotic surgery. *Journal of Laparoscopic & Advanced Surgical Techniques*, 2004.
- [34] HAGN, U. et al. Dlr mirosurge: a versatile system for research in endoscopic telesurgery. 2010.
- [35] TOBERGTE, A. et al. The sigma.7 haptic interface for mirosurge: A new bi-manual surgical console. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.
- [36] ISOKOSKI, P.; SPRINGARE, J. (Ed.). *Haptics: Perception, Devices, Mobility, and Communication: Part 2*. [S.l.]: Springer, 2012. (EuroHaptics 2012 Conference, v. 2).
- [37] HAMZA-LUP, F. G. et al. A survey of visuo-haptic simulation in surgical training. 2010.
- [38] KESNER, S. B.; HOWE, R. D. Discriminating tissue stiffness with a haptic catheter: Feeling the inside of the beating heart. 2011.
- [39] KING, H. H. et al. Acceleration compensation for vehicle based telesurgery on earth or in space. *IEEE International Conference on Intelligent Robots and Systems*, 2008.
- [40] DIOLAITI, N. et al. Stability of haptic rendering: Discretization, quantization, time delay, and coulomb effects. *IEEE TRANSACTIONS ON ROBOTICS*, v. 22, 2006.
- [41] FORCE DIMENSION. *Omega.X User Manual*. 1.6. ed. [S.l.], December 2011.
- [42] MEKABOT. *Meka A2 Compliant Arm User Guide*. [S.l.], October 2013.
- [43] MEKABOT. *Force Torque Sensor User Guide*. [S.l.], October 2013.
- [44] ATI - INDUSTRIAL AUTOMATION. *Six-Axis Force/Torque Sensor System - Compilation of Manuals*. [S.l.].
- [45] QUIGLEY, M. et al. Ros: an open-source robot operating system. 2009.
- [46] BERGEN, G. Dual numbers: Simple math, easy c++ coding, and lots of tricks. *GDC Europe*, 2009. Disponível em: <[www.gdcvault.com/play/10103/Dual-Numbers-Simple-Math-Easy](http://www.gdcvault.com/play/10103/Dual-Numbers-Simple-Math-Easy)>.
- [47] HAMILTON, W. R. Elements of quaternions. 1886.
- [48] MCDONALD, J. Teaching quaternions is not complex. *Computer Graphics Forum*, v. 29, p. 2447–2455, December 2010.



- [49] MORRIS, D. et al. Haptic feedback enhances force skill learning. *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, 2007.
- [50] CONTI, F.; KHATIB, O. Spanning large workspaces using small haptic devices. *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005.

