

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

TAISER TADEU TEIXEIRA BARROS

**MODELAGEM E IMPLEMENTAÇÃO NO
ROS DE UM CONTROLADOR PARA
MANIPULADORES MÓVEIS**

Porto Alegre
2014

TAISER TADEU TEIXEIRA BARROS

**MODELAGEM E IMPLEMENTAÇÃO NO
ROS DE UM CONTROLADOR PARA
MANIPULADORES MÓVEIS**

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. Walter Fetter Lages

Porto Alegre
2014

TAISER TADEU TEIXEIRA BARROS

**MODELAGEM E IMPLEMENTAÇÃO NO
ROS DE UM CONTROLADOR PARA
MANIPULADORES MÓVEIS**

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. Walter Fetter Lages, UFRGS
Doutor pelo Instituto Tecnológico de Aeronáutica (ITA) – São José dos Campos, SP, Brasil

Banca Examinadora:

Prof. Dr. Edson Prestes e Silva Junior, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof^a. Dr^a. Lucíola Campestrini, UFRGS
Doutora pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade Federal de Minas Gerais – Belo Horizonte, Brasil

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

Porto Alegre, maio de 2014.

DEDICATÓRIA

Dedico este trabalho aos meus pais, ao meu filho Matheus e à minha esposa Fabiane!
Amo vocês!

AGRADECIMENTOS

Agradeço aos meus pais por sempre prezarem por minha educação e formação!

À minha esposa por ser uma verdadeira companheira de todas as horas e por sua compreensão durante toda a jornada de meu mestrado!

Ao meu orientador Professor Doutor Walter Fetter Lages por todo o conhecimento que me foi passado e pelo apoio para realização deste trabalho!

Aos meus cunhados "dindo" Alcinei e "dinda" Mosa por todas as vezes que me hospedaram em sua casa, me tratando como um filho em todas as situações em que precisei ficar em Porto Alegre!

Ao colega Eduardo Henrique Maciel pelo auxílio com o ROS e pela parceria!

Ao Programa de Pós Graduação em Engenharia Elétrica da UFRGS por proporcionar formação de tamanha qualidade!

Ao SENAI pelo apoio financeiro nas apresentações de trabalhos e pela liberação de horários para frequentar o mestrado. E também aos colegas do SENAI por sempre estarem dispostos a me substituir e assim permitir minha ausência da escola nos momentos em que precisei!

That which does not kill us makes us stronger.
Friedrich Nietzsche

RESUMO

Este trabalho apresenta a modelagem matemática para um manipulador móvel composto por uma base móvel (o robô móvel Twil) e um manipulador (o manipulador WAM da Barrett). Os modelos cinemático e dinâmico para a base móvel, manipulador e manipulador móvel são apresentados.

Como o manipulador móvel é um sistema não linear, uma estratégia de controle baseada em linearização por realimentação da dinâmica da plataforma seguida por uma transformação não suave para tratar a não holonomicidade do modelo cinemático é proposta. Então o método de *backstepping* é utilizado para obter as entradas do modelo dinâmico. Um controlador de torque calculado é proposto para o manipulador, Estas técnicas de controle são utilizadas simultaneamente para controlar o manipulador móvel.

A implementação dos controladores propostos, na forma de *plugins* para o gerenciador de controladores é feita no ROS, assim os controladores são executados em tempo real. A maioria dos controladores existentes no ROS são do tipo SISO baseados em controle PID e independentes para cada junta, sendo que neste trabalho controladores MIMO não lineares são implementados.

Palavras-chave: Manipuladores móveis, Controlador não suave, Linearização por realimentação, Torque calculado, ROS.

ABSTRACT

This work presents a mathematical modelling for a mobile manipulator composed by a mobile base (the Twil mobile robot) and a manipulator (the Barrett WAM manipulator). The kinematic and dynamic models for the mobile base, the manipulator and the mobile manipulator are presented.

As the the mobilie manipulator is a non-linear system, a control strategy based on feedback linearization of the platform dynamics followed by a non-smooth transform to handle the non-holonomicity of its kinematic model is proposed. Then, the backstepping method is used to obtain the inputs for the dynamic model. A computed torque controller is proposed for the manipulator. These control techniques are used simultaneously to control the mobile manipulator.

The implementation of the proposed controllers is done in ROS as plugins for the controller manager so that the controllers run in real-time. Most controllers existing in ROS are independent joint SISO controllers based on the PID control law while in this work MIMO non-linear controllers are implemented.

Keywords: Mobile manipulators, Non-smooth controller, Feedback linearization, Computed torque, ROS.

LISTA DE ILUSTRAÇÕES

Figura 1:	Pose do robô em relação ao sistema inercial.	32
Figura 2:	Modelos de rodas universais.	33
Figura 3:	Sistemas de coordenadas das rodas.	34
Figura 4:	Roda orientável não-centrada (BEARINGBOYS, 2013).	35
Figura 5:	Roda esférica ou <i>caster ball</i>	39
Figura 6:	Modelo de roda esférica (BUYHAFELE, 2013).	39
Figura 7:	Robô omnidirecional, fabricante: Nexus Robot, modelo RB-Nex-07, fonte: (ROBOTSHOP, 2013a).	42
Figura 8:	Robô diferencial, fabricante: sgbotic, modelo RB-Das-09, fonte: (ROBOTSHOP, 2013b).	42
Figura 9:	Sistema de direção tipo Ackerman, adaptado de: (RCCRAWLER, 2013).	43
Figura 10:	Estrutura do robô móvel Twil.	52
Figura 11:	Velocidade linear e angular do robô móvel Twil.	54
Figura 12:	Representação em blocos do modelo do robô móvel.	57
Figura 13:	Sistema de coordenadas dos elos e respectivos parâmetros de Denavit e Hartenberg, adaptado de (FU; GONZALES; LEE, 1987).	62
Figura 14:	Manipulador Barrett WAM, adaptado de (BARRETT TECHNOLOGY, 2006).	65
Figura 15:	Sistemas de coordenadas do manipulador Barrett WAM, adaptado de BARRETT TECHNOLOGY (2006).	66
Figura 16:	Geometria de intersecção dos volumes das partes inferior e superior do manipulador WAM, adaptado de SINGH; CLAASSENS (2010).	67
Figura 17:	Posição do manipulador WAM com o cotovelo para fora, adaptado de SINGH; CLAASSENS (2010).	70
Figura 18:	Posição do manipulador WAM com o cotovelo para dentro, adaptado de SINGH; CLAASSENS (2010).	70
Figura 19:	Sistema de coordenadas do robô móvel Twil com relação a um sistema de referência.	80
Figura 20:	Diagrama de blocos do controlador a ser implementado para o robô móvel Twil.	87
Figura 21:	Estrutura de um pacote no ROS.	90
Figura 22:	Arquitetura de controladores de baixo nível no ROS.	93
Figura 23:	Laço de tempo real do ROS, adaptada de (MACIEL, 2014).	94
Figura 24:	Edição do modelo 3D do robô móvel Twil no SolidWorks®.	97
Figura 25:	Descrição URDF do robô móvel Twil.	98
Figura 26:	Estrutura do pacote <code>twil</code>	98

Figura 27:	Pacote <code>twil_control_gazebo</code>	99
Figura 28:	Pacote <code>twil_controllers</code>	100
Figura 29:	Pacote <code>twil_description</code>	101
Figura 30:	Pacote <code>twil_ident</code>	101
Figura 31:	Estrutura de diretórios do pacote do manipulador Barrett WAM.	102
Figura 32:	Pacote <code>wam_control_gazebo</code>	102
Figura 33:	Pacote <code>wam_controllers</code>	103
Figura 34:	Pacote <code>wam_description</code>	104
Figura 35:	Robô móvel Twil no Gazebo.	105
Figura 36:	Diagrama de nodos da simulação em malha aberta do robô móvel Twil.	106
Figura 37:	Degrau de torque na roda esquerda.	106
Figura 38:	Posição $x(t) \times y(t)$	107
Figura 39:	$x(t)$ e $y(t)$	107
Figura 40:	Velocidades das rodas.	108
Figura 41:	Degrau de torque na roda direita.	108
Figura 42:	Posição $x(t) \times y(t)$	109
Figura 43:	$x(t)$ e $y(t)$	109
Figura 44:	Velocidades das rodas.	110
Figura 45:	Degrau de torque nas rodas esquerda e direita.	110
Figura 46:	Posição $x(t) \times y(t)$	111
Figura 47:	$x(t)$ e $y(t)$	111
Figura 48:	Velocidades das rodas.	112
Figura 49:	Diagrama de nodos da identificação de parâmetros do robô móvel Twil.	112
Figura 50:	Evolução do valor estimado para K_5	113
Figura 51:	Evolução do valor estimado para K_6	113
Figura 52:	Evolução do valor estimado para K_7	114
Figura 53:	Evolução do valor estimado para K_8	114
Figura 54:	Diagonal da matriz de covariância referente ao parâmetro K_5	115
Figura 55:	Diagonal da matriz de covariância referente ao parâmetro K_6	116
Figura 56:	Diagonal da matriz de covariância referente ao parâmetro K_7	117
Figura 57:	Diagonal da matriz de covariância referente ao parâmetro K_8	118
Figura 58:	Diagrama de nodos da simulação da linearização do robô móvel Twil.	118
Figura 59:	Posições nos eixos X e Y , na simulação da linearização do robô móvel Twil com aceleração linear constante e aceleração angular nula.	119
Figura 60:	Posição $X \times Y$, na simulação da linearização do robô móvel Twil com aceleração linear constante e aceleração angular nula.	119
Figura 61:	Posições nos eixos X e Y , na simulação da linearização do robô móvel Twil com aceleração angular constante e aceleração linear nula.	120
Figura 62:	Posição $X \times Y$, na simulação da linearização do robô móvel Twil com aceleração angular constante e aceleração linear nula.	120
Figura 63:	Posições nos eixos X e Y , na simulação da linearização do robô móvel Twil com aceleração linear e angular constantes.	121
Figura 64:	Posição $X \times Y$, na simulação da linearização do robô móvel Twil com aceleração linear e angular constantes.	121
Figura 65:	Simulação do controlador <i>backstepping</i> - torque nas rodas.	122
Figura 66:	Simulação do controlador <i>backstepping</i> - posição X	123
Figura 67:	Simulação do controlador <i>backstepping</i> - posição Y	123
Figura 68:	Simulação do controlador <i>backstepping</i> - posição $X \times Y$	124

Figura 69:	Simulação do controlador <i>backstepping</i> - orientação.	124
Figura 70:	Simulação do controlador <i>backstepping</i> - η_1, η_2, u_1 e u_2	125
Figura 71:	Diagrama de nodos da simulação do manipulador Barrett WAM.	125
Figura 72:	Manipulador Barrett WAM no Gazebo.	125
Figura 73:	Posição das juntas.	126
Figura 74:	Torque aplicado nas juntas.	126
Figura 75:	Diagrama de blocos do controlador adaptativo proposto para o manipulador móvel.	128

LISTA DE TABELAS

Tabela 1:	Classificação de robôs móveis em função dos graus de mobilidade (δ_m) e dirigibilidade (δ_s).	41
Tabela 2:	Parâmetros das rodas do robô móvel Twil.	51
Tabela 3:	Parâmetros de Denavit e Hartenberg para o manipulador Barrett WAM configurado para sete graus de liberdade.	65
Tabela 4:	Parâmetros identificados e valores das diagonais das matrizes de covariância.	115

LISTA DE ABREVIATURAS

API	<i>Application Programming Interface</i>
CAD	<i>Computer Aided Design</i>
GID	<i>Generalized Inverse Dynamics</i>
IJU	<i>Idealized Joint Unit</i>
MIMO	<i>Multiple-Input-Multiple-Output</i>
OROCOS	<i>Open Robot Control Software</i>
PID	<i>Proporcional-Integral-Derivativo</i>
ROS	<i>Robot Operating System</i>
RPC	<i>Remote Procedure Call</i>
SISO	<i>Single-Input-Single-Output</i>
STL	<i>STereoLithography</i>
URDF	<i>Unified Robot Description Format</i>
WAM	<i>Whole Arm Manipulator</i>
XML	<i>Extensible Markup Language</i>

LISTA DE SÍMBOLOS

a	Distância do centro do robô até o eixo das rodas;
α	Ângulo entre os sistemas do robô e da roda;
b	Distância do centro do robô até o plano central das rodas;
β	Ângulo das rodas orientáveis;
β_c	Ângulo das rodas orientáveis centradas;
β_{nc}	Ângulo das rodas orientáveis não-centradas;
$B(x)$	Dinâmica do modelo cinemático de pose;
c	Distância do centro do robô até o eixo de giro do suporte da roda orientável não centrada;
d	Distância do centro da roda orientável não centrada e seu eixo de giro;
δ_m	Grau de mobilidade de um robô móvel;
δ_s	Grau de dirigibilidade de um robô móvel;
δ_M	Grau de manobrabilidade de um robô móvel;
e	Módulo do erro de posição em coordenadas polares;
γ	Ângulo da direção da velocidade nula do ponto de contato da roda com o solo;
h	Coefficiente de ponderação > 0 ;
I	Momento de inércia;
λ	Coefficiente de ponderação > 0 ;
λ, μ	Coefficientes de Lagrange;
K_i	Constantes do modelo;
N	Número de rodas;
N_f	Número de rodas fixas;
N_c	Número de rodas centradas;
N_o	Número de rodas não-centradas;
N_u	Número de rodas universais;
η	Vetor de entrada do modelo cinemático;

$\{X_0, Y_0, \theta_0\}$	Sistema de coordenadas inercial;
$\{X_c, Y_c, \theta_c\}$	Sistema de coordenadas associado ao robô móvel;
$\{X_w, Y_w, \theta_w\}$	Sistema de coordenadas associado à roda;
x_c, y_c, θ_c	Coordenadas do centro de gravidade do robô móvel;
${}^0\xi_c$	Pose do robô móvel;
iP_j	Posição da origem do sistema $\{j\}$ em relação ao sistema $\{i\}$;
iR_j	Matriz de rotação do sistema de coordenadas $\{j\}$ em relação ao sistema de coordenadas $\{i\}$;
$\hat{\Theta}$	Versor da coordenada Θ ;
x	Variável de estado;
$\rho(x)$	Rank de x ;
φ	Deslocamento angular da roda;
Ψ	Ângulo de X_w em relação a X_c ;
ψ	Ângulo do erro de posição em coordenadas polares;
ζ	Velocidade angular das rodas orientáveis centradas;
T	Energia cinética;
τ_c	Torque aplicado para orientação das rodas centradas;
τ_φ	Torque aplicado para rotação das rodas;
τ_{no}	Torque aplicado para orientação das rodas não centradas;
$Tr A$	Traço da matriz A ;
u	Vetor de entrada do modelo no espaço de estados;
v	Vetor de entradas de controle;
v_l	Velocidade linear;
V	Candidata à função de <i>Lyapunov</i> ;
ν	Entrada do modelo cinemático;
ω	Velocidade angular;

SUMÁRIO

1	INTRODUÇÃO	29
1.1	Manipuladores Móveis	29
1.2	ROS - Robot Operating System	30
1.3	Objetivos	30
1.4	Estrutura da dissertação	30
2	MODELAGEM DO MANIPULADOR MÓVEL	31
2.1	Introdução	31
2.2	Modelo Cinemático e Dinâmico da Base Móvel	31
2.2.1	Tipos de Rodas	32
2.2.2	Modelagem das Rodas Fixas e Orientáveis Centradas	33
2.2.3	Modelagem das Rodas Orientáveis Não-Centradas	36
2.2.4	Modelagem das Rodas Universais	37
2.2.5	Rodas Esféricas	38
2.2.6	Modelo Cinemático de Pose	39
2.2.7	Modelo Cinemático de Configuração	44
2.2.8	Modelo Dinâmico de Configuração	45
2.2.9	Modelo Dinâmico de Pose	51
2.2.10	Modelo do robô móvel Twil	51
2.3	Modelo Cinemático do Manipulador Barrett WAM	61
2.3.1	Notação de Denavit e Hartenberg	62
2.3.2	Modelo Cinemático Direto do Manipulador Barrett WAM	64
2.3.3	Modelo Cinemático Inverso do Manipulador Barrett WAM	65
2.4	Modelo Dinâmico do Manipulador Barrett WAM	73
2.4.1	Velocidades de Juntas	74
2.4.2	Energia Cinética do Manipulador	75
2.4.3	Energia Potencial do Manipulador	76
2.4.4	Equações de Movimento do Manipulador	76
2.5	Modelo Cinemático do Manipulador Móvel	77
3	PROPOSTA DE CONTROLE	79
3.1	Introdução	79
3.2	Estratégias de Controle	79
3.2.1	Linearização por Realimentação da Dinâmica	79
3.2.2	Transformação de coordenadas	79
3.2.3	Controlador não suave	80
3.2.4	<i>Backstepping</i>	82

3.2.5	Identificação de parâmetros	85
3.3	Controlador do robô móvel Twil	87
3.4	Controle do manipulador Barrett WAM	88
4	IMPLEMENTAÇÃO NO ROS (ROBOT OPERATING SYSTEM)	89
4.1	Introdução	89
4.2	ROS	89
4.2.1	Pacotes do ROS	89
4.2.2	<i>Stacks</i> do ROS	90
4.2.3	Nodos do ROS	90
4.2.4	Mecanismos de comunicação entre nodos do ROS	91
4.2.5	Controladores do ROS	91
4.2.6	Laço de tempo real do ROS	93
4.2.7	Unified Robot Description Format - URDF	96
4.3	Pacote do robô móvel Twil	97
4.3.1	Pacote <code>twil_control_gazebo</code>	97
4.3.2	Pacote <code>twil_controllers</code>	99
4.3.3	Pacote <code>twil_description</code>	99
4.3.4	Pacote <code>twil_ident</code>	99
4.4	Pacote do manipulador Barrett WAM	100
4.4.1	Pacote <code>wam_control_gazebo</code>	102
4.4.2	Pacote <code>wam_controllers</code>	102
4.4.3	Pacote <code>wam_description</code>	103
5	EXPERIMENTOS	105
5.1	Introdução	105
5.2	Simulação do robô móvel Twil com controle em malha aberta	105
5.3	Identificação de parâmetros do robô móvel Twil	112
5.4	Simulação do robô móvel Twil utilizando linearização	115
5.5	Simulação do robô móvel Twil utilizando o controlador <i>backstepping</i>	122
5.6	Simulação do manipulador Barrett WAM	122
6	CONCLUSÕES	127
	REFERÊNCIAS	129
	APÊNDICE A REGRAS DE DERIVAÇÃO DE VETORES E MATRIZES	135
	APÊNDICE B ENERGIA CINÉTICA DA PLATAFORMA MÓVEL	139
B.1	Energia Cinética do Chassi	139
B.2	Energia Cinética de uma Roda Fixa	139
B.3	Energia Cinética de uma Roda Não Centrada	140
B.4	Energia Cinética Total	142
	APÊNDICE C DERIVADAS DA ENERGIA CINÉTICA DA PLATAFORMA MÓVEL	145
C.1	Derivadas da energia cinética em relação a ξ e $\dot{\xi}$	145
C.2	Derivadas da energia cinética em relação a β_o e $\dot{\beta}_o$	149
C.3	Derivadas da energia cinética em relação a φ e $\dot{\varphi}$	151
C.4	Resumo das Expressões	151

APÊNDICE D	DERIVADAS DE SEGUNDA ORDEM DAS VARIÁVEIS ${}^0\xi_C,$ β_0 E φ	153
APÊNDICE E	DESENHOS DO ROBÔ MÓVEL TWIL	155
APÊNDICE F	MATRIZ DE ROTAÇÃO QUE RELACIONA DOIS PONTOS	157

1 INTRODUÇÃO

1.1 Manipuladores Móveis

Um manipulador móvel é constituído basicamente por um manipulador montado sobre uma base móvel, com as pesquisas acerca deste tipo de sistema concentrando-se principalmente sobre controle de movimento envolvendo sua cinemática e dinâmica (MAILAH; PITOWARNO; JAMALUDDIN, 2005). Os manipuladores convencionais, que possuem uma base fixa, são caracterizados por possuírem um espaço de trabalho limitado. Desta forma, a idéia de acoplar os mesmos a uma plataforma móvel com rodas propicia um ganho na mobilidade do conjunto e conseqüentemente um aumento espaço de trabalho. Devido a sua maior mobilidade e habilidade, os manipuladores móveis tornaram-se tópicos de pesquisa (YU; CHEN, 2002), onde a integração entre o manipulador e o controle do movimento é o principal objetivo de estudo (FUKUSHIMA; HIROSE; HAYASHI, 1998; YAMAMOTO; YUN, 1999).

Ainda que a teoria sobre manipuladores móveis já seja objetivo de pesquisa a mais de uma década, trabalhos recentes mostram que o assunto possui uma ampla gama de possibilidades de pesquisa, assim como pode-se observar em IDE et al. (2011), onde um sistema de planejamento de trajetórias em tempo real é formulado como problema de Programação Quadrática para otimizar as entradas de controle e em HAMNER et al. (2010), que apresenta um manipulador móvel autônomo desenvolvido para realizar tarefas em um ambiente de manufatura da linha automotiva empregando controle de coordenadas junto com sistemas de visão e força para solucionar incertezas inerentes ao processo.

Considerar a manipulação e o movimento do manipulador móvel como problemas individuais torna a tarefa de criar o controle do sistema mais fácil (YAMAMOTO; YUN, 1999), porém uma tendência mostrada em trabalhos recentes, é tratar os manipuladores móveis em uma perspectiva de controlar todo o corpo do robô (*Whole Body Control*) como um único manipulador. Em NAGASAKA et al. (2010), propõe-se duas técnicas de controle: a GID (*Generalized Inverse Dynamics*) considerada uma forma efetiva de controle, que conforme resultado de simulações propicia aumento na capacidade da interação física de um robô humanóide, e a IJU (*Idealized Joint Unit*) que permite uma resposta mais precisa em relação ao torque. A aplicação destas técnicas tem por objetivo controlar o manipulador móvel considerando o mesmo como um único manipulador frente às suas restrições não holonômicas.

Um trabalho que demonstra a complexidade do assunto manipuladores móveis quando considerando os mesmos como uma única entidade é apresentado em (DIETRICH et al., 2012) onde é proposto o controle para um robô humanóide com 51 graus de liberdade, com restrições de projeto tais como evitar a colisão do robô com o próprio corpo e com objetos presentes no ambiente de trabalho.

1.2 ROS - Robot Operating System

ROS é um sistema baseado em mensagens, sendo composto por bibliotecas escritas em C++ e Python. A filosofia de criação do ROS é baseada no sistema operacional UNIX onde uma série de pequenas ferramentas são projetadas para trabalhar em conjunto. A origem do ROS deve-se a uma parceria entre a Indústria e Universidades (COUSINS, 2010). O ROS pode ser compreendido como um Sistema Distribuído (WONG, 2011), onde um determinado número de nós se comunica através de mensagens. Cada nó da rede pode operar em modo cliente/servidor, que permite interação direta entre os nós, ou ainda publicando/assinando mensagens, modo que é normalmente utilizado para uso com sensores.

Alguns trabalhos que utilizam o ROS incluem: a verificação da viabilidade de um algoritmo desenvolvido para planejamento do movimento de um robô móvel com chasis ajustável (BRUNNER; BRÜGGEMANN; SCHULZ, 2012), o desenvolvimento do controle de um robô móvel capaz de navegar de forma autônoma (ZAMAN; SLANY; STEINBAUER, 2011), a localização de objetos por robôs autônomos utilizando RFID (Radio Frequency Identification) (GONG et al., 2012) e a interface de sistemas para teste de um Veículo Autônomo Subaquático (DEMARCO; WEST; COLLINS, 2011).

1.3 Objetivos

Este trabalho tem como objetivo servir como referência na modelagem matemática de manipuladores móveis, principalmente na modelagem da base móvel, além de desenvolver um controlador para a base móvel e para o manipulador, com as respectivas implementações no laço de tempo real do ROS.

1.4 Estrutura da dissertação

Este trabalho está dividido em seis capítulos, da seguinte forma: Introdução. Capítulo 2 com o desenvolvimento matemático para obtenção dos modelos cinemáticos e dinâmicos. Capítulo 3 com a proposta do controlador para o manipulador móvel. Capítulo 4 com a proposta de implementação dos modelos e do controlador no ROS. Capítulo 5 com os resultados de simulação dos pacotes desenvolvidos no ROS. Capítulo 6 com as conclusões e sugestões para trabalhos futuros.

2 MODELAGEM DO MANIPULADOR MÓVEL

2.1 Introdução

Neste capítulo serão apresentados os modelos cinemático e dinâmico da base móvel e do manipulador que constituem o manipulador móvel que é objeto de estudo desta dissertação (além dos modelos cinemático e dinâmico do manipulador móvel). O desenvolvimento dos modelos cinemático e dinâmico da base móvel estão baseados no robô móvel TWIL e os modelos do manipulador estão baseados no WAM (Whole Arm Manipulator) da Barrett.

2.2 Modelo Cinemático e Dinâmico da Base Móvel

Diversos trabalhos contemplam a modelagem cinemática e dinâmica de robôs móveis (SICILIANO; KHATIB, 2008; SIDEK; SARKAR, 2008; GARCÍA, 2009; PETROV, 2010; DUŠEK; HONC; ROZSÍVAL, 2011), sendo que esta seção da dissertação vai seguir uma formulação semelhante à apresentada em (CAMPION; BASTIN; D'ANDRÉA-NOVEL, 1993; SICILIANO; KHATIB, 2008), a qual fornece uma descrição geral para robôs móveis com rodas, apresentando propriedades relativas à sua mobilidade.

Tal formulação supõe que as rodas do robô satisfazem a condição de rolar sem deslizar, ou seja, que a velocidade do ponto de contato da roda com o solo é nula. Pode-se fazer tal suposição uma vez que para o modelo desenvolvido, efeitos dinâmicos rápidos como os do deslizamento das rodas podem ser desconsiderados frente ao tempo dos efeitos dinâmicos do movimento global de toda a estrutura do robô móvel (SICILIANO; KHATIB, 2008).

Assume-se que o plano das rodas permanece na vertical durante a movimentação, com a rotação se dando em torno de um eixo horizontal cuja orientação pode ser fixa ou variável com relação a um sistema de coordenadas associado ao corpo do robô, $\{X_c, Y_c, \Theta_c\}$, descrito em relação ao sistema inercial $\{X_0, Y_0, \Theta_0\}$ como representado na Figura 1, com θ_c sendo o ângulo que define a orientação do robô.

A partir da Figura 1 pode-se definir o vetor ${}^0\xi_c$ que contém as coordenadas da pose do robô e a matriz 0R_c que expressa a orientação do robô em relação ao sistema inercial $\{X_0, Y_0, \Theta_0\}$.

$${}^0\xi_c = \begin{bmatrix} x_c \\ y_c \\ \theta_c \end{bmatrix} \quad (1)$$

$${}^0R_c = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) & 0 \\ \sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

A velocidade do robô descrita em relação ao sistema inercial pode ser dada por:

$${}^0\dot{\xi}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} \quad (3)$$

e descrita no sistema $\{X_0, Y_0, \Theta_0\}$ por:

$${}^c\dot{\xi}_c = {}^cR_0 {}^0\dot{\xi}_c \quad (4)$$

Com cR_0 dado por:

$${}^cR_0 = {}^0R_c^{-1} = {}^0R_c^T = \begin{bmatrix} \cos(\theta_c) & \sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

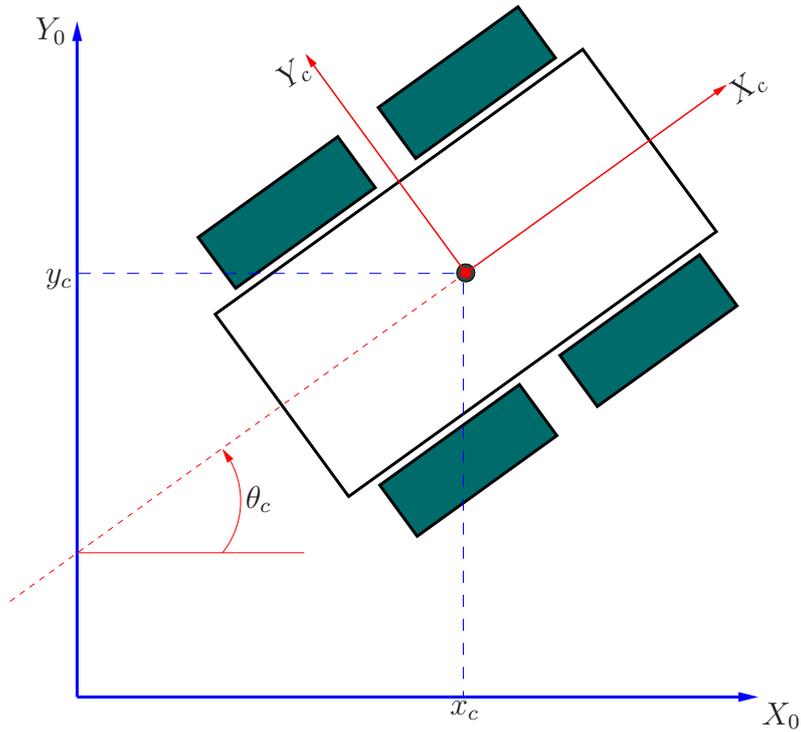


Figura 1: Pose do robô em relação ao sistema inercial.

2.2.1 Tipos de Rodas

Dois tipos de rodas são considerados neste desenvolvimento: rodas convencionais (assim como um pneu comum) e rodas universais (também conhecidas como rodas suecas, as quais incluem em sua estrutura roletes ou esferas que permitem a movimentação da roda em mais de uma direção). A Figura 2 apresenta alguns modelos de rodas universais disponíveis comercialmente. Destaca-se o fato de que a roda universal apresentada na Figura 2(a) é um caso particular da roda universal da Figura 2(b), onde o ângulo da

direção da velocidade nula do ponto de contato da roda com o solo é zero. Este ângulo é representado pelo parâmetro $\gamma \neq \pi/2$. Em situações onde $\gamma = \pi/2$ a roda universal passa a se comportar como uma roda convencional já que vai estar sujeita a uma restrição de não escorregamento.



(a) Roda universal ($\gamma = 0$), fabricante Vex Robotics, modelo 2.75" double roller, fonte: (VEXROBOTICS, 2013).



(b) Roda universal ($\gamma \neq 0$), fabricante: Fingertech Robotics, modelo CG wheel, fonte: (FINGERTECHROBOTICS, 2013).

Figura 2: Modelos de rodas universais.

Quatro casos particulares de rodas serão considerados, as rodas fixas, as rodas orientáveis centradas, as rodas orientáveis não-centradas e as rodas universais. A Figura 3 mostra o sistema de coordenadas definido para cada um dos tipos de rodas mencionado. A Figura 3(a) representa tanto as rodas fixas como as rodas orientáveis centradas, diferenciando-se as mesmas apenas pelo fato de que para as rodas orientáveis centradas β é variável, ou seja, o ângulo de orientação do plano da roda pode mudar durante a movimentação.

Também destaca-se a diferença fundamental entre uma roda orientável centrada e uma não-centrada, que reside no fato de que a roda orientável centrada precisa de algum elemento (um atuador elétrico por exemplo) que modifique seu ângulo β , enquanto que as rodas orientáveis não-centradas, podem se posicionar naturalmente na direção do movimento. Para cada roda foi associado um sistema $\{X_w, Y_w, \Theta_w\}$, com o eixo X_w apontando na direção do movimento da roda. A Figura 4 mostra uma roda orientável não-centrada disponível comercialmente.

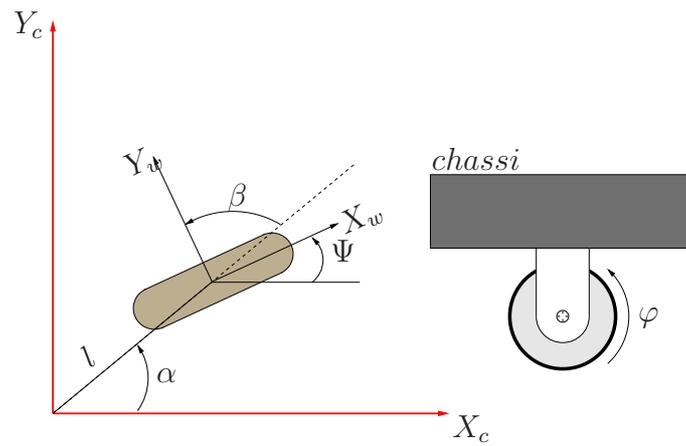
2.2.2 Modelagem das Rodas Fixas e Orientáveis Centradas

A velocidade do sistema de coordenadas $\{X_w, Y_w, \Theta_w\}$ com relação ao sistema inercial $\{X_0, Y_0, \Theta_0\}$ descrita no sistema $\{X_c, Y_c, \Theta_c\}$ é dada por LAGES (1998):

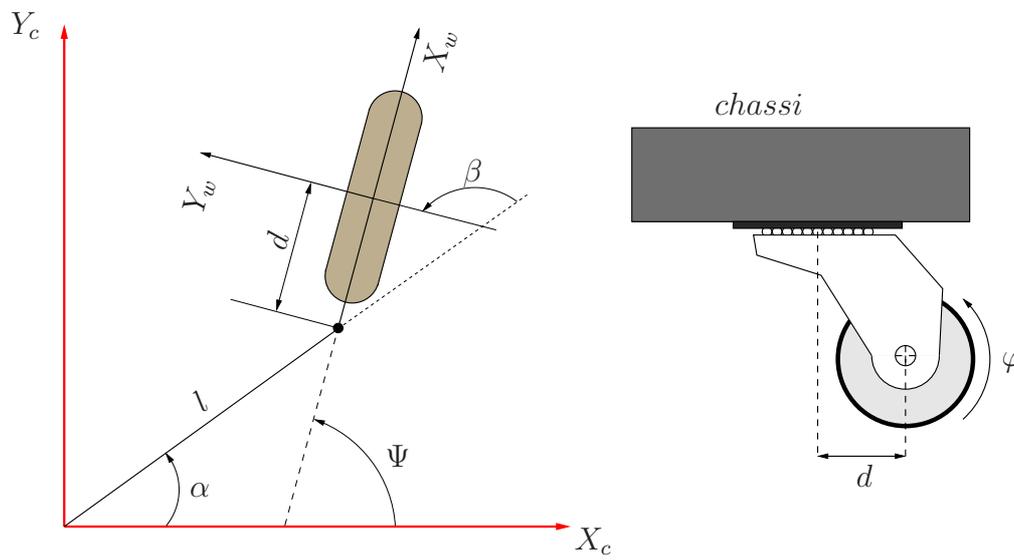
$${}^c\dot{\xi}_w = {}^c\dot{\xi}_c + {}^c\hat{\Theta}_c {}^c\xi_c^T {}^c\hat{\Theta}_c \times {}^cP_w \quad (6)$$

onde \times representa a operação de produto vetorial, e:

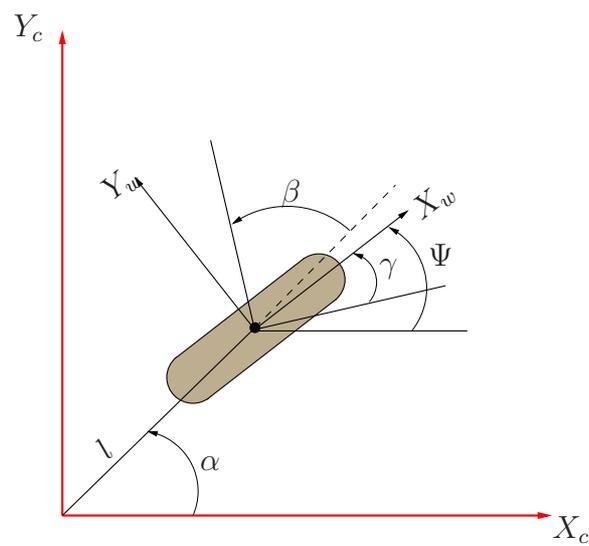
$${}^cP_w = \begin{bmatrix} l \cos \alpha \\ l \sin \alpha \\ \Psi \end{bmatrix} \quad (7)$$



(a) Roda fixa e orientável centrada.



(b) Roda orientável não-centrada.



(c) Roda universal.

Figura 3: Sistemas de coordenadas das rodas.



Figura 4: Roda orientável não-centrada (BEARINGBOYS, 2013).

é a posição da origem do sistema $\{X_w, Y_w, \Theta_w\}$ em relação a $\{X_c, Y_c, \Theta_c\}$ e ${}^c\hat{\Theta}_c$ é o versor da coordenada Θ representado no sistema $\{X_c, Y_c, \Theta_c\}$. Pode-se reescrever (6) no sistema $\{X_w, Y_w, \Theta_w\}$ associada ao centro da roda:

$${}^w\dot{\xi}_w = {}^wR_c {}^c\dot{\xi}_c + {}^wR_c {}^c\hat{\Theta}_c {}^c\dot{\xi}_c^T {}^c\hat{\Theta}_c \times {}^cP_w \quad (8)$$

onde wR_c é a matriz de rotação entre $\{X_w, Y_w, \Theta_w\}$ e $\{X_c, Y_c, \Theta_c\}$ dada por

$${}^wR_c = \begin{bmatrix} \cos \Psi & \sin \Psi & 0 \\ -\sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Ainda, pode-se reescrever (8) na forma:

$${}^w\dot{\xi}_w = {}^wR_c {}^c\dot{\xi}_c + {}^c\hat{\Theta}_c {}^c\dot{\xi}_c^T {}^c\hat{\Theta}_c \times {}^wR_c {}^cP_w \quad (10)$$

Uma vez que $\Psi = \alpha + \beta - \pi/2$ (vide Figura 3(a)) wR_c pode ser reescrito como:

$${}^wR_c = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

e utilizando-se (4) e substituindo wR_c e cP_w , pode-se reescrever (10) na forma:

$${}^w\dot{\xi}_w = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0 {}^c\dot{\xi}_c + {}^c\hat{\Theta}_c {}^cR_0 {}^c\dot{\xi}_c {}^c\hat{\Theta}_c \times \begin{bmatrix} l \sin \beta \\ l \cos \beta \\ \Psi \end{bmatrix} \quad (12)$$

O segundo termo de (12) pode ser escrito como:

$$\begin{aligned} {}^c\hat{\Theta}_c {}^cR_0 {}^c\dot{\xi}_c {}^c\hat{\Theta}_c \times \begin{bmatrix} l \sin \beta \\ l \cos \beta \\ \Psi \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0 {}^c\dot{\xi}_c \times \begin{bmatrix} l \sin \beta \\ l \cos \beta \\ \Psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_c \end{bmatrix} \times \begin{bmatrix} l \sin \beta \\ l \cos \beta \\ \Psi \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & -l \cos \beta \\ 0 & 0 & l \sin \beta \\ 0 & 0 & 0 \end{bmatrix} {}^cR_0 {}^c\dot{\xi}_c \end{aligned} \quad (13)$$

assim tem-se que

$${}^w\dot{\xi}_w = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0 {}^c\dot{\xi}_c \quad (14)$$

Como ${}^w\dot{\xi}_w = [r\dot{\varphi} \ 0 \ \dot{\theta}_c]^T$, pode-se obter:

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0^0 \dot{\xi}_c + \begin{bmatrix} -r\dot{\varphi} \\ 0 \\ -\dot{\theta}_c \end{bmatrix} = 0 \quad (15)$$

que representa as restrições do robô devido à utilização de rodas fixas ou orientáveis centradas. Pode-se ainda desmembrar (15) em:

$$[-\sin(\alpha + \beta) \ \cos(\alpha + \beta) \ l \cos \beta] {}^cR_0^0 \dot{\xi}_c + r\dot{\varphi} = 0 \quad (16)$$

$$[\cos(\alpha + \beta) \ \sin(\alpha + \beta) \ l \sin \beta] {}^cR_0^0 \dot{\xi}_c = 0 \quad (17)$$

que representam as restrições de movimento ao longo do plano da roda e ortogonal ao plano da roda respectivamente.

2.2.3 Modelagem das Rodas Orientáveis Não-Centradas

As rodas orientáveis não centradas (vide Figura 3(b)) caracterizam-se pelo eixo vertical de rotação do seu plano não passando pelo seu centro. Assim tem-se que a posição da origem do sistema $\{X_w, Y_w, \Theta_w\}$ em relação a $\{X_c, Y_c, \Theta_c\}$ é dada por:

$${}^cP_w = \begin{bmatrix} l \cos \alpha + d \cos \Psi \\ l \sin \alpha + d \sin \Psi \\ \Psi \end{bmatrix} = \begin{bmatrix} l \cos \alpha + d \sin(\alpha + \beta) \\ l \sin \alpha - d \cos(\alpha + \beta) \\ \Psi \end{bmatrix} \quad (18)$$

e que a velocidade do sistema $\{X_w, Y_w, \Theta_w\}$ em relação ao sistema inercial descrita no sistema de coordenadas $\{X_c, Y_c, \Theta_c\}$ pode ser dada por:

$${}^c\dot{\xi}_w = {}^c\dot{\xi}_c + {}^c\hat{\Theta}_c {}^c\dot{\xi}_c^T {}^c\hat{\Theta}_c \times {}^cP_w + \dot{\beta} {}^c\hat{\Theta}_c + \dot{\beta} {}^c\hat{\Theta}_c \times {}^cP_{dw} \quad (19)$$

onde:

$${}^cP_{dw} = \begin{bmatrix} d \cos \Psi \\ d \sin \Psi \\ \Psi \end{bmatrix} = \begin{bmatrix} d \sin(\alpha + \beta) \\ d \cos(\alpha + \beta) \\ \Psi \end{bmatrix} \quad (20)$$

é o deslocamento da origem do sistema $\{X_w, Y_w, \Theta_w\}$ em relação ao centro de giro da roda representado no sistema $\{X_c, Y_c, \Theta_c\}$.

Escrevendo (19) no sistema $\{X_w, Y_w, \Theta_w\}$ tem-se:

$${}^w\dot{\xi}_w = {}^wR_c {}^c\dot{\xi}_c + {}^wR_c {}^c\hat{\Theta}_c {}^c\dot{\xi}_c^T {}^c\hat{\Theta}_c \times {}^cP_w + \dot{\beta} {}^c\hat{\Theta}_c \times {}^cP_{dw} + {}^wR_c {}^c\hat{\Theta}_c \times {}^cP_{dw} \dot{\beta} \quad (21)$$

ou

$${}^w\dot{\xi}_w = {}^wR_c {}^c\dot{\xi}_c + {}^c\hat{\Theta}_c {}^c\dot{\xi}_c^T {}^c\hat{\Theta}_c \times {}^wR_c {}^cP_w + \dot{\beta} {}^c\hat{\Theta}_c \times {}^cP_{dw} + {}^c\hat{\Theta}_c \times {}^wR_c {}^cP_{dw} \dot{\beta} \quad (22)$$

E, utilizando-se (4) e substituindo cP_w , wR_c e ${}^cP_{dw}$, (22) pode ser reescrita como:

$$\begin{aligned}
{}^w \dot{\xi}_w &= \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + {}^c \hat{\Theta}_c {}^c R_0^0 \dot{\xi}_c {}^c \hat{\Theta}_c \times \begin{bmatrix} l \sin \beta + d \\ l \cos \beta \\ \Psi \end{bmatrix} \\
&+ \dot{\beta} {}^c \hat{\Theta}_c + {}^c \hat{\Theta}_c \times \begin{bmatrix} d \\ 0 \\ \Psi \end{bmatrix} \dot{\beta}
\end{aligned} \quad (23)$$

Os três últimos termos de (23) podem ser escritos na forma:

$$\begin{aligned}
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0^0 \dot{\xi}_c \times \begin{bmatrix} l \sin \beta + d \\ l \cos \beta \\ \Psi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d \dot{\beta} \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_c \end{bmatrix} \times \begin{bmatrix} l \sin \beta + d \\ l \cos \beta \\ \Psi \end{bmatrix} + \\
\begin{bmatrix} 0 \\ d \dot{\beta} \\ \dot{\beta} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & -l \cos \beta \\ 0 & 0 & l \sin \beta + d \\ 0 & 0 & 0 \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + \begin{bmatrix} 0 \\ d \dot{\beta} \\ \dot{\beta} \end{bmatrix}
\end{aligned} \quad (24)$$

tem-se que

$${}^w \dot{\xi}_w = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta + d \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + \begin{bmatrix} 0 \\ d \dot{\beta} \\ \dot{\beta} \end{bmatrix} \quad (25)$$

e como ${}^w \dot{\xi}_w = [r\dot{\varphi} \ 0 \ \dot{\theta}_c + \dot{\beta}]^T$ pode-se escrever:

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta + d \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + \begin{bmatrix} 0 \\ d \dot{\beta} \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} -r\dot{\varphi} \\ 0 \\ -\dot{\theta}_c - \dot{\beta} \end{bmatrix} = 0 \quad (26)$$

que representa as restrições do robô devido à utilização de rodas orientáveis não centradas. Esta expressão pode ser desmembrada em:

$$[-\sin(\alpha + \beta) \ \cos(\alpha + \beta) \ l \cos \beta] {}^c R_0^0 \dot{\xi}_c + r\dot{\varphi} = 0 \quad (27)$$

$$[\cos(\alpha + \beta) \ \sin(\alpha + \beta) \ l \sin \beta + d] {}^c R_0^0 \dot{\xi}_c + d\dot{\beta} = 0 \quad (28)$$

que representam respectivamente as restrições de movimento ao longo do plano da roda e ortogonal ao plano da roda respectivamente.

2.2.4 Modelagem das Rodas Universais

Para este tipo de roda é importante observar que o componente X_w não está mais alinhado com a direção do plano da roda (vide Figura 3(c)) e sim com a direção do componente nulo de velocidade do ponto de contato da roda com o solo. A velocidade do sistema de coordenadas $\{X_w, Y_w, \Theta_w\}$ com relação ao sistema inercial descrita no sistema $\{X_c, Y_c, \Theta_c\}$ é dada por (6), e que representada no sistema $\{X_w, Y_w, \Theta_w\}$ pode ser escrita como:

$${}^w \dot{\xi}_w = {}^w R_c {}^c \dot{\xi}_c + {}^c \hat{\Theta}_c {}^c \dot{\xi}_c^T {}^c \hat{\Theta}_c \times {}^w R_c {}^c P_w \quad (29)$$

com cP_w dado por (7).

A matriz de rotação wR_c entre os sistemas $\{X_w, Y_w, \Theta_w\}$ e $\{X_c, Y_c, \Theta_c\}$ é dada por:

$${}^wR_c = \begin{bmatrix} \sin(\alpha + \beta + \gamma) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta + \gamma) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (30)$$

observando-se que agora $\Psi = \alpha + \beta + \gamma - \pi/2$ e conseqüentemente pode-se escrever:

$${}^w\dot{\xi}_w = \begin{bmatrix} \sin(\alpha + \beta + \gamma) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta + \gamma) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0^0\dot{\xi}_c + {}^c\hat{\Theta}_c {}^cR_0^0\dot{\xi}_c {}^c\hat{\Theta}_c \times \begin{bmatrix} l \sin(\beta + \gamma) \\ l \cos(\beta + \gamma) \\ \Psi \end{bmatrix} \quad (31)$$

O segundo termo de (31) pode ser escrito na forma

$$\begin{aligned} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0^0\dot{\xi}_c \times \begin{bmatrix} l \sin(\beta + \gamma) \\ l \cos(\beta + \gamma) \\ \Psi \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_c \end{bmatrix} \times \begin{bmatrix} l \sin(\beta + \gamma) \\ l \cos(\beta + \gamma) \\ \Psi \end{bmatrix} = \\ \begin{bmatrix} 0 & 0 & -l \cos(\beta + \gamma) \\ 0 & 0 & l \sin(\beta + \gamma) \\ 0 & 0 & 0 \end{bmatrix} {}^cR_0^0\dot{\xi}_c & \quad (32) \end{aligned}$$

e assim:

$${}^w\dot{\xi}_w = \begin{bmatrix} \sin(\alpha + \beta + \gamma) & -\cos(\alpha + \beta + \gamma) & -l \cos(\beta + \gamma) \\ \cos(\alpha + \beta + \gamma) & \sin(\alpha + \beta + \gamma) & l \sin(\beta + \gamma) \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0^0\dot{\xi}_c \quad (33)$$

A velocidade do robô descrita no sistema $\{X_w, Y_w, \Theta_w\}$ é dada por:

$${}^w\dot{\xi}_w = \begin{bmatrix} r\dot{\varphi} \cos \gamma \\ {}^w\dot{\xi}_{wy} \\ \dot{\theta}_c \end{bmatrix} \quad (34)$$

e a restrição do movimento da roda pode ser descrita por:

$$[-\sin(\alpha + \beta + \gamma) \quad \cos(\alpha + \beta + \gamma) \quad l \cos(\beta + \gamma)] {}^cR_0^0\dot{\xi}_c + r\dot{\varphi} \cos \gamma = 0 \quad (35)$$

Deve-se observar que neste caso a componente normal de velocidade ${}^w\dot{\xi}_{wy}$ é desconhecida uma vez que a velocidade do ponto de contato nesta direção não é nula.

