

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EDUARDO HENRIQUE MACIEL

**DESENVOLVIMENTO DE UM MODELO
SIMPLIFICADO DOS MEMBROS
INFERIORES DE UM ROBÔ BÍPEDE
UTILIZANDO ROS**

Porto Alegre
2014

EDUARDO HENRIQUE MACIEL

**DESENVOLVIMENTO DE UM MODELO
SIMPLIFICADO DOS MEMBROS
INFERIORES DE UM ROBÔ BÍPEDE
UTILIZANDO ROS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Renato Ventura Bayan Henriques

CO-ORIENTADOR: Prof. Dr. Walter Fetter Lages

Porto Alegre
2014

EDUARDO HENRIQUE MACIEL

**DESENVOLVIMENTO DE UM MODELO
SIMPLIFICADO DOS MEMBROS
INFERIORES DE UM ROBÔ BÍPEDE
UTILIZANDO ROS**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade Federal de Minas Gerais – UFMG,
Brasil

Banca Examinadora:

Prof. Dr. Alexandre Balbinot, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Eduardo André Perondi, UFSC
Doutor pela Universidade Federal de Santa Catarina, UFSC, Brasil.

Prof. Dr. Vinícius Menezes de Oliveira, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE: _____
Prof. Dr. Alexandre Sanfelice Bazanella

Porto Alegre, Maio de 2014.

DEDICATÓRIA

Agradeço a minha família em especial a minha esposa Juliana Louise Knebel, pelo suporte ao longo desta jornada. Aos meus orientadores, Prof. Dr. Renato Ventura Bayan Henriques e o Prof. Dr. Walter Fetter Lages, pela atenção e ensino durante a realização deste trabalho. Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade. À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Fundação de Apoio à Pesquisa do Estado do Rio Grande do Sul (FAPERGS). Aos amigos, Anderson Antônio Giacomolli, Carlos Solon Soares Guimarães Junior e Thiago José Michelin.

RESUMO

Este trabalho apresenta o desenvolvimento de um modelo simplificado dos membros inferiores de um robô bípede, composto basicamente por uma cintura, dois fêmures, duas tíbias e dois pés. A estrutura mecânica do modelo em questão, possui seis graus de liberdade e tem as dimensões aproximadas de um ser humano de estrutura mediana. Seu sistema de controle e de geração de trajetórias é desenvolvido utilizando funcionalidades disponíveis no *Robot Operating System* (ROS), porém ao contrário da maioria dos controladores existentes no ROS, este projeto propõe implementar um pacote contendo um controlador multivariável (*multi-input, multi-output* (MIMO)), utilizando a técnica de controle por torque calculado. Para a geração de trajetórias das pernas do robô, implementam-se três tipos diferentes de geração, a interpolação linear, cubica e de quinto grau. Para os testes de validação do sistema de controle e de geração de trajetórias utiliza-se o simulador Gazebo.

Palavras-chave: Robô Bípede, Robot Operating System (ROS), Controlador Multivariável, Torque Calculado, Geração de Trajetória.

ABSTRACT

This work presents the development of a simplified model of a biped robot's lower limbs, composed basically by the waist, two femurs, two tibia and two feet. The model's mechanical structure has six degree of freedom and its dimensions are comparable to a human being's body. Its control and trajectory generation systems are developed making use of some features available in the Robot Operation System (ROS) tool. However, contrary to most of the controllers offered by ROS, this project suggests the implementation of a new package, including a MIMO (multi-input multi-output) controller, making use of the calculated torque technique. As for the trajectory generation system, three different methodologies are applied of the interpolation: linear, cubic and polynomial quintic. To validate both control and trajectory generation systems, the Gazebo simulator is used.

Keywords: Biped Robot, Robot Operating System, Computed Torque.

LISTA DE ILUSTRAÇÕES

Figura 1:	Tipos de cadeias cinemáticas.	26
Figura 2:	Tipos tradicionais de juntas.	26
Figura 3:	Robô WABIAN-2 possui 41 GDLs (OGURA et al., 2006).	27
Figura 4:	Plano sagital ou lateral, plano frontal e no plano transversal de um ser humano.	27
Figura 5:	Graus de liberdade de um robô bípede no plano sagital (CHEVAL- LIEREAU et al., 2009).	28
Figura 6:	Graus de liberdade de um robô bípede no plano frontal (CHEVAL- LIEREAU et al., 2009).	28
Figura 7:	Graus de liberdade de um robô bípede no plano transversal (CHE- VALLEREAU et al., 2009).	29
Figura 8:	Diagrama de blocos do controle por torque calculado (Modificada de (FU; GONZALEZ; LEE, 1987)).	34
Figura 9:	Robôs Bípedes desenvolvidos no Brasil.	36
Figura 10:	Robôs desenvolvidos pela Universidade Waseda.	37
Figura 11:	Robôs desenvolvidos pela Honda Motor Co., Ltd.	38
Figura 12:	Robôs desenvolvidos pela AIST e Kawada Industries Inc.	39
Figura 13:	Robôs desenvolvidos pela Universidade Técnica de Munique.	40
Figura 14:	Robôs desenvolvidos pela Boston Dynamics.	41
Figura 15:	Robôs desenvolvidos pela NASA Johnson Space Center.	41
Figura 16:	Robôs bípedes desenvolvidos utilizando o ROS.	42
Figura 17:	Exemplo de um nodo no ROS.	45
Figura 18:	Exemplo de um tópico no ROS.	45
Figura 19:	Exemplo de um serviço no ROS.	46
Figura 20:	Cadeia cinemática com topologia do tipo árvore (IOAN, 2013a).	46
Figura 21:	Estrutura de um elo no URDF (IOAN, 2013b).	47
Figura 22:	Estrutura de um junta no URDF (IOAN, 2013c).	48
Figura 23:	Robô Bípede Orbit.	49
Figura 24:	Parser de URDF para KDL.	50

Figura 25:	Diagrama de fluxo de dados de um controlador no ROS (DAVE, 2013a).	51
Figura 26:	Interface de ação entre o cliente e o servidor.	54
Figura 27:	Metodologia de desenvolvimento do robô bípede Orbit.	55
Figura 28:	Evolução do robô Orbit (MACIEL, 2011).	56
Figura 29:	Estruturas simplificadas dos membros inferiores.	57
Figura 30:	Ponto de origem da junta do robô Orbit.	57
Figura 31:	Esquema do Modelo cinemático do robô bípede Orbit.	58
Figura 32:	Diagrama contendo os nodos do projeto do robô bípede Orbit no ROS.	59
Figura 33:	Diagrama de fluxo de controle no ROS.	60
Figura 34:	Diagrama de fluxo de mensagens do gerador de trajetória do robô bípede Orbit no ROS.	61
Figura 35:	Diagrama de nodos do gerador de trajetória do robô bípede Orbit no ROS, utilizando interpolação linear.	63
Figura 36:	Sinal de referência e de posição da junta 1,2 e 3.	66
Figura 37:	Torque na junta 1.	66
Figura 38:	Torques nas juntas 2 e 3.	67
Figura 39:	Sinal de referência e de posição da junta 1.	68
Figura 40:	Torque na junta 1.	68
Figura 41:	Torques nas juntas 2 e 3.	69
Figura 42:	Sinal de referência e de posição da junta 1.	70
Figura 43:	Torque na junta 1.	70
Figura 44:	Torques nas juntas 2 e 3.	71
Figura 45:	Sinal de referência e de posição da junta 1.	72
Figura 46:	Torque na junta 1.	72
Figura 47:	Torques nas juntas 2 e 3.	73
Figura 48:	Sinal de referência e de posição da junta 1,2 e 3.	74
Figura 49:	Torque na junta 1.	75
Figura 50:	Torques nas juntas 2 e 3.	75
Figura 51:	Sinal de referência e de posição da junta 1.	76
Figura 52:	Torque na junta 1.	77
Figura 53:	Torques nas juntas 2 e 3.	77
Figura 54:	Sinal de referência e de posição da junta 1.	78
Figura 55:	Torque na junta 1.	79
Figura 56:	Torques nas juntas 2 e 3.	79
Figura 57:	Sinal de referência e de posição da junta 1.	80
Figura 58:	Torque na junta 1.	81
Figura 59:	Torques nas juntas 2 e 3.	81

LISTA DE TABELAS

Tabela 1:	Tabela de valores de sinais sem saturação para a trajetória ao degrau.	73
Tabela 2:	Tabela de valores de sinais sem saturação para a interpolação linear, cúbica e quártupla.	73
Tabela 3:	Tabela de valores de sinais sem saturação para a trajetória ao degrau.	82
Tabela 4:	Tabela de valores de sinais sem saturação para a interpolação linear, cúbica e quártupla.	82

LISTA DE ABREVIATURAS

AIST	Instituto Nacional Japonês de Ciência Industrial Avançada e Tecnologia
ASIMO	Advanced Step in Innovative Mobility
DAE	Digital Asset Exchange
DARPA	Defense Advanced Research Projects Agency
GDL	Graus de Liberdade
KDL	Kinematics and Dynamics Library
MIMO	Multi Input - Multi Output
NASA	National Aeronautics and Space Administration
OSRF	Open Source Robot Foundation
PD	Proporcional-Derivativo
PID	Proporcional-Integral-Derivativo
ROS	Robot Operating System
SISO	Single Input - Single Output
STL	STereoLithography
USP	Universidade de São Paulo
UFRJ	Universidade Federal do Rio de Janeiro
UNICAMP	Universidade Estadual de Campinas
URDF	Unified Robot Description Format
YAML	Yet Another Markup Language

LISTA DE SÍMBOLOS

q	Vetor de posição das juntas
qr	Vetor de referência de Posição das juntas
K_p	Matriz de ganho proporcional
K_d	Matriz de ganho derivativo
$D(q)$	Matriz de inércia
$H(q, \dot{q})$	Vetor de forças centrífugas e forças de Coriolis
$G(q)$	Vetor de forças gravitacionais
$D_n(q)$	Matriz de inércia nominal
$H_n(q, \dot{q})$	Vetor de forças centrífugas e forças de Coriolis nominal
$G_n(q)$	Vetor de forças gravitacionais nominal
τ	Vetor de torque aplicado nas juntas
L	Lagrangiano
K	Energia Cinemática
P	Energia Potencial
I	Momento de inércia
m_i	Massa do elo i
τ_i	Torque na junta i
z_0	$[0 \ 0 \ 1]^T$
${}^iR_0\omega_i$	Velocidade angular no elo i
${}^iR_0\dot{\omega}_i$	Aceleração angular no elo i
${}^iR_0\dot{v}_i$	Velocidade linear no elo i

- ${}^iR_0\bar{a}_i$ Aceleração linear no centro de gravidade do elo i
- ${}^iR_0p_i^*$ Posição da origem do sistema de coordenadas i em relação ao sistema de coordenadas $i-1$
- ${}^iR_0\bar{s}_i$ Posição do centro de gravidade do elo i em relação ao sistema de coordenadas i
- iR_0f_i Força atuante no elo i
- iR_0n_i Momento atuante no elo i
- ${}^iR_0I_i^0 R_i$ Tensor de inércia do elo i

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Objetivo	22
1.2	Organização do Trabalho	23
2	FUNDAMENTOS TEÓRICOS	25
2.1	Cinemática	25
2.2	Dinâmica	29
2.2.1	Formulação de Lagrange-Euler	30
2.2.2	Formulação de Newton-Euler	31
2.3	Sistema de Controle	34
2.4	Robôs Bípedes	35
2.5	Robôs Bípedes desenvolvidos utilizando ROS	42
3	MODELAGEM E SIMULAÇÃO	43
3.1	Robot Operating System (ROS)	44
3.1.1	Unified Robot Description Format (URDF)	46
3.1.2	Gazebo	49
3.1.3	Kinematics and Dynamics Library (KDL)	49
3.1.4	Sistema de Controle	50
3.1.5	Gerador de Trajetórias	53
4	METODOLOGIA	55
4.1	Desenvolvimento do Robô bípede Orbit	55
4.1.1	Estrutura Mecânica do Robô Bípede Orbit	56
4.1.2	Modelo Cinemático do Robô Bípede Orbit	57
4.1.3	Sistema de Controle do Robô Bípede Orbit	58
4.1.4	Gerador de Trajetórias	61

5	SIMULAÇÕES E RESULTADOS	65
5.1	Simulação sem Saturação no Sinal de Controle	65
5.1.1	Trajectoria ao Degrau	65
5.1.2	Interpolação Linear	67
5.1.3	Interpolação Cúbica	69
5.1.4	Interpolação Quíntupla	71
5.2	Simulação com Saturação no Sinal de Controle	74
5.2.1	Trajectoria ao Degrau	74
5.2.2	Interpolação Linear	76
5.2.3	Interpolação Cúbica	78
5.2.4	Interpolação Quíntupla	80
6	CONCLUSÕES E TRABALHOS FUTUROS	83
	REFERÊNCIAS	87
	APÊNDICE A ESTRUTURA MECÂNICA DO ROBÔ BÍPEDE ORBIT	93
A.1	Dimensões	93
A.2	Parâmetros dos Corpos Rígidos	94
	APÊNDICE B ATUADOR SCHUNK	99
B.1	Modelo PRL 80	99
B.2	Dimensões	100
B.3	Dados Técnicos	100
	APÊNDICE C URDF EXEMPLO	101
C.1	Cadeia cinemática de um robô genérico do tipo árvore	101
	APÊNDICE D ÁRVORE DE DIRETÓRIOS	105

1 INTRODUÇÃO

Os métodos de locomoção sobre o solo mais utilizados na robótica são por meio de rodas, trilhos ou pernas (LUM; ZRIBI; SOH, 1999). A locomoção que utiliza pernas, apresenta a característica de deslocamento descontínuo sobre o solo, permitindo o desvio de obstáculos, deslocar-se sobre superfícies inclinadas ou subir e descer escadas. Existem diversos grupos de pesquisa ao redor do mundo focados no desenvolvimento de robôs com pernas, em especial robôs dotados de duas pernas, denominados de robôs bípedes (HASEHAWA; ARAKAWA; FUKUDA, 2000). Atualmente estes robôs são capazes de imitar os movimentos humanos, devido as suas características antropomórficas.

Os robôs bípedes podem ser classificados em três categorias (WAARD; INJA; VISSER, 2013):

- Estáticos: seus movimentos são lentos e a estabilidade do seu sistema é descrita pela projeção do centro de gravidade, o que depende apenas das posições de cada junta. Estes robôs requerem atuadores com elevada potência para realização de movimentos;
- Passivos: este sistema não possui acionamento e não necessita de controle de torque ou de posição. O movimento de caminhada desse robôs é inicializado normalmente em posições e velocidades, tais que o fim do movimento é praticamente idêntico às condições iniciais;
- Dinâmicos: possuem articulações, pés e atuadores, dispostos em sua estrutura mecânica. A locomoção é realizada aproveitando as características naturais do caminhar e minimizando o consumo de energia.

No início dos estudos na área dos robôs bípedes a ênfase era apenas na locomoção e não na aplicação destes robôs em ambientes reais (RIEZENMAN, 2002), porém atualmente esses estudos estão tomando outros rumos, devido a semelhança estrutural entre um robô bípede e um ser humano, começou-se a pensar em tornar este robôs em

trabalhadores operacionais, substituindo os seres humanos em atividades que envolvam algum tipo de risco à saúde. Em teoria, os robôs bípedes podem realizar diversas atividades, como supervisão, manutenção e assistência em ambientes que hoje são ocupados apenas pelos seres humanos (SARDAIN; ROSTAMI; BESSONNET, 1998).

Porém ao comparar as habilidades de um ser humano as de um robô bípede, existem algumas diferenças, como por exemplo, a velocidade máxima de uma marcha atlética de um ser humano que pode variar de 11,5 a 14,8 km/h (WILLIAM, 2011). Em termos de velocidade, atualmente o Robô ASIMO destaca-se como sendo um dos bípedes mais rápidos, alcançando uma velocidade máxima que pode chegar a 9 km/h (COMPANY, 2011). Existem também fatores mecânicos e de controle que justificam essas diferenças. Uma diferença existente associada aos fatores mecânicos, são as configurações aplicadas aos N graus de liberdade (GDL) necessários para desenvolver um robô bípede similar a estrutura de um ser humano. Outro fator associado ao controle, está relacionado ao desenvolvimento de um algoritmo que evite as singularidades durante a locomoção (KURAZUME et al., 2005).

Na grande parte das técnicas de controle utilizadas em robôs bípedes leva-se em consideração o modelo parcial ou completo do robô, buscando um melhor desempenho para o seu sistema. Um dos sistemas de controle mais comuns encontrados no desenvolvimento de robôs bípedes, consiste em que cada junta seja controlada independente, ou seja, um conjunto de sistemas SISO (*Single input - Single output*) operando em paralelo. Desta forma projeta-se um controlador para cada junta, ignorando os efeitos de acoplamento entre as juntas. Frequentemente esses controladores são desenvolvidos utilizando uma estrutura de controle Proporcional-Derivativo (PD) ou uma estrutura de controle Proporcional-Derivativo-Integral (PID). Uma forma mais adequada de lidar com o problema do acoplamento entre juntas e as não linearidades em sistemas complexos como o de um robô bípede, é utilizar um sistema de controle MIMO (*Multi input, Multi output*). Com o uso desta técnica obtém-se o acréscimo de elementos dinâmicos na lei de controle, de modo a reduzir os efeitos de acoplamentos entre as juntas. Uma das técnicas mais usuais é a implementação de um sistema de controle por torque calculado (FU; GONZALEZ; LEE, 1987).

1.1 Objetivo

Este trabalho propõe-se em desenvolver um pacote para o *Robot Operating System* (ROS) (QUIGLEY et al., 2009), voltado a área de robôs bípedes, contendo:

- uma descrição em *Unified Robot Description Format* (URDF) do modelo simplificado dos membros inferiores de um robô bípede, compostos basicamente por uma cintura, dois fêmures, duas tíbias e dois pés. Essa estrutura mecânica possui seis graus de liberdade no plano sagital;

- um controlador multivariável (MIMO), que utiliza a técnica de controle por torque calculado.

1.2 Organização do Trabalho

No Capítulo 2 são apresentados os fundamentos teóricos, apresenta resumidamente alguns conceitos sobre cinemática e dinâmica na área dos robôs, bem como história e conceitos sobre os robôs bípedes. O Capítulo 3 descreve os conceitos sobre a modelagem e a simulação utilizando o ROS. No Capítulo 4 é apresentado a metodologia de desenvolvimento para o robô bípede denominado de Orbit. A apresentação das simulações e resultados obtido a partir da simulação e testes do sistema de controle são vista no Capítulo 5. No Capítulo 6 são feitas as considerações finais sobre o trabalho desenvolvido e sugestões para trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Os robôs bípedes possuem características antropomórficas que lhes permitem realizar um deslocamento de um ponto ao outro. Esse deslocamento é realizado por meio do uso de pernas, apresentando uma característica de movimento descontínuo sobre o solo.

Este capítulo apresenta resumidamente alguns conceitos sobre cinemática e dinâmica de robôs, bem como alguns conceitos sobre robôs bípedes.

2.1 Cinemática

A cinemática é parte da mecânica que estuda os movimentos sem considerar as forças que são responsáveis por estes movimentos. Os movimentos, em geral, são descritos por posições, velocidades e acelerações (MISTRY, 2008). Um dos pontos importantes na descrição da cinemática de um robô é a escolha do tipo de cadeia cinemática.

Por definição, uma cadeia cinemática é descrita com sendo um conjunto de elos (corpos rígidos) conectados por juntas (pares cinemáticos), podendo ser divididas em três tipos: cadeia aberta, cadeia fechada ou cadeia híbrida (IONESCU, 2003). Essa classificação é aplicada de acordo com a disposição dos elos que compõem a estrutura mecânica do robô. Em uma cadeia aberta, as ligações dos elos acontecem de forma sequencial e seguem apenas um caminho que liga um elo a outro. Já, na cadeia fechada, as ligações dos elos acontecem também de forma sequencial em circuito fechado, porém, para este caso, existem pelo menos dois caminhos que possibilitam a ligação de um elo a outro. A junção de uma cadeia cinemática aberta com uma cadeia cinemática fechada resulta em uma cadeia cinemática híbrida. A Figura 1 apresenta os tipos de cadeias cinemáticas.

As cadeias cinemáticas podem variar de acordo com as quantidades de elos e juntas, o que está relacionado diretamente ao número de graus de liberdade (GDL) (TSAI, 1999). A Figura 2 apresenta os tipos tradicionais de juntas que pode ser utilizadas em um robô.

Atualmente, os robôs bípedes são capazes de andar, correr ou até mesmo realizar um salto, mas, não possuem a flexibilidade em termos de números de GDLs. Para

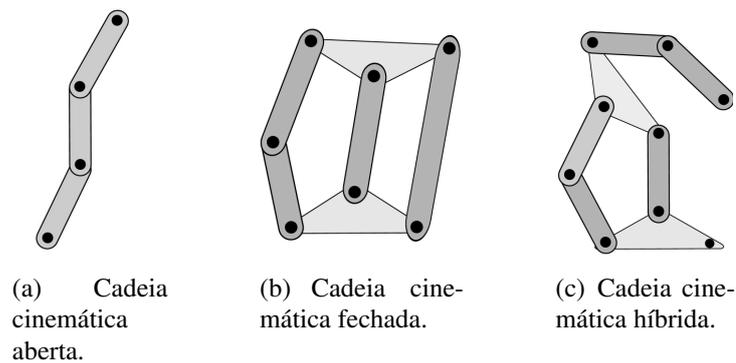


Figura 1: Tipos de cadeias cinemáticas.

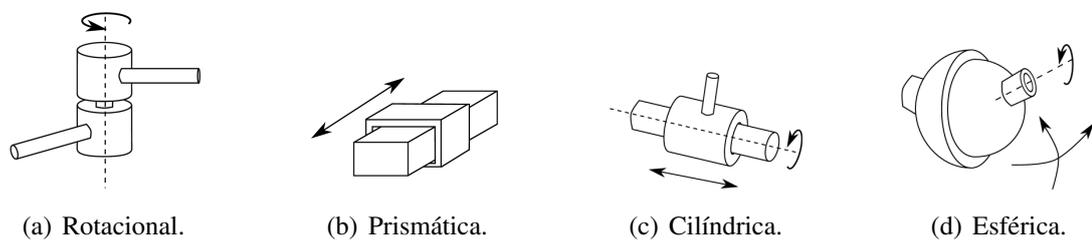


Figura 2: Tipos tradicionais de juntas.

serem eficientes em qualquer superfície sua elasticidade deve ser alta, já que o grau de elasticidade está relacionado ao GDL de cada articulação do robô (HURST; RIZZI, 2008).

As articulações de um robô são definidas por um ou mais GDLs, possibilitando a realização de movimentos no espaço bidimensional ou tridimensional. Por exemplo, quando o movimento relativo ocorre em um único eixo (x, y ou z), a articulação tem um GDL, caso o movimento se dê em mais de um eixo (x, y ou z), a articulação passa a ter dois ou mais GDL (SICILIANO; SCIavicco, 2009). A Figura 3 ilustra um robô bípede e seus respectivos GDLs.

Observa-se que quanto maior o número de GDLs, mais complexa é a cinemática, dinâmica e o controle do robô. O número de GDL de um robô está associado ao número de variáveis independentes, que permitem definir a posição de todas as partes de forma única (SICILIANO; SCIavicco, 2009).

A cinemática para robôs bípedes é baseada na mobilidade e na funcionalidade do corpo humano. No entanto, o corpo humano possui articulações e sistemas musculares extremamente complexos. Ao desenvolver um robô bípede, se faz necessário a análise dos graus de liberdade que se deseja ter em cada articulação. Essa abordagem consiste em realizar uma análise da estrutura do robô nos principais planos: Plano sagital ou lateral, plano frontal e no plano transversal. A Figura 4 mostra os planos do corpo

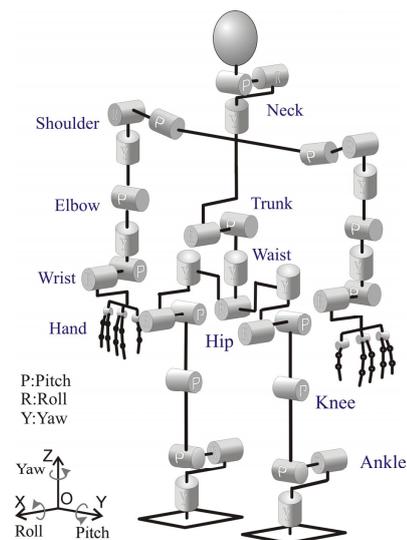


Figura 3: Robô WABIAN-2 possui 41 GDLs (OGURA et al., 2006).

humano. O andar humano e o andar robótico são realizados principalmente no plano sagital. É nesse plano que todos os bípedes têm o maior número de articulações e graus de liberdades (CHEVALLEREAU et al., 2009).

