



TRABALHO DE GRADUAÇÃO

**ESTUDO E CARACTERIZAÇÃO DE ARQUITETURA DE
CONTROLE DE BRAÇO ROBÓTICO COMPLACENTE**

Rafael Lima

Brasília, Dezembro de 2018



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

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

TRABALHO DE GRADUAÇÃO

ESTUDO E CARACTERIZAÇÃO DE ARQUITETURA DE CONTROLE DE BRAÇO ROBÓTICO COMPLACENTE

Rafael Lima

*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 Bernardes, FGA/UnB _____
Co-orientador

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

Prof. Hugo Tadashi Muniz Kussaba, ENE/UnB _____
Examinador interno

Brasília, Dezembro de 2018

FICHA CATALOGRÁFICA

LIMA, RAFAEL

Estudo e caracterização de arquitetura de controle em braço robótico complacente,

[Distrito Federal] 2018.

x, 54p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2018). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1. Robótica Colaborativa

2.Manipuladores Robóticos

3. Controle

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

LIMA, RAFAEL, (2018). Estudo e caracterização de arquitetura de controle em braço robótico complacente. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-n°022, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 54p.

CESSÃO DE DIREITOS

AUTOR: Rafael Lima

Estudo e caracterização de arquitetura de controle em braço robótico complacente.

GRAU: Engenheiro

ANO: 2018

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 Lima

email: raffaellimma@gmail.com

Agradecimentos

Absolutamente nada nesta vida se faz só. Para chegar até aqui devo a tantos que citando teria mais páginas que este trabalho. Tentarei resumir. Primeiramente agradeço a meus pais, Francisco e Zely pois devo tudo a eles, desde minha vida a cada aspecto construído nela. A meus irmãos Lorrane e Mateus, pela imensa parceria em qualquer situação. A minha namorada Lia, que me ajudou ao longo de tantas vezes que esqueci completamente de mim enquanto envolvido com alguma tarefa do curso. A minha família, tios, primos e avós por trazerem sempre a alegria em tudo. Aos dois grandes amigos Ciro e Pedro, que me inspiram a sempre persistir por um mundo melhor.

Gostaria de agradecer a tantos amigos que fiz ao longo dos anos na mecatrônica. Seria impossível manter a sanidade mental e chegar até o final sem ajuda de cada um de vocês. Em particular gostaria de agradecer à turma 27 de engenharia mecatrônica, pela amizade de cada um. São minha inspiração pelos talentos únicos, força de vontade e dedicação para seus sonhos. Agradeço especialmente ao Ataías, David Passos, Marina Moreira, Marcos Pereira, Jessé Barreto, Marco Emílio,, Rodrigo Texeira, Filipe Caixeta, Matheus Takahashi, Gabriel Cataldi, Victor Caveira, Guilherme, De Hong Jung, Lucas Fugita, Francisco Gomes, Eric Torlig, Cris Miranda. No Lara aos amigos, Gabriel Moises, Miguel Gutierrez, Eric Vale, Arthur, Bruno, Breno, Lucas Rato, Day, Pedro Perrucci, Roberto Baptista, Thiago Rocha, Cláudia Ochoa, David Fiorilo, Felipe Moreira e George Brindeiro. E aos amigos André, Arthur, William e Pamella.

Agradeço a cada um dos professores que me ajudaram em cada parte do caminho. Benilton e Felipe da sala de recursos. Na universidade: Clod Wagner, Célius Magalhães, Rathie Puspa, Arthur Vicentini, Fábio Menezes, Marcos Lamar, José Maurício, Renato Borges, Rafael Shayani, Adolfo Bauchspiess. João Ishihara, Holly, Ilene, Dean, Laura Kline, Ammar Basu e Ana Djuric. A Yuri Dumaresq e a Carla Koike, preceptores ao longo do curso que me ajudaram incontáveis vezes. Ao professor Antônio Padilha, responsável pelo começo dos meus estudos na área de saúde. E a Mohammad Mehr-mohammad, que me acompanhou durante o intercâmbio. E por fim para a professora Mariana, que me acompanhou durante toda a jornada, apoiando em tudo desde o início. E ao professor Geovany por me ajudar na orientação deste trabalho com diversas dicas extremamente precisas em vários momentos que eu estava completamente perdido.

A cada, inclusive aos muitos não citados aqui, um sincero Muito Obrigado!

Rafael Lima

RESUMO

Neste trabalho é investigado a arquitetura do Meka A2, um manipulador robótico antropomórfico composto por sete juntas complacentes. Este robô foi desenvolvido para uso em pesquisa na área de interação com pessoas. Embora constitua de uma plataforma bastante segura, em trabalhos anteriores foi observado uma dificuldade no controle de posição através de controladores cinemáticos. Desta forma foi feito um estudo de toda a arquitetura do robô para identificar quais aspectos contribuem para um desempenho ruim e possíveis condutas para mitigar os problemas.

Palavras Chave: Robôs Colaborativos, Atuadores Série-Elásticos, Sistemas em Tempo Real

ABSTRACT

This work aim to provide a better acknowledgement about the robot Meka A2 arm architecture and, the kinematics control problems reported. It's a robotics manipulator with seven compliant joints available at the Robotics and Automation Laboratory at UnB.

Keywords: Corobots, Serie Elastic Actuators, Real Time System

SUMÁRIO

1	Introdução.....	1
1.1	CONTEXTUALIZAÇÃO.....	1
1.1.1	BREVE HISTÓRICO.....	2
1.2	DEFINIÇÃO DO PROBLEMA	4
1.3	METODOLOGIA	5
1.3.1	REVISÃO TRABALHOS ANTERIORES	5
1.3.2	INVESTIGAÇÃO E EXPERIMENTAÇÃO A PARTIR DA PLATAFORMA	5
1.4	ESTRUTURA DO DOCUMENTO	6
2	Fundamentos	7
2.1	MANIPULADORES ROBÓTICOS	7
2.2	QUATÉRNIONS DUAIS	8
2.2.1	QUATÉRNIONS	8
2.2.2	NÚMEROS DUAIS	9
2.2.3	QUATÉRNIONS DUAIS	9
2.3	BRAÇO ROBÓTICO MEKA A2	9
2.4	ATUADORES SÉRIE ELÁSTICOS	10
2.4.1	ENCODER	12
2.4.2	HARMONIC DRIVE	12
2.4.3	MOTORES BRUSHLESS	13
2.4.4	PULSO DIFERENCIAL.....	14
2.5	PLATAFORMA DE CONTROLE	15
2.5.1	DSP CONTROL BOARDS	15
2.5.2	MEKA PC	16
2.6	M3	17
2.6.1	ARQUIVOS DE CONFIGURAÇÃO	18
2.6.2	CONTROLADORES.....	18
3	Desenvolvimento.....	20
3.1	DESEMPENHO DE SISTEMAS MECATRÔNICOS	20
3.2	AVALIAÇÃO PRELIMINAR	22
3.3	ESTUDO CONTROLADORES CINEMÁTICOS	22
3.4	LAYOUT EXPERIMENTOS	22

3.4.1	ROS.....	23
3.4.2	SHM_HUMANOID_INTERFACE.....	23
3.4.3	ROSBAGS	24
3.4.4	PYTHON API	24
4	Resultados.....	25
4.1	ESTUDO ARQUITETURA	25
4.1.1	SISTEMA MECÂNICO	25
4.1.2	SISTEMA DE CONTROLE.....	27
4.2	ESTUDO CONTROLADORES M3.....	28
4.2.1	CÓDIGO DE DEMONSTRAÇÃO.....	28
4.2.2	API PYTHON	29
4.3	ESTUDO CONTROLADORES CINEMÁTICOS	32
4.3.1	CONTROLE DE POSIÇÃO DE JUNTAS	32
4.3.2	AVALIAÇÃO NA EXECUÇÃO DE TRAJETÓRIA EM LINHA RETA	36
4.4	ESTUDO DE SATURAÇÃO DA VELOCIDADE DOS ATUADORES.....	37
4.4.1	SATURAÇÃO DO ATUADOR NA TRAJETÓRIA	40
4.5	ESTUDO CONTROLADORES CINEMÁTICOS	42
4.5.1	ANÁLISE INDIVIDUAL DOS CONTROLADORES DE JUNTA	42
4.5.2	CONTROLE DE RIGIDEZ	46
4.6	CONCLUSÃO	48
5	Conclusão	50
5.1	PERSPECTIVAS FUTURAS.....	50
REFERÊNCIAS BIBLIOGRÁFICAS		51
Anexos.....		53
I Descrição do conteúdo do CD		54

LISTA DE FIGURAS

1.1	Foto Robô Domo (Fonte: http://www.robotsvoice.com)	3
1.2	Meka M1 (Fonte: https://spectrum.ieee.org).....	3
1.3	Baxter em linha de produção	4
1.4	Braço Robótico Meka A2 disponível no Lara [1]	5
2.1	Robô Meka - Vista interna (Fonte: http://machinedesign.com)	10
2.2	Topologia Atuador Rígido [?].	11
2.3	Atuador Série Elástico Implementado por Pratt [2]	11
2.4	Encoder utilizado nas juntas do robô.....	12
2.5	Harmonic Drive (Fonte: https://www.orientalmotor.com)	13
2.6	Motor utilizado nas juntas do Ombro	14
2.7	Motor utilizado nas juntas do braço	14
2.8	Diagrama mecanismo diferencial do pulso [3]	15
2.9	Destaque Placas de controle entre as juntas do pulso e o cotovelo do braço	16
2.10	Diagrama da arquitetura do robô [4]	17
2.11	Arquitetura sistema de controle M3 [5]	18
2.12	Arquitetura dos controladores do Meka [5]	19
3.1	Representação do Desempenho de um sistema [6].....	21
3.2	Planejamento de Trajetória no Baxter com Auxílio do MoveIt [7].....	23
4.1	Arquitetura Controle C++	27
4.2	Interface Código de Demonstração [5].	29
4.3	Resposta a um degrau de entrada com Velocidade = 1 e Rigidez = 1	33
4.4	Resposta a um degrau de entrada com Velocidade = 1 e Rigidez = 0.5	34
4.5	Resposta a um degrau de entrada com Velocidade = 0.7 e Rigidez = 1	35
4.6	Experimento Trajetória na Vertical $dt = 10ms$	37
4.7	Experimento Trajetória na Vertical $dt = 2ms$	37
4.8	Resposta Controladores Juntas - Velocidade Máxima e 90% de rigidez	38
4.9	Resposta Controladores Juntas - Velocidade em 10%e rigidez em 90%.....	40
4.10	Resposta dos Controladores de Juntas para Velocidade em 70% e Stifness = 5	42
4.11	Resposta Juntas $J0$ para Velocidade em 70% e Stifness = 100%	43
4.12	Resposta Juntas $J1$ para Velocidade em 70% e Stifness = 100%	44
4.13	Resposta Juntas $J2$ para Velocidade em 70% e Stifness = 100%	44

4.14 Resposta Juntas <i>J</i> 4 para Velocidade em 70% e Stifness = 100%	45
4.15 Resposta Juntas <i>J</i> 5 para Velocidade em 70% e Stifness = 100%	45
4.16 Resposta Juntas <i>J</i> 6 para Velocidade em 70% e Stifness = 100%	46
4.17 Resposta Juntas <i>J</i> 6 para Velocidade em 100% e Stifness = 90%	47
4.18 Resposta Juntas <i>J</i> 6 para Velocidade em 100% e Stifness = 50%	47
4.19 Resposta Juntas <i>J</i> 6 para Velocidade em 100% e Stifness = 20%	48

LISTA DE TABELAS

4.1	Especificação Preliminar dos Atuadores, adaptado de [5]	26
4.2	Especificação dos Sensores, adaptado de [5]	27
4.3	Error Percentual para diferentes valores de velocidade (V) e rigidez (S)	35
4.4	Percentual Overshot para diferentes valores de velocidade (V) e rigidez (S)	35
4.5	Valor em regime permanente para diferentes valores de velocidade (V) e rigidez (S)	36

LISTA DE SÍMBOLOS

Símbolos Latinos

v	Velocidade linear	[m/s]
a	Aceleração linear	[m/s]
F	Força	[m/s]

Símbolos Gregos

ω	Velocidade angular	[rad/s]
α	Aceleração angular	[rad/s]
τ	Torque	[rad/s]

Grupos Adimensionais

i, k	Contador
------	----------

Subscritos

ref	referência
fer	ferramenta
sis	sistema
des	desejado

Sobrescritos

.	Variação temporal
-	Valor médio
^	Vetor Unitário

Siglas

API	Interface de Programação de Aplicações - <i>Application Program Interface</i>
RTOS	Sistema Operacional em Tempo Real - <i>Real Time Operating System</i>
ROS	Sistema Operativo de Robôs - <i>Robotic Operating System</i>
M3	Sistema de Controle M3
LARA	Laboratório de Automação e Robótica da UnB
DSP	Processador Digital de Sinais - <i>Digital Signal Processing</i>
SEA	Atuador Série Elástico - <i>Serie Elastic Actuator</i>

Capítulo 1

Introdução

"I never think of the future - it comes soon enough.- Albert Einstein

1.1 Contextualização

Toda tecnologia surge no intento de reduzir o gasto de energia na execução de alguma tarefa. Nela consolidamos todo o conhecimento envolvido em alguma técnica permitindo a execução com um esforço muito menor a partir da consolidação disto em um objeto. A este damos diversos nomes: ferramentas, máquinas, robôs, lápis e caneta. Essencialmente todos com a mesma missão de permitir realizar qualquer tarefa com a menor quantidade de energia possível.

As primeiras tecnologias utilizavam desta capacidade de executar as tarefas a partir de fenômenos naturais. Desta forma, o conhecimento aliado a propagação de calor do fogo, a fluidez da água e do vento e a rigidez da pedras nos permitiram conceber ferramentas cada vez mais complexas. Com o passar da história alcançamos o ponto dos próprios engenhos poderem ser iniciados e deixados a executarem tarefas sozinhos sem qualquer interferência de pessoas. No que então começamos a chamada automação.

Robôs industriais, ou manipuladores robóticos, como também são chamados, se tornaram parte essencial na produção de bens aonde alguma das etapas exijam capacidades sobre-humanas de força, precisão, repetibilidade ou ainda na tolerância a altas temperatura e materiais corrosivos. Garantindo assim consistência, rapidez e segurança em tarefas de soldagem, montagem, pintura e corte no cenário industrial. No entanto, estas mesmas características que conferem a robustez apreciada no ambiente industrial tornam os robôs perigosos. A força, peso e velocidades de operação compõem elementos de risco na execução de atividades na presença de pessoas.

Mesmo sendo uma tecnologia bastante interessante na automação de processos que envolvam a manipulação de objetos, o uso de braços robóticos pode oferecer diversos riscos. Para garantir a segurança em ambiente industrial, diversas soluções são utilizadas como grades de isolamento, botões de parada e sistemas de detecção de pessoas. Atuando em geral na desativação do robô como forma de mitigar o risco. No entanto para algumas atividade existe a necessidade da atuação conjunta de pessoas e máquinas. Para resolver este problema novos tipos de braços têm sido desenvolvidos incorporando características para segurança em cada aspecto do design, entre elas

emprego de estruturas mais leves, operação em velocidades reduzidas, protocolos para interrupção em caso de acidentes e complacência.

Ao que integra a categoria de robôs cooperativos, definidos pela norma ISO/TS 15066:2016¹, como sistemas desenvolvidos para serem intrinsecamente seguros no uso em atividade em colaboração com pessoas. Por definição, são projetados para que seja oferecido o menor risco possível. Primeiro, estes robôs devem oferecer alguma forma de complacência, isto é deve ceder em caso de uma eventual colisão ou no por uma intenção diferente de movimento por parte de pessoas próximas evitando que a energia transmitida possa causar algum dano. Segundo, devem permitir operação em velocidades menores para reduzir o stress das pessoas na presença do robô. E com isto ainda ser capaz de executar a tarefa desejada.

Muito embora sempre exista a visão da tecnologia como vilã no roubo de empregos ao permitir ao uso de menos energia na execução de uma tarefa a redução do custo gera novas oportunidade por aumenta a capacidade de carga. Ainda que junto exista a necessidade de ressignificar o papel das pessoas no desenvolvimento de alguma atividade, deslocando o papel de total responsabilidade pela execução para incorporar papéis de planejamento pelo auxílio da automação, a incorporação de novas tecnologias traz um maior alcance dos produtos gerados. A partir do uso da tecnologia e da automação as pessoas migram do papel de agentes transformadores da matéria para agentes transformadores do processo permitindo que novos conhecimentos possam ser sempre incorporados.

1.1.1 Breve histórico

Baseado no humanoide COG em humanoides anteriores, em 2004 foi desenvolvido por Aaron Edsinger e Jeff Weber o robô Domo, mostrado na figura 1.1. Em seu projeto foram incorporadas diversas características visando permitir segurança através do uso de juntas complacentes com percepção de força. Bem como características para uso contínuo do ponto de vista mecânico como o uso de atuadores série elásticos baseados em fuso de esferas, proteção de superaquecimento para os motores e modularização do design. Para o controle são utilizados DSP para cada uma das juntas do robô[8].

¹<https://www.iso.org/obp/ui/#iso:std:iso:ts:15066:ed-1:v1:en>

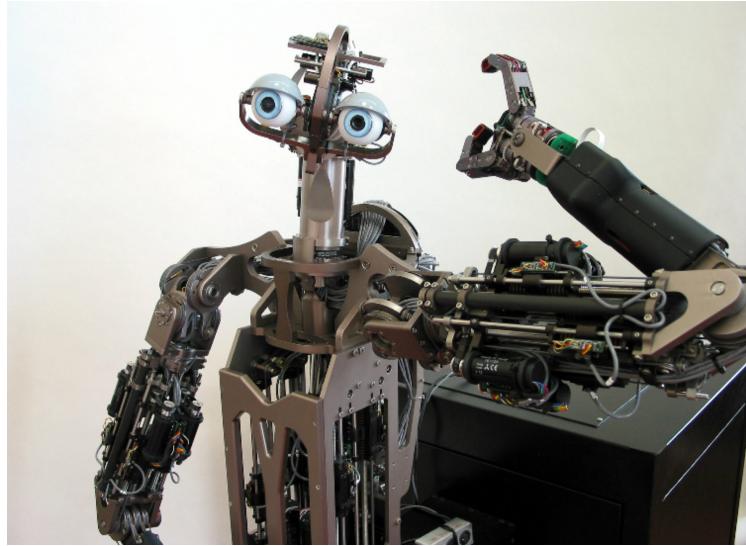


Figura 1.1: Foto Robô Domo (Fonte: <http://www.robotsvoice.com>)

Anos depois, os mesmos conceitos foram incorporados no robô Meka, figura 1.2, surgido como produto da empresa Meka Robotics fundada por Aaron e Jeff em 2006. Como pode ser notado pelas similaridades do ombro e design do braço em relação a figura 1.1. Este robô foi adquirido pelo LARA, porém em 2013 a empresa é adquirida pela Google e toda a documentação do Meka foi retirada do ar.