2.2.5 Rodas Esféricas

Além dos quatro tipos de rodas apresentadas, existem também as rodas esféricas. Uma vez que este tipo de roda não possui um único eixo de rotação, a roda não impõe restrições de movimento ao robô, ou seja, a roda pode movimentar-se em qualquer direção. As expressões que modelam o movimento de uma roda esférica, podem ser obtidas considerando os ângulos indicados na Figura 5, observando que o ângulo de rotação da roda em

relação ao eixo de giro não é restrito a uma única direção podendo assim a roda deslocar-se tanto na direção x como na direção y . Assim, seguindo-se o mesmo desenvolvimento para obter (16) e (17) pode-se escrever para as rodas esféricas:

$${}^w \dot{\xi}_w = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0^0 \dot{\xi}_c$$

observando-se agora que

$${}^w \dot{\xi}_w = \begin{bmatrix} {}^w \dot{\xi}_{wx} \\ {}^w \dot{\xi}_{wy} \\ \dot{\theta}_c \end{bmatrix}$$

ou seja, a roda apresenta componentes de velocidade não nulas, tanto na direção paralela (componente ${}^w \dot{\xi}_{wx}$) como na direção ortogonal (componente ${}^w \dot{\xi}_{wy}$) com relação ao seu ponto de contato com o solo, que não podem ser determinadas. Desta forma, pode-se concluir que as rodas esféricas não geram restrições de movimento e portanto a adição das mesmas no robô não alteram suas características de movimentação. A Figura 6 mostra uma roda esférica disponível comercialmente.

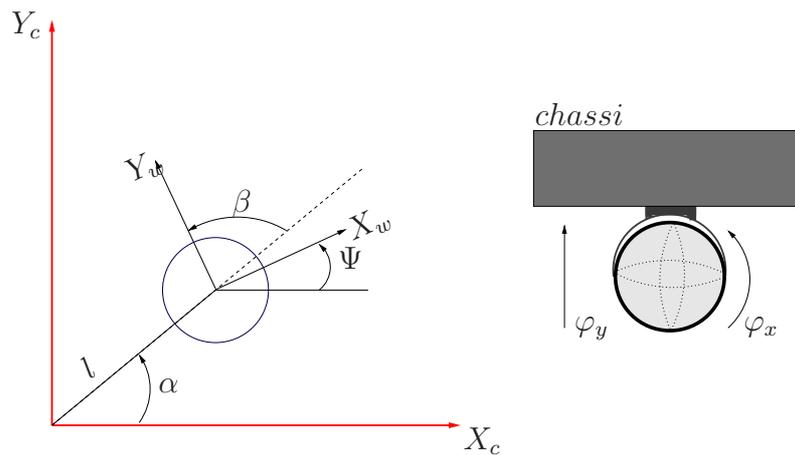


Figura 5: Roda esférica ou *caster ball*.



Figura 6: Modelo de roda esférica (BUYHAFELE, 2013).

2.2.6 Modelo Cinemático de Pose

Considerando-se um robô com um número qualquer de rodas N , sendo N_f rodas fixas, N_c rodas orientáveis centradas, N_o rodas orientáveis não-centradas e N_u rodas universais, pode-se reescrever as restrições (16, 17),(27, 28) e (35) na seguinte forma matricial

para representar as restrições à mobilidade do robô (CAMPION; BASTIN; D'ANDRÉA-NOVEL, 1993):

$$J_1(\beta_c, \beta_o) {}^c R_0^0 \dot{\xi}_c + J_2 \dot{\varphi} = 0 \quad (36)$$

$$C_1(\beta_c, \beta_o) {}^c R_0^0 \dot{\xi}_c + C_2 \dot{\beta}_o = 0 \quad (37)$$

com:

$$J_1(\beta_c, \beta_o) = \begin{bmatrix} J_{1f} \\ J_{1c}(\beta_c) \\ J_{1o}(\beta_o) \\ J_{1u} \end{bmatrix} \quad (38)$$

$$C_1(\beta_c, \beta_o) = \begin{bmatrix} C_{1f} \\ C_{1c}(\beta_c) \\ C_{1o}(\beta_o) \end{bmatrix} \quad (39)$$

$$C_2 = \begin{bmatrix} 0 \\ 0 \\ C_{2o} \end{bmatrix} \quad (40)$$

Os elementos J_{1f} , J_{1c} , J_{1o} e J_{1u} possuem dimensões $N_f \times 3$, $N_c \times 3$, $N_o \times 3$ e $N_u \times 3$ respectivamente, assim como C_{1f} , C_{1c} e C_{1o} possuem dimensões $N_f \times 3$, $N_c \times 3$, $N_o \times 3$. Ainda, J_2 é uma matriz diagonal com o raio das rodas (para rodas universais os raios são multiplicados por $\cos \gamma$) e C_{2o} é uma matriz diagonal com as distâncias d das N_o rodas não-centradas.

Se forem consideradas apenas as restrições referentes às rodas fixas e rodas orientáveis centradas, a partir de (37) pode-se escrever

$$C_1^*(\beta_c) {}^c R_0^0 \dot{\xi}_c = 0 \quad (41)$$

com

$$C_1^*(\beta_c) = \begin{bmatrix} C_{1f} \\ C_{1c}(\beta_c) \end{bmatrix} \quad (42)$$

e verifica-se de (41) que ${}^c R_0^0 \dot{\xi}_c$ pertence ao espaço nulo de $C_1^*(\beta_c)$, bem como as limitações à mobilidade do robô estão relacionadas com $\rho(C_1^*(\beta_c)) = \text{rank}$ de $C_1^*(\beta_c)$.

Para seguir com o desenvolvimento do modelo cinemático de pose da base móvel, utiliza-se as seguintes definições (CAMPION; BASTIN; D'ANDRÉA-NOVEL, 1993):

Definição 1 - Grau de Mobilidade de um robô móvel: é definido como

$$\delta_m = \dim \mathcal{N}(C_1^*(\beta_c)) \quad (43)$$

Definição 2 - Grau de Dirigibilidade de um robô móvel: é definido como o número de rodas orientáveis centradas (N_c) que podem ser orientadas de maneira independente para dirigir o robô.

$$\delta_s = \rho(C_{1c}(\beta_c)) \quad (44)$$

É evidente o fato de que $\rho(C_1^*(\beta_c)) \leq 3$, sendo assim, no caso em que $\rho(C_1^*(\beta_c)) = 3$ o robô não pode se movimentar pois obrigatoriamente ${}^cR_0^0\dot{\xi}_c = 0$ e portanto deve-se ter $\rho(C_1^*(\beta_c)) \leq 2$.

No caso em que $\rho(C_{1f}) = 2$ têm-se que o robô móvel possui no mínimo duas rodas fixas com planos não paralelos. A existência de mais que duas rodas fixas implica que seus eixos concorram para o mesmo centro de rotação instantâneo, e uma vez que este centro de rotação é fixo devido ao não paralelismo do plano das rodas, verifica-se que o único movimento possível do robô é a rotação em torno do centro de rotação instantâneo.

Elimina-se tal limitação impondo a condição $\rho(C_{1f}) \leq 1$ significando que nos casos em que o robô móvel possuir mais de uma roda convencional fixa, as mesmas devem estar posicionadas em um mesmo eixo.

Ainda, $\rho(C_1^*(\beta_c)) \leq \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$, sendo que nos casos onde $\rho(C_1^*(\beta_c)) < \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$ têm-se que as rodas orientáveis centradas estão posicionadas sobre o mesmo eixo das rodas fixas, fato este que impede as rodas orientáveis centradas de atuar na alocação do centro de rotação instantâneo, e assim para evitar esta situação assume-se que $\rho(C_1^*(\beta_c)) = \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$.

Também, para os casos onde o robô possua um número de rodas orientáveis centradas superior a δ_s , o movimento das $N_c - \delta_s$ rodas deve ser coordenado com as demais de maneira a garantir a existência do centro de rotação instantâneo.

A partir destas condições e das definições de (43) e (44) pode-se escrever as seguintes restrições relativas à configuração das rodas do robô móvel:

$$1 \leq \delta_m \leq 3 \quad (45)$$

com o limite inferior indicando que somente os casos onde é possível o movimento do robô são considerados.

$$0 \leq \delta_s \leq 2 \quad (46)$$

Considerando-se (46) o robô pode possuir no máximo duas rodas orientáveis centradas independentes.

$$2 \leq \delta_m + \delta_s \leq 3 \quad (47)$$

Ainda, em (47) verifica-se que o limite superior indica a condição $\rho(C_1^*(\beta_c)) \leq 2$, e assim $\delta_s = 2$, conseqüentemente $\delta_m = 1$. As configurações que permitem apenas rotação em torno de um centro instantâneo fixo ($\delta_m + \delta_s = 1$) não serão consideradas.

Conforme as expressões (45), (46) e (47) a Tabela 1 indica o conjunto dos termos δ_m e δ_s que satisfazem as restrições de configuração das rodas do robô móvel:

Tabela 1: Classificação de robôs móveis em função dos graus de mobilidade (δ_m) e dirigibilidade (δ_s).

Parâmetro	Valor				
δ_m	3	2	2	1	1
δ_s	0	0	1	1	2

Dos valores da Tabela 1, podem-se definir cinco classes de robôs móveis cada uma denominada robô móvel classe(δ_m, δ_s):

Classe (3,0) - Nesta classe os robôs móveis não possuem rodas fixas ou rodas orientáveis centradas ($\rho(C_1^*(\beta_c)) = 0$), assim, podem mover-se instantaneamente com qualquer orientação e em qualquer direção. Este tipo de robô é conhecido como robô omnidirecional. A Figura 7 mostra um exemplo deste tipo de robô disponível comercialmente.



Figura 7: Robô omnidirecional, fabricante: Nexus Robot, modelo RB-Nex-07, fonte: (ROBOTSHOP, 2013a).

Classe (2,0) - A característica desta classe de robôs é a presença de uma ou mais rodas fixas, sendo que a existência de mais de uma roda fixa implica que as mesmas estejam posicionadas em um eixo comum, lembrando que $\rho(C_1^*(\beta_c)) = \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$ e como $\rho(C_{1c}(\beta_c)) = \delta_s = 0$ tem-se obrigatoriamente que $\rho(C_{1f}) = 1$. É comum que os robôs desta classe operem em modo diferencial e exemplos típicos de estrutura similares a esta classe de robôs são a cadeira de rodas e os *Segways*[®]. A Figura 8 mostra um exemplo deste tipo de robô disponível comercialmente. Ainda, este será o tipo de robô utilizado como base móvel nesta dissertação.



Figura 8: Robô diferencial, fabricante: sgbotic, modelo RB-Das-09, fonte: (ROBOTSHOP, 2013b).

Classe (2,1) - Nesta classe os robôs possuem pelo menos uma roda orientável centrada não possuindo rodas fixas, uma vez que $\rho(C_{1f}) = 0$. A existência de mais de uma roda orientável centrada implica que o movimento das mesmas se dê de forma coordenada para

garantir $\rho(C_{1c}(\beta_c)) = \delta_s = 1$.

Classe (1,1) - Nesta classe de robôs tem-se $\rho(C_{1f}) = 1$, e sendo assim, deve-se ter uma roda fixa, ou várias sobre um mesmo eixo comum e uma ou mais rodas orientáveis centradas com o eixo não coincidente com o das rodas fixas. Na existência de várias rodas orientáveis centradas, deve-se garantir que seu movimento seja coordenado para que $\rho(C_{1c}(\beta_c)) = \delta_s = 1$. Um exemplo típico desta classe são os robôs que possuem uma estrutura com direção do tipo Ackerman (tipo carro). Uma estrutura de direção deste tipo está representada na Figura 9.



Figura 9: Sistema de direção tipo Ackerman, adaptado de: (RCCRAWLER, 2013).

Classe (1,2) - Nesta classe os robôs não possuem rodas fixas, uma vez que $\rho(C_{1f}) = 0$, possuindo duas ou mais rodas orientáveis centradas, sendo que neste caso, deve-se garantir que seu movimento seja coordenado para que $\rho(C_{1c}(\beta_c)) = \delta_s = 2$.

Considerando (41) tem-se que ${}^c R_0 {}^0 \dot{\xi}_c$ pertence ao espaço nulo de $C_1^*(\beta_c)$ e como ${}^c R_0^{-1} = {}^c R_0^T = {}^0 R_c$ pode-se escrever:

$${}^0 \dot{\xi}_c = {}^0 R_c \Sigma(\beta_c) \eta \quad (48)$$

com as colunas de $\Sigma(\beta_c)$ formando uma base do espaço nulo de $C_1^*(\beta_c)$.

Ainda, verifica-se facilmente que as dimensões de $\Sigma(\beta_c)$ e η serão sempre $3 \times \delta_m$ e $\delta_m \times 1$ e definindo-se $\zeta = \dot{\beta}_c$, (48) pode ser aumentada:

$$\begin{cases} {}^0 \dot{\xi}_c = {}^0 R_c \Sigma(\beta_c) \eta \\ \dot{\beta}_c = \zeta \end{cases} \quad (49)$$

representando o modelo no espaço de estados do sistema, onde as variáveis de estado são as coordenadas de pose (${}^0 \xi_c$) e as coordenadas angulares (β_c) e as entradas do sistema são η e ζ . Nos casos em que o robô móvel não possuir rodas orientáveis centradas ($N_c = 0$) (48) não precisa ser aumentada uma vez que Σ não depende de β_c .

O modelo cinemático de pose genérico para as cinco classes de robô apresentadas pode ser escrito na forma:

$$\dot{x} = B(x) u \quad (50)$$

onde,

$$x = \begin{cases} {}^0\xi_c, & N_c = 0 \\ \begin{bmatrix} {}^0\xi_c \\ \beta_c \end{bmatrix}, & N_c \geq 0 \end{cases}$$

$$B(x) = \begin{cases} {}^0R_c\Sigma, & N_c = 0 \\ \begin{bmatrix} {}^0R_c\Sigma(\beta_c) & 0 \\ 0 & I \end{bmatrix}, & N_c \geq 0 \end{cases}$$

$$u = \begin{cases} \eta, & N_c = 0 \\ \begin{bmatrix} \eta \\ \zeta \end{bmatrix}, & N_c \geq 0 \end{cases}$$

sendo este modelo de pose no espaço de estados o mais simples, capaz de fornecer uma descrição global do robô do ponto de vista do usuário.

2.2.7 Modelo Cinemático de Configuração

O modelo cinemático de configuração permite uma descrição do comportamento cinemático de todo o robô incluindo todas as variáveis de configuração. Uma vez que o modelo cinemático de pose foi obtido utilizando-se apenas um subconjunto de (36) e (37), que correspondem às restrições (41) impostas pelas rodas fixas e rodas orientáveis centradas, os ângulos β_o e φ não foram considerados.

Assim, o modelo cinemático de configuração utiliza as demais restrições para obter expressões que incluam estas variáveis. A partir de (36) e (37) pode-se escrever:

$$\dot{\beta}_o = -C_{2o}^{-1}C_{1o}(\beta_o) {}^cR_0\dot{\xi} \quad (51)$$

$$\dot{\varphi} = -J_2^{-1}J_1(\beta_c, \beta_o) {}^cR_0\dot{\xi} \quad (52)$$

Definindo-se:

$$D(\beta_o) = -C_{2o}^{-1}C_{1o}(\beta_o) \quad (53)$$

$$E(\beta_c, \beta_o) = -J_2^{-1}J_1(\beta_c, \beta_o) \quad (54)$$

e substituindo ${}^0\dot{\xi}_c$ de (48), pode-se escrever:

$$\dot{\beta}_o = D(\beta_o)\Sigma(\beta_c)\eta \quad (55)$$

$$\dot{\varphi} = E(\beta_c, \beta_o)\Sigma(\beta_c)\eta \quad (56)$$

Dado o vetor de coordenadas de configuração:

$$q = \begin{bmatrix} \xi \\ \beta_c \\ \beta_o \\ \varphi \end{bmatrix} \quad (57)$$

é possível reescrever (49), (55) e (56) como:

$$\dot{q} = S(q) u \quad (58)$$

que é o modelo cinemático de configuração. Onde:

$$S(q) = \begin{bmatrix} {}^0R_c \Sigma(\beta_c) & 0 \\ 0 & I \\ D(\beta_o) \Sigma(\beta_c) & 0 \\ E(\beta_c, \beta_o) \Sigma(\beta_c) & 0 \end{bmatrix} \quad (59)$$

$$u = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (60)$$

2.2.8 Modelo Dinâmico de Configuração

O modelo dinâmico de configuração permite descrever a dinâmica do robô incluindo os torques gerados por atuadores, além de permitir uma análise da configuração destes atuadores.

Os modelos cinemáticos apresentados previamente descrevem o comportamento do robô em função das velocidades das rodas. Porém fisicamente, as variáveis de entrada de um robô móvel são os torques produzidos pelos motores, que por sua vez são considerados nos modelos dinâmicos.

Este modelo será desenvolvido utilizando-se o formalismo de Lagrange, a partir do qual pode-se escrever que a dinâmica de um robô móvel com rodas é dada (LAGES, 1998; SICILIANO; KHATIB, 2008) por:

$$\left(\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\xi}} \right) - \frac{\partial L}{\partial \xi} \right)^T = {}^0R_c J_1^T(\beta_c, \beta_o) \lambda_1 + {}^0R_c C_1^T(\beta_c, \beta_o) \mu_1 \quad (61)$$

$$\left(\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\beta}_o} \right) - \frac{\partial L}{\partial \beta_o} \right)^T = C_2^T \mu_1 + \tau_o \quad (62)$$

$$\left(\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) - \frac{\partial L}{\partial \varphi} \right)^T = J_2^T \lambda_1 + \tau_\varphi \quad (63)$$

onde τ_φ e τ_o representam respectivamente os torques aplicados para rotação das rodas e orientação das rodas não-centradas. Ainda, L é a função de Lagrange dada por:

$$L = T - P \quad (64)$$

onde T é a energia cinética e P é a energia potencial. Uma vez que considera-se que o robô móvel vai se deslocar somente em uma superfície plana no plano horizontal, não há variação de energia potencial e assim pode-se considerar $L = T$ nas expressões relacionadas ao robô móvel. Ainda, λ_1 , μ_1 , são os coeficientes de Lagrange associados respectivamente às restrições representadas por (36) e (37).

Agora, pré multiplicando-se as expressões (61), (62), e (63) pelos termos cR_o , $D^T(\beta_o)$ e $E^T(\beta_c, \beta_o)$ e realizando a soma das expressões resultantes, obtém-se:

$$\begin{aligned}
& {}^c R_0 \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} \right)^T + D^T(\beta_o) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_o} \right) - \frac{\partial T}{\partial \beta_o} \right)^T + \\
& E^T(\beta_c, \beta_o) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\varphi}} \right) - \frac{\partial T}{\partial \varphi} \right)^T = (J_1(\beta_c, \beta_o) + J_2 E(\beta_c, \beta_o))^T \lambda_1 + \\
& (C_1(\beta_c, \beta_o) + C_2 D(\beta_o))^T \mu_1 + D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_{oc}) \tau_\varphi
\end{aligned} \tag{65}$$

Utilizando (53) e (54) e das definições de $D^T(\beta_o)$ e de $E^T(\beta_c, \beta_o)$ pode-se escrever:

$$J_1(\beta_c, \beta_o) + J_2 E(\beta_c, \beta_o) = 0 \tag{66}$$

$$C_1(\beta_c, \beta_o) + C_2 D(\beta_o) = 0 \tag{67}$$

e assim, os coeficientes de Lagrange (λ_1 e μ_1) são multiplicados por 0, sendo eliminados de (65) que resulta em:

$$\begin{aligned}
& {}^c R_0 \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} \right)^T + D^T(\beta_o) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_o} \right) - \frac{\partial T}{\partial \beta_o} \right)^T + \\
& E^T(\beta_c, \beta_o) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\varphi}} \right) - \frac{\partial T}{\partial \varphi} \right)^T = D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_o) \tau_\varphi
\end{aligned} \tag{68}$$

A energia cinética T de um robô móvel é dada por (vide apêndice B):

$$T = \frac{1}{2} {}^0 \dot{\xi}_c^T {}^c R_0^T \left[M(\beta_o) {}^c R_0 {}^0 \dot{\xi}_c + 2V(\beta_o) \dot{\beta}_o \right] + \frac{1}{2} \dot{\beta}_o^T I_\beta \dot{\beta}_o + \frac{1}{2} \dot{\varphi}^T I_\varphi \dot{\varphi} \tag{69}$$

onde $M(\beta_o)$, $V(\beta_o)$, I_β , e I_φ são funções dos parâmetros de massa e inércia dos vários corpos rígidos que constituem o robô.

Substituindo-se (69) em (68), eliminando as velocidades e acelerações $\dot{\xi}$, $\dot{\beta}_o$, $\dot{\varphi}$, $\dot{\beta}_c$, $\ddot{\xi}$, $\ddot{\beta}_o$, $\ddot{\varphi}$, $\ddot{\beta}_c$ através da utilização de (49), (55), e (56) bem como de suas derivadas (vide apêndices C e D) e substituindo os resultados das derivadas da energia cinética (vide apêndice C) em cada um dos três termos do lado esquerdo de (68) pode-se escrever:

$$\begin{aligned}
& {}^c R_0 \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} \right)^T = M(\beta_o) {}^c R_0 {}^0 \ddot{\xi}_c + M(\beta_o) Q^c {}^c R_0 \dot{\theta}_c {}^0 \dot{\xi}_c \\
& + \frac{d}{dt} (M(\beta_o)) {}^c R_0 {}^0 \dot{\xi}_c + Q^T \dot{\theta}_c M(\beta_o) {}^c R_0 {}^0 \dot{\xi}_c \\
& + V(\beta_o) \ddot{\beta}_o + \frac{d}{dt} V(\beta_o) \dot{\beta}_o + Q^T \dot{\theta}_c V(\beta_o) \dot{\beta}_o \\
& - {}^c R_0 K_v^T(\xi) M(\beta_o) {}^c R_0 {}^0 \dot{\xi}_c - {}^c R_0 K_v^T(\xi) V(\beta_o) \dot{\beta}_o \\
& D^T(\beta_o) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_o} \right) - \frac{\partial T}{\partial \beta_o} \right)^T = {}^0 \dot{\xi}_c^T {}^c R_0^T V(\beta_o) + {}^0 \dot{\xi}_c^T \frac{d}{dt} ({}^c R_0^T) V(\beta_o) \\
& + \dot{\beta}_o^T I_\beta - \frac{1}{2} {}^0 \dot{\xi}_c^T {}^c R_0^T G_\beta(f_M) - {}^0 \dot{\xi}_c^T {}^c R_0^T G_\beta(f_V) \\
& + {}^0 \dot{\xi}_c^T {}^c R_0^T \frac{d}{dt} (V(\beta_o))
\end{aligned}$$

$$E^T(\beta_c, \beta_o) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\varphi}} \right) - \frac{\partial T}{\partial \varphi} \right)^T = \ddot{\varphi}^T I_\varphi$$

e realizando a substituição destes resultados em (68) pode-se escrever:

$$\begin{aligned} & M(\beta_o) {}^c R_0 {}^0 \ddot{\xi}_c + M(\beta_o) Q^c R_0 \dot{\theta}_c {}^0 \dot{\xi}_c + \frac{d}{dt} (M(\beta_o)) {}^c R_0 {}^0 \dot{\xi}_c + \\ & Q^T \dot{\theta}_c M(\beta_o) {}^c R_0 {}^0 \dot{\xi}_c + D^T(\beta_o) \frac{d}{dt} (V^T(\beta_o)) {}^c R_0 {}^0 \dot{\xi}_c + V(\beta_o) \ddot{\beta}_o + \\ & Q^T \dot{\theta}_c V(\beta_o) \dot{\beta}_o - {}^c R_0 K_v^T(\xi) M(\beta_o) {}^c R_0 {}^0 \dot{\xi}_c - {}^c R_0 K_v^T(\xi) V(\beta_o) \dot{\beta}_o + \\ & D^T(\beta_o) V^T(\beta_o) {}^c R_0 {}^0 \ddot{\xi}_c + D^T(\beta_o) V^T(\beta_o) Q^c R_0 \dot{\theta}_c {}^0 \dot{\xi}_c + \\ & D^T(\beta_o) I_\beta \ddot{\beta}_o - \frac{1}{2} D^T(\beta_o) G_\beta^T(f_M) {}^c R_0 {}^0 \dot{\xi}_c - D^T(\beta_o) G_\beta^T(f_V) {}^c R_0 {}^0 \dot{\xi}_c \\ & + \frac{d}{dt} V(\beta_o) \dot{\beta}_o + E^T(\beta_c, \beta_o) I_\varphi \ddot{\varphi} = D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_o) \tau_\varphi \end{aligned} \quad (70)$$

De (48) e (55) tem-se que:

$${}^0 \dot{\xi}_c = {}^0 R_{c\Sigma}(\beta_c) \eta = {}^c R_0^T \Sigma(\beta_c) \eta$$

$$\dot{\beta}_o = D(\beta_o) \Sigma(\beta_c) \eta$$

E substituindo-se ${}^0 \dot{\xi}_c$ e $\dot{\beta}_o$ em (70):

$$\begin{aligned} & M(\beta_o) {}^c R_0 {}^0 \ddot{\xi}_c + M(\beta_o) Q^c R_0 \dot{\theta}_c {}^c R_0^T \Sigma(\beta_c) \eta + \frac{d}{dt} (M(\beta_o)) {}^c R_0 {}^c R_0^T \Sigma(\beta_c) \eta + \\ & Q^T \dot{\theta}_c M(\beta_o) {}^c R_0 {}^c R_0^T \Sigma(\beta_c) \eta + V(\beta_o) \ddot{\beta}_o + \frac{d}{dt} V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta + \\ & Q^T \dot{\theta}_c V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta - {}^c R_0 K_v^T(\xi) M(\beta_o) {}^c R_0 {}^c R_0^T \Sigma(\beta_c) \eta - \\ & {}^c R_0 K_v^T(\xi) V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta + D^T(\beta_o) V^T(\beta_o) {}^c R_0 {}^0 \ddot{\xi}_c + \\ & D^T(\beta_o) V^T(\beta_o) Q^c R_0 \dot{\theta}_c {}^c R_0^T \Sigma(\beta_c) \eta + D^T(\beta_o) \frac{d}{dt} (V^T(\beta_o)) {}^c R_0 {}^c R_0^T \Sigma(\beta_c) \eta + \\ & D^T(\beta_o) I_\beta \ddot{\beta}_o - \frac{1}{2} D^T(\beta_o) G_\beta^T(f_M) {}^c R_0 {}^c R_0^T \Sigma(\beta_c) \eta + E^T(\beta_c, \beta_o) I_\varphi \ddot{\varphi} - \\ & D^T(\beta_o) G_\beta^T(f_V) {}^c R_0 {}^c R_0^T \Sigma(\beta_c) \eta = D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_o) \tau_\varphi \end{aligned} \quad (71)$$

Observando-se que ${}^c R_0 {}^c R_0^T$ resulta em uma matriz identidade 3×3 pode reescrever (71) como:

$$\begin{aligned}
& M(\beta_o) {}^c R_0^0 \ddot{\xi}_c + M(\beta_o) Q \dot{\theta}_c \Sigma(\beta_c) \eta + \frac{d}{dt} (M(\beta_o)) \Sigma(\beta_c) \eta + \\
& Q^T \dot{\theta}_c M(\beta_o) \Sigma(\beta_c) \eta + V(\beta_o) \ddot{\beta}_o + \frac{d}{dt} V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta + \\
& Q^T \dot{\theta}_c V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta - {}^c R_0 K_v^T(\xi) M(\beta_o) \Sigma(\beta_c) \eta - \\
& {}^c R_0 K_v^T(\xi) V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta + D^T(\beta_o) V^T(\beta_o) {}^c R_0^0 \ddot{\xi}_c + \\
& D^T(\beta_o) V^T(\beta_o) Q \dot{\theta}_c \Sigma(\beta_c) \eta + D^T(\beta_o) \frac{d}{dt} (V^T(\beta_o)) \Sigma(\beta_c) \eta + \\
& D^T(\beta_o) I_\beta \ddot{\beta}_o - \frac{1}{2} D^T(\beta_o) G_\beta^T(f_M) \Sigma(\beta_c) \eta - D^T(\beta_o) G_\beta^T(f_V) \Sigma(\beta_c) \eta + \\
& E^T(\beta_c, \beta_o) I_\varphi \ddot{\varphi} = D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_o) \tau_\varphi \tag{72}
\end{aligned}$$

Substituindo-se os resultados da derivadas de segunda ordem das variáveis ${}^0 \xi_c$, β_o e φ (vide apêndice D) em (72) e definindo-se $M_1(\beta_o, \dot{\beta}_o) = \frac{d}{dt} (M(\beta_o))$ e $V_1(\beta_o, \dot{\beta}_o) = \frac{d}{dt} (V(\beta_o))$ tem-se:

$$\begin{aligned}
& M(\beta_o) Q^T \dot{\theta}_c \Sigma(\beta_c) \eta + M(\beta_o) f_s(\beta_c, \zeta) \eta + M(\beta_o) \Sigma(\beta_c) \dot{\eta} + \\
& M(\beta_o) Q \dot{\theta}_c \Sigma(\beta_c) \eta + M_1(\beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + Q^T \dot{\theta}_c M(\beta_o) \Sigma(\beta_c) \eta + \\
& V(\beta_o) f_d(\beta_c, \zeta) \Sigma(\beta_c) \eta + V(\beta_o) D(\beta_o) f_s(\beta_c, \zeta) \eta + \\
& V_1(\beta_o, \dot{\beta}_o) D(\beta_o) \Sigma(\beta_c) \eta + {}^c R_0 K_v^T(\xi) V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta + \\
& Q^T \dot{\theta}_c V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta - {}^c R_0 K_v^T(\xi) M(\beta_o) \Sigma(\beta_c) \eta - \\
& D^T(\beta_o) V^T(\beta_o) Q^T \dot{\theta}_c \Sigma(\beta_c) \eta + D^T(\beta_o) V^T(\beta_o) f_s(\beta_c, \zeta) \eta + \\
& D^T(\beta_o) V^T(\beta_o) Q \dot{\theta}_c \Sigma(\beta_c) \eta + D^T(\beta_o) V_1^T(\beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + \\
& D^T(\beta_o) I_\beta f_d(\beta_c, \zeta) \Sigma(\beta_c) \eta + D^T(\beta_o) I_\beta D(\beta_o) f_s(\beta_c, \zeta) \eta + \\
& D^T(\beta_o) V^T(\beta_o) \Sigma(\beta_c) \dot{\eta} + D^T(\beta_o) I_\beta D(\beta_o) \Sigma(\beta_c) \dot{\eta} - \\
& \frac{1}{2} D^T(\beta_o) G_\beta^T(f_M) \Sigma(\beta_c) \eta - D^T(\beta_o) G_\beta^T(f_V) \Sigma(\beta_c) \eta + \\
& E^T(\beta_c, \beta_o) I_\varphi f_e(\beta_c, \zeta, \beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + V(\beta_o) D(\beta_o) \Sigma(\beta_c) \dot{\eta} + \\
& E^T(\beta_c, \beta_o) I_\varphi E(\beta_c, \beta_o) f_s(\beta_c, \zeta) \eta + \\
& E^T(\beta_c, \beta_o) I_\varphi E(\beta_c, \beta_o) \Sigma(\beta_c) \dot{\eta} = D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_o) \tau_\varphi \tag{73}
\end{aligned}$$

Multiplicando-se agora os dois lados de (73) pelo termo $\Sigma^T(\beta_c)$ e definindo-se:

$$\begin{aligned}
f_1(\beta_c, \beta_o, \eta, \zeta) = & \Sigma^T(\beta_c) M(\beta_o) Q^T \dot{\theta}_c \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) M(\beta_o) f_s(\beta_c, \zeta) \eta + \\
& \Sigma^T(\beta_c) M(\beta_o) Q \dot{\theta}_c \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) M_1(\beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + \\
& \Sigma^T(\beta_c) V(\beta_o) f_d(\beta_c, \zeta) \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) V(\beta_o) D(\beta_o) f_s(\beta_c, \zeta) \eta + \\
& \Sigma^T(\beta_c) V_1(\beta_o, \dot{\beta}_o) D(\beta_o) \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) {}^c R_0 K_v^T(\xi) V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta + \\
& \Sigma^T(\beta_c) Q^T \dot{\theta}_c V(\beta_o) D(\beta_o) \Sigma(\beta_c) \eta - \Sigma^T(\beta_c) {}^c R_0 K_v^T(\xi) M(\beta_o) \Sigma(\beta_c) \eta - \\
& \Sigma^T(\beta_c) D^T(\beta_o) V^T(\beta_o) Q^T \dot{\theta}_c \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) D^T(\beta_o) V^T(\beta_o) f_s(\beta_c, \zeta) \eta + \\
& \Sigma^T(\beta_c) D^T(\beta_o) V^T(\beta_o) Q \dot{\theta}_c \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) D^T(\beta_o) V_1^T(\beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + \\
& \Sigma^T(\beta_c) D^T(\beta_o) I_\beta f_d(\beta_c, \zeta) \Sigma(\beta_c) \eta + \Sigma^T(\beta_c) D^T(\beta_o) I_\beta D(\beta_o) f_s(\beta_c, \zeta) \eta - \\
& \frac{1}{2} \Sigma^T(\beta_c) D^T(\beta_o) G_\beta^T(f_M) \Sigma(\beta_c) \eta - \Sigma^T(\beta_c) D^T(\beta_o) G_\beta^T(f_V) \Sigma(\beta_c) \eta + \\
& \Sigma^T(\beta_c) E^T(\beta_c, \beta_o) I_\varphi f_e(\beta_c, \zeta, \beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + \\
& \Sigma^T(\beta_c) E^T(\beta_c, \beta_o) I_\varphi E(\beta_c, \beta_o) f_s(\beta_c, \zeta) \eta + \Sigma^T(\beta_c) Q^T \dot{\theta}_c M(\beta_o) \Sigma(\beta_c) \eta \quad (74)
\end{aligned}$$

e com:

$$\begin{aligned}
H_1(\beta_c, \beta_o) = & \Sigma^T(\beta_c) [M(\beta_o) + D^T(\beta_o) V^T(\beta_o) + V(\beta_o) D(\beta_o) \\
& + D^T(\beta_o) I_o D(\beta_o) + E^T(\beta_c, \beta_o) I_\varphi E(\beta_c, \beta_o)] \Sigma(\beta_c) \quad (75)
\end{aligned}$$

pode-se escrever:

$$H_1(\beta_c, \beta_o) \dot{\eta} + f_1(\beta_c, \beta_o, \eta, \zeta) = \Sigma^T(\beta_c) [D^T(\beta_o) \tau_o + E^T(\beta_c, \beta_o) \tau_\varphi] \quad (76)$$

As expressões (49), (55), (56) e (76) formam o modelo dinâmico de configuração para um robô móvel com rodas genérico, sendo que neste modelo, os torques τ_φ e τ_o , representam os torques que podem ser aplicados para rotação e orientação das rodas. Para um robô real, apenas alguns dos torques citados são efetivamente aplicados, já que geralmente o número mínimo de atuadores necessários é utilizado.

Os vetores τ_φ e τ_o podem possuir componentes identicamente nulos com a condição de que a rotação e orientação das rodas às quais os torques estão associados possa ser efetuada através do acionamento das demais rodas do robô. O vetor dos torques fornecidos para rotação e orientação τ_m das rodas não centradas do robô pode ser obtido de:

$$\begin{bmatrix} \tau_o \\ \tau_\varphi \end{bmatrix} = P \tau_m \quad (77)$$

sendo P uma matriz $(N_o + N) \times N_m$ que seleciona os componentes do vetor $[\tau_o \ \tau_\varphi]^T$, que são efetivamente utilizados como entradas de controle. Desta forma (76) pode ser reescrita como:

$$H_1(\beta_c, \beta_o) \dot{\eta} + \Sigma^T(\beta_c) V(\beta_o) \dot{\zeta} + f_1(\beta_c, \beta_o, \eta, \zeta) = B(\beta_c, \beta_o) P \tau_m \quad (78)$$

onde $B(\beta_c, \beta_o) = \Sigma^T(\beta_c) [D^T(\beta_o) E^T(\beta_c, \beta_o)]$.

O número mínimo de atuadores que devem ser utilizados para uma determinada classe de robô móvel, está relacionada com $\rho(B(\beta_c, \beta_o) P)$, ou seja, o *rank* desta matriz deve ser completo para todos os valores de β_c e β_o . Nos casos onde esta condição não for satisfeita, não existirão valores de β_c e β_o para os quais o robô se tornará sub-atuado (não haverá graus de liberdade suficientes nas entradas de controle para determinar a alocação do centro instantâneo de rotação do robô). Para um robô da classe (2,0), o número mínimo de atuadores a ser utilizado é dois.

Definindo-se (LAGES, 1998; SICILIANO; KHATIB, 2008):

$$\beta = \begin{bmatrix} \beta_c \\ \beta_o \end{bmatrix} \quad (79)$$

$$q = \begin{bmatrix} \xi \\ \beta \\ \varphi \end{bmatrix} \quad (80)$$

$$u = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (81)$$

$$H(\beta) = \begin{bmatrix} H_1(\beta_c, \beta_o) & \Sigma^T(\beta_c) V(\beta_o) \\ V^T(\beta_o) \Sigma(\beta_c) & I_c \end{bmatrix} \quad (82)$$

$$f(\beta, u) = \begin{bmatrix} f_1(\beta_c, \beta_o, \eta, \zeta) \\ f_2(\beta_c, \beta_o, \eta, \zeta) \end{bmatrix} \quad (83)$$

$$G(\beta) = \begin{bmatrix} B(\beta_c, \beta_o) P & 0 \\ 0 & I \end{bmatrix} \quad (84)$$

$$\tau_0 = \begin{bmatrix} \tau_m \\ \tau_c \end{bmatrix} \quad (85)$$

pode-se reescrever o modelo dinâmico de configuração de forma simplificada como:

$$\begin{cases} \dot{q} = S(q) u \\ H(\beta) \dot{u} + f(\beta, u) = G(\beta) \tau_0 \end{cases} \quad (86)$$

sendo que $\dot{q} = S(q) u$ representa a cinemática do robô móvel tendo como entrada as componentes de u e como saída a posição do robô (obtida pela integração de \dot{q}). Ainda, $H(\beta) \dot{u} + f(\beta, u) = G(\beta) \tau_0$ representa a dinâmica do robô, onde observa-se que a entrada do modelo é o torque e as saídas são as componentes de u .

2.2.9 Modelo Dinâmico de Pose

Geralmente, para aplicações de controle o que interessa são basicamente as coordenadas de pose do robô ${}^0\xi_c$, sendo que os valores das variáveis internas β_o e φ podem ser ignorados, uma vez que não são de interesse. Assim, pode-se escrever o modelo dinâmico de pose do robô móvel como:

$$\begin{cases} \dot{x} = B(x) u \\ H(\beta) \dot{u} + f(\beta, u) = G(\beta) \tau_0 \end{cases} \quad (87)$$

sendo que este modelo descreve a dinâmica entre as coordenadas de pose x e entradas de controle τ_0 . As variáveis β_o e φ aparentemente desaparecem do modelo, mas dependendo da classe do robô as mesmas permanecem implicitamente na realimentação de estados (193).

2.2.10 Modelo do robô móvel Twil

O robô móvel Twil pertence a classe (2,0) e possui duas rodas fixas e uma roda de apoio do tipo *castor wheel* (vide Figura 3(b)) que caracteriza-se como uma roda orientável não centrada. Assim, tem-se que $N = 3$, $N_f = 2$, e $N_{nc} = 1$ e desta forma, vão existir 6 expressões que representam as restrições de mobilidade do robô: expressões (16) e (17) para cada uma das rodas fixas e expressões (27) e (28) para a roda orientável não centrada.

A Figura 10 representa a estrutura do robô móvel proposto e a Tabela 2 apresenta os parâmetros geométricos e das rodas do mesmo. Deve-se observar que para o robô móvel Twil, o sistema de coordenadas $\{X_c, Y_c\}$ está posicionado no centro do eixo das rodas.

Tabela 2: Parâmetros das rodas do robô móvel Twil.

Roda	1	2	3
l	b	b	c
d	0	0	d_3
α	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	π
β	π	0°	β_3

Observando a Figura 10 e considerando-se as restrições (17) para as rodas fixas e (28) para a roda não-centrada, pode-se escrever:

$$\begin{bmatrix} \cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & l_1 \sin \beta_1 \\ \cos(\alpha_2 + \beta_2) & \sin(\alpha_2 + \beta_2) & l_2 \sin \beta_2 \\ \cos(\alpha_3 + \beta_3) & \sin(\alpha_3 + \beta_3) & l_3 \sin(\beta_3) + d_3 \end{bmatrix} {}^cR_0 {}^0\dot{\xi}_c + \begin{bmatrix} 0 \\ 0 \\ d_3 \end{bmatrix} \dot{\beta}_3 = 0 \quad (88)$$

e substituindo os parâmetros das rodas tem-se:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ -\cos(\beta_3) & -\sin(\beta_3) & l_3 \sin(\beta_3) + d_3 \end{bmatrix} {}^cR_0 {}^0\dot{\xi}_c + \begin{bmatrix} 0 \\ 0 \\ d_3 \end{bmatrix} \dot{\beta}_3 = 0$$

que está na forma matricial proposta em (37). Considerando-se apenas as restrições referentes às rodas fixas (uma vez que o robô não possui rodas centradas), obtém-se de (42):

$$C_1^*(\beta_c) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} = C_1^*$$

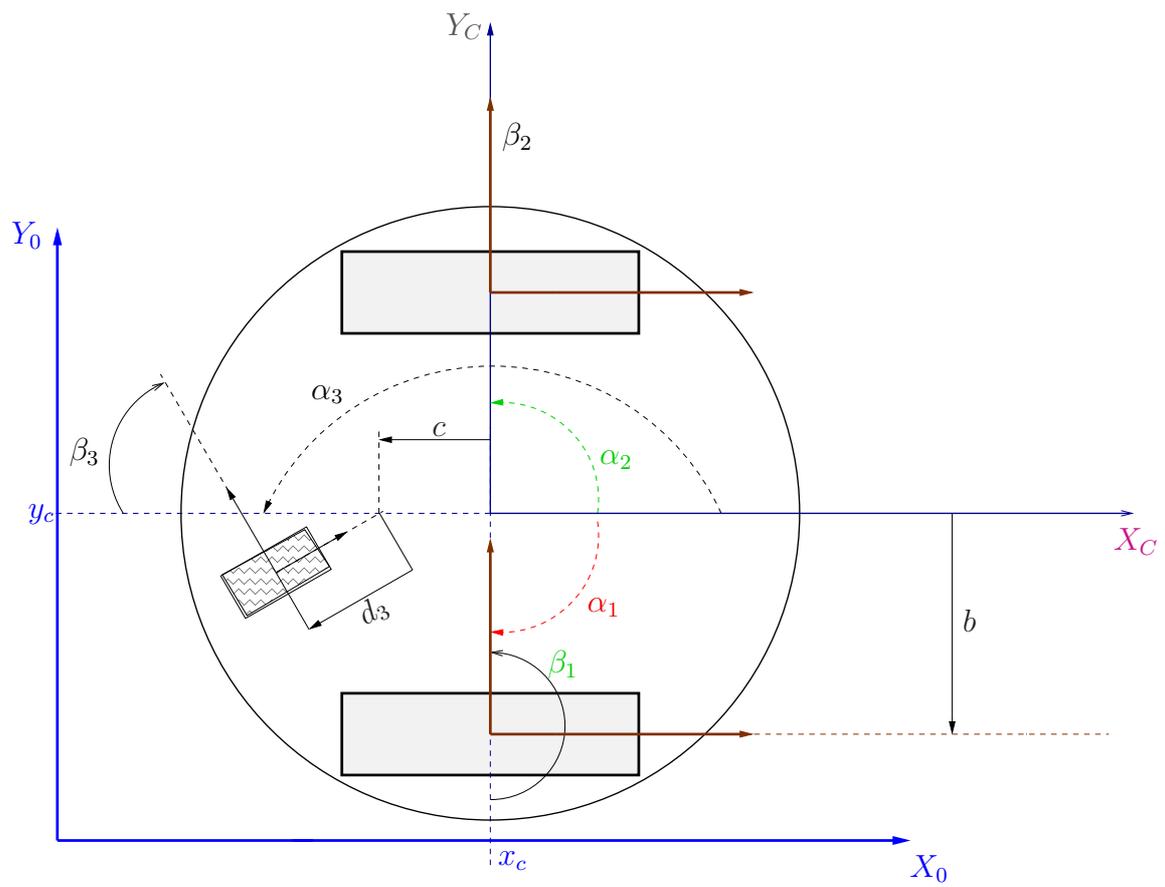


Figura 10: Estrutura do robô móvel Twil.

Assim como citado anteriormente, o robô móvel Twil pertence a classe (2,0), sendo que pode-se comprovar esta classificação verificando que $\rho(C_1^*) = 1$ e assim $\delta_m = \dim \mathcal{N}(C_1^*) = 3 - 1 = 2$, e $\delta_s = \rho(C_{1c}(\beta_c)) = 0$.