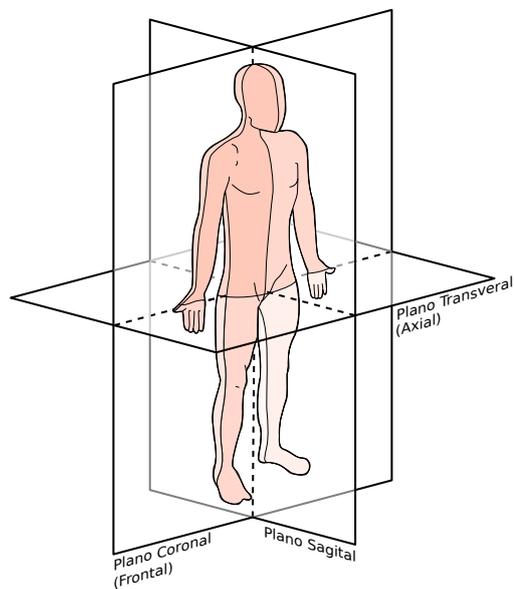


Figura 4: Plano sagital ou lateral, plano frontal e no plano transversal de um ser humano.

A estrutura apresentada na Figura 5 (a) corresponde à construção de um robô bípede simples, que tem somente duas articulações no quadril. Na Figura 5 (b) o robô bí-

pede possui duas articulações adicionais nos joelhos, e na última figura, a Figura 5 (c), complementa as estruturas anteriores com a adição de duas articulações adicionais nos tornozelos. Essa é a configuração que é mais frequentemente utilizada para a construção de protótipos de robôs bípedes (CHEVALLEREAU et al., 2009).

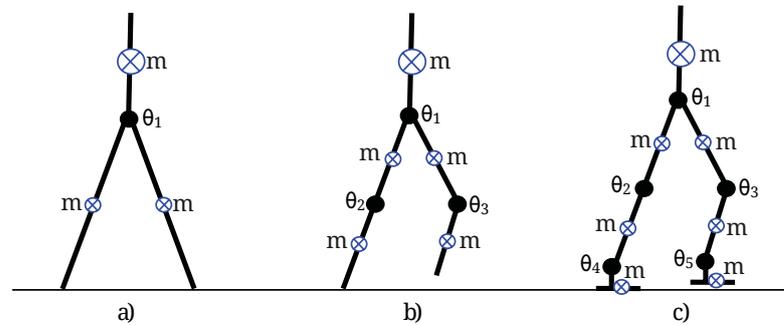


Figura 5: Graus de liberdade de um robô bípede no plano sagital (CHEVALLEREAU et al., 2009).

Ao realizar uma análise das articulações no plano frontal pode-se verificar a importância da estabilidade lateral para os pés. Na Figura 6 pode-se verificar a solução adotada para que o robô bípede tenha uma melhor estrutura mecânica e seja mais estável ao se locomover. A estabilidade vista pelo plano frontal depende muito da posição da postura da perna em relação ao centro de gravidade do robô bípede (CHEVALLEREAU et al., 2009).

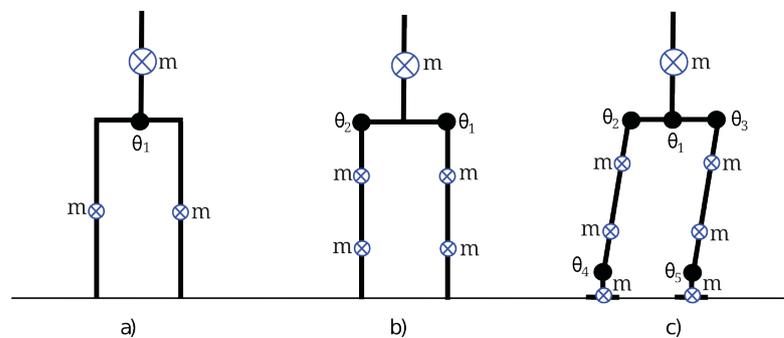


Figura 6: Graus de liberdade de um robô bípede no plano frontal (CHEVALLEREAU et al., 2009).

A estrutura da Figura 6 (a) permite o deslocamento do centro de gravidade, com esse deslocamento é possível obter uma flexão lateral que é realizada apenas no quadril, sem mexer as pernas. Ao utilizar a estrutura da Figura 6 (b) torna-se possível mover lateralmente o centro de gravidade, devido às articulações do quadril. Dessa forma é possível obter um movimento independente de cada perna e manter o equilíbrio do

centro de gravidade para cada perna. A estrutura da Figura 6 (c) permite efetuar um deslocamento do centro de gravidade por meio da combinação de um duplo movimento de inclinação do quadril e dos tornozelos. Mantendo os pés no plano, o que garante uma melhor aderência ao solo (CHEVALLEREAU et al., 2009). As articulações no plano transversal só existem em robôs bípedes mais complexos. Estes robôs têm de uma a cinco articulações no plano transversal.

A estrutura da Figura 7 (a) tem movimento de rotação do tronco e permite compensar as reações no plano transversal, devido ao movimento da perna de equilíbrio. A estrutura da Figura 7 (b) permite que o robô gire a perna de equilíbrio por um movimento de rotação interno-externa, o que lhe permite fazer mudanças de direção ao andar. Finalmente, a estrutura da Figura 7 (c) possui mais flexibilidade que as estruturas anteriores, permitindo orientar o pé quando ele está prestes a tocar no solo (CHEVALLEREAU et al., 2009).

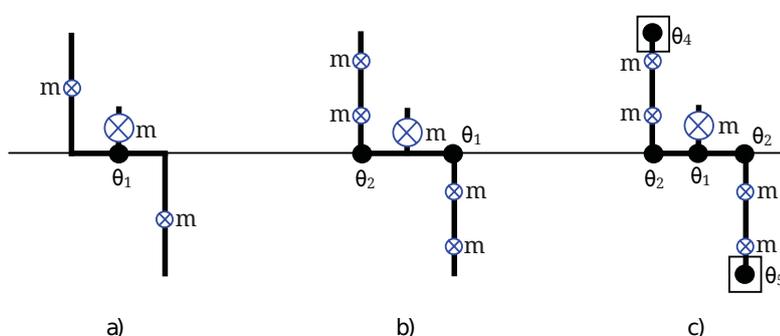


Figura 7: Graus de liberdade de um robô bípede no plano transversal (CHEVALLEREAU et al., 2009).

2.2 Dinâmica

O modelo dinâmico de um robô com n graus de liberdade pode ser descrito como sendo um conjunto de equações matemáticas que representam o seu comportamento dinâmico (FU; GONZALEZ; LEE, 1987). Essas equações podem ser representadas por:

$$\tau = D(q)\ddot{q} + H(q, \dot{q}) + G(q) \quad (1)$$

onde, q é o vetor de posição das juntas, $D(q)$ é a matriz de inércia, $H(q, \dot{q})$ é o vetor de forças centrífugas e forças de coriolis, $G(q)$ é o vetor de forças gravitacionais e, finalmente τ é o vetor de torques nas juntas.

Esta seção aborda duas formulações clássicas, que servem para a obtenção de um modelo dinâmico: a) Formulação de Lagrange-Euler e b) Formulação de Newton-Euler.

2.2.1 Formulação de Lagrange-Euler

Segundo FU; GONZALEZ; LEE (1987); CRAIG (1989) a formulação de Lagrange-Euler tem por base a mecânica lagrangiana, e resulta em um modelo estruturado. Com o uso deste método, é possível identificar os termos de momento de inércia, forças centrífugas, forças de coriolis e forças gravitacionais envolvidas no modelo. A formulação Lagrangiana é definida como sendo a diferença entre a energia cinética (K) e a energia potencial (P) de um sistema mecânico, sendo:

$$L = K - P \quad (2)$$

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i [Tr(U_{ij} J_i U_{ik}^T) \dot{q}_j \dot{q}_k] + \sum_{i=1}^n m_i g ({}^0 A_i {}^i \bar{r}_i) \quad (3)$$

onde, na expressão (3) vê-se que o primeiro somatório representa a energia cinética de um manipulador, a qual pode ser descrita por uma formula escalar, como uma função da posição e da velocidade da junta. O segundo somatório representa a energia potencial total armazenada no manipulador.

As equações de movimento para o manipulador são dada por:

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (4)$$

onde, τ_i é o vetor $n \times 1$ dos torques do atuador, expresso por meio de:

$$\tau_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{k=1}^n \sum_{m=1}^n H_{ikm} \dot{q}_k \dot{q}_m + C_i \quad (5)$$

ou na forma matricial:

$$\tau_i(t) = D(q(t)) \ddot{q}(t) + H(q(t), \dot{q}(t)) + C(q(t)) \quad (6)$$

onde,

- $\tau_i(t)$ é um vetor $n \times 1$, que descreve o torque aplicado em uma determinada junta i , expresso por:

$$\tau_i(t) = [\tau_1(t) \tau_2(t) \dots \tau_n(t)]^T \quad i = 1, 2, \dots, n; \quad (7)$$

- $q(t)$ é um vetor $n \times 1$, que descreve o ângulo das juntas do manipulador;

$$q(t) = [q_1(t) q_2(t) \dots q_n(t)]^T; \quad (8)$$

- $\dot{q}(t)$ é um vetor $n \times 1$, que descreve a velocidade para um conjunto de determinadas posições de junta;

$$\dot{q} = [\dot{q}_1(t) \dot{q}_2(t) \dots \dot{q}_n(t)]^T; \quad (9)$$

- $\ddot{q}(t)$ é um vetor $n \times 1$, que descreve um a aceleração para uma determinada junta;

$$\ddot{q} = [\ddot{q}_1(t) \ddot{q}_2(t) \dots \ddot{q}_n(t)]^T; \quad (10)$$

- $D(q(t))$ é uma matriz $n \times n$, é relacionada ao momento de inercia do sistema;

$$D(q(t)) = \sum_{j=\max(i,k)}^n Tr(U_{jk}J_jU_{ji}^T) \quad i, k = 1, 2, \dots, n; \quad (11)$$

- $H(q(t), \dot{q}(t))$ é uma matriz $n \times 1$, que descreve a forças centrífugas e de coriolis do sistema;

$$H(q(t), \dot{q}(t)) = [h_1 h_2 \dots h_n]^T; \quad (12)$$

- $C(q(t))$ é uma matriz $n \times n$, que descreve um vetor de forças gravitacionais do sistema:

$$C(q(t)) = (c_1, c_2, \dots, c_n)^T; \quad (13)$$

sendo C_i expressado por:

$$C_i = \sum_{j=\max(j=i)}^n (-m_j g U_{ji}^j \bar{r}_j) \quad i = 1, 2, \dots, n \quad (14)$$

e g é a aceleração da gravidade, enquanto que ${}^j\bar{r}_j$ é a coordenada do centro de gravidade do elo j_i em relação ao sistema de coordenadas j .

2.2.2 Formulação de Newton-Euler

Embora a formulação de Lagrange-Euler descreva praticamente todos os fenômenos físicos relacionados aos torques aplicados nas junta de um robô, o método é ineficiente computacionalmente. No entanto segundo FU; GONZALEZ; LEE (1987); CRAIG (1989) a formulação de Newton-Euler é mais eficiente computacionalmente e possibilita a implementação das leis de controle em tempo real, pois reduz o número de operações matemáticas a serem executadas no laço de controle. As equações dinâmicas da formulação de Newton-Euler, formam um conjunto de equações recursivas compactas chamadas de diretas e reversas que podem ser aplicadas a um robô de elos sequenciais.

As equações recursivas diretas operam com informações cinemáticas de velocidade, aceleração angular e aceleração translacional, relacionando os elos desde a base ao órgão terminal do manipulador. As equações recursivas inversas operam com as forças exercidas desde o efetuador até a base do manipulador, os torques nas juntas são calculados a partir destas forças.

Seja uma matriz de rotação 3×3 , que transforma qualquer vetor no sistema de coordenadas de referência $(x_{i-1}, y_{i-1}, z_{i-1})$ para o sistema (x_i, y_i, z_i) .

- A transformação na ordem direta ocorre da seguinte forma para as interações:
 $i = 1, 2, \dots, n$

$${}^i R_0 \boldsymbol{\omega}_i = \begin{cases} {}^i R_{i-1} ({}^{i-1} R_0 \boldsymbol{\omega}_{i-1} + z_0 \dot{q}_i) & \text{junta rotacional} \\ {}^i R_{i-1} ({}^{i-1} R_0 \boldsymbol{\omega}_{i-1}) & \text{junta prismática} \end{cases} \quad (15)$$

$${}^i R_0 \dot{\boldsymbol{\omega}}_i = \begin{cases} {}^i R_{i-1} [{}^{i-1} R_0 \dot{\boldsymbol{\omega}}_{i-1} + z_0 \ddot{q}_i + ({}^{i-1} R_0 \boldsymbol{\omega}_{i-1}) z_0 \dot{q}_i] & \text{junta rotacional} \\ {}^i R_{i-1} ({}^{i-1} R_0 \dot{\boldsymbol{\omega}}_{i-1}) & \text{junta prismática} \end{cases} \quad (16)$$

$${}^i R_0 \dot{v}_i = \begin{cases} ({}^i R_0 \dot{\boldsymbol{\omega}}_i) ({}^i R_0 p_i^*) + ({}^i R_0 \boldsymbol{\omega}_i) [({}^i R_0 \boldsymbol{\omega}_i) ({}^i R_0 p_i^*)] + {}^i R_{i-1} ({}^{i-1} R_0 \dot{v}_{i-1}) & \text{junta rotacional} \\ {}^i R_{i-1} (z_0 \ddot{q}_i + {}^{i-1} R_0 \dot{v}_{i-1}) + ({}^i R_0 \dot{\boldsymbol{\omega}}_i) ({}^i R_0 p_i^*) + 2({}^i R_0 \boldsymbol{\omega}_i) ({}^i R_{i-1} z_0 \dot{q}_i) + ({}^i R_0 \boldsymbol{\omega}_i) [({}^i R_0 \boldsymbol{\omega}_i) ({}^i R_0 p_i^*)] & \text{junta prismática} \end{cases} \quad (17)$$

$${}^i R_0 \bar{a}_i = ({}^i R_0 \dot{\boldsymbol{\omega}}_i) ({}^i R_0 \bar{s}_i) + ({}^i R_0 \boldsymbol{\omega}_i) [({}^i R_0 \boldsymbol{\omega}_i) ({}^i R_0 \bar{s}_i)] + {}^i R_0 \dot{v}_i \quad (18)$$

- A transformação na ordem inversa ocorre da seguinte forma para as interações:
 $i = n, n-1, \dots, 1$

$${}^i R_0 f_i = {}^{i+1} R_{i+1} ({}^i R_0 f_{i+1}) + m_i {}^i R_0 \bar{a}_i \quad (19)$$

$${}^i R_0 n_i = {}^i R_{i+1} [{}^{i+1} R_0 n_{i+1} + ({}^i R_0 p_i^*) ({}^{i+1} R_0 f_{i+1})] + ({}^i R_0 p_i^* + {}^i R_0 \bar{s}_i) ({}^i R_0 F_i) + ({}^i R_0 I_i^0 R_i) ({}^i R_0 \dot{\boldsymbol{\omega}}_i) + ({}^i R_0 \boldsymbol{\omega}_i) [({}^i R_0 I_i^0 R_i) ({}^i R_0 \boldsymbol{\omega}_i)] \quad (20)$$

$$\tau_i = \begin{cases} ({}^iR_0n_i)^T ({}^iR_{i-1}z_o) + b_i\dot{q}_i & \text{junta rotacional} \\ ({}^iR_0f_i)^T ({}^iR_{i-1}z_o) + b_i\dot{q}_i & \text{junta prismática} \end{cases} \quad (21)$$

onde,

- m_i : Massa do elo i ;
- τ_i : Torque na junta i ;
- z_o : $[0 \ 0 \ 1]^T$;
- ${}^iR_0\omega_i$: Velocidade angular no elo i , representado no sistema de coordenadas i ;
- ${}^iR_0\dot{\omega}_i$: Aceleração angular no elo i , representado no sistema de coordenadas i ;
- ${}^iR_0\dot{v}_i$: Velocidade linear no elo i , representado no sistema de coordenadas i ;
- ${}^iR_0\ddot{a}_i$: Aceleração linear no centro de gravidade do elo i , representado no sistema de coordenadas i ;
- ${}^iR_0p_i^*$: Posição da origem do sistema de coordenadas i em relação ao sistema de coordenadas $i - 1$ representado no sistema de coordenadas i ;
- ${}^iR_0\bar{s}_i$: Posição do centro de gravidade do elo i em relação ao sistema de coordenadas i representado no sistema de coordenadas i ;
- iR_0f_i : Força atuante no elo i representado no sistema de coordenadas i ;
- iR_0n_i : Momento atuante no elo i representado no sistema de coordenadas i ;
- ${}^iR_0I_i^0 R_i$: Tensor de inércia do elo i em torno do seu centro de massa representado em um sistema de coordenadas com origem no centro de massa e com a mesma orientação do sistema de coordenadas associado ao elo i .

Da aplicação do algoritmo de Newton-Euler, obtêm-se como esperado a Equação (1).

2.3 Sistema de Controle

A área de pesquisas em robôs bípedes e humanóides tem passado por uma grande evolução nas técnicas de controle. Os robôs bípedes possuem uma dinâmica complexa e seus movimentos tendem a se equivaler aos movimentos humanos. Além disso, existe um fator que está ligado à segurança que deve ser implementado neste tipo de robô, de modo que eles possam cooperar com os seres humanos em atividades rotineiras. Com este objetivo, a técnica de controle de torque permitem que os robôs bípedes atendam a requisitos desta ordem (DEL PRETE; SENTIS, 2013).

Segundo os autores (HEMAMI; KATBAB, 1982; LEE, 1988; HURMUZLU, 1998; YANG, 1994; JALICS; HEMAMI; CLYMER, 1997), a técnica de controle por torque calculado pode ser utilizada no desenvolvimento de controladores de movimentos para robôs bípedes. Esta técnica utiliza o modelo dinâmico do robô para implementar uma linearização por realimentação (SLOTINE; LI, 1991), por meio da estrutura de controle apresentada na Figura 8. Segundo esta representação, as linhas cheias indicam entradas para os blocos que são multiplicadas pelo "ganho" do bloco, enquanto as linhas tracejadas indicam as entradas que são apenas utilizadas para calcular a função descrita pelo bloco.

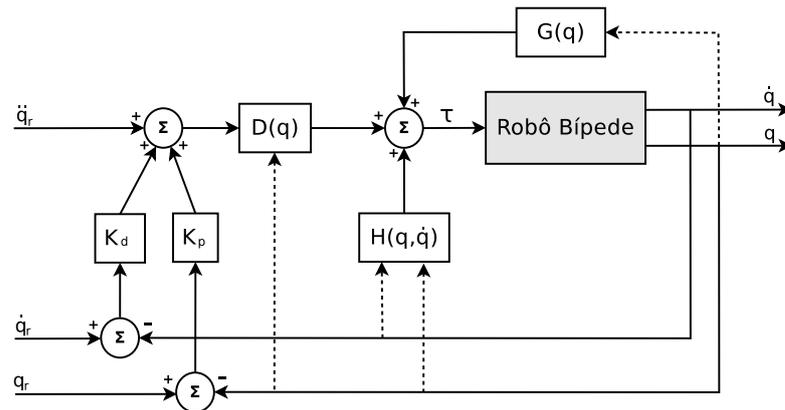


Figura 8: Diagrama de blocos do controle por torque calculado (Modificada de (FU; GONZALEZ; LEE, 1987)).

Nota-se no diagrama de blocos da Figura 8, que o controlador PD é utilizado para controlar o sistema linear resultante. Assim, a lei de controle resulta:

$$\tau = D_n(q)[\ddot{q}_r + K_d(\dot{q}_r - \dot{q}) + K_p(q_r - q)] + H_n(q, \dot{q}) + G_n(q) \quad (22)$$

onde,

- q_r : é a referência de posição;

- K_p : é a matriz de ganhos proporcionais;
- K_d : é a matriz de ganhos derivativos;
- $D_n(q)$: é a matriz de inércia;
- $H_n(q, \dot{q})$: é o vetor de forças centrífugas e forças de Coriolis;
- $G_n(q)$: Vetor de forças gravitacionais.

Note-se que se pode utilizar a forma clássica das expressões de Newton-Euler para calcular a lei de controle por torque calculado, fazendo-se:

$$\ddot{q} = \ddot{q}_r + K_d(\dot{q}_r - \dot{q}) + K_p(q_r - q) \quad (23)$$

Se não houver erros nos parâmetros do modelo, ou seja, se $D(q) = D_n(q)$, $H(q, \dot{q}) = H_n(q, \dot{q})$, $G(q) = G_n(q)$ obtém-se:

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \quad (24)$$

A Equação (24) demonstra que da escolha das matrizes K_p e K_d de forma diagonal, obtêm-se em malha fechada um sistema desacoplado, onde o comportamento do erro de cada junta obedecerá a uma equação diferencial de segunda ordem, na qual pode-se determinar com a frequência natural ω_n e o coeficiente de amortecimento ξ_n por meio da escolha adequada das matrizes de ganho do controlador ¹.

2.4 Robôs Bípedes

Esta seção apresenta uma breve história sobre os principais robôs bípedes, bem como algumas pesquisas que estão sendo realizadas na áreas dos robôs bípedes. Apresenta-se também dois robôs bípedes que utilizam o ROS como plataforma de desenvolvimento.

Algumas pesquisas e desenvolvimentos na área de robôs bípedes no Brasil têm sido realizadas em alguns importantes centros de pesquisas, tais como: UFRJ, USP e a UNICAMP. Existem alguns trabalhos realizados no Brasil, como, por exemplo, o trabalho de BEZERRA; ZAMPIERI (2004), onde os autores descrevem o estado da arte dos robôs bípedes e o trabalho de SIQUEIRA; TERRA (2006), onde os autores apresentam o uso de um sistema de controle baseado no método H_∞ , que é aplicado em um robô bípede de pequeno porte.

Na Figura 9 (a) tem-se o protótipo de um robô bípede de pequeno porte desenvolvido pela UFRJ (TORRES, 2006). Seu caminhar é estático e possui apenas um atuador em cada perna, dispondo de um mecanismo independente que realiza a variação do seu centro de gravidade. Sua velocidade de deslocamento é baixa, pois o seu centro de

¹Para maiores esclarecimento vide (OGATA, 2010).

gravidade é constantemente deslocado para a perna de apoio durante a sua locomoção. O controle deste robô é simples, pois utiliza chaves de fim de curso para ativar e desativar o seu sistema de controle, determinando-se o início e o fim de cada movimento realizado durante sua locomoção.

Na Figura 9 (b) tem-se o protótipo de um robô bípede de pequeno porte desenvolvido pela USP. O trabalho de SANTANA (2005) descreve o desenvolvimento de um robô bípede antropomórfico, que serve como ferramenta de auxílio ao estudo sobre a marcha humana aos profissionais da área da saúde. O robô desenvolvido por Santana é capaz de reproduzir os padrões de marcha humana, cujo os dados para realizar essa marcha. O robô possui dez graus de liberdade e apresenta um caminhar estático, seu sistema de controle é baseado em uma técnica simples e muito conhecida, denominada de controle independente por juntas. Sistema esse que considera que cada atuador realiza uma ação de forma independente.

Na Figura 9 (c) tem-se o protótipo de um robô bípede de pequeno porte desenvolvido pelo Centro de Tecnologia da UNICAMP. Seu caminhar é estático e tem o propósito de realizar a locomoção em ambientes desconhecidos, podendo se deslocar para frente ou para trás a uma velocidade aproximada de 1 cm/s (BEZERRA, 2002). Esse protótipo é composto por sete elos em alumínio (dois pés, duas pernas, duas coxas e uma pelve), seis juntas de rotação (duas no tornozelo, duas nos joelhos e duas no quadril) e seis servomotores.