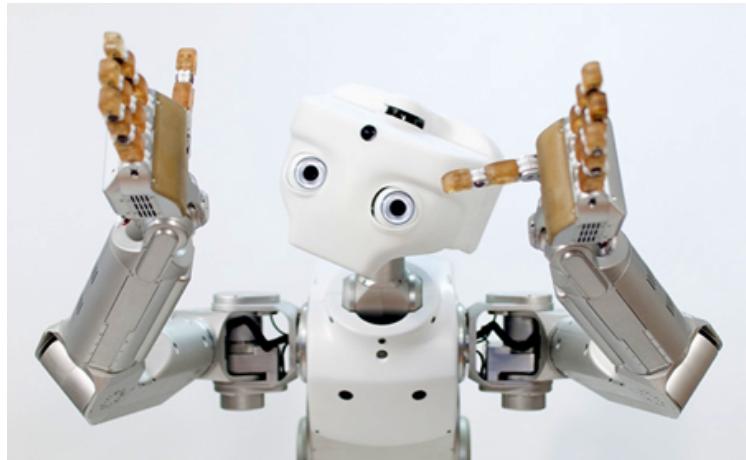


Figura 1.2: Meka M1 (Fonte: <https://spectrum.ieee.org>)

Da mesma linha de trabalhos surgiu a empresa Rethink Robotics fundada por Rodney Brooks, então professor do MIT em 1995. Estes aspectos estudados nos humanoides desenvolvidos no MIT como o COG e o DOMO permitiram que manipuladores robóticos pudessem ter um custo reduzido e atuar com um pouco mais de flexibilidade no ambiente, permitindo a presença de pessoa em proximidade.

Estes projetos tiveram particular importância pois muitos dos conceitos explorados vieram a compor patentes e futuramente partes dos robôs Baxter (Figura 1.3) e Meka (Figura 1.2) ,

respectivamente das empresas Rethink Robotics e Meka Robótics.

Embora com design completamente diferente, o Meka e o Baxter têm como semelhança o uso atuadores série elásticos em cada uma das juntas como elemento de segurança. Desta forma qualquer pessoa pode manusear os braços de ambos robôs além de proteger a estrutura em caso de colisão[9]. No entanto este sistema também confere ao robô uma maior suscetibilidade a influência de outros estímulos externos como a ação da gravidade na dinâmica do robô.

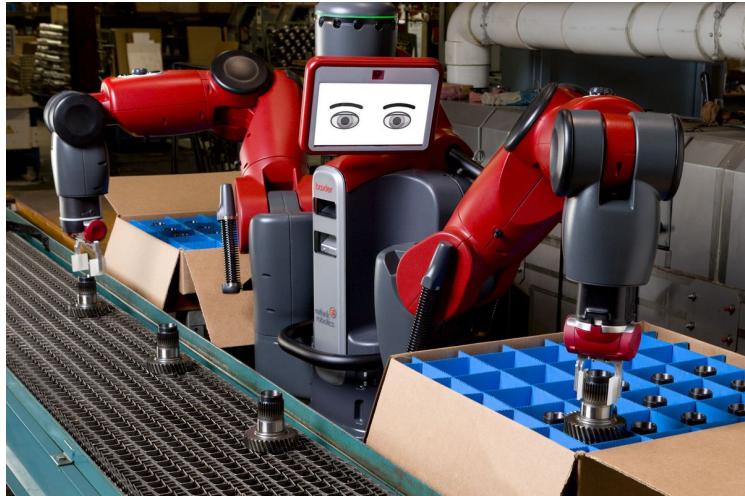


Figura 1.3: Baxter em linha de produção

A compensação da gravidade é resolvida de forma diferente em cada uma das plataformas. No Baxter a compensação da gravidade é feita por compensação mecânica por uma série de molas e via software através do modelo dinâmico em conjunto da biblioteca Orococos KDL[10]. E no Meka esta é feita apenas por software pela KDL.[5]

1.2 Definição do problema

No Laboratório de Automação e Robótica (LARA - UnB) encontra-se disponível o braço robótico Meka A2 fornecido pela Meka Robotics. Daqui em diante será denominado apenas Meka. O Meka é um braço complacente antropomórfico composto por sete juntas e uma garra desenvolvido para pesquisas na área de interação com pessoas. Em trabalhos anteriores do LARA [1] e [11] foi observado um desempenho ineficiente nos controles cinemáticos implementados em contraste ao resultado observado em outras plataformas robóticas e em simulação. Tal arquitetura não encontra-se devidamente documentada e detalhada. Desta forma, neste trabalho, propõe-se realizar uma caracterização mais detalhada desta arquitetura de controle para compreender sua implicação no desempenho dos controles cinemáticos testados e propor melhorias que resultem em trajetórias mais precisas levando em conta as características próprias do sistema.



Figura 1.4: Braço Robótico Meka A2 disponível no Lara [1]

1.3 Metodologia

Este trabalho foi desenvolvido em três etapas: estudo sobre teoria clássica de controle de manipuladores, revisão e acompanhamento de trabalhos anteriores feitos no LARA utilizando o Meka e por fim a investigação e experimentação usando a plataforma. Esta investigação foi feita a partir dos métodos de análise e síntese do processo. Por meio da análise foram estudados cada um dos componentes utilizados tanto de hardware como software em seu detalhes. Em conjunto alguns dos fenômenos foram sintetizados por meio de simulações visando ampliar a compreensão de quais fatores ocasionam ao Meka um desempenho inferior outros a braços robóticos avaliados pelos trabalhos de Murilo [12] e Marcos [1].

1.3.1 Revisão trabalhos anteriores

Inicialmente foi feito um estudo sobre os trabalho anteriores desenvolvidos na plataforma no Laboratório de Automação e Robótica (LARA - UnB) bem como estratégias clássicas no controle de robôs industriais. Alguns destes trabalhos puderam ser acompanhados ao longo da execução como o caso dos trabalhos desenvolvidos por Marcos Pereira [1] e pelo Rafael Koji [11] de modo a facilitar a transmissão do conhecimento relacionado as contribuições de cada um ao projeto bem como presenciar as dificuldades relacionadas ao controle do braço robótico.

1.3.2 Investigação e experimentação a partir da plataforma

Após o estudo preliminar foi adotado uma metodologia de investigação baseada em ciclos compostos por 3 etapas, descritas a seguir. Estas etapas são baseadas nos princípios de desenvolvimento ágil² e têm como objetivo acelerar e otimizar os esforços.

²<http://agilemanifesto.org>

1. Análise e Formulação de Hipóteses
2. Formulação de Testes para as Hipóteses
3. Avaliação de Hipóteses (Testes Experimentais e Estudo teórico)

Cada ciclo possui duração variável de acordo com o nível de aprofundamento necessário para satisfazer os objetivos levantados para cada momento do projeto. O objetivo central é ao final de um ciclo é trazer algum aprimoramento quanto ao entendimento do sistema ou ainda quanto ao comportamento final na realização da tarefa.

No inicio de cada ciclo é feito um estudo do sistema através da modelagem do sistema e da definição de uma métrica para avaliar o comportamento. Então o comportamento do sistema real é comparado com o modelo adotado e as expectativas de comportamento. Os objetivos são traduzidos em uma métrica, compondo um conjunto de indicadores que permita avaliar se tarefa foi executada e como foi o processo de execução. A cada novo aspecto avaliado podem ser introduzidos novos indicadores em conjunto aos antigos ou modificados.

Com base nisto são levantadas hipóteses quanto ao comportamento quanto ao fato de satisfazer ou não o modelo adotado e possíveis condutas para aprimorar os resultados dentro da métrica proposta. Para cada uma destas condutas é levantado um teste de verificação que pode incluir consulta bibliográfica, simulações a partir do modelo e experimentos com o Meka. Os resultados desta etapa são então levados para a etapa de observação e confrontados novamente e assim o ciclo se repete.

Resultados positivos são incorporados e resultados negativos avaliados quanto a possíveis correções. Tudo é registrado para permitir futuras avaliações. Como efeito foi feito um registro das atividades e todo material produzido, incluindo registro de dados dos experimentos e scripts usados para processar os dados, foi colocado no GitHub³ de forma a permitir o uso futuro na validação do comportamento após modificações futuras.

1.4 Estrutura do Documento

Este trabalho está dividido em 5 capítulos visando facilitar o entendimento. Estes estão organizados da seguinte forma:

- *Cap. 1 Introdução* : Breve Apresentação e Histórico sobre Robótica Colaborativa.
- *Cap. 2 Fundamentos* : Conceitos teóricos e tecnologias usadas no trabalho.
- *Cap. 3 Desenvolvimento* : Metodologia usada na investigação dos problemas e análise dos dados.
- *Cap. 4 Resultados* : Dados obtidos ao longo do trabalho.
- *Cap. 5 Conclusão* : Resultado final do trabalho e propostas de trabalhos futuros.

³www.github.com/lara-unb/Meka

Capítulo 2

Fundamentos

“Se eu vi mais longe, foi por estar sobre ombros de gigantes”(Isaac Newton)

Neste capítulo serão apresentados os aspectos teóricos e técnicos das tecnologias envolvidas no controle implementado no manipulador robótico Meka A2. Os quais estão dispostos na seguinte ordem: modelagem, sistemas mecânicos, embarcados e por fim sistemas de controle.

2.1 Manipuladores Robóticos

Segundo a ISO8373:2012¹, robôs são mecanismos de atuação programáveis em dois ou mais eixos com algum grau de autonomia, capazes de mover-se no ambiente em ordem de executar alguma tarefas. São dotados de sistemas de controle como forma de garantir a percepção e os acionamentos conforme solicitado. Manipuladores, por sua vez, são uma classe de robôs formados por uma série de segmentos conectados através de juntas ou com deslizamento entre si, com o propósito de pegar ou mover objetos de um lugar a outro.

Para o desenvolvimento deste trabalho foi utilizado o manipulador robótico Meka A2. Sendo composto por sete juntas com atuadores série elásticos. O Meka É um braço desenvolvido para pesquisa em tarefas em interação com pessoas de maneira segura em razão do baixo peso e da capacidade de ceder em contato com forças externas e por isto denominado complacente. O controle cinemático foi feito através de Quaternions Duaís, acrescentando mais uma camada ao controle em cascata das posições de junta já implementado.

Para controlar qualquer coisa é preciso definir um modelo matemático, que permita relacionar as características físicas às ações desejadas, permitindo assim que este seja comandado de alguma forma a partir das relações lógicas expressas no modelo usado. Em manipuladores robóticos o objeto de interesse para controle é o movimento no espaço como forma de permitir a execução da tarefa. Usualmente são adotadas duas abordagens denominadas cinemática e dinâmica.

Na cinemática estamos interessado apenas na geometria do movimento de pontos e objetos rígidos a partir da posição de cada ponto de referência e a variações no tempo, expressas como

¹<https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>

a velocidade e aceleração. Não são levados em conta as causas do movimento, isto é as forças e torques envolvidos.

Na representação dinâmica, em oposição a cinemática, analisa as causa e efeitos do movimento através do estudo das forças envolvidas. Desta forma outras características são levadas em consideração como distribuição de massa de cada um dos segmentos e a interação com a força externas como a gravidade ou a interação com outros objetos. O que a torna uma representação mais complexa e computacionalmente mais cara. E também demanda do lado dos atuadores o controle de torque no lugar do controle de posição, sendo reservado inicialmente para aplicações em que o controle do torque no efetuador é crucial para a execução da tarefa.

2.2 Quatérnions Duais

Quatérnions Duais são números duais em que as partes primárias e secundárias são quatérnios. São utilizados neste trabalho para as representações cinemáticas das posições e velocidades das juntas e do efetuador do braço.

2.2.1 Quatérnions

Quatérnios foram introduzidos em 1848 por Hamilton [13] como forma de permitir representar as rotações no espaço da mesma maneira que é feito com números complexos para rotações no plano através da multiplicação por uma potência do número imaginário \mathbf{i} . São definidos da seguinte forma:

$$H = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \quad (2.1)$$

Em que w, x, y e z são os quatro coeficientes do Quatérnion compostos por números Reais. e \mathbf{i} , \mathbf{j} e \mathbf{k} são unidades imaginárias para os quais são definidas as seguintes propriedades:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1 \quad (2.2)$$

$$\mathbf{ij} = \mathbf{k}, \mathbf{jk} = \mathbf{i}, \mathbf{ki} = \mathbf{j} \quad (2.3)$$

$$\mathbf{ji} = \mathbf{k}, \mathbf{kj} = \mathbf{i}, \mathbf{ik} = \mathbf{j} \quad (2.4)$$

Cada um dos coeficiente representa uma rotação em torno de um dos 3 eixos cartesianos. Observe que o produto não é comutativo, o que também é uma propriedade de rotações. O conjunto dos Quartérnions representa uma álgebra, constituindo um espaço vetorial com operação binária de multiplicação de vetores, propriedade distributiva e associativa sobre a soma de vetores.

2.2.2 Números Duais

Quatérnions tais os números complexos representam bem a magnitude e orientação no espaço, no entanto não permitem a representação de uma posição em particular [14], algo que é uma necessidade para algumas aplicações como o movimento de corpos rígidos. Para resolver este problema Clifford introduziu os números duais em 1871.

O conjunto dos números duais é definido como um conjunto em \mathbf{R}^2 que amplia as propriedades do números reais através da adição do elemento nilpotente $\eta^2 = 0$ em que $\eta \neq 0$. São representados de forma similar as números complexos:

$$Z = a + \eta b \quad (2.5)$$

Em razão de suas propriedades é uma forma de representação que vêm ganhando espaço em várias áreas com a robótica [15] e na computação gráfica para renderização [16] por permitirem o cálculo de forma mais eficiente com menos distorções.

2.2.3 Quatérnions Duais

Combinando os Quatérnions de Hamilton e os números duais de Clifford temos os chamados Quatérnions Duais. Supondo dois quatérnions Q_r e Q_d , um quatérnio dual pode ser definidos

$$H = Q_r + \eta Q_d \quad (2.6)$$

Onde

$$H = (q_{r_1} + q_{r_2}\mathbf{i} + q_{r_3}\mathbf{j} + q_{r_4}\mathbf{k}) + \eta((q_{d_1} + q_{d_2}\mathbf{i} + q_{d_3}\mathbf{j} + q_{r_4}\mathbf{k})) \quad (2.7)$$

A representação em quatérnions duais traz a vantagens em relação a outras representações usadas na robótica de eliminar as singularidades além de trazer uma representação mais compacta e portanto computacionalmente mais eficiente, uma vez que são utilizados 8 parâmetros ao invés de 12 em comparação com Matrizes Homogenias.[15]

2.2.3.1 DQ Robotics

Para este trabalho a representação do robô usando quatérnions duais e operações matemáticas relacionadas são feitas com auxílio da biblioteca DQ Robotics. Nela estão implementados as principais operações algébricas com Quatérnions Duais bem como alguns controladores nas linguagens Matlab, Python e C++.

2.3 Braço Robótico Meka A2

Cada uma das juntas do Meka A2 utiliza um atuador série elástico (Seção 2.4) composto por um motor Brushless em conjunto com uma redução baseada em engrenamento por onda de

deformação, comercialmente denominado *Harmonic Drive* (Seção 2.4.2). O controle é feito por placas desenvolvidas pelo fabricante comandadas por sua vez por um computador embarcado conectado via Ethernet. Como forma de entender melhor o comportamento do robô, cada um destes componentes foi analisado de forma separada quanto as suas características de resposta no controle da posição em cada junta.



Figura 2.1: Robô Meka - Vista interna (Fonte: <http://machinedesign.com>)

2.4 Atuadores Série Elásticos

Cada uma juntas do Meka é um Atuador Série Elástico. Estes são uma implementação de atuador complacente descrito por Pratt e Williamson [9]. São basicamente constituídos de uma mola colocada em série entre um atuador e o ponto de saída e visam permitir uma maior segurança na operação com objetos rígidos, bem como na interação com outros robôs e pessoas. Em atuadores rígidos, o uso de redução permite que uma alta velocidade do motor seja traduzida em um alto torque gerando uma grande inércia. Esta resistência a mudança no movimento por interferência externa representa um risco na execução de tarefas em conjunto com pessoas. Tendo em vista que um impacto nesta situação pode causar grande danos. Assim, quando ocorre uma colisão, muita energia é transmitida ao objeto de contato bem como ao dente da engrenagem de saída, resultando internamente em um fratura no mecanismo da junta do robô. Ao se colocar um elemento elástico como uma mola, parte desta energia é absorvida e distribuída gradualmente reduzindo assim a possibilidade de fratura.

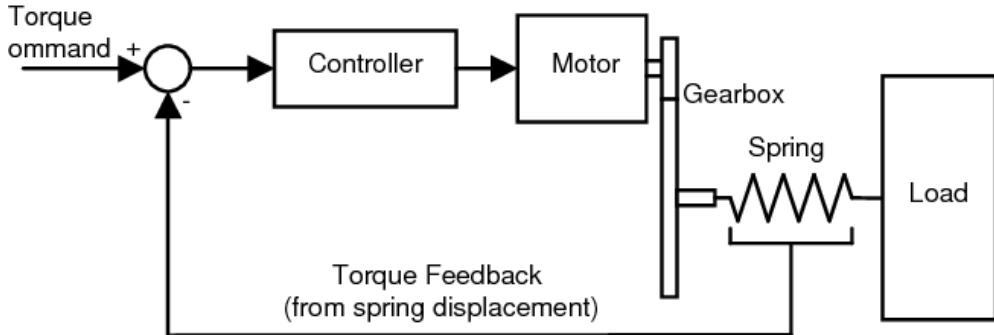


Figura 2.2: Topologia Atuador Rígido [?]