De (48), tem-se que as colunas de $\Sigma(\beta_c)$ formam uma base do espaço nulo de $C_1^*(\beta_c)$, e portanto:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = 0$$

e assim, tem-se que $\sigma = [\sigma_1 \ \sigma_2 \ \sigma_3]^T$ tal que $\sigma_2 = 0$, e por sua vez Σ é escolhido como:

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (89)$$

atendendo a condição $\sigma_2 = 0$ e com rank completo. E assim o modelo cinemático de pose do robô móvel Twil é:

$${}^0\dot{\xi}_c = {}^0R_c\Sigma\eta = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) & 0 \\ \sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \eta = \begin{bmatrix} \cos(\theta_c) & 0 \\ \sin(\theta_c) & 0 \\ 0 & 1 \end{bmatrix} \eta \quad (90)$$

$$\text{Deve-se observar que para o caso do robô móvel Twil } B(x) = {}^0R_c\Sigma = \begin{bmatrix} \cos(\theta_c) & 0 \\ \sin(\theta_c) & 0 \\ 0 & 1 \end{bmatrix}$$

uma vez que $N_c = 0$. De (90) pode-se escrever:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta_c\eta_1 \\ \sin\theta_c\eta_1 \\ \eta_2 \end{bmatrix} \quad (91)$$

ou seja, conforme representação da Figura 11, a velocidade linear do robô é dada por:

$$v_l = \sqrt{\dot{x}_c^2 + \dot{y}_c^2} = \sqrt{\cos^2\theta_c\eta_1^2 + \sin^2\theta_c\eta_1^2} = \eta_1$$

e para a velocidade angular:

$$\omega = \dot{\theta}_c = \eta_2$$

Se forem consideradas agora as restrições (16) e (27) para cada roda, pode-se escrever na forma matricial:

$$\begin{bmatrix} -\sin(\alpha_1 + \beta_1) & \cos(\alpha_1 + \beta_1) & l_1 \cos(\beta_1) \\ -\sin(\alpha_2 + \beta_2) & \cos(\alpha_2 + \beta_2) & l_2 \cos(\beta_2) \\ -\sin(\alpha_3 + \beta_3) & \cos(\alpha_3 + \beta_3) & l_3 \cos(\beta_3) \end{bmatrix} {}^cR_0 {}^0\dot{\xi}_c + \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \dot{\varphi} = 0 \quad (92)$$

e substituindo os parâmetros da Tabela 2 pode-se escrever:

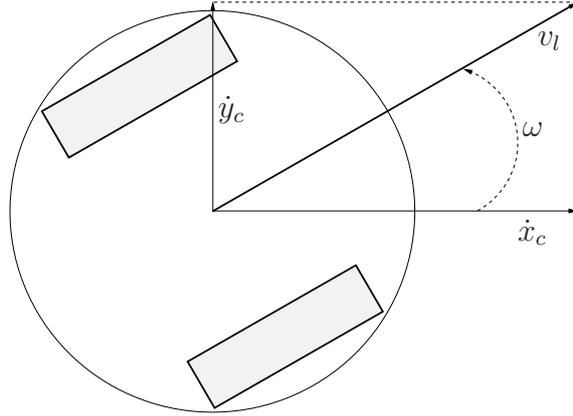


Figura 11: Velocidade linear e angular do robô móvel Twil.

$$\begin{bmatrix} -\sin\left(\alpha_1 + \frac{\pi}{2} - \alpha_1\right) & \cos\left(\alpha_1 + \frac{\pi}{2} - \alpha_1\right) & l_1 \cos\left(\frac{\pi}{2} - \alpha_1\right) \\ -\sin\left(\alpha_2 + \frac{\pi}{2} - \alpha_2\right) & \cos\left(\alpha_2 + \frac{\pi}{2} - \alpha_2\right) & l_2 \cos\left(\frac{\pi}{2} - \alpha_2\right) \\ -\sin(\pi + \beta_3) & \cos(\pi + \beta_3) & l_3 \cos(\beta_3) \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \dot{\varphi} = 0$$

ou:

$$\begin{bmatrix} -1 & 0 & l_1 \sin(\alpha_1) \\ -1 & 0 & l_2 \sin(\alpha_2) \\ \sin(\beta_3) & -\cos(\beta_3) & l_3 \cos(\beta_3) \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \dot{\varphi} = 0$$

A partir da Figura 10 pode-se verificar que $l_1 \sin(\alpha_1) = -b$ e $l_2 \sin(\alpha_2) = b$, e assim tem-se

$$\begin{bmatrix} -1 & 0 & -b \\ -1 & 0 & b \\ \sin(\beta_3) & -\cos(\beta_3) & l_3 \cos(\beta_3) \end{bmatrix} {}^c R_0^0 \dot{\xi}_c + \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \dot{\varphi} = 0$$

que está na forma (36). Utilizando (55) pode-se escrever

$$D(\beta_{nc}) = -C_{2nc}^{-1} C_{1nc}(\beta_3) = D(\beta_3) = -\frac{1}{d_3} \begin{bmatrix} -\cos(\beta_3) & -\sin(\beta_3) & l_3 \sin(\beta_3) + d_3 \end{bmatrix}$$

$$D(\beta_3) = \begin{bmatrix} \frac{\cos(\beta_3)}{d_3} & \frac{\sin(\beta_3)}{d_3} & -\frac{l_3 \sin(\beta_3)}{d_3} - 1 \end{bmatrix}$$

ainda,

$$\dot{\beta}_3 = D(\beta_3) \Sigma \eta$$

De (56) pode-se escrever:

$$\begin{aligned}
 E(\beta_c, \beta_{nc}) &= E(\beta_3) = -J_2^{-1} J_1(\beta_c, \beta_{nc}) \\
 &= - \begin{bmatrix} \frac{1}{r_1} & 0 & 0 \\ 0 & \frac{1}{r_2} & 0 \\ 0 & 0 & \frac{1}{r_3} \end{bmatrix} \begin{bmatrix} -1 & 0 & -b \\ -1 & 0 & b \\ \sin(\beta_3) & -\cos(\beta_3) & l_3 \cos(\beta_3) \end{bmatrix} \\
 &= - \begin{bmatrix} \frac{1}{r_1} & 0 & \frac{b}{r_1} \\ \frac{1}{r_2} & 0 & -\frac{r_1 b}{r_2} \\ -\frac{\sin(\beta_3)}{r_3} & \frac{\cos(\beta_3)}{r_3} & -l_3 \frac{\cos(\beta_3)}{r_3} \end{bmatrix}
 \end{aligned}$$

e

$$\dot{\varphi} = E(\beta_3) \Sigma \eta$$

Definindo-se agora o vetor de coordenadas de configuração q :

$$q = \begin{bmatrix} {}^0\xi_c \\ \beta_3 \\ \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix} \quad (93)$$

pode-se escrever o modelo cinemático de configuração do robô móvel Twil:

$$\dot{q} = S(q)\eta = \begin{bmatrix} \cos(\theta_c) & 0 \\ \sin(\theta_c) & 0 \\ 0 & 1 \\ \frac{\cos(\beta_3)}{d_3} & -\left(\frac{l_3}{d_3} \sin(\beta_3) + 1\right) \\ \frac{1}{r_1} & \frac{b}{r_1} \\ \frac{1}{r_2} & -\frac{r_1 b}{r_2} \\ -\frac{\sin(\beta_3)}{r_3} & -\left(\frac{l_3}{r_3}\right) \cos(\beta_3) \end{bmatrix} \eta \quad (94)$$

O modelo dinâmico do robô móvel Twil pode ser obtido utilizando-se (78), porém será utilizado a seguir o modelo desenvolvido em (LAGES, 1998):

$$\dot{x} = \begin{bmatrix} S\nu \\ f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T M S)^{-1} \end{bmatrix} \tau \quad (95)$$

com o estado x definido como

$$x = \begin{bmatrix} q \\ \nu \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ \theta \\ \varphi_r \\ \varphi_l \\ \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} \quad (96)$$

e:

$$f_2 = (S^T MS)^{-1} \left(-S^T M \dot{S} \nu - S^T V \right) \quad (97)$$

$$M = \begin{bmatrix} m & 0 & m_c d \sin(\theta) & 0 & 0 \\ 0 & m & -m_c d \cos(\theta) & 0 & 0 \\ m_c d \sin(\theta) & -m_c d \cos(\theta) & I_p & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \quad (98)$$

$$V = [m_c d \dot{\theta}^2 \cos(\theta) \quad m_c d \dot{\theta}^2 \sin(\theta) \quad 0 \quad 0 \quad 0]^T \quad (99)$$

$$S = \begin{bmatrix} c(d \sin(\theta) + b \cos(\theta)) & c(b \cos(\theta) - d \sin(\theta)) \\ c(b \sin(\theta) - d \cos(\theta)) & c(b \sin(\theta) + d \cos(\theta)) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (100)$$

Observa-se agora das definições da seção 2.2.7 que o estado é dado por:

$$x = \begin{bmatrix} q \\ u \end{bmatrix} \quad (101)$$

com q e u definidos respectivamente em (57) e (60).

Uma vez que os estados são diferentes, faz-se necessário uma mudança de variáveis que permita obter uma versão de (95) em função da variável u ao invés de ν , consequentemente obtendo o modelo do robô móvel Twil com o estado (101). Este procedimento é mais simples do que a obtenção do modelo proposto por (78), através dos cálculos de (61) a (63). Deve-se observar que agora $c = \frac{r}{2b}$ e d é a distância entre a origem do sistema de coordenadas inercial $\{X_c, Y_c\}$ e o centro de gravidade do robô, b permanece sendo a distância das rodas ao eixo de simetria.

Neste modelo o acoplamento entre a dinâmica e a cinemática do robô é realizada por $\nu = [\dot{\varphi}_r \quad \dot{\varphi}_l]^T$, enquanto que em (86) o acoplamento é feito por $\eta = [v_l \quad \omega]^T$, sendo que estas relações são influenciadas diretamente pela escolha de Σ . Assim para transformar o modelo (95) no modelo (86) é necessário fazer uma mudança de variáveis como será desenvolvido a seguir.

As duas últimas linhas de (95), representam a dinâmica do robô, e as linhas anteriores representam sua cinemática. Este modelo pode ser interpretado como uma cascata entre a dinâmica do robô (entrada τ e saída ν) e sua cinemática (entrada ν e saída q) assim como representado na Figura 12.

Ainda, considerando-se somente a dinâmica do robô (95) pode-se escrever:

$$\dot{\nu} = f_2 + (S^T MS)^{-1} \tau \quad (102)$$

Desenvolvendo f_2 utilizando (97) tem-se que:

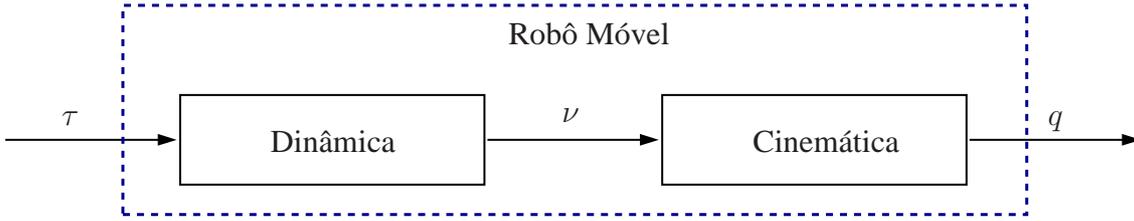


Figura 12: Representação em blocos do modelo do robô móvel.

$$S^T MS = \begin{bmatrix} c(d \sin(\theta) + b \cos(\theta)) & c(b \sin(\theta) - d \cos(\theta)) & c & 1 & 0 \\ c(b \cos(\theta) - d \sin(\theta)) & c(b \sin(\theta) + d \cos(\theta)) & -c & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} m & 0 & m_c d \sin(\theta) & 0 & 0 \\ 0 & m & -m_c d \cos(\theta) & 0 & 0 \\ m_c d \sin(\theta) & -m_c d \cos(\theta) & I_p & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}$$

$$\begin{bmatrix} c(d \sin(\theta) + b \cos(\theta)) & c(b \cos(\theta) - d \sin(\theta)) \\ c(b \sin(\theta) - d \cos(\theta)) & c(b \sin(\theta) + d \cos(\theta)) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (103)$$

$$S^T MS = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \quad (104)$$

onde:

$$\begin{aligned} s_{11} &= s_{22} = c^2 d^2 m + c^2 b^2 m + 2c^2 d^2 m_c + I_p c^2 + I_w \\ s_{12} &= s_{21} = c^2 b^2 m - c^2 d^2 m - 2c^2 d^2 m_c - I_p c^2 \end{aligned}$$

Calculando a inversa de (104) tem-se:

$$(S^T MS)^{-1} = \begin{bmatrix} K_1 & K_2 \\ K_2 & K_1 \end{bmatrix} \quad (105)$$

com

$$\begin{aligned} K_1 &= \frac{c^2 d^2 m + c^2 b^2 m + 2c^2 d^2 m_c + I_p c^2 + I_w}{K_{11}} \\ K_2 &= -\frac{c^2 b^2 m - c^2 d^2 m - 2c^2 d^2 m_c - I_p c^2}{K_{11}} \end{aligned}$$

$$\begin{aligned}
K_{11} &= 4c^4b^2d^2m^2 + 2c^2d^2mI_w + 8c^4b^2d^2mm_c + 4c^4b^2mI_p + 2c^2b^2mI_w \\
&+ 4c^2d^2m_cI_w + 2I_pI_wc^2 + I_w^2
\end{aligned}$$

Tem-se também:

$$\dot{S} = \frac{\partial}{\partial \theta} S \dot{\theta} = \begin{bmatrix} cd \cos(\theta) - cb \sin(\theta) & -cb \sin(\theta) - cd \cos(\theta) \\ cb \cos(\theta) + cd \sin(\theta) & cb \cos(\theta) - cd \sin(\theta) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \dot{\theta} \quad (106)$$

e portanto

$$\begin{aligned}
-S^T M \dot{S} \nu - S^T V &= \begin{bmatrix} c^2 b d m_c & 2c^2 b d m + c^2 b d m_c \\ -2c^2 b d m - c^2 b d m_c & -c^2 b d m_c \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} \dot{\theta} \\
&- \begin{bmatrix} c b d m_c \\ c b d m_c \end{bmatrix} \dot{\theta}^2
\end{aligned} \quad (107)$$

Uma vez que:

$$\dot{\theta} = c (\dot{\varphi}_r - \dot{\varphi}_l) \quad (108)$$

tem-se:

$$\begin{aligned}
-S^T M \dot{S} \nu - S^T V &= \begin{bmatrix} \frac{c^2 b d m_c}{2} & -c^2 b d m - \frac{c^2 b d m_c}{2} \\ -c^2 b d m - \frac{c^2 b d m_c}{2} & \frac{c^2 b d m_c}{2} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l \\ \dot{\varphi}_l^2 - \dot{\varphi}_r \dot{\varphi}_l \end{bmatrix} \\
&- \begin{bmatrix} \frac{c b d m_c}{4} (\dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l - \dot{\varphi}_r \dot{\varphi}_l + \dot{\varphi}_l^2) \\ \frac{c b d m_c}{4} (\dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l - \dot{\varphi}_r \dot{\varphi}_l + \dot{\varphi}_l^2) \end{bmatrix} \\
&= \begin{bmatrix} \frac{c^2 b d m_c}{2} - \frac{c b d m_c}{4} & -c^2 b d m - \frac{c^2 b d m_c}{2} - \frac{c b d m_c}{4} \\ -c^2 b d m - \frac{c^2 b d m_c}{2} - \frac{c b d m_c}{4} & \frac{c^2 b d m_c}{2} - \frac{c b d m_c}{4} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l \\ \dot{\varphi}_l^2 - \dot{\varphi}_r \dot{\varphi}_l \end{bmatrix}
\end{aligned} \quad (109)$$

Assim, pode-se escrever:

$$f_2 = \begin{bmatrix} K_1 & K_2 \\ K_2 & K_1 \end{bmatrix} \begin{bmatrix} c b d \left(\frac{c m_c}{2} - \frac{m_c}{4} \right) & c b d \left(-c m - \frac{c m_c}{2} - \frac{m_c}{4} \right) \\ c b d \left(-c m - \frac{c m_c}{2} - \frac{m_c}{4} \right) & c b d \left(\frac{c m_c}{2} - \frac{m_c}{4} \right) \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l \\ \dot{\varphi}_l^2 - \dot{\varphi}_r \dot{\varphi}_l \end{bmatrix} \quad (110)$$

ou ainda,

$$f_2 = \begin{bmatrix} K_3 & K_4 \\ K_4 & K_3 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l \\ \dot{\varphi}_l^2 - \dot{\varphi}_r \dot{\varphi}_l \end{bmatrix} \quad (111)$$

com

$$K_3 = cbd \left(K_1 \left(\frac{cm_c}{2} - \frac{m_c}{4} \right) + K_2 \left(-cm - \frac{cm_c}{2} - \frac{m_c}{4} \right) \right)$$

$$K_4 = cbd \left(K_1 \left(-cm - \frac{cm_c}{2} - \frac{m_c}{4} \right) + K_2 \left(\frac{cm_c}{2} - \frac{m_c}{4} \right) \right)$$

$$K_3 = K_1 \frac{c^2 b d m_c}{2} - K_1 \frac{c b d m_c}{4} - K_2 c^2 b d m - K_2 \frac{c^2 b d m_c}{2} - K_2 \frac{c b d m_c}{4}$$

$$= \frac{c^4 b^3 d m^2 - c^4 b d^3 m^2 - 2c^4 b d^3 m m_c - I_p c^4 b d m}{K_{11}}$$

$$+ \frac{2c^4 b^3 d m m_c + I_w c^2 b d m_c - c^4 b d^3 m_c^2}{K_{12}} - \frac{I_w c b d m_c}{K_{13}}$$

$$K_{12} = 8c^4 b^2 d^2 m^2 + 4c^2 d^2 m I_w + 16c^4 b^2 d^2 m m_c + 8c^4 b^2 m I_p + 4c^2 b^2 m I_w$$

$$+ 8c^2 d^2 m_c I_w + 4I_p I_w c^2 + 2I_w^2$$

$$K_{13} = 16c^4 b^2 d^2 m^2 + 8c^2 d^2 m I_w + 32c^4 b^2 d^2 m m_c + 16c^4 b^2 m I_p + 8c^2 b^2 m I_w$$

$$+ 16c^2 d^2 m_c I_w + 8I_p I_w c^2 + 4I_w^2$$

$$K_4 = -K_1 c^2 b d m - K_1 \frac{c^2 b d m_c}{2} - K_1 \frac{c b d m_c}{4} + K_2 \frac{c^2 b d m_c}{2} - K_2 \frac{c b d m_c}{4}$$

$$= -\frac{c^4 b d^3 m^2 + c^4 b^3 d m^2 + 2c^4 b d^3 m m_c + I_p c^4 b d m + I_w c^2 b d m}{K_{11}}$$

$$- \frac{2c^4 b^3 d m m_c + I_w c^2 b d m_c}{K_{12}}$$

$$- \frac{2c^3 b d^3 d m m_c + 4c^3 b d^3 m_c^2 + 2I_p c^3 b d m_c + I_w c b d m_c}{K_{13}}$$

De (102):

$$\dot{v} = \begin{bmatrix} \ddot{\varphi}_r \\ \ddot{\varphi}_l \end{bmatrix} = \begin{bmatrix} K_3 & K_4 \\ K_4 & K_3 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r^2 - \dot{\varphi}_r \dot{\varphi}_l \\ \dot{\varphi}_l^2 - \dot{\varphi}_r \dot{\varphi}_l \end{bmatrix} + \begin{bmatrix} K_1 & K_2 \\ K_2 & K_1 \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (112)$$

e uma vez que (LAGES, 1998):

$$v = \frac{r(\dot{\varphi}_r + \dot{\varphi}_l)}{2} \quad (113)$$

$$\omega = c(\dot{\varphi}_r - \dot{\varphi}_l) \quad (114)$$

tem-se:

$$\dot{\varphi}_r = \frac{v}{r} + \frac{\omega}{2c} \quad (115)$$

$$\dot{\varphi}_l = \frac{v}{r} - \frac{\omega}{2c} \quad (116)$$

e:

$$\ddot{\varphi}_r = \frac{\dot{v}}{r} + \frac{\dot{\omega}}{2c} \quad (117)$$

$$\ddot{\varphi}_l = \frac{\dot{v}}{r} - \frac{\dot{\omega}}{2c} \quad (118)$$

Assim, substituindo-se (115), (116), (117) e (118) em (112) tem-se:

$$\begin{bmatrix} \frac{\dot{v}}{r} + \frac{\dot{\omega}}{2c} \\ \frac{\dot{v}}{r} - \frac{\dot{\omega}}{2c} \end{bmatrix} = \begin{bmatrix} K_3 & K_4 \\ K_4 & K_3 \end{bmatrix} \begin{bmatrix} \frac{v\omega}{rc} + \frac{\omega^2}{2c^2} \\ -\frac{v\omega}{rc} + \frac{\omega^2}{2c^2} \end{bmatrix} + \begin{bmatrix} K_1 & K_2 \\ K_2 & K_1 \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (119)$$

Somando-se e subtraindo-se as duas equações de (119) e substituindo $c = \frac{r}{2b}$ obtem-se

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & \frac{2K_3b^2}{r} + \frac{2K_4b^2}{r} \\ \frac{2K_3}{r} - \frac{2K_4}{r} & 0 \end{bmatrix} \begin{bmatrix} v\omega \\ \omega^2 \end{bmatrix} + \begin{bmatrix} \frac{K_1r}{2b} + \frac{K_2r}{2b} & \frac{K_1r}{2b} + \frac{K_2r}{2b} \\ \frac{K_1r}{2b} - \frac{K_2r}{2b} & -\frac{K_1r}{2b} + \frac{K_2r}{2b} \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (120)$$

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & K_5 \\ K_6 & 0 \end{bmatrix} \begin{bmatrix} v\omega \\ \omega^2 \end{bmatrix} + \begin{bmatrix} K_7 & K_7 \\ K_8 & -K_8 \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (121)$$

com

$$K_5 = \frac{2K_3b^2}{r} + \frac{2K_4b^2}{r} = -\frac{8c^4b^3d^3mm_c + 4I_p c^4b^3dm + 2I_w c^2b^3dm}{r(K_{11})}$$

$$-\frac{2c^4b^3d^3m_c^2}{r(K_{12})} - \frac{4c^3b^3d^3dmm_c + 8c^3b^3d^3m_c^2 + 4I_p c^3b^3dm_c + 4I_w cb^3dm_c}{r(K_{13})}$$

$$K_6 = \frac{2K_3}{r} - \frac{2K_4}{r} = \frac{2c^4b^3dm^2 + I_w c^2b^3dm}{rK_{14}} +$$

$$\frac{4c^4b^3dmm_c + 2I_w c^2b^3dm_c - c^4bd^3m_c^2}{r(K_{11})} + \frac{2c^3bd^3dmm_c + 4c^3bd^3m_c^2 + 2I_p c^3b^3dm_c}{r(K_{12})}$$

$$K_{14} = 2c^4b^2d^2m^2 + c^2d^2mI_w + 4c^4b^2d^2mm_c + 2c^4b^2mI_p + c^2b^2mI_w$$

$$+ 2c^2d^2m_cI_w + I_pI_wc^2 + \frac{I_w^2}{2}$$

$$K_7 = \frac{K_1r}{2} + \frac{K_2r}{2} = \frac{r(2c^2d^2m + 4c^2d^2m_c + 2I_p c^2 + I_w)}{K_{12}}$$

$$K_8 = \frac{K_1 r}{2b} - \frac{K_2 r}{2b} = \frac{r(2c^2 b^2 m + I_w)}{K_{15}}$$

$$\begin{aligned} K_{15} &= 8c^4 b^3 d^2 m^2 + 4c^2 b d^2 m I_w + 16c^4 b^3 d^2 m m_c + 8c^4 b^3 m I_p + 4c^2 b^3 m I_w \\ &+ 8c^2 b d^2 m_c I_w + 4b I_p I_w c^2 + 2b I_w^2 \end{aligned}$$

E utilizando (86) e (121), o modelo dinâmico de configuração do robô móvel Twil pode ser expresso por:

$$\begin{cases} \dot{q} = S(q) u \\ \dot{u} = \begin{bmatrix} 0 & K_5 \\ K_6 & 0 \end{bmatrix} \begin{bmatrix} u_1 u_2 \\ u_2^2 \end{bmatrix} + \begin{bmatrix} K_7 & K_7 \\ K_8 & -K_8 \end{bmatrix} \tau \end{cases} \quad (122)$$

que pode ser reescrita na forma proposta em (86), como:

$$\begin{cases} \dot{q} = S(q) u \\ \dot{u} = -f(u) + G\tau \end{cases} \quad (123)$$

onde:

$$\begin{aligned} f(u) &= - \begin{bmatrix} 0 & K_5 \\ K_6 & 0 \end{bmatrix} \begin{bmatrix} u_1 u_2 \\ u_2^2 \end{bmatrix} \\ G &= \begin{bmatrix} K_7 & K_7 \\ K_8 & -K_8 \end{bmatrix} \end{aligned}$$

com os termos K_5 , K_6 , K_7 e K_8 sendo constantes dependendo apenas dos parâmetros geométricos e de inércia do robô.

2.3 Modelo Cinemático do Manipulador Barrett WAM

Um manipulador pode ser representado por uma cadeia cinemática de elos que podem ser conectados entre si por diferentes tipos de juntas, principalmente juntas rotacionais e prismáticas. Considera-se que uma das pontas da cadeia cinemática vai estar fixa, enquanto que a outra ponta pode movimentar-se.

O movimento de toda a cadeia cinemática é determinado pela composição do movimento de cada elo com referência ao elo anterior, assim por exemplo, para verificar qual a posição de um objeto manipulado no espaço é necessário descrever a posição e orientação do elo terminal com relação à base. Dois problemas principais são estudados dentro da cinemática do robô: o problema da cinemática direta e o problema da cinemática inversa. O problema da cinemática direta busca determinar a pose do elo terminal do manipulador com relação a um sistema de coordenadas base, dados o vetor dos ângulos das n juntas $q(t) = [q_1(t) \ q_2(t) \ \cdots \ q_n(t)]^T$ e os demais parâmetros geométricos dos elos. Já o problema da cinemática inversa parte de uma pose desejada para o elo terminal do manipulador e busca obter os ângulos de junta, o problema pode ter várias soluções ou não ter solução.

Há uma grande diversidade de literatura técnica compreendendo a teoria cinemática de manipuladores como pode ser encontrado em livros clássicos como (PAUL, 1981; CRAIG, 2005; FU; GONZALES; LEE, 1987; SICILIANO et al., 2009; SPONG; HUTCHINSON; VIDYASAGAR, 2006), dentre outros.

2.3.1 Notação de Denavit e Hartenberg

A notação de Denavit e Hartenberg é uma convenção de como estabelecer os sistemas de coordenadas para cada elo, de forma a sistematizar o processo de obtenção da cinemática direta do manipulador (FU; GONZALES; LEE, 1987). Utilizando-se a mesma é possível descrever a relação entre dois sistemas de coordenadas, utilizando-se apenas quatro parâmetros.

2.3.1.1 Regras Básicas

A partir da Figura 13 podem-se estabelecer (FU; GONZALES; LEE, 1987) as seguintes regras referentes ao eixos $[x \ y \ z]$ associados a um sistema de coordenadas:

1. O eixo z_{i-1} está ao longo do eixo de movimentação da i -ésima junta;
2. O eixo x_i é normal a z_{i-1} e aponta no sentido que se afasta do mesmo;
3. O eixo y_i deve ser disposto de forma a completar o sistema de coordenadas obedecendo a regra da mão direita.

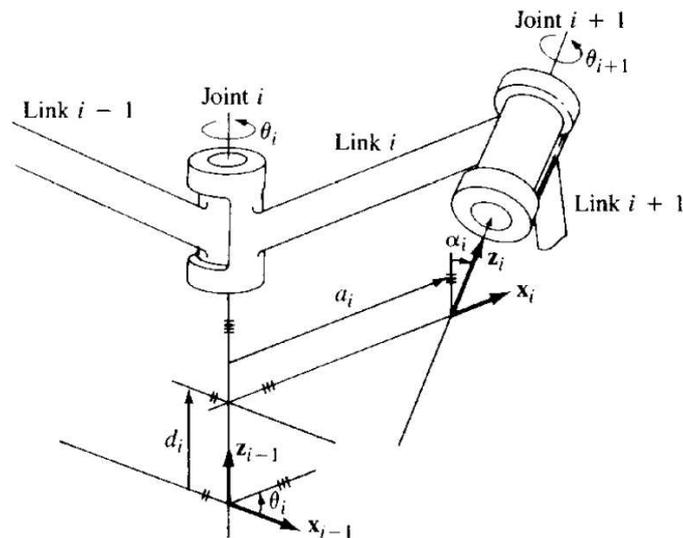


Figura 13: Sistema de coordenadas dos elos e respectivos parâmetros de Denavit e Hartenberg, adaptado de (FU; GONZALES; LEE, 1987).

2.3.1.2 Atribuição dos Sistemas de Coordenadas

1. *Sistema de coordenadas da base:* o sistema de coordenadas da base x_0, y_0, z_0 deve ser estabelecido na base de apoio do robô sendo que o eixo z_0 deve estar sobre o eixo da junta 1 apontando para o ombro do robô. Os eixos x_0 e y_0 podem ser estabelecidos de maneira conveniente desde que formem um sistema ortonormal;

2. *Sistema de coordenadas das juntas*: para cada uma das juntas $i = 1 \cdots (n - 1)$ do robô, repita:
 - (a) *Eixo da junta* - alinhar z_i com o eixo da junta $i + 1$ (rotacional ou prismática);
 - (b) *Origem do sistema i* - localize a origem do sistema i na intersecção de z_i com z_{i-1} ou na intersecção da normal comum a z_i e z_{i-1} e o eixo z_i ;
 - (c) *Eixo x_i* - estabelecer $x_i = \pm(z_{i-1} \times z_i) / \| z_{i-1} \times z_i \|$ ou sobre a normal comum entre z_{i-1} e z_i no caso em que os mesmos são paralelos;
 - (d) *Eixo y_i* - estabelecer $y_i = +(z_i \times x_i) / \| z_i \times x_i \|$ de forma a completar o sistema.
3. *Sistema de coordenadas da garra*: é comum que a n -ésima junta do robô seja rotacional. Alinhe z_n na mesma direção que z_{n-1} apontando no sentido de se afastar do robô. Alinhar x_n para que seja normal a z_{n-1} e a z_n . Ainda, y_n deve completar o sistema.
4. *Parâmetros das juntas e do elos*: para cada $i = 1 \cdots n$ repita:
 - (a) d_i - é a distância da origem do sistema $i - 1$ até a intersecção dos eixos z_{i-1} e x_i , medida sobre z_{i-1} . É a variável de junta no caso da junta i ser prismática;
 - (b) a_i - é a distância da intersecção de z_{i-1} e x_i até a origem do sistema i , medida sobre x_i ;
 - (c) θ_i - é o ângulo de rotação em torno de z_{i-1} medido de x_{i-1} até x_i . É a variável de junta no caso da junta i ser rotacional;
 - (d) α_i - é o ângulo de rotação em torno de x_i medido de z_{i-1} à z_i .

2.3.1.3 Transformação entre Sistemas de Coordenadas a partir dos Parâmetros de Denavit e Hartenberg

Das definições dos parâmetros de Denavit e Hartenberg tem-se que para um ponto P_i expresso no sistema de coordenadas i , o mesmo poderá ser expresso em um outro sistema de coordenadas $i - 1$ através da seguinte sequência de transformações:

1. Rotação de um ângulo θ_i em torno de z_{i-1} , para alinhar x_{i-1} com x_i ;

$$T_{z,\theta_i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (124)$$

2. Translação de d_i ao longo de z_{i-1} para fazer x_{i-1} coincidente com x_i ;

$$T_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (125)$$

3. Translação de a_i ao longo de x_i para fazer as origens e os eixos x coincidentes;

$$T_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (126)$$

4. Rotação de um ângulo α_i em torno de x_i para fazer os dois sistemas coincidentes;

$$T_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (127)$$

Logo, pode-se escrever:

$$\begin{aligned} {}^{i-1}T_i &= T_{z,d_i} T_{z,\theta_i} T_{x,a_i} T_{x,\alpha_i} \\ &= \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (128)$$

e para a transformação inversa:

$$\begin{aligned} {}^{i-1}T_i^{-1} &= {}^i T_{i-1} = \\ &= \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 & -a_i \\ -\cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & \sin(\alpha_i) & -d_i\sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & -d_i\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (129)$$

2.3.2 Modelo Cinemático Direto do Manipulador Barrett WAM

Este trabalho é baseado no manipulador Barrett WAM. A Figura 14 mostra o manipulador utilizando como elemento terminal uma garra com três dedos.

A cinemática direta do manipulador Barrett WAM pode ser obtida utilizando-se os parâmetros de Denavit e Hartenberg da Tabela 3, juntamente com a aplicação da matriz de transformação homogênea (128) entre cada um dos sistemas de coordenadas de cada um dos elos do manipulador (BARRETT TECHNOLOGY, 2006). Assim, tem-se para a cinemática direta do manipulador:

$${}^{base}T_{garra} = {}^{base}T_0 T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_{garra} \quad (130)$$

A relação entre o sistema de coordenadas da base e o sistema de coordenadas do elo 0 não obedecem a convenção de Denavit e Hartenberg, podendo ser representada pela transformação:



Figura 14: Manipulador Barrett WAM, adaptado de (BARRETT TECHNOLOGY, 2006).

$${}^{base}T_0 = \begin{bmatrix} 1 & 0 & 0 & 0.22 \\ 0 & 1 & 0 & 0.14 \\ 0 & 0 & 1 & 0.346 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (131)$$

onde os valores de translação podem ser obtidos de BARRETT TECHNOLOGY (2006).

A Figura15 representa a definição dos sistemas de coordenadas do manipulador.

Tabela 3: Parâmetros de Denavit e Hartenberg para o manipulador Barrett WAM configurado para sete graus de liberdade.

Elo i	$a_i(m)$	$\alpha_i(rad)$	$d_i(rad)$	$\theta_i(rad)$
0	0	$-\pi$	0	0
1	0	$-\frac{\pi}{2}$	0	θ_1
2	0	$\frac{\pi}{2}$	0	θ_2
3	0.045	$-\frac{\pi}{2}$	0.55	θ_3
4	-0.045	$\frac{\pi}{2}$	0	θ_4
5	0	$-\frac{\pi}{2}$	0.3	θ_5
6	0	$\frac{\pi}{2}$	0	θ_6
7	0	0	0.06	θ_7
garra	0	0	0	

2.3.3 Modelo Cinemático Inverso do Manipulador Barrett WAM

O modelo Cinemático Inverso do manipulador Barrett WAM apresentado nesta seção segue a proposta de SINGH; CLAASSENS (2010), onde o modelo é desenvolvido utilizando método geométrico (FU; GONZALES; LEE, 1987).

O problema cinemático inverso de um manipulador caracteriza-se em encontrar as variáveis de junta (θ_1 a θ_7 neste caso) dadas a posição (${}^{base}P_{garra}^d$) e orientação (${}^{base}R_{garra}^d$) desejada da garra do manipulador. Para a proposta apresentada, a posição desejada do punho do robô pode ser obtida utilizando-se:

$${}^{base}P_{5_{org}}^d = {}^{base}P_{garra}^d - {}^{base}R_{garra}^d d_{garra} {}^{garra}\hat{Z}_{garra} \quad (132)$$

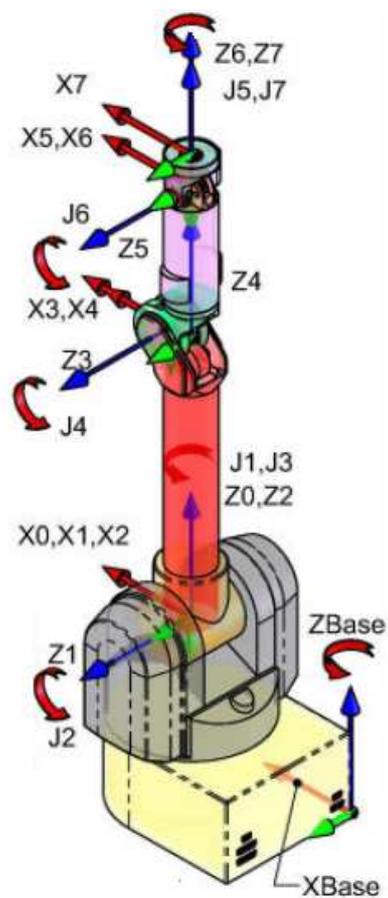


Figura 15: Sistemas de coordenadas do manipulador Barrett WAM, adaptado de BARRETT TECHNOLOGY (2006).

A análise das possíveis posições do manipulador é realizada tendo como base o esquema representado na Figura 16. Nesta análise geométrica, considera-se que as partes inferior e superior do manipulador geram um volume de trabalho correspondente a uma casca esférica, sendo que a intersecção destas, pode gerar uma circunferência, um ponto ou um conjunto nulo (neste caso a posição desejada para o punho do robô não é atingível). A parte inferior do manipulador corresponde aos elementos compreendidos da base até a junta 4, e a parte superior do manipulador compreende os elementos da junta 4 até a ferramenta. Todas as possíveis soluções são calculadas considerando-se a posição normalizada do punho do manipulador posicionado verticalmente acima da junta da base.

O pontos O e B representados na Figura 16 estão localizados na origem do sistema 1 e na origem do sistema 5 respectivamente, ainda, o ponto A é a origem do sistema 3, que coincide com a origem do sistema 4.

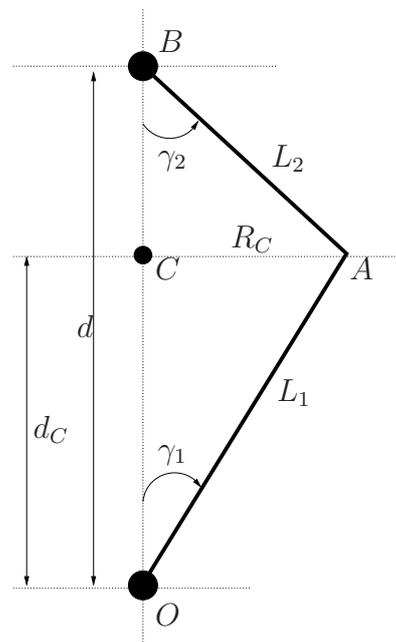


Figura 16: Geometria de intersecção dos volumes das partes inferior e superior do manipulador WAM, adaptado de SINGH; CLAASSENS (2010).

Se os segmentos L_1 e L_2 pudessem girar livremente em torno dos pontos O e B respectivamente, a ponta de cada segmento geraria uma casca esférica com centros em O e B . Mas uma vez que os segmentos estão conectados no ponto A , a única solução possível é a intersecção das cascas esféricas geradas, sendo que esta intersecção pode resultar em uma circunferência, um ponto (que é equivalente a uma circunferência com raio zero) ou nula, sendo que interpreta-se a intersecção nula como um caso onde não há solução para o problema cinemático inverso.

Ou seja, as possíveis posições do ponto A descrevem um círculo de raio R_C em torno do ponto C que está a uma distância d_C do ponto O .

$$d_C = L_1 \cos(\alpha_1) = d - L_2 \cos(\alpha_2) \quad (133)$$

$$R_C = L_1 \sin(\alpha_1) = L_2 \sin(\alpha_2) \quad (134)$$

Da geometria apresentada na Figura 16, os ângulos γ_1 e γ_2 podem ser calculados como segue:

$$\gamma_2 = \cos^{-1} [(d^2 + L_2^2 - L_1^2) / (2dL_2)] \quad (135)$$

$$\gamma_1 = \sin^{-1} [L_2 \sin(\gamma_2) / (L_1)] \quad (136)$$

O comprimento do segmento L_1 é calculado em relação aos sistemas de coordenadas 1, sendo que o valor resultante é uma constante que não depende das variáveis de junta:

$$L_1 = |{}^1P_{3org} - {}^1P_{1org}| \quad (137)$$

$${}^1P_{1org} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (138)$$

$$\begin{aligned} {}^1P_{3org} &= {}^1T_2 {}^2T_3 {}^3P_{3org} = \\ &\begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & 0 \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & a_3 \cos(\theta_3) \\ \sin(\theta_3) & 0 & \cos(\theta_3) & a_3 \sin(\theta_3) \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \\ &\begin{bmatrix} \cos(\theta_2) a_3 \cos(\theta_3) + \sin(\theta_2) d_3 \\ \sin(\theta_2) a_3 \cos(\theta_3) - \cos(\theta_2) d_3 \\ a_3 \sin(\theta_3) \\ 1 \end{bmatrix} \end{aligned} \quad (139)$$

portanto:

$$L_1 = \sqrt{a_3^2 + d_3^2} \quad (140)$$

e o comprimento do segmento L_2 é calculado em relação aos sistemas de coordenadas 3, sendo que da mesma forma que para o segmento L_1 , o resultado é uma constante que portanto não depende das variáveis de junta:

$$L_2 = |{}^3P_{5org} - {}^3P_{3org}| \quad (141)$$

$${}^3P_{3org} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (142)$$

$$\begin{aligned} {}^3P_{5org} &= {}^3T_4 {}^4T_5 {}^5P_{5org} = \\ &\begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & a_4 \cos(\theta_4) \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & a_4 \sin(\theta_4) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \\ &\begin{bmatrix} d_5 \sin(\theta_4) + a_4 \cos(\theta_4) \\ -d_5 \cos(\theta_4) + a_4 \sin(\theta_4) \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (143)$$

$$L_2 = \sqrt{a_4^2 + d_5^2} \quad (144)$$

Ainda, para calcular o valor do comprimento do segmento d utilizam-se as coordenadas dos pontos B e O , sendo que o mesmo é independente das variáveis de juntas, dependendo apenas de ${}^0P_{5org}^d$, que pode ser calculado de (132).

$$d = |B - O| \quad (145)$$

$${}^1B = {}^1P_{5org}^d = {}^1T_0 {}^0P_{5org}^d \quad (146)$$

E considerando que ${}^0P_{5org}^d$ vai ter a forma $[{}^0x_{5org}^d \quad {}^0y_{5org}^d \quad {}^0z_{5org}^d \quad 1]^T$ pode-se escrever:

$${}^1B = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^0x_{5org}^d \\ {}^0y_{5org}^d \\ {}^0z_{5org}^d \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) {}^0x_{5org}^d + \sin(\theta_1) {}^0y_{5org}^d \\ -{}^0z_{5org}^d \\ -\sin(\theta_1) {}^0x_{5org}^d + \cos(\theta_1) {}^0y_{5org}^d \\ 1 \end{bmatrix} \quad (147)$$

E com:

$${}^1O = {}^1P_{1org} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (148)$$

tem-se para d :

$$d = \sqrt{({}^0x_{5org}^d)^2 + ({}^0y_{5org}^d)^2 + ({}^0z_{5org}^d)^2} \quad (149)$$

Para calcular os ângulos de juntas θ_1 à θ_7 , são consideradas as geometrias apresentadas nas Figuras 18 e 17 que representam respectivamente o manipulador configurado com o cotovelo para fora e para dentro, e onde pode-se verificar as posições dos pontos L_J e U_J .

As possíveis posições para os pontos L_J e U_J também descrevem circunferências semelhantes a descrita pelas possíveis posições do ponto A . A partir deste ponto do trabalho, somente o ponto L_J será considerado uma vez que não há necessidade de analisar o ponto U_J para resolver o problema cinemático inverso. A circunferência referente a L_J tem raio R_{LJ} e está a uma distância d_{LJ} do ponto O .

Observando a Figura 17 pode-se verificar que:

$$d_{LJ} = d_c - l_L \sin(\theta_L) \quad (150)$$

$$R_{LJ} = R_c + l_L \cos(\theta_L) \quad (151)$$

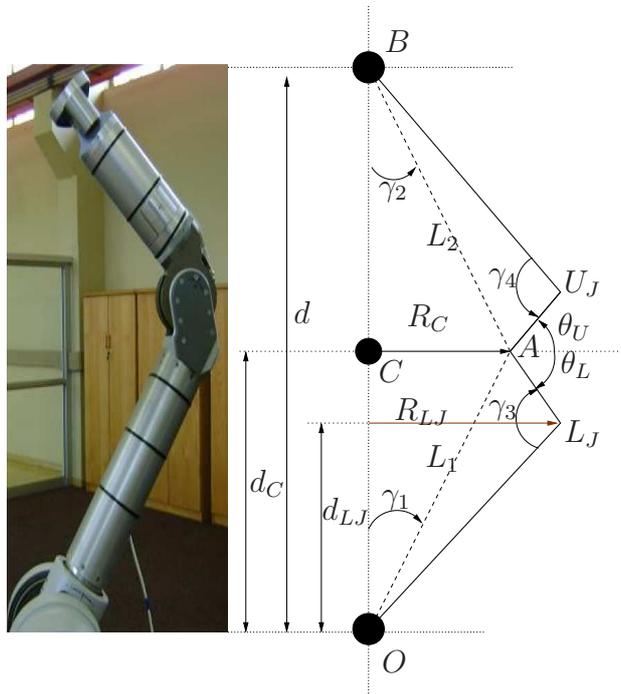


Figura 17: Posição do manipulador WAM com o cotovelo para fora, adaptado de SINGH; CLAASSENS (2010).

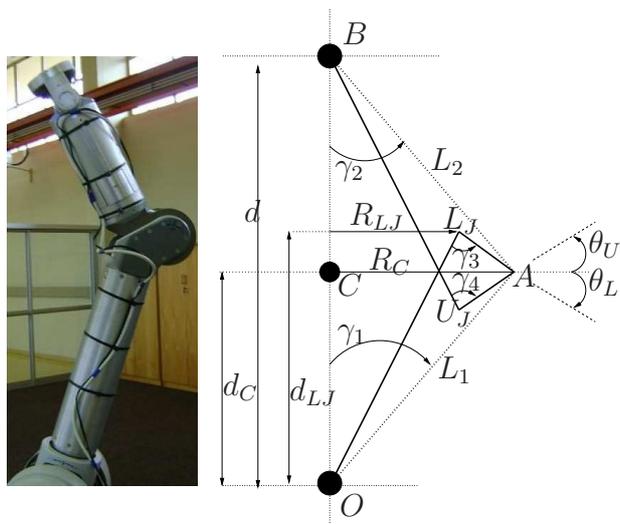


Figura 18: Posição do manipulador WAM com o cotovelo para dentro, adaptado de SINGH; CLAASSENS (2010).

onde l_L é o tamanho do "offset" do elo, ou seja, a distância do ponto A ao ponto L_J , e $l_L = 0.06364m$ (BARRETT TECHNOLOGY, 2006).

Para a posição com o cotovelo para fora tem-se que:

$$\gamma_3 = \frac{\pi}{2} - \text{atan2}(|a_4|, d_5) \quad (152)$$

$$\gamma_4 = \frac{\pi}{2} - \text{atan2}(a_3, d_3) \quad (153)$$

$$\theta_L = \pi - \left(\frac{\pi}{2} - \gamma_1\right) - \gamma_3 \quad (154)$$

$$\theta_U = \pi - \left(\frac{\pi}{2} - \gamma_2\right) - \gamma_4 \quad (155)$$

E para a posição com o cotovelo para dentro tem-se:

$$\theta_L = \gamma_3 - \left(\frac{\pi}{2} - \gamma_1\right) \quad (156)$$

$$\theta_U = \gamma_4 - \left(\frac{\pi}{2} - \gamma_2\right) \quad (157)$$

Deve-se observar que as possíveis posições para o ponto A descrevem uma circunferência centrada no ponto C , com raio R_C a uma distância d_C do ponto O , e que as possíveis posições para o ponto L_J descrevem uma circunferência centrada em C_{LJ} , com raio R_{LJ} a uma distância d_{LJ} do ponto O . Ainda, as Figuras 16, 17 e 18 se referem a uma posição normalizada do robô sendo que o eixo OB pode estar em qualquer orientação espacial (diferente da vertical).

As circunferências descritas pelas possíveis posições dos pontos A e L_J são paralelos e em torno do eixo OB . Pode-se parametrizar os mesmos através de seu raio R , distância D ao ponto O e ângulo $\phi \in [-\pi, \pi]$. Ou seja, tem-se cada ponto sobre uma das circunferências dada por:

$$C_n(R, D, \phi) = \begin{bmatrix} R \cos(\phi) \\ R \sin(\phi) \\ D \end{bmatrix} \quad (158)$$

Assim, as possíveis posições para o ponto A na configuração normalizada são dadas por $C_n(R_c, d_c, \phi)$ e as possíveis posições para o ponto L_J são dadas por $C_n(R_{LJ}, d_{LJ}, \phi)$, sendo que considerando-se o fato de que o eixo OB em geral não está na vertical, as circunferências normalizadas precisam ser rotacionadas para a sua posição real.

Sendo R_n a matriz de rotação que move a posição desejada do punho do manipulador para a posição normalizada equivalente (Figura 16), tem-se:

$$\begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} = R_n {}^0 P_{5org}^d \quad (159)$$

A matriz R_n é a matriz de rotação que rotaciona um ponto em torno de um eixo arbitrário, transformando-o em outro, cujo cálculo é mostrado no apêndice F. Assim, as possíveis posições para o ponto A são dadas por:

$${}^0A = R_n^T C_n(R_c, d_c, \phi) = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \quad (160)$$

e as possíveis posições para o ponto L_J são dadas por:

$${}^0L_J = R_n^T C_n(R_{LJ}, d_{LJ}, \phi) = \begin{bmatrix} x_{LJ} \\ y_{LJ} \\ z_{LJ} \end{bmatrix} \quad (161)$$

Uma vez que $\gamma_3, \gamma_4, \theta_L$ e θ_U foram calculados, pode-se seguir:

$$\theta_1 = \arctan(y_{LJ}, x_{LJ}) \quad (162)$$

$$\theta_2 = \arccos\left(\frac{z_{LJ}}{d_3}\right) \quad (163)$$

$$\theta_3 = \text{atan2}(\sin(\theta_3), \cos(\theta_3)) \quad (164)$$

Com $\sin(\theta_3)$ e $\cos(\theta_3)$ calculados de:

$$\begin{bmatrix} \cos(\theta_1) \cos(\theta_2) & -\sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2) & \cos(\theta_1) \\ -\sin(\theta_2) & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix} = \left(\begin{bmatrix} x_{LJ} \\ y_{LJ} \\ z_{LJ} \end{bmatrix} - \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \right) \frac{1}{a_3} \quad (165)$$

$$\theta_4 = (\theta_U + \theta_L), \text{ cotovelo para fora} \quad (166)$$

$$\theta_4 = -(\theta_U + \theta_L), \text{ cotovelo para dentro} \quad (167)$$

Os ângulos θ_5 e θ_6 podem ser calculados a partir do deslocamento da posição desejada da garra em relação a posição desejada do punho, representada no sistema de coordenadas 4, ou seja:

$$\begin{aligned} {}^4D_{garra}^5 &= {}^4T_{base} \text{ }^{base}P_{garra}^d - {}^4T_{base} \text{ }^{base}P_5^d \\ &= {}^4R_{base} (\text{ }^{base}P_{garra}^d - \text{ }^{base}P_5^d) = \begin{bmatrix} x_D \\ y_D \\ z_D \\ 1 \end{bmatrix} \end{aligned} \quad (168)$$

E:

$$\theta_5 = \text{atan2}(y_D, x_D) + \pi, \text{ cotovelo para fora} \quad (169)$$

$$\theta_5 = \text{atan2}(y_D, x_D), \text{ cotovelo para dentro} \quad (170)$$

$$\theta_6 = \text{atan2}\left(z_D, \sqrt{x_D^2 + y_D^2}\right) - \frac{\pi}{2}, \text{ cotovelo para fora} \quad (171)$$

$$\theta_6 = \pi - \text{atan2}\left(z_D, \sqrt{x_D^2 + y_D^2}\right), \text{ cotovelo para dentro} \quad (172)$$

Observando que $[x_D \ y_D \ z_D]^T$ pode ser calculado de (168). Para proceder com o cálculo de θ_7 , tem-se de (130):

$${}^{base}R_{garra}^d = {}^{base}R_0 {}^0R_6^d {}^6R_7^d {}^7R_{garra} \quad (173)$$

E deve-se observar que ${}^{base}R_{garra}^d$ é especificado e conhecido, ${}^{base}R_0$ e ${}^7R_{garra}$ são constantes e conhecidos e ${}^0R_6^d$ pode ser calculado a partir de $\theta_1 \cdots \theta_6$ que já foram determinados. E de (173) pode-se obter:

$${}^6R_7^d = {}^6R_0^d {}^{base}R_{garra}^d {}^{garra}R_7 \quad (174)$$

Por outro lado, dado que $a_7 = \alpha_7 = 0$ tem-se:

$${}^6R_7^d = \begin{bmatrix} \cos(\theta_7) & -\sin(\theta_7) & 0 & 0 \\ \sin(\theta_7) & \cos(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & 0 & 0 \\ r_{21} & r_{22} & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (175)$$

e portanto

$$\theta_7 = \text{atan2}(r_{21}, r_{11}) \quad (176)$$

2.4 Modelo Dinâmico do Manipulador Barrett WAM

O modelo dinâmico do manipulador Barrett WAM será desenvolvido utilizando-se o equacionamento de Lagrange-Euler, que em conjunto com a notação de Denavit-Hartenberg propicia a obtenção de um algoritmo de descrição de forma compacta para as equações de movimento do manipulador (FU; GONZALES; LEE, 1987). O algoritmo proposto é representado por operações matriciais e facilita tanto a implementação como a análise computacional.