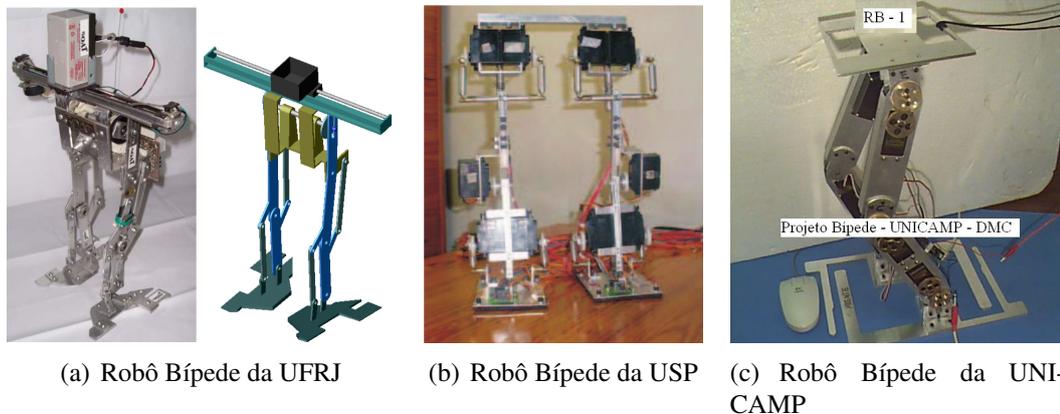
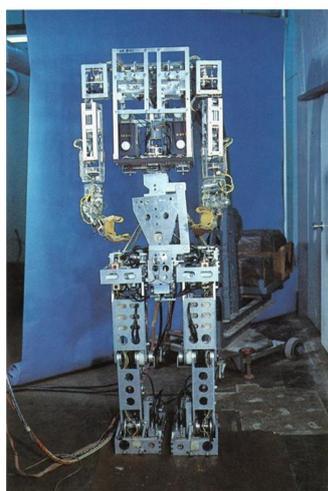


Figura 9: Robôs Bípedes desenvolvidos no Brasil.

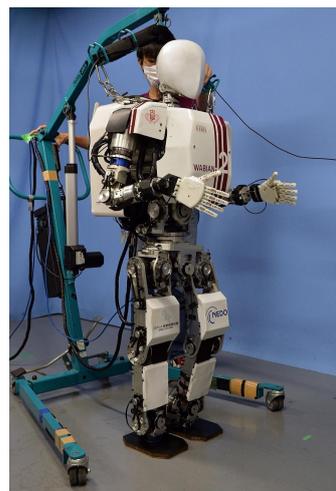
Existem diversos grupos de pesquisa ao redor do mundo focados no desenvolvimento de robôs bípedes (HASEHAWA; ARAKAWA; FUKUDA, 2000), sendo capazes de imitar os movimentos ou até mesmo o comportamento humano (WAARD; INJA; VISSER, 2013).

O grupo de pesquisa sobre robôs bípedes e humanoides da Universidade Waseda no Japão tem uma longa história de estudos e desenvolvimentos. Este grupo de pes-

quiza vem fazendo muitas contribuições significativas desde 1966. O Dr. Ichiro Kato, professor da Universidade Waseda, é conhecido como o pai da robótica dos bípedes, dedicando sua vida à pesquisa no desenvolvimento e locomoção dos bípedes, realizando grandes avanços na área (KATO et al., 1974). Em 1973, a Universidade Waseda desenvolveu o seu primeiro robô denominado de WABOT-1, que tornou-se a primeira de uma série bem conhecida de robôs bípedes capaz de andar. Atualmente a Universidade Waseda continua suas investigação e estudos na área dos bípedes. A Figura 10 (a) e a Figura 10 (b) ilustra as versões dos bípedes da Universidade Waseda.



(a) Robô WABOT-1 (1973)



(b) Robô WABIAN-2 (2010)

Figura 10: Robôs desenvolvidos pela Universidade Waseda.

Um dos robôs humanóides mais famosos atualmente é o robô ASIMO, desenvolvido pela empresa japonesa Honda.

Com o objetivo de disponibilizar uma melhor qualidade de vida as pessoas, a Honda iniciou em 1986 suas pesquisas na área de robôs bípedes, procurando desenvolver robôs que pudessem se adaptar bem às condições de vida em sociedade e substituir alguns trabalhos que são tradicionalmente realizados somente por seres humanos.

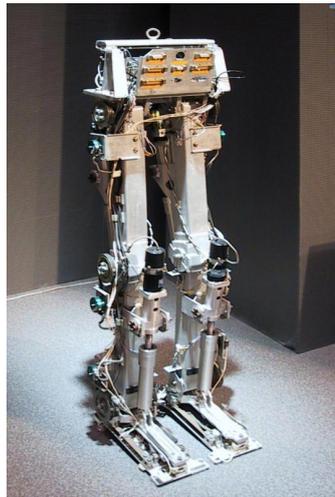
Neste tipo de pesquisa, a Honda elencou alguns tópicos importantes como pré-requisitos para esse tipo de desenvolvimento, dentro deles estão: desviar obstáculos, subir e descer escadas, caminhar em terrenos não planos e operar em uma vasta gama de ambientes (HIROSE; HAIKAWA; TAKENAKA, 1998).

A Figura 11 (a) ilustra o primeiro protótipo da Honda, da geração E0. Esse modelo era considerado lento, pois levava 30 segundos para realizar um passo, sua caminhada era estática e realizada apenas em linha reta. Nos anos seguintes, de 1987 a 1993, novos avanços foram realizados com os robôs da geração E2 e E6. Os robôs da geração E2 foram os primeiros robôs da Honda a possuir habilidades de locomoção dinâmica

e de subir e descer escadas. Na geração E6, o sistema de controle foi aprimorado e a série recebeu sensores para mapear o ambiente. Em 1993, novos modelos foram lançados, como o protótipo P1, que possuía braços para manipular objetos e auxiliá-lo no equilíbrio durante a caminhada. No ano de 1996, o robô P2 apresentou uma grande evolução em relação aos modelos anteriores. Possuía sistema de visão, comunicação sem fio, capacidade de compensar forças externas, detetor de colisão e conseguia caminhar em ambientes não planos.

Em 2000, a Honda apresentou ao público o novo modelo do robô ASIMO. A Figura 11 (b) ilustra esse modelo. Até o presente momento, esse modelo é um dos robôs bípedes mais avançados do mundo. Com 1 metro e 20 centímetros de altura, pesando 43 Kg, possui as dimensões adequadas para a interação em ambientes domésticos, podendo abrir portas, ligar e desligar luzes, subir e descer escadas, etc.

Atualmente, muitas melhorias vêm sendo aplicadas no ASIMO, como mapeamento de visão, reconhecimento facial e gestual, reconhecimento de fala, comunicação com pessoas, gerenciamento do consumo de energia (SAKAGAMI et al., 2002). Para o planejamento de trajetórias, foi implementado o reconhecimento de obstáculos, onde sua programação é otimizada para que o sistema realize uma parada suave ou para desviar obstáculos (CHESTNUTT et al., 2005).



(a) Robô Honda E0



(b) Robô ASIMO

Figura 11: Robôs desenvolvidos pela Honda Motor Co., Ltd.

Outros robôs humanoides que possuem habilidades similares ao ASIMO, pertencem a série chamada HRP, desenvolvidos pela AIST e Kawada Industries Inc. Um dos robôs desta série é o HRP-2, ilustrado na Figura 12 (a). Este robô, diferente do ASIMO, tem a capacidade de se levantar a partir do chão, de frente ou de costas, algo que o

robô ASIMO da Honda não é capaz de fazer. O outro modelo é o HRP-4C, ilustrado na Figura 12 (b), consegue imitar movimentos da cabeça e face de um ser humano, bem como executar passos de dança (KANEKO et al., 2011).



(a) Robô HRP-2



(b) Robô HRP-4

Figura 12: Robôs desenvolvidos pela AIST e Kawada Industries Inc.

Em 2000, a universidade técnica de Munique desenvolveu o robô JOHNNIE, ilustrado na Figura 13 (a). Trata-se de um robô bípede autônomo e seu objetivo inicial era o de realizar uma caminhada antropomórfica dinamicamente estável, onde seja possível movimentos de curvas ou em linha reta em superfícies planas ou irregulares. O robô JOHNNIE tem 1 metro e 80 centímetros de altura, 48 Kg, 23 graus de liberdade, é autônomo em termos de sensores, atuadores e poder computacional (PFEIFFER; LOFFLER; GIENGER, 2002) e sua velocidade de caminhada é de 2,4 Km/h (LOHMEIER et al., 2004).

Seu sistema de controle é composto por três camadas. Na camada superior é realizado o cálculo das trajetórias e a mudança entre os diferentes padrões de caminhada. Na segunda camada ocorre as ações de estabilidade do tronco, por meio dos sensores de força/torque, para manter o seu balanço e, na terceira camada, é realizado o controle de posição, velocidade e aceleração das juntas através de uma estratégia de realimentação linear e PID com observador de atrito (GIENGER; LOFFLER; PFEIFFER, 2001). Possui também um sistema de visão que lhe permite localizar os obstáculos e optar por desviar, passar por cima ou pisar sobre eles.

Em 2009, a mesma universidade lançou o robô humanoide LOLA, ilustrado na Fi-

gura 13 (b), com 25 GDLs, sendo 7 GDLs para cada perna. Esse modelo realiza movimentos rápidos que se assemelham aos movimentos de um ser humano, podendo chegar a velocidade de 5 Km/h (LOHMEIER; BUSCHMANN; ULBRICH, 2009).

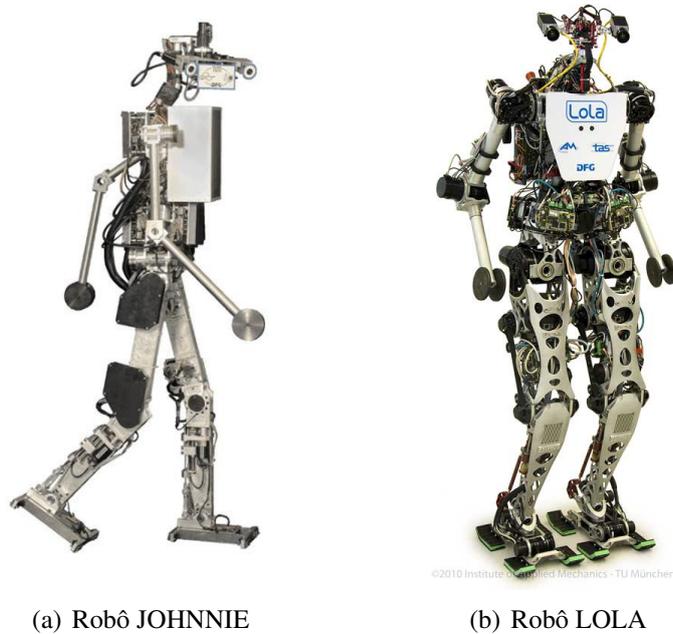
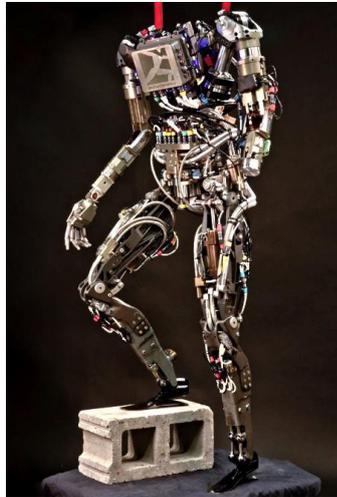


Figura 13: Robôs desenvolvidos pela Universidade Técnica de Munique.

A Figura 14 (a) ilustra o robô bípede PETMAN lançado em 2009 pela empresa Boston Dynamics (RAIBERT, 2010). O protótipo do PETMAN também demonstrou a capacidade de se equilibrar durante a caminhada em um nível superior ao demonstrado pelos seus antecessores. A partir de 2013, o robô ATLAS, ilustrado na Figura 14 (b), vem sendo desenvolvido, com o apoio da DARPA Robotics Challenge.

Em dezembro de 2013, a NASA, apresentou seu novo modelo na área de robôs bípedes. A Figura 15 (b) ilustra o robô Valkyrie. Esse projeto é derivado do Robonaut, ilustrado na Figura 15 (a), que está trabalhando na estação espacial internacional. Este novo robô tem a altura de 1,9 metros, 125 Kg e possui 44 graus de liberdade. A NASA pretende utilizar esse robô para entrar em zonas de desastres, com objetivo de realizar buscas e possíveis salvamentos. A NASA vem desenvolvendo a próxima geração de robôs humanoides, e a perspectiva da NASA é solucionar alguns problemas de controle que existem ainda na área dos robôs bípedes, como, por exemplo, realizar tarefas dinâmicas com mais agilidade em diferentes ambientes.



(a) Robô PETMAN



(b) Robô ATLAS

Figura 14: Robôs desenvolvidos pela Boston Dynamics.



(a) Robô Robonaut



(b) Robô Valkyrie

Figura 15: Robôs desenvolvidos pela NASA Johnson Space Center.

2.5 Robôs Bípedes desenvolvidos utilizando ROS

Atualmente, existem alguns modelos de robôs bípedes que utilizam o ROS. A Figura 16 ilustra dois dos robôs bípedes que utilizam o ROS.

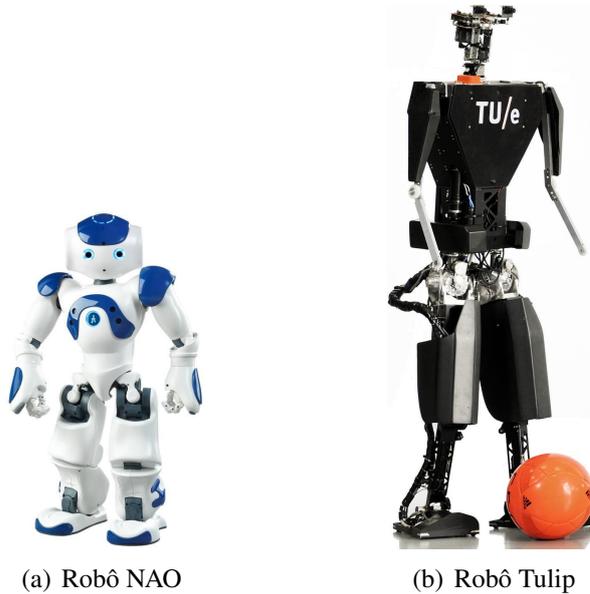


Figura 16: Robôs bípedes desenvolvidos utilizando o ROS.

Em 2004 a empresa Francesa Aldebaran Robotics desenvolveu o robô NAO. Este modelo possui 58 centímetros de altura e 25 graus de liberdade, rede de sensores, dispositivos de comunicação, um processador Intel ATOM 1,6 Ghz, sistema operacional Linux e bateria de 27,6 W/h que fornece ao robô uma autonomia de 1,5 horas ou mais, dependendo do uso (GOUAILLIER et al., 2009). Informações deste projeto estão disponíveis no repositório do ROS (ARMIN, 2013).

Em 2009 foi desenvolvido o robô humanoide denominado de Tulip pelo grupo de dinâmica e controle do departamento de engenharia mecânica da universidade de tecnologia de Eindhoven em cooperação com a universidade de tecnologia de Delft e a universidade de Twente. O robô possui 125 centímetros de altura, pesa cerca de 25 Kg, e tem duas pernas, cada uma com seis GDLs. Possui uma unidade de medição inercial e na cabeça um sistema de câmera estereoscópica. Forças de contato com o solo são medidas através de sensores de pressão sob seus pés. A pesquisa se concentra na dinâmica e no equilíbrio do ser humano, visão computacional, localização e planejamento de trajetória para garantir a operação autônoma dos pés do robô em ambientes naturais, tais como casas, escritórios e hospitais (VAN ZUTVEN et al., 2013). Informações deste projeto estão disponíveis no repositório do ROS (PIETER, 2013).

3 MODELAGEM E SIMULAÇÃO

Atualmente é comum a utilização de simulações computacionais na área da robótica, com o intuito de antever a resposta dinâmica de um sistema mecânico. Ao realizar a simulação do comportamento de um sistema, o tempo de desenvolvimento de um projeto pode ser reduzido, pois não é necessário a construção de um modelo físico para a obtenção do comportamento do sistema modelado. Nas últimas três décadas, foram desenvolvidos vários programas com o foco em simulação de multicorpos, formalismo e métodos computacionais. Segundo SCHIEHLEN (1990), quando se faz necessário escolher um programa para obter uma simulação mais realista para um problema específico, julga-se importante pesquisar e analisar os tipos de programas que estão disponíveis no mercado atual. Pode-se utilizar como base os seguintes requisitos:

1. Topologia de sistema: é importante definir a topologia do sistema para obter a solução do problema proposto. Em geral, os sistemas mecânicos podem ser classificados como sendo mecanismos de cadeias abertas, cadeias fechadas e cadeias híbridas.
2. Elementos de modelagem: em geral, todos os programas disponibilizam a modelagem de corpos rígidos, partículas, juntas e elementos passivos como molas ou amortecedores. Alguns programas disponibilizam a modelagem de corpos flexíveis e elementos ativos, como atuadores.
3. Sistemas de coordenadas: a maioria dos programas de simulação permitem a utilização de sistemas de coordenadas inerciais.
4. Métodos computacionais: existem os métodos computacionais numéricos e simbólicos, ou até mesmo a combinação entre os dois métodos.
5. Métodos dinâmicos: os métodos mais utilizados em simulação são as equações de Lagrange-Euler e as equações de Newton-Euler.
6. Métodos de simulação: são recursos de integração numérica para a resolução das equações diferenciais.

7. Tipos de equações: as equações geradas pelo programa de simulação consistem de equações diferenciais ordinárias ou equações algébricas. Alguns programas possibilitam realizar a geração das equações destes dois tipos já linearizadas.
8. Pré-processamento: por algum tempo não era possível realizar uma modelagem matemática com o auxílio de uma interface. Os dados eram inseridos via editor de texto e compilados, obtendo-se assim um resultado. Atualmente, os simuladores disponibilizam interfaces que simplificam a entrada de dados.
9. Pós-processamento: no início do uso de simuladores e até hoje, a análise das grandezas envolvidas na simulação é descrita por meio de gráficos. Atualmente, simuladores mais modernos reproduzem animações muito realísticas, permitindo dessa forma que o usuário tenha uma visão ampla e integrada do comportamento dinâmico do sistema.
10. Área de aplicação: atualmente, existem a disposição do usuário diversos programas de simulação que possibilitam uma ampla gama de aplicações da teoria de multicorpos, incluindo veículos ferroviários e rodoviários, aeronaves, satélites, robôs, máquinas e até mesmo estudos de ergonomia.

Após o estudo e análise dos requisitos descritos por SCHIEHLEN (1990), a escolha do programa de simulação utilizado neste trabalho foi o Gazebo (?). O Gazebo é uma ferramenta voltada ao desenvolvimento de aplicações para a área da robótica e está integrado ao ROS. Atualmente existe uma grande comunidade internacional que utiliza esse programa. Um dos motivos mais importantes de se utilizar este simulador é a disponibilidade de obter acesso a múltiplas ferramentas relacionadas à física de multicorpos no Gazebo, incluindo *middlewares* como o ODE (RUSSELL, 2005) e o Simbody (SHERMAN; SETH; DELP, 2011).

A seguir serão apresentados alguns conceitos básicos sobre o ROS e algumas ferramentas que fazem parte do escopo do presente trabalho.

3.1 Robot Operating System (ROS)

O ROS inclui bibliotecas e ferramentas que ajudam no desenvolvimento de aplicações voltadas à área da robótica, incluindo abstração de *hardware*, *drivers*, bibliotecas específicas, visualizadores, transmissão de mensagens, gerenciamento de pacotes e muitas outras funcionalidades.

O ROS é organizado utilizando o conceito de pacotes. Um pacote pode conter nodos, bibliotecas, conjuntos de dados, arquivos de configurações e *plugins* externos. O objetivo desses pacotes é fornecer de forma simplificada funcionalidades úteis ao desenvolvedor, de modo que o programa possa ser facilmente reutilizado.

No ROS pode-se criar uma rede em que os processos podem estar interligados. Qualquer nodo do sistema pode acessar a rede, interagir com outros nodos por meio de troca de mensagens. Essas interações podem ser realizadas por tópicos ou por serviço. A seguir, para melhor entendimento do funcionamento do ROS, descrevem-se os conceitos de *nó*, *mensagem*, *tópico* e *serviço*.

1. **Nó (*Nodo*):** pode ser descrito como um processo, podendo realizar um cálculo, enviar e/ou receber informações de outros nodos e utilizar e/ou prestar serviços (KEN, 2013a). A Figura 17 mostra um exemplo de dois nodos, sendo um nodo responsável pelo envio da mensagem (*talker*) e outro pela recepção (*listener*). Essa interconexão entre nodos é feita por meio de um tópico, publicando uma mensagem para o nodo de destino.

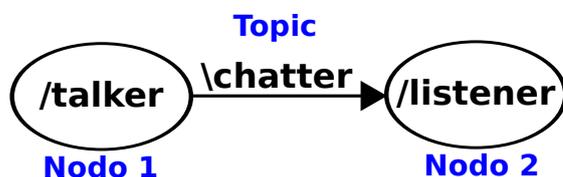


Figura 17: Exemplo de um nodo no ROS.

2. **Mensagem:** as mensagens carregam a informação de um nodo para o outro. As mensagens são simplesmente uma estrutura de dados equivalente a uma estrutura de C++ (KEN, 2013b).
3. **Tópico (*topic*):** um tópico é a publicação de mensagens realizada entre os nodos (KEN, 2013c). A Figura 18 mostra um exemplo de aplicação.

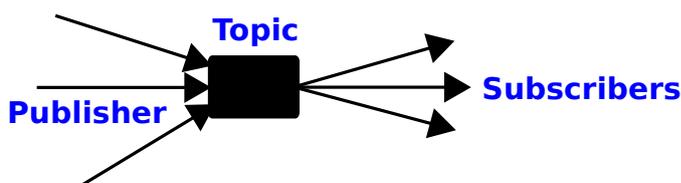


Figura 18: Exemplo de um tópico no ROS.

4. **Serviço (*service*):** serviços são as formas pelas quais uma mensagem envia um pedido (*request*) a partir de um nodo para outro, sendo retornado uma resposta (*response*) com outra mensagem. A Figura 19 ilustra um exemplo da aplicação (KEN, 2013d).

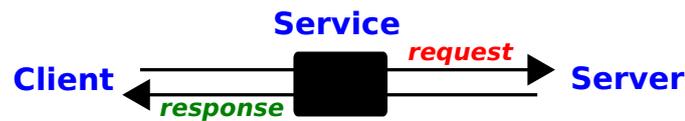


Figura 19: Exemplo de um serviço no ROS.

3.1.1 Unified Robot Description Format (URDF)

Para descrever um robô no ROS, uma das formas possíveis é utilizar o pacote denominado de URDF (IOAN, 2009). Este pacote possibilita descrever a cinemática, dinâmica, modelos de colisões, materiais dos componentes, coloração e texturas do robô. Atualmente, existe uma restrição importante¹ a ser destacada ao implementar um robô utilizando-se URDF: o desenvolvimento deve seguir uma topologia do tipo árvore, o que impossibilita a descrição de robôs paralelos utilizando o URDF.

Para exemplificar o desenvolvimento de um robô em URDF, a Figura 20 representa uma cadeia cinemática genérica que utiliza a topologia do tipo árvore, constituída por um conjunto de quatro elos, nomeados de `link1` a `link4` conectados por 3 juntas, nomeados de `joint1` a `joint3`. Esta cadeia cinemática pode ser descrita em URDF, como descrito na listagem C.1 disponibilizado no Apêndice C.

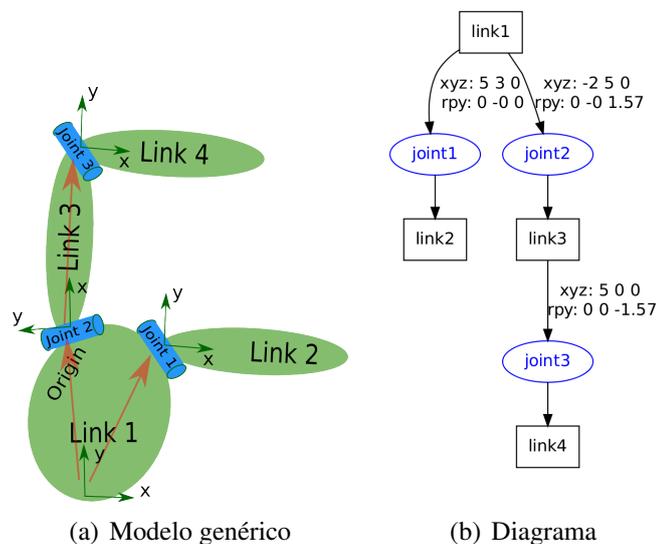


Figura 20: Cadeia cinemática com topologia do tipo árvore (IOAN, 2013a).

Os principais componentes dos parâmetros de um arquivo URDF, descritas na Listagem C.1 são apresentados a seguir.

¹Atualmente ocorre a falta de informação sobre essas restrições no site do ROS.