# ANEXOS

## A. FLUXOGRAMA VREP-MEKA-INTERFACE



Figura A.1: Fluxograma do Robot Controlling Node

## B. FLUXOGRAMA OMEGA7-ROS-INTERFACE

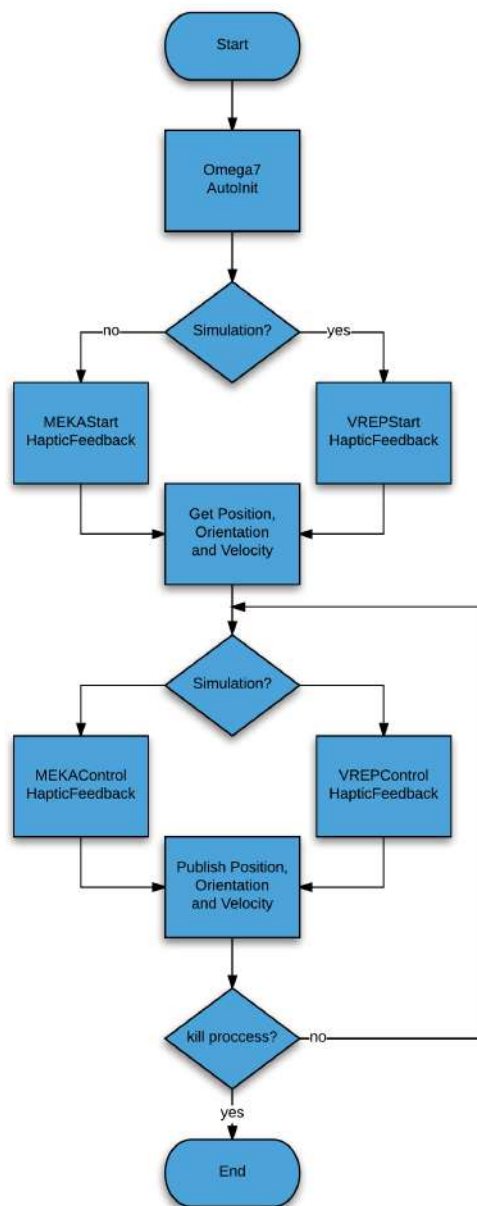


Figura B.1: Fluxograma do Omega7 ROS Interface

## C. FLUXOGRAMA DO HAPTIC CONTROL

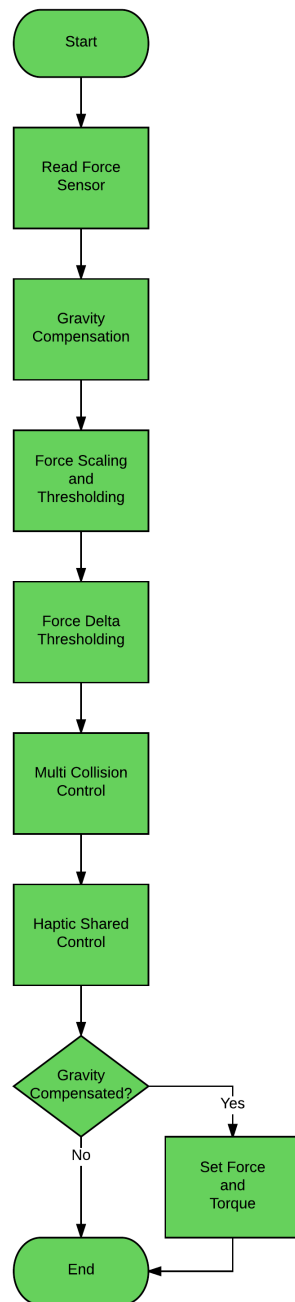


Figura C.1: Fluxograma do Haptic Control



## D. PESOS DE REFERÊNCIA



Figura D.1: Peso de referência para 500g



Figura D.2: Peso de referência para 1000g



Figura D.3: Peso de referência para 2000g

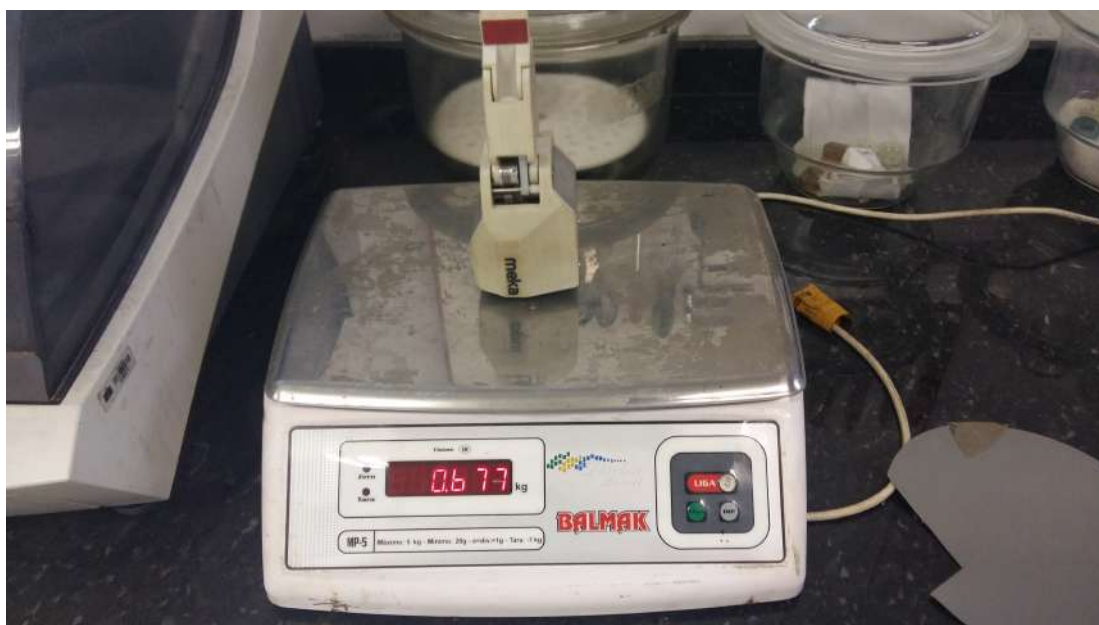


Figura D.4: Peso da garra completa



Figura D.5: Peso da placa de acoplamento



Figura D.6: Peso da contenção da placa com a garra

## E. GRÁFICOS DE CALIBRAÇÃO

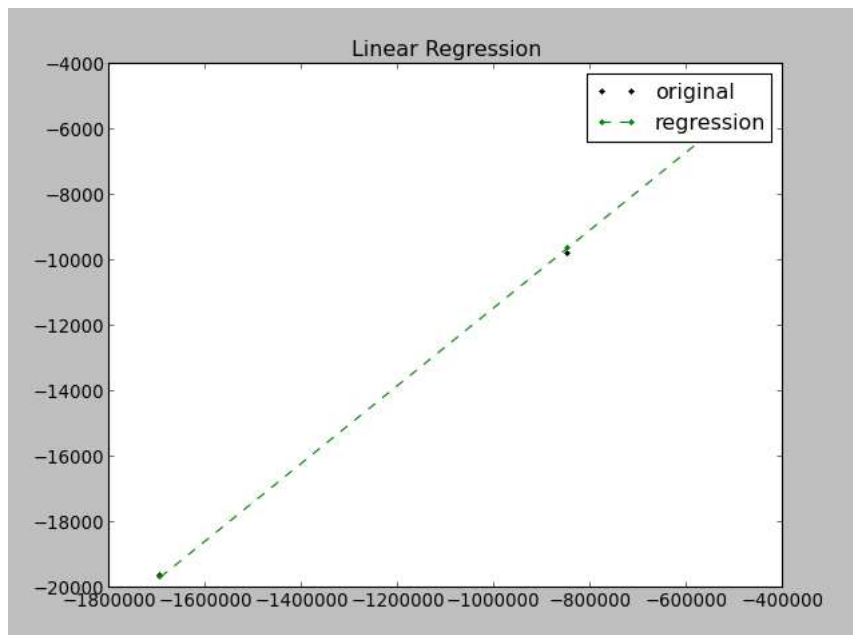


Figura E.1: Primeiro gráfico de saída da calibração **cl**

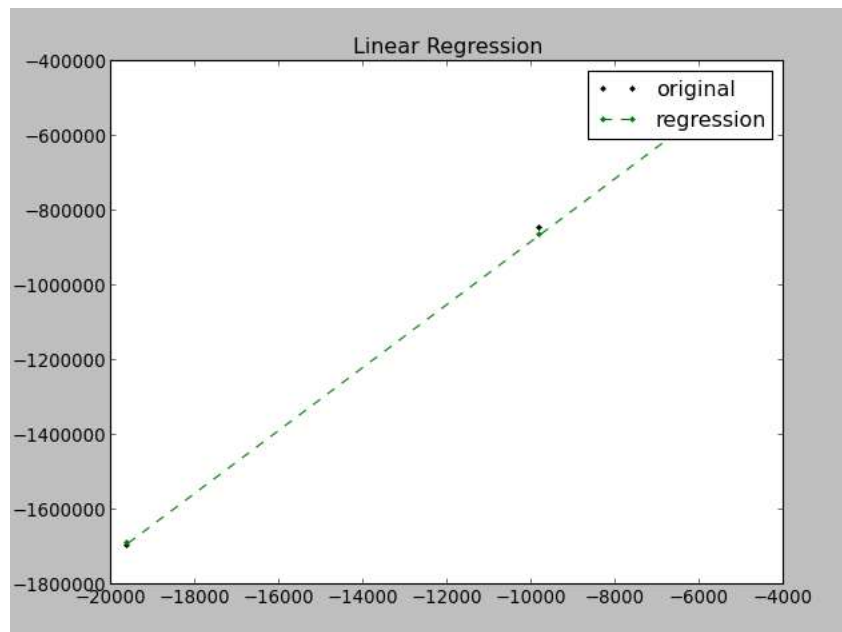


Figura E.2: Segundo gráfico de saída da calibração **cl**

## F. ESPECIFICAÇÕES TÉCNICAS OMEGA7

## 8. technical information

### omega.3

<b>workspace</b>	translation	Ø 160 mm x L 110mm
<b>forces</b>	continuous	12.0 N
<b>resolution</b>	linear	< 0.01 mm
<b>stiffness</b>	closed loop	14.5 N/mm
<b>dimensions</b>	height	270 mm
	width	300 mm
	depth	350 mm
<b>interface</b>	standard	USB 2.0
	rate	up to 8 KHz
<b>power</b>	universal	100V - 240V
<b>calibration</b>	automatic driftless	
<b>structure</b>	delta-based parallel kinematics active gravity compensation	
<b>user input</b>	1 programmable button	
<b>safety</b>	velocity monitoring	
	electromagnetic damping	