Para obter o modelo dinâmico para um manipulador de n graus de liberdade utiliza-se o equacionamento de Lagrange-Euler:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1, 2, \dots, n \quad (177)$$

onde,

L - Função de Lagrange a qual representa a diferença entre a energia cinética T e a energia potencial P ($L = T - P$);

T - Energia cinética total do manipulador;

P - Energia potencial total do manipulador;

q_i - Coordenadas generalizadas do manipulador;

τ_i - Torque generalizado aplicado no manipulador na junta i para acionamento do elo i .

2.4.1 Velocidades de Juntas

O equacionamento de Lagrange-Euler exige o conhecimento da energia cinética do sistema, assim, faz-se necessário também o conhecimento das velocidades envolvidas. Sendo conhecido um ponto fixo do manipulador dado por:

$${}^i r_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (178)$$

sua velocidade pode ser obtida conforme expressão (179).

$${}^0 v_i \equiv v_i = \frac{d}{dt} ({}^0 r_i) = \frac{d}{dt} ({}^0 A_i {}^i r_i) = \left(\sum_{j=1}^i \frac{\partial {}^0 A_i}{\partial q_j} \dot{q}_j \right) {}^i r_i \quad (179)$$

A matriz ${}^0 A_i$ tem a forma:

$${}^{i-1} A_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

para o caso de juntas rotacionais. E:

$${}^{i-1} A_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

para o caso de juntas prismáticas.

A derivada parcial de ${}^0 A_i$ em relação a q_j pode ser calculada pela utilização do operador Q , que para o caso de juntas rotacionais tem a forma:

$$Q = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

e para o caso de juntas prismáticas:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Assim, pode-se escrever:

$$\frac{\partial^0 A_i}{\partial q_j} = \begin{cases} {}^0 A_1^1 A_2 \cdots {}^{j-2} A_{j-1} Q_j^{j-1} A_j \cdots {}^{i-1} A_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases}$$

Para simplificação pode-se definir $U_{ij} \triangleq \frac{\partial^0 A_i}{\partial q_j}$, e assim:

$$U_{ij} = \begin{cases} {}^0 A_{j-1} Q_j^{j-1} A_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases}$$

A velocidade de um ponto fixo do manipulador pode ser reescrita na forma:

$$v_i = \left(\sum_{j=1}^i U_{ij} \dot{q}_j \right) {}^i r_i \quad (180)$$

O efeito da interação entre as juntas do manipulador pode ser expresso por:

$$\frac{\partial U_{ij}}{\partial q_k} \triangleq U_{ijk} = \begin{cases} {}^0 A_{j-1} Q_j^{j-1} A_{k-1} Q_k^{k-1} A_i & \text{para } i \geq k \geq j \\ {}^0 A_{k-1} Q_k^{k-1} A_{j-1} Q_j^{j-1} A_i & \text{para } i \geq j \geq k \\ 0 & \text{para } i < j \text{ ou } i < k \end{cases} \quad (181)$$

que pode ser interpretada como os efeitos das juntas j e k para cada ligamento i .

2.4.2 Energia Cinética do Manipulador

Calculada a velocidade da junta de cada um dos elos do manipulador, é possível realizar-se o cálculo da energia cinética dos mesmos. Seja T_i a energia cinética do elo i com $i = 1, 2, \dots, n$ e dT_i a energia cinética de uma partícula de massa diferencial dm , pode-se escrever

$$dT_i = \frac{1}{2} (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2) dm = \frac{1}{2} T_r (v_i v_i^T) dm$$

$$dT_i = \frac{1}{2} T_r \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} ({}^i r_i dm {}^i r_i^T) U_{ir}^T \dot{q}_p \dot{q}_r \right]$$

A energia cinética total será obtida pela soma da energia cinética de cada um dos termos diferenciais sendo representada assim pela integral

$$T_i = \int dT_i = \frac{1}{2} T_r \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} \left(\int {}^i r_i {}^i r_i^T dm \right) U_{ir}^T \dot{q}_p \dot{q}_r \right]$$

com o termo $\int {}^i r_i {}^i r_i^T dm$ representando o momento de inércia do ligamento i . Assim pode-se definir

$$J_i = \int {}^i r_i {}^i r_i^T dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix}$$

e escrever para a energia cinética do manipulador

$$T = \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i [T_r (U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r] \quad (182)$$

2.4.3 Energia Potencial do Manipulador

A energia potencial do manipulador será representada pela letra P , e para cada elo do manipulador tem-se:

$$P_i = -m_i g {}^0 \bar{r}_i = -m_i g ({}^0 A_i {}^i \bar{r}_i)$$

com g representando a gravidade. Calculando agora para cada elo do manipulador têm-se que:

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n -m_i g ({}^0 A_i {}^i \bar{r}_i)$$

2.4.4 Equações de Movimento do Manipulador

Conforme definição da função de Lagrange tem-se que $L = T - P$ que permite escrever:

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i [T_r (U_{ij} J_i U_{ik}^T) \dot{q}_j \dot{q}_k] + \sum_{i=1}^n m_i g ({}^0 A_i {}^i \bar{r}_i) \quad (183)$$

E aplicando o equacionamento de Lagrange-Euler em (183) obtém-se:

$$\begin{aligned} \tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} &= \sum_{j=1}^n \sum_{k=1}^j T_r (U_{jk} J_j U_{ji}^T) \ddot{q}_k + \\ &\sum_{j=1}^n \sum_{k=1}^j \sum_{m=1}^j T_r (U_{jkm} J_j U_{ji}^T) \dot{q}_k \dot{q}_m - \sum_{j=1}^n m_j g U_{ji}^j \bar{r}_j \end{aligned} \quad (184)$$

que para $i = 1, 2, \dots, n$ pode ser expressa de forma simplificada como:

$$\tau_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m + c_i \quad (185)$$

ou ainda na forma matricial

$$\tau(t) = D(q(t))\ddot{q}(t) + h(q(t), \dot{q}(t)) + c(q(t)) \quad (186)$$

Ainda, $D(q(t))$ representa uma matriz simétrica ($n \times n$) relacionada aos elementos inerciais do manipulador e dada por:

$$D_{ik} = \sum_{j=\max(i,k)}^n \text{Tr}(U_{jk}J_jU_{ji}^T) \quad i, k = 1, 2, \dots, n \quad (187)$$

$h(q(t), \dot{q}(t))$ é um vetor de dimensões $n \times 1$ contendo informações sobre as forças centrífugas e de Coriolis com os elementos dados por:

$$\begin{aligned} h(q, \dot{q}) &= [h_1 \quad h_2 \quad \dots \quad h_n]^T \\ h_i &= \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m \quad i, k = 1, 2, \dots, n \\ h_{ikm} &= \sum_{j=\max(i,k,m)}^n \text{Tr}(U_{jk}J_jU_{ji}^T) \quad i, k, m = 1, 2, \dots, n \end{aligned} \quad (188)$$

E $c(q(t))$ é um vetor relativo à força gravitacional com dimensões $n \times 1$ dado por:

$$\begin{aligned} c(q) &= [c_1 \quad c_2 \quad \dots \quad c_n]^T \\ c_i &= \sum_{j=1}^n (-m_j g U_{ji}^j \bar{r}_j) \end{aligned} \quad (189)$$

2.5 Modelo Cinemático do Manipulador Móvel

O modelo cinemático do manipulador móvel pode ser obtido utilizando-se uma transformação de coordenadas que relacione o modelo cinemático do manipulador com o modelo cinemático da base móvel.

De (130) tem-se a transformação de coordenadas do manipulador Barrett WAM da garra para a base. Para formar o manipulador móvel, propõe-se a instalação do manipulador Barrett WAM sobre o chassi do robô móvel Twil, assim pode-se definir uma transformação que relacione este acoplamento na forma:

$$\text{chassi}T_{base} = \begin{bmatrix} \cos(\theta_{base}) & -\sin(\theta_{base}) & 0 & x_{base} \\ \sin(\theta_{base}) & \cos(\theta_{base}) & 0 & y_{base} \\ 0 & 0 & 1 & z_{base} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (190)$$

onde x_{base} , y_{base} e z_{base} indicam a posição do sistema de coordenadas da base do manipulador em relação ao sistema de coordenadas do topo da plataforma do robô móvel. As medidas em relação aos eixos x e y (BARRETT TECHNOLOGY, 2006), indicam que

o sistema de coordenadas está referenciado em uma das arestas da base do manipulador. Assim, para centralizar o manipulador em relação ao robô móvel é necessário que se considerem estes deslocamentos. Já a medida relativa ao eixo z se refere à espessura da chapa metálica do disco superior do robô móvel T_{wil} e ao fato de que o sistema de coordenadas está localizado no plano central em relação à espessura da chapa.

Faz-se necessário também definir uma relação entre o sistema da base-chassi em relação ao sistema inercial de coordenadas. Assim, da transformação do robô móvel T_{wil} (2) pode-se escrever:

$${}^0T_{chassi} = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) & 0 & x_c \\ \sin(\theta_c) & \cos(\theta_c) & 0 & y_c \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (191)$$

E então pode-se escrever para o modelo cinemático da manipulador móvel que:

$${}^0T_{garra} = {}^0T_{chassi} {}^{chassi}T_{base} {}^{base}T_{garra} \quad (192)$$

3 PROPOSTA DE CONTROLE

3.1 Introdução

Neste capítulo será apresentada uma proposta de controle para o manipulador móvel. Serão apresentadas estratégias de controle utilizando linearização, controle não suave, e *backstepping* para controle da base móvel, bem como uma análise de estabilidade do sistema utilizando a teoria de Lyapunov. Com relação ao controle do manipulador, apresenta-se a lei de torque calculado. Ainda, será apresentado um experimento para identificação de parâmetros necessários no modelo da base móvel.

3.2 Estratégias de Controle

3.2.1 Linearização por Realimentação da Dinâmica

Utilizando a realimentação de estados:

$$\tau_0 = G^\dagger(\beta)(H(\beta)v + f(\beta, u)) \quad (193)$$

onde $G^\dagger(\beta)$ denota uma pseudo-inversa à esquerda de $G(\beta)$. O modelo dinâmico de configuração de um robô móvel com rodas pode ser simplificado, assumindo a forma:

$$\begin{cases} \dot{x} = B(x)u \\ \dot{u} = v \end{cases} \quad (194)$$

Para o robô móvel Twil a realimentação de estados proposta em (193) pode ser calculada, verificando que no presente caso G é inversível, e sendo assim tem-se que:

$$\tau = G^{-1}(v + f(u)) \quad (195)$$

com:

$$G^{-1} = \begin{bmatrix} \frac{1}{2K_7} & \frac{1}{2K_8} \\ \frac{1}{2K_7} & -\frac{1}{2K_8} \end{bmatrix} \quad (196)$$

3.2.2 Transformação de coordenadas

É de interesse estabilizar o robô em um ponto arbitrário x_r , para qualquer posição e orientação $[x_{r1} \ x_{r2} \ x_{r3}]^T$. Isto pode ser realizado através da transformação de coordenadas $\bar{x}(x, x_r)$, estabelecendo um novo sistema referencial de coordenadas $\{X_{r1}, X_{r2}\}$ na posição de referência $[x_{r1} \ x_{r2}]^T$ com ângulo x_{r3} assim como apresentado na Figura 19.

Assim, a transformação de coordenadas de $\{X_1, X_2\}$ para $\{X_{r1}, X_{r2}\}$ consiste em uma translação e uma rotação do ângulo x_{r3} . Pode-se verificar que $\bar{x}_3 = x_3 - x_{r3}$ e então, a transformação de coordenadas $\bar{x}(\cdot, \cdot)$ é obtida por:

$$\bar{x} = \begin{bmatrix} R(x_{r3}) & 0 \\ 0 & 1 \end{bmatrix} (x - x_r) \quad (197)$$

onde $R(x_{r3})$ é uma matriz de rotação de duas dimensões, ou seja:

$$R(x_{r3}) = \begin{bmatrix} \cos x_{r3} & \sin x_{r3} \\ -\sin x_{r3} & \cos x_{r3} \end{bmatrix} \quad (198)$$

Portanto, para o sistema $\dot{\bar{x}} = f(\bar{x}, g(\bar{x}))$ sendo estável em $\bar{x} = 0$, tem-se $\dot{x} = f(x, g(x))$ estável em $x = 0$. Assim, para estabilizar o sistema em um ponto arbitrário qualquer x_r utilizando uma lei de controle g que conduz o estado à origem, basta utilizar $g(\bar{x})$.

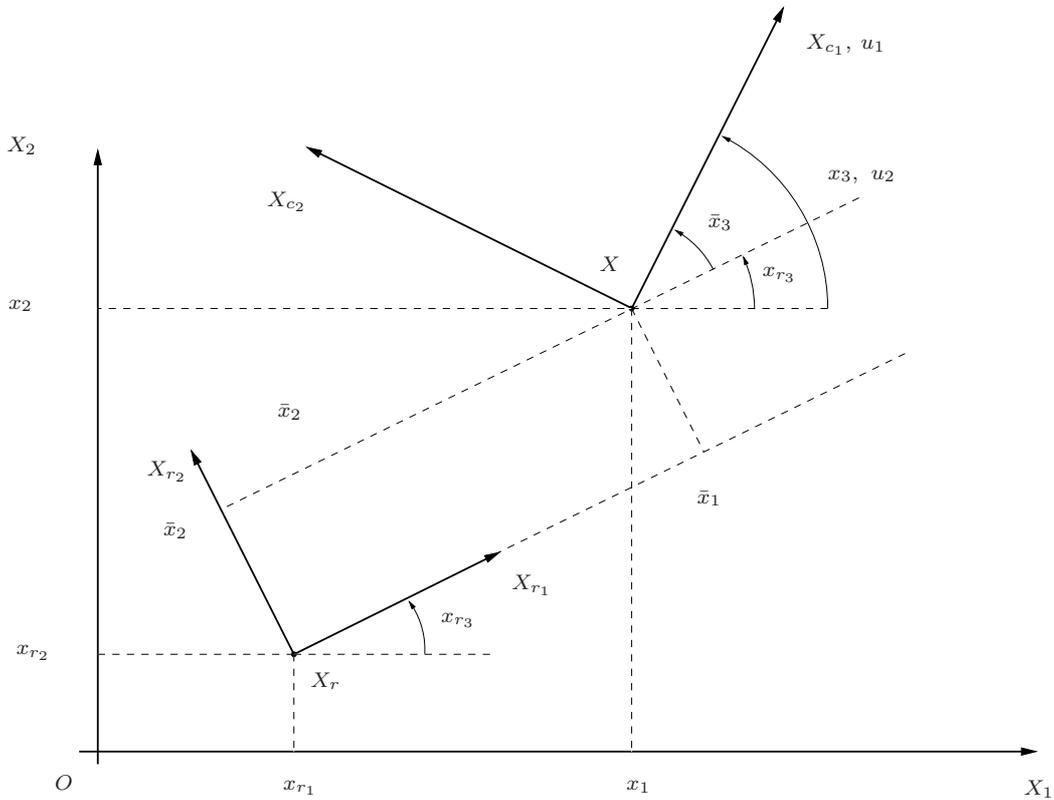


Figura 19: Sistema de coordenadas do robô móvel Twil com relação a um sistema de referência.

3.2.3 Controlador não suave

Robôs móveis com sistema de acionamento diferencial (caso do robô móvel Twil) são sistemas não-holonômicos (CAMPION; BASTIN; D'ANDRÉA-NOVEL, 1993). Uma importante definição sobre o controle de sistemas não-holonômicos foi feita por Brockett (BROCKETT, 1982) que demonstrou que não é possível estabilizar assintoticamente um sistema em um determinado ponto arbitrário através de uma lei de controle suave com realimentação e invariante no tempo, por mais que o sistema seja controlável (ASTOLFI, 1994).

Os meios para estabilizar assintoticamente um sistema conforme as condições de Brockett são sistemas de controle do tipo variantes no tempo (POMET et al., 1992; TEEL; MURRAY; WALSH, 1995; GODHAVN; EGELAND, 1997; REHMAN; RAFIQ; RAZA, 2011), controle não-suave (ASTOLFI, 1994; SØRDALEN, 1993; CANUDAS DE WIT; SØRDALEN, 1992) e leis de controle híbridas (LUCIBELLO; ORIOLO, 2001).

Neste trabalho, será obtido um controlador não suave baseado na transformação não suave de coordenadas (197), com restrições que estão associadas à cinemática, mas não à dinâmica do robô móvel Twil. Considerando a transformação de coordenadas (BARROS; LAGES, 2012):

$$e = \sqrt{\bar{x}_1^2 + \bar{x}_2^2} \quad (199)$$

$$\psi = \text{atan2}(\bar{x}_2, \bar{x}_1) \quad (200)$$

$$\alpha = \bar{x}_3 - \psi \quad (201)$$

$$\eta_1 = u_1 \quad (202)$$

$$\eta_2 = u_2 \quad (203)$$

(87) pode ser reescrita na forma:

$$\begin{cases} \dot{e} = \cos \alpha \eta_1 \\ \dot{\psi} = \frac{\sin \alpha}{e} \eta_1 \\ \dot{\alpha} = -\frac{\sin \alpha}{e} \eta_1 + \eta_2 \\ \dot{\eta}_1 = v_1 \\ \dot{\eta}_2 = v_2 \end{cases} \quad (204)$$

Para o projeto do controlador não suave, serão utilizadas apenas as primeiras três expressões de (204), tomando como entrada a variável η . Na seção 3.2.4 será utilizada a técnica de *backstepping* para voltar a considerar a entrada como sendo v assim como na expressão (204). Considerando agora:

$$\begin{cases} \dot{e} = \cos \alpha \eta_1 \\ \dot{\psi} = \frac{\sin \alpha}{e} \eta_1 \\ \dot{\alpha} = -\frac{\sin \alpha}{e} \eta_1 + \eta_2 \end{cases} \quad (205)$$

será utilizada a análise de Lyapunov para verificar a estabilidade do sistema. Escolhendo como função candidata:

$$V = \frac{1}{2} (\lambda_1 e^2 + \lambda_2 \alpha^2 + \lambda_3 \psi^2) \quad (206)$$

tem-se que:

$$\dot{V} = \lambda_1 e \dot{e} + \lambda_2 \alpha \dot{\alpha} + \lambda_3 \psi \dot{\psi} \quad (207)$$

$$\dot{V} = \lambda_1 e \cos \alpha \eta_1 - \lambda_2 \alpha \frac{\sin \alpha}{e} \eta_1 + \lambda_2 \alpha \eta_2 + \lambda_3 \psi \frac{\sin \alpha}{e} \eta_1 \quad (208)$$

Agora, para que \dot{V} seja positiva definida pode-se fazer a escolha das entradas de controle:

$$\eta_1 \triangleq -\gamma_1 e \cos \alpha \quad (209)$$

e

$$\eta_2 \triangleq -\gamma_2 \alpha - \gamma_1 \sin \alpha \cos \alpha + \gamma_1 \lambda_3 \psi \frac{\sin \alpha}{\lambda_2 \alpha} \cos \alpha \quad (210)$$

assim, pode-se reescrever (208) como:

$$\dot{V} = -\gamma_1 \lambda_1 e^2 \cos^2 \alpha - \gamma_2 \lambda_2 \alpha^2 \leq 0 \quad (211)$$

Lembrando que V é limitada inferiormente, e como pode-se observar de (211), V é não crescente, ainda \dot{V} é uniformemente contínua pois \ddot{V} é limitada. Baseando-se no lema de Barbalat (POPOV, 1973) tem-se que $\dot{V} \rightarrow 0$ e de (211) pode-se escrever $-\gamma_1 \lambda_1 e^2 \cos^2 \alpha - \gamma_2 \lambda_2 \alpha^2 \rightarrow 0$. Uma vez que γ_2 e λ_2 , são termos positivos, para que $\gamma_2 \lambda_2 \alpha^2 \rightarrow 0$ é necessário que $\alpha \rightarrow 0$. Fazendo $\alpha = 0$ em $-\gamma_1 \lambda_1 e^2 \cos^2 \alpha - \gamma_2 \lambda_2 \alpha^2 \rightarrow 0$ tem-se que $-\gamma_1 \lambda_1 e^2 \rightarrow 0$ e com γ_1 e λ_1 sendo termos positivos pode-se concluir que $e \rightarrow 0$.

Ainda faz-se necessário verificar a convergência para zero da variável ψ . Para isto segue a análise do sistema em malha fechada, dado por:

$$\begin{cases} \dot{e} = -\gamma_1 e \cos^2 \alpha \\ \dot{\psi} = -\gamma_1 \sin \alpha \cos \alpha \\ \dot{\alpha} = -\gamma_2 \alpha + \gamma_1 \frac{\lambda_3}{\lambda_2} \psi \frac{\sin \alpha}{\alpha} \cos \alpha. \end{cases} \quad (212)$$

Tem-se que $\ddot{\alpha} < \infty$ e conseqüentemente $\dot{\alpha}$ é uniformemente contínua. Como $\alpha \rightarrow 0$ pelo lema de Barbalat tem-se $\dot{\alpha} \rightarrow 0$. Agora com $\alpha \rightarrow 0$ e $\dot{\alpha} \rightarrow 0$ pode-se reescrever a terceira expressão de (212) como:

$$\gamma_1 \frac{\lambda_3}{\lambda_2} \psi \frac{\sin \alpha}{\alpha} \cos \alpha = 0 \quad (213)$$

e uma vez que γ_1 , λ_2 e λ_3 são termos positivos, pode-se concluir que:

$$\psi \rightarrow 0. \quad (214)$$

3.2.4 Backstepping

A técnica de *backstepping* permite o desenvolvimento de controladores que buscam estabilizar sistemas dinâmicos não lineares, estando dentre as técnicas mais importantes para controle de sistemas não lineares (TANNER; KYRIAKOPOULOS, 2003) sendo utilizada em diversas aplicações, a exemplo do trabalho apresentado em FIERRO; LEWIS (1995).

Ainda que (209-210) possam estabilizar a primeira expressão de (87), as mesmas não podem estabilizar (87) totalmente uma vez que sua entrada é v e não u . Pode-se observar entretanto que (87) pode ser visto como uma cascata entre dois subsistemas e neste caso é possível utilizar a técnica de *backstepping* (KOKOTOVIĆ, 1992) para obter uma expressão para v a partir de u .

Aplicando as transformações (199-201,197) à (87) é possível escrever:

$$\dot{e} = \cos \alpha u_1 \quad (215)$$

$$\dot{\psi} = \frac{\sin \alpha}{e} u_1 \quad (216)$$

$$\dot{\alpha} = -\frac{\sin \alpha}{e} u_1 + u_2 \quad (217)$$

$$\dot{u}_1 = v_1 \quad (218)$$

$$\dot{u}_2 = v_2 \quad (219)$$

E somando-se $\cos \alpha(\eta_1 - \eta_1)$ a (215), $\frac{\sin \alpha}{e}(\eta_1 - \eta_1)$ a (216) e $-\frac{\sin \alpha}{e}(\eta_1 - \eta_1) + (\eta_2 - \eta_2)$ a (217), as expressões permanecem inalteradas:

$$\begin{cases} \dot{e} = \cos \alpha u_1 + \cos \alpha(\eta_1 - \eta_1) \\ \dot{\psi} = \frac{\sin \alpha}{e} u_1 + \frac{\sin \alpha}{e}(\eta_1 - \eta_1) \\ \dot{\alpha} = -\frac{\sin \alpha}{e} u_1 + u_2 - \frac{\sin \alpha}{e}(\eta_1 - \eta_1) + (\eta_2 - \eta_2) \\ \dot{u}_1 = v_1 \\ \dot{u}_2 = v_2 \end{cases} \quad (220)$$

podendo as mesmas serem rearranjadas na forma:

$$\begin{cases} \dot{e} = \cos \alpha \eta_1 + \cos \alpha(u_1 - \eta_1) \\ \dot{\psi} = \frac{\sin \alpha}{e} \eta_1 + \frac{\sin \alpha}{e}(u_1 - \eta_1) \\ \dot{\alpha} = -\frac{\sin \alpha}{e} \eta_1 + \eta_2 - \frac{\sin \alpha}{e}(u_1 - \eta_1) + (u_2 - \eta_2) \\ \dot{u}_1 = v_1 \\ \dot{u}_2 = v_2 \end{cases} \quad (221)$$

e definindo-se:

$$e_1 \triangleq u_1 - \eta_1 \quad (222)$$

$$e_2 \triangleq u_2 - \eta_2 \quad (223)$$

$$\bar{v}_1 \triangleq v_1 - \dot{\eta}_1 \quad (224)$$

$$\bar{v}_2 \triangleq v_2 - \dot{\eta}_2 \quad (225)$$

obtem-se:

$$\begin{cases} \dot{e} = \cos \alpha \eta_1 + \cos \alpha e_1 \\ \dot{\psi} = \frac{\sin \alpha}{e} \eta_1 + \frac{\sin \alpha}{e} e_1 \\ \dot{\alpha} = -\frac{\sin \alpha}{e} \eta_1 + \eta_2 - \frac{\sin \alpha}{e} e_1 + e_2 \\ \dot{e}_1 = \bar{v}_1 \\ \dot{e}_2 = \bar{v}_2 \end{cases} \quad (226)$$

Agora, substituindo-se η_1 e η_2 de (209) e (210) pode-se escrever:

$$\begin{cases} \dot{e} = -\gamma_1 e \cos \alpha^2 + \cos \alpha e_1 \\ \dot{\psi} = -\gamma_1 \sin \alpha \cos \alpha + \frac{\sin \alpha}{e} e_1 \\ \dot{\alpha} = -\gamma_2 \alpha + \gamma_1 \frac{\lambda_3}{\lambda_2} \cos \alpha \frac{\sin \alpha}{\alpha} \psi - \frac{\sin \alpha}{e} e_1 + e_2 \\ \dot{e}_1 = \bar{v}_1 \\ \dot{e}_2 = \bar{v}_2 \end{cases} \quad (227)$$

Seja a seguinte candidata à função de Lyapunov:

$$V_1 = \frac{1}{2} (\lambda_1 e^2 + \lambda_2 \alpha^2 + \lambda_3 \psi^2 + \lambda_4 e_1^2 + \lambda_5 e_2^2) \quad (228)$$

Sua derivada temporal é dada por:

$$\dot{V}_1 = \lambda_1 e \dot{e} + \lambda_2 \alpha \dot{\alpha} + \lambda_3 \psi \dot{\psi} + \lambda_4 e_1 \dot{e}_1 + \lambda_5 e_2 \dot{e}_2 \quad (229)$$

e substituindo-se agora as expressões (227) em (229) resulta em:

$$\begin{aligned} \dot{V}_1 &= -\gamma_1 \lambda_1 e^2 \cos^2 \alpha - \gamma_2 \lambda_2 \alpha^2 + \lambda_1 e \cos \alpha e_1 \\ &\quad - \lambda_2 \alpha \frac{\sin \alpha}{e} e_1 + \lambda_3 \psi \frac{\sin \alpha}{e} e_1 \\ &\quad + \lambda_4 e_1 \bar{v}_1 + \lambda_5 e_2 \bar{v}_2 \end{aligned} \quad (230)$$

Então, escolhendo-se:

$$\begin{aligned} \bar{v}_1 &\triangleq -\gamma_4 e_1 - \frac{\lambda_1}{\lambda_4} \cos \alpha + \frac{\lambda_2}{\lambda_4} \alpha \frac{\sin \alpha}{e} \\ &\quad - \frac{\lambda_3}{\lambda_4} \psi \frac{\sin \alpha}{e} \end{aligned} \quad (231)$$

$$\bar{v}_2 \triangleq -\gamma_5 e_2 - \frac{\lambda_2}{\lambda_5} \alpha \quad (232)$$

obtem-se como resultado:

$$\begin{aligned} \dot{V}_1 &= -\gamma_1 \lambda_1 e^2 \cos^2 \alpha - \gamma_2 \lambda_2 \alpha^2 - \gamma_4 \lambda_4 e_1^2 \\ &\quad - \gamma_5 \lambda_5 e_2^2 \leq 0 \end{aligned} \quad (233)$$

que prova que V_1 é de fato uma função de Lyapunov para o sistema (227). Além disto, como \dot{V}_1 é uniformemente contínua, tem-se do lema de Barbalat (POPOV, 1973) que $\dot{V}_1 \rightarrow 0$, o que implica que $e \rightarrow 0$, $\alpha \rightarrow 0$, $e_1 \rightarrow 0$ e $e_2 \rightarrow 0$. Resta ainda provar que ψ converge para zero. Aplicando o lema de Barbalat para $\dot{\alpha}$ segue-se que $\dot{\alpha} \rightarrow 0$ em (227), o que implica que $\psi \rightarrow 0$.

Assim, a lei de controle para o sistema (87) é dada por:

$$v_1 = \bar{v}_1 + \dot{\eta}_1 \quad (234)$$

$$v_2 = \bar{v}_2 + \dot{\eta}_2 \quad (235)$$

e com:

$$\begin{aligned}\dot{\eta}_1 &= \gamma_1^2 e \cos^3 \alpha - \gamma_1 \gamma_2 e \alpha \sin \alpha \\ &+ \gamma_1^2 \frac{\lambda_3}{\lambda_2} e \cos \alpha \frac{\sin^2 \alpha}{\alpha} \psi\end{aligned}\quad (236)$$

$$\begin{aligned}\dot{\eta}_2 &= \gamma_2^2 \alpha - 2\gamma_1 \gamma_2 \frac{\lambda_3}{\lambda_2} \psi \frac{\sin \alpha}{\alpha} \cos \alpha \\ &+ \gamma_1 \gamma_2 \alpha \cos^2 \alpha - \gamma_1^2 \frac{\lambda_3}{\lambda_2} \cos^3 \alpha \frac{\sin \alpha}{\alpha} \psi \\ &- \gamma_1 \gamma_2 \alpha \sin^2 \alpha + \gamma_1^2 \frac{\lambda_3}{\lambda_2} \cos \alpha \frac{\sin^3 \alpha}{\alpha} \psi \\ &- \gamma_1^2 \frac{\lambda_3}{\lambda_2} \cos^2 \alpha \frac{\sin^2 \alpha}{\alpha} - \gamma_1 \gamma_2 \frac{\lambda_3}{\lambda_2} \psi \\ &+ \gamma_1^2 \frac{\lambda_3^2}{\lambda_2^2} \cos^3 \alpha \frac{\sin \alpha}{\alpha^2} \psi^2 + \gamma_1^2 \frac{\lambda_3^2}{\lambda_2^2} \cos \alpha \frac{\sin^3 \alpha}{\alpha^2} \psi^2 \\ &+ \gamma_1^2 \frac{\lambda_3^2}{\lambda_2^2} \cos^2 \alpha \frac{\sin^2 \alpha}{\alpha^3} \psi^2\end{aligned}\quad (237)$$

tem-se:

$$\begin{aligned}v_1 &= -\gamma_4 e_1 - \frac{\lambda_1}{\lambda_4} \cos \alpha + \frac{\lambda_2}{\lambda_4} \alpha \frac{\sin \alpha}{e} \\ &+ \gamma_1^2 e \cos^3 \alpha - \gamma_1 \gamma_2 e \alpha \sin \alpha \\ &+ \gamma_1^2 \frac{\lambda_3}{\lambda_2} e \cos \alpha \frac{\sin^2 \alpha}{\alpha} \psi\end{aligned}\quad (238)$$

$$\begin{aligned}v_2 &= -\gamma_5 e_2 - \frac{\lambda_2}{\lambda_5} \alpha \\ &+ \gamma_2^2 \alpha - 2\gamma_1 \gamma_2 \frac{\lambda_3}{\lambda_2} \psi \frac{\sin \alpha}{\alpha} \cos \alpha \\ &+ \gamma_1 \gamma_2 \alpha \cos^2 \alpha - \gamma_1^2 \frac{\lambda_3}{\lambda_2} \cos^3 \alpha \frac{\sin \alpha}{\alpha} \psi \\ &- \gamma_1 \gamma_2 \alpha \sin^2 \alpha + \gamma_1^2 \frac{\lambda_3}{\lambda_2} \cos \alpha \frac{\sin^3 \alpha}{\alpha} \psi \\ &- \gamma_1^2 \frac{\lambda_3}{\lambda_2} \cos^2 \alpha \frac{\sin^2 \alpha}{\alpha} - \gamma_1 \gamma_2 \frac{\lambda_3}{\lambda_2} \psi \\ &+ \gamma_1^2 \frac{\lambda_3^2}{\lambda_2^2} \cos^3 \alpha \frac{\sin \alpha}{\alpha^2} \psi^2 + \gamma_1^2 \frac{\lambda_3^2}{\lambda_2^2} \cos \alpha \frac{\sin^3 \alpha}{\alpha^2} \psi^2 \\ &+ \gamma_1^2 \frac{\lambda_3^2}{\lambda_2^2} \cos^2 \alpha \frac{\sin^2 \alpha}{\alpha^3} \psi^2\end{aligned}\quad (239)$$

3.2.5 Identificação de parâmetros

O modelo (122) não é o mesmo descrito em URDF (*Unified Robot Description Format*), por ser mais simples e não incluir alguns detalhes construtivos do robô como peças com distribuição de massa não uniforme como a CPU, placas de controle, cabeamento,

etc. Para contornar a dificuldade de obter um modelo algébrico tal como (122) considerando estes detalhes, propõe-se a identificação destes parâmetros (AGUIRRE, 2007) a fim de garantir uma maior precisão nos dados utilizados para projeto do controlador do robô. Do modelo de configuração (122) do robô móvel Twil tem-se:

$$\dot{u} = \begin{bmatrix} 0 & K_5 \\ K_6 & 0 \end{bmatrix} \begin{bmatrix} u_1 u_2 \\ u_2^2 \end{bmatrix} + \begin{bmatrix} K_7 & K_7 \\ K_8 & -K_8 \end{bmatrix} \tau \quad (240)$$

sendo que (240) pode ser desmembrada em:

$$\dot{u}_1 = K_5 u_2^2 + K_7 (\tau_1 + \tau_2) \quad (241)$$

$$\dot{u}_2 = K_6 u_1 u_2 + K_8 (\tau_1 - \tau_2) \quad (242)$$

Assim, utilizando as duas expressões (241) e (242) pode-se identificar dois parâmetros em cada uma das expressões, sendo este processo mais simples que realizar a identificação simultânea dos quatro parâmetros K_5 , K_6 , K_7 e K_8 .

$$y_1(k+1) = \dot{u}_1(k) \simeq \frac{u_1(k+1) - u_1(k)}{T} = K_5 u_2^2(k) + K_7 (\tau_1(k) + \tau_2(k)) \quad (243)$$

$$y_1(k+1) = \begin{bmatrix} u_2^2(k) \\ \tau_1(k) + \tau_2(k) \end{bmatrix}^T \begin{bmatrix} K_5 \\ K_7 \end{bmatrix} \quad (244)$$

e definindo-se:

$$\phi_1(k) = \begin{bmatrix} u_2^2(k) \\ \tau_1(k) + \tau_2(k) \end{bmatrix} \quad (245)$$

$$\theta_1(k) = \begin{bmatrix} K_5 \\ K_7 \end{bmatrix} \quad (246)$$

pode-se reescrever (244) como:

$$y_1(k+1) = \phi_1^T(k) \theta_1(k) \quad (247)$$

$$y_2(k+1) = \dot{u}_2(k) \simeq \frac{u_2(k+1) - u_2(k)}{T} = K_6 u_1(k) u_2(k) + K_8 (\tau_1(k) - \tau_2(k)) \quad (248)$$

$$y_2(k+1) = \begin{bmatrix} u_1(k) u_2(k) \\ \tau_1(k) - \tau_2(k) \end{bmatrix}^T \begin{bmatrix} K_6 \\ K_8 \end{bmatrix} \quad (249)$$

e definindo-se:

$$\phi_2(k) = \begin{bmatrix} u_1(k)u_2(k) \\ \tau_1(k) - \tau_2(k) \end{bmatrix} \quad (250)$$

$$\theta_2(k) = \begin{bmatrix} K_6 \\ K_8 \end{bmatrix} \quad (251)$$

pode-se reescrever (249) como:

$$y_2(k+1) = \phi_2^T(k)\theta_2(k) \quad (252)$$

Utilizando o algoritmo de mínimos quadrados recursivo (AGUIRRE, 2007) pode-se estimar θ_i através de:

$$\hat{y}_i(n+1) = \phi_i^T(n)\hat{\theta}_i(n) \quad (253)$$

$$K_i(n) = \frac{P_i(n-1)\phi_i(n)}{1 + \phi_i^T(n)P_i(n-1)\phi_i(n)} \quad (254)$$

$$\hat{\theta}_i(n+1) = \hat{\theta}_i(n) + K_i(n)(y_i(n+1) - \hat{y}_i(n+1)) \quad (255)$$

$$P_i(n) = (I - K_i(n)\theta_i^T(n))P_i(n-1) \quad (256)$$

3.3 Controlador do robô móvel Twil

O controlador proposto para o robô Twil será implementado (no ROS) conforme diagrama de blocos apresentado na Figura 20, com o bloco da linearização sendo implementado com base na formulação da Seção 3.2.1, o bloco de transformação de coordenadas será implementado seguindo o desenvolvimento apresentado na Seção 3.2.2, o bloco de transformação descontínua será implementado seguindo o desenvolvimento apresentado nas Seções 3.2.2 e 3.2.3 e o bloco do controlador *backstepping* será implementado seguindo o desenvolvimento da Seção 3.2.4.

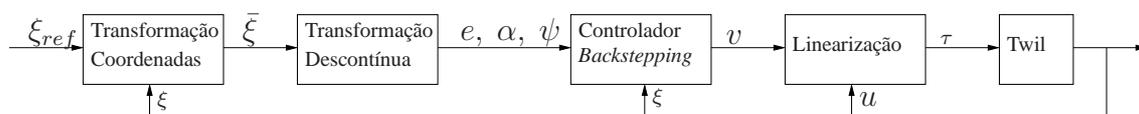


Figura 20: Diagrama de blocos do controlador a ser implementado para o robô móvel Twil.

3.4 Controle do manipulador Barrett WAM

Para o controle do manipulador Barrett WAM, será utilizada a lei de controle por torque calculado (FU; GONZALES; LEE, 1987):

$$\begin{aligned} \tau(t) = & D_n(q) [\ddot{q}_r(t) + K_d(\dot{q}_r(t) - \dot{q}(t)) + K_p(q_r(t) - q(t))] \\ & + h_n(q(t), \dot{q}(t)) + c_n(q(t)) \end{aligned} \quad (257)$$

onde q_r é o vetor de referência das variáveis de junta e K_d e K_p são respectivamente as matrizes de ganhos derivativo e proporcional do controlador. Substituindo $\tau(t)$ da lei de controle (257) em (186) tem-se:

$$\begin{aligned} D(q(t)) \ddot{q}(t) + h(q(t), \dot{q}(t)) + c(q(t)) = & \quad (258) \\ D_n(q) [\ddot{q}_r(t) + K_d(\dot{q}_r(t) - \dot{q}(t)) + K_p(q_r(t) - q(t))] + h_n(q(t), \dot{q}(t)) + c_n(q(t)) \end{aligned}$$

Se

$$\begin{aligned} D(q) &= D_n(q) \\ h(q(t), \dot{q}(t)) &= h_n(q(t), \dot{q}(t)) \\ c(q(t)) &= c_n(q(t)) \end{aligned}$$

e definindo-se $e(t) \triangleq q_r(t) - q(t)$ e $\dot{e}(t) \triangleq \dot{q}_r(t) - \dot{q}(t)$ a expressão (258) se reduz à forma:

$$D(q)[\ddot{e}(t) + K_d\dot{e}(t) + K_p e(t)] = 0 \quad (259)$$

Sendo a matriz $D(q)$ sempre não singular, as matrizes dos ganhos K_d e K_p podem ser escolhidas apropriadamente de forma que as raízes do polinômio característico de (259) possuam a parte real negativa e assim o vetor de erro $e(t)$ pode ser levado à zero assintoticamente.

4 IMPLEMENTAÇÃO NO ROS (*ROBOT OPERATING SYSTEM*)

4.1 Introdução

Neste Capítulo será apresentado o ROS (*Robot Operating System*), assim como os conceitos básicos sobre modelagem URDF (*Unified Robot Description Format*) necessários para criar o modelo do manipulador móvel. A estrutura dos pacotes criados para implementar os controladores no ROS é apresentada com uma descrição detalhada da implementação dentro do laço de tempo real do ROS bem como sua estrutura geral de controladores.

4.2 ROS

O ROS é um *framework* para desenvolvimento de *software* para robôs. É constituído por uma coleção de ferramentas, bibliotecas e convenções que buscam simplificar a tarefa de criar comportamentos complexos e robustos para robôs em diversas plataformas robóticas (QUIGLEY et al., 2009). Foi desenvolvido para promover a colaboração entre desenvolvedores de robótica, permitindo por exemplo que diferentes grupos de pesquisa possam cooperar através da troca de experiências reutilizando código.

O ROS é um sistema baseado em mensagens, sendo composto por bibliotecas escritas em linguagens como C++ e Python. A filosofia de criação do ROS é baseada no sistema operacional UNIX onde uma série de pequenas ferramentas são projetadas para trabalhar em conjunto. A origem do ROS deve-se a uma parceria entre a Indústria e Universidades (COUSINS, 2010). O ROS pode ser compreendido como um sistema distribuído (WONG, 2011), onde um determinado número de nós, cada um rodando determinada aplicação se comunica através de mensagens.

O ROS é um dos sistemas operacionais disponíveis para robôs mais utilizado (KERR; NICKELS, 2012) com trabalhos sendo desenvolvidos tanto no ambiente industrial (ROBOT OPERATING SYSTEM, 2014a) como no ambiente acadêmico a exemplo de cursos que ensinam conceitos básicos de robótica utilizando o ROS como um dos componentes de um laboratório virtual (CORRELL; WING; COLEMAN, 2013). A versão do ROS utilizada neste trabalho é a FUERTE.

4.2.1 Pacotes do ROS

A sistemática de operação do ROS está baseada nos pacotes, os quais contém por exemplo códigos-fonte e arquivos executáveis, oferecendo uma camada de abstração para programação onde os arquivos estão organizados de maneira apropriada e podem ser facil-

mente localizados (GITHUB, 2014). Sendo um sistema operacional baseado em pacotes sua principal característica é a de que arquivos podem referenciar outros arquivos de maneira simples, neste caso, definindo que determinado pacote A possui dependência de outro pacote B. A estrutura de um pacote está representada na Figura 21.

nome_do_pacote	
<code>src/</code> Códigos-fonte
<code>bin/</code> Arquivos executáveis
<code>lib/</code> Bibliotecas compartilhadas
<code>build/</code> Arquivos compilados
<code>launch/</code> Arquivos de inicialização do pacote
<code>include/</code> Arquivos de cabeçalho
<code>manifest.xml</code> Informações básicas do pacote
<code>CMakeList.txt</code> Definições sobre a compilação

Figura 21: Estrutura de um pacote no ROS.

4.2.2 Stacks do ROS

O *stack* é o principal mecanismo de distribuição de software utilizado no ROS, sendo que os pacotes do ROS são organizados em *stacks*. Assim como o objetivo da utilização dos pacotes é criar uma coletânea de códigos que facilitem sua reutilização, o objetivo de um *stack* é a simplificação do compartilhamento destes códigos (ROBOT OPERATING SYSTEM, 2014b).

Cada *stack* possui um número de versão associado podendo declarar dependência com relação a outros *stacks*, e esta dependências também possuem um número de versão garantindo maior estabilidade de desenvolvimento. Ainda, os *stacks* reúnem pacotes que de forma conjunta proporcionam funcionalidades como navegar entre diferentes *stacks* e manipular os mesmos.

A criação de um *stack* no ROS pode ser efetuada de forma manual ou utilizando a ferramenta *roscreeate-stack* de geração automática, sendo que a criação manual é extremamente simples uma vez que um *stack* é simplesmente um diretório dependente das variáveis de ambiente `ROS_ROOT` ou `ROS_PACKAGE_PATH` com uma arquivo *stack.xml* associado e todos os arquivos contidos neste diretório são considerados como parte do *stack*.

4.2.3 Nodos do ROS

Um nodo é um processo capaz de realizar tarefas como cálculos por exemplo. Em um robô por exemplo, estima-se utilizar uma grande quantidade de nodos, onde pode-se ter um nodo controlando os motores das rodas, um nodo realizando localização, um nodo atualizando a visualização gráfica do sistema, e assim sucessivamente (ROBOT OPERATING SYSTEM, 2014b). A utilização dos nodos no ROS traz uma série de benefícios ao sistema de forma geral, como por exemplo uma maior tolerância à falhas uma vez que problemas com um único nodo são isolados ao mesmo.

Ainda, a complexidade de código associada a um nodo é inferior quando comparada com uma implementação monolítica. Os detalhes de implementação dos nodos são ocultados de maneira apropriada uma vez que os mesmos fornecem uma API (*Application*

Programming Interface) simplificada que permite implementações alternativas mesmo em outras linguagens de programação.

4.2.4 Mecanismos de comunicação entre nodos do ROS

O ROS utiliza mecanismos que permitem que os nodos possam se comunicar. Dentre estes mecanismos estão os tópicos, os serviços e os *action servers*.

4.2.4.1 Tópicos

Os tópicos são um mecanismo *publisher/subscriber* através do qual os nós podem trocar mensagens (ROBOT OPERATING SYSTEM, 2014b), possuindo uma semântica anônima com relação a qual o nodo que publica ou subscreve a mensagem, assim o nodo não possui a informação de qual o seu parceiro na comunicação.

Ao invés disso o nodo que precisar de um determinado dado em específico deve subscrever no tópico apropriado, assim como nodos que publicam dados devem publicar no tópico apropriado. Um único tópico pode ter uma série de nodos publicando e subscrevendo ao mesmo tempo.

4.2.4.2 Serviços

Os nodos são destinados à transmissão unidirecional de dados, e para os casos onde há a necessidade de execução de alguma RPC por parte do nodo, ou seja, receber uma confirmação em relação a um pedido o nodo deve utilizar um serviço para tal fim (ROBOT OPERATING SYSTEM, 2014b).

O modelo *publisher/subscriber* de mensagens é extremamente flexível mas a grande quantidade de trocas unidirecionais não é apropriada para requisições do tipo RPC ou interações do tipo requisição/resposta que são geralmente solicitadas em sistemas distribuídos. Neste caso, o ROS oferece um serviço, que é um processo que possui uma estrutura do tipo requisição/resposta, permitindo também operações do tipo RPC entre nodos (ROBOT OPERATING SYSTEM, 2014b).