Por esta razão, introduzir alguma forma de complacência é, pelas normas ISO 10218-1:2011 e ISO/TS 15066:2016 um requisito de segurança para robô colaborativos (co-bots). Em particular, os uso de atuadores complacentes também facilita a interação entre manipuladores robóticos objetos rígidos, uma vez que a força percebida pelo atuador é reduzida pela contração interna da mola. O que por sua vez diminui a necessidade de alta precisão na operação como garantia de reduzir o esforço percebido pelo robô no contato com superfícies rígidas. Na figura 2.3 temos um atuador série elástico implementado por Pratt. Esta versão utiliza como redução engrenagens planetárias, uma mola plana e um motor dc, sinalizados na figura por a, b e c respectivamente.

Tendo que a mola está em série entre o atuador e a carga, a contração da mola corresponde diretamente a força aplicada pelo motor na saída. O torque fornecido pode ser então obtido pela lei de Hooke ($\tau = -k \cdot \Delta x$). Bastando que seja feita a medida da posição no atuador e na junta.

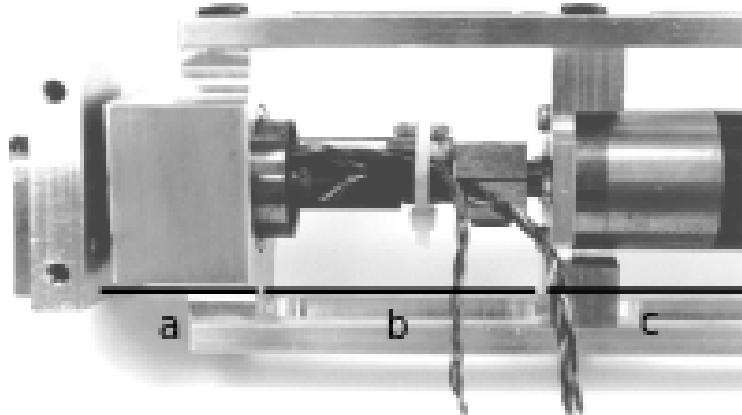


Figura 2.3: Atuador Série Elástico Implementado por Pratt [2]

E desta forma a partir de medida de posição da motor e da junta após a mola o torque de saída pode ser medido e permite a implementação de um controle de torque preciso e de baixo custo [2]. Em particular no Meka esta forma de atuação é implementado usando um motor Brushless em conjunto com um Harmonic drive e uma placa de controle DSP para controle do torque. O comportamento elástico é adicionado pela soma da elasticidade do Harmonic Drive e demais componentes na transmissão de cada junta e modulado pelo controlador de torque implementado por

placas DSP embarcadas dentro do braço. Desta forma a rigidez do atuador poder ser controlada via software como uma mola virtual. Em que a rigidez da mola é toda emulada por um sistema de controle a partir da referência do controle de posição e do controle de torque. Em resumo, quando a posição varia em relação a referência o controlador de baixo nível aumenta o torque de maneira proporcional ao erro da posição emulando o comportamento de uma mola, porém com rigidez totalmente ajustada pelo controlador. Desta forma a complacência e a rigidez pode ser ajustada dinamicamente por software [17].

Da mesma forma que o acréscimo de uma mola introduz uma maior segurança e permite um excelente controle de torque, traz o custo de uma menor precisão no controle de posição e algumas características dinâmicas não desejadas como oscilações e respostas mais lentas. Um diferencial do Meka em seu lançamento é o uso de sistemas complacentes em todas as juntas ao invés de apenas em relação ao efetuador, conferindo mais segurança no contato com superfícies rígidas como em operação na proximidade de pessoas.

2.4.1 Encoder

Para cada junta são utilizados encoders para a medidas das grandezas a posição e o torque aplicado. Um encoder é tradicionalmente um sensor de posição. No entanto por conta da estrutura do atuador série elástico é possível avaliar também o torque aplicado na junta levando em conta a dinâmica da mola, ou componente elástico do atuador. Desta forma, seguindo a lei de Hooke, o torque avaliado será a diferença da medida de posição da saída do motor e da junta vezes a rigidez da junta.



Figura 2.4: Encoder utilizado nas juntas do robô

2.4.2 Harmonic Drive

Harmonic Drive, também denominado “Engrenamento por ondas de deformação”, é um sistema mecânico para redução de velocidade baseado no deslizamento de uma membrana flexível em torno

de um engrenamento. Na figura ?? temos detalhados os 3 componentes principais: as duas partes rígidas denotadas na figura como *Circular Spline* e *Wave Generator* e parte flexível *Flex Spline*. Com a rotação do gerador de ondas a membrana flexível vai deslizando e movendo ao longo do engrenamento da parte mais externa gerando uma rotação em uma velocidade reduzida em relação ao velocidade de entrada. Na figura 2.5 é apresentado o encoder utilizado em cada uma das juntas do robô.

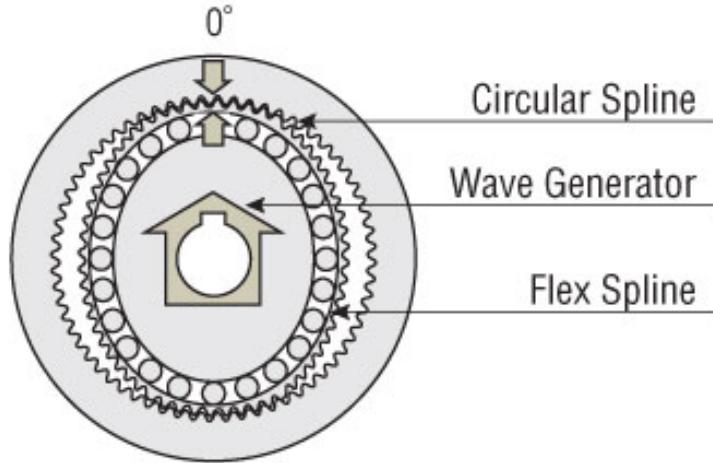


Figura 2.5: Harmonic Drive (Fonte: <https://www.orientalmotor.com>)

Por ter baixo deslizamento e folga junto a uma alta capacidade de redução em um tamanho compacto é um tipo de transmissão bastante comum na área de robótica. No entanto acrescenta atrito e flexibilidade às juntas. Esta característica de flexibilidade permite uma implementação extremamente compacta de um atuador série elástico, desde que rigidez possa ser modelada.

2.4.3 Motores Brushless

Motores de corrente continua sem escovas (comumente motores Brushless) são motores síncronos controlados via corrente contínua. Oferecem baixa manutenção e alta performance pela redução do atrito entre as partes internas em relação a motores DC tradicionais. Tipicamente motores síncronos são controlados via corrente alternada e rotação é sincronizada com a frequência da tensão aplicada. Em um motor Brushless é utilizado um inversor ou uma fonte chaveada para converter o sinal elétrico de corrente contínua em um sinal de corrente alternada com uma frequência definida. Desta forma, em conjunto com um circuito de controle interno em malha fechada, a velocidade pode ser ajustada e mantida com precisão. Nas figuras 2.6 e 2.7 são mostrados dois dos motores utilizados no Meka.



Figura 2.6: Motor utilizado nas juntas do Ombro



Figura 2.7: Motor utilizado nas juntas do braço

Em razão dos componentes eletrônicos usados para controle interno, motores Brushless são mais caros que motores magnéticos permanentes. No entanto, apresentam um custo de manutenção inferior uma vez que as partes internas estão sujeitas a um menor desgaste por atrito. Permitem um controle de velocidade preciso através do sensores de corrente por efeito Hall. Tendo ganhado espaço por garantirem um controle de velocidade melhor por um custo menor em comparação com servo-motores além de serem menores.

2.4.4 Pulso Diferencial

Para o movimento do pulso é utilizado um mecanismo diferencial (figuras 2.8), baseado em dois atuadores de forma que os movimento de *Pitch* e *Yaw* do efetuador estão sempre acoplados. Para se mover em *Pitch* são acionados ambos motores na mesma direção e para *Yaw* estes são acionados em direções opostas. Um controlador no PC é utilizado para separar os movimentos dentro antes do envio ao DSP.

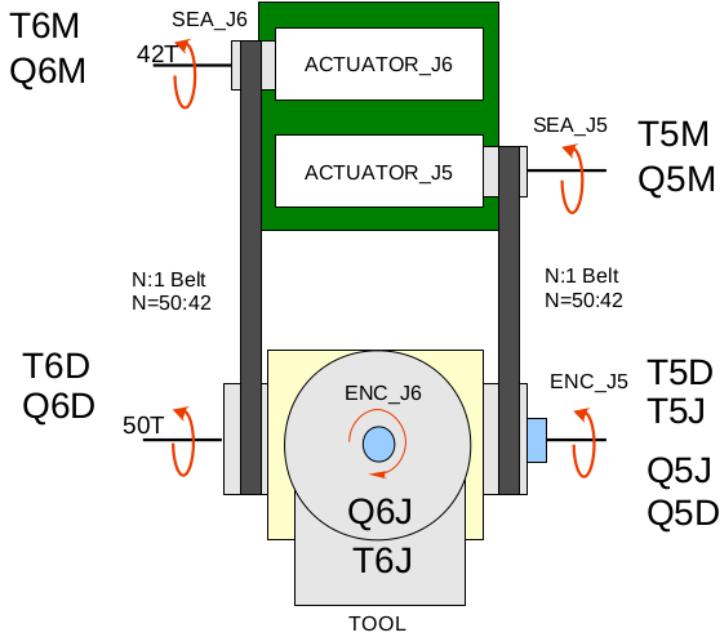


Figura 2.8: Diagrama mecanismo diferencial do pulso [3]

2.5 Plataforma de Controle

O Meka é controlado pelo conjunto de dois sistemas: um embarcado diretamente no braço para o controle das juntas e um externo sendo executado em PC com Ubuntu 12.04 com Kernel RTOS Xenomai em conjunto do sistema M3 da Meka Robotics e do framework ROS. No sistema embarcado são feitas todas as interfaces com os sensores e atuadores. Enquanto no PC é feito o controle de cada uma das juntas e demais tarefas. A comunicação entre estes dois se dá por uma porta Ethernet e opera no protocolo proprietário EtherCAT. Desta forma o sistema pode ser facilmente expandido com o acréscimo de novos dispositivos e sensores.

2.5.1 DSP Control Boards

O controle de cada junta a partir do acionamento dos motores e da leitura dos sensores é feito pela placa DSP desenvolvida pela Meka Robotics. Nela estão implementados o controle de posição, velocidade, torque e rigidez de cada atuador. Cada uma das placas possui uma interface de comunicação EtherCAT ligada a um concentrador dentro do tronco do robô. Sendo este ligado através da porta Ethernet a um computador para comando de cada uma das partes do robô.



Figura 2.9: Destaque Placas de controle entre as juntas do pulso e o cotovelo do braço

O uso de DSP e um barramento de comunicação de auto desempenho no lugar facilita o controle em tempo real em relação a controles baseados em PC, além de oferecer modularidade e velocidade de inicialização[8]. Desta forma os controles implementados no PC podem operar em frequências menores.

2.5.2 Meka PC

Toda a parte de controle é feita no PC. Para iniciar a comunicação do PC com o robô é executado um script que dispara os processos relacionados a comunicação com o robô via Socket pelo protocolo EtherCAT e a interface para Python. Um vez que o script está em execução o robô pode ser controlado por uma das interfaces da M3 entre elas a API em Python ou pelo ROS, conforme ilustrado no diagrama apresentado na figura 2.10.

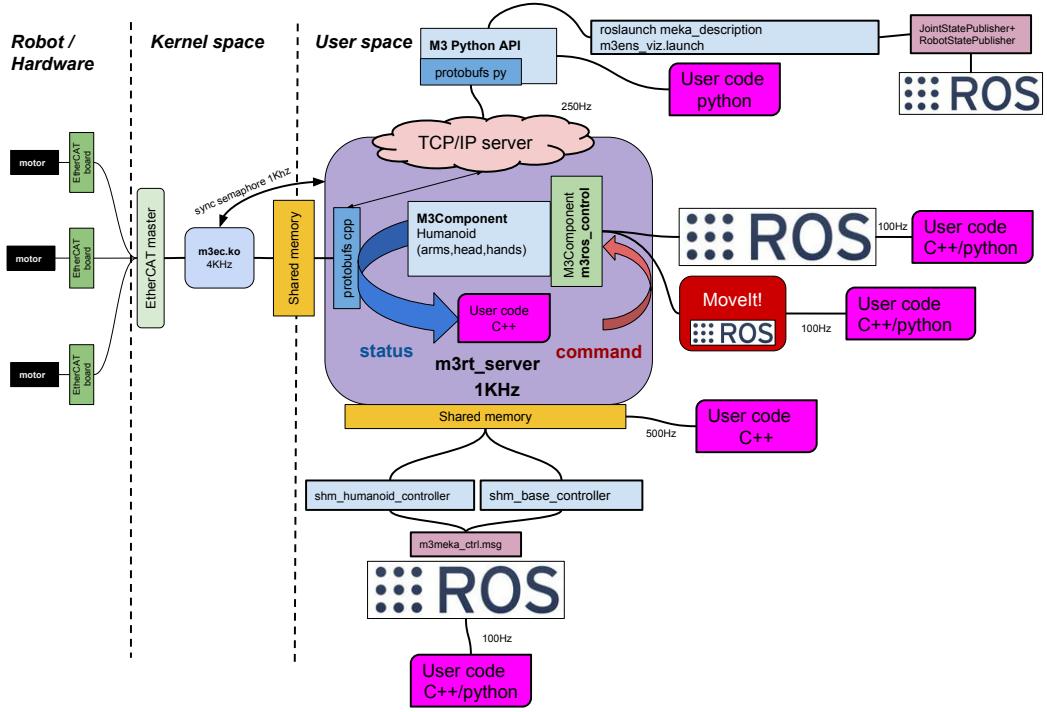


Figura 2.10: Diagrama da arquitetura do robô [4]

2.6 M3

M3 é o sistema de controle em tempo real para os robôs desenvolvidos pela Meka Robotics. Representa o core do robô em termos de controle e responde por todos os acionamentos, sensores e ações do Meka. É bem extensa e baseada em módulos visando permitir a integração de múltiplos tipos de hardware. Possui código aberto e atualmente este encontra-se todo disponível nos repositórios do LARA² e da Ensta Paris³.

²https://github.com/lara-unb/Meka/tree/master/meka_robots_source/mekabot

³<https://github.com/ahoarau/mekabot>

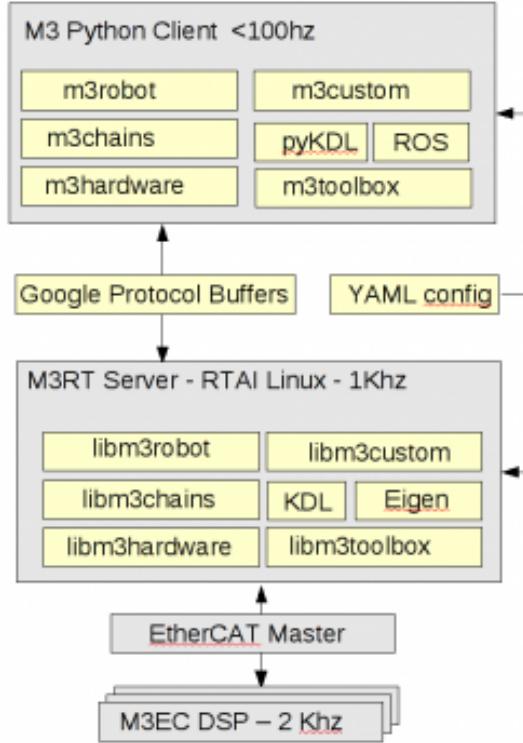


Figura 2.11: Arquitetura sistema de controle M3 [5]

2.6.1 Arquivos de Configuração

Todas as configurações do Meka são guardadas através de arquivos *.yml*, localizados dentro da pasta *config* de cada diretório. Em particular, dentro da pasta *m3ene* estão tudo que é particular de configuração, drivers e ajuste para o funcionamento do robô do LARA. Esta pasta é baseada na *m3ens* da Ensta Paris. Nela consta tudo para que o hardware completo disponível do robô e respectivos controladores funcionem, incluindo informações sobre sensor de força, atuadores, garra e parâmetros do modelo dinâmico do robô e da garra.

2.6.2 Controladores

O controle do Meka é feito em estrutura de cascata por uma série de controladores interagindo entre si. Cada nível opera na execução de uma tarefa possibilitando ao nível superior uma maior abstração do sistema e do objetivo a ser cumprido. Para garantir a estabilidade é preciso a dinâmica de cada controlador seja mais rápida. São estes, ordenados da dinâmica mais rápida para a mais lenta:

1. Controladores do Motor (velocidade e torque)

2. Controle do Atuador Série Elástico (posição, velocidade e torque)
3. Controle de Posição do Efetuador (Modelo Cinemático)
4. Controlador de Trajetória

No Meka estes estão implementados a partir de uma série de controladores em cascata executados no PC descritos da figura 2.12.

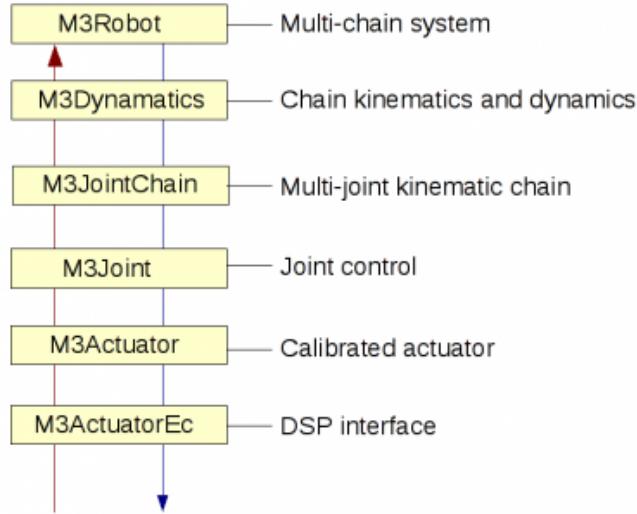


Figura 2.12: Arquitetura dos controladores do Meka [5]

A compensação da gravidade é feita a partir do cálculo do modelo dinâmico pela *M3Dynamics* e em seguida passada como uma ação de controle em *feedforward* para o torque das juntas em *M3Joint* para ser por fim executado em *M3ActuatorEc* [5].