1. A *tag robot*, inicializa um bloco onde é descrito o nome do robô, características dos elos e tipo de juntas. Esta descrição é disponibilizada em todos os subsistemas do ROS que estão ativos no momento da simulação;
2. A *tag link* (vide a Listagem 3.1), descreve um corpo rígido com uma inércia atrelada ao seu sistema e algumas características visuais, como pode-se visualizar na Figura 21. Existe também a possibilidade de configurar o tipo de geometria (caixa, cilindro, esfera ou arquivos externos tridimensionais, nas extensões, STL (MARSHALL, 1989) ou DAE), material e propriedades de colisão de um elo. A *tag inertial* possibilita especificar os parâmetros de inércia do elo: centro de massa (*xyz*), orientação dos eixos principais (*rpy*) e a massa (*mass*) do elo.

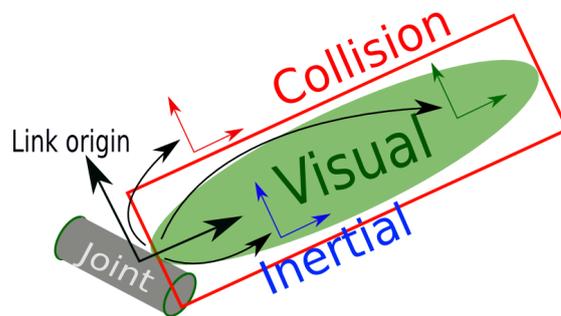


Figura 21: Estrutura de um elo no URDF (IOAN, 2013b).

Listagem 3.1: *tag link* em URDF.

```

1 <link name="link1">
2   <inertial>
3     <origin xyz="0 0 0" rpy="0 0 0"/>
4     <mass value="1.0" />
5     <inertia ixx="1.0" ixy="0.0" ixz="0.0"
6       iyy="1.0" iyz="0.0" izz="1.0"/>
7   </inertial>
8 </link>

```

3. A *tag joint* (Listagem 3.2), descreve a cinemática e a dinâmica para uma junta, disponibiliza parâmetros de limites de segurança para a estrutura e inclui especificações para o elo pai (*parent*) e para o elo filho (*child*). A *tag joint* também possibilita nomear a junta, selecionar o seu tipo (*revolute continuous*, *prismatic*, *fixed*, *floating* e *planar*), possibilita ao usuário especificar a posição (*xyz*) e orientação (*rpy*) em relação ao elo pai e o eixo de movimentação (*axis*) para cada junta. A Figura 22 ilustra um dos principais pontos para o funcionamento de uma junta no URDF.

Nota-se que o sistema de coordenadas da junta é fixado no ponto de intersecção entre as juntas, e é dessa forma que deve ser exportado o sistema de coordenadas de uma junta ao se desenvolver um projeto utilizando-se uma ferramenta de CAD ².

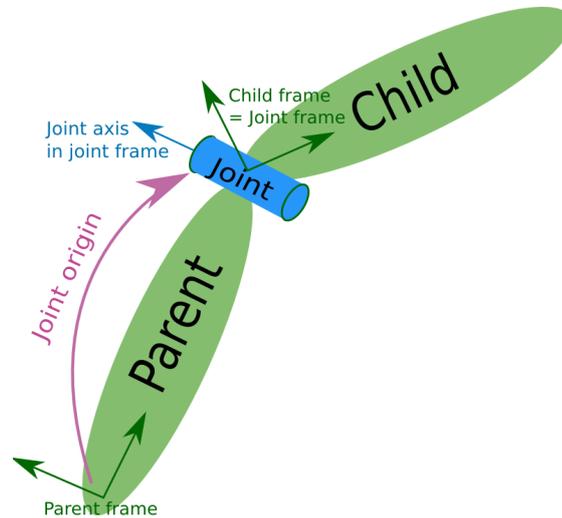


Figura 22: Estrutura de uma junta no URDF (IOAN, 2013c).

Listagem 3.2: tag joint em URDF.

```

1 <joint name="joint1" type="continuous">
2   <parent link="link1"/>
3   <child link="link2"/>
4   <origin xyz="5 3 0" rpy="0 0 0" />
5   <axis xyz="-0.9 0.15 0" />
6 </joint>

```

4. A tag `transmission` (Listagem 3.3), descreve o tipo de transmissão e atuador que são aplicadas nas juntas, tipo de interface de *hardware* e também permite realizar a escolha da redução mecânica mais adequada para o motor.

Listagem 3.3: tag transmission em URDF.

```

1 <transmission name="trans1">
2   <type>transmission_interface/SimpleTransmission</type>
3   <joint name="joint1"/>
4   <actuator name="motor1">
5     <hardwareInterface>EffortJointInterface</hardwareInterface>
6     <mechanicalReduction>1</mechanicalReduction>

```

²Normalmente, sistemas CAD possuem uma notação diferente das notações de coordenadas relacionadas ao desenvolvimento de robôs, o que pode ocasionar erros na simulação.

```

7 </actuator>
8 </transmission>

```

Para realizar a simulação de um robô descrito em URDF, pode-se utilizar o programa Gazebo, que será descrito a seguir.

3.1.2 Gazebo

Gazebo é um simulador de robôs. O programa gera tanto sinais de sensores e interações físicas entre objetos, quanto simulações envolvendo física de corpos rígidos. A Figura 23 ilustra o robô bípede Orbit simulado no Gazebo.

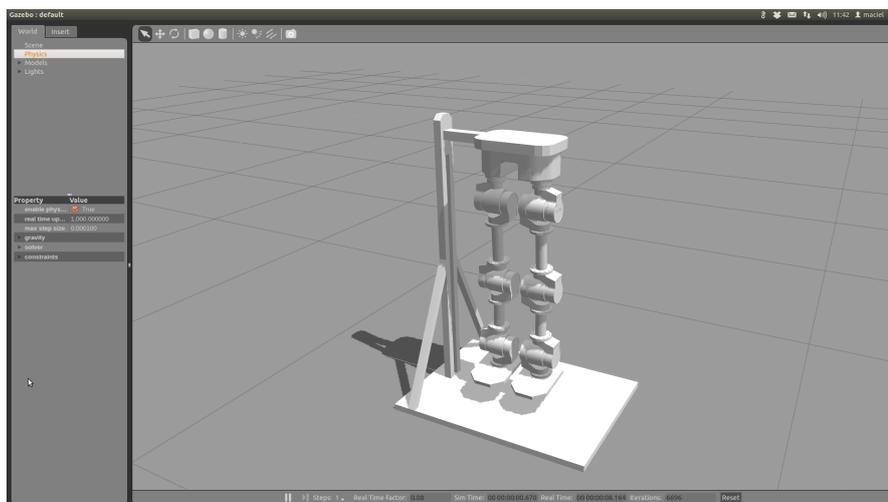


Figura 23: Robô Bípede Orbit.

O Gazebo foi originalmente concebido para ajudar no processo de desenvolvimento de algoritmos para plataformas robóticas. O uso deste tipo de programa ajuda a diminuir o tempo de desenvolvimento de um protótipo e a evitar alguns problemas comuns, como, por exemplo, alguns comportamentos inesperados ou perigosos do robô durante os testes de algoritmos.

3.1.3 Kinematics and Dynamics Library (KDL)

Para realizar o cálculo da cinemática ou dinâmica de um robô no ROS, pode-se utilizar o pacote denominado de KDL (BRUYNINCKX, 2001). Este pacote disponibiliza o suporte ao uso de geometrias primitivas e também suporta cadeias cinemáticas de vários tipos, como, por exemplo, robôs seriais, humanóides, robôs paralelos, robôs móveis e outros diversos modelos.

Para realizar cálculos de cinemática ou dinâmica de um robô utilizando o KDL, muitas vezes é necessário ler os parâmetros de um arquivo URDF. Para isso, utiliza-se

o `kdl_parser`. A Figura 24 ilustra a criação de uma cadeia cinemática do tipo árvore sendo lida de um arquivo URDF. Nota-se que as *tags* do URDF são convertidas para parâmetros que podem ser interpretados e calculados pelo KDL (DAVE, 2013b).

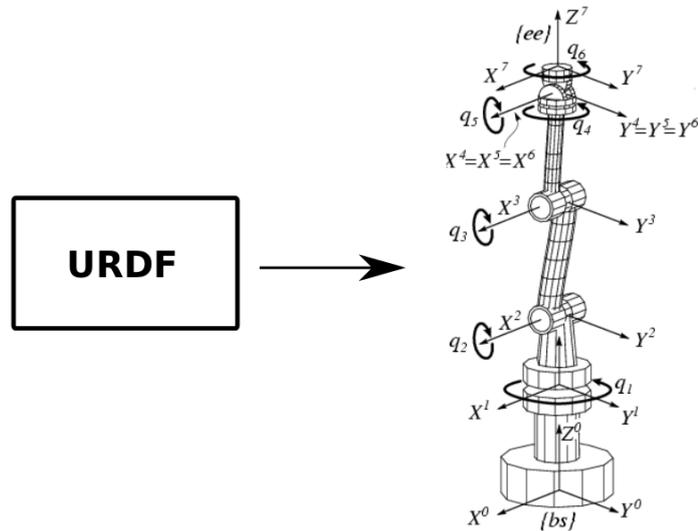


Figura 24: Parser de URDF para KDL.

Ao realizar este *parser*, deve-se tomar cuidado em relação à matriz de inércia do elo, pois o KDL especifica a matriz de inércia no centro da junta e o URDF especifica a matriz de inércia no centro do elo. É possível encontrar esses detalhes descritos no código fonte do `kdl_parser.cpp`.

No pacote KDL existem algumas classes importantes para serem utilizadas em um desenvolvimento de robôs que utilizam um sistema de controle por torque calculado:

1. `KDL::Chain`: Encapsula uma estrutura de interligações baseadas na cinemática do tipo serial;
2. `KDL::Tree`: Encapsula uma estrutura de interligações baseadas na cinemática do tipo árvore;
3. `KDL::ChainIdSolver_RNE`: É utilizada para implementar o algoritmo recursivo de Newton-Euler (FEATHERSTONE, 1987).

3.1.4 Sistema de Controle

A implementação de controladores que utilizam o ROS pode ser realizada utilizando o pacote `ros_control`. A Figura 25 mostra uma visão geral do ROS juntamente com

o Gazebo, onde está a disposição do usuário uma camada de simulação (*Simulation*) que tem *plugins* que implementam uma interface entre o Gazebo e o ROS. Também existe a possibilidade de implementar os testes diretamente no robô utilizando o conceito do bloco *Hardware* e, por fim a camada *Hardware Resource Interface Layer* faz interface entre o controlador (*Controller Manager*) e a estrutura de simulação ou a estrutura de *Hardware* do robô.

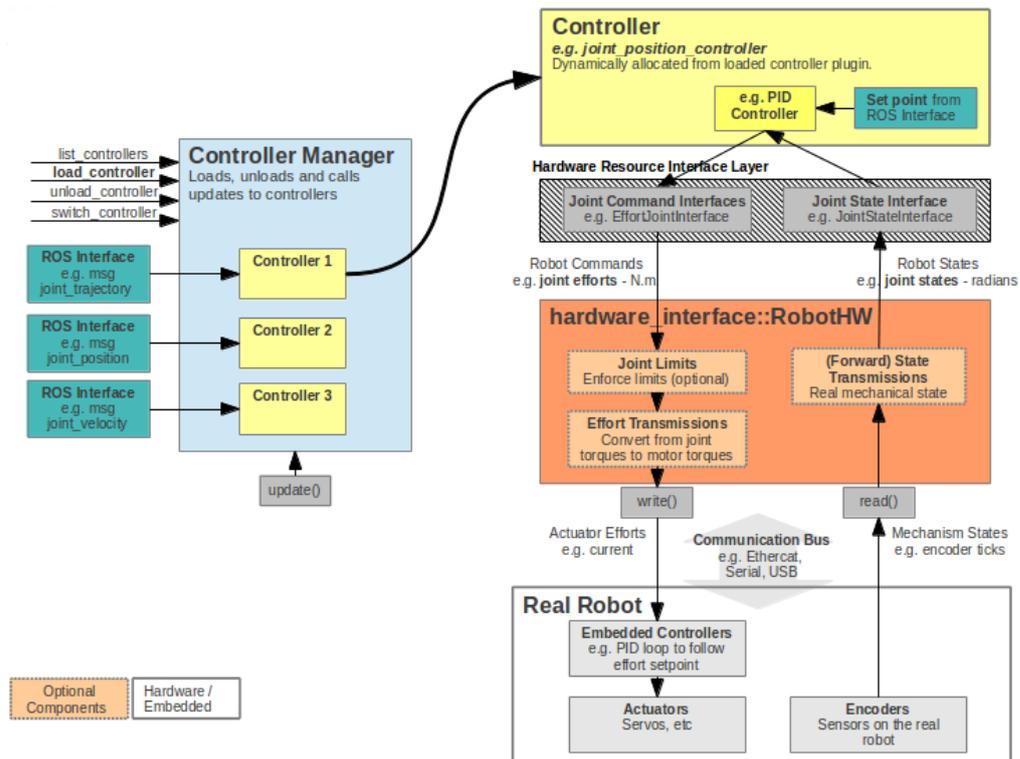


Figura 25: Diagrama de fluxo de dados de um controlador no ROS (DAVE, 2013a).

Como pré-requisito para o uso de controladores existentes no ROS ou para realizar a implementação de novos controladores no ROS, é necessário instalar os seguintes pacotes `ros_control` e `ros_controllers`. Para utilizar o pacote `ros_control` no projeto de um robô, é necessário adicionar a *tag* `transmission` no arquivo URDF. Um exemplo de como adicionar essa *tag* no URDF pode ser visto na Listagem 3.3. Além da *tag* `transmission`, também deve-se adicionar o *plugin* `gazebo_ros_control` no arquivo URDF do projeto, que serve para implementar o tipo de transmissão adicionada em cada junta e carregar o `hardwareInterface` apropriado para gerenciar o controlador. Como padrão, este *plugin* traz algumas interfaces desenvolvidas pela

própria comunidade do ROS. Porém, o usuário pode desenvolver sua própria interface, tendo em vista a necessidade de solucionar algum problema específico. Para adicionar o *plugin* `gazebo_ros_control` no URDF, deve-se adicionar a *tag* `gazebo`, conforme código exemplo da Listagem 3.4.

Listagem 3.4: Código do *plugin* `gazebo_ros_control` para URDF.

```

1 <gazebo>
2 <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
3   <robotNamespace>/MYROBOT</robotNamespace>
4 </plugin>
5 </gazebo>

```

O *plugin* também fornece uma interface baseada no `pluginlib`, que faz uma interface entre o Gazebo e o `ros_control`, servindo para simulação de mecanismos mais complexos, como, por exemplo, molas, dobradiças, ligamentos, etc.

O *RobotHWSim* fornece acesso em nível de *API*, que serve para ler e comandar propriedades conjuntas durante a simulação do Gazebo. A respectiva classe `RobotHWSim` deve ser especificada no modelo do URDF, e é carregada quando o modelo do robô é inicializado na simulação. A Listagem 3.5 descreve como realizar essa codificação interna ao arquivo URDF.

Listagem 3.5: Código para customização do *plugin* `gazebo_ros_control`.

```

1 <gazebo>
2 <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
3   <robotNamespace>/MYROBOT</robotNamespace>
4   <robotSimType>gazebo_ros_control/DefaultRobotHWSim</robotSimType>
5 </plugin>
6 </gazebo>

```

Normalmente, ao se utilizar controladores já implementados no ROS, o usuário se depara com controladores que envolvem o uso do clássico controle PID. Para parametrizar os ganhos do controlador deve-se criar um arquivo `yaml` que será carregado em um arquivo `launch`. A Listagem 3.6 descreve um exemplo do uso de configuração de parâmetros do controlador, onde utilizar o `joint_state_controller` como sensor e o `joint1_position_controller` como um controlador de posição.

Listagem 3.6: Arquivo de configuração dos parâmetros para o controlador.

```

1 MYROBOT:
2   # Publish all joint states
3   joint_state_controller:
4     type: joint_state_controller/JointStateController
5     publish_rate: 50
6
7   # Position Controllers
8   joint1_position_controller:

```

```

9   type: effort_controllers/JointPositionController
10  joint: joint1
11  pid: {p: 100.0, i: 0.01, d: 10.0}

```

Para desenvolver um arquivo de lançamento para um controlador utilizando o pacote `ros_control` juntamente com o arquivo yml de configuração para os ganhos, pode-se seguir os passos conforme mostrado no código exemplo da Listagem 3.7.

Listagem 3.7: Arquivo de configuração dos parâmetros para o controlador.

```

1 <launch>
2 <!-- Load joint controller configurations
3 from YAML file to parameter server -->
4 <rosparam file="$(find MYROBOT_control)/config/MYROBOT_control.yaml"
5 command="load"/>
6
7 <!-- load the controllers -->
8 <node name="controller_spawner" pkg="controller_manager"
9 type="spawner" respawn="false"
10 output="screen" ns="/MYROBOT"
11 args="joint1_position_controller joint_state_controller"/>
12
13 <!-- convert joint states to
14 TF transforms for rviz, etc -->
15 <node name="robot_state_publisher" pkg="robot_state_publisher"
16 type="robot_state_publisher" respawn="false" output="screen">
17 <remap from="/joint_states" to="/MYROBOT/joint_states" />
18 </node>
19 </launch>

```

3.1.5 Gerador de Trajetórias

A geração de trajetória utilizada no ROS pode ser tratada como sendo uma troca de mensagens entre os nodos. Em alguns casos essas mensagens podem ser trocadas utilizando o conceito de serviço no ROS. Pode-se utilizar para a implementação de um serviço no ROS, o pacote `actionlib`, que fornece ferramentas para o desenvolvimento de um servidor (*ActionServer*) de troca de mensagens, podendo ser inicializado, cancelado ou consultado a qualquer momento durante a sua execução.

O pacote `actionlib` também fornece uma interface para o desenvolvimento de um cliente (*ActionClient*) que irá enviar a solicitação ao servidor. O *ActionServer* e o *ActionClient* se comunicam por meio de um protocolo denominado de *ROS Action Protocol*, que é baseado na troca de mensagens entre os nodos. O cliente e o servidor realizam entre eles uma solicitação de objetivo ³ (*goal*) para que se realize uma ação desejada. A Figura 26 ilustra essa troca de mensagens de uma forma simplificada.

³Esta solicitação do objetivo é realizada no cliente para o servidor.

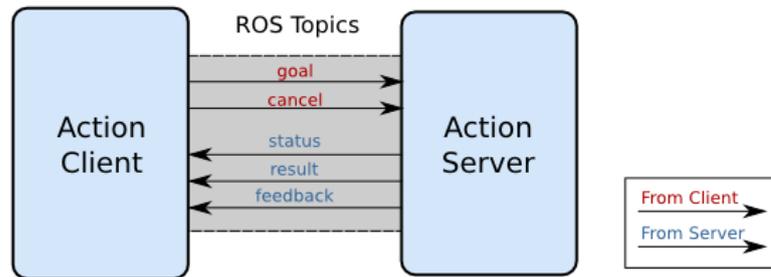


Figura 26: Interface de ação entre o cliente e o servidor.

1. *Goal*: É utilizado para enviar novos pontos de via do cliente para servidor, que posteriormente passa um vetor de referência de posição e de velocidade para o controlador;
2. *Cancel*: É utilizado para enviar uma solicitação de cancelamento do serviço que o servidor está realizando em um determinado instante de tempo;
3. *Status*: É utilizado para notificar o cliente sobre o estado atual de cada *Goal* no sistema;
4. *Feedback*: É utilizado para enviar ao cliente uma informação auxiliar periodicamente em resposta a um *Goal*;
5. *Result*: É utilizado para enviar uma informação auxiliar de uma só vez após a conclusão de um *Goal*.

Para realizar a geração de trajetórias utilizando diferentes interpolação, pode-se utilizar o pacote KDL, que disponibiliza a classe `KDL::VelocityProfile_Spline`, que implementa três diferentes tipos de interpolações: linear, cúbica e de quinto grau. Essa implementação deve ser realizada no servidor e o cliente fica encarregado de enviar os objetivo para cada junta.

4 METODOLOGIA

Para realizar o desenvolvimento e a simulação de um modelo simplificado dos membros inferiores de um robô bípede utilizando o ROS, se faz necessária a realização de um estudo sobre quais modelos e tipos de estruturas mecânicas que estão sendo utilizados atualmente nas pesquisas na área dos robôs bípedes. Também é importante realizar o estudo sobre modelagem das estruturas mecânicas utilizando uma ferramenta CAD e sobre como projetar sistemas robóticos utilizando o ROS. Algumas dessas etapas foram apresentadas em capítulos anteriores deste trabalho.

4.1 Desenvolvimento do Robô bípede Orbit

Para este trabalho, propõe-se uma metodologia de desenvolvimento de um robô bípede que permite realizar movimentos das pernas no plano sagital. A Figura 27 apresenta os principais componentes envolvidos no desenvolvimento deste trabalho.

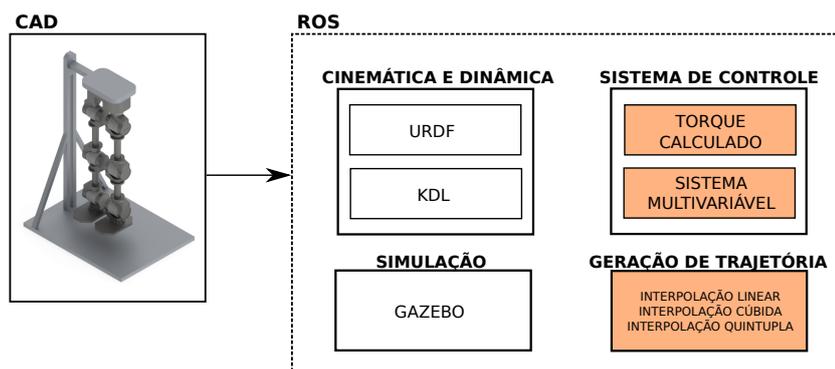


Figura 27: Metodologia de desenvolvimento do robô bípede Orbit.

4.1.1 Estrutura Mecânica do Robô Bípede Orbit

A estrutura mecânica é baseada no trabalho de MACIEL (2011), onde iniciou-se o desenvolvimento de um robô bípede de pequeno porte. Este protótipo não possuía um sistema de controle e sua aplicação era apenas voltada para realizar um simples movimento do pé do robô em malha aberta. A Figura 28 apresenta a evolução do robô Orbit até o presente momento.

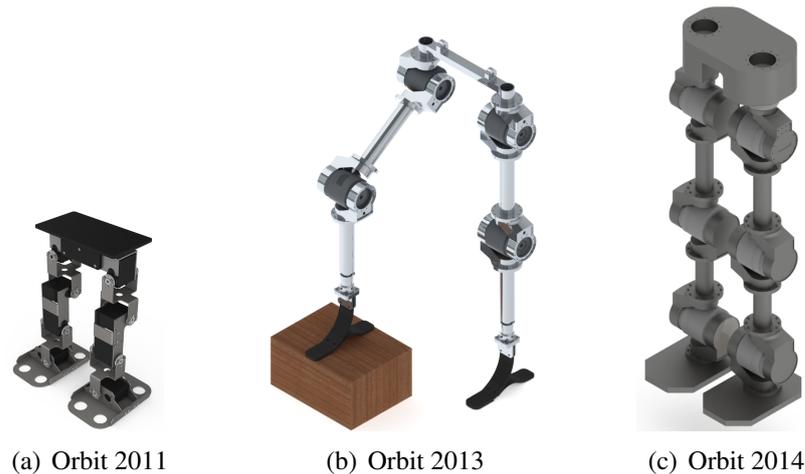


Figura 28: Evolução do robô Orbit (MACIEL, 2011).

O projeto da Figura 28 (c) procura seguir um modelo simplificado dos membros inferiores do corpo humano, dessa forma foi desenvolvido uma estrutura contendo uma cintura, dois fêmures, duas tíbias e dois pés.

No trabalho de MORECKI; EKIEL; FIDELUS (1984), o autor descreve que um ser humano possui no total trinta GDLs nos membros inferiores de cada perna. No trabalho de SIAS; ZHENG (1990), o autor realizou testes com diferentes números de GDLs e obteve a conclusão de que um robô bípede pode ser desenvolvido com apenas quatro GDLs em cada perna, mas, para que os movimentos tenham uma boa aproximação do corpo humano, o robô deve possuir oito GDLs em cada perna. Já, no trabalho de BEZERRA (2002), o autor desenvolve uma simulação de um robô bípede utilizando apenas três GDLs em cada perna para a locomoção. Como o foco desse trabalho não está voltado na locomoção do robô bípede, a estrutura inicial apresenta apenas três DoFs em cada perna, conforme mostra a Figura 29 (b).