### omega.6 extension

<b>workspace</b>	rotation	240 x 140 x 320 deg
<b>resolution</b>	angular	0.09 deg
<b>option</b>	available for right or left hand	
<b>user input</b>	1 programmable button	

### omega.7 extension

<b>workspace</b>	rotation	240 x 140 x 180 deg
	gripper	25 mm
<b>force</b>	continuous	± 8 N
<b>resolution</b>	angular	0.09 deg
	linear	0.006 mm
<b>option</b>	available for right or left hand	

### software

<b>OS</b>	Microsoft	Windows XP (SP2) / Vista / 7
	Linux	kernel 2.6 / 3.0
	Apple	OS X 10.6 / 10.7
	QNX	Neutrino 6.3 / 6.5
<b>SDK</b>	haptic SDK	
	robotic SDK	



## **G. ESPECIFICAÇÕES TÉCNICAS MEKA A2**

Item	Value	
Weight (Kg)	11.4	
Payload (Kg)	2	
Reach (mm)	600	
Gripper mount	Dovetail quick-connect (3) M3 flathead on 33mm BC	
Base mount	Side mount (6) M6x1.0 or bottom mount (6) M3x0.5	
Material	Aluminum Alloy	
	6061-T6	

## Electrical

The A2 can be powered from either a wall supply or batteries. While the average current consumption is low during normal operation, the supply must be able to source high momentary (~60A/10-100ms) peak currents. For this reason, we recommend using a bulk power supply with a large capacitor over a switching supply.

- The A2 internal Power Board is fused at 30A (Fuse: Digikey# 283-2332-ND)
- The maximum voltage input should be kept below 30V at all times.
- A slow blow 20A fuse must be placed in line with the Power Supply or battery supply.
- Power cables should be at least 12AWG
- Power cable length should be kept below 3m to limit the cable inductance. If longer length is required, an large aluminum capacitor should be added close to the robot, with a value depending on the cable length.

Item	Value	
Supply Voltage Nominal	24V	
Supply Voltage Min	20V	
Supply Voltage Max	30V	
Quiescent Power	10W	
Average Power	17W	
Peak Power	120W	
Reverse Protection	Yes, External fuse required	
Wall Voltage US	110VAC	
Wall Voltage International	220VAC	
Fused Current	30A Slow Blow	
Power Supply	Bulk, 24V/10A, fused	

## Actuation

A2 R4 Prelim Specs	J0	J1	J2	J3	J4	J5	J6
Torque Cont (Nm)	14.6	16.7	9.6	9.6	1.9	2.3	2.3
Torque Mom (Nm)	40.0	40.0	20.0	20.0	4.0	8.0	8.0
Theta Min (Deg)	-80.0	-25.0	-85.0	0.0	-110.0	-60.0	-60.0
Theta Max (Deg)	200.0	150.0	85.0	133.0	110.0	60.0	60.0
Stiffness (Nm/rad)	417.0	417.0	190.0	190.0	23.0	46.0	46.0
Motor	Maxon RE40 150W 24V 14887	Maxon RE40 150W 24V 14887	Maxon EC45 Flat 70W 24V 397172	Maxon EC45 Flat 70W 24V 397172	Maxon EC32 Flat 15W 24V 267121	Maxon EC32 Flat 15W 24V 267121	Maxon EC32 Flat 15W 24V 267121
Gear Ratio HD (:1)	120.00	120.00	100.00	100.00	100.00	50.00	50.00
Gear Ratio Belt (:1)	1	N/A	N/A	N/A	N/A	1.19	1.19
Gear Ratio Net (:1)	120.00	120.00	100.00	100.00	100.00	59.52	59.52
Current peak (A)	11.0	11.0	5.4	5.4	0.8	0.8	0.8
Current cont (A)	5.8	5.8	3.3	3.3	0.5	0.5	0.5
Max Vel (Rad/s)	4.6	4.6	4.5	4.5	2.0	3.4	3.4

## Sensing

Sensor	Description
Force	ContElec Vert-X13 Absolute Encoder; 14bits
Motor angle	N/A
Joint angle	Vert-X13 14-bit Absolute Encoder; 14 bits
Motor current	U-V channel low-side 50Khz sampling; 12 bits
Temperature	Ambient sensing inside each link
Motor temperature	N/A; Derived from current, ambient temp, model
6 Axis Force Torque Wrist	Optional. ATI Mini-40E

## Kinematics

The A2R4 kinematic and mass parameters are stored in its Kinematics and Dynamics Config File, as shown in this example. This file [mekabot.com/wiki/doku.php?id=user\\_guides:arm\\_a2r4](http://mekabot.com/wiki/doku.php?id=user_guides:arm_a2r4)