No ROS os serviços são identificados por um nome (uma *string*), permitindo que um cliente faça uma chamada ao serviço enviando uma mensagem e esperando pela resposta.

4.2.4.3 Action Servers

Em sistemas de grande porte baseados no ROS ocorrem situações onde um usuário pode querer enviar uma requisição a um nodo a fim de executar determinada tarefa, e também receber uma resposta referente a requisição enviada. Este procedimento pode ser obtido através de utilização de serviços.

Porém, nos casos em que um serviço exige um longo tempo de execução, o usuário pode querer cancelar a requisição enviada durante sua execução ou receber informações sobre o *status* de execução da requisição. Assim é possível a criação de *action servers* utilizando o pacote *actionlib* que fornece ferramentas para a criação dos mesmos.

Os *action servers* executam tarefas de longo período de execução que podem ser preemptadas e fornecem também uma interface para que clientes possam enviar solicitações.

4.2.5 Controladores do ROS

O ROS ainda não possui uma boa capacidade para implementação de controladores avançados assim como o proposto neste trabalho. A princípio, o ROS não é um sistema de tempo real apesar de que pode apresentar algumas características de tempo real se for

executado em um sistema *Linux* com *PREEMPT_RT kernel patch*. Mesmo neste caso, existe um único laço de tempo real onde todos os controladores serão supervisionados por um gerenciador de controladores (pertencente à classe *ControllerManager*).

Este laço de tempo real roda a uma taxa fixa de 1kHz, a qual não é adequada para atender todas as tarefas de tempo real em um sistema complexo. Neste caso, uma alternativa para sistemas que necessitem executar muitas tarefas de tempo real com diferentes taxas é a utilização do *framework* OROCOS (*Open Robot Control Software*) (BRUYNINCKX, 2001) como uma camada de baixo nível para implementação da etapa de tempo real do sistema.

Do ponto de vista da engenharia de controle, o ROS apresenta algumas particularidades quando se quer implementar um controlador. Uma delas é a nomenclatura, pois o que o ROS chama de controlador não é necessariamente um controlador na nomenclatura de um sistema de controle. No ROS um controlador é um *plugin* para o gerenciador de controladores que está implementando a interface de controle, a qual tipicamente possui uma interface para a *joint command interface* e, ou *joint state interface*. Observe que um controlador no ROS pode efetuar uma função que não é necessariamente a função de um controlador em um sistema de controle. Como exemplo têm-se o *joint_state_controller*, que por sua nomenclatura poderia ser interpretado como um controlador das juntas no espaço de estados, porém trata-se de um *publisher* dos valores das posições e velocidades das juntas.

Outro problema é que aparentemente a infraestrutura do ROS foi concebida para controladores com apenas uma entrada e uma saída (SISO - *Single-Input-Single-Output*) que gerenciam apenas um valor escalar de referência, o valor de um sensor e geram uma saída escalar.

Além disso, há discussões em fóruns de desenvolvedores do ROS que assumem a possibilidade de desenvolver controladores com múltiplas entradas e múltiplas saídas (MIMO - *Multiple-Input-Multiple-Output*) através da composição de controladores SISO. O que não se aplica. Leis avançadas de controle de robôs são intrinsecamente do tipo MIMO e não podem ser decompostas em um conjunto de leis do tipo SISO, sem falar no problema de sincronizar muitos controladores SISO, de forma a se comportarem como um único controlador MIMO. Entretanto neste trabalho, a implementação de um controlador no ROS é apresentada, buscando um melhor entendimento das capacidades e limitações do ROS com relação a implementação de controladores de baixo nível. Implementações similares utilizando o *framework* OROCOS para implementar capacidades de tempo real podem ser verificadas em IORIS; LAGES; SANTINI (2012); LAGES; IORIS; SANTINI (2014); SANTINI; LAGES (2010).

A Figura 22 mostra a arquitetura de baixo nível dos controladores no ROS. Os controladores são *plugins* carregados pelo gerenciador de controladores, o qual pode carregar, descarregar, ativar e desativar os controladores. A seção 4.2.6 traz o detalhamento da arquitetura apresentada na Figura 22.

Cada ciclo de controle é realizado a uma frequência de 1kHz. Em cada ciclo a função *update()* de cada um dos controladores ativos é invocada sequencialmente e assim cada controlador pode realizar sua tarefa dentro do ciclo. Se o controlador tiver que executar em uma taxa menor, o mesmo deverá implementar uma subamostragem independente. Do ponto de vista de controle digital, o modelo é o de um controlador de tempo contínuo implementado em um computador digital com uma taxa de amostragem tão rápida que os efeitos da digitalização podem ser ignorados. Deve-se notar entretanto que a taxa de 1kHz nem sempre é suficiente para desconsiderar os efeitos da digitalização em qualquer

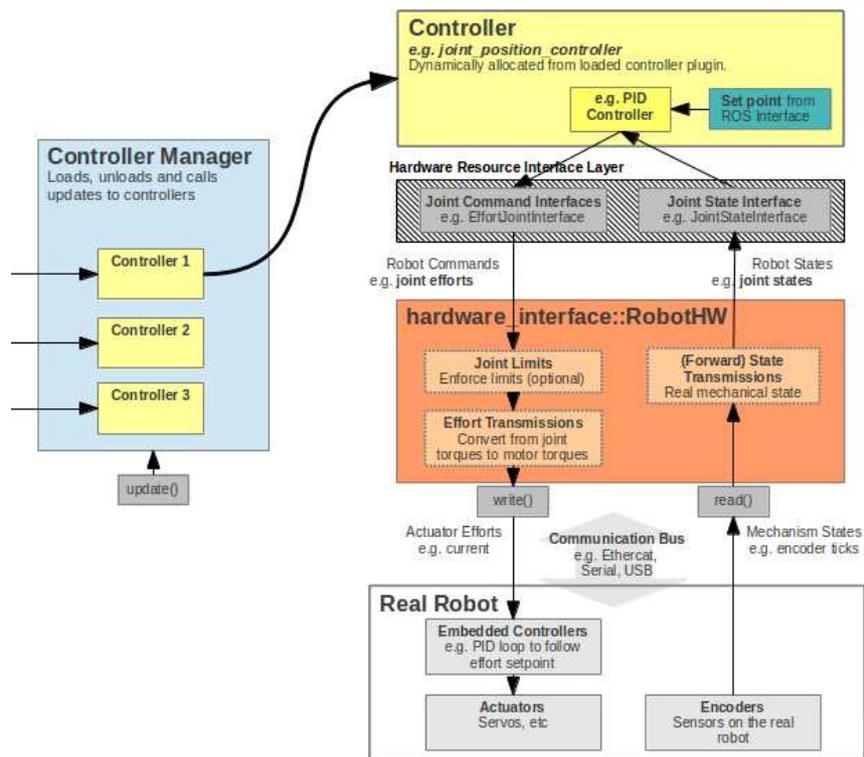


Figura 22: Arquitetura de controladores de baixo nível no ROS.

sistema robótico, principalmente nos que implementam controle de força e controle de impedância.

O modelo de um controlador de tempo contínuo é apropriado para controladores não lineares, assim como o controlador proposto neste trabalho. A função `update()` do controlador proposto neste trabalho implementa a lei de controle definida por (197), (199-201), (238),(239) e (257).

4.2.6 Laço de tempo real do ROS

A Figura 23 mostra um diagrama do laço de controle no ROS. Os blocos retangulares representam as classes envolvidas na implementação deste laço e as setas as invocações de funções ou acesso de variáveis através de ponteiros. No caso de funções, a ponta da seta indica a classe que implementa a função e a origem da seta a classe que invoca a função.

No caso de ponteiros, a ponta da seta indica a classe que armazena a variável. As elipses representam tópicos que são publicados ou assinados pelas classes mostradas no diagrama. Os blocos em azul são as classes que precisam ser implementadas especificamente para um determinado robô através da derivação das classes base representadas no diagrama. As demais classes são as mesmas para qualquer robô.

A classe `RobotHW`, implementa a interface do ROS com o *hardware* do robô. No caso de uma simulação, como neste trabalho, esta classe é derivada para uma classe `RobotSim`, que implementa a interface com o simulador Gazebo. Por sua vez, a classe `RobotSim`, é derivada para a classe `RobotSimTwil`, que implementa as particularidades da interface do ROS com a simulação do robô Twil, e para a classe `RobotSimWam`, que implementa as particularidades da interface do ROS com a simulação do robô Barrett WAM.

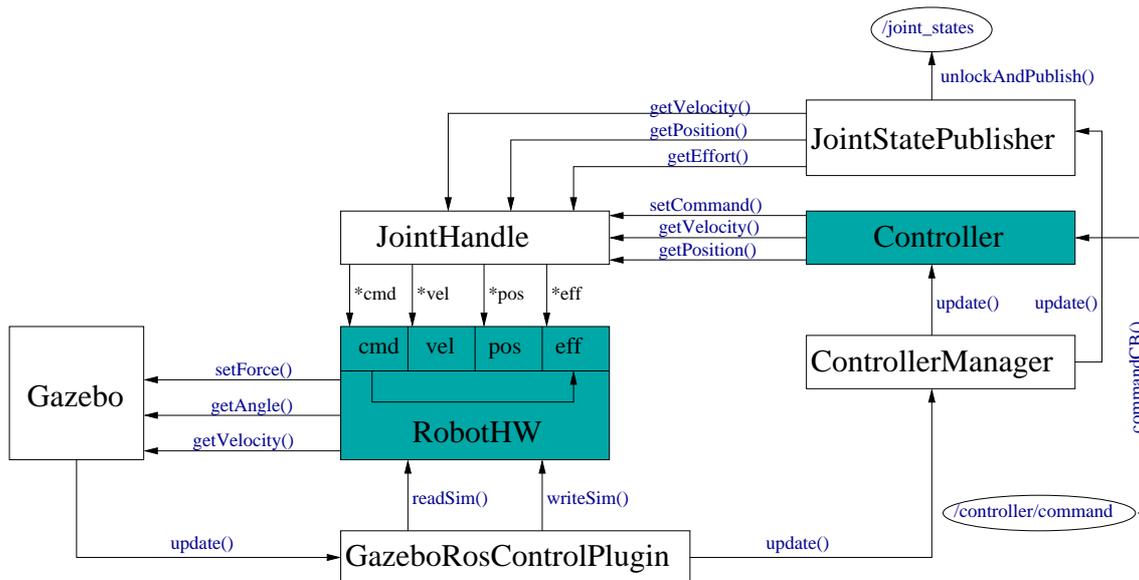


Figura 23: Laço de tempo real do ROS, adaptada de (MACIEL, 2014).

A classe `GazeboRosControlPlugin` implementa o sincronismo de tempo entre o simulador Gazebo e o ROS, em particular com o gerenciador de controladores do ROS. Quando se está trabalhando em simulação isto é necessário porque o tempo é simulado pelo simulador Gazebo e portanto o ROS deve operar com base no tempo simulado e não com base no tempo real. O gerenciador de controladores do ROS é implementado pela classe `ControllerManager`.

A sua função é executar um a um, em sequência, os controladores que encontram-se ativos a cada instante de controle. Cabe ressaltar que na nomenclatura do ROS controlador não é exatamente o mesmo que controlador na nomenclatura da teoria de sistemas de controle. Para o ROS, controlador é qualquer classe que é carregada pelo gerenciador de controladores (e para tanto é obrigado a implementar determinadas funções) e tem a sua função `update()` invocada a cada instante de controle.

Não necessariamente esta classe implementa uma lei de controle ou aciona algum controlador, pode simplesmente amostrar um sensor e publicar os dados, ou ter qualquer outra função no sistema. No diagrama da Figura 23, as classes `JointStatePublisher` e `Controller` são controladores. A classe `JointStatePublisher` apenas publica o estado do robô no tópico `/joint_states` e não tem a função de um controlador no sentido de sistemas de controle. Já a classe `Controller` implementa um controlador no sentido de sistemas de controle.

A referência é recebida através do tópico `/controller/command`. Na realidade, esta classe é derivada, dependendo do caso, para a classe `JointEffortController`, que implementa para cada junta um *forward* da referência para comando do atuador, mantendo o sistema em malha aberta, ou para a classe `CartLinearizingController`, que implementa a linearização (195).

Ou para a classe `NonSmoothBackstepController`, que implementa o controlador não suave descrito no Capítulo 3, ou ainda para a classe `ComputedTorque` que implementa o controlador por torque calculado descrito no Capítulo 3.

A classe `JointHandle` implementa uma abstração das juntas do robô, disponibilizando funções para leitura da posição, velocidade e esforço nas juntas do robô e para aplicação de esforços nas juntas. Estas funções acessam através de ponteiros as respec-

tivas variáveis nos objetos da classe `RobotHW`, que implementam efetivamente o acesso ao *hardware* ou à simulação do robô.

Na nomenclatura do ROS, a estrutura de controle mostrada na Figura 23 é denominado laço de tempo real e opera com um período constante. É importante notar que só existe um laço de tempo real no qual todos os controladores são executados e portanto, todos os controladores executam com a mesma frequência. Se algum controlador tiver que ser executado com uma frequência menor, ele próprio deve implementar uma subamostragem, o que força os períodos dos diversos controladores serem múltiplos do período do controlador de menor período.

A operação do laço de tempo real segue a seguinte sequência:

1. O simulador Gazebo é o elemento responsável pelo tempo simulado. E a cada vencimento do período de controle é invocada pelo Gazebo a função `update()` da classe `GazeboRosControlPlugin`.
2. A função `update()` da classe `GazeboROSControlPlugin` invoca a função `readSim()` da classe `RobotSim` (derivada da classe `RobotHW`).
3. A função `readSim()` da classe `RobotSim` faz a leitura dos sensores do robô, através das funções `getAngle()` e `getVelocity()` da biblioteca do Gazebo e armazena estas leituras em variáveis internas.
4. A função `update()` da classe `GazeboROSControlPlugin` invoca a função `update()` da classe `ControllerManager`.
5. A função `update()` da classe `ControllerManager` invoca, em sequência, as funções `update()` de todos os controladores ativos.
6. A função `update()` de cada controlador ativo, que são implementados por classes derivadas da classe `Controller`, obtém a leitura dos sensores tipicamente através das funções `getPosition()` e `getVelocity` da classe `JointHandle`, calcula a lei de controle e aciona os atuadores através da função `setCommand()` da classe `JointHandle`.
7. As funções `getPosition()` e `getVelocity` da classe `JointHandle` apenas retornam valores que foram armazenados pela função `readSim()` e a função `setCommand()` apenas armazena o valor do comando para ser usado pela função `writeSim()` da classe `GazeboRosControlPlugin`. No caso de ser usado o robô real, estas funções acessariam os sensores e atuadores reais. No caso da classe `RobotSimTwil`, derivada da classe `RobotSim` para o robô Twil, a função `setCommand()` também armazena o esforço na variável `eff`, para simular um sensor de esforço na junta já que esta simulação não é feita pelo Gazebo.
8. A função `update()` da classe `GazeboROSControlPlugin` é responsável por invocar a função `writeSim()` da classe `RobotSim` (derivada da classe `RobotHW`).
9. A função `writeSim()` da classe `RobotSim` obtém o esforço a ser aplicado na junta de sua variável interna e usa a função `setForce()` da biblioteca do Gazebo para aplicar o esforço na junta.

Cabe ressaltar que esta mesma estrutura, inclusive as mesmas classes, seriam utilizadas em caso de acionamento do robô real ao invés de simulação. Apenas os blocos Gazebo (simulador) e GazeboRosControlPlugin não existiriam e a classe RobotHW seria especializada pela classe (RobotHWTwil, por exemplo) responsável por acionar diretamente a *hardware* do robô e implementar o período de controle através da chamada da função `update()` pertencente à classe `ControllerManager`. O restante do sistema permaneceria exatamente o mesmo.

4.2.7 Unified Robot Description Format - URDF

O URDF é um formato XML para representação de modelos de robôs (ROBOT OPERATING SYSTEM, 2014c) utilizado para descrever propriedades cinemáticas e dinâmicas. Uma grande quantidade de robôs já foram modelados (ROBOT OPERATING SYSTEM, 2014d) utilizando este formato, como por exemplo os robôs Denso VS060, Pioneer P3DX e Robotino. Utilizando-se o formato URDF é possível simular um robô dentro de ambientes de simulação específicos como o Gazebo e o V-REP, assim como realizado em MICHIELETTO; ZANIN; MENEGATTI (2013), onde um modelo proposto para o robô NAO foi testado com o ROS.

Um arquivo URDF descreve a forma como os diferentes elos e juntas de um robô estão conectados entre si, além de apresentar propriedades físicas dos mesmos, como a geometria e momentos de inércia. Uma propriedade importante de um arquivo URDF é a possibilidade de descrever a geometria de um robô utilizando formas simples como cubos, esferas e cilindros, porém nem sempre estas formas conseguem descrever robôs reais com geometrias complexas (como o robô móvel Twil).

Nos casos onde é necessário descrever uma geometria mais complexa é possível utilizar arquivos gerados em ferramentas de CAD *Computer Aided Design*, como por exemplo o SolidWorks® que foi utilizado para desenhar os componentes do robô móvel Twil. A conversão dos arquivos do SolidWorks® para URDF foi efetuada através da utilização de um *add-in* (ROS.ORG, 2013). A Figura 24 mostra a ferramenta SolidWorks® com o modelo 3D do robô móvel Twil. Ainda, um trecho do código URDF gerado pode ser observado na Figura 25 onde os seguintes elementos podem ser analisados:

- `<robot name="twil" >`: definição do nome do robô;
- `<link name="chassis" >`: define um elo do robô (o elo chassis neste caso), propriedades físicas como a massa (`<mass value="6.4923"/>`) e a inércia (`<inertia= ... >`) podem ser associadas ao elo;
- `<geometry> <mesh filename="package://twil/meshes/chassis.STL"/> </geometry>`: este item referencia um arquivo do tipo STereoLithography (STL), o qual descreve somente a geometria da superfície de um objeto tridimensional, sem representar cor, textura ou outra propriedade comum aos arquivos de CAD (WIKIPEDIA, 2014). Através deste arquivo o simulador Gazebo pode construir o objeto gráfico que vai representar o elo;
- `<joint name="right_wheel_suport_joint" type="fixed" >`: define uma junta fixa (sem movimento) do robô, neste caso a junta do suporte de uma roda fixa.

Durante a simulação o Gazebo interpreta as informações referentes aos elementos da geometria do robô (elos e juntas por exemplo) utilizando estas informações para exibir graficamente a estrutura do robô.

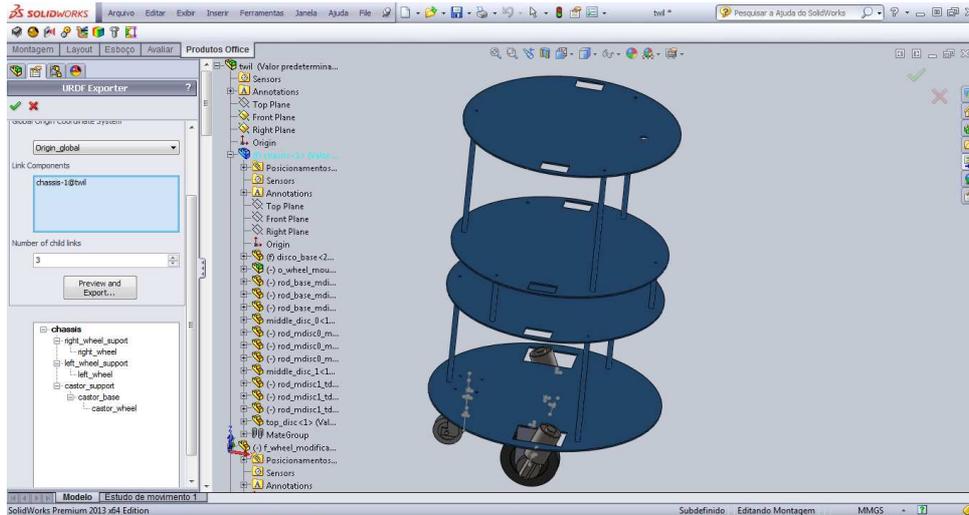


Figura 24: Edição do modelo 3D do robô móvel Twil no SolidWorks®.

4.3 Pacote do robô móvel Twil

O software desenvolvido no ROS é organizado na forma de pacotes. Sendo que um pacote pode conter códigos fonte e executáveis de nodos do ROS, bibliotecas do ROS, arquivos de configuração dentre outros arquivos que possam ser agrupados de forma lógica. Um pacote do ROS é simplesmente um diretório no sistema operacional corrente que contém um arquivo `package.xml`. Um pacote é a menor parte da estrutura de uma versão de software no ROS.

Para simplificar a forma como o código é compartilhado, os pacotes são organizados em *stacks* do ROS, que são coleções de pacotes que de forma conjunta provêm determinada funcionalidade. De forma similar aos pacotes, os *stacks* são diretórios do sistema operacional corrente que possui um arquivo do tipo `stack.xml`, sendo que os pacotes contidos neste diretório são considerados como parte do *stack*. É importante observar que os nomes dos pacotes devem ser únicos em todo o sistema, o que significa que dois ou mais pacotes não podem possuir o mesmo nome, ainda que os pacotes estejam em *stacks* diferentes.

A Figura 26 mostra o *stack* do robô móvel twil, o qual foi criado para agrupar o software do robô móvel Twil. Os arquivos `CMakeList.txt` e `Makefile` são utilizados pelo sistema de *build* do ROS que é baseado no `cmake`, o arquivo `stack.xml` define o *stack* e suas dependências, com os diretórios sendo pacotes que estão incluídos no *stack*.

No diretório `twil_control_gazebo` está a implementação da interface entre o ROS e a simulação do robô móvel Twil no Gazebo, no diretório `twil_controllers` está a implementação dos controladores propostos na seção 3. Ainda, no diretório `twil_description` está a descrição URDF do robô móvel Twil e a implementação do nodo do ROS que efetua a identificação de parâmetros está no diretório `twil_ident`.

4.3.1 Pacote `twil_control_gazebo`

O pacote `twil_control_gazebo` apresentado na Figura 27 implementa a interface entre o ROS e a simulação do robô móvel Twil no Gazebo. Novamente, `CMakeList.txt` e `Makefile` são arquivos utilizados pelo sistema de *build*, `manifest.xml` é o arquivo que define o pacote, `robot_sim_plugins.xml` é o arquivo que descreve a

```

1 <robot name="twil">
2   <link name="chassis">
3     <inertial>
4       <origin xyz="6.3955E-06 -2.1963E-17 0.27338" rpy="0 0 0" />
5       <mass value="6.4923" />
6       <inertia ixx="0.67525" ixy="0.0014553" ixz="-0.00017525"
7         iyy="0.69058" iyz="-6.3289E-18" izz="0.28611" />
8     </inertial>
9     <visual>
10      <origin xyz="0 0 0" rpy="0 0 0" />
11      <geometry>
12        <mesh filename="package://twil/meshes/chassis.STL" />
13      </geometry>
14      <material name="">
15        <color rgba="0.75294 0.75294 0.75294 1" />
16      </material>
17    </visual>
18    <collision>
19      <origin xyz="0 0 0" rpy="0 0 0" />
20      <geometry>
21        <mesh filename="package://twil/meshes/chassis.STL" />
22      </geometry>
23    </collision>
24  </link>
25  .
26  .
27  .
28  <joint name="right_wheel_suport_joint" type="fixed">
29    <origin xyz="0 -0.161 -0.002" rpy="0 0 0" />
30    <parent link="chassis" />
31    <child link="right_wheel_suport" />
32    <axis xyz="0 0 0" />
33  </joint>
34  .
35  .
36  .
37 </robot>

```

Figura 25: Descrição URDF do robô móvel Twil.

```

twil/
├── CMakeList.txt
├── Makefile
├── stack.xml
├── twil_control_gazebo/
├── twil_controllers/
├── twil_description/
└── twil_ident/

```

Figura 26: Estrutura do pacote twil.

interface entre o ROS e o Gazebo, `robot_sim_twil.cpp` é o código fonte que define `RobotSimTwil` que é a implementação da classe `RobotHW` apresentado na Figura 23. O diretório `lib` contém as bibliotecas compartilhadas com os arquivos binários para a classe `RobotSimTwil`.

```

twil_control_gazebo/
├── CMakeLists.txt
├── lib/
│   └── libtwil_control_gazebo.so
├── Makefile
├── manifest.xml
├── robot_sim_plugins.xml
├── src/
│   └── robot_sim_twil.cpp

```

Figura 27: Pacote `twil_control_gazebo`.

4.3.2 Pacote `twil_controllers`

O pacote `twil_controllers` apresentado na Figura 28 implementa o controle proposto na seção 3. Os arquivos contidos no diretório `config` especificam parâmetros como as juntas do robô, ganhos e taxa de atualização para os controladores. Os arquivos do diretório `src` são a implementação dos controladores descritos nas seções 3.2.1 e 3.2.4. As classes `CartLinearizingController` e `NonSmoothBackstepController` implementam os respectivos controladores. Estas classes são ambas derivadas da classe `Controller` como mostra a Figura 23, enquanto o diretório `include` contém os arquivos com a declaração destas classes.

O diretório `lib` contém as bibliotecas compartilhadas com os arquivos binários para estas classes. Ainda, o arquivo `twil_controllers_plugins.xml` especifica que as classes que implementam os controladores são plugins para o gerenciador de controladores do ROS. Os arquivos contidos no diretório `launch` são utilizados para carregar os controladores com seus respectivos arquivos de configuração.

4.3.3 Pacote `twil_description`

O pacote `twil_description` apresentado na Figura 29 contém a descrição no formato URDF para o robô móvel Twil assim como detalhado na seção 4.2.7. Os arquivos contidos no diretório `xacro` descrevem os parâmetros geométricos e de massa dos diferentes corpos que compõem o robô móvel Twil e o diretório `meshes` contém os arquivos no formato STL (*STereoLithography*) que descrevem o formato dos corpos.

Os arquivos do diretório `launch` são utilizados para carregar o modelo do robô móvel Twil no servidor de parâmetros do ROS. O arquivo `twil.launch` carrega o modelo do robô móvel Twil no servidor de parâmetros sendo utilizado com o robô atual que estiver sendo simulado, enquanto que o arquivo `twil_sim.launch` carrega o modelo do robô móvel Twil no servidor de parâmetros e roda o simulador Gazebo.

4.3.4 Pacote `twil_ident`

O pacote `twil_ident` apresentado na Figura 30 é um nodo do ROS que implementa a identificação de parâmetros descrita na seção 3.2.5. Novamente o código fonte está no

```

twil_controllers/
├── CMakeLists.txt
├── config/
│   ├── effort_control.yaml
│   ├── linearizing_control.yaml
│   └── nonsmooth_backstep_control.yaml
├── include/
│   ├── twil_controllers/
│   │   ├── cart_linearizing_controller.h
│   │   └── nonsmooth_backstep_controller.h
├── launch/
│   ├── cart_linearizing.launch
│   ├── joint_effort.launch
│   └── nonsmooth_backstep.launch
├── lib/
│   └── libtwil_controllers.so
├── Makefile
├── manifest.xml
├── src/
│   ├── cart_linearizing_controller.cpp
│   └── nonsmooth_backstep_controller.cpp
└── twil_controllers_plugins.xml

```

Figura 28: Pacote `twil_controllers`.

diretório `src`, e há um arquivo para carregar o nodo no diretório `launch`. Entretanto, como este pacote implementa um nodo do ROS e não um *plugin* há também um diretório `bin` com o código executável do nodo.

4.4 Pacote do manipulador Barrett WAM

Para permitir a simulação do manipulador Barrett WAM no simulador Gazebo com o controlador sendo executado pelo ROS o pacote `ufrgs_wam` foi criado. A estrutura do pacote está apresentada na Figura 31. O diretório `wam_control_gazebo` possui arquivos de interface de comunicação entre o ROS e o Gazebo, o diretório `wam_controllers` possui os arquivos para implementação do controlador de torque calculado e o diretório `wam_description` possui os arquivos de descrição da estrutura do manipulador.

Assim como para o robô móvel `twil`, o pacote do manipulador Barrett WAM contém arquivos `CMakeList.txt` e `Makefile` utilizados pelo sistema de *build*, e um arquivo `stack.xml` que define o `stack` e suas dependências. No diretório `wam_control_gazebo` está a implementação da interface entre o ROS e a simulação do manipulador Barret WAM, no diretório `wam_controllers` está a implementação dos controladores propostos na seção 3, mais especificamente do controlador de torque calculado, e no diretório `wam_description` está a descrição URDF do manipulador.

```

twil_description
├── launch/
│   ├── twil.launch
│   └── twil_sim.launch
├── Makefile
├── manifest.xml
├── meshes/
│   ├── battery_bosch_12v.stl
│   ├── castor_base.stl
│   ├── castor_support.stl
│   ├── castor_wheel.stl
│   ├── chassis.stl
│   ├── cpu.stl
│   ├── fixed_wheel.stl
│   ├── fixed_wheel_support.stl
│   ├── left_wheel_support.stl
│   └── right_wheel_support.stl
├── xacro/
│   ├── battery_bosch_12v.urdf.xacro
│   ├── castor_base.urdf.xacro
│   ├── castor_support.urdf.xacro
│   ├── castor_wheel.urdf.xacro
│   ├── chassis.urdf.xacro
│   ├── cpu.urdf.xacro
│   ├── eurocard.urdf.xacro
│   ├── fixed_wheel_support.urdf.xacro
│   ├── fixed_wheel.urdf.xacro
│   ├── twil.urdf.xacro
│   └── twil_wam.urdf.xacro

```

Figura 29: Pacote twil_description.

```

twil_ident/
├── bin/
│   └── ident
├── CMakeLists.txt
├── launch/
│   └── ident.launch
├── Makefile
├── manifest.xml
├── src/
│   └── ident.cpp

```

Figura 30: Pacote twil_ident.

```

ufrgs_wam/
├── CMakeList.txt
├── Makefile
├── stack.xml
├── wam_control_gazebo/
├── wam_controllers/
└── wam_description/

```

Figura 31: Estrutura de diretórios do pacote do manipulador Barrett WAM.

4.4.1 Pacote `wam_control_gazebo`

O pacote `wam_control_gazebo` apresentado na Figura 32 implementa a interface entre o ROS e a simulação do manipulador Barret WAM no Gazebo. Novamente, `CMakeList.txt` e `Makefile` são arquivos utilizados pelo sistema de *build*, `manifest.xml` é o arquivo que define o pacote, `robot_sim_plugins.xml` é o arquivo que descreve a interface entre o ROS e o Gazebo, `robot_sim_wam.cpp` é o código fonte que define `RobotSimWam` que é a implementação da classe `RobotHW` apresentado na Figura 23. O diretório `lib` contém as bibliotecas compartilhadas com os arquivos binários para a classe `RobotSimWam`.

```

wam_control_gazebo/
├── CMakeLists.txt
├── lib/
│   └── libwam_control_gazebo.so
├── Makefile
├── manifest.xml
├── robot_sim_plugins.xml
├── src/
│   └── robot_sim_wam.cpp

```

Figura 32: Pacote `wam_control_gazebo`.

4.4.2 Pacote `wam_controllers`

O pacote `wam_controllers` apresentado na Figura 33 implementa o controle proposto na seção 3. Os arquivos contidos no diretório `config` especificam parâmetros como as juntas do robô, ganhos e taxa de atualização para os controladores. Os arquivos do diretório `src` são a implementação do controlador descrito na seção 3.4. A classe `ComputedTorqueController` implementa o respectivo controlador. Esta classe é derivada da classe `Controller` como mostra a Figura 23, enquanto o diretório `include` contém os arquivos com a declaração destas classes.

O diretório `lib` contém as bibliotecas compartilhadas com os arquivos binários para estas classes. Ainda, o arquivo `wam_controllers_plugins.xml` especifica que as classes que implementam os controladores são plugins para o gerenciador de controladores do ROS. Os arquivos contidos no diretório `launch` são utilizados para carregar os controladores com seus respectivos arquivos de configuração.

```

wam_controllers/
├── CMakeLists.txt
├── config/
│   └── computed_torque_control.yaml
├── include/
│   ├── wam_controllers/
│   │   └── computed_torque_controller.h
├── launch/
│   └── computed_torque.launch
├── lib/
│   └── libwam_controllers.so
├── Makefile
├── manifest.xml
├── src/
│   └── computed_torque_controller.cpp
└── wam_controllers_plugins.xml

```

Figura 33: Pacote `wam_controllers`.

4.4.3 Pacote `wam_description`

O pacote `wam_description` apresentado na Figura 34 contém a descrição no formato URDF para o manipulador Barrett WAM assim como detalhado na seção 4.2.7. Os arquivos contidos no diretório `xacro` descrevem os parâmetros geométricos e de massa dos diferentes corpos que compõem o manipulador e o diretório `meshes` contém os arquivos no formato STL (*STereoLithography*) que descrevem o formato dos corpos.

Os arquivos do diretório `launch` são utilizados para carregar o modelo do manipulador no servidor de parâmetros do ROS. O arquivo `wam.launch` carrega o modelo do manipulador no servidor de parâmetros sendo utilizado com o robô atual que estiver sendo simulado, enquanto que o arquivo `wam_sim.launch` carrega o modelo do manipulador no servidor de parâmetros e roda o simulador Gazebo.

```
wam_description
├── launch/
│   ├── wam.launch
│   └── wam_sim.launch
├── Makefile
├── manifest.xml
├── meshes/
│   ├── wam1.stl
│   ├── wam2.stl
│   ├── wam3.stl
│   ├── wam4.stl
│   ├── wam5.stl
│   ├── wam6.stl
│   ├── wam7.stl
│   └── wambase.stl
├── xacro/
│   ├── wam_base.urdf.xacro
│   ├── wam_bhand.urdf.xacro
│   ├── wam_j1.urdf.xacro
│   ├── wam_j2.urdf.xacro
│   ├── wam_j3.urdf.xacro
│   ├── wam_j4.urdf.xacro
│   ├── wam_j5.urdf.xacro
│   ├── wam_j6.urdf.xacro
│   ├── wam_j7.urdf.xacro
│   ├── wam_tool_plate.urdf.xacro
│   ├── wam.urdf_original.xacro
│   └── wam.urdf.xacro
```

Figura 34: Pacote wam_description.

5 EXPERIMENTOS

5.1 Introdução

Neste capítulo serão apresentados os resultados das simulações efetuadas no Gazebo com os controladores sendo executados no ROS. São apresentados os resultados para o robô móvel Twil com controle em malha aberta, controle por linearização, identificação de parâmetros e controlador *backstepping*. Também para o manipulador Barrett WAM é apresentado o resultado da simulação com o controlador de torque calculado.

5.2 Simulação do robô móvel Twil com controle em malha aberta

Nesta simulação, o robô móvel Twil é acionado via controladores do ROS, que implementam apenas um *bypass* (*effort_controllers*) da referência para a saída. Isto equivale a acionar o robô móvel Twil em malha aberta.

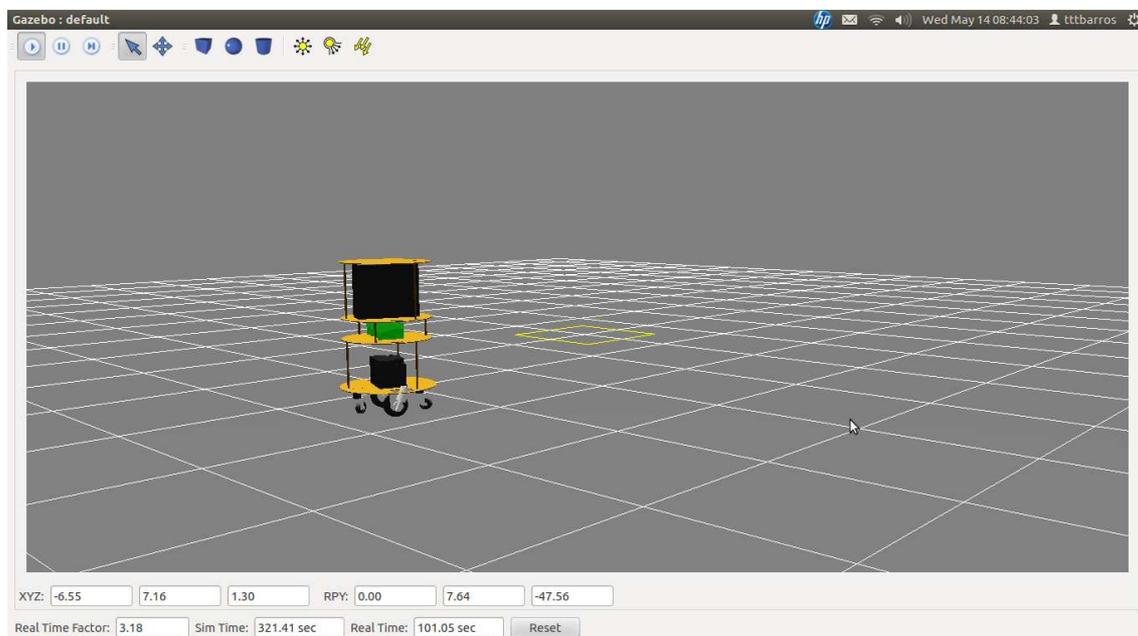


Figura 35: Robô móvel Twil no Gazebo.

Nesta simulação são apresentados os resultados com o robô Twil operando em malha aberta, ou seja, utilizando o controlador padrão `JointEffortController` do ROS. Este controlador leva o sinal de referência para a saída de controle, assim, utilizar este controlador tem o mesmo efeito do robô operando em malha aberta. A entrada do sistema

utilizada foi um degrau de torque aplicado nas rodas. A Figura 35 mostra o robô móvel Twil no Gazebo e a Figura 36 mostra o diagrama de nodos e tópicos. Os nodos são representados por elipses e os tópicos por retângulos.

O controlador não é representado como um nodo pois o mesmo configura-se como um *plugin* que é carregado pelo gerenciador de controladores e assim é incluído no nodo `/gazebo` que representa o simulador Gazebo. Os tópicos que iniciam com `/gazebo/` são intrínsecos ao simulador e utilizados para na configuração e monitoramento de seu comportamento, não sendo utilizados neste trabalho.

Os tópicos `/twil/right_wheel_joint_effort_controller_command` e `/twil/left_wheel_joint_effort_controller_command` são utilizados pelos controladores das rodas direita e esquerda respectivamente para receber suas referências.

O controlador `JointEffortController` é utilizado, com os tópicos sendo os comandos de torque para as rodas. O tópico `/joint_states` é onde os estados das juntas são publicados pelo `JointStateController`, sendo que usualmente o estado é somente a posição e velocidade das juntas.

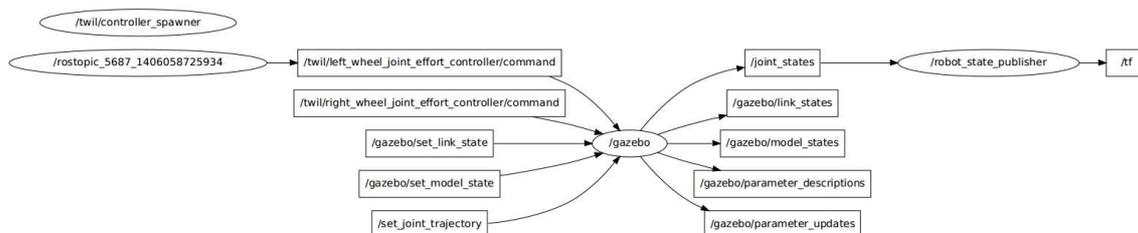


Figura 36: Diagrama de nodos da simulação em malha aberta do robô móvel Twil.

A Figura 37 mostra o degrau de torque aplicado e as Figuras 38, 39, 40 mostram respectivamente a posição X x Y , as posições individuais nos eixos X e Y e as velocidades das rodas do robô móvel Twil.

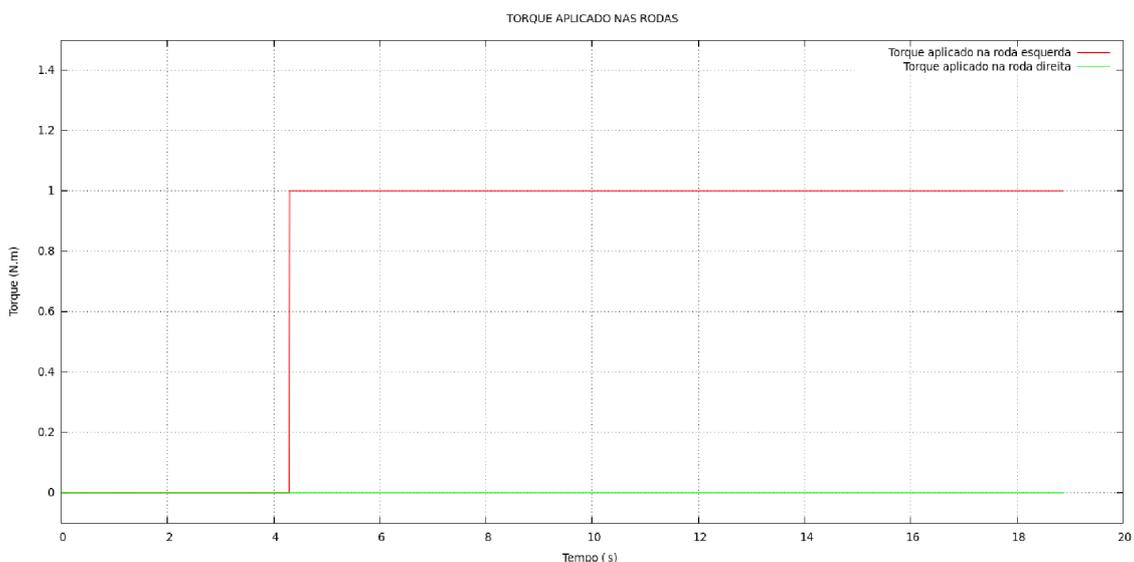


Figura 37: Degrau de torque na roda esquerda.

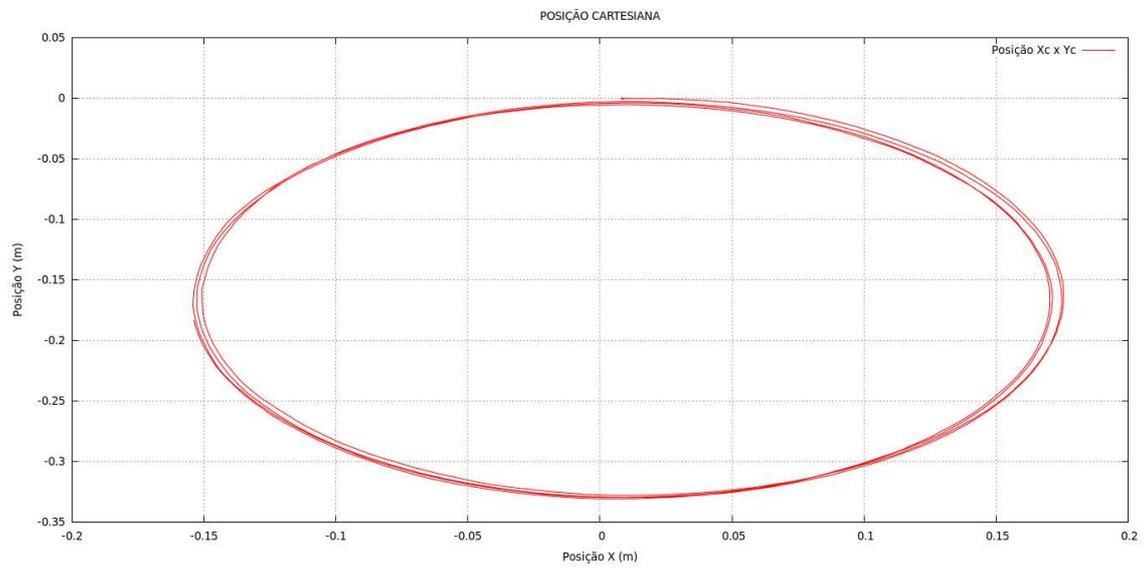


Figura 38: Posição $x(t) \times y(t)$.

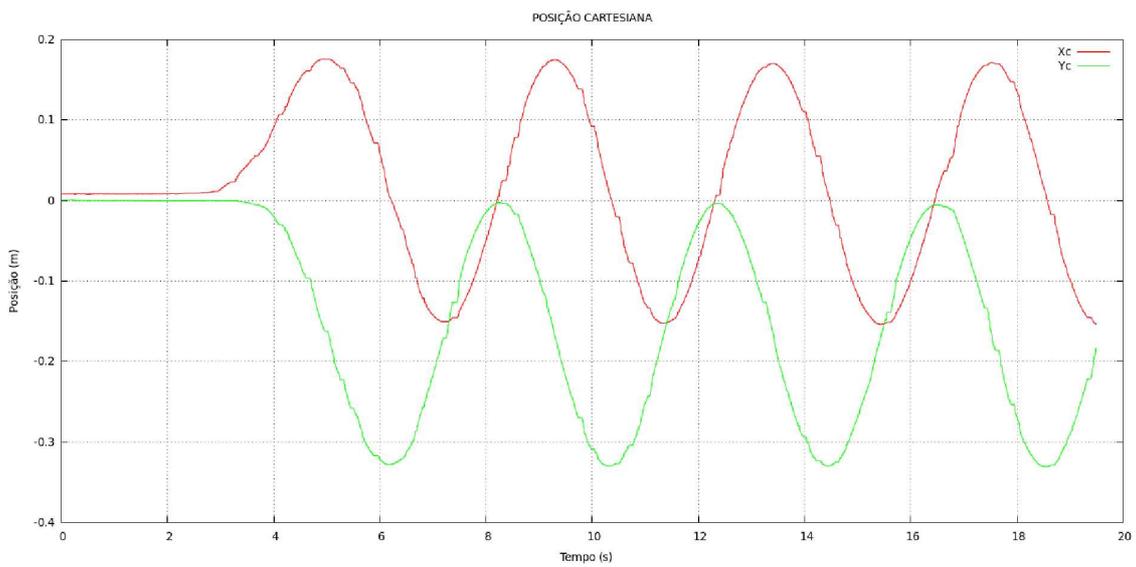


Figura 39: $x(t)$ e $y(t)$.

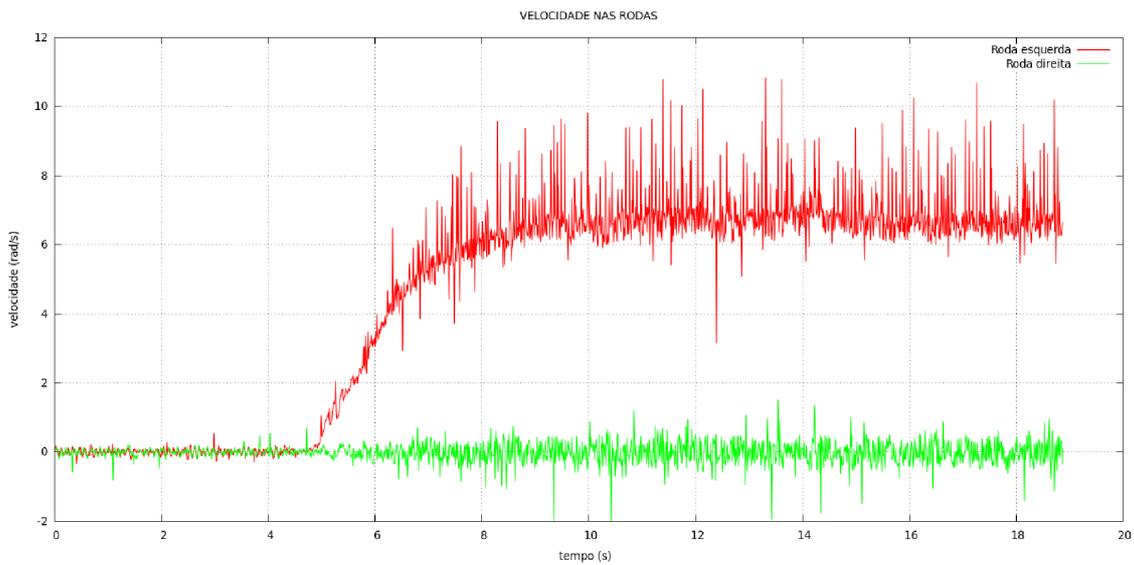


Figura 40: Velocidades das rodas.

A Figura 41 mostra o degrau de torque aplicado e as Figuras 42, 43, 44 mostram respectivamente a posição X x Y , as posições individuais nos eixos X e Y e as velocidades das rodas do robô móvel Twil.