Para o projeto da estrutura mecânica do robô bípede Orbit utilizou-se o programa SolidWorks. Um dos motivos do uso deste programa para o projeto, é a de possuir um *Add-Ins* denominado de SW2URDF (STEPHEN, 2012), que possibilita exportar a estrutura mecânica tridimensional para URDF, simplificando, desta forma, a criação do modelo interno ao ROS.

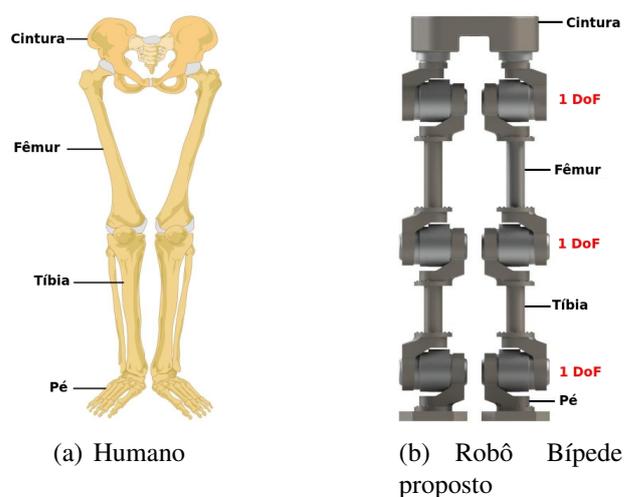


Figura 29: Estruturas simplificadas dos membros inferiores.

Ao utilizar o SW2URDF, deve-se ficar atento ao realizar a exportação do modelo, pois se faz necessário especificar para cada junta o ponto de origem junto ao eixo de rotação. A Figura 30 ilustra o ponto onde deve estar localizada a coordenada do eixo de rotação e a origem da junta no desenho tridimensional.

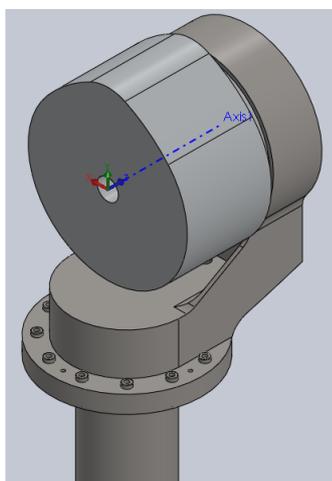


Figura 30: Ponto de origem da junta do robô Orbit.

4.1.2 Modelo Cinemático do Robô Bípede Orbit

O modelo cinemático do Orbit possui sete corpos rígidos, sendo unidos entre si por meio de juntas de rotação com um GDL, representados na Figura 31.

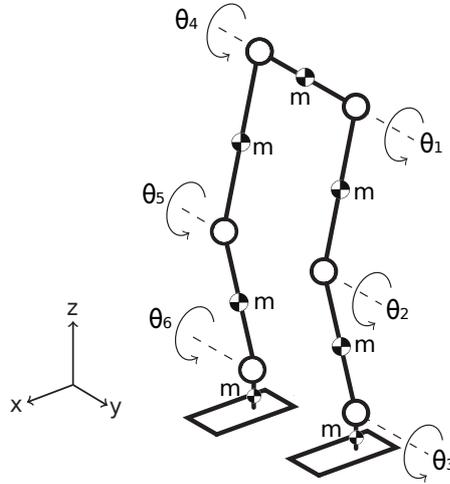


Figura 31: Esquema do Modelo cinemático do robô bípede Orbit.

O robô Orbit é modelado da mesma forma que um robô serial. Neste caso, possui duas cadeias cinemáticas desacopladas, onde se aplicam sucessivas transformações em seus sistemas de coordenadas adjacentes. Considera-se que o sistema de coordenadas do referencial inercial está localizado na cintura do robô e que as origens dos demais sistemas de coordenadas estão localizadas nas juntas de rotação.

Observando-se a Figura 31, nota-se que todas as rotações são em torno do semi eixo Z positivo, pois todos os eixos Z são paralelos. Portanto, pode-se obter todas as matrizes de transformação de coordenadas como segue:

$$R_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad i = 1, \dots, 6 \quad (25)$$

4.1.3 Sistema de Controle do Robô Bípede Orbit

O sistema de controle desenvolvido para o robô bípede Orbit utiliza uma abordagem multivariável, juntamente com a técnica de controle por torque calculado. A escolha do controlador por torque calculado deve-se ao fato de que a Equação (24) pode levar o erro para zero, o que, para o bípede significa que ao comandar um sinal para uma junta específica, o torque converge rapidamente para o valor de referência e as não-linearidades são compensadas pela estrutura de controle.

Para determinar os ganhos do controlador, escolhe-se as matrizes de ganho K_p e K_d

fixando a frequência natural ω_n e o índice de amortecimento ξ_n : $K_p = \text{diag}(\omega_n^2)$ e $K_d = \text{diag}(2\xi_n\omega_n)$. Escolhendo-se empiricamente os valores para $\xi_n = 1,0$ e para $\omega_n = 20\pi$ rad/s, para que o sistema tenha um comportamento hiper-amortecido (BAZANELLA; SILVA, 2005):

$$K_p = \begin{bmatrix} 15791,4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15791,4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15791,4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15791,4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15791,4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15791,4 \end{bmatrix} \quad (26)$$

$$K_d = \begin{bmatrix} 251,327 & 0 & 0 & 0 & 0 & 0 \\ 0 & 251,327 & 0 & 0 & 0 & 0 \\ 0 & 0 & 251,327 & 0 & 0 & 0 \\ 0 & 0 & 0 & 251,327 & 0 & 0 \\ 0 & 0 & 0 & 0 & 251,327 & 0 \\ 0 & 0 & 0 & 0 & 0 & 251,327 \end{bmatrix} \quad (27)$$

A Figura 32, mostra o diagrama de nodos e tópicos apresentado pelo programa `rqt` do ROS. Esses nodos são utilizado para realizar a simulação do sistema de controle do robô Orbit. O nodo `/robot_state_publisher` publica o estado do robô (posição em radianos para cada elo do robô) no tópico `tf`. O tópico `/orbit/joint_states`, é publicado pelo nodo `gazebo`, informando a posição e velocidade de cada junta. O nodo `/computed_torque_controller` e o `JointMIMOEffortController` foram implementados especialmente para esse trabalho ¹, permitindo a implementação de um controlador por torque calculado em uma estrutura de controle multivariável dentro do ROS.

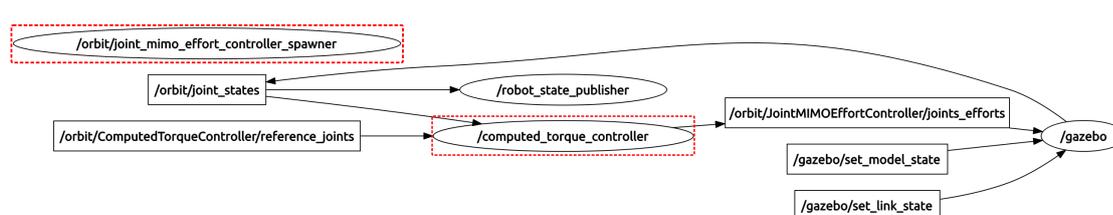


Figura 32: Diagrama contendo os nodos do projeto do robô bípode Orbit no ROS.

A Figura 33 mostra o fluxo dos sinais de controle no ROS. Pode-se observar que o componente em cinza denominado de `JointMIMOEffortController` e o componente em cinza denominado de `ComputerTorqueController` fazem parte do

¹Retângulos em vermelho na Figura 32.

pacote desenvolvido especialmente para esse projeto. Os demais componentes são implementados e disponibilizados pela comunidade do ROS. A seguir descreve-se resumidamente os componentes do diagrama de fluxo de controle e suas interações.

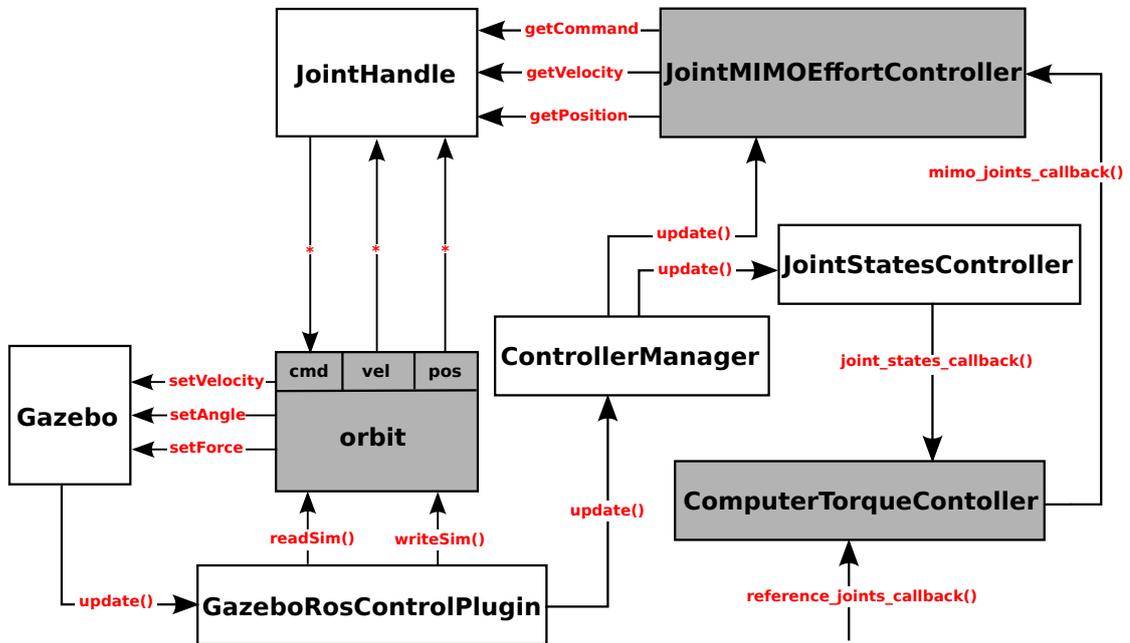


Figura 33: Diagrama de fluxo de controle no ROS.

- Gazebo: É o componente responsável pela visualização da simulação e é o responsável pela base de tempo do sistema ².
- GazeboRosControlPlugin: Disponibiliza a interface entre o Gazebo e o ROS;
- JointHandle: É utilizado para ler os sensores e enviar as referências para o componente Orbit via ponteiro;
- Orbit: É um *plugin* que é inserido no URDF, disponibilizando para a simulação os comandos de `setVelocity`, `setAngle` e `setForce` vindos do componente JointHandle;
- ControllerManager: De acordo com a Figura 25, o ControllerManager atualiza os componentes JointStateController (disponibiliza os sinais dos sensores de posição, velocidade e torque que são enviados por meio de tópico

²Para as simulações apresentadas neste trabalho utilizou-se 1 milissegundo.

pelo Gazebo) e `JointMIMOEffortController`. Esse componente também possui a opção de listar, conectar, desconectar e realizar a troca entre controladores que estão disponíveis no sistema.

- `JointMIMOEffortController`: É um *bypass* que repassa os vetores de torques calculado no componente `ComputerTorqueController` para o Gazebo. Esse componente possui as características de um controlador multivariável, que envia os valores dos torques simultaneamente para todas as juntas do robô Orbit;
- `ComputerTorqueController`: Este componente recebe um vetor de posições e velocidades desejadas por meio do `reference_joint_callback()` que são enviados pelo gerador de trajetórias via tópico `reference_joints`. Com esses valores de referência, o sinal de controle é calculado e com o resultado tem-se o vetor de torques generalizados que são enviado para o componente `JointMIMOEffortController` via `computer_torque_callback()`, obtendo, dessa forma, uma ação na junta desejada.

As linhas pontilhadas significam que as variáveis são acessadas diretamente no código do simulador pelos ponteiros, ao invés de serem acessadas através de chamada de funções, como representado pelas linhas cheias.

4.1.4 Gerador de Trajetórias

Para realizar os testes no sistema de controle utilizou-se uma interpolação linear, cúbica e quártica. O desenvolvimento deste gerador de trajetórias segue os conceitos visto na Seção 3. A Figura 34 ilustra o fluxo de mensagens envolvidas no processo de geração de trajetória do robô bípede Orbit no ROS, onde o nó `orbit_trajectory_client` é substituído de acordo com o tipo de interpolação.

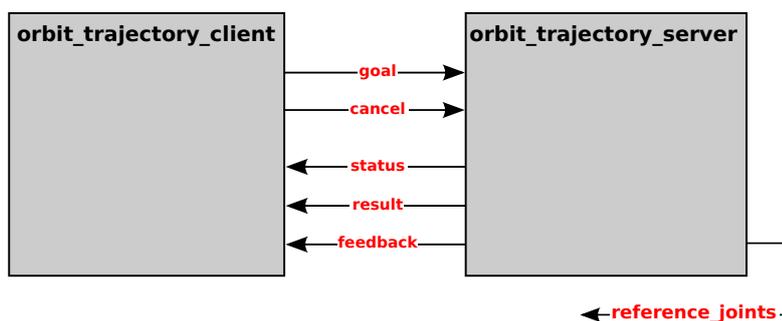


Figura 34: Diagrama de fluxo de mensagens do gerador de trajetória do robô bípede Orbit no ROS.

A Figura 35, ilustra o diagrama de nodos do gerador realizada pela interpolação linear do robô Orbit apresentado pelo programa `rqt` do ROS. Esses nodos são utilizados para realizar a troca de mensagens entre o servidor e o cliente, onde o nodo `orbit_trajectory_client` envia os pontos de via de uma trajetória para o nodo `orbit_trajectory_server`, que processa esses dados e envia um vetor de posições e velocidades desejadas para o `ComputerTorqueController` por meio da mensagem `JointMIMOEfforts`. Para obter o vetor de aceleração utilizado na interpolação de quinto grau, realiza-se no nodo `orbit_trajectory_server` a derivada da velocidade que é enviado pelo nodo `orbit_trajectory_client`.

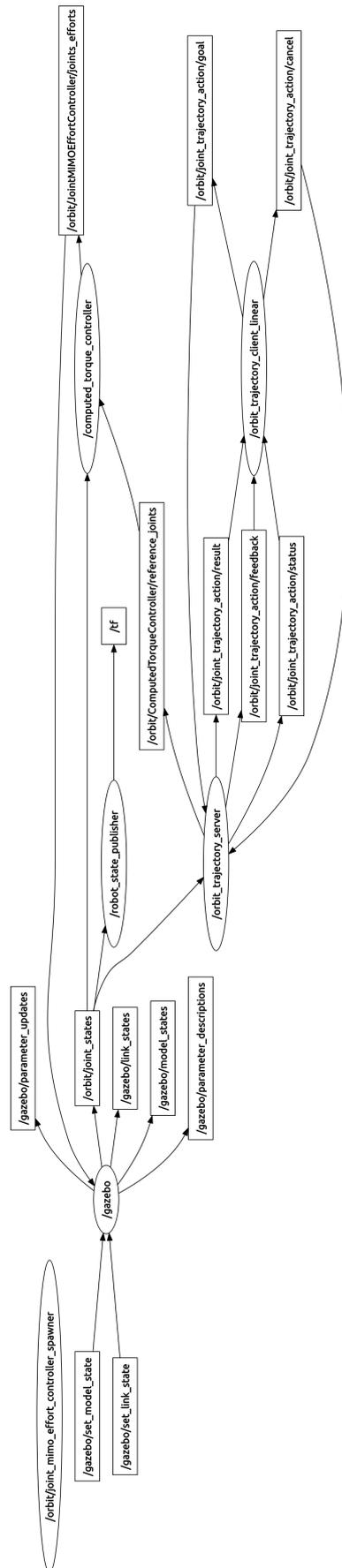


Figura 35: Diagrama de nodos do gerador de trajetória do robô bípede Orbit no ROS, utilizando interpolação linear.

5 SIMULAÇÕES E RESULTADOS

Neste capítulo apresentam-se as simulações e resultados da aplicação do controlador por torque calculado utilizando-se o sistema de controle multivariado proposto. Apresentam-se as simulações realizadas na perna esquerda ¹ do robô bípede Orbit. São apresentadas duas seções: a) Simulação sem saturação no sinal de controle e b) Simulação com saturação no sinal de controle. Cabe salientar que as constantes físicas são simuladas no ambiente do Gazebo e não tem-se o controle de tais variáveis.

5.1 Simulação sem Saturação no Sinal de Controle

5.1.1 Trajetória ao Degrau

A seguir, seguem-se os resultados da aplicação de uma trajetória ao degrau de referência de 0,1 radianos de amplitude na junta 1, disposta na cintura do robô Orbit. Pode-se visualizar na Figura 31, que o ângulo θ_1 é atribuído ao sistema de coordenadas da junta 1 da cintura, visto que esta é a junta sujeita ao maior esforço, devido ao acoplamento dos outros elos que compõem a perna esquerda do robô.

A Figura 36 ilustra o sinal de referência e de posição da junta 1. Pode-se perceber que o sinal de posição segue o sinal de referência, com erro ² e mantém-se próximo de zero. Este comportamento deve-se a estrutura de controle por torque calculado, o que comprova à eficiência da Equação (24).

A Figura 37 ilustra que o sinal de torque na junta 1 excede o valor do torque nominal de 20,7 Nm ³, com um pico máximo de torque de 1058 Nm.

Percebe-se na Figura 38 que as juntas 2 e 3 também apresentam torques. Estes torques são oriundos das forças de compensação necessárias para que as juntas do joelho e do pé, respectivamente θ_2 e θ_3 , mantenham-se nos seus valores de referência, permanecendo em sua posição inicial (zero). A Figura 36 mostra o resultado do sinal de posição

¹Como o robô é simétrico, os resultados são duais.

²Vide Tabela 1.

³Vide Tabela 6, Apêndice B.

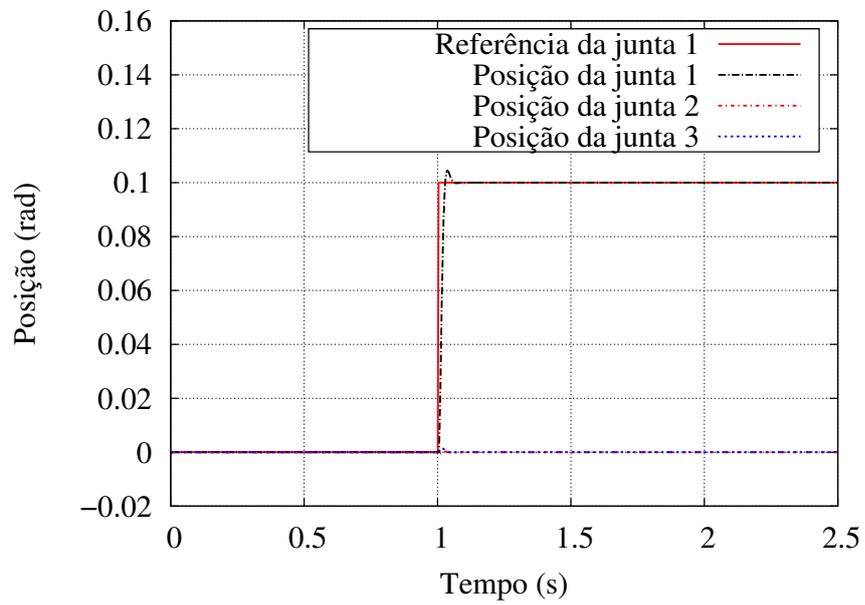


Figura 36: Sinal de referência e de posição da junta 1,2 e 3.

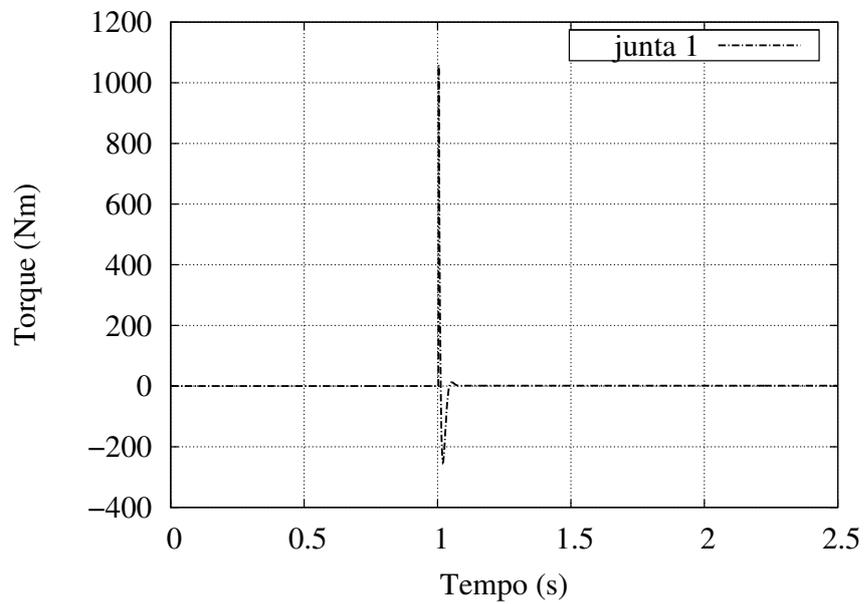


Figura 37: Torque na junta 1.

em relação ao sinal de referência das juntas 1,2 e 3.

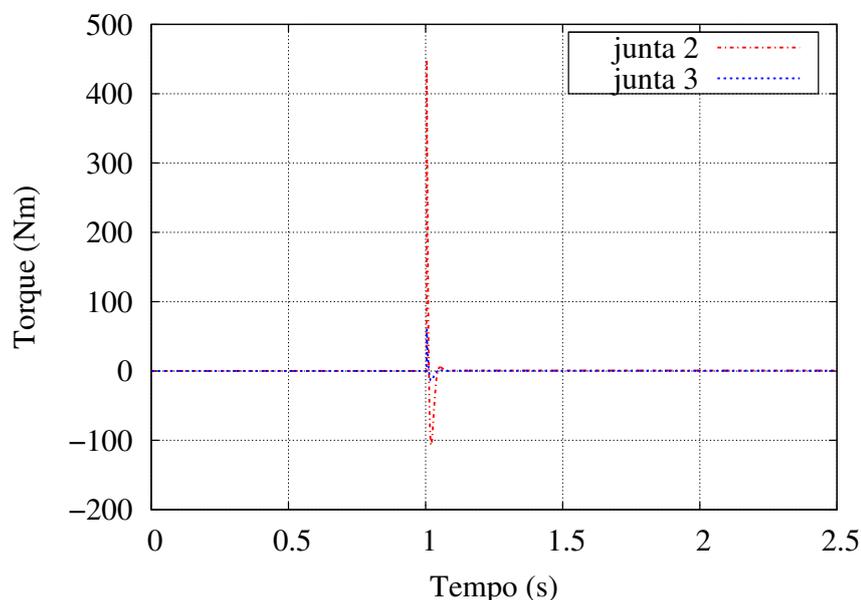


Figura 38: Torques nas juntas 2 e 3.

A seguir apresentam-se os três outros tipos de entrada de referência e seus respectivos resultados.

5.1.2 Interpolação Linear

Diferentemente da trajetória ao degrau a interpolação linear apresenta uma trajetória mais suave, como pode ser visto na Figura 39.

Pode-se notar na Figura 39 que a resposta segue o sinal de referência com erro ⁴ aproximadamente nulo. Esse comportamento deve-se ao fato das matrizes de ganho K_p e K_d não terem sido sintonizadas. Salienta-se também que o torque diminuiu sua magnitude se comparado a trajetória ao degrau, reduzindo o torque a valores menores.

Na Figura 40 pode-se verificar que, devido à suavização do sinal de referência, onde o torque na junta 1 é reduzido a picos de 4,24 Nm nos dois pontos de inflexão no instante de tempo de 1 s e 2 s. Percebe-se que os torques das juntas 2 e 3 apresentam uma redução em sua magnitude quando comparado ao apresentado na Figura 38. Parte desta redução deve-se à existência de saturação do sinal de torque e a outra parte deve-se a transição mais suave da interpolação linear comparada a trajetória ao degrau. A Figura 41 apresenta esses resultados.

⁴Vide Tabela 2.