## H. MODELO CINEMÁTICO MEKA A2

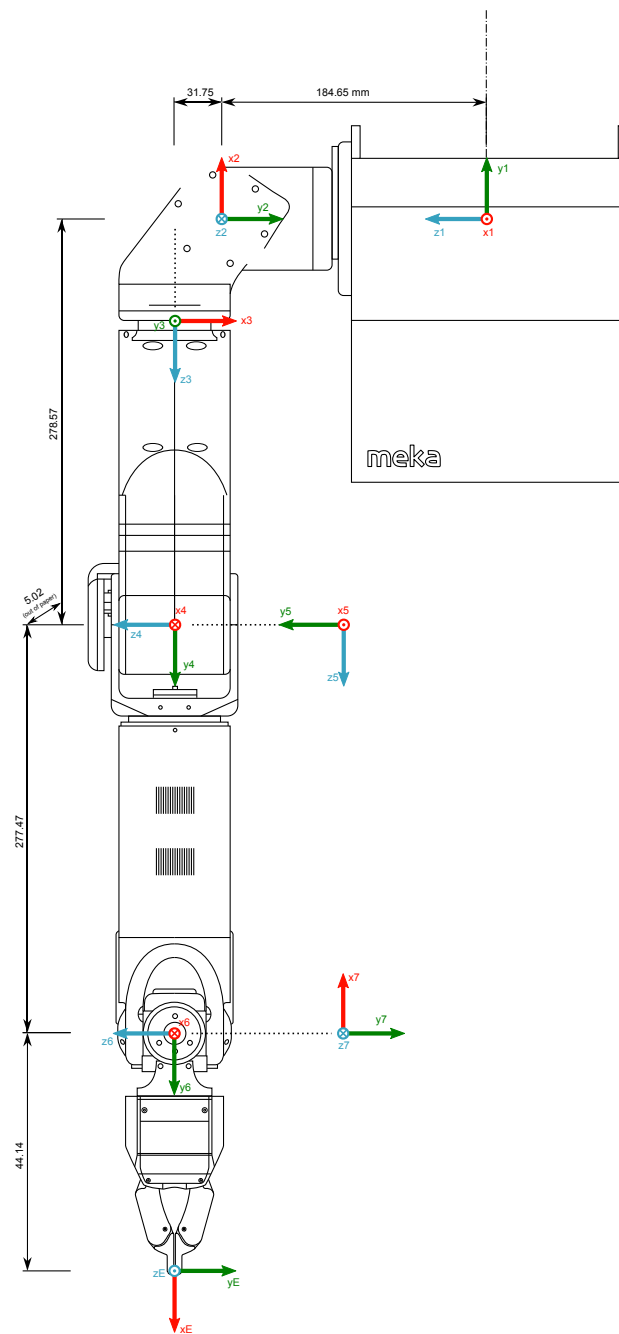


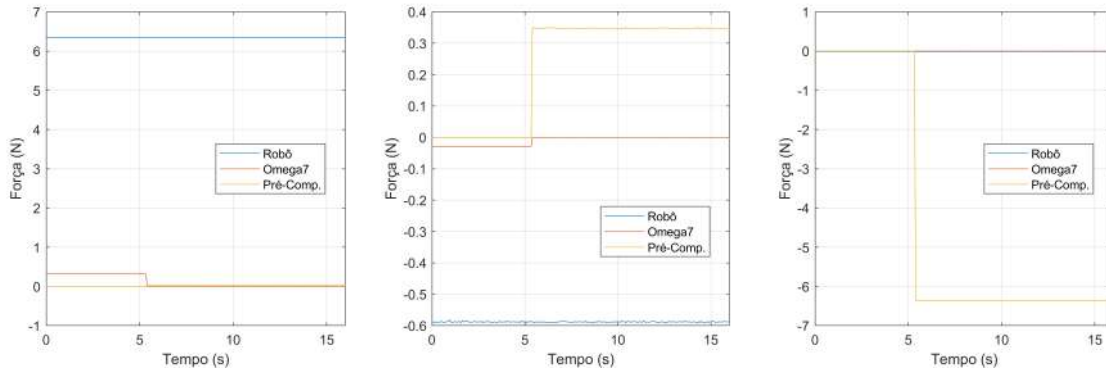
Figura H.1: Modelo cinemático do Meka A2

# I. GRÁFICOS DE FORÇA

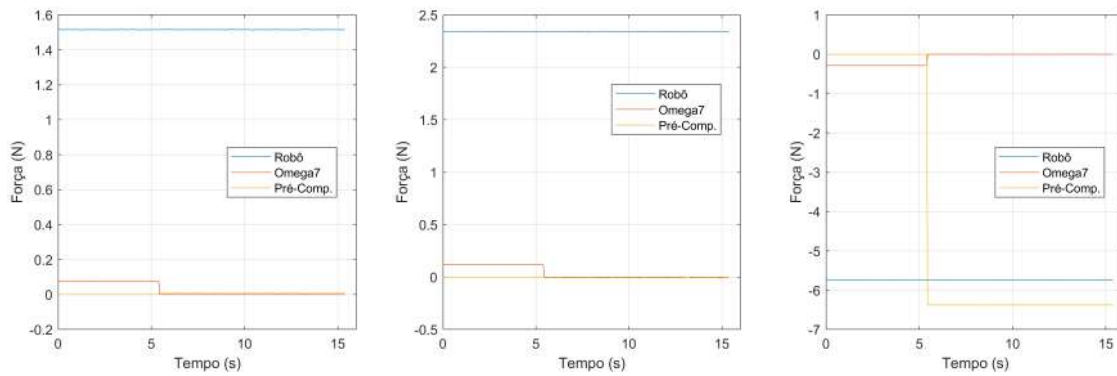
Os gráficos apresentados neste apêndice representam as forças X, Y e Z, respectivamente, dos resultados obtidos a partir dos testes realizados conforme a secção 4.1. A legenda apresenta as forças do manipulador como azul, a saída na interface háptica Omega7 em laranja e, por fim, os valores de força antes sofrerem a compensação da gravidade.

## I.1 Simulação

### I.1.1 Teste com o robô parado

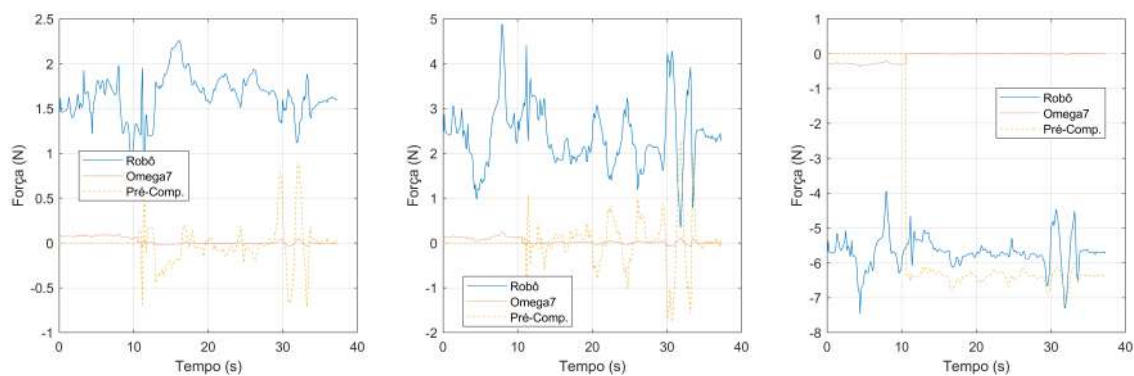


Gráficos para a posição zero das juntas do robô.