Capítulo 3

Desenvolvimento

Simple, clear purpose and principles give rise to complex, intelligent behavior. Complex rules and regulations give rise to simple and stupid behavior. (Hock Dee)

3.1 Desempenho de Sistemas Mecatrônicos

Em trabalhos anteriores foram propostos e implementados vários controladores cinemáticos como forma de corrigir o efeito das pertubações devido a interação com uma agente externo e da ação da gravidade. No entanto o desempenho foi abaixo do apresentado pela código de demonstração do fabricante para o robô e pelos mesmos controladores no uso em outra plataforma explicitando a necessidade de um maior estudo da arquitetura do manipulador robótico Meka A2. Com base nisto foi feito uma análise dos componentes do braço e a implementação dos sistemas de controle das juntas em software e hardware. Neste capítulo será descrito os métodos de investigação utilizados bem como experimentos efetuados.

Em [1] Marcos propôs diversas métricas e avaliar os controladores cinemáticos implementados. Enquanto foi possível um excelente detalhamento da desempenho de cada um dos controladores, foi descoberto limitações na interação com a plataforma. Neste trabalho será analisado o robô enquanto sistema observando as implicações dos componentes mecânicos, sistemas embarcados e sistemas de controle operando de forma conjunta. O objetivo é avaliar dentro do que já foi implementado os limites de desempenho permitidos pela plataforma para permitir um melhor desempenho em trabalhos futuros.

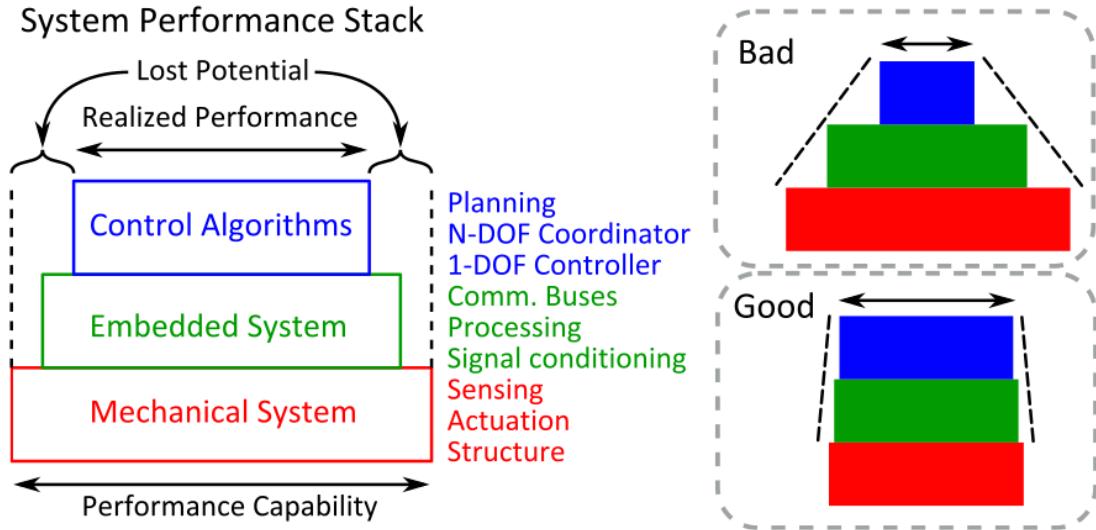


Figura 3.1: Representação do Desempenho de um sistema [6]

Por se tratar de um sistema mecatrônico, naturalmente existem 3 camadas: Sistema Mecânico, Sistema Embarcado e Algorítimos de Controle conforme ilustrado na figura 3.1 proposta por N. A. Paine em [6] para o desenvolvimento de um sistema série elástico de alta performance. Para o Meka estas camadas estão implementados na seguinte forma:

- Sistema Mecânico (Atuadores Série Elásticos)
 - Motores Brushless
 - Harmonic Drive
- Sistema Embarcado (Atuadores Série Elásticos)
 - Placa DSP para controle dos motores
 - Hub EtherCAT
 - Interface EtherCAT para o computador
 - PC com Linux Ubuntu 12.04 e Kernel RTOS Xenomai
- Algoritmos de Controle
 - Controle de Torque (DSP)
 - Controle de Posição com Compensação da Gravidade (PC)
 - Interface ROS
 - Controle Cinemático
 - Gerador de Trajetória

Como forma de alcançar um maior desempenho com o Meka foram avaliadas cada parte quanto a possíveis limitações e assim orientar trabalhos futuros no controle do robô.

3.2 Avaliação Preliminar

Inicialmente foi feito um estudo a partir do código de demonstração do fabricante feito em Python para avaliar quais as possíveis forma de controlar o braço. Em que foi levantando todos os controladores implementados na biblioteca m3 e testados individualmente através da API em Python. Foram analisados os seguintes controladores disponíveis pelo M3:

- Controle de Posição
- Controle de Posição com compensação da gravidade
- Controle de Torque
- Controle de Torque com compensação da gravidade
- Controle de Velocidade

Para avaliar o comportamento em conjunto dos controladores cinemáticos apenas o controlador de posição com compensação da gravidade foi utilizado, uma vez que este é usado pelo ROS.

3.3 Estudo Controladores Cinemáticos

Para definir um ponto de referência para os ensaios e testes foram avaliados os controladores implementados por Marcos Pereira. A partir da melhor configuração para cada um dos controladores foram feitos experimentos ajustando os parâmetros da velocidade de atuação e nível rigidez para avaliar a influência nos resultados. Os experimentos foram executados com base em duas trajetórias pré-definidas: deslocamento em linha reta na vertical e o desenho de um quadrado a partir de dois deslocamentos na vertical e dois na horizontal.

Nos experimentos de deslocamento em linha reta foram avaliados o intervalo de tempo até robô começar a responder e o controle estabilizar para diferentes taxas de amostragem da trajetória. Enquanto nos experimentos com a trajetória de quadrado foram avaliadas a resposta do robô para diferentes valores de ajuste dos controladores de junta quanto aos parâmetros de velocidade e rigidez.

O estudo dos tempos de atuação foi obtido repetindo o mesmo experimento para o deslocamento em linha reta na vertical em diferentes condições de velocidade e rigidez. Neste experimento foi estudado o tempo necessário para o atuador começar a se mover dado um comando bem o tempo necessário para o controle convergir.

3.4 Layout Experimentos

Como forma de avaliar a resposta do sistema para controle dos ângulos de junta foi proposto um teste com o uso de um degrau de referência de entrada em malha fechada. Este experimento

foi efetuado utilizando as interfaces da M3 definidas pela API em Python e pelo ROS. Para ambos casos os dados obtidos foram registrados em arquivo utilizando a ferramenta rosbag com auxílio do nó *shm_humanoid_interface*.

3.4.1 ROS

Robotics Operating System (ROS) é um *middleware* desenvolvido para a robótica. Desenvolver um robô é uma tarefa muito complexa para uma pessoa só ou mesmo um grupo de pesquisa, de modo que o ROS atua conectando diversos *frameworks*, bibliotecas permitindo o uso de uma variedade de tipos de hardware e linguagens de programação [18]. De igual forma atua conectando toda a comunidade de robótica em uma linguagem comum. Seu desenvolvimento começou em 2007 reunindo conceitos de diversos projetos de software aberto existentes até então e com o passar dos anos se tornou um padrão dentro da comunidade, contando com implementação para diversos robôs comerciais e inclusive uma versão completa voltada para a indústria.

Como exemplo temos o módulo *MovitIt* (figura 3.2) que reúne várias aplicações para facilitar o planejamento de trajetórias bem como a definição dos parâmetros de ajuste dos controladores com auxílio de métodos heurísticos através da biblioteca OMPL [19].

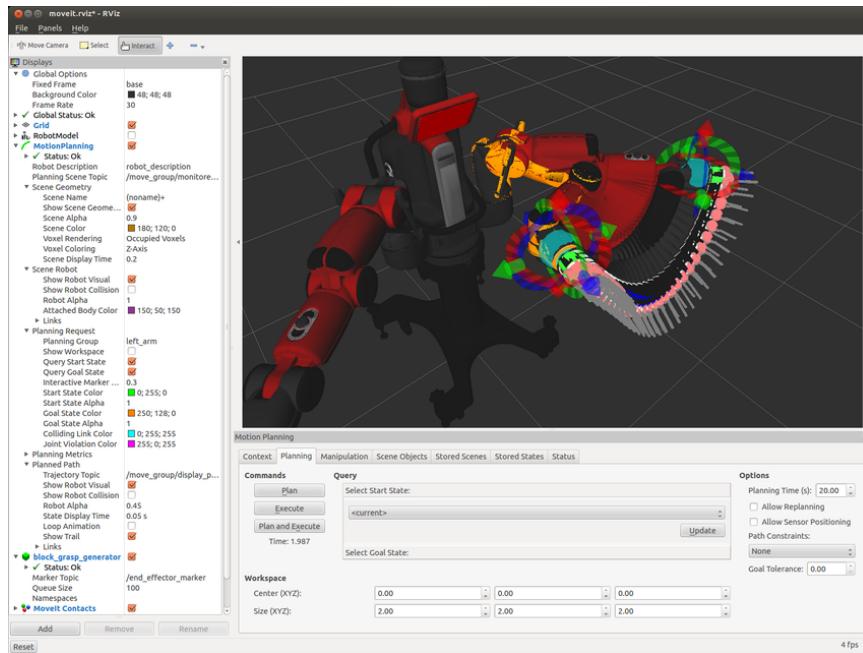


Figura 3.2: Planejamento de Trajetória no Baxter com Auxílio do MoveIt [7]

3.4.2 *shm_humanoid_interface*

Este é um nó do ROS disponibilizado pela Meka Robotics para a comunicação via ROS com o Meka. Representa uma das possíveis interfaces com a M3 através do acesso direto a memória compartilhada. Em 2.10 são detalhados todas as possíveis interfaces. Através deste nó são gerados dois tópicos no ros: */humanoid_state* e */humanoid_command*. A cada interação a informação

atual do tópico */hummanoid_command* é passada para a memória compartilhada e a leitura dos sensores é então escrita no tópico */hummanoid_state*. Na versão atual são somente interpretados os comando de posição das juntas através dos modos de controle com e sem compensação da gravidade.

3.4.3 Rosbags

Rosbags¹ é um utilitário do ROS para o registro dos eventos em cada tópico. A informação publicada nos tópicos é registrada em formato *yaml* e permite que posteriormente possa ser reproduzida em tópicos tal uma simulação. Desta forma dispensa a necessidade de manter o hardware conectado para analisar o comportamento do robô ao longo do tempo bem como permitir a análise de diferentes experimentos diretamente pelas ferramentas do ROS como o *ros_plot*. Ou ainda, permite conectar os resultados de um hardware real com uma simulação como forma de efetuar uma validação cruzada do comportamento real em comparação do comportamento simulado. Para tal basta executar o programa e indicar quais tópicos serão monitorados. Ao final da execução é registrado um arquivo com as informações com a extensão *.bag*. Internamente os dados são registrados em formato *.yaml* o que permite a leitura direta.

Neste trabalho todos os experimentos foram registrados usando rosbag para permitir o uso futuro por outras pessoas. No entanto, como a representação da informação é um pouco verbosa gerando um arquivo grande, foi utilizado o programa *bag2csv* para converter para o formato *.csv* para facilitar o processamento. Em seguida o programa *sed* para pequenos ajustes e a separação da leitura dos tópicos */hummanoid_state* e */hummanoid_command* em diferentes arquivos. Este processo foi incorporado a um Makefile para permitir o processamento rápido de todos os arquivos *.bag* utilizando o recurso de processamento paralelo da ferramenta *GNU make*².

3.4.4 Python API

A M3 possui uma interface através de uma API em Python que permite uma liberdade maior de controle uma vez que possui mais recursos já implementados. Além dos modos de controle por posição estão também disponíveis o controle direto por torque e por velocidade das juntas e o controle por PWM dos motores. Na M3 estão implementados um controlador para compensação da gravidade para o controle de torque e de posição a partir da biblioteca KDL com base no momento dinâmico do braço descrito no arquivo *m3ene/meke_doc/ma26/m3dynamics_right_ma26.yaml*. Este arquivo pode ser alterado para permitir o controle com um atuador com peso diferente, sem ter que alterar os parâmetros dos controladores.

¹<https://wiki.ros.org/rosbag>

²https://www.gnu.org/software/make/manual/html_node/Parallel.html

Capítulo 4

Resultados

“De forma mais simples, uma vez que cada pedaço de matéria no Universo é, de alguma forma, afetado por todos os outros pedaços de matéria do Universo, é teoricamente possível extrapolar a totalidade da criação – cada sol, cada planeta, suas órbitas, sua composição e sua história econômica e social a partir de, digamos, um pedaço de pão-de-ló.”(Guia do Mochileiro da Galáxia, Douglas Adams)

4.1 Estudo Arquitetura

Inicialmente foi feito o estudo do código e documentação em busca de identificar tecnologias e algoritmos envolvidos no controle do Meka tendo como intuito identificar possíveis referências que ajudasse a guiar os experimentos a serem realizados e eventuais pontos críticos para o desempenho. Como a documentação oficial do Meka foi tirada fora do ar e apenas estão disponíveis as documentações geradas pela comunidade, foi feito um estudo para retirar possíveis pistas através do conhecimento deixado no código dentro do robô. Tal foi somente possível, pois por se tratar de uma plataforma voltada para pesquisa todo o código fonte e documentação estava disponível dentro do PC auxiliar.

4.1.1 Sistema Mecânico

Foram avaliados as limitações de atuação e sensoriamento através das especificações fornecidas na documentação. Estes valores, embora possam não corresponder a situação atual do sistema servem como um primeiro parâmetro para orientar as análises feitas posteriormente pelo outros métodos. A documentação original do Meka foi obtida através de arquivos de backup armazenados no PC de desenvolvimento, uma vez que o site oficial está indisponível.

4.1.1.1 Atuadores

A documentação oficial disponibilizada no site original da Meka Robotics encontra-se atualmente disponível apenas no PC uma vez que o site está fora do ar desde a compra da empresa pela Google em 2013. A partir da antiga documentação temos os seguintes parâmetros da especificação do robô, registrados na tabela 4.1. Nesta tabela são indicando os limites permitidos pela plataforma segundo especificação de projeto, obviamente com o desgaste e o tempo estes valores podem sofrer alterações.

Tabela 4.1: Especificação Preliminar dos Atuadores, adaptado de [5]

A2.R4 Spec	Ombro		Braço		Antebraço		Pulso
	J_0	J_1	J_2	J_3	J_4	J_5	J_6
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
Rigidez (Nm/rad)	417.0	417.0	190.0	190.0	23.0	46.0	46.0
Redução HD (:1)	120.00	120.00	100.00	100.00	100.00	50.0	50.0
Redução Correia (:1)	1	N/A	N/A	N/A	N/A	1.19	1.19
Redução Engrenagens (:1)	120.00	120.00	100.00	100.00	100.00	59.52	59.52
Corrente Pico (A)	11.0	11.0	5.4	5.4	0.8	0.8	0.8
Corrente Cont. (A)	5.8	5.8	3.3	3.3	0.5	0.5	0.5
Velocidade Max. (Rad/s)	4.6	4.6	4.5	4.5	2.0	3.4	3.4

Da tabela 4.1 podemos perceber que os motores de cada junta possuem diferentes limites de velocidade e atuação de modo que um mesmo comando de velocidade pode levar um motor de uma junta a saturação enquanto outro ainda possui uma boa faixa de operação. Notadamente a maior diferença está entre a junta do pulso (J_4) em relação as demais. Como a rigidez também varia em cada junta, podemos perceber que as juntas do pulso (J_4 , J_5 e J_6) são bem mais sensíveis a perturbação que as demais.

Por ser um braço antropomórfico, o grupos de juntas do pulso (J_4 , J_5 e J_6) e do ombro (J_0 , J_1 e J_2) acabam atuando em composição como uma junta esférica respondendo pela orientação do ponto de referência no centro das juntas. Para uma tarefa de translação de um ponto de referência localizado no pulso estas não atuam. Enquanto para o caso do deslocamento de um efetuador acoplado ao pulso não há necessidade de velocidades grandes como forma de garantir a tarefa de posicionamento no espaço, uma vez que o erro/deslocamento é pequeno em comparação as juntas do ombro.

4.1.1.2 Sensores

Com base na documentação temos que não existe nenhuma forma de medida da velocidade, esta é apenas estimada a partir de diferenças finitas pela medida posição diretamente no PC. De modo que a medida final obtida para velocidade é replata de ruídos que por sua vez são atenuados dentro da M3 utilizando um filtro butterworth. Na tabela 4.2 são relacionados todos os sensores disponíveis.

Tabela 4.2: Especificação dos Sensores, adaptado de [5]

Sensor	Descrição
Força	ContLec Vert-X13 Absolute Encoder; 14 bits
Angulo Motor	N/A
Ângulo Junta	ContLec Vert-X13 Absolute Encoder; 14 bits
Corrente Motor	U-V channel low-side 50Khz sampling; 12 bits
Temperatura	Temperatura ambiente dentro de cada link
Temperatura Motor	N/A; Derivada da corrente, temperatura ambiente e modelo
Força e Torque no Puslo	Sensor de 6 Eixos ATI Mini-40E

Como em um atuador série elástico a medida do torque é feita de maneira indireta, relacionando a rigidez da junta e a medida de deslocamento nota-se o uso do encoder como sensor de força. A sensibilidade a uma força externa varia conforme a rigidez da junta, uma vez que todas utilizam tipo de sensor. Tendo que as juntas relacionadas ao movimento do pulso $J4, J5, J6$ são de menor rigidez e serão também mais sensíveis as perturbações. Além disto o pulso utiliza um mecanismo diferencial de modo que movimentação pura nas direções *Pitch* e *Roll* são obtidas através de um controlador aliando o movimento de dois atuadores.

4.1.2 Sistema de Controle

O código opera a partir de uma estrutura de controle em cascata com uma parte embarcada diretamente no manipulador operando em alta frequência para o controle do torque, uma camada operando em tempo real (1Khz) no PC em comunicação com o robô a partir da biblioteca M3 e por fim uma camada em soft real time operando a partir da interface com o ROS ou com a API em Python. Na figura 4.1 é detalhado alguns aspectos da arquitetura observados.