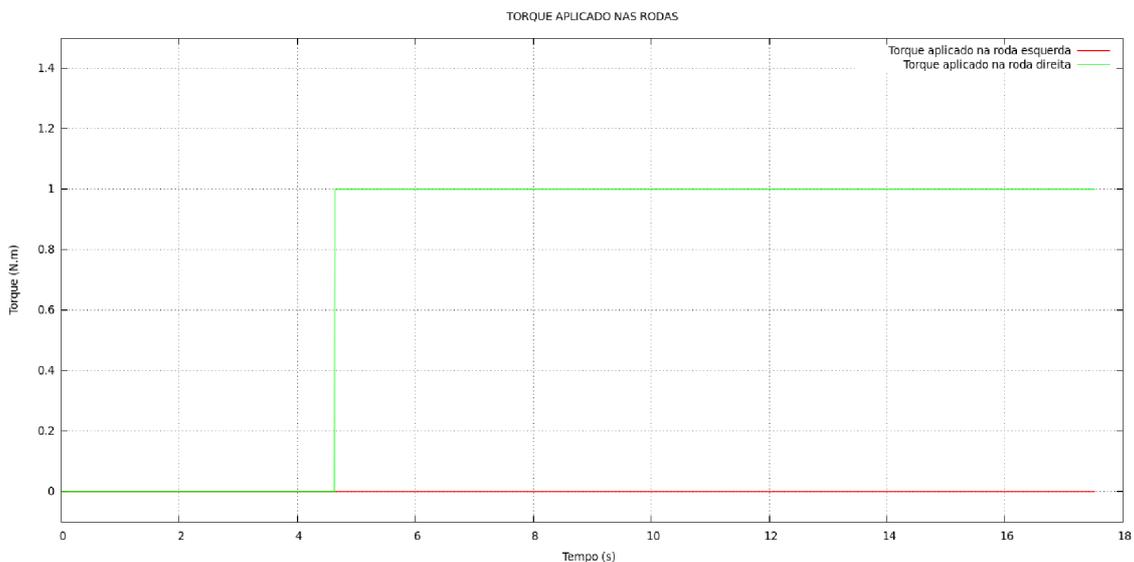


Figura 41: Degrau de torque na roda direita.

Pode-se verificar pela análise dos gráficos que o comportamento do robô móvel Twil com os sinais de torque aplicados em malha aberta apresentaram resultados condizentes com o esperado, assim como pode-se concluir por exemplo das Figuras 38 e 42 cujo gráfico descreve trajetórias elípticas, uma vez que há torque aplicado somente em uma roda do robô para cada caso.

A Figura 45 mostra o degrau de torque aplicado e as Figuras 46, 47, 48 mostram respectivamente a posição X x Y , as posições individuais nos eixos X e Y e as velocidades das rodas do robô móvel Twil.

Na Figura 46 pode-se observar que o robô inicialmente descreve uma trajetória elíptica

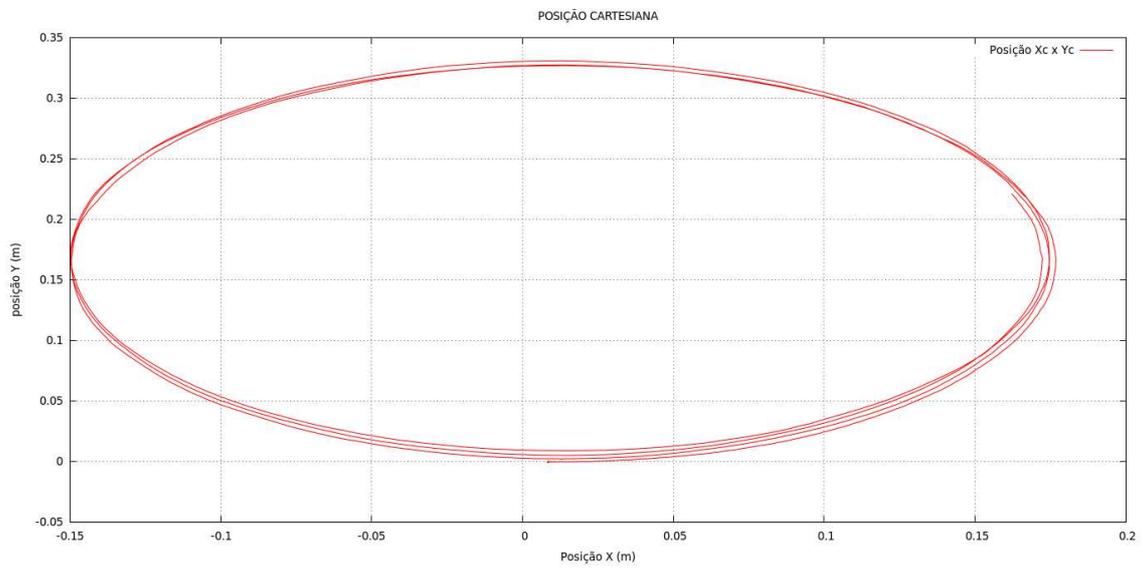


Figura 42: Posição $x(t) \times y(t)$.

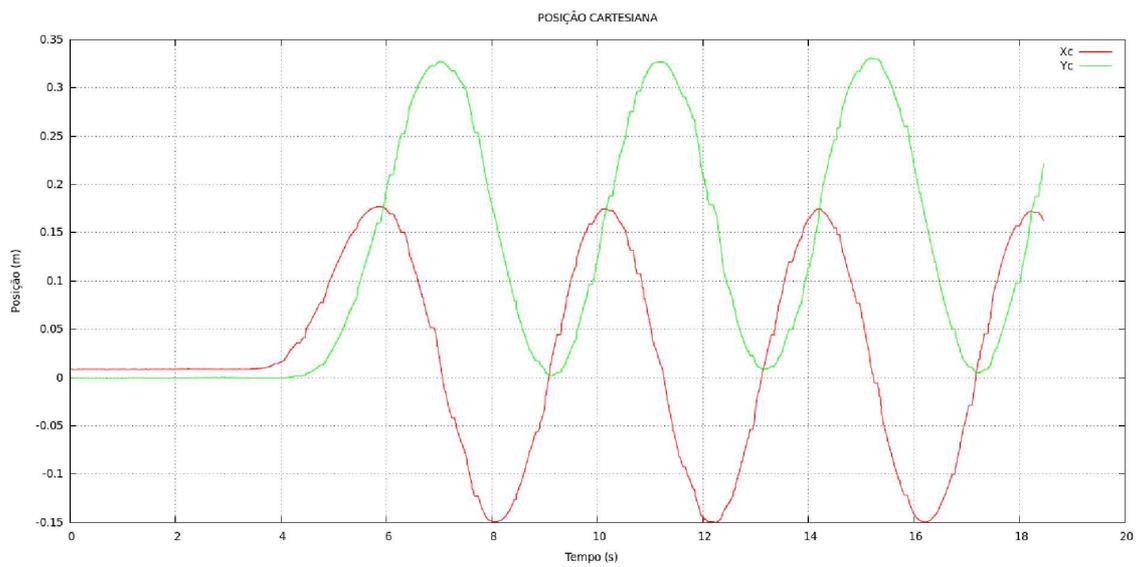


Figura 43: $x(t)$ e $y(t)$.

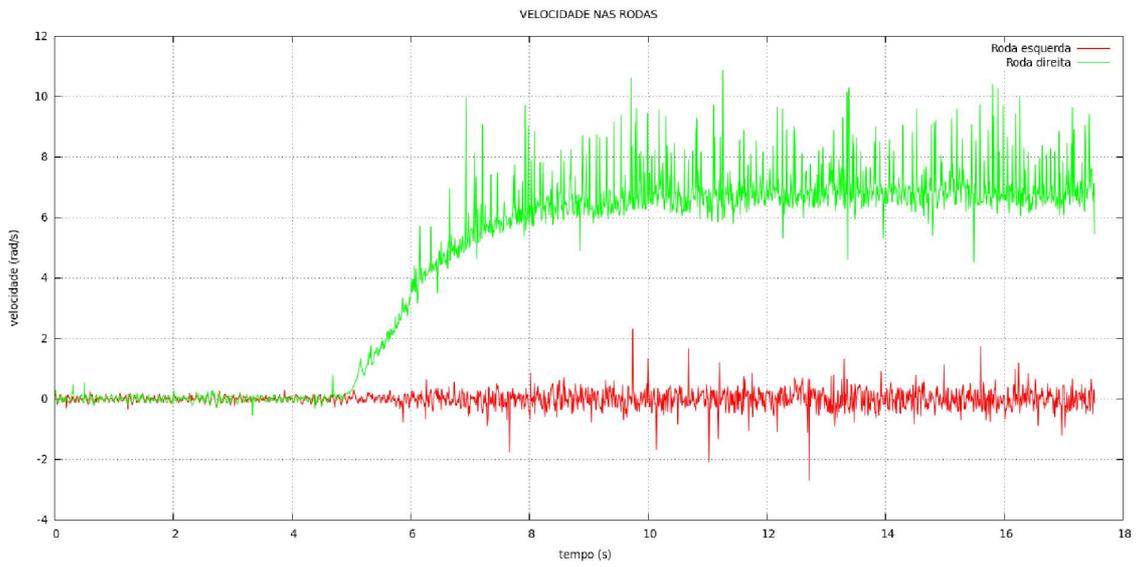


Figura 44: Velocidades das rodas.

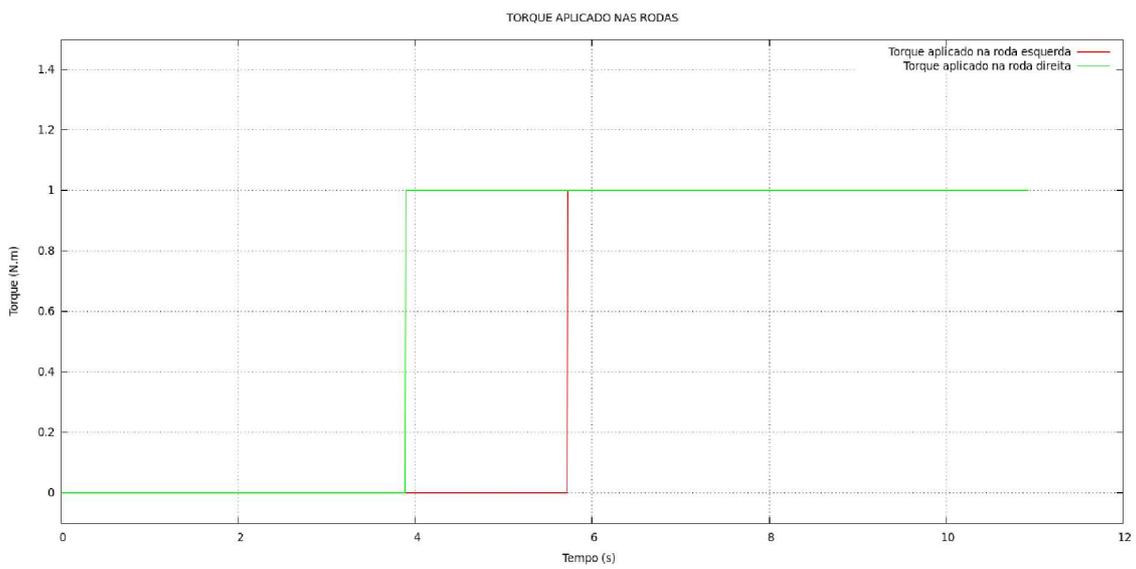


Figura 45: Degrau de torque nas rodas esquerda e direita.

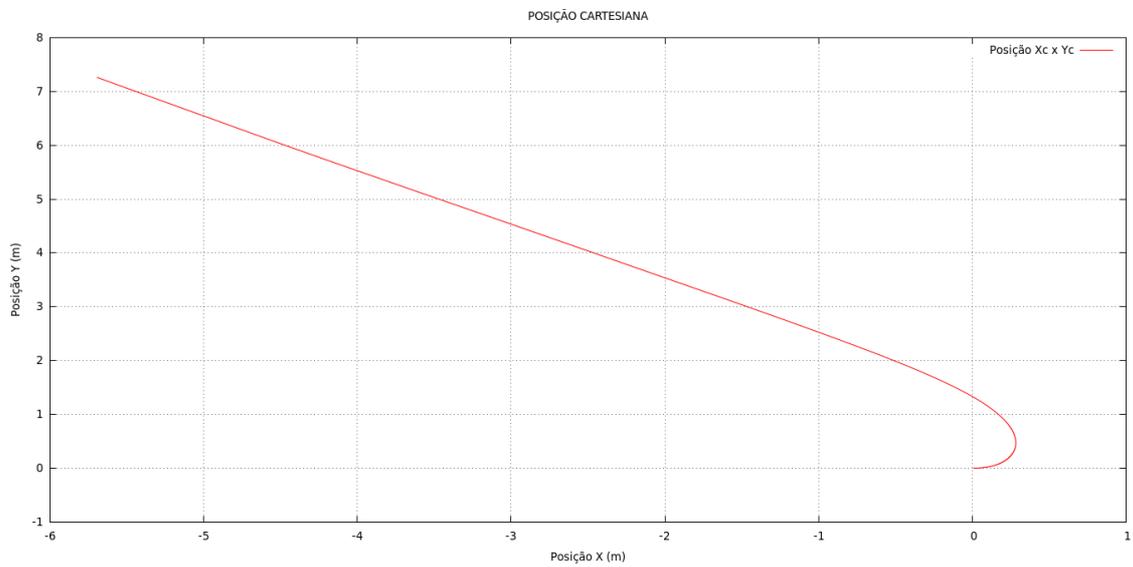


Figura 46: Posição $x(t) \times y(t)$.

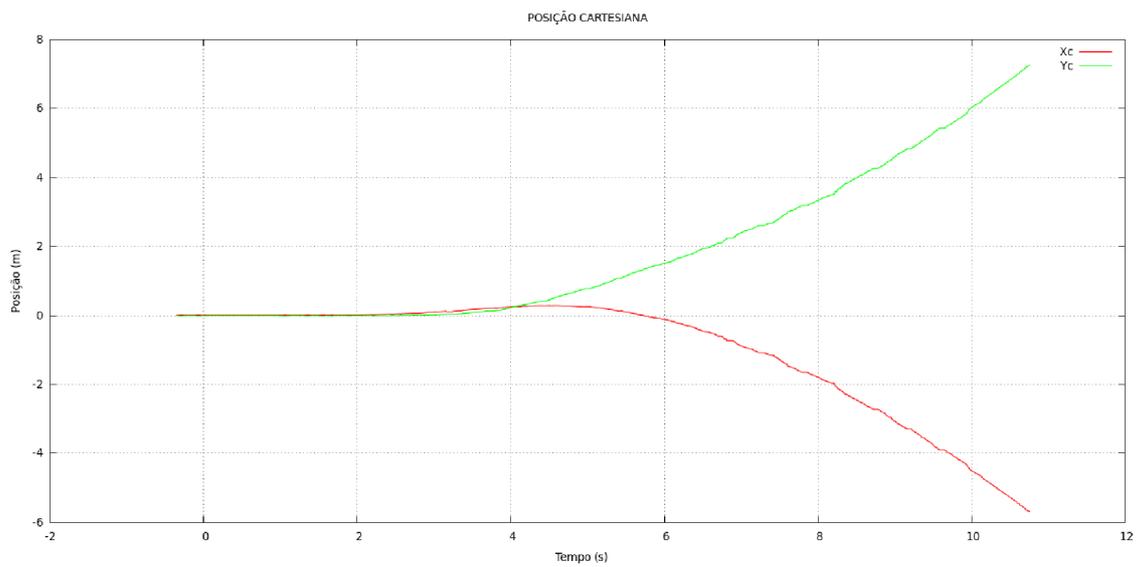


Figura 47: $x(t)$ e $y(t)$.

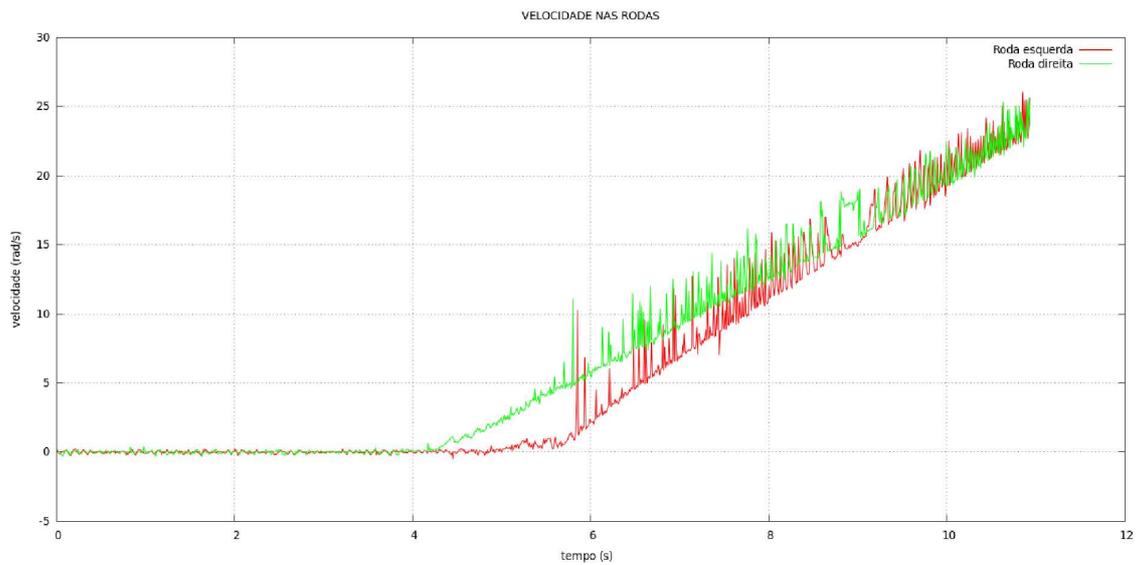


Figura 48: Velocidades das rodas.

e posteriormente segue uma reta já que conforme pode-se observar na Figura 45 há um atraso entre a aplicação de torque na roda direita e roda esquerda, fazendo assim com o robô primeiro inicie a trajetória elíptica e só então quando os torques são iguais comece a descrever uma trajetória em linha reta.

5.3 Identificação de parâmetros do robô móvel Twil

Esta seção apresenta os resultados da simulação do identificador de parâmetros proposto para o robô móvel Twil conforme descrito na seção 3.2.5 do Capítulo 3. Os resultados apresentados são referentes aos parâmetros identificados K_5 , K_6 , K_7 e K_8 , assim como os valores das diagonais das matrizes de covariância correspondentes. A Figura 49 mostra o diagrama de nodos da identificação de parâmetros.

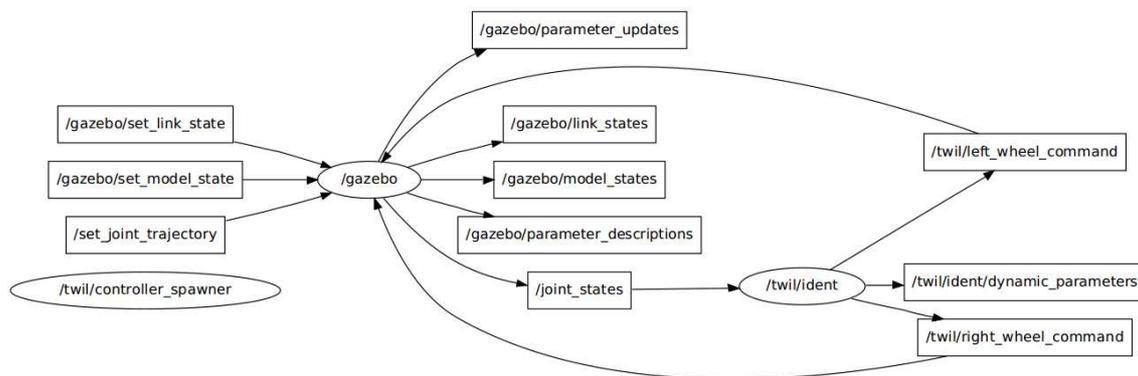


Figura 49: Diagrama de nodos da identificação de parâmetros do robô móvel Twil.

As Figuras 50, 51, 52 e 53 mostram respectivamente as curvas dos valores identificados para os parâmetros K_5 , K_6 , K_7 e K_8 . Deve-se observar que o horizonte do tempo de simulação das Figuras 50 e 51 é diferente dos demais, pelo fato de que os valores de pico destas duas figuras apresentam-se nos momentos iniciais de simulação e assim foi utilizada uma ampliação para que estes valores possam ser visualizados.

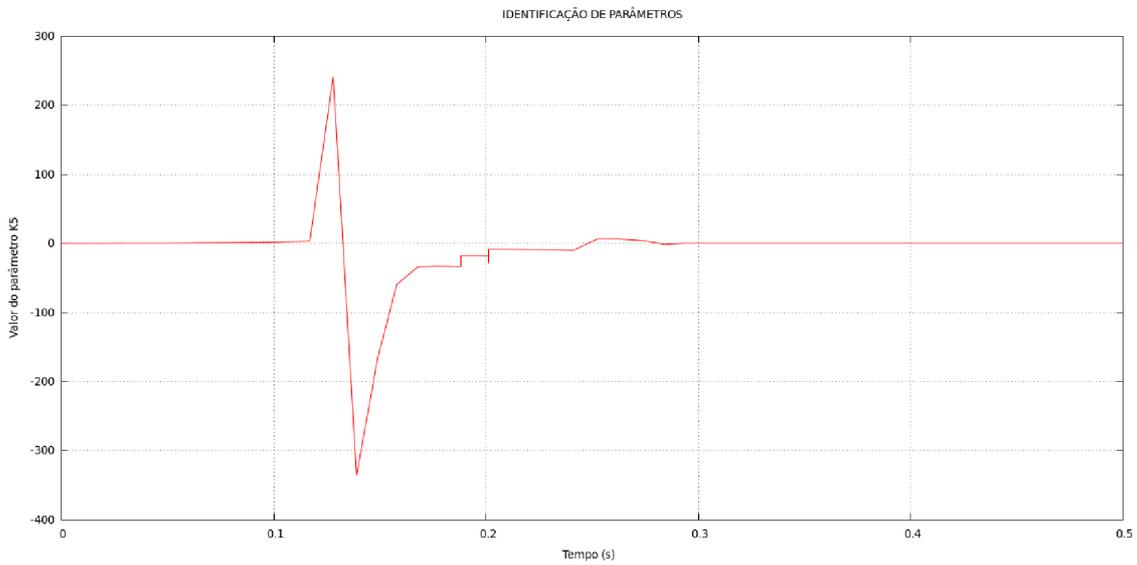


Figura 50: Evolução do valor estimado para K_5 .

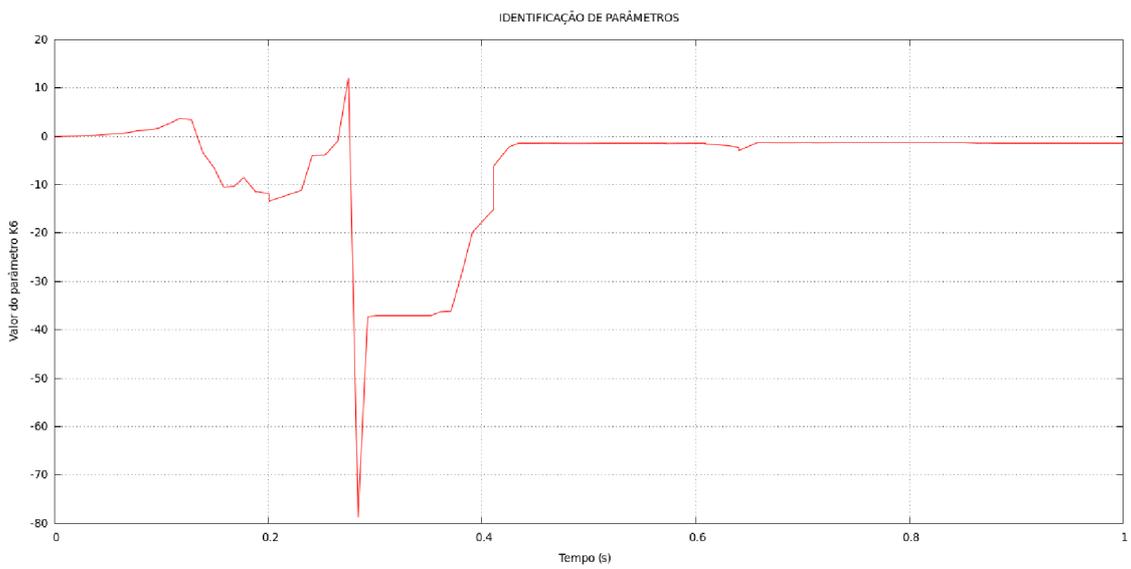
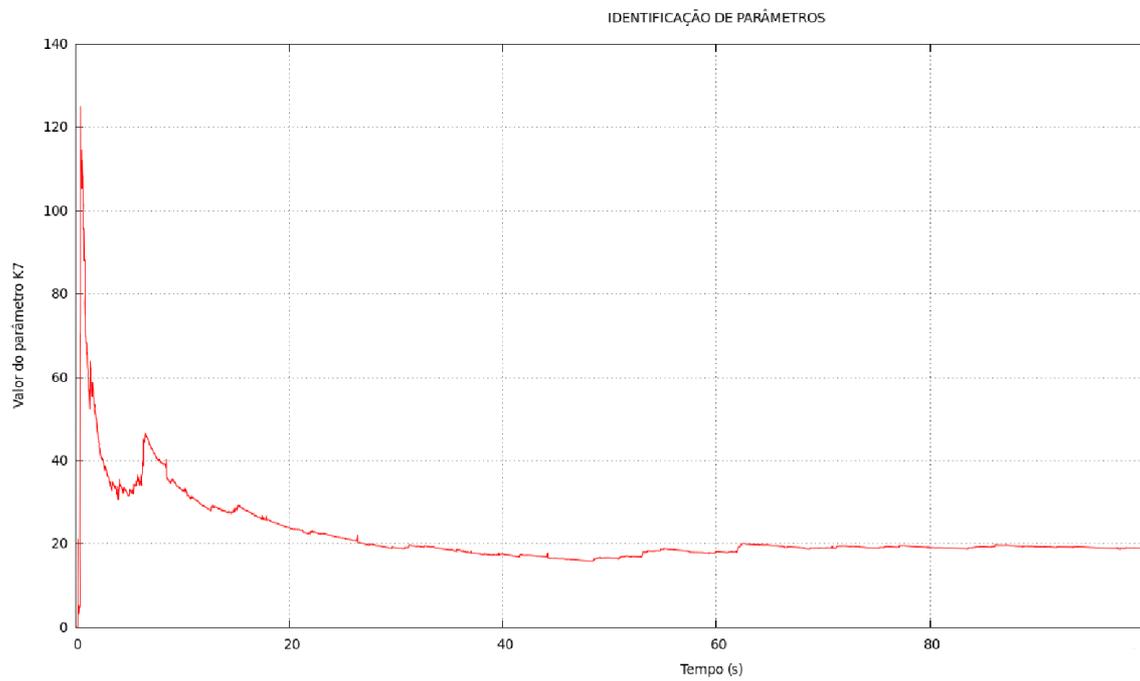
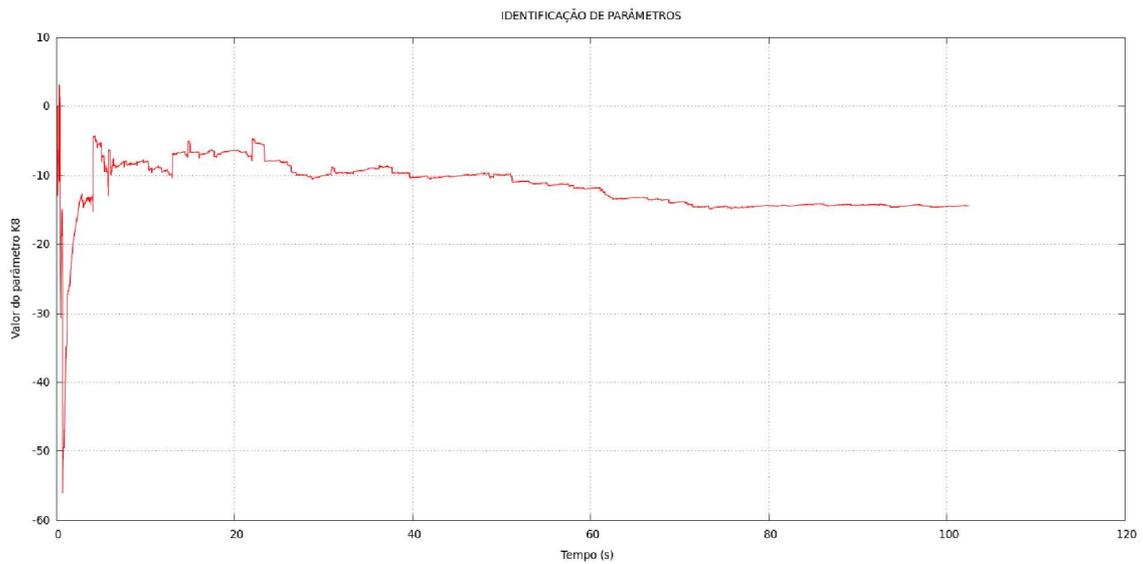


Figura 51: Evolução do valor estimado para K_6 .

Figura 52: Evolução do valor estimado para K_7 .Figura 53: Evolução do valor estimado para K_8 .

Nas Figuras 54, 55, 56 e 57 são apresentadas as curvas dos valores das matrizes de covariância relativas aos parâmetros. Os valores utilizados na implementação do controlador linearizante são apresentados na Tabela 4.

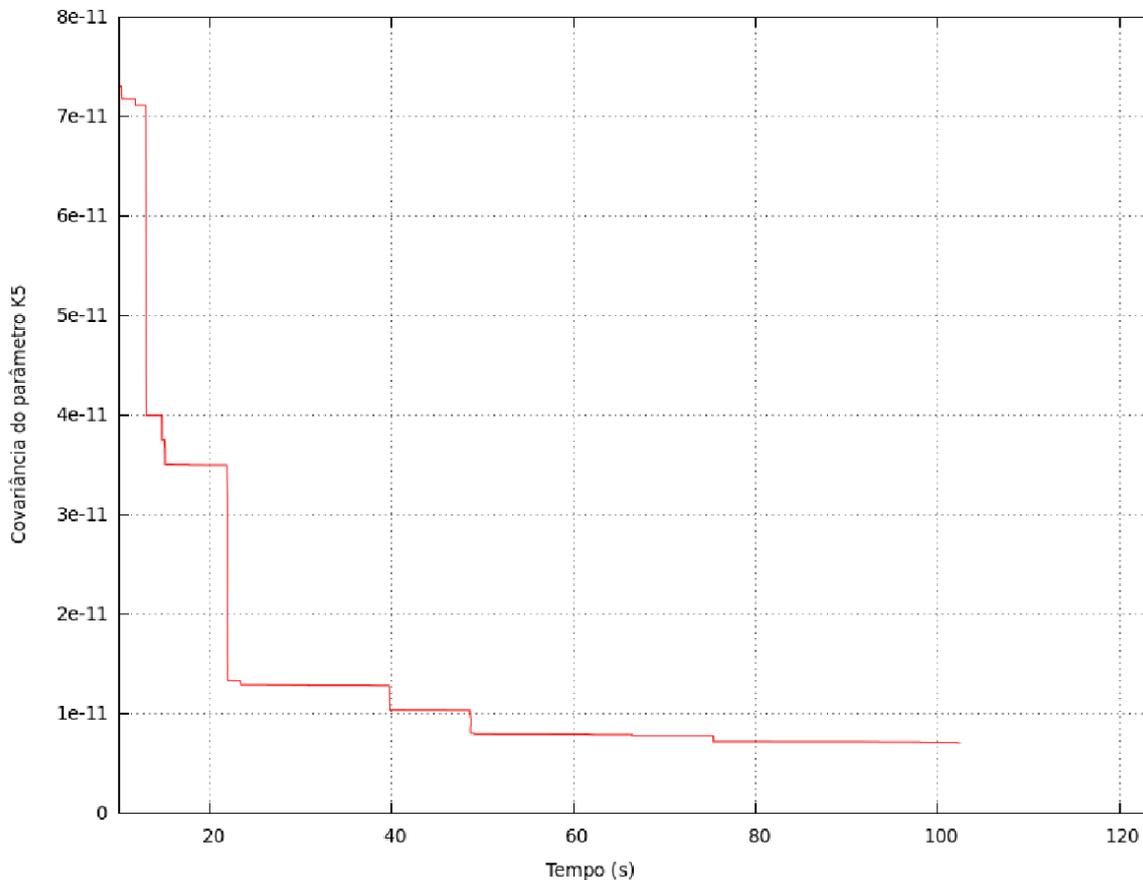


Figura 54: Diagonal da matriz de covariância referente ao parâmetro K_5 .

Tabela 4: Parâmetros identificados e valores das diagonais das matrizes de covariância.

Parâmetro	Valor	Covariância
K_5	0.0043125446	$7.04281951252e^{-12}$
K_6	0.1851039578	$1.08705914620e^{-09}$
K_7	18.7807217774	$1.96179198744e^{-06}$
K_8	-14.3839401898	$1.94971312561e^{-06}$

5.4 Simulação do robô móvel Twil utilizando linearização

Nesta simulação, o robô móvel Twil é acionado utilizando a linearização (195). As entradas da simulação são as acelerações lineares e angulares. A Figura 58 mostra o diagrama de nodos da simulação.

A primeira simulação realizada consistiu na aplicação de um sinal com aceleração linear constante ($0.075m/s^2$) e aceleração angular nula. A Figura 59 mostra as posições individuais nos eixos X e Y e a Figura 60 mostra a posição $X \times Y$, a análise desta figura

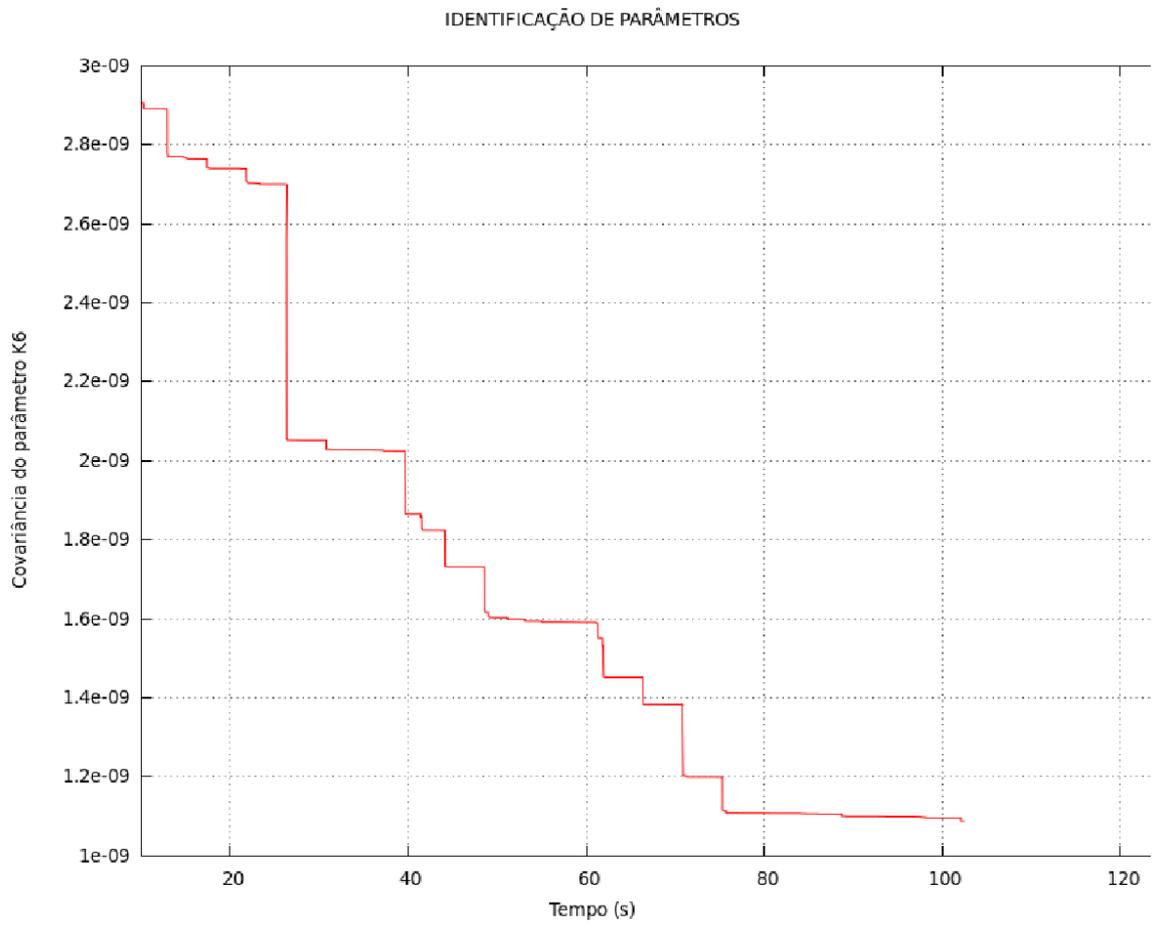


Figura 55: Diagonal da matriz de covariância referente ao parâmetro K_6 .

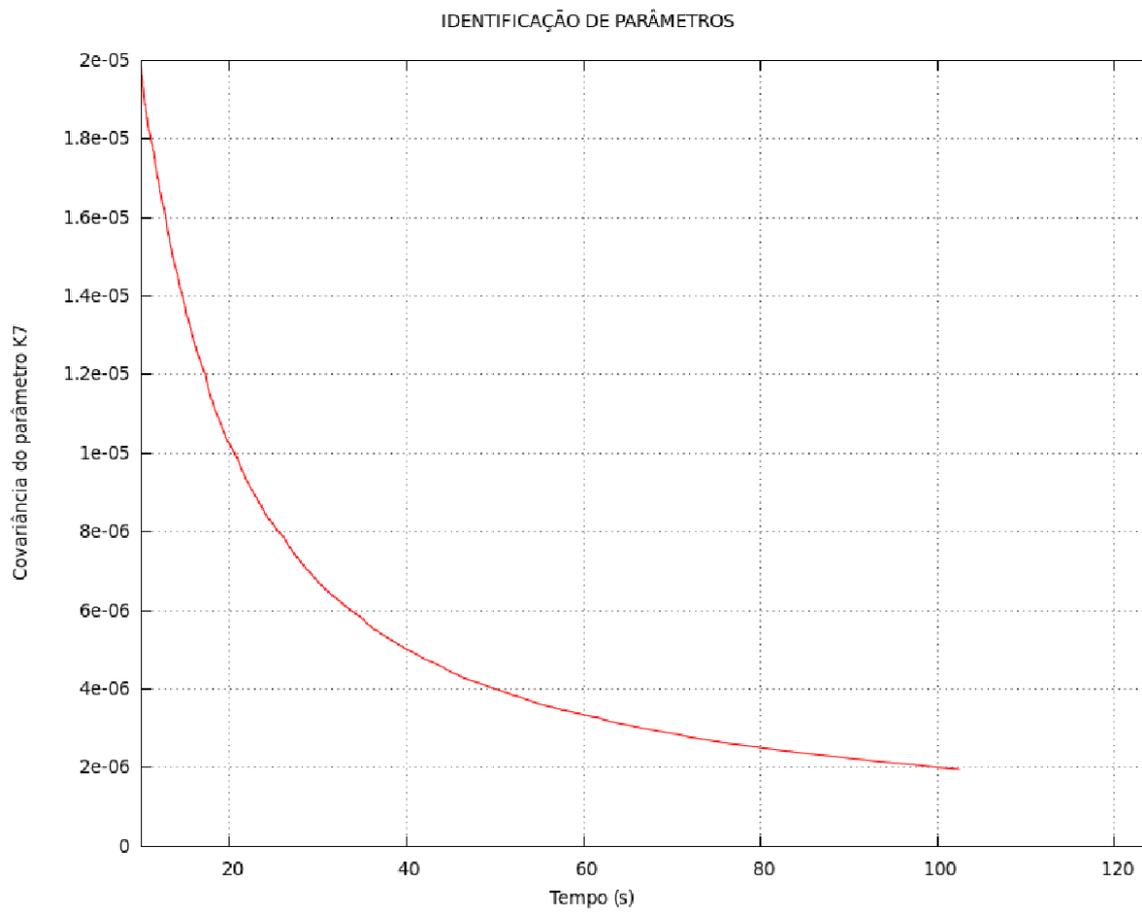


Figura 56: Diagonal da matriz de covariância referente ao parâmetro K_7 .

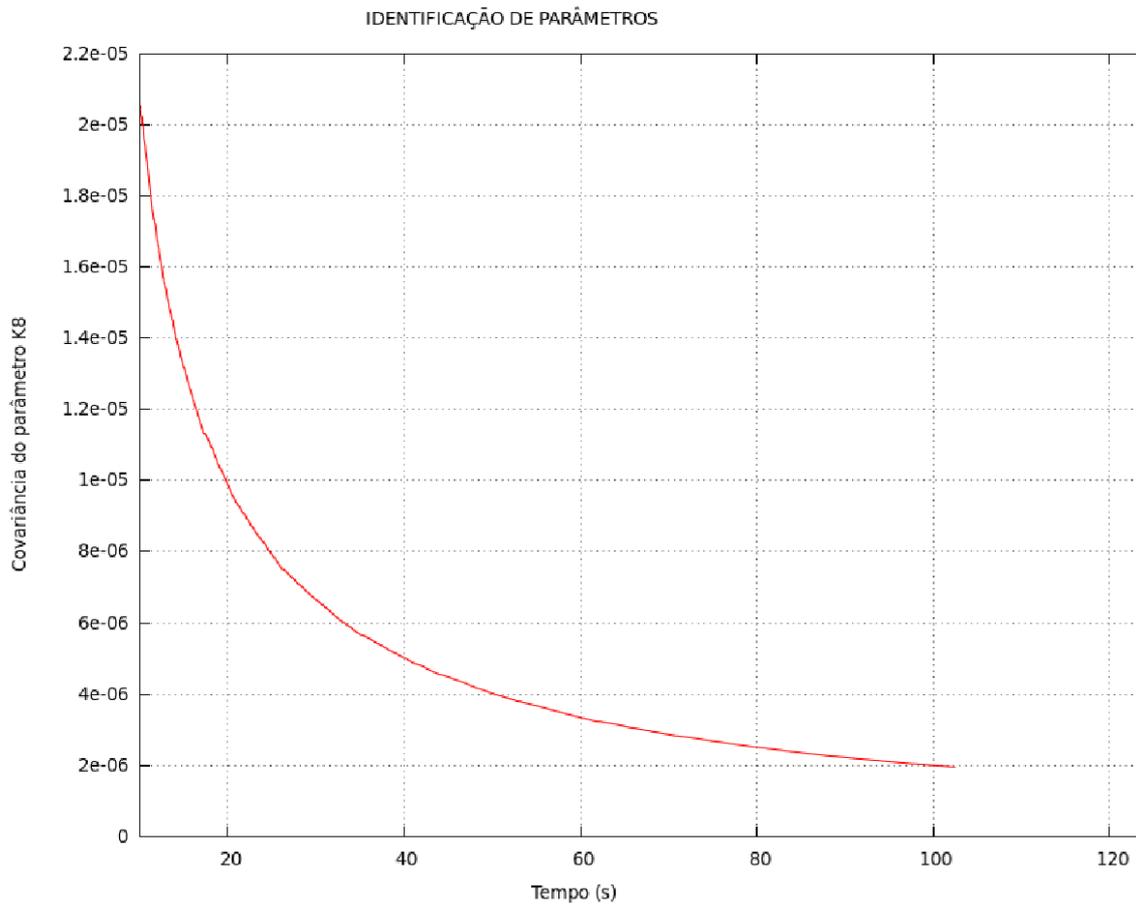


Figura 57: Diagonal da matriz de covariância referente ao parâmetro K_8 .

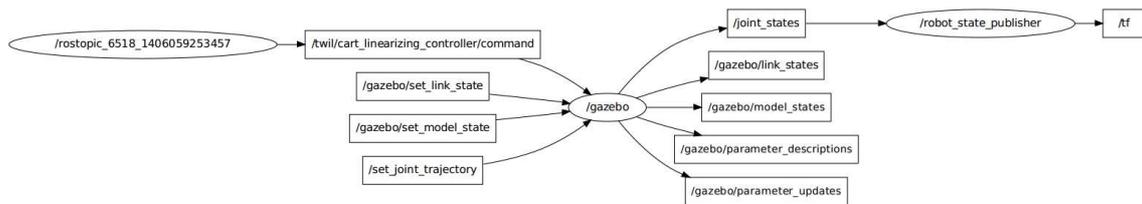


Figura 58: Diagrama de nodos da simulação da linearização do robô móvel Twil.

permite concluir que o robô se desloca em um trajeto que se aproxima de uma linha reta. O trajeto não é o de uma linha reta pois a linearização não é perfeita uma vez que é baseada no modelo do robô móvel Twil, o qual também não é exato e possui alguns parâmetros que não são conhecidos exatamente. Deve-se observar que mesmo na simulação o modelo utilizado para a linearização (121) não é exato uma vez que o simulador Gazebo utiliza seu próprio modelo de robô móvel calculando os parâmetros diretamente da descrição URDF do mesmo.

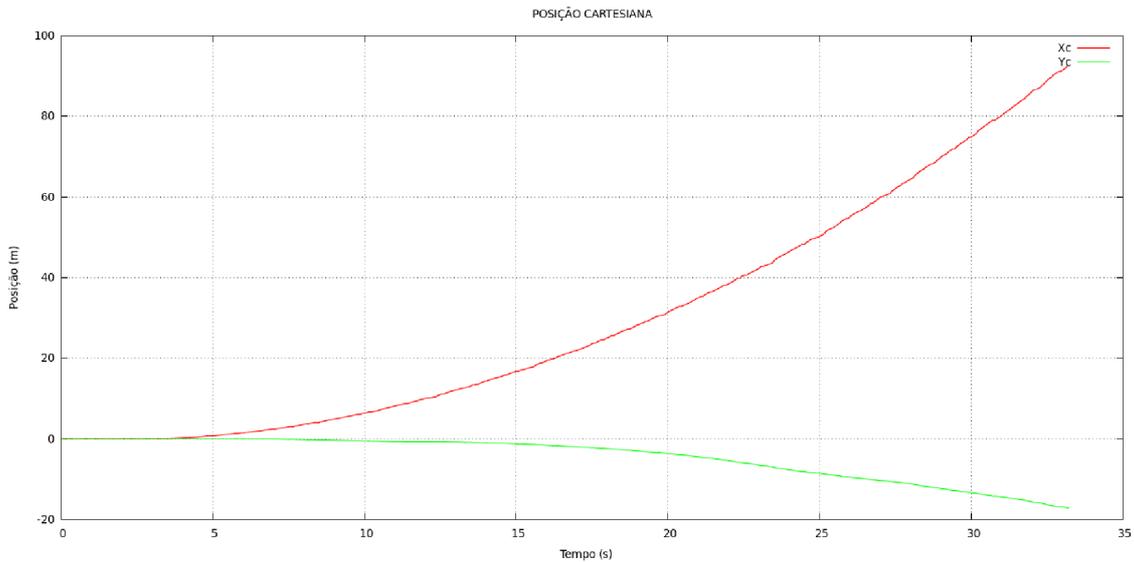


Figura 59: Posições nos eixos X e Y , na simulação da linearização do robô móvel Twil com aceleração linear constante e aceleração angular nula.

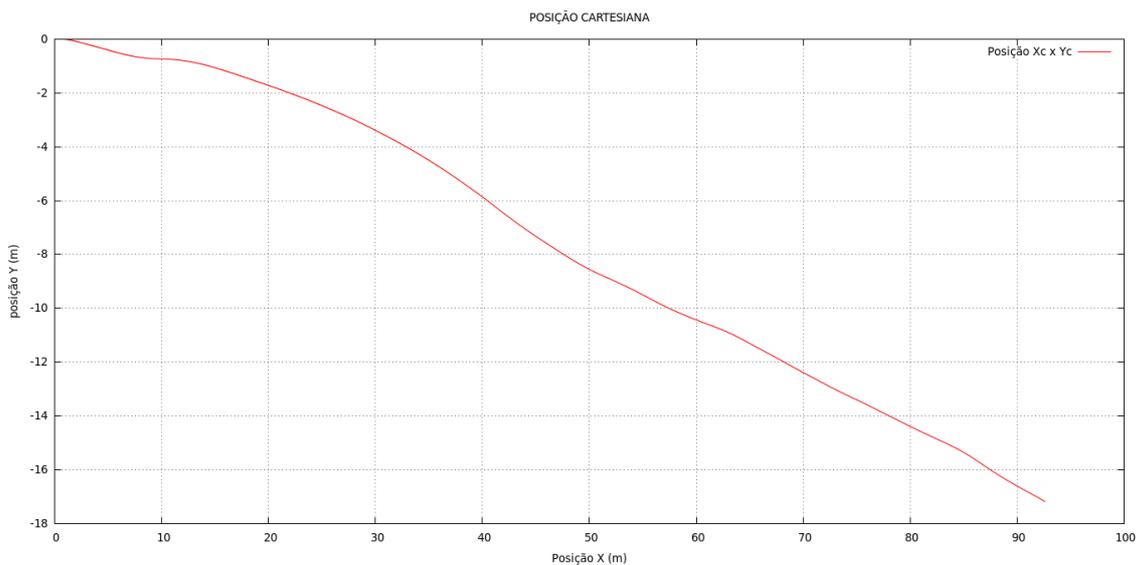


Figura 60: Posição $X \times Y$, na simulação da linearização do robô móvel Twil com aceleração linear constante e aceleração angular nula.