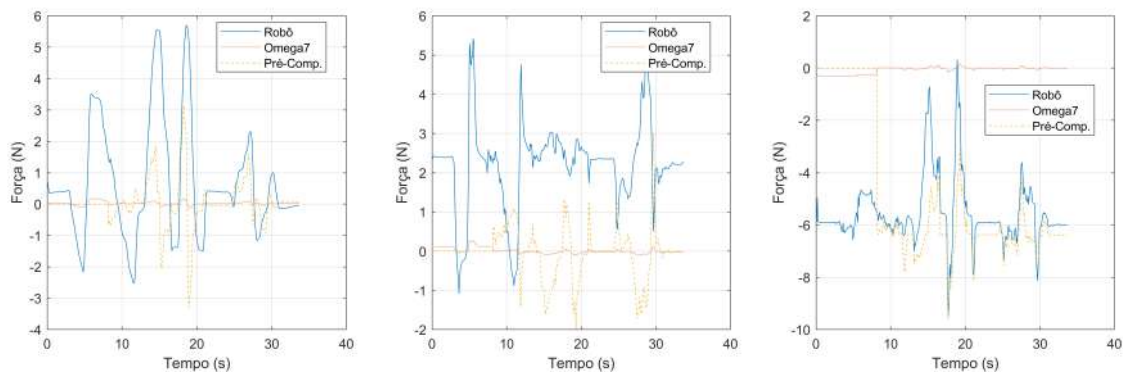


Gráficos para a posição inicial da Figura 3.5.

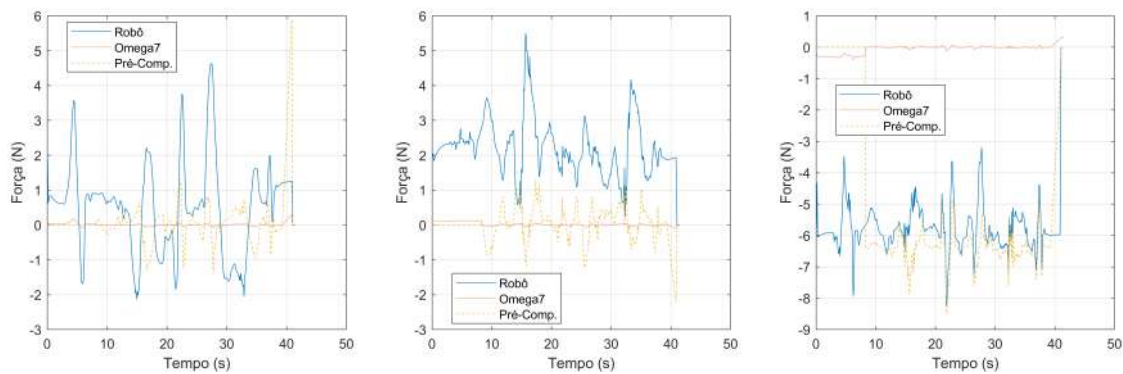
### I.1.2 Teste com o manipulador em movimento



Gráficos para o manipulador em movimentação sem a orientação.

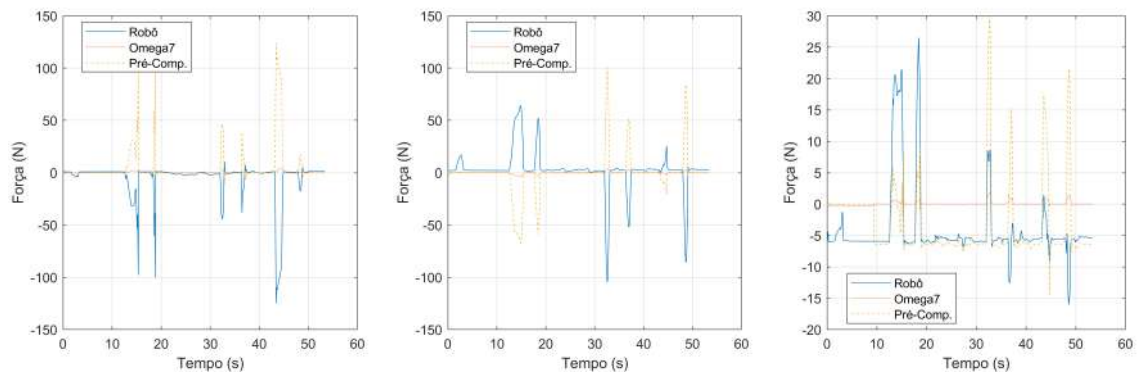


Gráficos para o manipulador em movimentação somente com a orientação.

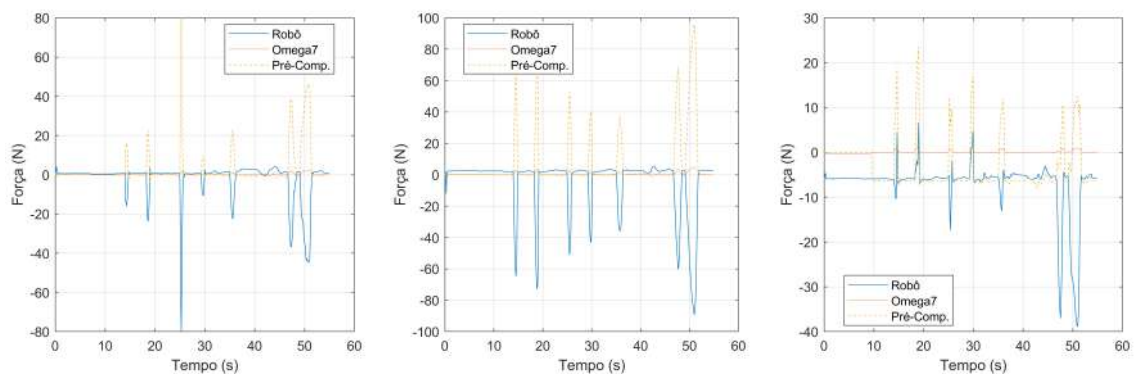


Gráficos para o manipulador em movimentação livre.