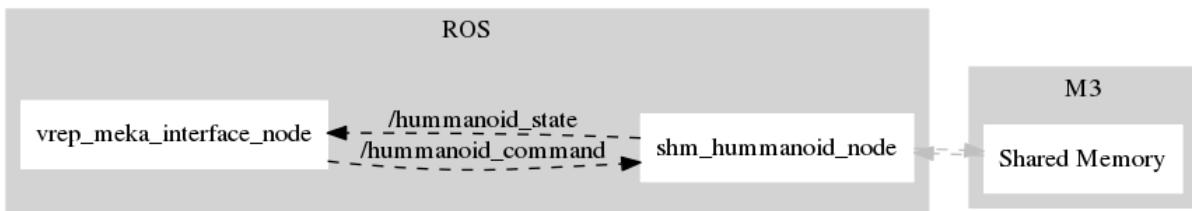


Figura 4.1: Arquitetura Controle C++

Todo o código relacionado ao controle do robô é disponibilizado pelo fabricante e compõe a biblioteca M3, implementada em Python e com interface em C++. Dentro do programa de interface com o ROS pela análise do código foi possível observar alguns pontos importantes dos controladores controladores de posição com e sem compensação da gravidade. Os valores de velocidade e torque informados para o controle são completamente ignorados. Além disto a taxa de envio da informação obtida pelo ROS para a memória compartilhada é de apenas $100Hz$ definidos *hardcode*. Foi feito um teste para alterar este valor para $1KHz$ porém o robô ficou completamente instável. De novo que testes similares não foram repetidos para evitar danos ao robô.

Notadamente o valor de $100Hz$ é bem próximo das taxas de amostragem de $8ms$ ($125hz$) e $20ms$ ($50Hz$) utilizadas pelos controladores implementados em C++ obtidas para os melhores resultados em [1] obtidos por Marco Pereira. Sendo estes valores definidos pelo tempo mínimo necessário para efetuar todos os cálculos relacionados ao controladores cinemáticos implementados usando quatérnions duais.

4.2 Estudo Controladores M3

O sistema de controle M3 fornece diversos controladores e incorporados a plataforma. Para avaliar o desempenho dos controladores já existentes foram feitos alguns experimentos, detalhados a seguir.

4.2.1 Código de Demonstração

São disponibilizados alguns códigos de exemplo em Python e C++, espalhados ao longo do código fonte da M3. Para estudo sobre controlar o robô a partir do código em Python foi tomado como referência o estudo dos códigos de demonstração e a documentação disponibilizada no site ReadTheDocs¹. Neste processo foi adicionada mais uma pose no código de demonstração com o nome de referência *Kojiref* com os valores usados para a posição inicial pelo Rafael Koji em [11].

Para referência do uso dos controladores foi tomado como parâmetro inicial o código de demonstração sobre o controle de rigidez disponibilizado pelo fabricante, cuja interface gráfica é mostrada na figura 4.2. Na interface estão disponíveis algumas poses para mostrar o robô em posição estática, uma trajetória montada por uma sequência de poses e uma trajetória de um círculo elaborada a partir de splines. É possível ajustar a velocidade máxima e a rigidez do controlador de forma dinâmica, permitindo a avaliar o impacto na resposta do robô.

¹<https://mekabot.readthedocs.io/en/latest/>



Figura 4.2: Interface Código de Demonstração [5]

A partir da execução do código de demonstração foi possível perceber que a posição das juntas varia conforme o valor de rigidez é ajustado. O valor da velocidade máxima e da rigidez variam de 0 a 1 internamente do código representando uma porcentagem do valor máximo possível. Ao que foi notado que mesmo com a rigidez no máximo o braço continua demonstrando um comportamento complacente. Com a rigidez em zero o controle é instável e o robô não consegue manter a posição e para valores abaixo de 0.2 o desempenho na execução de uma trajetória é visivelmente ruim.

4.2.2 API Python

Como a maior parte do código da M3 está implementada em Python, para avaliar os controladores de junta de baixo nível foi utilizado a API para Python por ser a opção mais completa em termos de recursos. Sendo utilizada para todos os testes iniciais. No entanto, para registrar os valores do sensor foram registrados via *rosbag* a partir das leitura do tópico */hummanoid_state*. Foi escolhido usar o programa *rosbag* em conjunto com o nó de ros *shm_hummanoid_interface* no lugar do registro direto em arquivo para facilitar a reproduzibilidade dos experimentos e análises feitas. Através interface da API em Python para a M3 estão disponíveis os seguintes controladores, ordenados do de mais alto nível para o de mais baixo nível:

- Posição
- Velocidade
- Torque
- PWM

Os modos de controle de posição estão ainda disponíveis com e sem controle de compensação da gravidade a partir do modelo dinâmico inverso da M3 e obtido pelo algoritmo de dinâmica inversa de cadeia cinemática da biblioteca Orocos KDL².

²<http://www.orocos.org/KDL>

4.2.2.1 Avaliação do Atraso de Comunicação

A comunicação com o robô é feita através de uma memória compartilhada. Assim, o primeiro experimento foi levantar o tempo gasto pela instrução *proxy.step()* que atualiza esta memória pegando as medidas dos sensores e informando os comandos para os controladores. Este teste foi feito através avaliando o intervalo de tempo entre cada chamada após sucessivas interações tendo como resultado o valor médio de 16ms entre cada chamada indicando que a frequência máxima possível para um controlador implementado com base na API é de $f = 1/16ms = 62.5Hz$. Isto, obviamente, desconsiderando qualquer tempo extra gasto com as operações realizadas pelo próprio controlador.

4.2.2.2 Avaliação do Acumulo do Erro

Para verificar o acumulo do erro após sucessivas interações foi feito o teste de fechar a malha com braço na posição de repouso. Isto é, controle deveria ler os dados atuais dos sensores e passar diretamente para o robô na expectativa de que o robô ficasse parado. No entanto foi observado que o braço começou a subir lentamente, indicando valores cada vez maiores para os ângulos medidos e uma distância cada vez maior da posição de origem.

Por outro ao ser enviado a mesma posição de referência suscetivamente, o braço desloca até o ponto desejado e permanece parado, mostrando que o controle para um referência em degrau é estável. O que viabilizou os testes de identificação a partir do uso de uma entrada em Degrau.

4.2.2.3 Acompanhamento de Referência

No intuito de avaliar o comportamento dos controladores com um referência com variação fixa da velocidade foram então feitos testes para uma entrada em rampa. Como se trata de um controle feito de maneira discreta, foram passados valores de ângulos em sequência incrementados por um valor contante e separados por um pequeno tempo de espera emulando uma entrada em rampa porém discretizada no tempo.

Para estudo da velocidade de resposta do controlador, o valor incrementado entre um ângulo e outro foi mantido enquanto o intervalo de tempo era ajustado. Neste experimento foi observado que para intervalos muito pequenos de tempo o robô não consegue acompanhar mantendo uma velocidade constante. Em tais casos a velocidade é mantida aproximadamente constante e com o acúmulo do erro ocorre saltos periódicos para compensar o atraso em relação a referência. Este fato decorre da interação entre o controlador de posição das juntas e o controlador de torque. O erro acumulado é corrigido com a passagem de um torque mais alto produzindo um salto na posição, seguido de uma leve oscilação para compensar o efeito da inércia.

4.2.2.4 Resposta ao Degrau

Feito uma avaliação preliminar para o experimento de identificação da planta foi aplicado um degrau para cada uma das juntas individualmente com registro dos dados. Este teste foi definido a partir dos seguintes passos:

1. Começa com a junta na posição 0 graus
2. Envia o comando para ir para posição 45 graus
3. Mantém a referência da posição em 45 graus por 2s
4. Envia o comando para ir para posição 0 graus
5. Mantém a referência da posição em 0 graus por 2s

Para a API em Python estes passos foram executados 3 vezes para cada uma das juntas. Um experimento semelhante foi efetuado a partir de um programa em C++ baseado no código elaborado em trabalhos anteriores. Para ambos casos a saída dos sensores foi registrada a partir do ROS a partir do programa *rosbag* e somente para os testes feitos em C++ foram também registradas as mensagens de controle. Uma vez que os comandos passados via API são enviados diretamente para o serializador de mensagens sem passar pelo ROS e portanto foram utilizados apenas para uma avaliação preliminar do comportamento dos controladores de junta. Mais detalhes do experimento em C++ na seção 4.3.1.

4.2.2.5 Controle de Posição sem Compensação da Gravidade

O controle de posição sem o uso de compensação da gravidade foi avaliado em estudo e não foi possível atingir a posição desejada de 45 graus partindo do zero em nenhuma das juntas. Na maioria dos casos o braço movia apenas 10 graus. O que explicita o fato do impacto de toda a estratégia de compensação da gravidade ser feita por software. A precisão do controle, depende da qualidade do modelo dinâmico e da estratégia utilizada. Esta solução que o sistema de controle possa ser sempre atualizado, uma vez que não está embutido no projeto de hardware, no entanto fica preso a qualidade do modelo.

4.2.2.6 Controle de Velocidade

Na hipótese de utilizar um controle de baixa nível de velocidade ao invés de posição, foi feito apenas um ensaio utilizando o controle de velocidade disponível na API em Python. No entanto ao ser definido velocidade zero a braço robótico ficou completamente rígido e passou a ignorar comandos para outros valores de referência de velocidade. O que demonstrou um esforço grande sendo feito pelo controle de torque e pelos motores. Somente quando foi alterado o modo de controle para posição que o braço voltou a mover normalmente. Para evitar qualquer dano a plataforma, não foram feitos outros testes neste modo de operação. Em [3] este modo de controle

foi classificado como instável, recomendando evitar o uso pois o sinal da velocidade possui muito ruído pois o sinal é estimado no computador através de diferenças finitas a partir da leitura da posição.

4.3 Estudo Controladores Cinemáticos

Em C++ não estão disponíveis todos os modos de operação para o uso a partir do ROS, embora ao final a mensagem seja serializada e passada para os mesmos programas. No nó *shm_hummanoid* está implementado somente o controle por posição com e sem compensação da gravidade. Em decorrência disto, foi analisado somente o controlador de posição com compensação da gravidade por ter sido o utilizado em trabalhos anteriores. Os tempos foram obtidos a partir do *ros :: time* e registrado nas mensagens como *timestamp*. Foi necessário alterar o código do *vrep_meka_interface* para incluir o *timestamp* na mensagem para permitir a análise dos dados registrados via *rosbag*. Em razão disto, alguns dos experimentos anteriores não foram incluídos no estudo do tempo de resposta por conta de inexistir um referencial de tempo comum para as mensagens do tópico */hummanoid_state* e */hummanoid_command*.

4.3.1 Controle de Posição de Juntas

Para avaliar a resposta do controlador em C++ foi aplicado um degrau para cada uma das juntas individualmente com registro dos dados. Este teste foi definido a partir dos seguintes passos:

1. Começa com a junta na posição 0 rad
2. Envia o comando para ir para posição 1 rad
3. Mantém a referência da posição em 1 rad por 4s
4. Envia o comando para ir para posição 0 rad
5. Mantém a referência da posição em 1 rad por 4s

Estes passos foram repetidos, ajustando-se os os parâmetros de rigidez e velocidade máxima do controlador. A velocidade máxima de cada junta pode ser ajustada para um valor entre 0 e 1, de igual forma a rigidez do braço gerada pelo controlador também pode ser ajustada entre 0 e 1. Valores maiores que 1 foram testados porém não houve mudança significativa em relação a comportamento com valor 1. Muito embora em trabalhos anteriores na plataforma fosse usado *rigidez = 5*. Vale reforçar que estes são parâmetros usados pelos controladores no PC (posição e compensação da gravidade) de forma que não é possível obter diretamente os parâmetros de rigidez e velocidade máxima pela resposta do estado atual fornecida por */hummanoid_state* pois a complacência do controlador é ajustada por software.

Já no experimento feito de referência mostrado na 4.4, pode-se notar que existe um atraso de cerca de 20ms entre o comando e o recebimento do sinal de resposta do atuador no tópico.

Também percebe-se um erro grande em regime permanente das juntas do pulso (5 e 6). Percebe-se uma longa rampa indicando aonde o controle da posição passou a atuar como um controle tudo ou nada passando a velocidade máxima possível. Curiosamente, embora sejam motores distintos em cada uma das juntas com diferentes limites de velocidade a resposta pelo gráfico percebe-se uma velocidade próxima de $V = 1.5\text{rad/s}$. O que mostra que os controladores de posição estão ajustados para operarem na faixa linear comum a todos os motores, que é abaixo da velocidade máxima do junta mais lenta $J4$ de $V_4 = 2.0\text{rad/s}$, conforme registrado na tabela 4.1. De forma que um controlador mais sofisticado pode ser desenvolvido de modo a permitir explorar melhor a capacidade das juntas mais rápidas.

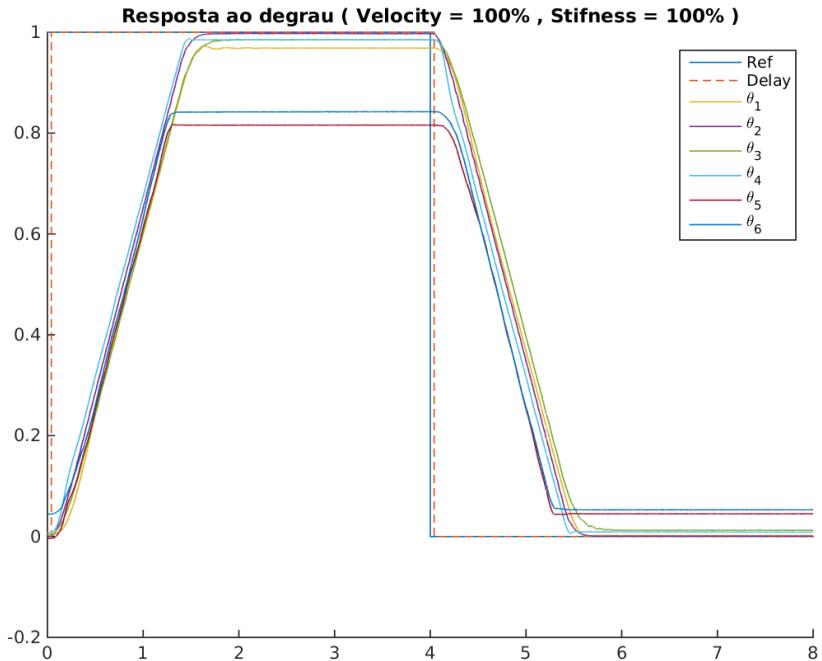


Figura 4.3: Resposta a um degrau de entrada com Velocidade = 1 e Rígidez = 1

Pela figura 4.4 observa-se que a variação da rigidez do controlador implica em um erro maior em regime permanente nas juntas. O controlador converge em um tempo próximo mas notadamente aumenta o erro das juntas do pulso. Este fato também é percebido no código de demonstração no ajuste da rigidez. Tendo sido notado que ao mudar o valor do parâmetro de rigidez na interface com o robô parado sem alterar a posição de referência desejada o pulso desloca significativamente de posição.

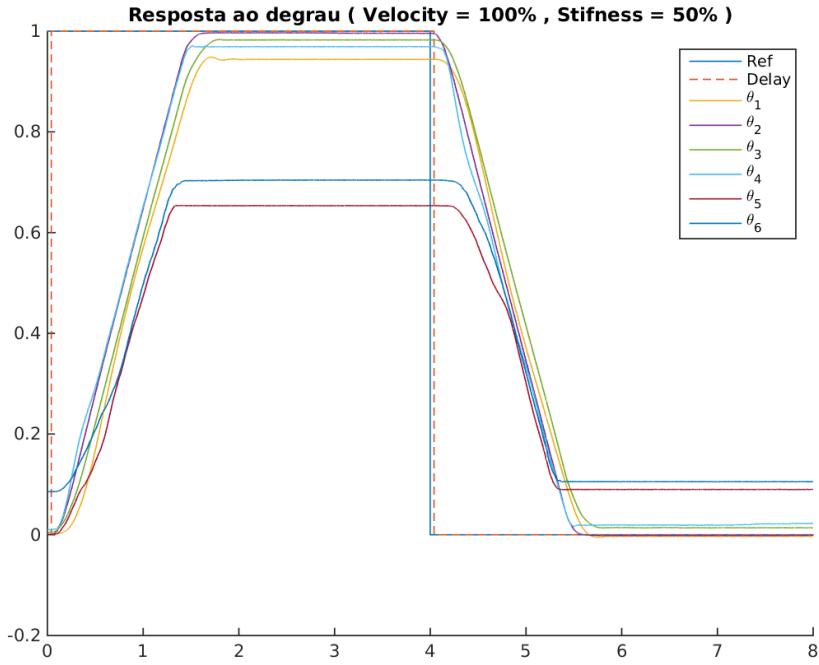


Figura 4.4: Resposta a um degrau de entrada com Velocidade = 1 e Rigidez = 0.5

Como esperado, no terceiro experimento ao se alterar a velocidade para 70% o tempo necessário para cada uma das juntas atingir a posição aumentou, tendo em vista a fase em que o controlador utiliza a velocidade máxima permitida, o erro em regime permanente para cada uma permanece com pouca alteração. No entanto é percebido que a inclinação da curva durante as fase de controle entre o tempo $t = 1s$ e $2s$ com a velocidade máxima muda de forma diferente para cada uma das juntas. Estes dados encontram-se na figura 4.5.

No 3 experimentos o atraso obtido foi próximo de 20ms. Este valor é próximo dos valores de intervalo de tempo obtidos para os melhores resultados dos controladores implementados em [1] e também próximo do valor obtido na seção 4.2.2.1. O que reforça a tese boa parte do atraso ocorre devido a camadas abaixo do nó de ROS. Notadamente, quando o controlador opera com frequências maiores, o impacto do atraso da comunicação passa a ser mais significativo levando a instabilidade caso este não seja considerado no dinâmica implementada como obtido experimentalmente por Marcos. Para uma análise mais completa da resposta das juntas é sugerido o uso de variações menores de ângulo como forma de evitar a não linearidade devido a saturação dos motores.

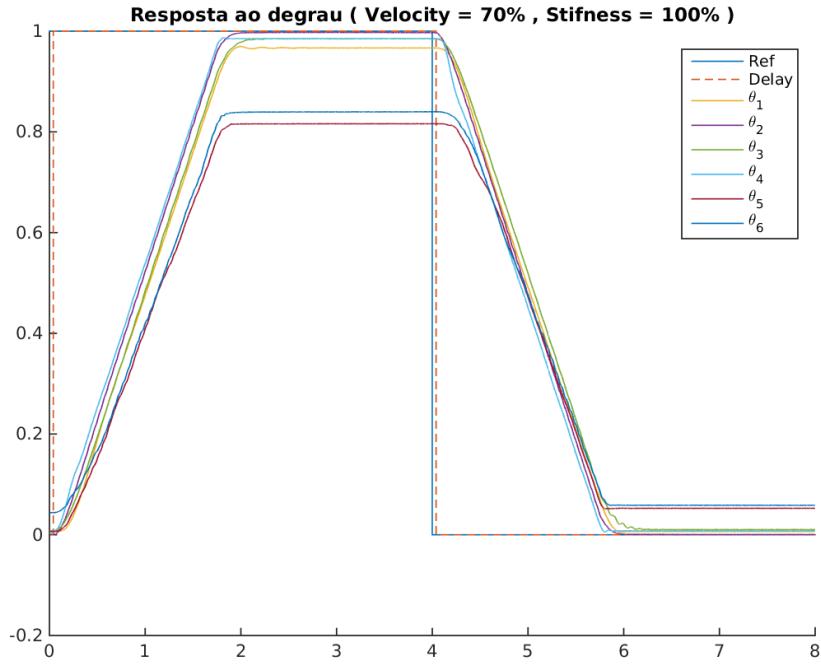


Figura 4.5: Resposta a um degrau de entrada com Velocidade = 0.7 e Rigidez = 1

Tomando-se o diferença entre o valor passado de referência e o valor alcançado tempo foram obtidos os erro em regime permanente, registrados na tabela 4.3.