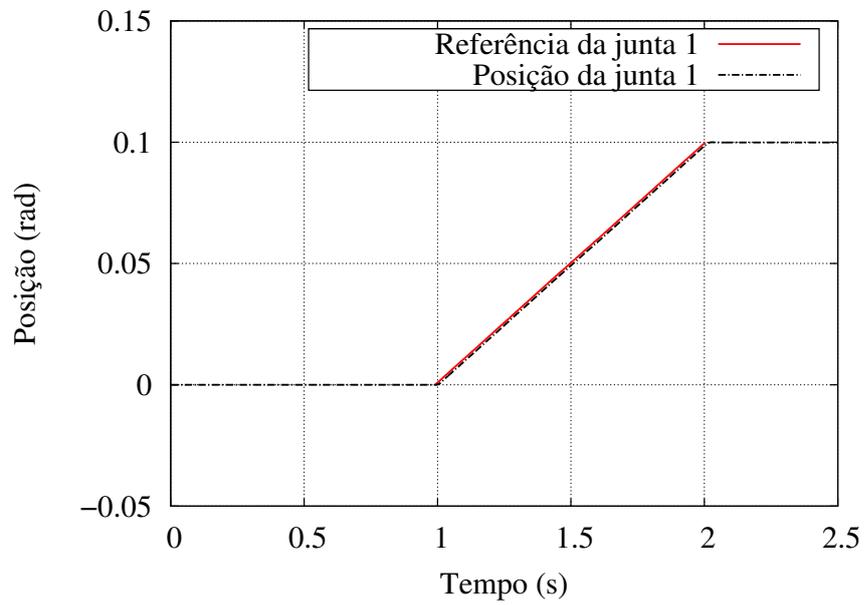


Figura 39: Sinal de referência e de posição da junta 1.

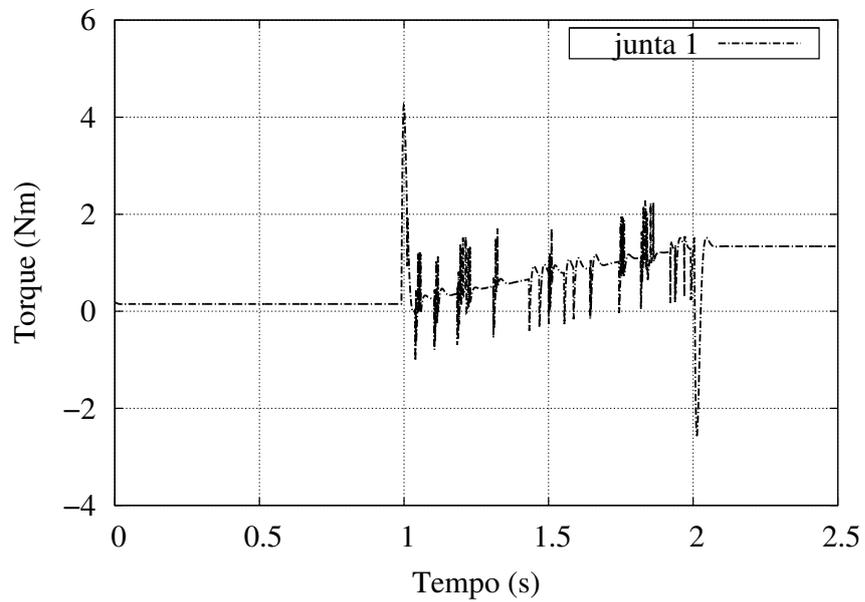


Figura 40: Torque na junta 1.

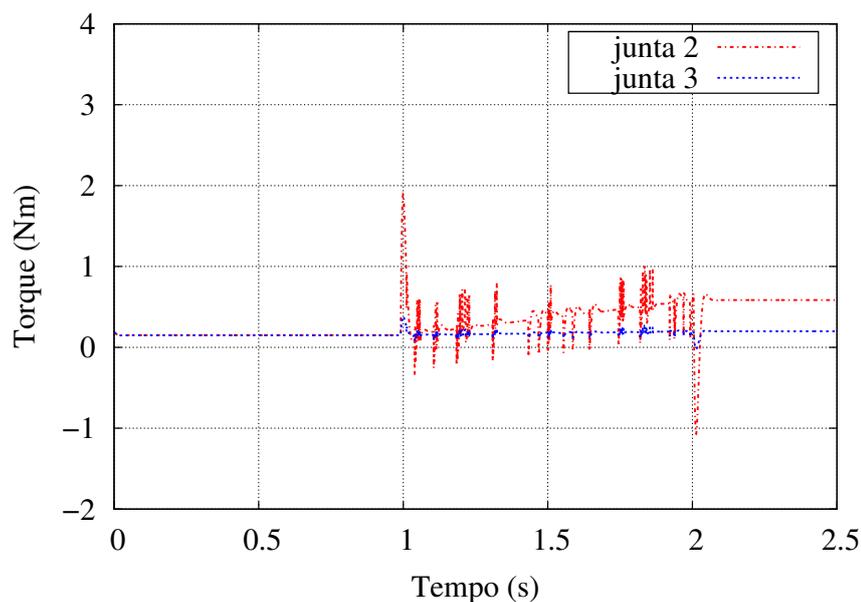


Figura 41: Torques nas juntas 2 e 3.

5.1.3 Interpolação Cúbica

Como forma de suavizar a trajetória seguida pela junta 1, optou-se por testar uma interpolação cúbica, esperando, dessa forma, uma variação mais suave da velocidade e da aceleração durante a interpolação cúbica, se comparadas com os resultados gerados na trajetória ao degrau e a na interpolação linear.

Pode-se notar que a resposta segue o sinal de referência com erro⁵ aproximadamente nulo. Esse comportamento deve-se ao fato das matrizes de ganho K_p e K_d não terem sido sintonizadas, comprovando a eficiência do controlador por torque calculado. Salienta-se também que o torque diminuiu sua magnitude se comparado a interpolação linear, reduzindo o torque a valores menores.

Na Figura 43 pode-se verificar que devido à suavização do sinal de referência, o torque na junta 1 tem o pico máximo em 2,20 Nm. Percebe-se que o torque das juntas 2 e 3 apresentam uma redução significativa em sua magnitude quando comparados ao apresentado na Figura 44. Parte desta redução deve-se à existência de saturação do sinal de torque e a outra parte deve-se à transição mais suave da interpolação cúbica se comparada a interpolação linear. A Figura 44 apresenta esses resultados.

⁵Vide Tabela 2.

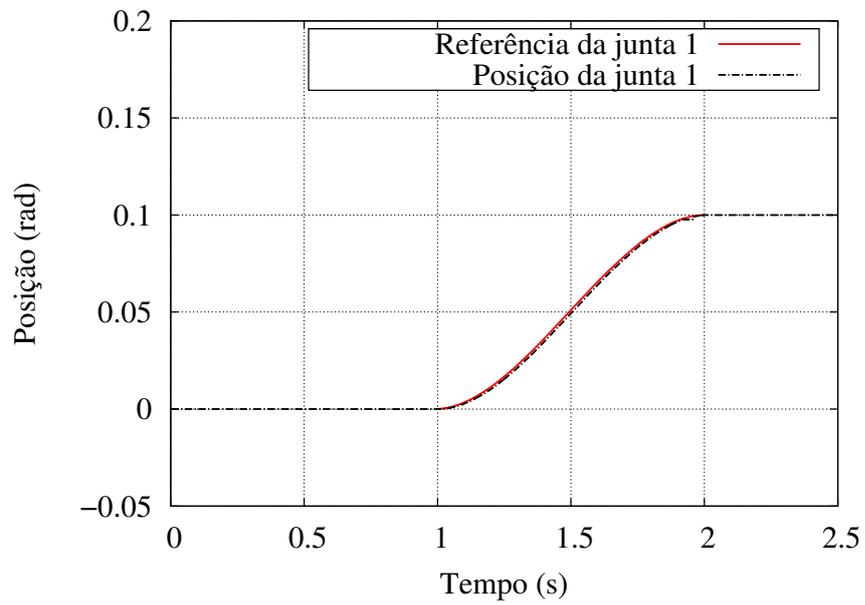


Figura 42: Sinal de referência e de posição da junta 1.

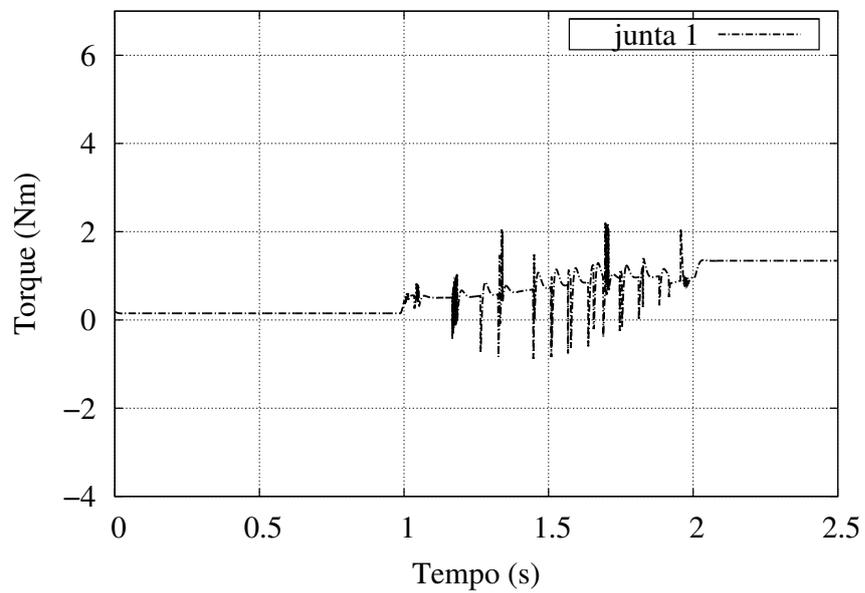


Figura 43: Torque na junta 1.

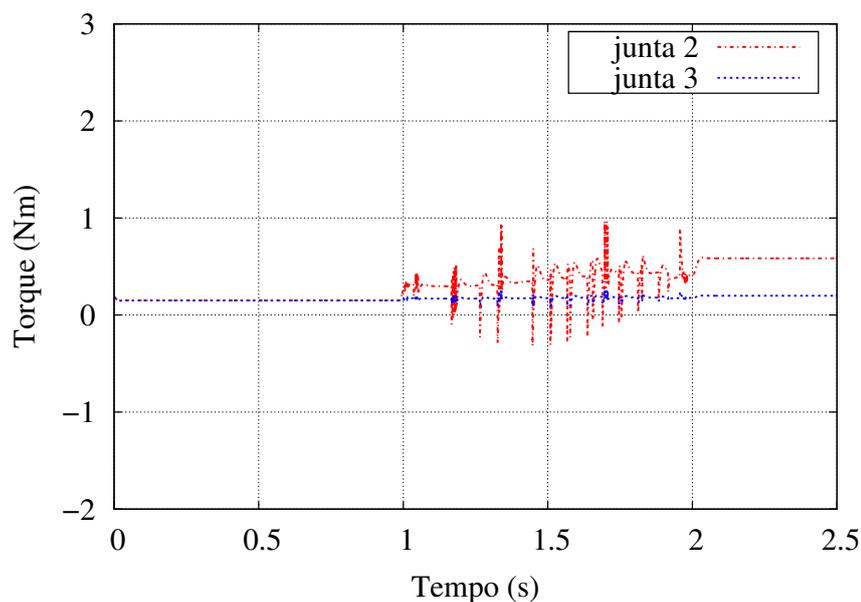


Figura 44: Torques nas juntas 2 e 3.

5.1.4 Interpolação Quíntupla

Como forma de suavizar a trajetória seguida pela junta 1, optou-se por testar uma interpolação quíntupla, onde se espera que as velocidades e acelerações tenham variações mais suaves se comparadas às demais trajetórias. Verifica-se na Figura 8 que ao utilizar uma interpolação quíntupla como entrada de referência, utiliza-se tanto a posição, quanto a velocidade e a aceleração no cálculo (termos de inércia, de forças centrífugas, de coriolis e dos termos gravitacionais), o que tende a tornar melhor o resultado de seguimento da trajetória.

Para a trajetória da Figura 45, pode-se notar na que a resposta segue o sinal de referência com erro ⁶ aproximadamente nulo.

Na Figura 46 pode-se verificar que devido à suavização do sinal de referência, onde o torque na junta 1 tem o pico máximo é de 2,76 Nm. Percebe-se que o torque das juntas 2 e 3 não apresenta uma redução significativa em sua magnitude comparado a interpolação cúbica. A Figura 44 demonstra esses resultados.

⁶Vide Tabela 2.

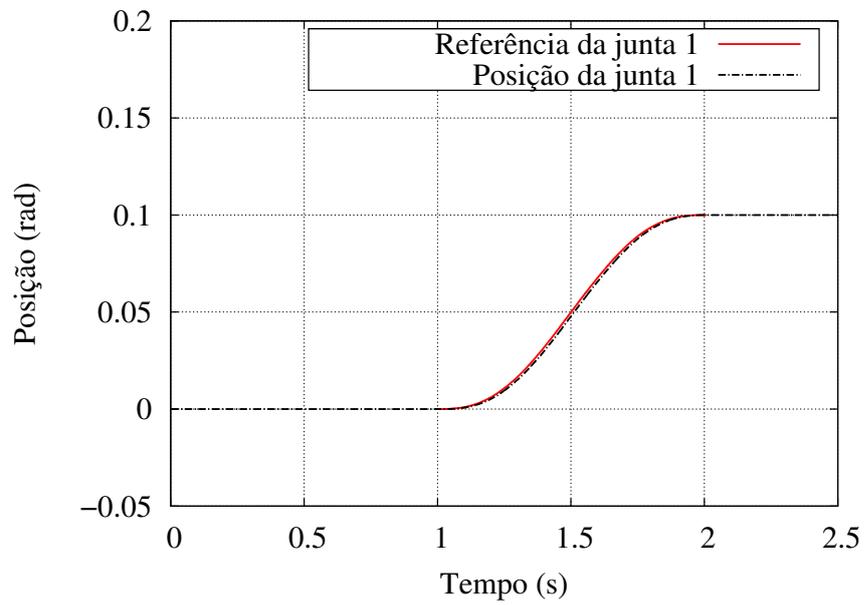


Figura 45: Sinal de referência e de posição da junta 1.

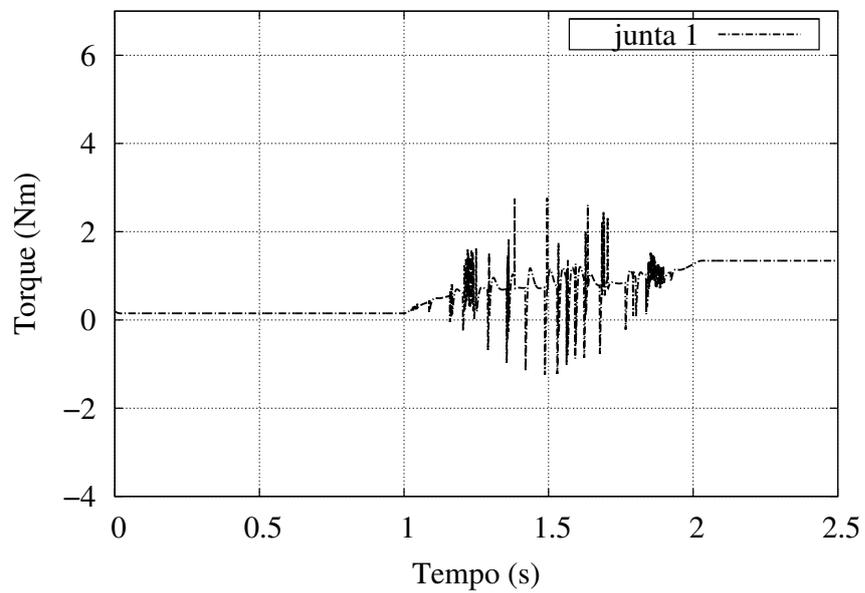


Figura 46: Torque na junta 1.

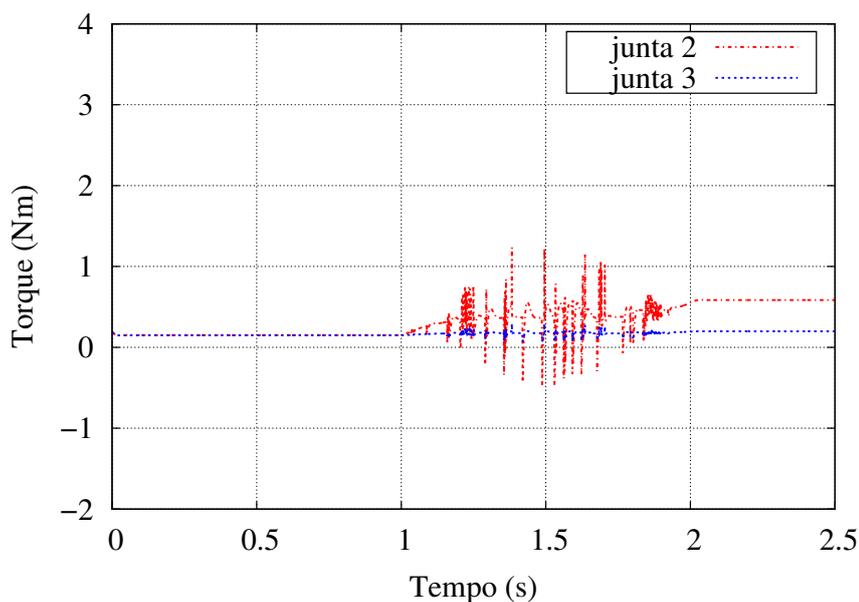


Figura 47: Torques nas juntas 2 e 3.

Apresenta-se a seguir um resumo dos resultados obtidos na junta 1 para a simulação sem saturação. Na Tabela 1, o valor de ultrapassagem máxima (M_P), o erro em regime permanente (ϵ_{RP}) e o torque máximo (τ_{MAX}), para a trajetória ao degrau são apresentados:

Tabela 1: Tabela de valores de sinais sem saturação para a trajetória ao degrau.

Referência	M_P [%]	ϵ_{RP} [rad]	τ_{MAX} [Nm]
Trajetoária ao Degrau	4,5	0,0955	1058

Na Tabela 2, o erro de posição (ϵ_R) e o torque máximo τ_{MAX} , para a interpolação linear, cúbica e quártupla são apresentados:

Tabela 2: Tabela de valores de sinais sem saturação para a interpolação linear, cúbica e quártupla.

Referência	ϵ_R [rad]	τ_{MAX} [Nm]
Interpolação Linear	1,0e-04	4,24
Interpolação Cúbica	1,0e-04	2,20
Interpolação Quártupla	1,0e-04	2,76

Na Tabela 1 verifica-se que o controlador respondeu satisfatoriamente com um erro em M_P e ϵ_{RP} aproximadamente nulos. Como a simulação é sem saturação, o torque

apresenta um valor absurdo. Na tabela 2 verifica-se que o erro de regime é praticamente nulo e os torques permanecem em valores de baixa magnitude para os casos de interpolação linear, cúbica e quártupla.

5.2 Simulação com Saturação no Sinal de Controle

5.2.1 Trajetória ao Degrau

A seguir, apresentam-se os resultados da aplicação de uma trajetória ao degrau de referência de 0,78 radianos de amplitude na junta 1. O valor de referência é diferente do aplicado na simulação sem saturação no sinal de controle, pois uma referência menor que 0,78 radianos não apresenta saturação no sinal de controle.

A Figura 48 ilustra o sinal de referência e de posição da junta 1. Pode-se perceber que o sinal de posição segue o sinal de referência, com erro ⁷ e mantém-se próximo de zero. Este comportamento deve-se a estrutura de controle por torque calculado, o que comprova a eficiência da Equação 24.

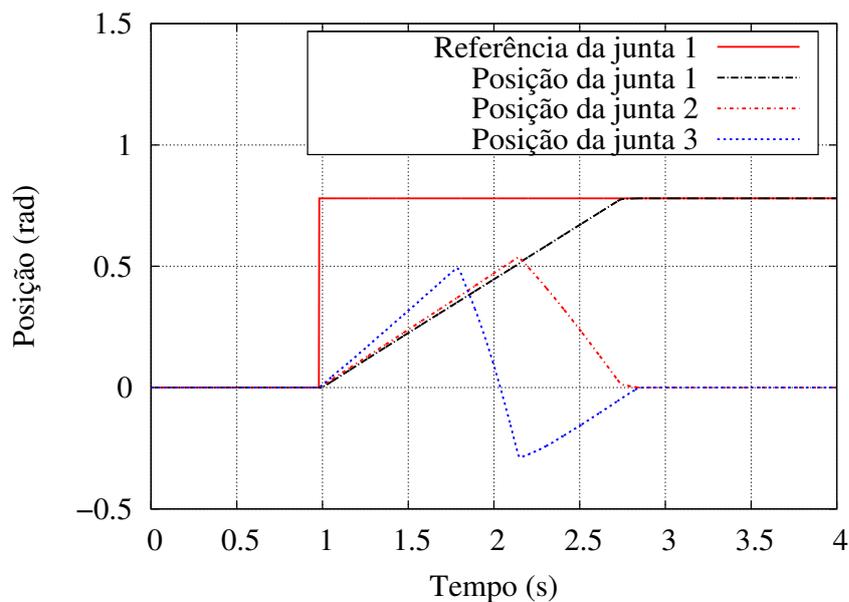


Figura 48: Sinal de referência e de posição da junta 1,2 e 3.

A Figura 49 mostra que o sinal de torque na junta 1 não excede o valor do torque nominal de 20,7 Nm ⁸.

⁷Vide Tabela 1.

⁸Vide Tabela 6, Apêndice B.

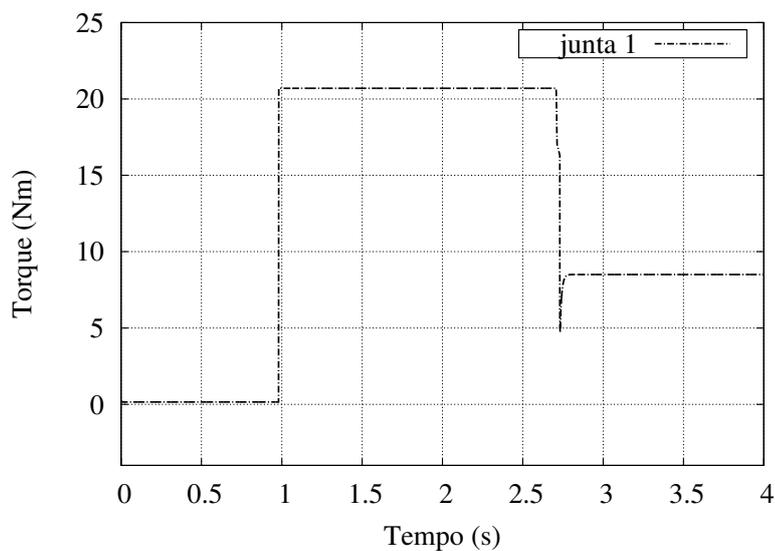


Figura 49: Torque na junta 1.

Percebe-se na Figura 50, a existência de torque nas juntas 2 e 3. Estes torques são oriundos das forças de compensação necessárias para que as juntas do joelho (θ_2) e do pé (θ_3), mantenham-se em suas posições iniciais.

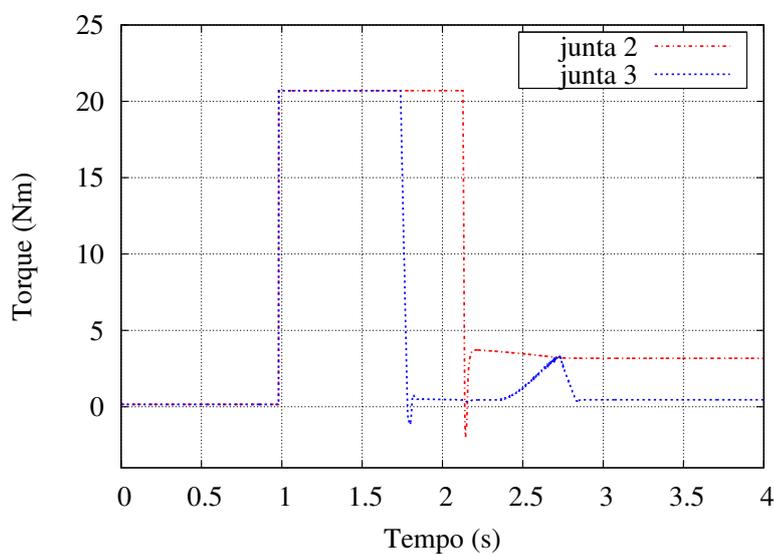


Figura 50: Torques nas juntas 2 e 3.

A seguir apresentam-se os três outros tipos de entrada de referência e seus respectivos resultados.

5.2.2 Interpolação Linear

Pode-se notar na Figura 51 que a resposta segue o sinal de referência com ⁹ erro aproximadamente nulo. Salienta-se também que o torque diminuiu sua magnitude se comparado a trajetória ao degrau, reduzindo o torque a valores menores.