### I.1.3 Teste de colisão



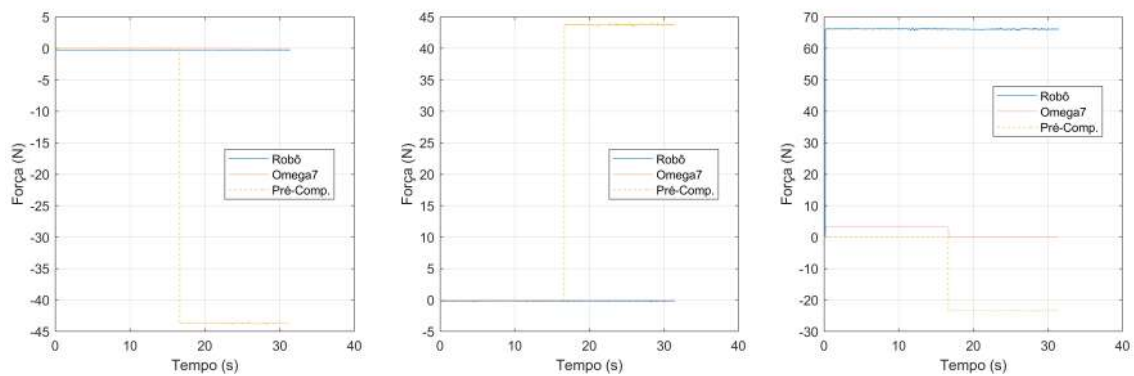
Gráficos para o manipulador em colisão A.



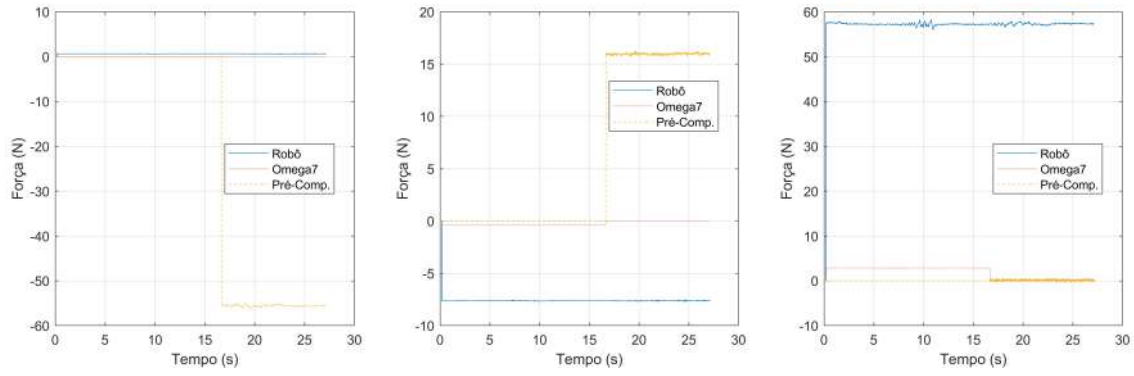
Gráficos para o manipulador em colisão B.

## I.2 Testes com o robô real

### I.2.1 Teste com o manipulador parado

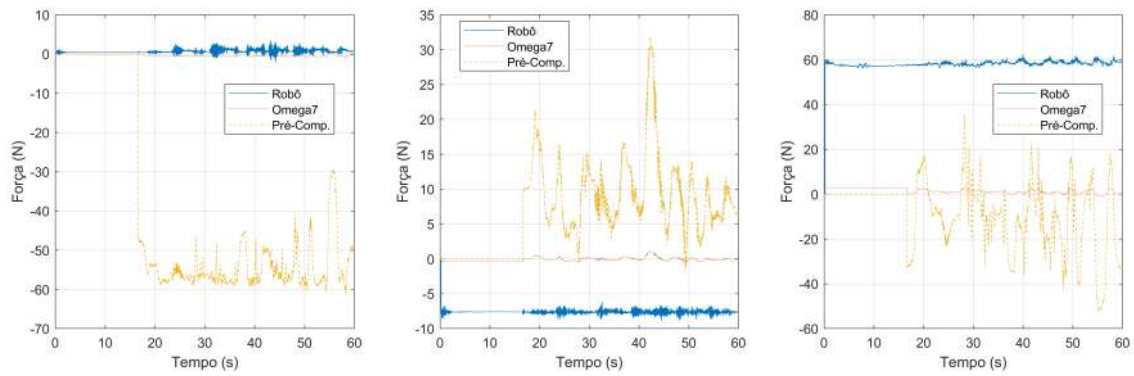


Gráficos para a posição zero das juntas do robô.

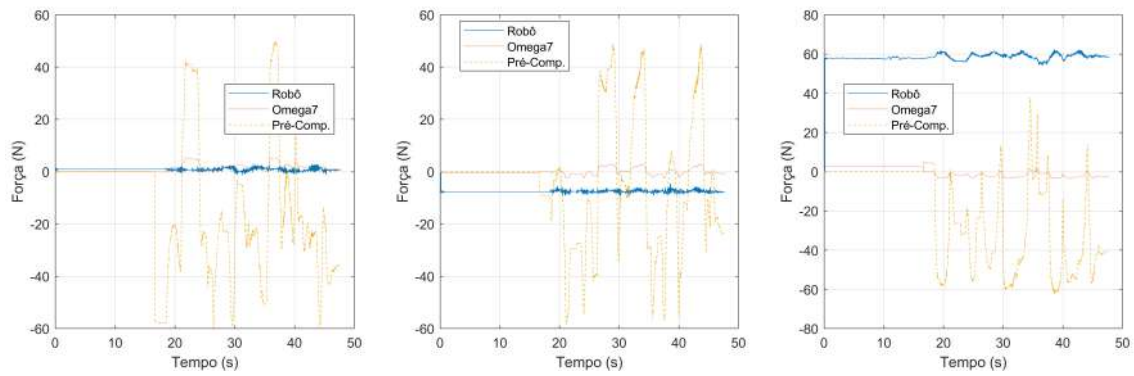


Gráficos para a posição inicial da Figura .