Tabela 4.3: Error Percentual para diferentes valores de velocidade (V) e rigidez (S)

V	S	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1.00	1.00	-Inf	0.031544	0.0025193	0.014824	0.014865	0.18436	0.15769
1.00	0.50	-Inf	0.056071	0.0036188	0.017413	0.031003	0.34662	0.29566
0.70	1.00	-Inf	0.033305	0.0026167	0.015199	0.014783	0.18398	0.16037
1.00	0.50	-Inf	0.056453	0.0035564	0.017352	0.03028	0.32879	0.28538

Tabela 4.4: Percentual Overshot para diferentes valores de velocidade (V) e rigidez (S)

V	S	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1.00	1.00	0	1.0041	1.0001	1.0002	1.0025	1.0016	0
1.00	0.50	0	1.0049	1.0004	1.0008	1.0019	1.0007	0
0.70	1.00	0	1.003	1.0002	1.0004	1.0016	0	0
1.00	0.50	0	1.005	1.0004	1.0009	1.0026	0	0

Tabela 4.5: Valor em regime permanente para diferentes valores de velocidade (V) e rigidez (S)

V	S	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1.00	1.00	0.001841	0.96846	0.99748	0.98518	0.98513	0.81564	0.84231
1.00	0.50	0.003698	0.94393	0.99638	0.98259	0.969	0.65338	0.70434
0.70	1.00	0.0020654	0.96667	0.99738	0.9848	0.98522	0.81602	0.83963
1.00	0.50	0.0047216	0.94355	0.99644	0.98265	0.96972	0.67121	0.71462

4.3.2 Avaliação na execução de Trajetória em linha reta

Os controladores de posição do efetuador foram avaliado a partir da estratégia de discretização dos pontos e frequência de amostragem. Para tal foi reduzido o número de ponto e trajetória foi simplificada para apenas um deslocamento em linha reta de 10cm na vertical. Para primeira análise foi utilizado os controladores implementados anteriormente e foram feitos os seguintes testes:

1. Trajetória dividida em 100 pontos, intervalo de $8ms$
2. Trajetória dividida em 100 pontos, intervalo de $100ms$

Foi observado que o robô leva um tempo até começar a mexer e um tempo até o controle estabilizar. De modo que está sempre atrasado em relação a referência. Para o período $8ms$ foram necessárias 25 interações até o robô começar a se mover, com o período de $100ms$ o robô começo a se mover já na segunda interação sugerindo um atraso de $200ms$ entre o tempo que o comando é passado via ROS e o tempo que o robô executa o comando.

Assim foram feitos novos experimentos para as seguintes variações:

1. Trajetória dividida em 100 pontos, *velocidade em 100%*, intervalo de $2ms$
2. Trajetória dividida em 100 pontos, *velocidade em 100%*, intervalo de $10ms$

No primeiro experimento, figura 4.6, o robô executou a trajetória sem problemas movendo para cima com velocidade constante.

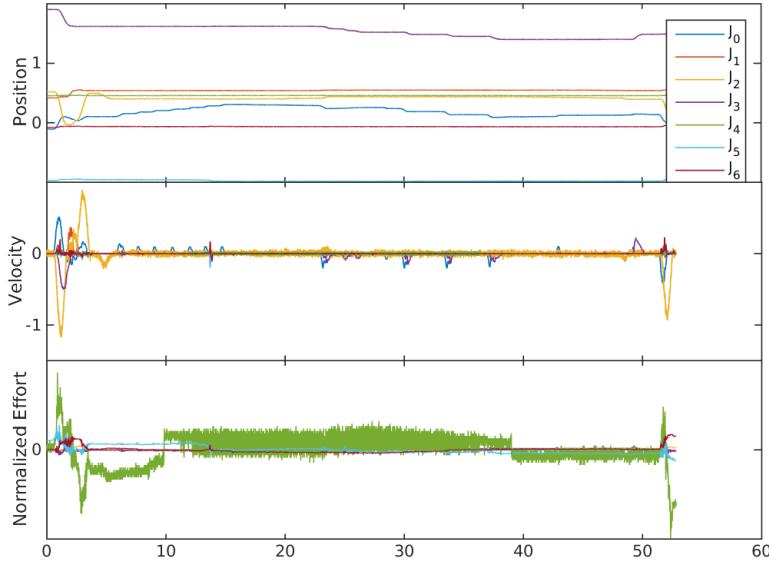


Figura 4.6: Experimento Trajetória na Vertical $dt = 10ms$

Já quando o intervalo de tempo entre os pontos é muito abaixo de $20ms$ é percebido saltos periódicos com intervalo aproximado de $5s$ devido ao acumulo do erro nas juntas. Nota-se que as juntas movem em velocidade constante por um tempo e em seguida rola o salto. Dá para notar que as juntas $J5$ e $J6$ são as que possuem a maior variação no esforço ao longo da trajetória.

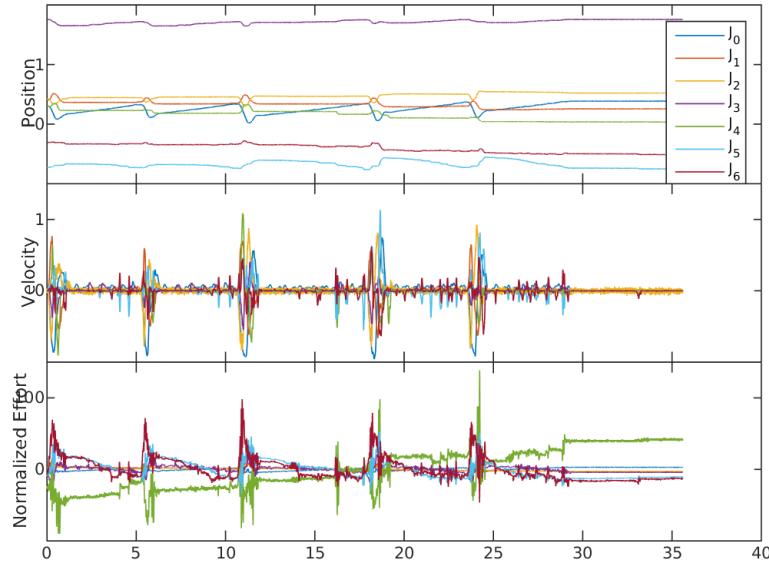


Figura 4.7: Experimento Trajetória na Vertical $dt = 2ms$

4.4 Estudo de saturação da velocidade dos atuadores

Em extensão ao estudo feito na seção 4.3.1, foi analisado a resposta da velocidade fornecida pelo tópico ao longo da trajetória. Pelas figuras 4.3, 4.4 e 4.5 foi notado que a velocidade é

aproximadamente constante ao longo dos momentos de maior deslocamento e com valores próximos de $v_{max} = 0.5\text{rad/s}$ para todas as juntas. Para avaliar o comportamento ao longo do tempo foi comparado a resposta fornecida no tópico do ROS /hummanoid_state para os valores de *position*, *velocity* e *effort*, respectivamente posição, velocidade e torque nas juntas, conforme mostrado nas figuras 4.8 e 4.9.

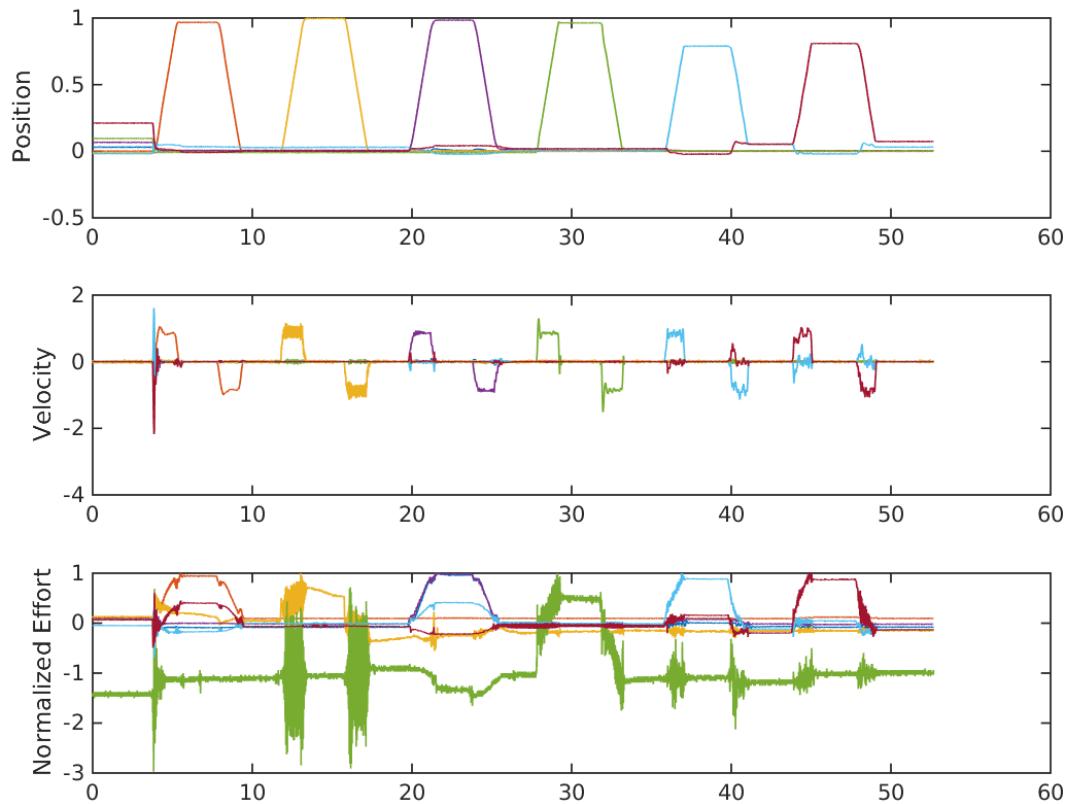


Figura 4.8: Resposta Controladores Juntas - Velocidade Máxima e 90% de rigidez

A medida do torque varia bastante de junta para junta, uma vez que o esforço necessário para mover o braço inteiro é bem menor que para mover cada uma das partes. Em razão disto foi feito um ajuste nos valores de *effort* a dividindo os valores ao longo da trajetória pelo valor máximo observado em cada junta para permitir avaliar o formato da resposta do torque de forma igual para todas as juntas.

Obviamente a velocidade é apenas uma estimativa da velocidade real a partir do valor de posição que chega no computador. Pois, como mostrado na tabela 4.2 não existe um sensor de velocidade embarcado em cada uma das juntas, apenas sensores de posição. Este valor é notadamente ruidoso pois devido ao atraso de comunicação e detecção, a medida dos valores de posição também sofre de não linearidades, apresentando momentos em que esta não

varia.

Ao que é percebido a origem do ruído na medida da velocidade: em momentos que o informação da posição é consultada na memória compartilhada antes da informação ser atualizada gera uma velocidade nula, uma vez que não houve variação do dado. Consultando os códigos foi notado que o valor da medida da estimativa é filtrado por um filtro passa baixa de Butterworth de 3 ordem com frequência de corte em $80Hz$ para cada uma das juntas. O que atenua um pouco do efeito da estimativa da derivada porém ainda assim a confiabilidade da medida é bem menor que a de posição e torque.

Dado que a velocidade máxima da junta mais rápida é de $4.6rad/s$ a variação máxima esperada é de $f_0 = 1/(2\pi/4.6rad/s) = 0.732Hz$, A frequência de corte $f_{filt} = 80Hz$ representa em torno $10f_0$. Como este tipo de filtro não leva em consideração o modelo físico do processo, apenas a resposta em frequência, outros tipos de filtro podem ser implementados, visando melhorar a qualidade da estimativa.

Para avaliar a situação em que a velocidade máxima é abaixo do necessário para permitir o controlador convergir foi refeito o experimento com o mesmo valor de rigidez e 10% da velocidade máxima (Figura 4.9). Nesta situação nota-se uma variação maior para as velocidades apresentadas para as juntas de menor velocidade $J4$, $J5$ e $J6$. A junta do cotovelo $J4$ em particular foi a que apresentou o maior erro. Também não foi possível atingir o ponto de referência no intervalo dado. Em lugar temos uma resposta em formato triangular indicando a saturação da velocidade na subida e na decida. O que também é possível para notar no gráfico da velocidade.

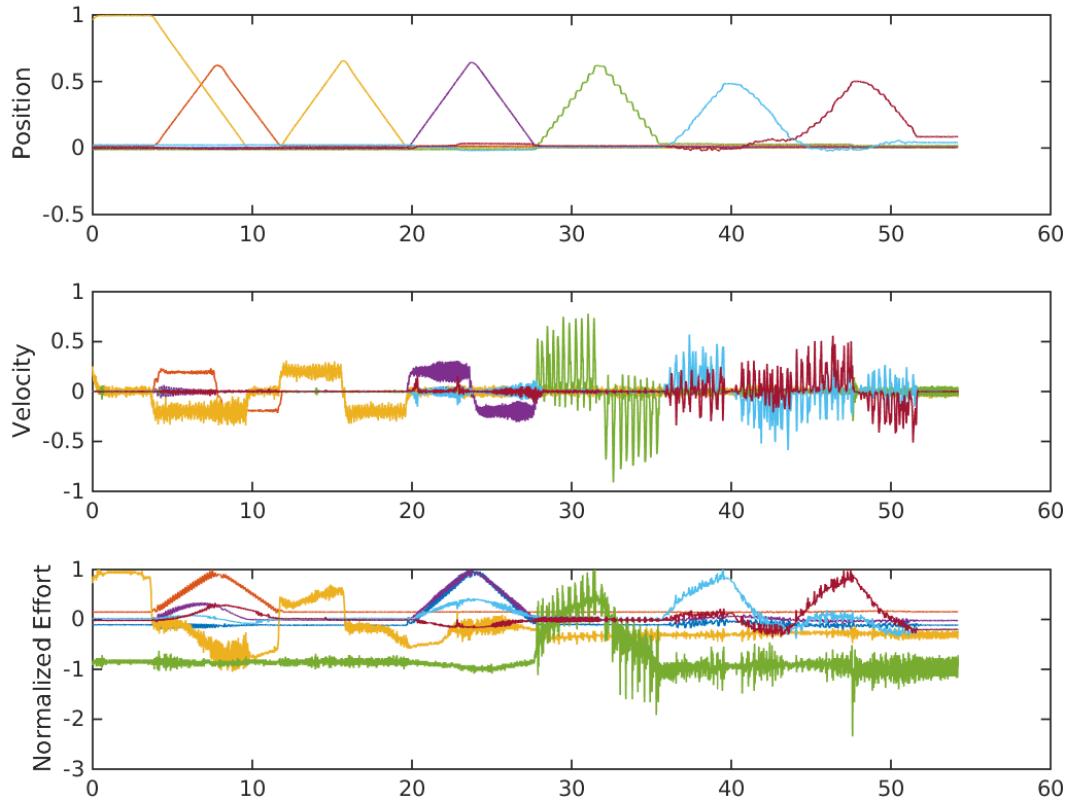


Figura 4.9: Resposta Controladores Juntas - Velocidade em 10%e rigidez em 90%

Uma vez que os modelos usados para o controle são lineares, a existência de comportamentos não lineares como saturação representam perturbações e podem gerar erros ou até instabilidade no controle caso não consideradas. Em razão disto uma atenção especial foi dada a este aspecto visando identificar a contribuição na perca de desempenho. A grande dificuldade é que ocorrem ao longo de todo o processo, principalmente devido a discretização, gerando um acumulo do efeito com o aumento da quantidade de camadas em cascata.

4.4.1 Saturação do Atuador na Trajetória

O efeito de saturação do atuador ocorre quando é solicitado uma ação de torque ou velocidade acima do que o atuador permite. Em diversos sistemas isto é limitado através de algum dispositivo de proteção evitando danificar o atuador. Do lado do controle, este é percebido como uma resposta mais lenta que o esperado para aquele determinado ganho em simulação podendo levar a instabilidade no controle do sistema real devido ao acumulo do erro no integrador.

Como o controle da posição é feito em colaboração pela ação de cada uma das juntas, este problema é mascarado na métrica do erro de posição. A contribuição de cada junta em manipulador não cartesiano muda de acordo com a posição, o atraso devido a saturação pode fazer com

que o robô tenha um erro maior para determinadas posições no decorrer da trajetória que outras. Outro efeito é a saturação de um motor levar a um uso maior dos outros motores na tentativa de compensar o erro acumulado. Como são utilizados diferentes motores para cada uma das juntas, conferindo limitações específicas na atuação, em particular na velocidades máximas que cada junta pode atingir. Por conta disto, é percebido um maior uso das juntas com velocidade máxima maior para compensar o erro acumulado pela juntas com velocidade máxima menor, quando o controle começa solicitar uma velocidade maior que a permitida pelos outros motores. Em particular no Meka, este fenômeno leva a um uso maior das juntas do ombro para corrigir orientação ao acumulado pela saturação das juntas do pulso.

Em um manipulador robótico composto por juntas rotacionais, a tarefa de deslocamento no espaço é distribuída entre todas as juntas ao longo de praticamente toda a região do espaço de trabalho. O deslocamento de cada junta produz um arco que inevitavelmente gera movimentação em pelo menos duas direções no espaço e uma rotação. Desta forma a tarefa de atingir uma posição e orientação no espaço é distribuída por várias juntas, permitindo um esforço menor em um determinado atuador. No entanto, em particular para um braço antropomórfico a orientação é resolvida de forma conjunta pelo pulso e pelo ombro gerando um amplo espaço nulo a permitindo para uma mesma posição do pulso, diversas posições para o cotovelo.