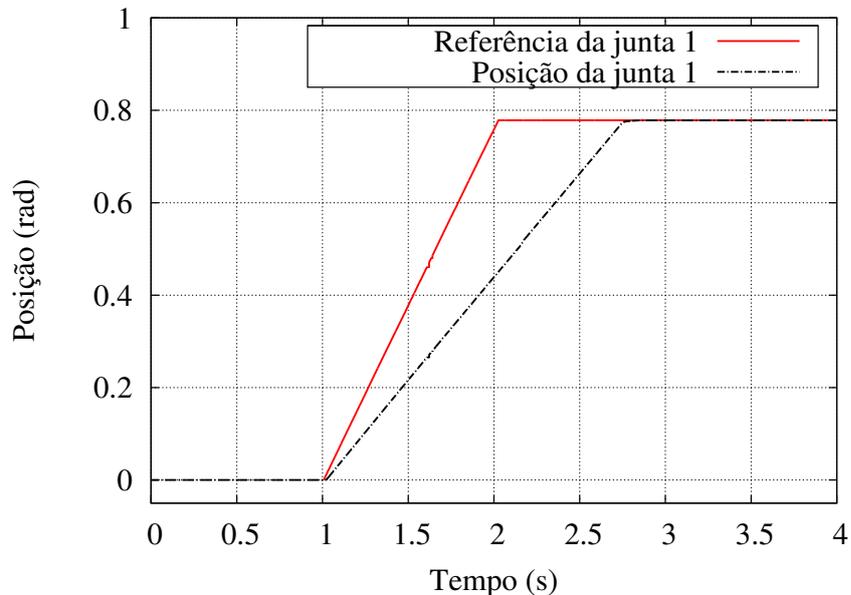


Figura 51: Sinal de referência e de posição da junta 1.

Na Figura 52 pode-se verificar que devido a suavização do sinal de referência, o torque na junta 1 é reduzido ao pico máximo de 4,23 Nm. Percebe-se que os torques das juntas 2 e 3, apresenta uma redução significativa em sua magnitude quando comparado ao apresentado na Figura 38. Parte desta redução deve-se a existência de saturação do sinal de torque e a outra parte deve-se a transição mais suave da interpolação linear comparada a trajetória ao degrau. A Figura 53 apresenta esses resultados.

⁹Vide Tabela 2.

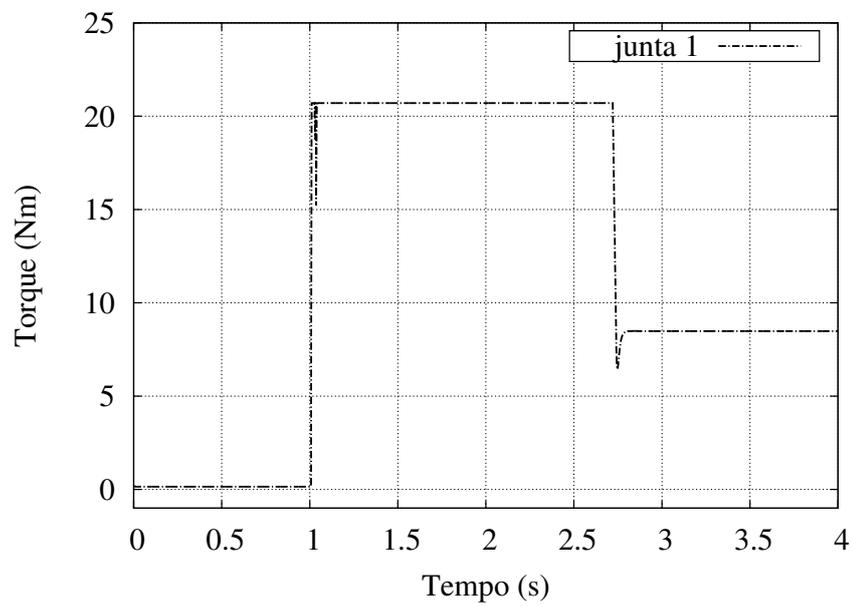


Figura 52: Torque na junta 1.

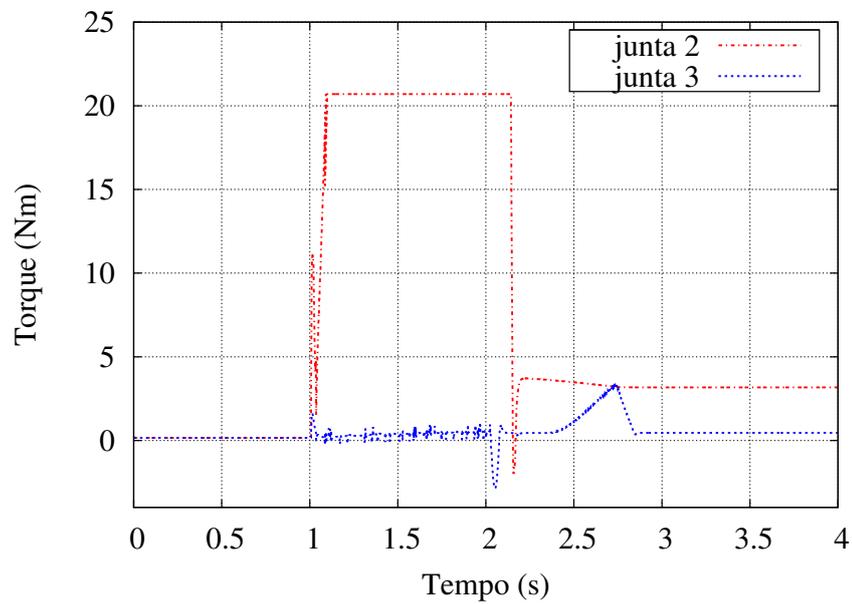


Figura 53: Torques nas juntas 2 e 3.

5.2.3 Interpolação Cúbica

Como forma de suavizar a trajetória seguida pela junta 1, optou-se por testar uma interpolação cúbica. Esperando dessa forma uma variação mais suave da velocidade e da aceleração realizada durante a trajetória, se comparadas com a trajetória ao degrau e a interpolação linear.

Pode-se notar que também no caso da interpolação cúbica, a resposta segue o sinal de referência com erro ¹⁰ aproximadamente nulo.

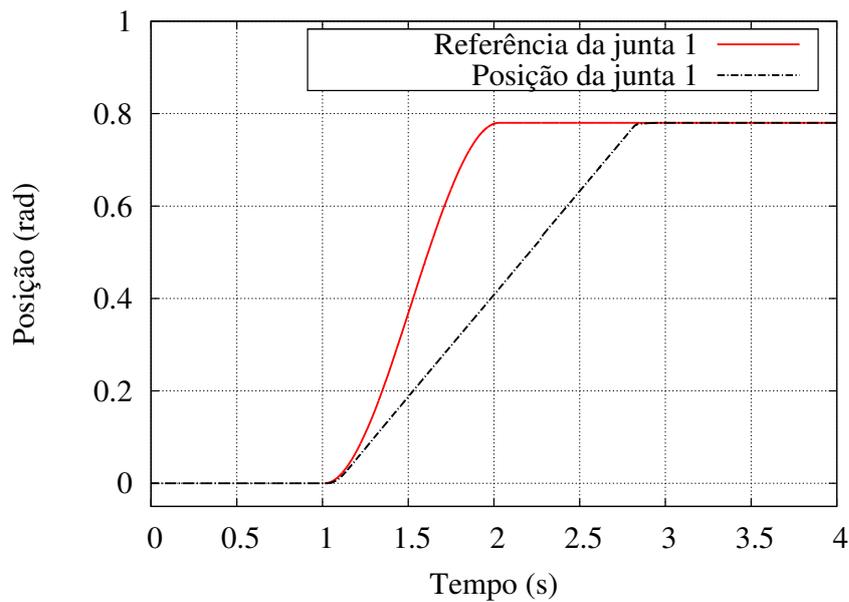


Figura 54: Sinal de referência e de posição da junta 1.

Na Figura 55 pode-se verificar que o torque na junta 1 tem seu pico máximo em 2,53 Nm. Percebe-se que os torques das juntas 2 e 3 não apresentam uma redução significativa em sua magnitude quando comparado a interpolação linear, porém não apresentam mais os picos de sinal nos pontos de inflexão, como pode ser visto na Figura 56.

¹⁰Vide Tabela 2.

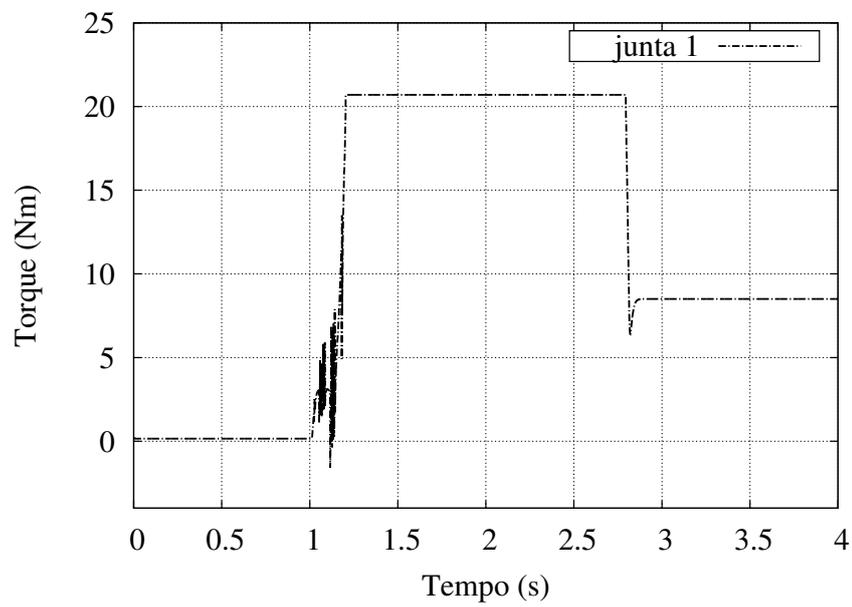


Figura 55: Torque na junta 1.

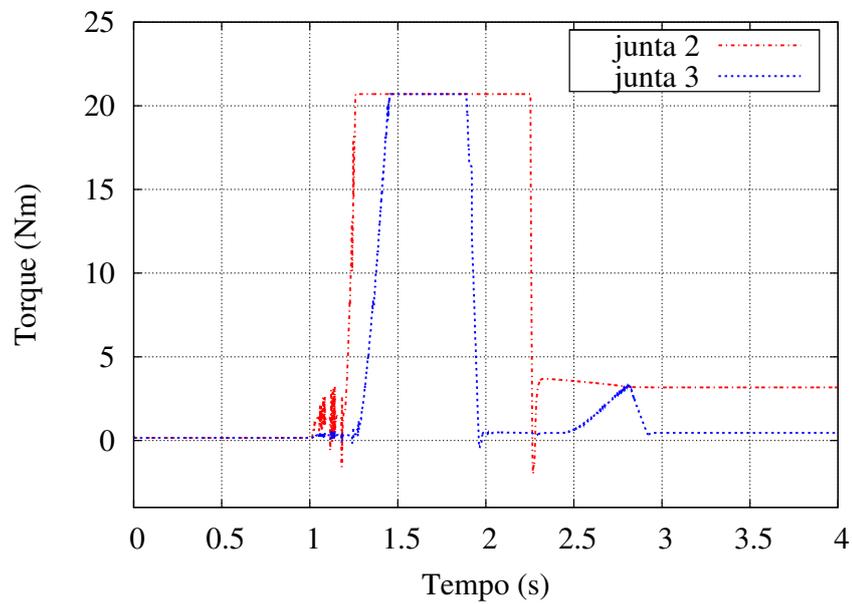


Figura 56: Torques nas juntas 2 e 3.

5.2.4 Interpolação Quintupla

Da mesma forma que a simulação sem saturação, optou-se por testar uma interpolação quintupla. Espera-se que as velocidades e acelerações tenham variações mais suave quando comparada à interpolação cúbica. Verifica-se na Figura 8 que, ao utilizar uma interpolação quintupla como entrada de referência de posição, velocidade e aceleração são consideradas no calculo (termos das forças inerciais, de forças centrífugas e de coriolis e de termos gravitacionais).

Pode-se notar na que a resposta segue o sinal de referência com erro ¹¹ aproximadamente nulo.

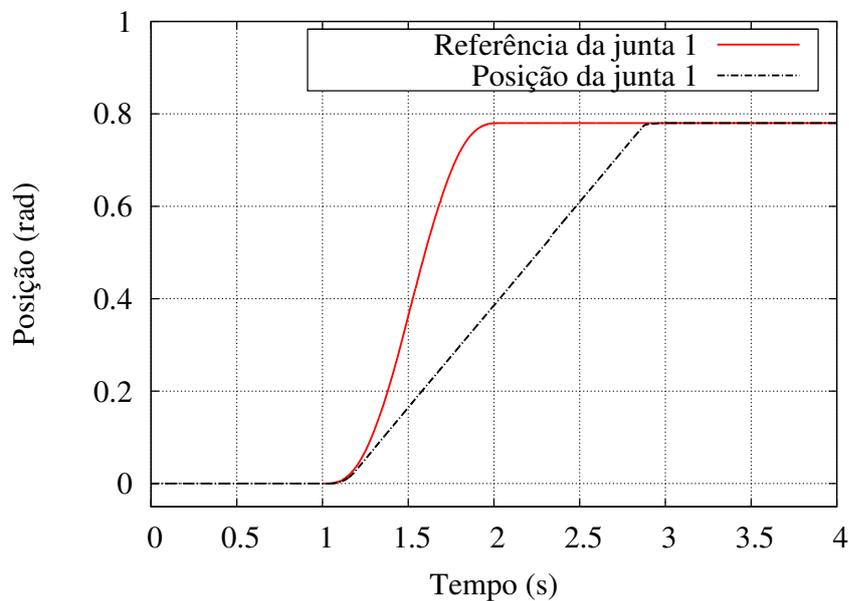


Figura 57: Sinal de referência e de posição da junta 1.

Na Figura 58 pode-se verificar que o torque na junta 1 aumentou sua magnitude quando comparado ao torque da junta 1 da interpolação cúbica, seu pico máximo de 2,73 Nm. Percebe-se que os torques das juntas 2 e 3 não apresentam uma redução significativa em sua magnitude quando comparado a interpolação cúbica. A Figura 59 mostra esses resultados.

¹¹Vide Tabela 2.

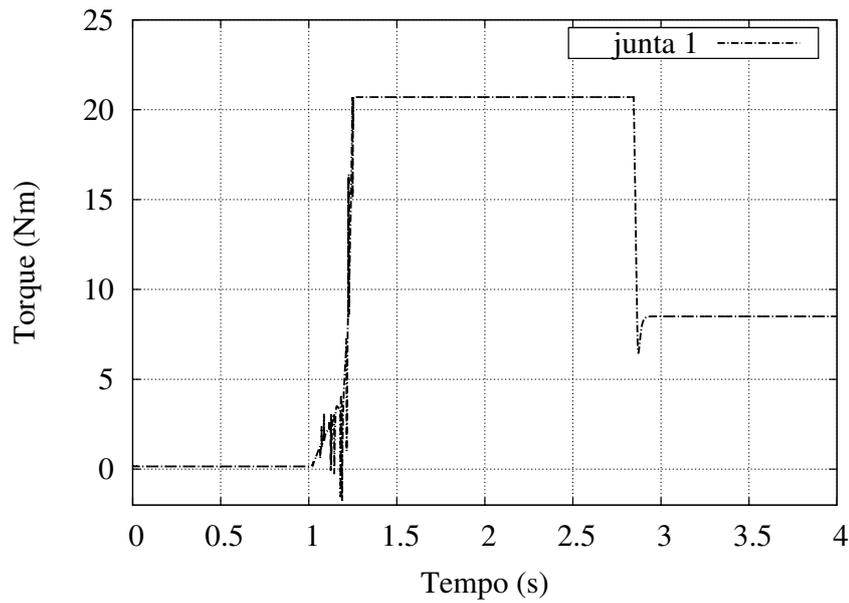


Figura 58: Torque na junta 1.

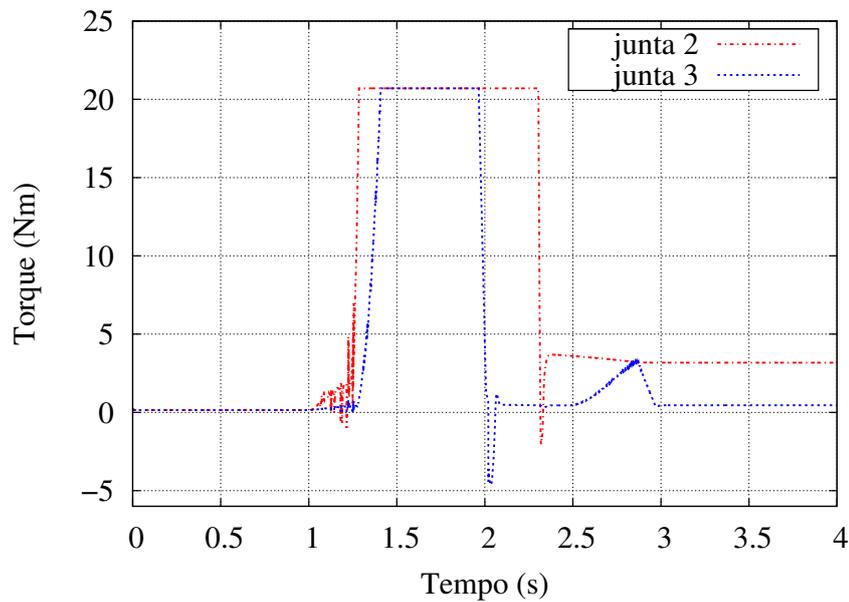


Figura 59: Torques nas juntas 2 e 3.

Apresenta-se a seguir um resumo dos resultados obtidos na junta 1 para a simulação com saturação. Assim, na Tabela 3 tem-se o valor de ultrapassagem máxima (M_P), o

erro em regime permanente (ϵ_{RP}) e o torque máximo (τ_{MAX}), para a trajetória ao degrau são apresentados:

Tabela 3: Tabela de valores de sinais sem saturação para a trajetória ao degrau.

Referência	M_P [%]	ϵ_{RP} [rad]	τ_{MAX} [Nm]
Trajectoria ao Degrau	0,3	0,0997	20,7

onde, o erro de posição (ϵ_R) e o torque máximo τ_{MAX} , para a interpolação linear, cúbica e quártupla são:

Tabela 4: Tabela de valores de sinais sem saturação para a interpolação linear, cúbica e quártupla.

Referência	ϵ_R [rad]	τ_{MAX} [Nm]
Interpolação Linear	1,0e-04	20,7
Interpolação Cúbica	1,0e-04	20,7
Interpolação Quártupla	1,0e-04	20,7

Nas Tabelas 3 e 4 verifica-se que o erro é nulo e como tem-se a saturação nos valores do sinal de controle o torque máximo permanece em valores de baixa magnitude.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresenta a implementação de um pacote para o ROS, contendo um modelo simplificado dos membros inferiores de um robô bípede ¹ com seis graus de liberdade e um sistema de controle multivariável, utilizando a técnica de controle por torque calculado. Por meio da implementação deste pacote foram realizados testes de funcionamento do sistema de controle utilizando uma trajetória ao degrau e um gerador de trajetória, com o uso de interpolação linear, cúbica e quártupla.

A contribuição desse trabalho está na implementação de um pacote para a área de controle de robô bípedes, disponibilizando aos pesquisadores o conceito de como é realizada a implementação de um controlador genérico no ROS.

Sobre a estrutura de controle no ROS, a grande maioria dos controladores utilizam uma arquitetura SISO e uma estrutura de controle PID. Esse conceito deve-se pelo motivo de que a maioria dos atuadores disponíveis no mercado já vem com uma estrutura de controle de PID implementada no próprio *hardware*. Neste trabalho, ao contrário, foi implementado um controlador multivariável utilizando a técnica de controle por torque calculado. O nível de dificuldade para a implementação desse sistema de controle foi alto, devido à carência de uma boa documentação do ROS e da API do KDL. Inicialmente as dificuldades de implementação do controlador por torque calculado foram:

- Realizar o *parser* do arquivo URDF para o KDL, pois ao realizar este *parser* deve-se tomar um certo cuidado em relação à matriz de inércia, pois o KDL especifica a matriz de inércia no centro da junta e o URDF especifica a matriz de inércia no centro do elo;
- Identificar e entender o funcionamento das interligações baseadas na cinemática do tipo serial (KDL::Chain) e do tipo árvore (KDL::Tree). A ideia é ler os parâmetros do URDF e montar a cadeia cinemática no KDL utilizando basicamente a classe `kdl_parser::treeFromString`, criando uma cadeia cinemática para cada perna;

¹Modelado em uma ferramenta CAD, SolidWorks.

- Identificar e entender o funcionamento do algoritmo de recursividade de Newton-Euler (KDL: `:ChainIdSolver_RNE`) para gerar um vetor de torques para cada junta.

Na parte da implementação do controlador multivariável, a dificuldade foi no entendimento do funcionamento do fluxo de controle no ROS, pois a falta de um documento que explique de um forma clara e objetiva o fluxo de controle tornou a implementação complexa. A Figura 25 ilustra esse fluxo de controle. A partir do estudo e entendimento de alguns algoritmos que estão envolvidos neste fluxo de controle, foi desenvolvido para este trabalho um diagrama que descreve o fluxo de controle de uma forma mais objetiva e clara, disponibilizando a outros pesquisadores uma visão de como esse fluxo ocorre no ROS. Esse diagrama pode ser visto na Figura 33.

Os resultados sobre a implementação do pacote foram satisfatórios, bem como os resultados da implementação do controlador multivariado por torque calculado, tendo em vista que os parâmetros de ganho do controlador não foram sintonizados, pois não é o objetivo deste trabalho. Os resultados foram divididos em simulação sem limite de saturação no sinal de controle e simulação com limite de saturação no sinal de controle, onde os melhores resultados apresentados, como esperado, foram nas simulações com limite de saturação no sinal de controle, pois utilizam-se nessas simulações os limites reais de um atuador² e verifica-se que o erro é nulo, obtendo torque máximo de baixa magnitude.

Ao utilizar o ROS como plataforma de simulação e testes de sistemas robóticos, observa-se uma redução no tempo de desenvolvimento, devido à reutilização de alguns pacotes que já estão disponíveis pela comunidade do ROS, como, por exemplo, o pacote `joint_states_publisher` (DAVID, 2013) e o pacote `robot_state_publisher` (WIM, 2013). Por outro lado, devido a carência de uma boa documentação sobre o funcionamento de alguns desses pacotes, a curva de aprendizado para realizar a implementação do pacote proposto para esse trabalho foi lenta e complexa, devido a interpretação dos pacotes já existentes ou na tentativa de reutilizá-los.

Entre os possíveis trabalhos futuros, está a criação de um protótipo do robô bípede Orbit em escala 1:1. O pacote implementado no presente trabalho será o ponto de partida para o desenvolvimento de um pacote voltado ao acionamento do *hardware* do robô. A estrutura de controle multivariável por torque calculado será aquela ilustrada na Figura 33. Também propõem-se implementar um sistema de controle em tempo real para o robô.

²Vide Tabela 6

REFERÊNCIAS

- ARMIN, H. **Wiki of the NAO Robot**. Disponível em: <<http://wiki.ros.org/Robots/Nao>>. Acesso em: 28 out. 2013.
- BAZANELLA, A. S.; SILVA, J. M. G. **Sistemas de Controle: princípios e métodos de projeto**. 1th. ed. [S.l.]: UFRGS, 2005.
- BEZERRA, C. A. B. **Desenvolvimento de um Robô Bípede para Locomoção em Ambiente Desestruturado**. 2002. 204p. Tese (Doutorado em Engenharia Mecânica) — Universidade Estadual de Campinas, São Paulo, 2002.
- BEZERRA, C.; ZAMPIERI, D. Biped Robots: the state of art. In: HMM2004 INTERNATIONAL SYMPOSIUM ON HISTORY OF MACHINES AND MECHANISMS, 2004, Cassino Frosinone. **Proceedings...** [S.l.: s.n.], 2004. p.371–389.
- BRUYNINCKX, H. Open Robot Control Software: the OROCOS project. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2001, Seoul. **Proceedings...** [S.l.: s.n.], 2001. p.2523–2528.
- CHESTNUTT, J. E. et al. Footstep Planning for the Honda ASIMO Humanoid. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2005, Barcelona. **Proceedings...** [S.l.: s.n.], 2005. p.629–634.
- CHEVALLEREAU, C. et al. **Bipedal Robots: modeling, design and walking synthesis**. [S.l.]: Wiley, 2009.
- COMPANY honda motor. **Evolution of ASIMO Robot**. Disponível em: <<http://asimo.honda.com/asimo-specs/>>. Acesso em: 05 jun. 2013.
- CRAIG, J. J. **Introduction to Robotics, Mechanics and Control**. 2th. ed. [S.l.]: Addison Wesley, 1989.