A segunda simulação realizada consistiu na aplicação de um sinal com aceleração angular constante ($0.075m/s^2$) e aceleração linear nula. A Figura 61 mostra as posições

individuais nos eixos X e Y e a Figura 62 mostra a posição $X \times Y$, a análise desta figura permite concluir que o robô descreve uma trajetória circular condizente com o fato de possuir apenas aceleração angular aplicada.

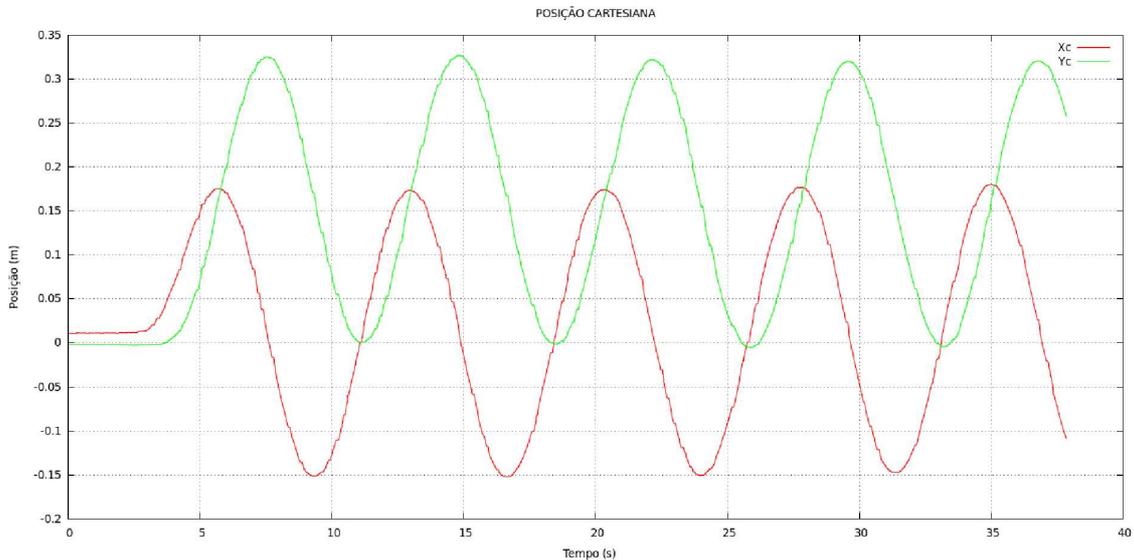


Figura 61: Posições nos eixos X e Y , na simulação da linearização do robô móvel Twil com aceleração angular constante e aceleração linear nula.

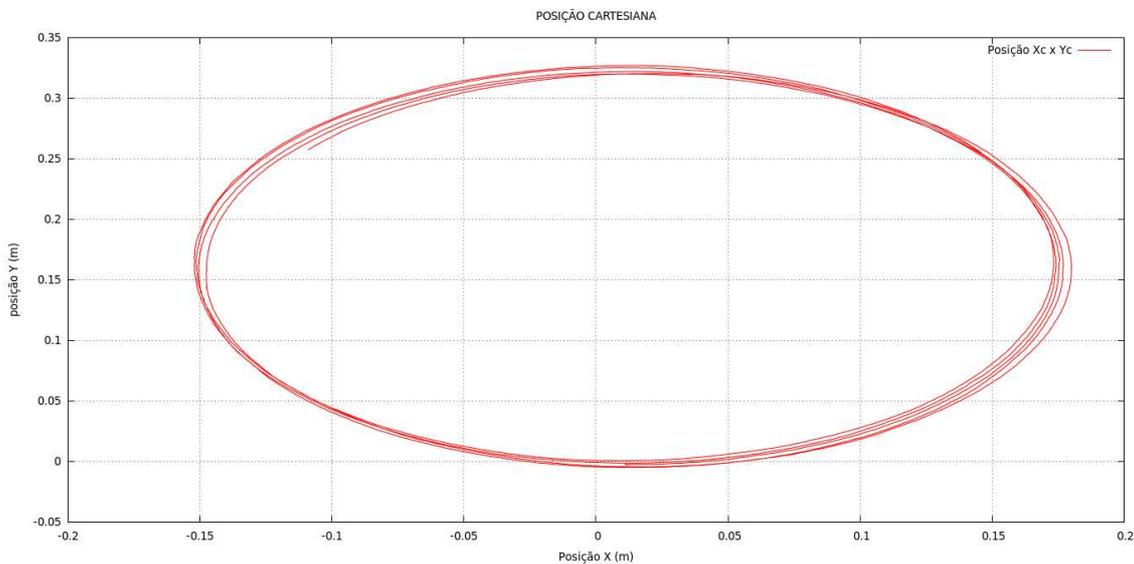


Figura 62: Posição $X \times Y$, na simulação da linearização do robô móvel Twil com aceleração angular constante e aceleração linear nula.

A terceira simulação realizada consistiu na aplicação de um sinal com aceleração linear e aceleração angular constantes ($0.075m/s^2$). A Figura 63 mostra as posições individuais nos eixos X e Y e a Figura 64 mostra a posição $X \times Y$, a análise desta figura permite concluir que o robô descreve uma trajetória em espiral, uma vez que a aplicação tanto de sinais de aceleração linear como angular, faz com que o robô afaste-se cada vez mais de seu ponto de origem pois nesta situação a componente de aceleração linear faz com que cada círculo descrito pelo robô tenha um acréscimo no raio descrito.

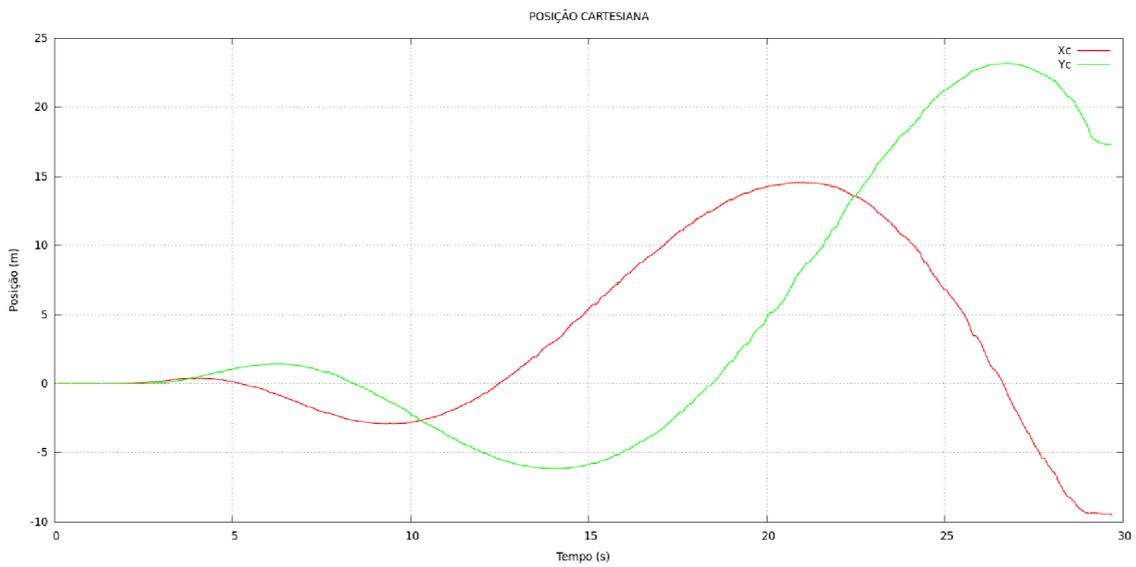


Figura 63: Posições nos eixos X e Y , na simulação da linearização do robô móvel Twil com aceleração linear e angular constantes.

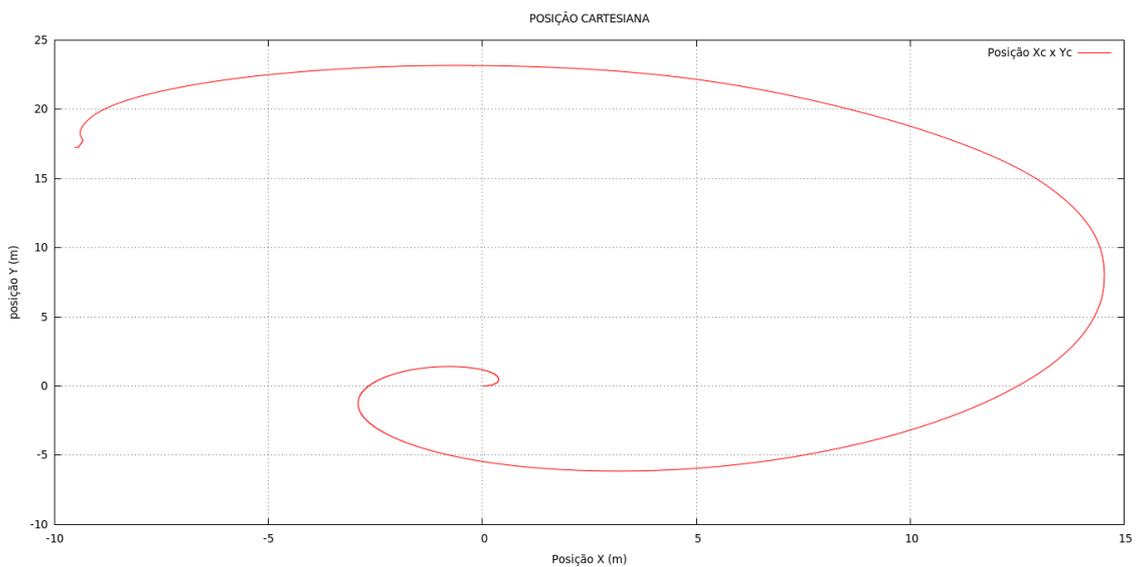


Figura 64: Posição $X \times Y$, na simulação da linearização do robô móvel Twil com aceleração linear e angular constantes.

5.5 Simulação do robô móvel Twil utilizando o controlador *backstepping*

Nesta simulação foram verificados os resultados referentes ao controlador *backstepping* proposto na seção 3.2.4, sendo que as Figuras 65, 66, 67, 68, 69 e 70 mostram respectivamente os torques nas rodas do robô, a posição do robô no eixo X , a posição do robô no eixo Y , a posição $X \times Y$ do robô, a orientação do robô e a evolução dos sinais η_1 , η_2 , u_1 e u_2 .

Pode-se concluir da análise das figuras que o controlador apresenta resultados condescizentes uma vez que o erro apresentado é pequeno e em parte deve-se à qualidade da estimacão da pose do robô, baseada em odometria. Ainda, na Figura 70 tem-se que os sinais u tendem a convergir para os sinais η em concordância com a teoria do controle *backstepping*.

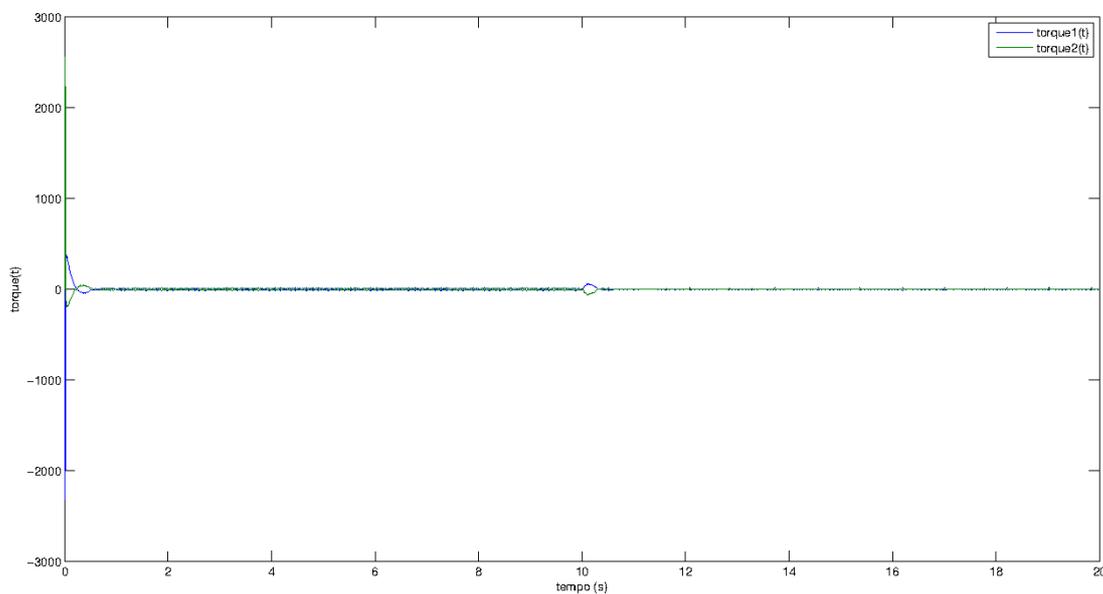


Figura 65: Simulação do controlador *backstepping* - torque nas rodas.

5.6 Simulação do manipulador Barrett WAM

Esta seção mostra a simulação do manipulador Barrett WAM com o controlador de torque calculado. A Figura 71 mostra o diagrama de nodos onde é possível observar quais são os tópicos que publicam e subscrevem no ROS e no Gazebo e a Figura 72 mostra o manipulador no Gazebo.

As Figuras 73 e 74 mostram respectivamente os resultados das posições das juntas e os torques aplicados em cada junta durante a simulação. Partindo-se de uma condição inicial com todas as juntas em zero, foi aplicado um degrau de 0.05 radianos na referência de posição de cada junta para evitar que o controlador por torque calculado gerasse valores muito elevados de torque. Para realizar grandes deslocamentos e evitar torques elevados deve-se usar um gerador de trajetória.

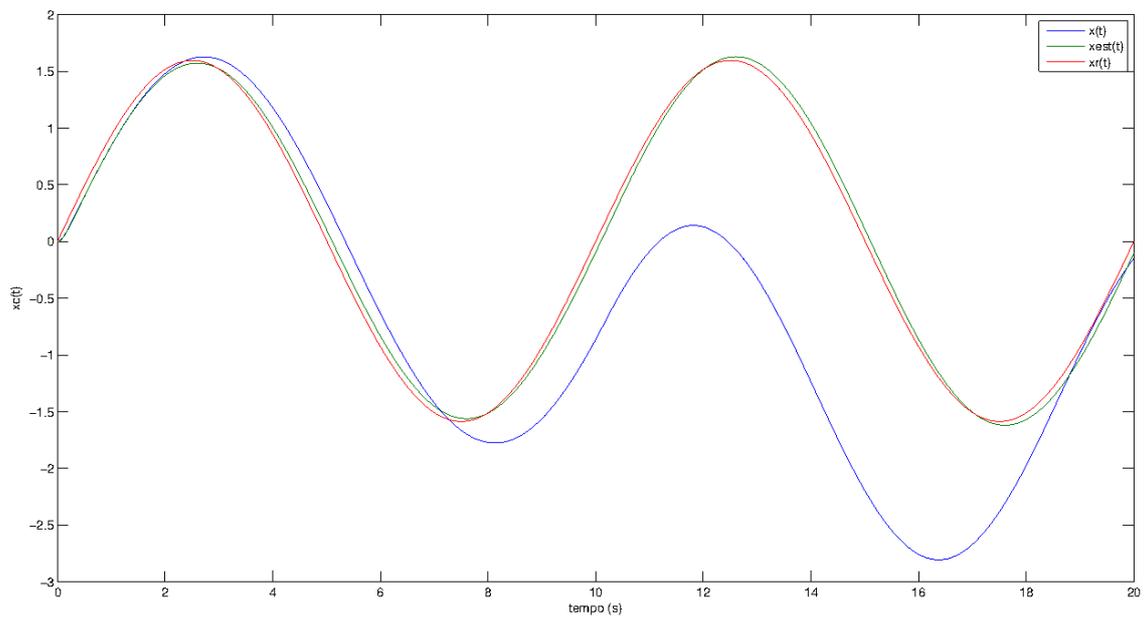


Figura 66: Simulação do controlador *backstepping* - posição X .

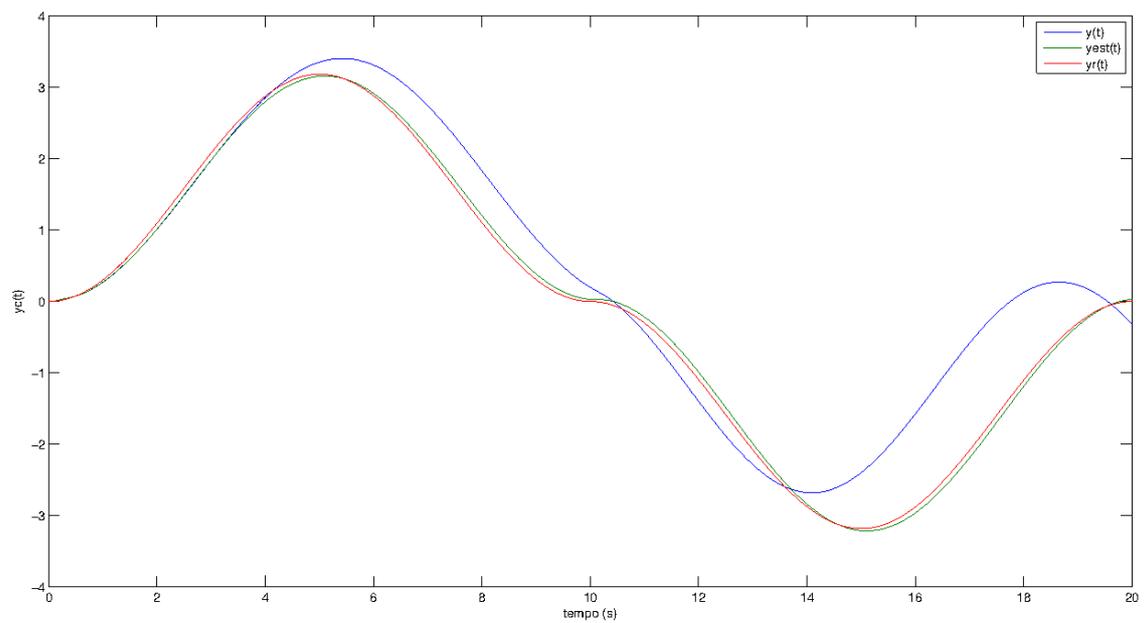


Figura 67: Simulação do controlador *backstepping* - posição Y .

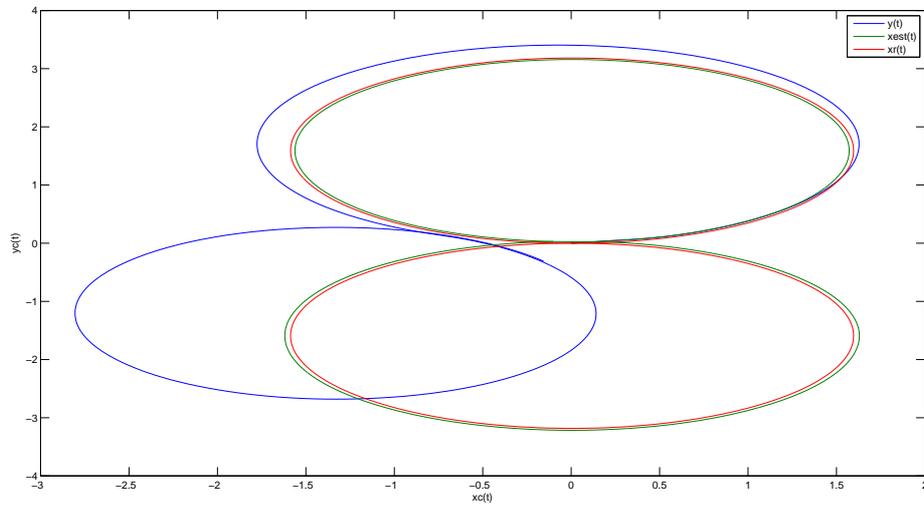


Figura 68: Simulação do controlador *backstepping* - posição $X \times Y$.

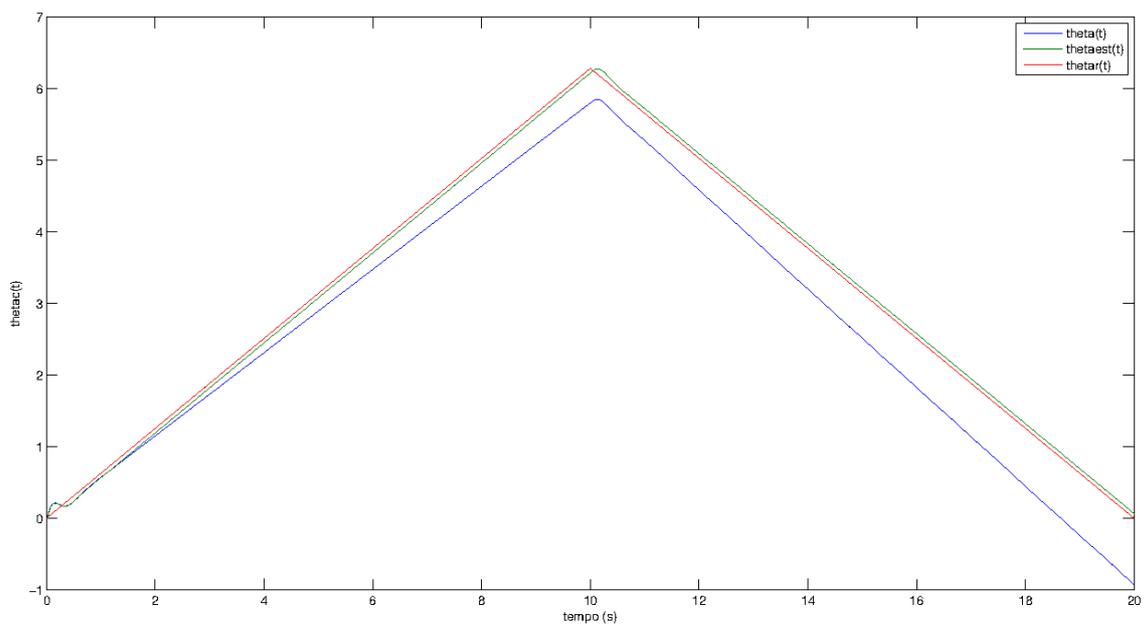


Figura 69: Simulação do controlador *backstepping* - orientação.

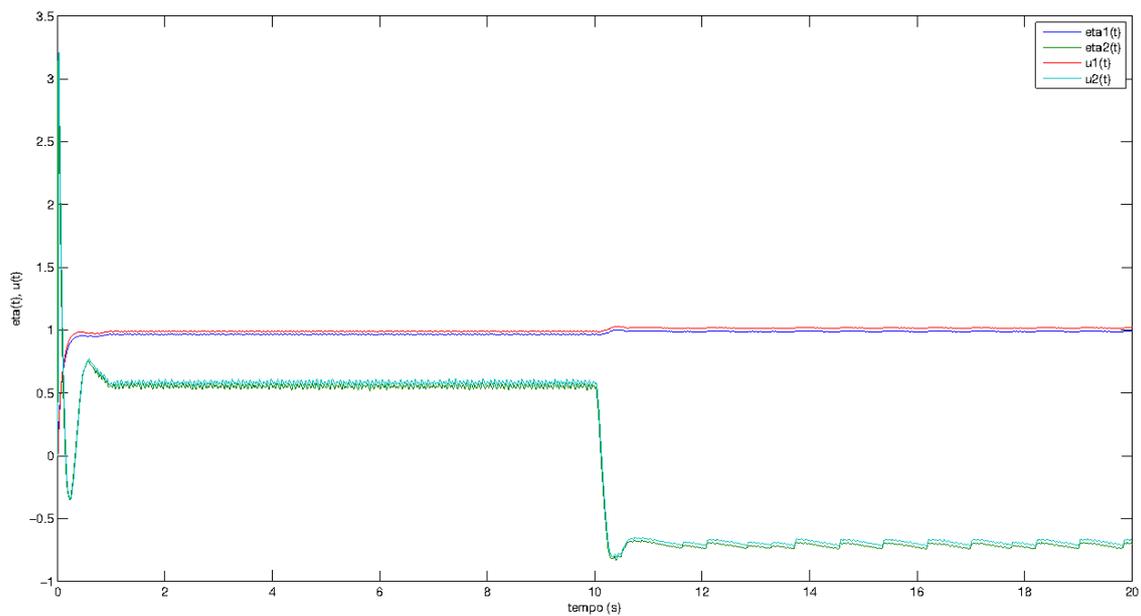


Figura 70: Simulação do controlador *backstepping* - η_1 , η_2 , u_1 e u_2 .

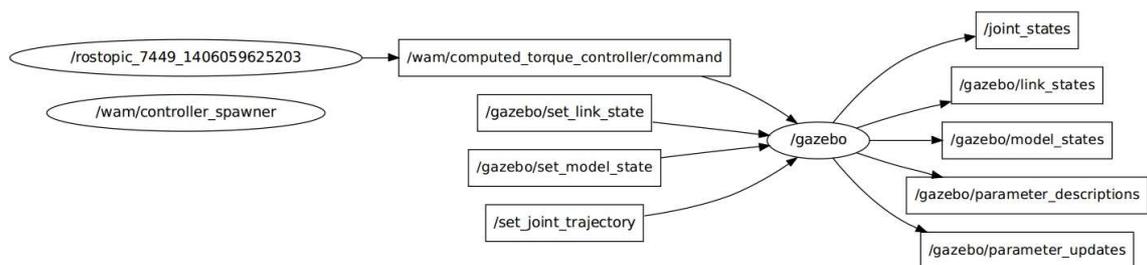


Figura 71: Diagrama de nós da simulação do manipulador Barrett WAM.

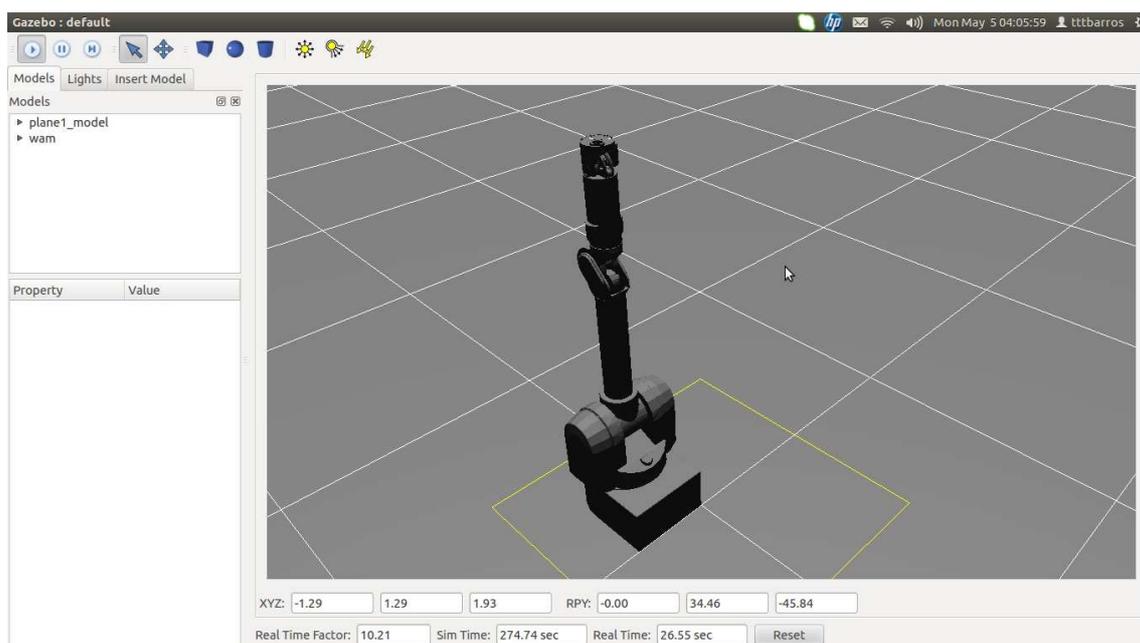


Figura 72: Manipulador Barrett WAM no Gazebo.

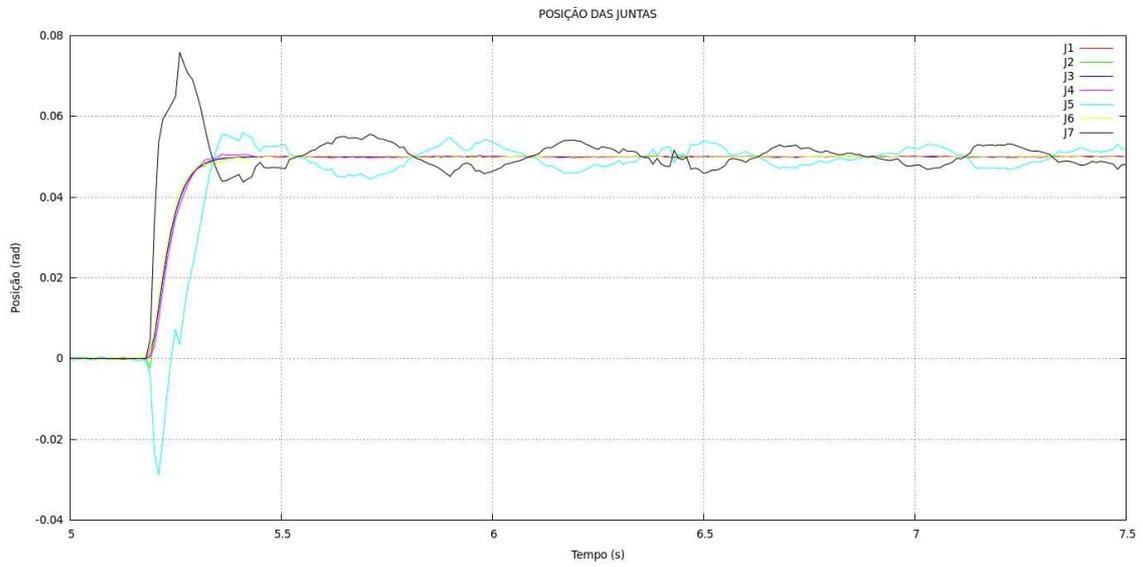


Figura 73: Posição das juntas.

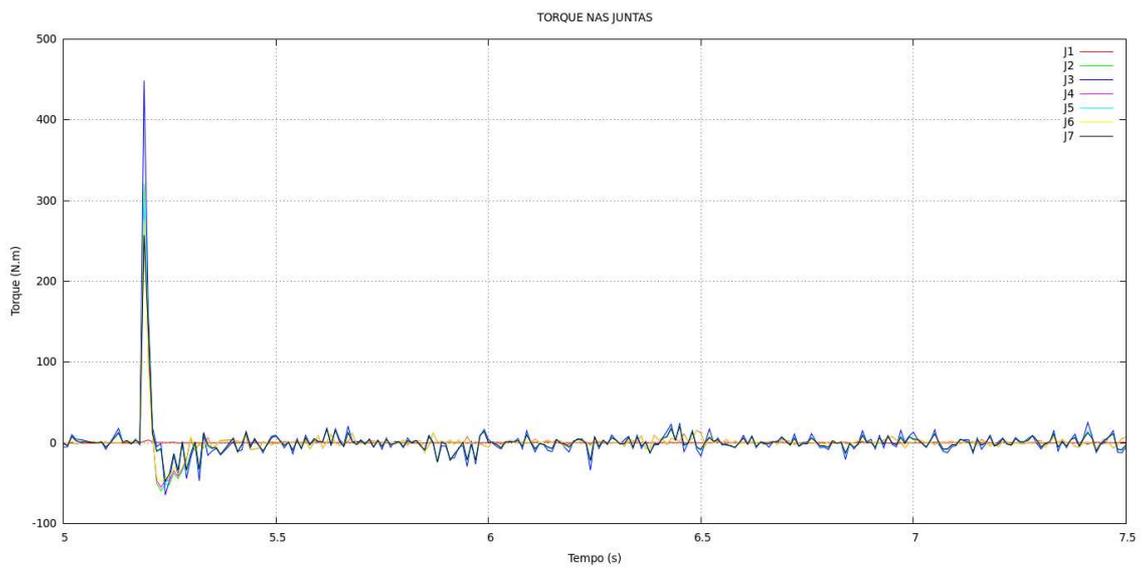


Figura 74: Torque aplicado nas juntas.

6 CONCLUSÕES

Este trabalho apresentou a modelagem matemática de um manipulador móvel. Os modelos matemáticos obtidos expressam as propriedades cinemáticas e dinâmicas do manipulador móvel. Utilizando os modelos desenvolvidos, também foi possível propor uma estratégia de controle para o manipulador móvel.

As ferramentas matemáticas disponibilizadas, permitem que novas análises sejam realizadas, como por exemplo, adequar a formulação da energia cinética da base móvel para permitir a integração de um número qualquer de rodas orientáveis centradas e rodas universais. A obtenção das expressões do modelo dinâmico da base móvel apresentado está totalmente descrito, podendo ser utilizado como referência para futuros trabalhos, sendo que durante o desenvolvimento deste modelo, observou-se que existem poucas referências que apresentem a formulação para obtenção do modelo dinâmico no que se trata de literatura de robôs móveis, diferentemente da literatura referente a robôs manipuladores, a qual possui vasta quantidade de material. Pode-se associar a este fato também a questão de que poucos trabalhos utilizam o modelo dinâmico de robôs móveis para desenvolver o controle, ao invés disso utilizando o modelo cinemático.

A modelagem matemática desenvolvida permitiu que as técnicas de controle de linearização por realimentação, *backstepping* e torque calculado fossem estudadas, juntamente com os tópicos de transformação linear e sistemas não holonômicos.

O sistema operacional para robôs ROS foi utilizado para testar a proposta de controle, sendo que os resultados apresentados na etapa de linearização e identificação permitem concluir que o controle proposto apresentou desempenho satisfatório uma vez que o modelo matemático desenvolvido para o manipulador é diferente do modelo simulado pelo ROS, sendo assim, esta diferença entre os modelos exige uma boa performance do controlador. Ainda uma descrição detalhada do funcionamento do laço de tempo real do ROS foi apresentada, sendo que a própria documentação do ROS não traz este nível de detalhamento.

Ao mesmo tempo, os controladores apresentados no ROS são em sua grande maioria, baseados em controladores PID e do tipo SISO, sendo que neste trabalho um controlador não linear do tipo MIMO foi desenvolvido, atendendo as características de sistema não holonômico apresentadas pelo robô móvel. Este tipo de implementação no ROS ainda é pouco difundido, sendo que este trabalho serve como base para futuras implementações que busquem utilizar o ROS como plataforma de desenvolvimento.

Os pacotes desenvolvidos para o robô móvel Twil e para o manipulador Barrett WAM servem como base para futuros desenvolvimentos, tanto do ponto de vista de documentação como de códigos de referência. Por mais que a documentação sobre o ROS seja vasta, ainda são poucos os trabalhos que exploram diferentes estratégias de controle e o modo de como implementar os mesmos.

Como trabalhos futuros propõe-se: o estudo de uma estratégia de controle adaptativo similar ao apresentado na Figura 75 que permita controlar a base móvel e o manipulador simultaneamente. Uma vez que o manipulador seja acoplado sobre o robô móvel Twil, os parâmetros de massa e inércia do manipulador móvel vão mudar em função da configuração do manipulador, sendo que estas alterações não foram consideradas na modelagem.

A indentificação *on-line* dos parâmetros do modelo permitiria ajustá-los para as variações provocadas pelo movimento do manipulador, e alterações nos parâmetros só será efetuada quando a covariância associada for pequena, para garantir a confiabilidade dos novos valores.

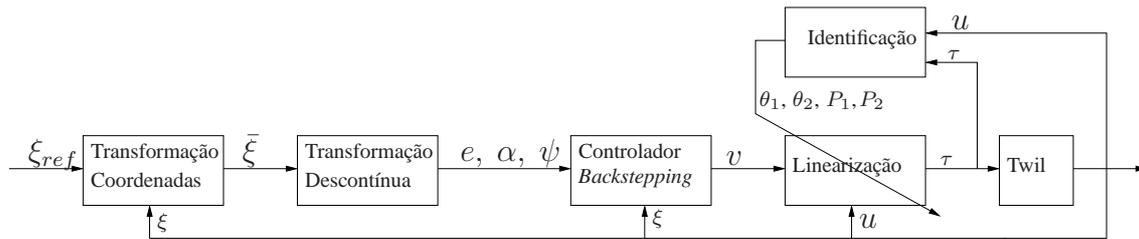


Figura 75: Diagrama de blocos do controlador adaptativo proposto para o manipulador móvel.

Ainda, o desenvolvimento de um modelo dinâmico conjunto da plataforma móvel com o manipulador pode permitir o desenvolvimento de novas estratégias de controle que venham a aumentar a robustez do sistema. Propõe-se também testar as estratégias de controle com o manipulador móvel real para verificação de desempenho e propostas de melhoria.

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à Identificação de Sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais**. 3^a.ed. Belo Horizonte: UFMG, 2007.
- ASTOLFI, A. On The Stabilization of Nonholonomic Systems. In: IEEE AMERICAN CONFERENCE ON DECISION AND CONTROL, 33., 1994, Lake Buena Vista. **Proceedings...** Piscataway: IEEE Press, 1994. p.3481–3486.
- BARNES, R. J. **Matrix Differentiation (and some other stuff)**. Disponível em: <<http://www.atmos.washington.edu/~dennis/MatrixCalculus.pdf>>. Acesso em: 10 Jun. 2013.
- BARRETT TECHNOLOGY. **Document: d1001 - wam arm user's guide**. AE.00.ed. Cambridge: Barrett Technology, 2006.
- BARROS, T.; LAGES, W. F. Development of a Firefighting Robot for Educational Competitions. In: RIE - INTERNATIONAL CONFERENCE ON ROBOTICS IN EDUCATION, 3., 2012, Prague. **Proceedings...** Prague: MATFYZPRESS, 2012. p.47–54.
- BEARINGBOYS. **2BZD75GR 75mm Grey Rubber Tyre Twin Wheel Castor - Swivel 4 Bolt Unbraked**. Disponível em: <http://www.bearingboys.co.uk/2BZD75GR_75mm_Grey_Rubber_Tyre_Twin_Wheel_Castor_-_Swivel_4_Bolt_Unbraked-1033-p#>. Acesso em: 10 Fev. 2013.
- BROCKETT, R. W. **New Directions in Applied Mathematics**. New York: Springer-Verlag, 1982.
- BRUNNER, M.; BRÜGGEMANN, B.; SCHULZ, D. Towards Autonomously Traversing Complex Obstacles with Mobile Robots with Adjustable Chassis. In: INTERNATIONAL CARPATHIAN CONTROL CONFERENCE (ICCC), 13., 2012, Podbanské, Slovak Republic. **Proceedings...** Podbanské: Slovak Society for Applied Cybernetics and Informatics, 2012. p.63–68.
- BRUYNINCKX, H. Open Robot Control Software: the orocos project. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2001, Seoul. **Proceedings...** Hoboken: IEEE Press, 2001. p.2523–2528.

BUYHAFELE. **Ball Caster, steel, yellow chrome, 6 kilogram, 13mm**. Disponível em: <http://www.buyhafele.com/index.php?main_page=product_info&products_id=16758>. Acesso em: 22 Mar. 2013.

CAMPION, G.; BASTIN, G.; D'ANDRÉA-NOVEL, B. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. **IEEE Transactions on Robotics and Automation**, Atlanta, v.12, n.1, p.47–62, May 1993.

CANUDAS DE WIT, C.; SØRDALEN, O. J. Exponential Stabilization of Mobile Robots with Nonholonomic Constraints. **IEEE Transactions on Automatic Control**, Piscataway, v.37, n.11, p.1791–1797, Nov. 1992.

CORRELL, N.; WING, R.; COLEMAN, D. A One-Year Introductory Robotics Curriculum for Computer Science Upperclassmen. **IEEE Transactions on Education**, Piscataway, v.56, p.54–60, Feb. 2013.

COUSINS, S. Welcome to ROS Topics. **IEEE Robotics and Automation Magazine**, Piscataway, v.17, p.13–14, Mar. 2010.

CRAIG, J. J. **Introduction to Robotics: mechanics and control**. Upper Saddle River: Pearson Education, Incorporated, 2005.

DEMARCO, K.; WEST, M. E.; COLLINS, T. R. An implementation of ROS on the Yellowfin autonomous underwater vehicle (AUV). In: OCEANS 2011, 2011, Hilton Waikoloa Village. **Proceedings...** Kona: IEEE Oceanic Engineering Society, 2011. p.1–7.

DIETRICH, A. et al. Reactive Whole-Body Control: dynamic mobile manipulation using a large number of actuated degrees of freedom. **IEEE Robotics and Automation Magazine**, Piscataway, v.19, p.20–33, Jun. 2012.

DUŠEK, F.; HONC, D.; ROZSÍVAL, P. Mathematical Model of Differentially Steered Mobile Robot. In: INTERNATIONAL CONFERENCE ON PROCESS CONTROL, 18., 2011, Tatranská Lomnica. **Proceedings...** Bratislava: Slovak University of Technology in Bratislava, 2011. p.221–229.

FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In: IEEE CONFERENCE ON DECISION AND CONTROL, 34., 1995, New Orleans. **Proceedings...** New Orleans: IEEE Press, 1995. v.4, p.3805–3810.

FINGERTECHROBOTICS. **FingerTech Mecanum Wheels (Set of 4)-Competition Grade**. Disponível em: <http://www.fingertechrobotics.com/prodimages/wheels/CG_wheel_A_lg.jpg>. Acesso em: 02 Jan. 2013.

FU, K. S.; GONZALES, R. C.; LEE, C. S. G. **Robotics Control, Sensing, Vision and Intelligence**. New York: McGraw-Hill, 1987. (Industrial Engineering Series).

FUKUSHIMA, E. F.; HIROSE, S.; HAYASHI, T. Basic Manipulation Considerations For The Articulated Body Mobile Robot. In: IEEE/RSJ INTL. CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 1998, British Columbia. **Proceedings...** Victoria: IEEE Press, 1998. p.386–393.

GARCÍA, J. F. C. **Análisis Cinemático, Dinámico y Control en Tiempo Real de un Vehículo Guiado Automáticamente**. 2009. 137p. Tesis de Maestría en Ciencias (Maestría en Ciencias en Ingeniería Mecatrónica) — Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, 2009.

GITHUB. **ROS Tutorial 5. Basic ROS Programming**. Disponível em: <<https://github.com/uscrs-art/uscrs-art/wiki/ROS-Tutorial-5.-Basic-ROS-Programming>>. Acesso em: 14 Fev. 2014.

GODHAVN, J.; EGELAND, O. A Lyapunov Approach to Exponential Stabilization of Nonholonomic Systems in Power Form. **IEEE Transactions on Automatic Control**, Piscataway, New Jersey, United States, v.42, n.7, p.1028–1032, July 1997.

GONG, S. et al. ROS-based Object Localization Using RFID and Laser Scan. In: INTERNATIONAL CONFERENCE ON INFORMATION AND AUTOMATION (ICIA), 2012, Shenyang. **Proceedings...** Piscataway: IEEE Robotics and Automation Society, 2012. p.406–411.

HAMNER, B. et al. An autonomous mobile manipulator for assembly tasks. **Autonomous Robot**, Hingham, v.28, p.131–149, Jan. 2010.

IDE, S. et al. Real-time Trajectory Planning for Mobile Manipulator Using Model Predictive Control with Constraints. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS ROBOTS AND AMBIENT INTELLIGENCE (URAI), 8., 2011, Incheon. **Proceedings...** Detroit: IEEE Press, 2011. p.244–249.

IORIS, D.; LAGES, W. F.; SANTINI, D. C. Integrating the OROCOS Framework and the Barrett WAM Robot. In: WORKSHOP ON APPLIED ROBOTICS AND AUTOMATION, 5., 2012, Bauru. **Proceedings...** Bauru: Unesp, 2012. p.1–8.

KERR, J.; NICKELS, K. Robot operating systems: bridging the gap between human and robot. In: SOUTHEASTERN SYMPOSIUM ON SYSTEM THEORY (SSST), 44., 2012, Jacksonville. **Proceedings...** Jacksonville: IEEE, 2012. p.99–104.

KOKOTOVIĆ, P. V. Developments in Nonholonomic Control Problems. **IEEE Control Systems Magazine**, Piscataway, New Jersey, United States, v.12, n.3, p.7–17, June 1992.

LAGES, W. F. **Controle e Estimção de Posição e Orientação de Robôs Móveis**. 1998. 180p. Tese (Doutorado em Engenharia Eletrônica e Computação) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 1998.

LAGES, W. F.; IORIS, D.; SANTINI, D. An Architecture for Controlling the Barrett WAM Robot Using ROS and OROCOS. In: JOINT 45TH INTERNATIONAL SYMPOSIUM ON ROBOTICS AND 8TH GERMAN CONFERENCE ON ROBOTICS, 2014, Munich. **Proceedings...** Berlin: VDE, 2014.

LUCIBELLO, P.; ORIOLO, G. Robust stabilization via iterative state steering with an application to chained-form systems. **Automatica**, Amsterdam, v.37, n.1, p.71–79, Jan. 2001.

MACIEL, E. H. **Desenvolvimento de um Modelo Simplificado dos Membros Inferiores de um Robô Bípede Utilizando ROS**. 2014. 109p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

MAILAH, M.; PITOWARNO, E.; JAMALUDDIN, H. Robust Motion Control for Mobile Manipulator Using Resolved Acceleration and Proportional-Integral Active Force Control. **International Journal of Advanced Robotic Systems**, Skudai, v.2, p.125–134, Dec. 2005.

MICHIELETTO, S.; ZANIN, D.; MENEGATTI, E. NAO Robot Simulation for Service Robotics Purposes. In: EUROPEAN MODELLING SYMPOSIUM - EMS, 2013, Manchester. **Proceedings...** Malta: IEEE Press, 2013. p.477–482.

MURRAY, R. M.; LI, Z.; SASTRY, S. **A Mathematical Introduction to Robotics Manipulation**. Boca Raton: CRC Press, 1994.

NAGASAKA, K. et al. Whole-body Cooperative Force Control for a Two-Armed and Two-Wheeled Mobile Robot Using Generalized Inverse Dynamics and Idealized Joint Units. **International Conference on Robotics and Automation**, Alaska, p.3377–3383, May 2010.

PAUL, R. P. **Robot Manipulators: mathematics, programming and control**. Cambridge: MIT Press, 1981. (The MIT Press Series in Artificial Intelligence).

PETROV, P. Modeling and Adaptive Path Control of a Differential Drive Mobile Robot. In: WSEAS INTERNATIONAL CONFERENCE ON AUTOMATIC CONTROL, MODELLING & SIMULATION, 12., 2010, Stevens Point. **Proceedings...** Wisconsin: World Scientific and Engineering Academy and Society (WSEAS), 2010. p.403–408.

POMET, J. B. et al. A Hybrid Strategy for the Feedback Stabilization of Nonholonomic Mobile Robots. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1992, Nice. **Proceedings...** Piscataway: IEEE Press, 1992. p.129–134.

POPOV, V. M. **Hyperstability of Control Systems**. Berlin: Springer-Verlag, 1973. (Die Grundlehren der mathematischen Wissenschaften, v.204).

QUIGLEY, M. et al. ROS: an open-source robot operating system. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION ICRA, WORKSHOP ON OPEN SOURCE ROBOTICS, 2009, Kobe. **Proceedings...** Piscataway: IEEE Press, 2009. p.1–6.

RCCRAWLER. **RC Crawler Steering**. Disponível em:
<<http://www.rccrawler.com/forum/general-crawlers/366295-rc-crawler-steering.html>>. Acesso em: 10 Fev. 2013.