Tais relações geométricas indicam que a tarefa de manter uma orientação possui uma relação com o movimento necessário pelas juntas notadamente diferente da tarefa de manter uma posição. De forma a permitir separar a complacência da posição ou da orientação conforme necessário para a tarefa. Ao se utilizar a representação cinemática por Quaternions Duais tanto a referência da posição como a da orientação estão acopladas. Um bom controlador por outro lado necessita de uma rápida resposta a uma perturbação, gerando uma rigidez maior ao invés de uma complacência. Como para os experimentos em questão a orientação de referência era mantida como a orientação inicial do robô, parte do esforço de controle era direcionado em manter a orientação. O que gerava uma perturbação na posição. Este problema em composição com diferença de velocidade dos atuadores do pulso e da ombro fazia com que uma perturbação na orientação percebida no pulso provoca-se uma movimentação juntas do ombro e por consequência uma movimentação da posição do cotovelo. O que gerava uma aproximação do cotovelo para a base provocando em cascata uma maior chance colisão do braço com o tronco e uma dificuldade em manter a estabilidade as referências do controle através da cinemática diferencial. Pois o robô se aproximava do limite da junta e de regiões com bloqueio de movimento.

As consequências deste fenômeno foram atenuadas pelos primeiros trabalhos com o acréscimo de uma espuma de proteção no tronco do robô, minimizando o risco de dano. Em [1] foi resolvida pelo Marcos Pereira pela escolha de posições iniciais com o cotovelo distante do tronco e operando em uma região limitada por um plano paralelo a frente do robô. Em [11], Rafael Koji, apresenta como solução a separação do acoplamento do controle das 2 juntas do pulso ($j5$ e $j6$) com a redução do modelo cinemático do manipulador para 5 juntas apenas. O que permitindo o controle de posição em malha fechada com 5 graus de liberdade e a orientação em malha aberta pelas juntas do pulso, reduzindo as instabilidades. Em um experimento disponível no YouTube³ Luís

³<https://sites.utexas.edu/hcrl/2010/11/09/experiments-on-prioritized-compliant-control/>

Sentis demonstra a utilização de campos potenciais como forma de manter a postura do robô e evitar a colisão enquanto permite a priorização do controle complacente em uma das 3 direções X , Y e Z a partir da arquitetura implementada em [20] no Meka.

4.5 Estudo Controladores Cinemáticos

4.5.1 Análise individual dos Controladores de Junta

Para avaliar a variação da resposta dos controladores cinemáticos implementado primeiramente foi tomado como referência os resultados de trabalhos passados. No qual foi refeito o experimento da trajetória de um quadrado com todos os controladores ajustados para os melhores parâmetros obtidos em [1], enquanto os parâmetros dos controladores de baixo nível foram mantidos. Neste experimento (Figura 4.10) foi acrescentado a resposta da velocidade e do torque obtidos a partir do tópico */humanoid_state*.

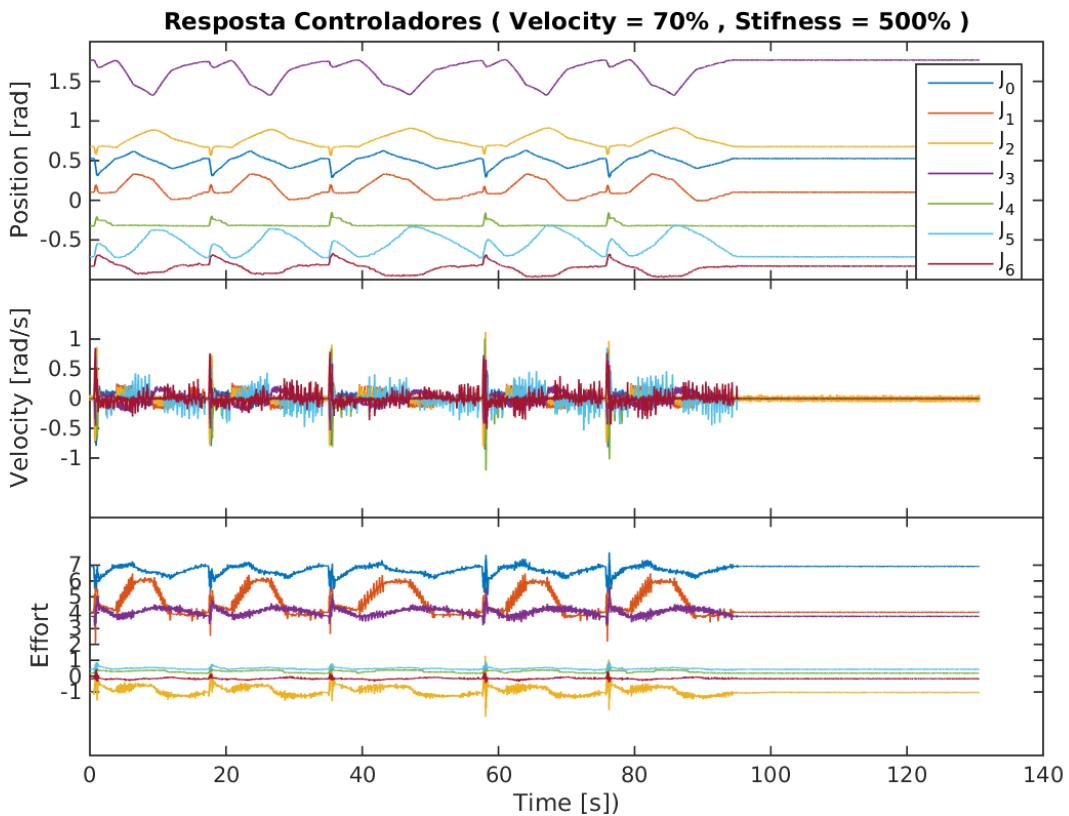


Figura 4.10: Resposta dos Controladores de Juntas para Velocidade em 70% e Stiffness = 5

Apenas analisando a figura 4.10 podemos perceber alguns saltos a cada 18s na posição e na velocidades das juntas. Cada salto denota a transição entre um controlador e outro em decorrência do movimento de retorno a posição inicial. No que pode ser observado que cada controlador leva um tempo diferente para cumprir a trajetória, porém a resposta de todos é bem similar no formato da trajetória. Embora contenha toda a movimentação, este gráfico permite apenas uma visão geral

do comportamento. Assim, a partir destes dados, foram extraídos os dados de cada junta como forma de detalhar a contribuição individual para o erro avaliado e o desenvolvimento da trajetória.

Os gráficos de cada junta são apresentados nas figuras 4.11 a 4.16. Para todas é notado um salto no valor da referência. No entanto, ao observar a resposta de cada junta temos uma atenuação da variação rápida gerada pela transição entre um controlador e outro. Isto se deve a comportamento complacente do atuador série elástico que atua como um filtro passa baixa. A energia introduzida pela variação brusca é absorvida pela elasticidade do atuador e devolvida gradualmente ao sistema como observado mais claramente pelo gráfico de esforço de cada junta em que o salto na referência de posição é acompanhado de imediato por um salto no torque e por uma decida um pouco mais suave na curva da resposta do sensor.

Dentre todas as juntas, a do ombro $J0$ apresentou o menor erro em relação a referência de controle, além de manter um torque aproximadamente constante ao longo de toda trajetória, figura 4.11. Algo esperado por ser a junta mais com o motor mais rápido e a maior rigidez.

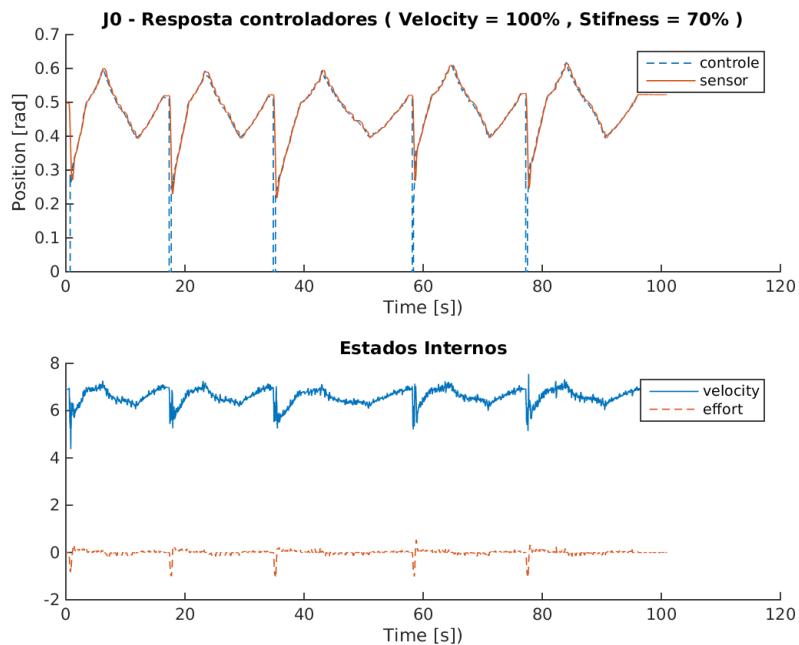


Figura 4.11: Resposta Juntas $J0$ para Velocidade em 70% e Stiffness = 100%

Para as demais juntas do ombro $J1$ e $J2$ o erro também foi pequeno e o torque foi aproximadamente constante. Sofrendo um pouco mais de variação em relação a $J0$ mas ainda dentro de uma margem pequena, como mostrado pelas figura 4.12 e 4.13.

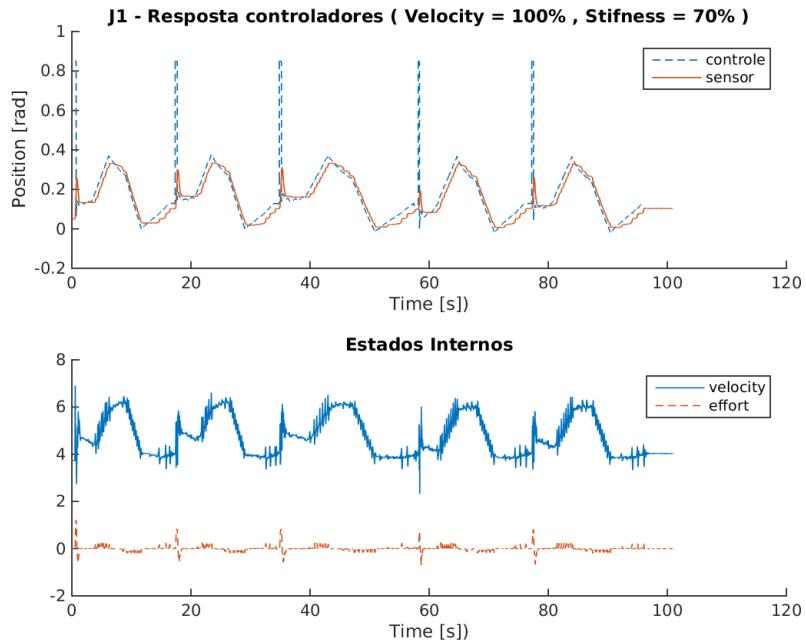


Figura 4.12: Resposta Juntas J_1 para Velocidade em 70% e Stiffness = 100%

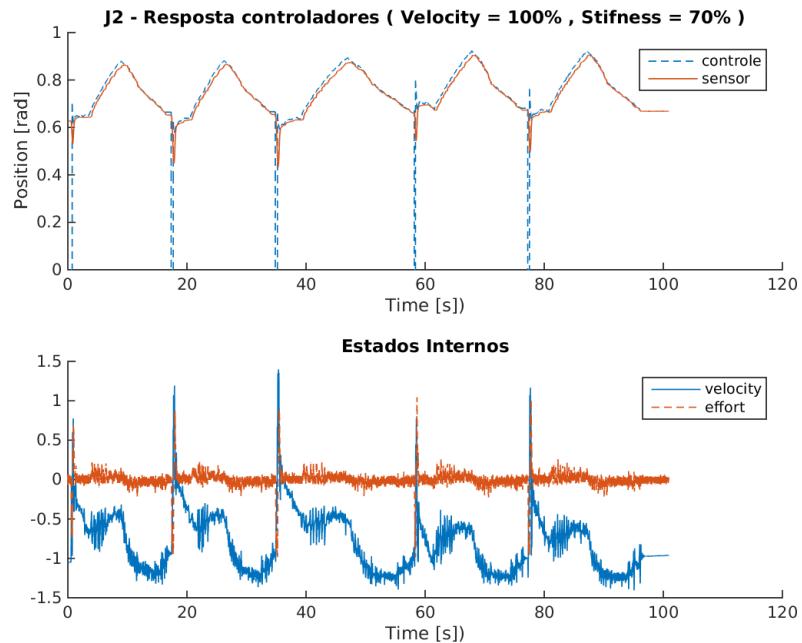


Figura 4.13: Resposta Juntas J_2 para Velocidade em 70% e Stiffness = 100%

A junta J_4 é a mais lenta dentre todas as juntas. A partir da figura 4.14, nota-se uma resposta bem diferente em relação as demais. A junta não consegue acompanhar da mesma forma a referência e leva a uma acumulo de erro. Observa-se que velocidade sofre saltos próximos de 0.2rad/s e o torque oscila bastante em determinadas regiões. Indicando que o controle esta trabalhando em regiões além da região linear de resposta da junta.

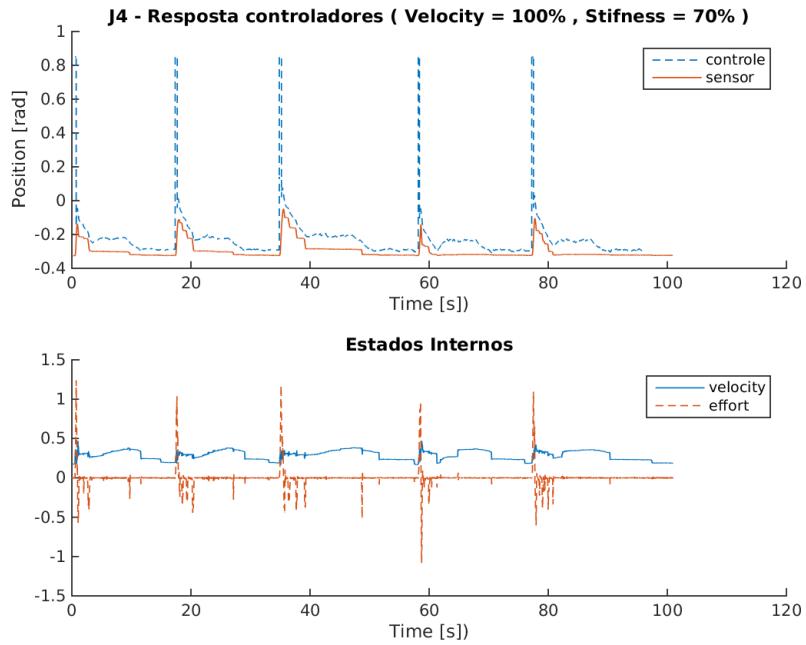


Figura 4.14: Resposta Juntas J_4 para Velocidade em 70% e Stiffness = 100%

As juntas do pulso J_5 e J_6 estão acopladas mecanicamente pelo mecanismo de diferencial e separadas em *Yall* e *Pitch* através de um controlador dentro do PC. Em ambas é percebido o maior erro no acompanhamento da referência, similar ao ocorrido no experimento de resposta a degrau na figura 4.5. Porém é notado uma alta variação no torque da exercido pela junta nas figuras 4.15 e 4.16.

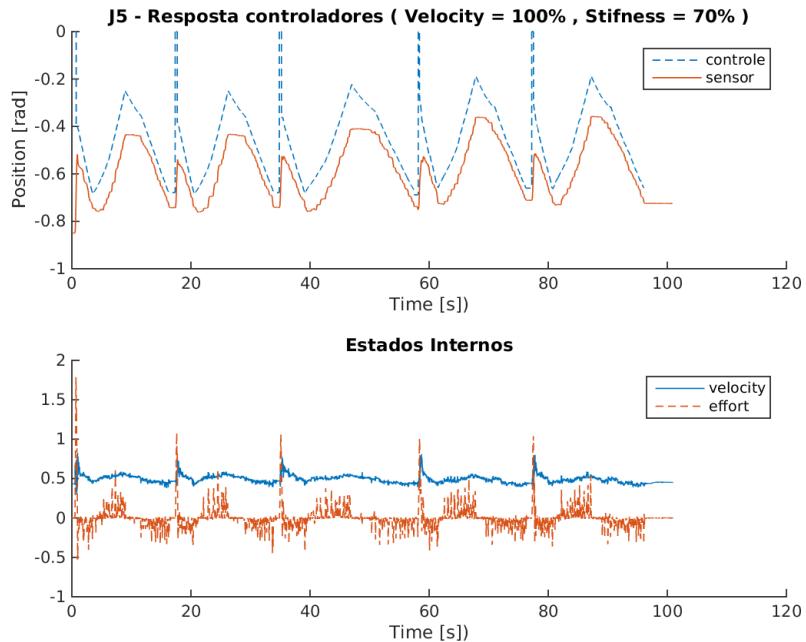


Figura 4.15: Resposta Juntas J_5 para Velocidade em 70% e Stiffness = 100%

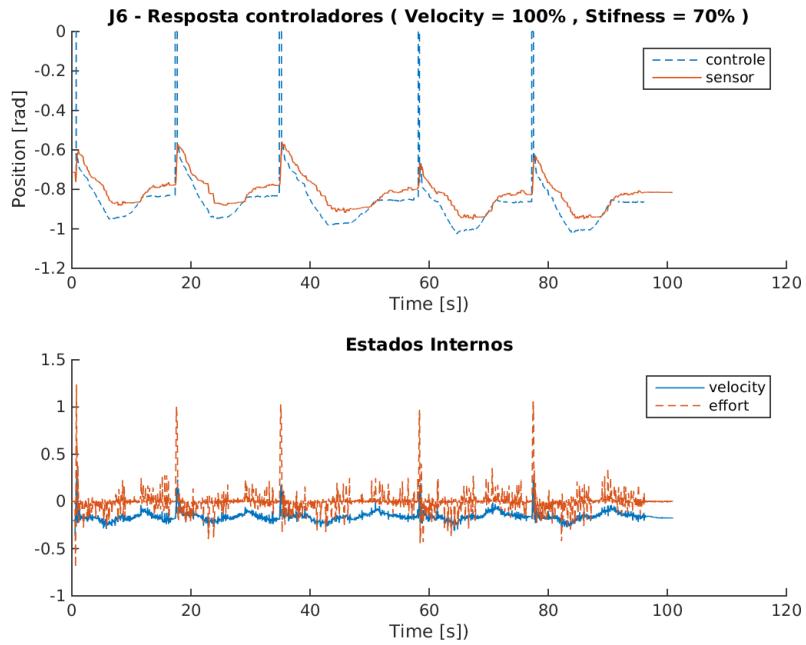


Figura 4.16: Resposta Juntas J_6 para Velocidade em 70% e Stifness = 100%

Naturalmente são as juntas mais próximo do efetuador e portanto toda a oscilação produzida pelas demais juntas é transmitida diretamente ao pulso. No entanto, tendo uma rigidez mais baixa, um erro grande em regime permanente acaba tendo um impacto grande no erro da orientação do efetuador. Já que dado a geometria do robô a orientação da posição de referência do efetuador está ligada diretamente a posição das juntas do pulso. Se considerar ainda que o ponto de referência usado está deslocado em relação ao centro do pulso diferencial, este erro de orientação será ainda refletido também em um erro de posição. Ao que concluímos que a maior contribuição para o erro da posição ocorre nas juntas associadas ao movimento do pulso.