DAVE, C. **Wiki of the ros_control with Gazebo**. Disponível em: <http://gazebosim.org/wiki/Tutorials/1.9/ROS_Control_with_Gazebo>. Acesso em: 22 abr. 2013.

DAVE, C. **Wiki of the ROS for kdl_parser**. Disponível em: <http://wiki.ros.org/kdl_parser>. Acesso em: 25 set. 2013.

DAVID, L. **Wiki of the ROS for Joint States Publisher**. Disponível em: <http://wiki.ros.org/joint_state_publisher>. Acesso em: 22 set. 2013.

DEL PRETE, A.; SENTIS, L. **Torque-Controlled Humanoids Robots**. Disponível em: <<http://www.codyco.eu/index.php/workshop-humanoids2013>>. Acesso em: 22 set. 2013.

FEATHERSTONE, R. **Robot Dynamics Algorithms**. [S.l.]: Springer-Verlag, 1987.

FU, K. S.; GONZALEZ, R. C.; LEE, C. S. G. **Robotics Control, Sensing, Vision and Intelligence**. 2th. ed. [S.l.]: McGrawHill, 1987.

GIENGER, M.; LOFFLER, K.; PFEIFFER, F. Towards the Design of a Biped Jogging Robot. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2001, Seoul. **Proceedings...** [S.l.: s.n.], 2001. p.4140–4145.

GOUAILLIER, D. et al. Mechatronic Design of NAO Humanoid. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2009, Kobe. **Proceedings...** [S.l.: s.n.], 2009. p.769–774.

HASEHAWA, Y.; ARAKAWA, T.; FUKUDA, F. Trajectory Generation for Bipedal Locomotion Robots. **Mechatronics**, [S.l.], v.10, n.1-2, p.67–89, Feb. 2000.

HEMAMI, H.; KATBAB, A. Constrained Inverted Pendulum Model For Evaluating Upright Postural Stability. **Journal of Dynamic Systems, Measurement and Control**, [S.l.], v.104, p.343–349, Jan. 1982.

HIROSE, M.; HAIKAWA, Y.; TAKENAKA, T. The Development of Honda Humanoid Robot. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1998, Leuven. **Proceedings...** [S.l.: s.n.], 1998. p.1321–1326.

HURMUZLU, Y. **Dynamics and Control of Bipedal Robots**. [S.l.]: Springer Berlin Heidelberg, 1998.

HURST, J. W.; RIZZI, A. A. Series Compliance for an Efficient Running Gait. **IEEE Robotics and Automation Magazine**, [S.l.], v.15, n.3, p.42–51, Sept. 2008.

IOAN, S. **Wiki of the URDF**. Disponível em: <<http://wiki.ros.org/urdf>>. Acesso em: 07 fev. 2014.

IOAN, S. **Wiki of the URDF for Model Element**. Disponível em: <<http://wiki.ros.org/urdf/XML/model>>. Acesso em: 07 mar. 2013.

IOAN, S. **Wiki of the URDF for Link Element**. Disponível em: <<http://wiki.ros.org/urdf/XML/link>>. Acesso em: 07 mar. 2013.

IOAN, S. **Wiki of the URDF for Joint Element**. Disponível em: <<http://wiki.ros.org/urdf/XML/joint>>. Acesso em: 07 mar. 2013.

IONESCU, T. Terminology for Mechanisms and Machine Science. **Mechanism and Machine Theory**, [S.l.], v.38, n.7-10, p.597–1111, Apr. 2003.

JALICS, L.; HEMAMI, H.; CLYMER, B. D. A Control Strategy for Terrain Adaptive Bipedal Locomotion. **Autonomous Robots** 4, [S.l.], v.4, n.3, p.243–257, Sept. 1997.

KANEKO, K. et al. Humanoid Robot HRP-4 - Humanoid Robotics Platform With LightWeight and Slim Body. In: IROS INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2011, San Francisco. **Proceedings...** [S.l.: s.n.], 2011. p.4400–4407.

KATO, I. et al. **Information-Power Machine with Senses and Limbs WABOT 1**. [S.l.]: Springer-Verlag, 1974.

KEN, C. **Wiki of the Node in the ROS**. Disponível em: <<http://wiki.ros.org/Nodes>>. Acesso em: 07 mar. 2013.

KEN, C. **Wiki of the Messages in the ROS**. Disponível em: <<http://wiki.ros.org/Messages>>. Acesso em: 07 mar. 2013.

KEN, C. **Wiki of the Topic in the ROS**. Disponível em: <<http://www.ros.org/wiki/Topics>>. Acesso em: 07 mar. 2013.

KEN, C. **Wiki of the Service in the ROS**. Disponível em: <<http://wiki.ros.org/Services>>. Acesso em: 07 mar. 2013.

KURAZUME, R. et al. Straight Legged Walking of a Biped Robot. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2005, Edmonton. **Proceedings...** [S.l.: s.n.], 2005. p.337–343.

LEE, T. T. Trajectory Planning and Control of a 3-link Biped Robot. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1988, Philadelphia. **Proceedings...** [S.l.: s.n.], 1988. p.820–823.

LOHMEIER, S.; BUSCHMANN, T.; ULBRICH, H. Humanoid Robot LOLA. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2009, Kobe. **Proceedings...** [S.l.: s.n.], 2009. p.775–780.

LOHMEIER, S. et al. Computer System and Control of Biped JOHNNIE. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2004, New Orleans. **Proceedings...** [S.l.: s.n.], 2004. p.4222–4227.

LUM, H. K.; ZRIBI, M.; SOH, Y. C. Planning and Control of a Biped Robot. **International Journal of Engineering Science**, [S.l.], v.37, n.10, p.1319–1349, Aug. 1999.

MACIEL, E. H. **Controle de Movimentação do Pé de um Robô Bípede**. 2011. 51p. Trabalho de Conclusão (Graduação) — Faculdade Anglo-Americano, Caxias do Sul, 2011.

MARSHALL, B. **Standard Data Format for Fabbers**. Disponível em: <<http://www.ennex.com/fabbers/StL.asp>>. Acesso em: 09 mar. 2013.

MISTRY, M. Inverse Kinematics with Floating Base and Constraints for Full Body Humanoid Robot Control. In: IEEE-RAS INTERNATIONAL CONFERENCE ON HUMANOID ROBOTS, 2008, Daejeon. **Proceedings...** [S.l.: s.n.], 2008. Não paginado.

MORECKI, A.; EKIEL, J.; FIDELUS, K. **Cybernetic Systems of Limb Movements in Man, Animals and Robots**. [S.l.]: E. Horwood, 1984.

OGATA, K. **Modern Control Engineering**. 5th. ed. [S.l.]: Prentice Hall, 2010.

OGURA, Y. et al. Development of a New Humanoid Robot WABIAN-2. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2006, Orlando. **Proceedings...** [S.l.: s.n.], 2006. p.76–81.

PFEIFFER, F.; LOFFLER, K.; GIENGER, M. The Concept of Jogging JOHNNIE. In: IEEE-ICRA INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2002, Washington. **Proceedings...** [S.l.: s.n.], 2002. p.3129–3135.

PIETER, V. Z. **Wiki of the Tulip Robot**. Disponível em: <<http://wiki.ros.org/Robots/TULip>>. Acesso em: 09 fev. 2013.

QUIGLEY, M. et al. ROS: an open-source robot operating system. In: ICRA WORKSHOP ON OPEN SOURCE SOFTWARE, 2009, Kobe. **Proceedings...** [S.l.]: IEEE Press, 2009. Não paginado.

- RAIBERT, M. Dynamic Legged Robot for Rough Terrain. In: IEEE INTERNATIONAL CONFERENCE ON HUMANOID ROBOTS, 2010, Nashville. **Proceedings...** [S.l.: s.n.], 2010. p.1.
- RIEZENMAN, M. J. Robots Stand on Own Two Feet. In: IEEE SPECTRUM, 2002, San Diego. **Proceedings...** [S.l.: s.n.], 2002. v.39, p.24–25.
- RUSSELL, S. **Open Dynamics Engine**. Disponível em: <<http://www.ode.org>>. Acesso em: 28 ago. 2012.
- SAKAGAMI, Y. et al. The Intelligent ASIMO: system overview and integration. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2002, Lausanne. **Proceedings...** [S.l.: s.n.], 2002. v.3, p.2478–2483.
- SANTANA, R. E. S. **Projeto de um Robô Bípede para a Reprodução da Marcha Humana**. 2005. 188p. Tese (Mestrado em Engenharia Mecânica) — Universidade de São Paulo, São Paulo, 2005.
- SARDAIN, P.; ROSTAMI, M.; BESSONNET, G. Open Robot Control Software: the OROCOS project. In: IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS - PART A: SYSTEMS AND HUMANS, 1998, Edmonton. **Proceedings...** [S.l.: s.n.], 1998. v.28, p.823–828.
- SCHIEHLEN, W. O. **Multibody Systems Handbook**. [S.l.]: Springer-Verlag, 1990.
- SHERMAN, M. A.; SETH, A.; DELP, S. L. Simbody:multibody dynamics for biomedical research. **Procedia IUTAM**, [S.l.], v.2, n.2, p.241–261, Jan. 2011.
- SIAS, F. R. J.; ZHENG, Y. F. How Many Degrees-of-freedom Does a Biped Need. In: IEEE INTERNATIONAL WORKSHOP ON INTELLIGENT ROBOTS AND SYSTEM, 1990, Ibaraki. **Proceedings...** [S.l.: s.n.], 1990. p.297–302.
- SICILIANO, B.; SCIAVICCO, L. **Robotics: modelling, planning and control**. [S.l.]: Springer-Verlag, 2009.
- SIQUEIRA, A. A. G.; TERRA, M. H. Nonlinear H-infinity Control Applied to Biped Robots. In: IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL, 2006, Munich. **Proceedings...** [S.l.: s.n.], 2006. p.2190–2195.
- SLOTINE, J. J. E.; LI, W. A. **Applied Nonlinear Control**. [S.l.]: Prentice Hall, 1991.
- STEPHEN, B. **SolidWorks to URDF Exporter**. Disponível em: <http://wiki.ros.org/sw_urdf_exporter>. Acesso em: 15 fev. 2013.

TORRES, S. O. A. **Avaliação de Protótipo Mecatrônico de Locomoção Bípede**. 2006. 111p. Tese (Mestrado em Engenharia Mecânica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

TSAI, L. **Robot Analysis**: the mechanics of serial and parallel manipulators. [S.l.]: Wiley, 1999.

VAN ZUTVEN, P. W. M. et al. Tech United Eindhoven Robocup Adult Size Humanoid Team Description. In: ANNUAL ROBOCUP INTERNATIONAL SYMPOSIUM, 17., 2013, Vienna. **Proceedings...** [S.l.: s.n.], 2013. Não paginado.

WAARD, M. de; INJA, M.; VISSER, A. Analysis of Flat Terrain for the Atlas Robot. In: JOINT CONFERENCE OF AI & ROBOTICS AND 5TH ROBOCUP IRAN OPEN INTERNATIONAL SYMPOSIUM, 3., 2013, Qazvin. **Proceedings...** [S.l.: s.n.], 2013. v.3, p.1–6.

WILLIAM, D. M. **Fisiologia do Exercício**: nutrição, energia e desempenho humano. 7th. ed. [S.l.]: Guanabara Koogan, 2011.

WIM, M. **Wiki of the ROS for Robot State Publisher**. Disponível em: <http://wiki.ros.org/robot_state_publisher>. Acesso em: 21 set. 2013.

YANG, J. A control Study of a Kneeless Biped Locomotion System. **Journal of the Franklin Institute**, [S.l.], v.331, p.125–143, Mar. 1994.

APÊNDICE A ESTRUTURA MECÂNICA DO ROBÔ BÍ-PEDE ORBIT

Neste apêndice, estão relacionados aos desenhos da estrutura mecânica do robô bípede Orbit, como dimensões e parâmetros dos corpos rígidos.

A.1 Dimensões

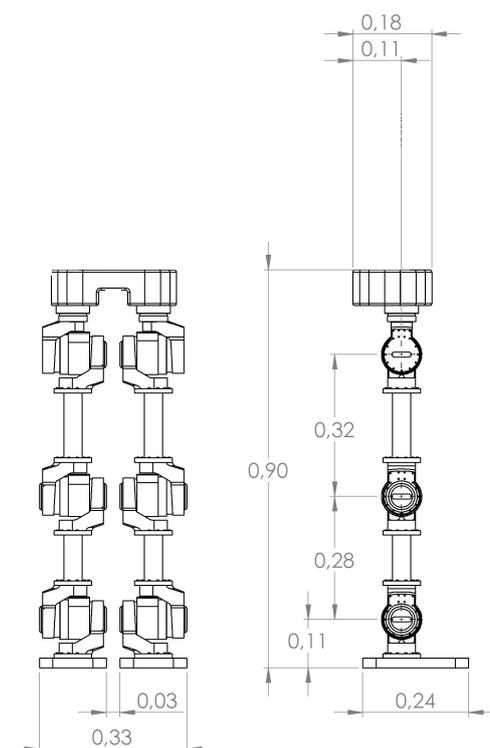


Figura 60: Dimensões da estrutura proposta (m).

A.2 Parâmetros dos Corpos Rígidos

Esta seção descreve o peso (kg), centro de massa ($CoM(S)$) e matriz de inercia ($[I_o(S)]_R$) de cada sólido (S) que compõem a estrutura mecânica do robô bípede Orbit.

A posição do centro de massa e a matriz de inercia está descrito em relação ao sistema de coordenadas local do sólido (C), sendo (o, R).

Todos os sólidos (C) e o sistema de coordenadas local são descritos em relação à postura zero: de pé, com as pernas esticadas.

$$CoM(S) = \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_{(o,R)} \quad (m) \quad (28)$$

$$[I_o(S)]_R = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}_R \quad (kg * m^2) \quad (29)$$



Figura 61: Cintura

$$\begin{aligned} \text{Peso} &= 0.011 \\ CoM(S) &= \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_{(o,R)} \\ [I_o(S)]_R &= \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}_R \end{aligned}$$



Figura 62: Fêmur perna esquerda

$$\text{Peso} = 0.99705$$

$$\text{CoM}(S) = \begin{bmatrix} 4.3507E-08 \\ -0.13576 \\ 0.010901 \end{bmatrix}_{(o,R)}$$

$$[I_o(S)]_R = \begin{bmatrix} 0.021866 & -1.02E-09 & 7.074E-11 \\ 0.00000 & 0.0017346 & 0.0040132 \\ 0.00000 & 0.00000 & 0.021066 \end{bmatrix}_R$$

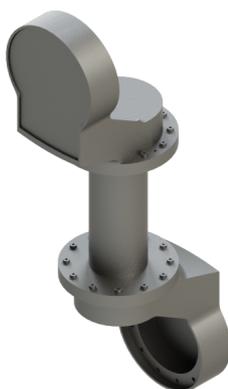


Figura 63: Tíbia perna esquerda

$$\text{Peso} = 0.98468$$

$$\text{CoM}(S) = \begin{bmatrix} 3.2552E-08 \\ -0.1172 \\ 0.010901 \end{bmatrix}_{(o,R)}$$

$$[I_o(S)]_R = \begin{bmatrix} 0.016465 & -3.3684E-11 & -2.7363E-10 \\ 0.00000 & 0.00173 & 0.0034283 \\ 0.00000 & 0.00000 & 0.015665 \end{bmatrix}_R$$

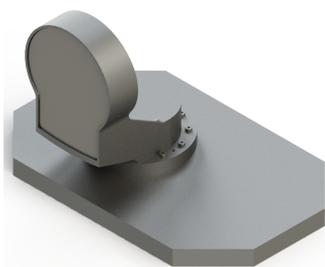


Figura 64: Pé perna esquerda

$$\text{Peso} = 1.0074$$

$$\text{CoM}(S) = \begin{bmatrix} 0.015223 \\ -0.050221 \\ 0.024383 \end{bmatrix}_{(o,R)}$$

$$[I_o(S)]_R = \begin{bmatrix} 0.003588 & -0.00067902 & -0.00020753 \\ 0.00000 & 0.003698 & 0.00051726 \\ 0.00000 & 0.00000 & 0.0048067 \end{bmatrix}_R$$



Figura 65: Pé perna esquerda

$$\text{Peso} = 0.99705$$

$$CoM(S) = \begin{bmatrix} 4.3461E-08 \\ -0.13576 \\ 0.010901 \end{bmatrix}_{(o,R)}$$

$$[I_o(S)]_R = \begin{bmatrix} 0.021866 & -1.0172E-09 & 7.1171E-11 \\ 0.00000 & 0.0017346 & 0.0040132 \\ 0.00000 & 0.00000 & 0.021066'' \end{bmatrix}_R$$



Figura 66: Pé perna esquerda

$$\text{Peso} = 0.98468$$

$$CoM(S) = \begin{bmatrix} 3.2666E-08 \\ -0.1172 \\ 0.010901 \end{bmatrix}_{(o,R)}$$

$$[I_o(S)]_R = \begin{bmatrix} 0.016465 & -4.8383E-11 & 2.7547E-10 \\ 0.00000 & 0.00173 & 0.00173 \\ 0.00000 & 0.00000 & 0.015665'' \end{bmatrix}_R$$



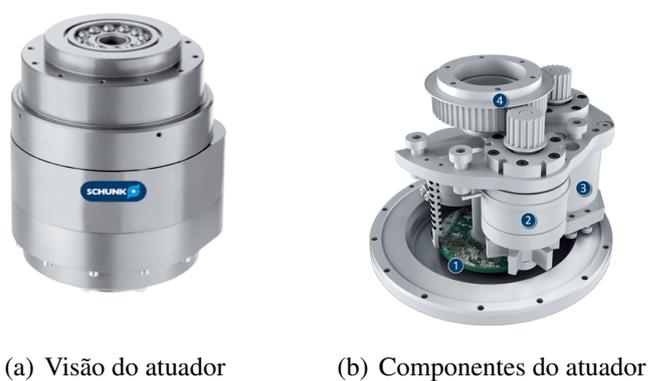
Figura 67: Pé perna esquerda

$$\begin{aligned}
 & \text{Peso} = 1.0074 \\
 & CoM(S) = \begin{bmatrix} -0.015223 \\ -0.050221 \\ 0.024383 \end{bmatrix}_{(o,R)} \\
 & [I_o(S)]_R = \begin{bmatrix} 0.003588 & 0.00067903 & 0.00020753 \\ 0.00000 & 0.003698 & 0.00051726 \\ 0.00000 & 0.00000 & 0.0048067'' \end{bmatrix}_R
 \end{aligned}$$

APÊNDICE B ATUADOR SCHUNK

Este apêndice apresenta um pequeno descritivo sobre o atuador fabricado pela empresa alemã Schunk.

B.1 Modelo PRL 80



(a) Visão do atuador

(b) Componentes do atuador

Figura 68: Modelo PRL 80.

A Figura 68b indica os seguinte componentes:

Tabela 5: Tabela de componentes do motor PRL80 da Schunk.

Número	Descrição
1	Eletrônica de potência e de controle
2	Servo Motor
3	Harmonic Drive
4	Engrenagens

B.2 Dimensões

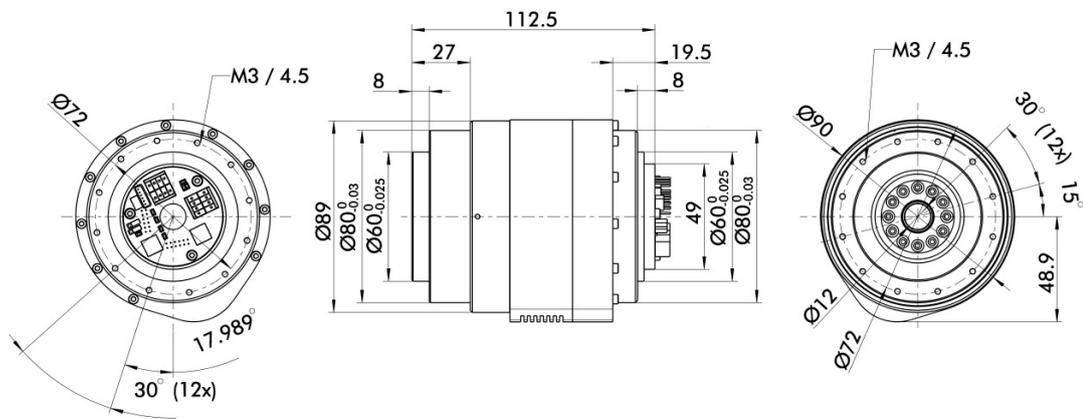


Figura 69: Dimensões do motor PRL80.

B.3 Dados Técnicos

Tabela 6: Tabela de parâmetros do motor PRL80 da Schunk.

Item	Valor
Potência[W]	80
Tipo	Brushless DC Servo Motor
Freios	Integrado
Torque Nominal[Nm]	20.7
Torque Máximo[Nm]	41.4
Angulo(>)[graus]	> 360.00
Peso[Kg]	1.2
Velocidade angular Máxima[graus]	25
Aceleração Máxima[graus]	100
Transmissão	552:1

APÊNDICE C URDF EXEMPLO

Este apêndice descreve a cadeia cinemática que representa a estrutura em URDF da Figura 20.

C.1 Cadeia cinemática de um robô genérico do tipo árvore

Listagem C.1: URDF para a cadeia cinemática da Figura 20.

```
1 <?xml version="1.0"?>
2 <robot name="robot_generic">
3
4   <link name="link1">
5     <inertial>
6       <origin xyz="0 0 0" rpy="0 0 0"/>
7       <mass value="1.0" />
8       <inertia ixx="1.0" ixy="0.0" ixz="0.0"
9         iyy="1.0" iyz="0.0" izz="1.0"/>
10    </inertial>
11  </link>
12
13  <link name="link2">
14    <inertial>
15      <origin xyz="0 0 0" rpy="0 0 0"/>
16      <mass value="1.0" />
17      <inertia ixx="1.0" ixy="0.0" ixz="0.0"
18        iyy="1.0" iyz="0.0" izz="1.0"/>
19    </inertial>
20  </link>
21
22  <link name="link3">
23    <inertial>
24      <origin xyz="0 0 0" rpy="0 0 0"/>
25      <mass value="1.0" />
26      <inertia ixx="1.0" ixy="0.0" ixz="0.0"
27        iyy="1.0" iyz="0.0" izz="1.0"/>
```

```

28   </inertial>
29 </link>
30
31 <link name="link4">
32   <inertial>
33     <origin xyz="0 0 0" rpy="0 0 0"/>
34     <mass value="1.0" />
35     <inertia ixx="1.0" ixy="0.0" ixz="0.0"
36       iyy="1.0" iyz="0.0" izz="1.0"/>
37   </inertial>
38 </link>
39
40 <joint name="joint1" type="continuous">
41   <parent link="link1"/>
42   <child link="link2"/>
43   <origin xyz="5 3 0" rpy="0 0 0" />
44   <axis xyz="-0.9 0.15 0" />
45 </joint>
46
47 <joint name="joint2" type="continuous">
48   <parent link="link1"/>
49   <child link="link3"/>
50   <origin xyz="-2 5 0" rpy="0 0 1.57" />
51   <axis xyz="-0.707 0.707 0" />
52 </joint>
53
54 <joint name="joint3" type="continuous">
55   <parent link="link3"/>
56   <child link="link4"/>
57   <origin xyz="5 0 0" rpy="0 0 -1.57" />
58   <axis xyz="0.707 -0.707 0" />
59 </joint>
60
61 <transmission name="trans1">
62   <type>transmission_interface/SimpleTransmission</type>
63   <joint name="joint1"/>
64 <actuator name="motor1">
65   <hardwareInterface>EffortJointInterface</hardwareInterface>
66   <mechanicalReduction>1</mechanicalReduction>
67 </actuator>
68 </transmission>
69
70 <transmission name="trans2">
71   <type>transmission_interface/SimpleTransmission</type>
72   <joint name="joint2"/>
73 <actuator name="motor2">
74   <hardwareInterface>EffortJointInterface</hardwareInterface>
75   <mechanicalReduction>1</mechanicalReduction>
76 </actuator>

```

```
77 </transmission>
78
79 <transmission name="trans3">
80   <type>transmission_interface/SimpleTransmission</type>
81   <joint name="joint3"/>
82   <actuator name="motor3">
83     <hardwareInterface>EffortJointInterface</hardwareInterface>
84     <mechanicalReduction>1</mechanicalReduction>
85   </actuator>
86 </transmission>
87
88 </robot>
```

APÊNDICE D ÁRVORE DE DIRETÓRIOS

Este apêndice apresenta a árvore de diretórios que fazem parte do pacote desenvolvido para o robô bípede Orbit no ROS.