## I.2.2 Teste com o manipulador em movimento

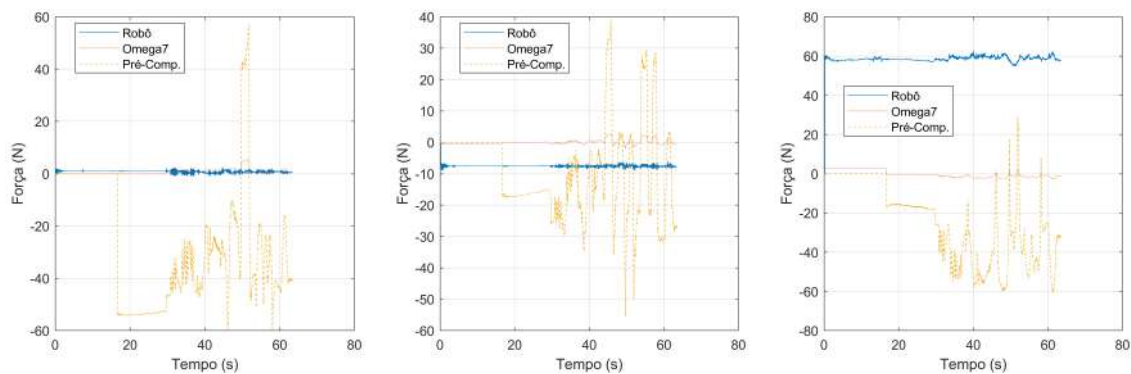


Gráficos para o manipulador em movimentação sem a orientação.



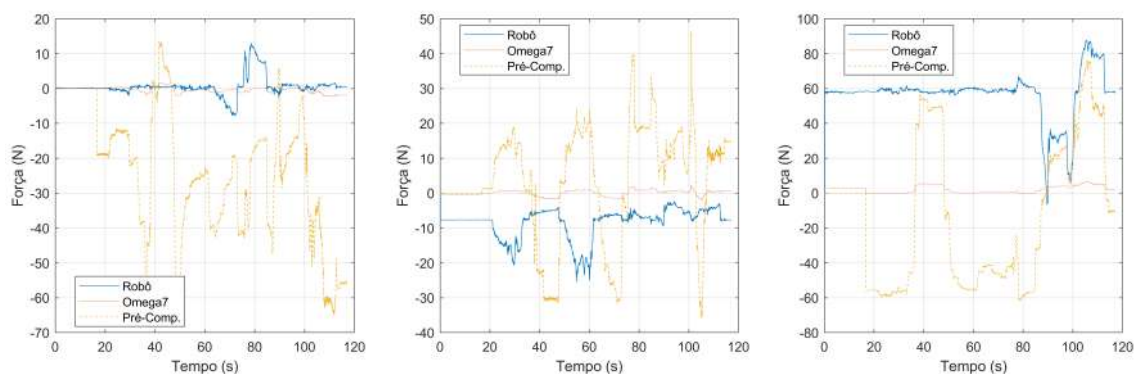
Gráficos para o manipulador em movimentação somente com a orientação.



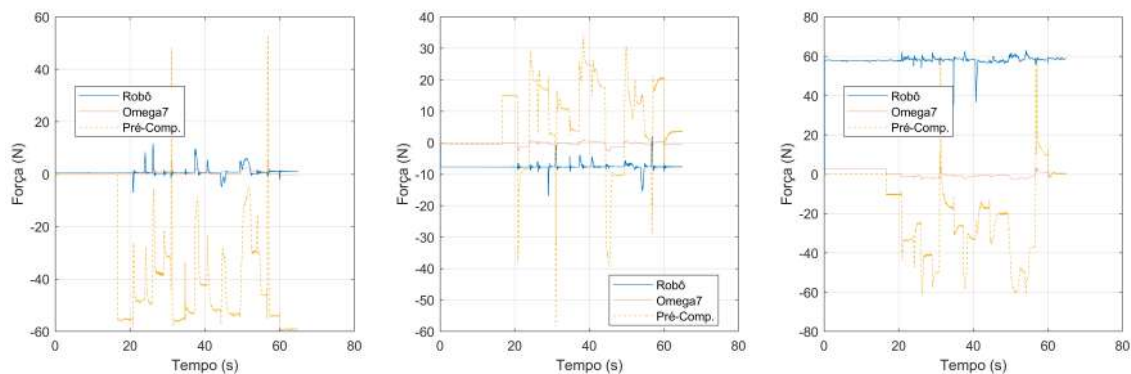


Gráficos para o manipulador em movimentação livre.

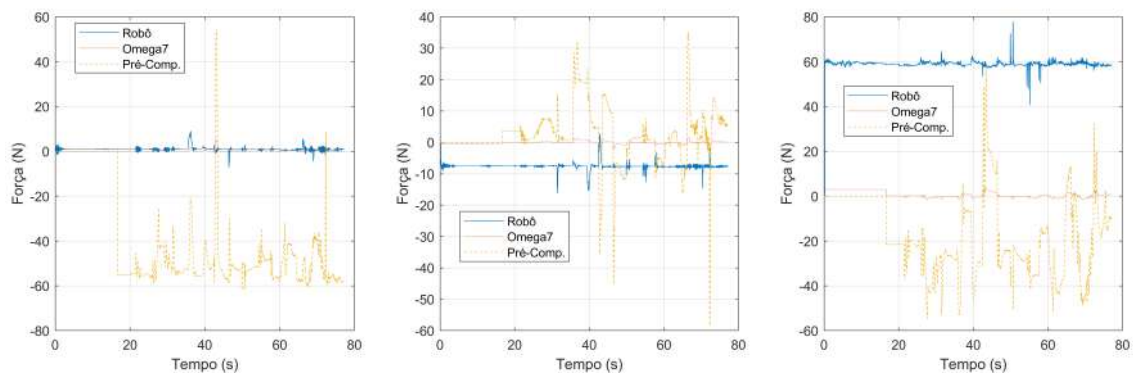
### I.2.3 Teste de colisão



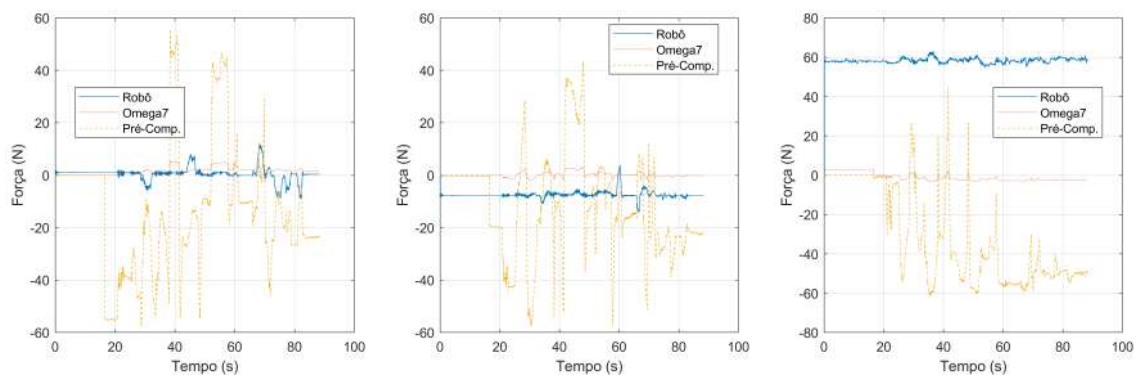
Gráficos para o manipulador parado sobre o efeito de forças de longa duração.