4.5.2 Controle de Rigidez

O mesmo experimento foi feito para vários valores de rigidez. Nas figuras 4.17, 4.18 e 4.19 são apresentados a resposta da junta J_6 do pulso, para rigidez igual a 90%, 50% e 20% respectivamente. Os experimentos foram feitos para todo as juntas no entanto esta foi a que apresentou a maior variação. Onde foi observado que conforme a rigidez é reduzida o erro aumenta em relação ao controle. Com 20% de rigidez o atuador varia pouco a posição e o maior dentre os 3 casos.

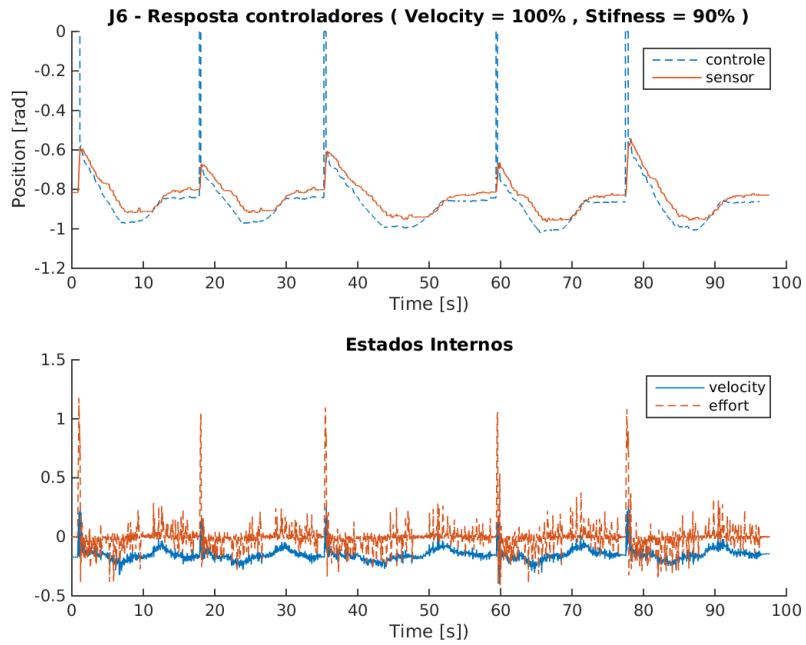


Figura 4.17: Resposta Juntas J_6 para Velocidade em 100% e Stifness = 90%

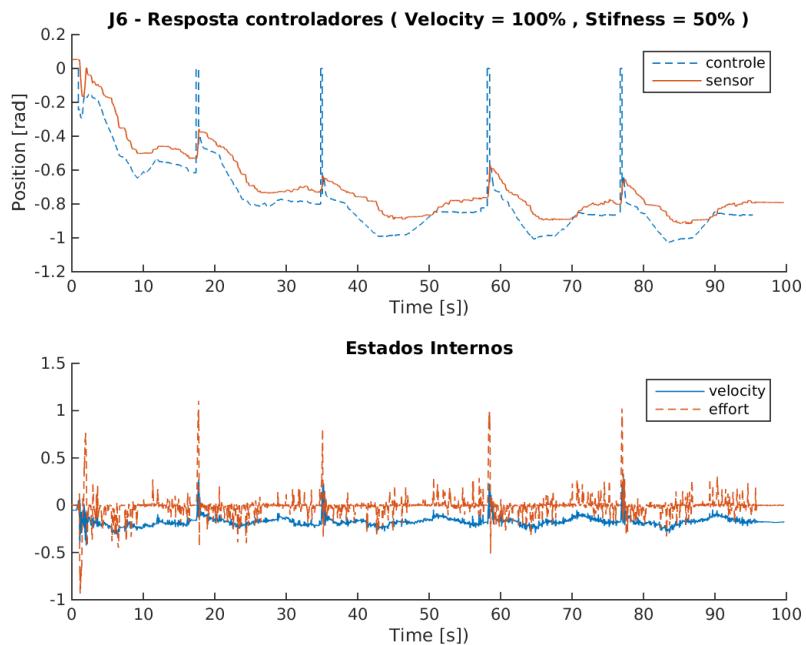


Figura 4.18: Resposta Juntas J_6 para Velocidade em 100% e Stifness = 50%

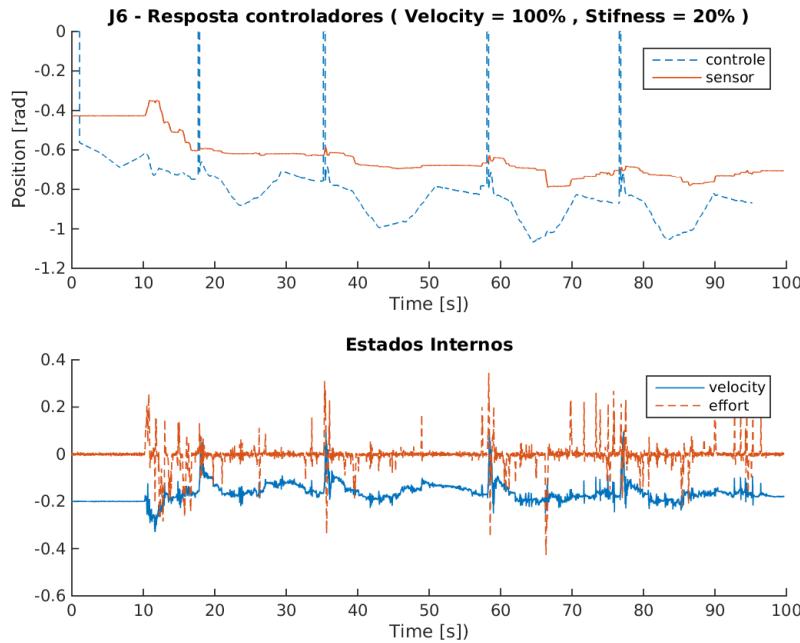


Figura 4.19: Resposta Juntas J6 para Velocidade em 100% e Stiffness = 20%

4.6 Conclusão

Juntando todas as análise foi percebido que o erro do pulso está tendo um peso significativo nos resultados obtidos para o erro. Pois como comentado, devido a geometria do braço, o pulso têm um papel significativo na rotação. E este efeito gera também a translação do ponto de referência do efetuador. O que mostra que a rotação está tendo um impacto significativo no movimento do braço que até então era inexplicado.

Observa-se também uma proporção inadequada entre a unidade de erro de posição e de orientação. Como a referência dos controladores é passada através de Quaternions Duais Unitários, a posição e a orientação estão sempre acompladas. Ainda que durante toda a trajetória a orientação de referência seja mantida, esta não é livre e compete com a tarefa de acompanhar a referência de posição da trajetória.

Como a descrição dos parâmetros do robô têm a distância sendo a avaliada em metros enquanto a orientação é medida sem radianos, esta proporção acaba influênciando na resposta do controlador. Embora seja unidades as unidades padrão, a proporção entre as unidades influência dinâmica do erro. Tal pode ser explicada da seguinte forma: Resgatando da definição de Quaternions Duais Unitários, temos que os primeiros 4 elementos estão relacionados com a orientação e como se trata de um Quaternion Unitário, cada deve variar entre -1 e 1 . De modo que a variação de 2 grau é gera um coeficiente $\text{sen}(\theta_x/2) = \text{sen}(2/2) = \text{sen}(1)$ que é aproximadamente $\pi/180 = 0.017$. Como os parâmetros do robôs são representados em metros, podemos avaliar o coeficiente gerado pelo deslocamento de 2cm com $t_x/2 = 0.02/2 = 0.01$. Dado que o cálculo do erro definido pelos controladores em [1] trata todos os coeficientes do quatérnion dual com o mesmo peso, temos que o desvio de 2graus gera um impacto maior que o desvio de 2cm . O que é um pouco desproporcional se considerarmos que o lado do quadrado efetuado pela trajetória é de 10cm e foco da avaliação era a resposta da translação.

O Meka foi implementado com componentes de alto desempenho seguindo a filosofia do mercado na sua época de lançamento. Embora hoje, outras tecnologias estão sendo incorporadas ao projeto de robôs complacentes como o uso de sistemas freios, esta plataforma ainda representa uma ótima plataforma para o estudo de controladores dinâmicos bem como cada uma dos componentes envolvidos. Como exemplo, o uso de atuadores série elásticos, embora estejam implementados no Meka com motores e sistemas de transmissão de ponta, podem também ser incorporados em outros projetos como solução aos problemas de instabilidade do controle na interação com superfícies rígidas em aplicações para locomoção. Ou ainda em aplicação de forças na área de fisioterapia.

No que tange ao sistema de controle implementado, foi percebido que nos trabalhos anteriores foi alcançado o limite de desempenho dos controladores de baixo nível implementado no sistema M3. No entanto, dado que o sistema de controle é completamente open-source e possui um sistema embarcado capaz de uma desempenho maior ainda existe um enorme potencial que pode ser explorado para uso em trabalhos futuros. Uma vez que as soluções implementadas incorporam técnicas de filtragem e controle dinâmico que podem ser aprimoradas.

Os controladores de posição de juntas foram ajustados para responderem de forma similar para todas as juntas. Embora esta seja uma estratégia que permite que a resposta controle seja similar e facilite a avaliação dos controladores cinemáticos, na prática o comportamento o comportamento de cada junta é bem diferente. Para um ganho de desempenho, pode ser explorado a possibilidade de controladores ajustados para respostas diferentes para cada uma das juntas. Em conjunto com as características particulares da geometria do braço, um desempenho melhor pode ser atingido, por permitir as juntas do ombro operarem com uma resposta mais rápida.

A técnica atualmente implementada para compensação da gravidade usa apenas a posição das juntas para uma gerar um torque em feedforward a partir do modelo cinemático e dinâmico do robô. Outras técnicas podem ser utilizadas para que as velocidade final e o torque possam também ser ajustadas. Também foi notado que o acoplamento entre a posição e a orientação introduzido por quatérnions duais pode ser explorado a partir das características das juntas do robô, uma vez que a composição do ombro e do pulso podem atuar como dois guibas em oposição permitindo um estudo maior da dinâmica da rotação no movimento do efetuador.

Capítulo 5

Conclusão

“What we need is more people specialize in the impossible” (Theodore Roethke)

5.1 Perspectivas Futuras

Neste trabalho foi apresentado um detalhamento maior da arquitetura de controle implementada no Meka permitindo conhecer os fundamentos das tecnologias envolvidas no desenvolvimento de manipuladores robóticos cooperativos bem como vivenciar os principais desafios em manter um manipulador robótico seguro e um desempenho razoável na execução de alguma tarefa. Foi buscado ir a origem inicial de cada problema, tanto do ponto de vista histórico como na análise de cada componente utilizado. Robótica colaborativa ainda representa um tema com bastante espaço de pesquisa e trabalho pois robôs estão cada vez mais acessíveis devido a redução no preço dos computadores, o uso de sistemas série elásticos para controle preciso de torque e bem como a disponibilidade de soluções open-source entre outras tecnologias entre outros fatores. Soluções robóticas sempre estiveram no plano de tecnologias de alto padrão e distantes da realidade da maioria porém têm sido crescente os esforços na difusão.

Apesar de constituir uma plataforma mais acadêmica, o Meka ainda pode ser amplamente utilizado para estudos na área de controle e principalmente para interação com pessoas ou outros robôs. Para trabalhos futuros na plataforma é sugerido a identificação do modelo dos atuadores série elástico disponíveis na plataforma visando a implementação de um controle de juntas mais preciso. No intuito de apresentar a modelagem completa do braço, uma vez que esta não está disponível. Também é apresentado a possibilidade de estudo em controle multivariado a partir de espaço de estados para permitir explorar os efeitos das não linearidade em composição com o modelo cinemático do robô, gerando uma solução que controle ao mesmo a cinemática e dinâmica do movimento. Além de filtros mais sofisticados de forma a permitir o uso da informação do sensor de força em conjunto com os demais sensores. Os conceitos implementados na plataforma podem também compor partes de outros projetos como explorado por outras universidades que passaram a produzir seus próprios atuadores, braços e plataformas robóticas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] PEREIRA, M. S. *Trajectory control of anthropomorphic compliant manipulator with Dual Quaternion Based Kinematic Controllers*. [S.l.: s.n.], 2016.
- [2] PRATT, G. A. et al. Stiffness isn't everything. In: *experimental robotics IV*. [S.l.]: Springer, 1997. p. 253–262.
- [3] WELCOME to meka-docs documentation! - meka-docs 0.1 documentation 2017. 2017. Disponível em: <<https://mekadocs.readthedocs.io/en/latest/>>.
- [4] HOARAU, A. *Mekabot Wiki*. 2015. Disponível em: <<https://github.com/ahoarau/mekabot/wiki>>.
- [5] ROBOTICS, M. *Meka Documentation*. 2013. Disponível em: <mekabot.com/wiki/doku.php>.
- [6] PAINE, N. A. *High-performance series elastic actuation*. Tese (Doutorado), 2014.
- [7] COLEMAN, D. et al. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.
- [8] EDSINGER-GONZALES, A.; WEBER, J. Domo: a force sensing humanoid robot for manipulation research. In: *Humanoids*. [S.l.: s.n.], 2004. v. 1.
- [9] PRATT, G. A.; WILLIAMSON, M. M. Series elastic actuators. In: IEEE. *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*. [S.l.], 1995. v. 1, p. 399–406.
- [10] ROBOTICS, R. *Baxter Documentation*. 2015. Disponível em: <<http://sdk.rethinkrobotics.com/wiki/GravityCompensation>>.
- [11] KOJI, R. *Teleoperação de robô manipulador através de interface haptica com feedback de força*. [S.l.: s.n.], 2017.
- [12] MARINHO, M. M. *Robot-aided Endoscope Control under Laparoscopic Surgery Constraints using LAPAROSCOPIC SURGERY CONSTRAINTS USING DUAL QUATERNIONS*. [S.l.: s.n.], 2014.
- [13] HAMILTON, W. R. Xi. on quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 33, n. 219, p. 58–60, 1848.

- [14] CLIFFORD. Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, s1-4, n. 1, p. 381–395, 1871.
- [15] ADORNO, B. V. *Two-arm manipulation: from manipulators to enhanced human-robot collaboration*. Tese (Doutorado) — Université Montpellier 2, 2011.
- [16] KAVAN, L. et al. Dual quaternions for rigid transformation blending. *Trinity College Dublin, Tech. Rep. TCD-CS-2006-46*, 2006.
- [17] ABEYRATHNA, P. et al. Simulation on active vibration suppression using virtual spring-damper combination. In: IEEE. *Circuits, Controls and Communications (CCUBE), 2013 International conference on*. [S.l.], 2013. p. 1–6.
- [18] QUIGLEY, M. et al. Ros: an open-source robot operating system. In: KOBE, JAPAN. *ICRA workshop on open source software*. [S.l.], 2009. v. 3, n. 3.2, p. 5.
- [19] SUCAN, I. A.; MOLL, M.; KAVRAKI, L. E. The open motion planning library. *IEEE Robotics Automation Magazine*, v. 19, n. 4, p. 72–82, Dec 2012. ISSN 1070-9932.
- [20] SENTIS, L. *Synthesis and control of whole-body behaviors in humanoid systems*. [S.l.: s.n.], 2007.
- [21] ADORNO, B. V. Robot kinematic modeling and control based on dual quaternion algebra—part i: Fundamentals. 2017.
- [22] MARINHO, M. M.; BERNARDES, M. C.; BO, A. P. A programmable remote center-of-motion controller for minimally invasive surgery using the dual quaternion framework. In: IEEE. *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS EMBS International Conference on)*. Sao Paulo, Brazil, 2014. v. 1, p. 339–344. ISSN 2155-1774.
- [23] FIGUEREDO, L. F. C. et al. Robust kinematic control of manipulator robots using dual quaternion representation. In: *2013 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2013. p. 1949–1955. ISSN 1050-4729.
- [24] IYER, S. S. *Modeling and testing of a series elastic actuator with controllable damping*. Tese (Doutorado) — Worcester Polytechnic Institute, 2012.
- [25] SICILIANO, B. et al. *Robotics - Modelling, Planning and Control*. [S.l.]: Springer, 2009.
- [26] SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot Modeling And Control*. [S.l.]: John Wiley & Sons, INC., 2005.
- [27] LEE, C. et al. Generalization of series elastic actuator configurations and dynamic behavior comparison. In: MULTIDISCIPLINARY DIGITAL PUBLISHING INSTITUTE. *Actuators*. [S.l.], 2017. v. 6, n. 3, p. 26.

ANEXOS

I. DESCRIÇÃO DO CONTEÚDO DO CD

Este trabalho é open source e está disponível no Github através do link <https://github.com/akafael/undergrad-thesis>. Dentro do CD e do repositório estão as seguintes pastas:

- **doc**: Arquivo PDF do Relatório e Resumo (*Somente no CD*)
- **tex**: Código fonte do relatório
 - **figs**: Figuras Utilizadas
 - **tables**: Tabelas em formato *.tex*
- **src**: Código utilizado para gerar as figuras
 - **dots**: códigos em DOT para esquemas
 - **matlab**: códigos em Matlab para os gráficos
 - **logs**: dados dos experimentos feitos
 - **org**: registro das atividades desenvolvidas em formato org-mode

Em particular no repositório do Github, o arquivo em PDF do relatório está disponível no link <https://github.com/akafael/undergrad-thesis/releases>.