Gráficos para o manipulador parado sobre o efeito de forças de curta duração.



Gráficos para o manipulador em movimento sem orientação.

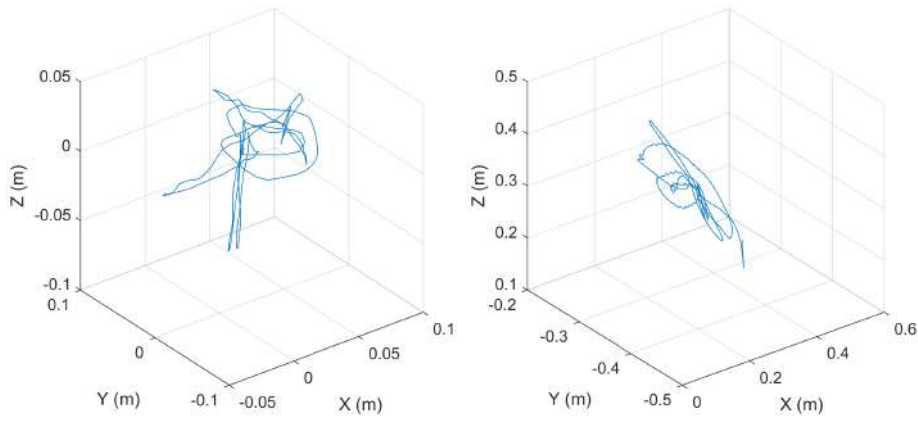


Gráficos para o manipulador em movimento com orientação.

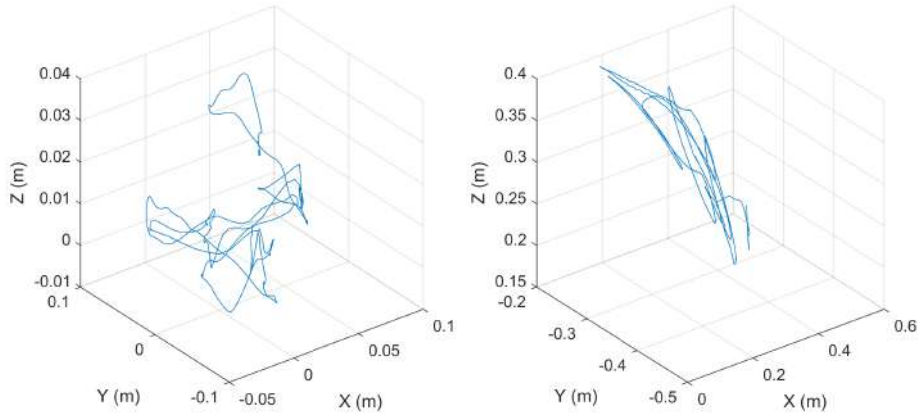
## J. GRÁFICOS DE POSIÇÃO

Os gráficos apresentados neste apêndice representam as trajetórias seguidas pelo efetuador terminal e a interface háptica, respectivamente, dos resultados obtidos a partir dos testes realizados conforme a secção 4.1.

### J.1 Simulação

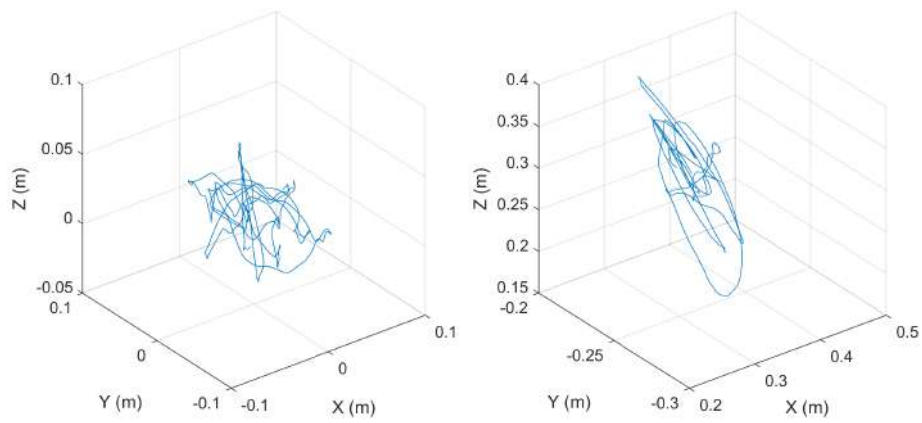


Gráficos para a trajetória do movimento realizado sem orientação.

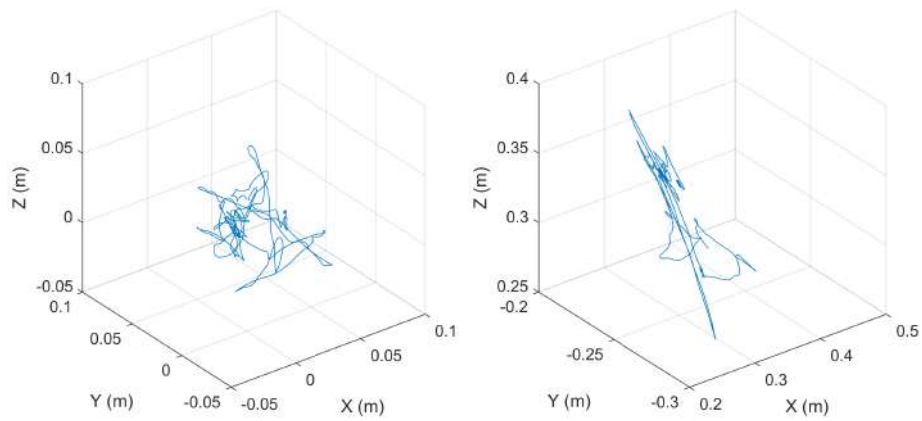


Gráficos para a trajetória do movimento realizado com orientação.

## J.2 Robô real



Gráficos para a trajetória do movimento realizado sem orientação.



Gráficos para a trajetória do movimento realizado com orientação.