REHMAN, F.; RAFIQ, M.; RAZA, Q. Time-varying Stabilizing Feedback Control for a Sub-class of Nonholonomic Systems. **European Journal of Scientific Research**, Victoria, Mahé, Seychelles, v.53, n.3, p.346–358, May. 2011.

ROBOT OPERATING SYSTEM. **ROS - INDUSTRIAL**. Disponível em:
<<http://rosindustrial.org/>>. Acesso em: 20 Fev. 2014.

ROBOT OPERATING SYSTEM. **Robot Operating System**. Disponível em:
<<http://www.ros.org/>>. Acesso em: 20 Jan. 2014.

ROBOT OPERATING SYSTEM. **Urdf**. Disponível em:
<<http://wiki.ros.org/urdf>>. Acesso em: 14 Fev. 2014.

ROBOT OPERATING SYSTEM. **Robots**. Disponível em:
<<http://wiki.ros.org/Robots>>. Acesso em: 14 Fev. 2014.

ROBOTSHOP. **3WD Compact Omni-Directional Arduino Compatible Mobile Robot Kit**. Disponível em: <<http://www.robotshop.com/ca/3wd-omni-directional-arduino-mobile-robot-kit-3.html>>. Acesso em: 02 Jan. 2013.

ROBOTSHOP. **Tetra-DS IV Mobile Robot Platform**. Disponível em:
<<http://www.robotshop.com/productinfo.aspx?pc=RB-Das-09&lang=en-US>>. Acesso em: 02 Jan. 2013.

ROS.ORG. **solid works urdf exporter**. Available at:
<http://wiki.ros.org/sw_urdf_exporter>. Access: December 2013.

SANTINI, D. C.; LAGES, W. F. An Architecture for Robot Control Based on the OROCOS Framework. In: WORKSHOP ON APPLIED ROBOTICS AND AUTOMATION, 4., 2010, Bauru. **Proceedings...** Bauru: Unesp, 2010. p.1–10.

SICILIANO, B. et al. **Robotics: modelling, planning and control**. London: Springer, 2009.

SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics**. Stanford: Springer-Verlag, 2008.

SIDEK, N.; SARKAR, N. Dynamic Modeling and Control of Nonholonomic Mobile Robot With Lateral Slip. In: WSEAS INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING, ROBOTICS AND AUTOMATION, 7., 2008. **Proceedings...** Wisconsin: WSEAS: address =, 2008. p.66–74.

SINGH, G. K.; CLAASSENS, J. An Analytical Solution for the Inverse Kinematics of a Redundant 7DoF Manipulator with link Offsets. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2010, Taipei. **Proceedings...** Taipei: IEEE, 2010. p.2976–2982.

SØRDALEN, O. J. **Feedback Control of Nonholonomic Mobile Robots**. 1993. 140p. Tese (Doutorado em Ciência da Computação) — The Norwegian Institute of Technology, Trondheim, 1993.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. New York: Wiley, 2006.

TANNER, H. G.; KYRIAKOPOULOS, K. J. Backstepping for Nonsmooth Systems. **Automatica**, Tarrytown, v.39, n.7, p.1259–1265, July 2003.

TEEL, A. R.; MURRAY, R. M.; WALSH, G. C. Non-holonomic Control Systems: from steering to stabilization with sinusoids. **International Journal of Control**, London, UK, v.62, n.4, p.849–870, 1995.

VEXROBOTICS. **2.75 Omni Directional Wheel - Double Roller (2-pack)**. Disponível em: <<http://www.vexrobotics.com/276-1902.html>>. Acesso em: 02 Jan. 2013.

WIKIPEDIA. **STL (file format)**. Available at:

<[http://en.wikipedia.org/wiki/STL_\(file_format\)](http://en.wikipedia.org/wiki/STL_(file_format))>. Access: March 2014.

WONG, B. **Cooperation leads to smarter robots**. Disponível em:

<<http://electronicdesign.com/android/cooperation-leads-smarter-robots>>. Acesso em: 18 Out. 2013.

YAMAMOTO, Y.; YUN, X. Unified Analysis on Mobility and Manipulability of Mobile Manipulators. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1999, Detroit. **Proceedings...** Piscataway: IEEE Press, 1999. p.1200–1206.

YU, Q.; CHEN, I.-M. A General Approach to the Dynamics of Nonholonomic Mobile Manipulator Systems. **Journal of Dynamic Systems, Measurement, and Control**, Singapore, v.124, p.512–521, Dec. 2002.

ZAMAN, S.; SLANY, W.; STEINBAUER, G. ROS-based Mapping, Localization and Autonomous Navigation Using a Pioneer 3-DX Robot and Their Relevant Issues. In: ELECTRONICS, COMMUNICATIONS AND PHOTONICS CONFERENCE (SIECPC), 2011, Riyadh. **Proceedings...** Piscataway: IEEE Press, 2011. p.1–5.

APÊNDICE A REGRAS DE DERIVAÇÃO DE VETORES E MATRIZES

A obtenção das expressões demonstradas neste trabalho são baseadas nas regras de derivação apresentadas neste apêndice. Estas regras foram formuladas seguindo as propostas de BARNES (2006) e MURRAY; LI; SASTRY (1994).

1. Vetores

Exceto em casos específicos, por definição todos os vetores utilizados serão vetores coluna. Assim, para um dado vetor x com n componentes tem-se

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

2. Derivada de um vetor com relação a outro vetor

Dados os vetores x e y com dimensões $n \times 1$ e $m \times 1$ respectivamente, a derivada de y em relação a x será dada por

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \dots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

3. Derivada de um escalar com relação a um vetor

Dados um escalar α e um vetor x com dimensão $n \times 1$, a derivada de α em relação a x será dada por

$$\frac{\partial \alpha}{\partial x} = \left[\frac{\partial \alpha}{\partial x_1} \quad \frac{\partial \alpha}{\partial x_2} \quad \dots \quad \frac{\partial \alpha}{\partial x_n} \right]$$

4. Derivada de um escalar com relação a uma matriz

Dados um escalar α e uma matriz A com dimensões $m \times n$, a derivada de α em relação a A será dada por

$$\frac{\partial \alpha}{\partial A} = \begin{bmatrix} \frac{\partial \alpha}{\partial a_{11}} & \frac{\partial \alpha}{\partial a_{12}} & \cdots & \frac{\partial \alpha}{\partial a_{1n}} \\ \frac{\partial \alpha}{\partial a_{21}} & \frac{\partial \alpha}{\partial a_{22}} & \cdots & \frac{\partial \alpha}{\partial a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \alpha}{\partial a_{m1}} & \frac{\partial \alpha}{\partial a_{m2}} & \cdots & \frac{\partial \alpha}{\partial a_{mn}} \end{bmatrix}$$

5. *Derivada de uma matriz com relação a um escalar*

Dados um escalar α e uma matriz A com dimensões $m \times n$, a derivada de A em relação a α será dada por

$$\frac{\partial A}{\partial \alpha} = \begin{bmatrix} \frac{\partial a_{11}}{\partial \alpha} & \frac{\partial a_{12}}{\partial \alpha} & \cdots & \frac{\partial a_{1n}}{\partial \alpha} \\ \frac{\partial a_{21}}{\partial \alpha} & \frac{\partial a_{22}}{\partial \alpha} & \cdots & \frac{\partial a_{2n}}{\partial \alpha} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial a_{m1}}{\partial \alpha} & \frac{\partial a_{m2}}{\partial \alpha} & \cdots & \frac{\partial a_{mn}}{\partial \alpha} \end{bmatrix}$$

6. *Derivada de um vetor com relação a um escalar*

Dados um vetor x com dimensão $n \times 1$ e um escalar α , a derivada de x em relação a α será dada por

$$\frac{\partial x}{\partial \alpha} = \begin{bmatrix} \frac{\partial x_1}{\partial \alpha} \\ \frac{\partial x_2}{\partial \alpha} \\ \vdots \\ \frac{\partial x_n}{\partial \alpha} \end{bmatrix}$$

7. *Derivada de termos quadráticos de um vetor*

Dado um vetor x com dimensão $n \times 1$ tem-se que

$$\frac{\partial x^T x}{\partial x} = 2x^T$$

Prova:

Considerando-se que o termo $x^T x$ é uma função do vetor x na forma

$$x^T x = f(x) = f(x_1, x_2, \dots, x_n) = x_1^2 + x_2^2 + \cdots + x_n^2$$

pode-se escrever

$$\begin{aligned} \frac{\partial f(x)}{\partial x} &= \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \frac{\partial f(x)}{\partial x_2} & \cdots & \frac{\partial f(x)}{\partial x_n} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial (x_1^2 + x_2^2 + \cdots + x_n^2)}{\partial x_1} & \frac{\partial (x_1^2 + x_2^2 + \cdots + x_n^2)}{\partial x_2} & \cdots & \frac{\partial (x_1^2 + x_2^2 + \cdots + x_n^2)}{\partial x_n} \end{bmatrix} \\ &= [2x_1 \quad 2x_2 \quad \cdots \quad 2x_n] = 2 [x_1 \quad x_2 \quad \cdots \quad x_n] = 2x^T \end{aligned}$$

8. *Derivada envolvendo o produto de termos quadráticos de um vetor por uma matriz*

Dado um escalar α na forma quadrática $\alpha = x^T A x$ com x tendo dimensão $n \times 1$ e a A tendo dimensões $n \times n$, onde A não depende de x , tem-se que

$$\frac{\partial \alpha}{\partial x} = x^T (A + A^T)$$

e no caso particular onde A é simétrica

$$\frac{\partial \alpha}{\partial x} = x^T (A + A^T) = x^T (A + A) = 2x^T A$$

Prova:

Conforme definição de α pode-se escrever

$$\alpha = \sum_{j=1}^n \sum_{i=1}^n a_{ij} x_i x_j$$

derivando-se α com relação ao k -ésimo ($k = 1, 2, \dots, n$) termo de x tem-se

$$\frac{\partial \alpha}{\partial x_k} = \sum_{j=1}^n a_{kj} x_j + \sum_{i=1}^n a_{ik} x_i$$

e conseqüentemente

$$\frac{\partial \alpha}{\partial x} = x^T A^T + x^T A = x^T (A + A^T)$$

9. Derivada do produto de dois vetores

Dado um vetor x com dimensão $n \times 1$ e um vetor y com dimensão $m \times 1$ tem-se que

$$\frac{\partial x^T y}{\partial x} = y^T$$

Prova:

Considerando-se que o termo $x^T y$ é uma função dos vetores x e y na forma

$$x^T y = f(x, y) = x_1 y_1 + x_2 y_2 + \dots + x_n y_m$$

pode-se escrever

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &= \left[\frac{\partial f(x, y)}{\partial x_1} \quad \frac{\partial f(x, y)}{\partial x_2} \quad \dots \quad \frac{\partial f(x, y)}{\partial x_n} \right] \\ &= \left[\frac{\partial (x_1 y_1 + x_2 y_2 + \dots + x_n y_m)}{\partial x_1} \quad \frac{\partial (x_1 y_1 + x_2 y_2 + \dots + x_n y_m)}{\partial x_2} \quad \dots \quad \frac{\partial (x_1 y_1 + x_2 y_2 + \dots + x_n y_m)}{\partial x_n} \right] \\ &= [y_1 \quad y_2 \quad \dots \quad y_m] = y^T \end{aligned}$$

10. *Derivada envolvendo produto de vetores com uma matriz*

Para um escalar α dado por $\alpha = y^T A x$ onde y tem dimensão $m \times 1$, x tem dimensão $n \times 1$, A tem dimensão $m \times n$, y e x são funções do vetor z e A não depende de z , tem-se que

$$\frac{\partial \alpha}{\partial z} = x^T A^T \frac{\partial y}{\partial z} + y^T A^T \frac{\partial x}{\partial z}$$

No caso particular onde $x = y$, tem-se α dado por $\alpha = x^T A x$ tem-se

$$\frac{\partial \alpha}{\partial z} = x^T (A + A^T) \frac{\partial x}{\partial z}$$

e ainda, se A for simétrica tem-se que

$$\frac{\partial \alpha}{\partial z} = 2x^T A \frac{\partial x}{\partial z}$$

APÊNDICE B ENERGIA CINÉTICA DA PLATAFORMA MÓVEL

Este desenvolvimento demonstra a obtenção de (69) que expressa a energia cinética da base móvel do manipulador móvel que é objetivo deste trabalho. A energia cinética T da base móvel, pode ser expressa pela soma das energias cinéticas individuais do chassi (T_c), de cada uma das rodas fixas (T_f) e de cada uma das rodas não centradas (T_o). Assim, pode-se escrever

$$T = T_c + \sum_{i=1}^{nf} T_{f_i} + \sum_{i=1}^{no} T_{o_i}.$$

Para cada uma destas parcelas, utiliza-se a definição da energia cinética na forma $\frac{1}{2}v^T m v$, verificando-se qual a velocidade v e a matriz relativa às massas m de cada parte da base móvel.

B.1 Energia Cinética do Chassi

A energia cinética relativa ao chassi do robô móvel pode ser expressa por

$$T_c = \frac{1}{2} \left({}^c R_0^0 \dot{\xi}_c \right)^T M_c {}^c R_0^0 \dot{\xi}_c = \frac{1}{2} {}^0 \dot{\xi}_c^T {}^c R_0^T M_c {}^c R_0^0 \dot{\xi}_c$$

onde

$$M_c = \begin{bmatrix} m_c & 0 & 0 \\ 0 & m_c & 0 \\ 0 & 0 & I_c \end{bmatrix}$$

com m_c sendo a massa e I_c o momento de inércia do chassi em torno do eixo Z_c .

B.2 Energia Cinética de uma Roda Fixa

Para a energia cinética de cada uma das rodas fixas utiliza-se:

$$T_f = \frac{1}{2} {}^w \dot{\xi}_w^T M_f {}^w \dot{\xi}_w + \frac{1}{2} \dot{\phi}_f^T I_{f_{\hat{y}}} \dot{\phi}_f$$

onde

$$M_f = \begin{bmatrix} m_f & 0 & 0 \\ 0 & m_f & 0 \\ 0 & 0 & I_{f\hat{z}} \end{bmatrix}$$

com m_f sendo a massa de cada uma das rodas fixas, $I_{f\hat{y}}$ o momento de inércia da roda em relação ao eixo y , $I_{f\hat{z}}$ o momento de inércia da roda em relação ao eixo Z_c e ${}^w\dot{\xi}_w$ expressa a velocidade à qual cada uma das rodas fixas do robô está submetida (14). Assim, tem-se que:

$$T_f = \frac{1}{2} {}^0\dot{\xi}_c^T {}^cR_0^T \begin{bmatrix} \sin(\alpha_f + \beta_f) & \cos(\alpha_f + \beta_f) & 0 \\ -\cos(\alpha_f + \beta_f) & \sin(\alpha_f + \beta_f) & 0 \\ -l \cos(\beta_f) & l \sin(\beta_f) & 1 \end{bmatrix} \begin{bmatrix} m_f & 0 & 0 \\ 0 & m_f & 0 \\ 0 & 0 & I_{f\hat{z}} \end{bmatrix} \\ \begin{bmatrix} \sin(\alpha_f + \beta_f) & -\cos(\alpha_f + \beta_f) & -l \cos(\beta_f) \\ \cos(\alpha_f + \beta_f) & \sin(\alpha_f + \beta_f) & l \sin(\beta_f) \\ 0 & 0 & 1 \end{bmatrix} {}^cR_0^0\dot{\xi}_c + \frac{1}{2} \dot{\varphi}_f^T I_{f\hat{y}} \dot{\varphi}_f$$

E efetuando-se a multiplicação das três matrizes, obtém-se

$$T_f = \frac{1}{2} {}^0\dot{\xi}_c^T {}^cR_0^T \begin{bmatrix} m_f & 0 & -\sin(\alpha_f) l m_f \\ 0 & m_f & \cos(\alpha_f) l m_f \\ -\sin(\alpha_f) l m_f & \cos(\alpha_f) l m_f & l^2 m_f + I_{f\hat{z}} \end{bmatrix} {}^cR_0^0\dot{\xi}_c + \frac{1}{2} \dot{\varphi}_f^T I_{f\hat{y}} \dot{\varphi}_f$$

Ainda, definindo-se

$$M_f = \begin{bmatrix} m_f & 0 & -\sin(\alpha_f) l m_f \\ 0 & m_f & \cos(\alpha_f) l m_f \\ -\sin(\alpha_f) l m_f & \cos(\alpha_f) l m_f & l^2 m_f + I_{f\hat{z}} \end{bmatrix}$$

pode-se escrever

$$T_{f_i} = \frac{1}{2} {}^0\dot{\xi}_c^T {}^cR_0^T M_{f_i} {}^cR_0^0\dot{\xi}_c + \frac{1}{2} \dot{\varphi}_{f_i}^T I_{f\hat{y}_i} \dot{\varphi}_{f_i}$$

B.3 Energia Cinética de uma Roda Não Centrada

A energia cinética de cada uma das rodas não centradas do robô pode ser obtida utilizando-se (25) que expressa a velocidade à qual cada uma destas rodas está submetida. Assim:

$$T_o = \frac{1}{2} {}^w\dot{\xi}_w^T M_o {}^w\dot{\xi}_w + \frac{1}{2} \dot{\varphi}_o^T I_{o\hat{y}} \dot{\varphi}_o$$

onde

$$M_o = \begin{bmatrix} m_o & 0 & 0 \\ 0 & m_o & 0 \\ 0 & 0 & I_{o\hat{z}} \end{bmatrix}$$

observando-se que m_o compreende a massa da roda e do mecanismo de sustentação da mesma (o qual caracteriza o tipo de movimentação da roda), além disso, o cálculo de I_{o_z} considera apenas a massa da roda. Substituindo-se ${}^w\dot{\xi}_w$ de (25) no primeiro termo da expressão de T_o obtém-se:

$$T_o = \frac{1}{2} \left({}^0\dot{\xi}_c^T {}^c R_0^T \begin{bmatrix} \sin(\alpha_o + \beta_o) & \cos(\alpha_o + \beta_o) & 0 \\ -\cos(\alpha_o + \beta_o) & \sin(\alpha_o + \beta_o) & 0 \\ -l \cos(\beta_o) & l \sin(\beta_o) + d & 1 \end{bmatrix} + [0 \quad d\dot{\beta}_o \quad \dot{\beta}_o] \right) \\ \begin{bmatrix} m_o & 0 & 0 \\ 0 & m_o & 0 \\ 0 & 0 & I_{o_z} \end{bmatrix} \left(\begin{bmatrix} \sin(\alpha_o + \beta_o) & -\cos(\alpha_o + \beta_o) & -l \cos \beta_o \\ \cos(\alpha_o + \beta_o) & \sin(\alpha_o + \beta_o) & l \sin \beta_o + d \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0 {}^0\dot{\xi}_c + \begin{bmatrix} 0 \\ d\dot{\beta}_o \\ \dot{\beta}_o \end{bmatrix} \right) \\ + \frac{1}{2} \dot{\varphi}_o^T I_{o_{\hat{y}}} \dot{\varphi}_o$$

$$T_o = \frac{1}{2} \left({}^0\dot{\xi}_c^T {}^c R_0^T \begin{bmatrix} \sin(\alpha_o + \beta_o) m_o & \cos(\alpha_o + \beta_o) m_o & 0 \\ -\cos(\alpha_o + \beta_o) m_o & \sin(\alpha_o + \beta_o) m_o & 0 \\ -l \cos(\beta_o) m_o & (l \sin(\beta_o) + d) m_o & I_{o_z} \end{bmatrix} \right. \\ \left. + [0 \quad d\dot{\beta}_o m_o \quad \dot{\beta}_o I_{o_z}] \right) \left(\begin{bmatrix} \sin(\alpha_o + \beta_o) & -\cos(\alpha_o + \beta_o) & -l \cos \beta_o \\ \cos(\alpha_o + \beta_o) & \sin(\alpha_o + \beta_o) & l \sin \beta_o + d \\ 0 & 0 & 1 \end{bmatrix} {}^c R_0 {}^0\dot{\xi}_c \right. \\ \left. + \begin{bmatrix} 0 \\ d\dot{\beta}_o \\ \dot{\beta}_o \end{bmatrix} \right) + \frac{1}{2} \dot{\varphi}_o^T I_{o_{\hat{y}}} \dot{\varphi}_o$$

Considerando agora as seguintes definições

$$M_O(\beta_o) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

onde

$$\begin{aligned} m_{11} &= m_o \\ m_{12} = m_{21} &= 0 \\ m_{13} = m_{31} &= -l m_o \sin(\alpha_o) + m_o d \cos(\alpha_o + \beta_o) \\ m_{22} &= m_o \\ m_{23} = m_{32} &= l m_o \cos(\alpha_o) + m_o d \sin(\alpha_o + \beta_o) \\ m_{33} &= l^2 m_o + 2l \sin(\beta_o) d m_o + d^2 m_o + I_{o_z} \end{aligned}$$

$$V(\beta_o) = \begin{bmatrix} \cos(\alpha_o + \beta_o) m_o d \\ \sin(\alpha_o + \beta_o) m_o d \\ l \sin(\beta_o) m_o d + m_o d^2 + I_{o_z} \end{bmatrix}$$

$$I_{\beta\hat{z}} = (d^2m_o + I_{o\hat{z}})$$

tem-se que

$$\begin{aligned} T_o &= \frac{1}{2} \left({}^0\dot{\xi}_c^T c R_0^T M_O(\beta_o) c R_0 {}^0\dot{\xi}_c + {}^0\dot{\xi}_c^T c R_0^T V(\beta_o) \dot{\beta}_o + {}^0\dot{\xi}_c^T c R_0^T V(\beta_o) \dot{\beta}_o + \dot{\beta}_o^T I_{\beta\hat{z}} \dot{\beta}_o \right) \\ &+ \frac{1}{2} \dot{\varphi}_o^T I_{o\hat{y}} \dot{\varphi}_o \end{aligned}$$

E assim, pode-se escrever para a energia cinética de cada uma das rodas orientáveis não centradas:

$$T_{o_i} = \frac{1}{2} {}^0\dot{\xi}_c^T c R_0^T M_{O_i}(\beta_{o_i}) c R_0 {}^0\dot{\xi}_c + {}^0\dot{\xi}_c^T c R_0^T V(\beta_{o_i}) \dot{\beta}_{o_i} + \frac{1}{2} \dot{\beta}_{o_i}^T I_{\beta\hat{z}_i} \dot{\beta}_{o_i} + \frac{1}{2} \dot{\varphi}_{o_i}^T I_{o_i\hat{y}_i} \dot{\varphi}_{o_i}$$

B.4 Energia Cinética Total

Para o caso genérico de uma base móvel que contenha um número qualquer de rodas fixas e rodas orientáveis não centradas, a energia cinética total será calculada pela soma das parcelas da energia cinética do chassi, de cada uma das rodas fixas e de cada uma das rodas orientáveis não centradas, assim, pode-se escrever

$$\begin{aligned} T &= T_c + \sum_{i=1}^{nf} T_{f_i} + \sum_{i=1}^{no} T_{o_i} \\ &= \frac{1}{2} {}^0\dot{\xi}_c^T c R_0^T M_c c R_0 {}^0\dot{\xi}_c + \frac{1}{2} {}^0\dot{\xi}_c^T c R_0^T \sum_{i=1}^{nf} M_{f_i} c R_0 {}^0\dot{\xi}_c + \frac{1}{2} \sum_{i=1}^{nf} \dot{\varphi}_{f_i}^T I_{f_i\hat{y}_i} \dot{\varphi}_{f_i} \\ &+ \frac{1}{2} {}^0\dot{\xi}_c^T c R_0^T \sum_{i=1}^{no} M_{O_i}(\beta_{o_i}) c R_0 {}^0\dot{\xi}_c + {}^0\dot{\xi}_c^T c R_0^T \sum_{i=1}^{no} V(\beta_{o_i}) \dot{\beta}_{o_i} + \frac{1}{2} \sum_{i=1}^{no} \dot{\beta}_{o_i}^T I_{\beta\hat{z}_i} \dot{\beta}_{o_i} \\ &+ \frac{1}{2} \sum_{i=1}^{no} \dot{\varphi}_{o_i}^T I_{o_i\hat{y}_i} \dot{\varphi}_{o_i} \end{aligned} \quad (260)$$

Colocando agora o termo $\frac{1}{2} {}^0\dot{\xi}_c^T c R_0^T$ em evidência e definindo-se

$$M(\beta_o) \triangleq M_c + \sum_{i=1}^{nf} M_{f_i} + \sum_{i=1}^{no} M_{O_i}(\beta_{o_i}) \quad (261)$$

$$\dot{\beta}_o = \begin{bmatrix} \dot{\beta}_{o_1} \\ \vdots \\ \dot{\beta}_{o_{no}} \end{bmatrix} \quad (262)$$

$$V(\beta_o) = [V_1(\beta_{o_1}) \quad \cdots \quad V_n(\beta_{o_{no}})] \quad (263)$$

$$\dot{\varphi} = \begin{bmatrix} \dot{\varphi}_{f_1} \\ \vdots \\ \dot{\varphi}_{f_{nf}} \\ \dot{\varphi}_{o_1} \\ \vdots \\ \dot{\varphi}_{o_{no}} \end{bmatrix} \quad (264)$$

Com 264 considerando apenas rodas fixas e orientáveis não centradas.

$$I_{\varphi} = \text{diag} \left(I_{f_{\hat{y}_i}}, I_{o_{\hat{y}_i}} \right) \quad (265)$$

$$I_{\beta} = \text{diag} (I_{\beta \hat{z}_i}) \quad (266)$$

e observando-se que

$$\sum_{i=1}^{no} V(\beta_{o_i}) \dot{\beta}_{o_i} = [V_1(\beta_{o_1}) \quad \cdots \quad V_n(\beta_{o_{no}})] \begin{bmatrix} \dot{\beta}_{o_1} \\ \vdots \\ \dot{\beta}_{o_{no}} \end{bmatrix} = V(\beta_o) \dot{\beta}_o$$

pode-se reescrever (260) na forma

$$T = \frac{1}{2} {}^0 \xi_c^T {}^c R_0^T \left[M(\beta_o) {}^c R_0 {}^0 \dot{\xi}_c + 2V(\beta_o) \dot{\beta}_o \right] + \frac{1}{2} \dot{\beta}_o^T I_{\beta} \dot{\beta}_o + \frac{1}{2} \dot{\varphi}^T I_{\varphi} \dot{\varphi} \quad (267)$$

APÊNDICE C DERIVADAS DA ENERGIA CINÉTICA DA PLATAFORMA MÓVEL

Este desenvolvimento demonstra a obtenção das derivadas da energia cinética T dadas por (61), (62) e (63) utilizando as regras de derivação do apêndice A.

C.1 Derivadas da energia cinética em relação a ξ e $\dot{\xi}$

$$\frac{\partial T}{\partial \dot{\xi}} = \frac{\partial}{\partial \dot{\xi}} \left(\frac{1}{2} {}^0 \dot{\xi}^T {}^c R_0^T \left[M(\beta_o) {}^c R_0 {}^0 \dot{\xi} + 2V(\beta_o) \dot{\beta}_o \right] + \frac{1}{2} \dot{\beta}_o^T I_\beta \dot{\beta}_o + \frac{1}{2} \dot{\varphi}^T I_\varphi \dot{\varphi} \right)$$

e sendo:

$$\begin{aligned} \frac{\partial}{\partial \dot{\xi}} \left(\frac{1}{2} \dot{\beta}_o^T I_\beta \dot{\beta}_o \right) &= 0 \\ \frac{\partial}{\partial \dot{\xi}} \left(\frac{1}{2} \dot{\varphi}^T I_\varphi \dot{\varphi} \right) &= 0 \end{aligned}$$

pode-se escrever:

$$\frac{\partial T}{\partial \dot{\xi}} = \frac{\partial}{\partial \dot{\xi}} \left(\frac{1}{2} {}^0 \dot{\xi}^T {}^c R_0^T M(\beta_o) {}^c R_0 {}^0 \dot{\xi} + {}^0 \dot{\xi}^T {}^c R_0^T V(\beta_o) \dot{\beta}_o \right)$$

Uma vez que o termo ${}^c R_0^T M(\beta_o) {}^c R_0$ é simétrico e utilizando-se a regra de derivação 8 do apêndice A obtém-se:

$$\frac{\partial T}{\partial \dot{\xi}} = {}^0 \dot{\xi}^T {}^c R_0^T M(\beta_o) {}^c R_0 + \dot{\beta}_o^T V(\beta_o)^T {}^c R_0 \quad (268)$$

E derivando de (268) com relação ao tempo resulta em:

$$\begin{aligned}
\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) &= {}^0 \ddot{\xi}_c^T {}^c R_0^T M(\beta_o) {}^c R_0 + {}^0 \dot{\xi}_c^T \frac{d}{dt} ({}^c R_0^T) M(\beta_o) {}^c R_0 \\
&+ {}^0 \dot{\xi}_c^T {}^c R_0^T \frac{d}{dt} (M(\beta_o)) {}^c R_0 + {}^0 \dot{\xi}_c^T {}^c R_0^T M(\beta_o) \frac{d}{dt} ({}^c R_0) \\
&+ \ddot{\beta}_o^T V^T(\beta_o) {}^c R_0 + \dot{\beta}_o^T \frac{d}{dt} V^T(\beta_o) {}^c R_0 \\
&+ \dot{\beta}_o^T V^T(\beta_o) \frac{d}{dt} ({}^c R_0)
\end{aligned} \tag{269}$$

com as derivadas $\frac{d}{dt} ({}^c R_0)$, $\frac{d}{dt} ({}^c R_0^T)$, $\frac{d}{dt} (M(\beta_o))$ e $\frac{d}{dt} (V(\beta_o))$ dadas conforme (270), (271), (272) e (273) respectivamente.

Para o termo $\frac{d}{dt} ({}^c R_0)$ a operação de derivação em relação ao tempo pode ser substituída por uma multiplicação de matrizes (FU; GONZALES; LEE, 1987) e seguindo-se o desenvolvimento proposto obtém-se:

$$\begin{aligned}
\frac{d}{dt} ({}^c R_0) &= \frac{\partial}{\partial \theta_c} ({}^c R_0) \dot{\theta}_c = \frac{\partial}{\partial \theta_c} \left(\begin{bmatrix} \cos(\theta_c) & \sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \dot{\theta}_c = \\
\begin{bmatrix} -\sin(\theta_c) & \cos(\theta_c) & 0 \\ -\cos(\theta_c) & -\sin(\theta_c) & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}_c &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta_c) & \sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\theta}_c
\end{aligned}$$

E definindo-se o operador $Q = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, pode-se escrever:

$$\frac{d}{dt} ({}^c R_0) = Q {}^c R_0 \dot{\theta}_c \tag{270}$$

$$\frac{d}{dt} ({}^c R_0^T) = \left(\frac{d}{dt} ({}^c R_0) \right)^T = {}^c R_0^T Q^T \dot{\theta}_c \tag{271}$$

Para a derivada do termo $\frac{d}{dt} (M(\beta_o))$, observa-se a definição da matriz $M(\beta_o)$:

$$M(\beta_o) \triangleq M_C + \sum_{i=1}^{nf} M_{F_i} + \sum_{i=1}^{no} M_{O_i}(\beta_{o_i})$$

assim, tem-se:

$$\frac{d}{dt} (M(\beta_o)) = \frac{d}{dt} M_C + \sum_{i=1}^{nf} \frac{d}{dt} M_{F_i} + \sum_{i=1}^{no} \frac{d}{dt} M_{O_i}(\beta_{o_i})$$

sendo que as derivadas dos termos relativos às matrizes M_C e M_F resultam em zero uma vez que estas matrizes possuem apenas termos constantes. Então:

$$\frac{d}{dt} (M(\beta_o)) = \sum_{i=1}^{no} \frac{d}{dt} M_{O_i}(\beta_{o_i}) = \sum_{i=1}^{no} \left(\frac{\partial M_{O_i}(\beta_{o_i})}{\partial \beta_{o_i}} \dot{\beta}_{o_i} \right) \tag{272}$$

Como a matriz $M(\beta_o)$ é simétrica tem-se que:

$$M^T(\beta_o) = M(\beta_o)$$

e:

$$\left(\frac{d}{dt} (M(\beta_o)) \right)^T = \frac{d}{dt} (M^T(\beta_o)) = \frac{d}{dt} (M(\beta_o))$$

Assim, pode-se reescrever:

$$\begin{aligned} \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) \right)^T &= {}^c R_0^T M(\beta_o) {}^c R_0^0 \ddot{\xi}_c + {}^c R_0^T M(\beta_o) Q^c R_0 \dot{\theta}_c {}^0 \dot{\xi}_c \\ &+ {}^c R_0^T \frac{d}{dt} (M(\beta_o)) {}^c R_0^0 \dot{\xi}_c + {}^c R_0^T Q^T \dot{\theta}_c M(\beta_o) {}^c R_0^0 \dot{\xi}_c \\ &+ {}^c R_0^T V(\beta_o) \ddot{\beta}_o + {}^c R_0^T \frac{d}{dt} V(\beta_o) \dot{\beta}_o \\ &+ {}^c R_0^T Q^T \dot{\theta}_c V(\beta_o) \dot{\beta}_o \end{aligned}$$

Para a derivada do termo $\frac{d}{dt} (V^T(\beta_o))$, conforme (263) pode-se escrever:

$$\begin{aligned} \frac{d}{dt} V(\beta_o) &= \left[\frac{d}{dt} V_1(\beta_{o1}) \quad \cdots \quad \frac{d}{dt} V_n(\beta_{ono}) \right] \\ &= \left[\frac{\partial(V_1(\beta_{o1}))}{\partial \beta_{o1}} \dot{\beta}_{o1} \quad \cdots \quad \frac{\partial(V_n(\beta_{ono}))}{\partial \beta_{ono}} \dot{\beta}_{ono} \right] \end{aligned} \quad (273)$$

Derivada do termo $\frac{\partial T}{\partial \xi}$

$$\frac{\partial T}{\partial \xi} = \frac{\partial}{\partial \xi} \left(\frac{1}{2} {}^0 \dot{\xi}_c^T {}^c R_0^T \left[M(\beta_o) {}^c R_0^0 \dot{\xi}_c + 2V(\beta_o) \dot{\beta}_o \right] + \frac{1}{2} \dot{\beta}_o^T I_\beta \dot{\beta}_o + \frac{1}{2} \dot{\varphi}^T I_\varphi \dot{\varphi} \right)$$

e observando-se que:

$$\begin{aligned} \frac{\partial}{\partial \xi} \left(\frac{1}{2} \dot{\beta}_o^T I_\beta \dot{\beta}_o \right) &= 0 \\ \frac{\partial}{\partial \xi} \left(\frac{1}{2} \dot{\varphi}^T I_\varphi \dot{\varphi} \right) &= 0 \end{aligned}$$

pode-se escrever:

$$\frac{\partial T}{\partial \xi} = \frac{\partial}{\partial \xi} \left(\frac{1}{2} {}^0 \dot{\xi}_c^T {}^c R_0^T M(\beta_o) {}^c R_0^0 \dot{\xi}_c + {}^0 \dot{\xi}_c^T {}^c R_0^T V(\beta_o) \dot{\beta}_o \right)$$

Definindo-se $f(\xi) = {}^c R_0^0 \dot{\xi}_c$ e $f_V(\beta_o) = V(\beta_o) \dot{\beta}_o$ tem-se:

$$\frac{\partial T}{\partial \xi} = \frac{\partial}{\partial \xi} \left(\frac{1}{2} f(\xi)^T M(\beta_o) f(\xi) + f(\xi)^T f_V(\beta_o) \right)$$

com:

$$\frac{\partial}{\partial \xi} \left(\frac{1}{2} f(\xi)^T M(\beta_o) f(\xi) \right) = 2 \frac{1}{2} f(\xi)^T M(\beta_o) \frac{\partial f(\xi)}{\partial \xi}$$

e:

$$\frac{\partial}{\partial \xi} \left(f(\xi)^T f_V(\beta_o) \right) = f(\xi)^T \frac{\partial f_V(\beta_o)}{\partial \xi} + f_V(\beta_o)^T \frac{\partial f(\xi)}{\partial \xi}$$

assim tem-se que:

$$\begin{aligned} \frac{\partial T}{\partial \xi} &= f(\xi)^T M(\beta_o) \frac{\partial f(\xi)}{\partial \xi} + f_V(\beta_o)^T \frac{\partial f(\xi)}{\partial \xi} \\ &= {}^0 \dot{\xi}_c^T {}^c R_0^T M(\beta_o) \frac{\partial ({}^c R_0^0 \dot{\xi}_c)}{\partial \xi} + \beta_o^T V^T(\beta_o) \frac{\partial ({}^c R_0^0 \dot{\xi}_c)}{\partial \xi} \end{aligned} \quad (274)$$

Observando-se que:

$${}^c R_0^0 \dot{\xi}_c = \begin{bmatrix} \cos(\theta_c) & +\sin(\theta_c) & 0 \\ -\sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_c) \dot{x}_c + \sin(\theta_c) \dot{y}_c \\ -\sin(\theta_c) \dot{x}_c + \cos(\theta_c) \dot{y}_c \\ \dot{\theta}_c \end{bmatrix}$$

obtem-se para a derivada $\frac{\partial ({}^c R_0^0 \dot{\xi}_c)}{\partial \xi}$:

$$\begin{aligned} \frac{\partial ({}^c R_0^0 \dot{\xi}_c)}{\partial \xi} &= \begin{bmatrix} \frac{\partial(\cos(\theta_c)\dot{x}_c + \sin(\theta_c)\dot{y}_c)}{\partial x_c} & \frac{\partial(\cos(\theta_c)\dot{x}_c + \sin(\theta_c)\dot{y}_c)}{\partial y_c} & \frac{\partial(\cos(\theta_c)\dot{x}_c + \sin(\theta_c)\dot{y}_c)}{\partial \theta_c} \\ \frac{\partial(-\sin(\theta_c)\dot{x}_c + \cos(\theta_c)\dot{y}_c)}{\partial x_c} & \frac{\partial(-\sin(\theta_c)\dot{x}_c + \cos(\theta_c)\dot{y}_c)}{\partial y_c} & \frac{\partial(-\sin(\theta_c)\dot{x}_c + \cos(\theta_c)\dot{y}_c)}{\partial \theta_c} \\ \frac{\partial \dot{\theta}_c}{\partial x_c} & \frac{\partial \dot{\theta}_c}{\partial y_c} & \frac{\partial \dot{\theta}_c}{\partial \theta_c} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & -\sin(\theta_c) \dot{x}_c + \cos(\theta_c) \dot{y}_c \\ 0 & 0 & -\cos(\theta_c) \dot{x}_c - \sin(\theta_c) \dot{y}_c \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

E definindo-se:

$$K_v(\xi) = \begin{bmatrix} 0 & 0 & -\sin(\theta_c) \dot{x}_c + \cos(\theta_c) \dot{y}_c \\ 0 & 0 & -\cos(\theta_c) \dot{x}_c - \sin(\theta_c) \dot{y}_c \\ 0 & 0 & 0 \end{bmatrix}$$

pode-se reescrever (274) como:

$$\frac{\partial T}{\partial \xi} = {}^0 \dot{\xi}_c^T {}^c R_0^T M(\beta_o) K_v(\xi) + \beta_o^T V^T(\beta_o) K_v(\xi) \quad (275)$$

C.2 Derivadas da energia cinética em relação a β_o e $\dot{\beta}_o$

Derivada do termo $\frac{\partial T}{\partial \beta_o}$

$$\begin{aligned}\frac{\partial T}{\partial \dot{\beta}_o} &= \frac{\partial}{\partial \dot{\beta}_o} \left({}^0\dot{\xi}_c^T {}^c R_0^T V(\beta_o) \dot{\beta}_o + \frac{1}{2} \dot{\beta}_o^T I_\beta \dot{\beta}_o \right) \\ &= {}^0\dot{\xi}_c^T {}^c R_0^T V(\beta_o) + \dot{\beta}_o^T I_\beta\end{aligned}\quad (276)$$

Derivando-se (276) em relação ao tempo obtém-se:

$$\begin{aligned}\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_o} \right) &= {}^0\ddot{\xi}_c^T {}^c R_0^T V(\beta_o) + {}^0\dot{\xi}_c^T \frac{d}{dt} ({}^c R_0^T) V(\beta_o) + {}^0\dot{\xi}_c^T {}^c R_0^T \frac{d}{dt} (V(\beta_o)) \\ &\quad + \ddot{\beta}_o^T I_\beta\end{aligned}\quad (277)$$

$$\begin{aligned}\left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_o} \right) \right)^T &= V^T(\beta_o) {}^c R_0^0 \ddot{\xi}_c + V^T(\beta_o) \frac{d}{dt} ({}^c R_0) {}^0\dot{\xi}_c + \frac{d}{dt} (V^T(\beta_o)) {}^c R_0^0 \dot{\xi}_c \\ &\quad + I_\beta^T \ddot{\beta}_o \\ &= V^T(\beta_o) {}^c R_0^0 \ddot{\xi}_c + V^T(\beta_o) Q^c R_0 \dot{\theta}_c {}^0\dot{\xi}_c + \frac{d}{dt} (V^T(\beta_o)) {}^c R_0^0 \dot{\xi}_c \\ &\quad + I_\beta \ddot{\beta}_o\end{aligned}$$

Para derivar o termo $\frac{\partial T}{\partial \beta_o}$, define-se $f_M(\beta_o) = M(\beta_o) {}^c R_0^0 \dot{\xi}_c$ e assim:

$$\begin{aligned}\frac{\partial T}{\partial \beta_o} &= \frac{\partial}{\partial \beta_o} \left(\frac{1}{2} {}^0\dot{\xi}_c^T {}^c R_0^T M(\beta_o) {}^c R_0^0 \dot{\xi}_c + {}^0\dot{\xi}_c^T {}^c R_0^T V(\beta_o) \dot{\beta}_o \right) \\ &= \frac{\partial}{\partial \beta_o} \left(\frac{1}{2} {}^0\dot{\xi}_c^T {}^c R_0^T f_M(\beta_o) + {}^0\dot{\xi}_c^T {}^c R_0^T f_V(\beta_o) \right) \\ &= \frac{1}{2} {}^0\dot{\xi}_c^T {}^c R_0^T \frac{\partial f_M(\beta_o)}{\partial \beta_o} + {}^0\dot{\xi}_c^T {}^c R_0^T \frac{\partial f_V(\beta_o)}{\partial \beta_o}\end{aligned}\quad (278)$$

com $\frac{\partial f_M(\beta_o)}{\partial \beta_o}$ e $\frac{\partial f_V(\beta_o)}{\partial \beta_o}$ dadas respectivamente por (282) e (284)

$$\frac{\partial f_M(\beta_o)}{\partial \beta_o} = \frac{\partial \left(M(\beta_o) {}^c R_0^0 \dot{\xi}_c \right)}{\partial \beta_o}$$

e utilizando-se a definição de $M(\beta_o)$ pode-se escrever

$$\begin{aligned}\frac{\partial f_M(\beta_o)}{\partial \beta_o} &= \frac{\partial}{\partial \beta_o} \left(M_C {}^c R_0^0 \dot{\xi}_c \right) + \frac{\partial}{\partial \beta_o} \left(\sum_{i=1}^{nf} M_{F_i} {}^c R_0^0 \dot{\xi}_c \right) \\ &\quad + \frac{\partial}{\partial \beta_o} \left(\sum_{i=0}^{no} M_{O_i}(\beta_{o_i}) {}^c R_0^0 \dot{\xi}_c \right)\end{aligned}\quad (279)$$

Observando que M_C , M_{F_i} , cR_0 e ${}^0\dot{\xi}_c$ não dependem do termo β_o suas derivadas são nulas, e assim, pode-se reescrever (279) como

$$\frac{\partial f_M(\beta_o)}{\partial \beta_o} = \frac{\partial}{\partial \beta_o} \left(\sum_{i=0}^{no} M_{O_i}(\beta_{o_i}) {}^cR_0 {}^0\dot{\xi}_c \right) \quad (280)$$

E utilizando-se a regra de derivação 2 do apêndice A, pode-se reescrever a expressão (280) na forma

$$\frac{\partial f_M(\beta_o)}{\partial \beta_o} = \left[\frac{\partial}{\partial \beta_{o_1}} \left(\sum_{i=0}^{no} M_{O_i}(\beta_{o_i}) {}^cR_0 {}^0\dot{\xi}_c \right) \cdots \frac{\partial}{\partial \beta_{o_n}} \left(\sum_{i=0}^{no} M_{O_i}(\beta_{o_i}) {}^cR_0 {}^0\dot{\xi}_c \right) \right] \quad (281)$$

Uma vez que apenas um dos termos de cada somatório não resultará em zero devido à derivação em relação a apenas um ângulo β_{o_i} para cada termo do vetor, tem-se que

$$\frac{\partial f_M(\beta_o)}{\partial \beta_o} = \left[\frac{\partial}{\partial \beta_{o_1}} M_{O_1}(\beta_{o_1}) {}^cR_0 {}^0\dot{\xi}_c \cdots \frac{\partial}{\partial \beta_{o_n}} M_{O_n}(\beta_{o_n}) {}^cR_0 {}^0\dot{\xi}_c \right] \quad (282)$$

$$\begin{aligned} \frac{\partial f_V(\beta_o)}{\partial \beta_o} &= \frac{\partial (V(\beta_o) \dot{\beta}_o)}{\partial \beta_o} = \frac{\partial}{\partial \beta_o} \left(\sum_{i=1}^{no} V(\beta_{o_i}) \dot{\beta}_{o_i} \right) \\ &= \left[\frac{\partial}{\partial \beta_{o_1}} \left(\sum_{i=1}^{no} V(\beta_{o_i}) \dot{\beta}_{o_i} \right) \cdots \frac{\partial}{\partial \beta_{o_n}} \left(\sum_{i=1}^{no} V(\beta_{o_i}) \dot{\beta}_{o_i} \right) \right] \end{aligned} \quad (283)$$

E também para esta caso, apenas um dos termos de cada somatório não resultará em zero devido à derivação em relação a apenas um ângulo β_{o_i} e assim pode-se reescrever (283) como:

$$\frac{\partial f_V(\beta_o)}{\partial \beta_o} = \left[\frac{\partial}{\partial \beta_{o_1}} V(\beta_{o_1}) \dot{\beta}_{o_1} \cdots \frac{\partial}{\partial \beta_{o_n}} V(\beta_{o_n}) \dot{\beta}_{o_n} \right] \quad (284)$$

E definindo-se as notações:

$$G_\beta(f_M) = \frac{\partial f_M(\beta_o)}{\partial \beta_o}$$

$$G_\beta(f_V) = \frac{\partial f_V(\beta_o)}{\partial \beta_o}$$

pode-se reescrever (278) como:

$$\frac{\partial T}{\partial \beta_o} = \frac{1}{2} {}^0\dot{\xi}_c^T {}^cR_0^T G_\beta(f_M) + {}^0\dot{\xi}_c^T {}^cR_0^T G_\beta(f_V) \quad (285)$$

C.3 Derivadas da energia cinética em relação a φ e $\dot{\varphi}$

Derivada do termo $\frac{\partial T}{\partial \dot{\varphi}}$

$$\frac{\partial T}{\partial \dot{\varphi}} = \frac{\partial}{\partial \dot{\varphi}} \left(\frac{1}{2} \dot{\varphi}^T I_\varphi \dot{\varphi} \right) = \dot{\varphi}^T I_\varphi \quad (286)$$

Derivando-se (286) em relação ao tempo obtém-se:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\varphi}} \right) = \ddot{\varphi}^T I_\varphi \quad (287)$$

E para a derivada de T com relação a φ tem-se:

$$\frac{\partial T}{\partial \varphi} = 0 \quad (288)$$

C.4 Resumo das Expressões

Para o desenvolvimento do modelo dinâmico da base móvel utilizam-se as expressões (269), (275), (277), (285), (287) e (288):

Expressão (269):

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) &= {}^0 \ddot{\xi}_c^T {}^c R_0^T M(\beta_o) {}^c R_0 + {}^0 \dot{\xi}_c^T \frac{d}{dt} ({}^c R_0^T) M(\beta_o) {}^c R_0 \\ &+ {}^0 \dot{\xi}_c^T {}^c R_0^T \frac{d}{dt} (M(\beta_o)) {}^c R_0 + {}^0 \dot{\xi}_c^T {}^c R_0^T M(\beta_o) \frac{d}{dt} ({}^c R_0) \\ &+ \ddot{\beta}_o^T V^T(\beta_o) {}^c R_0 + \dot{\beta}_o^T \frac{d}{dt} V^T(\beta_o) {}^c R_0 \\ &+ \dot{\beta}_o^T V^T(\beta_o) \frac{d}{dt} ({}^c R_0) \end{aligned}$$

Expressão (275):

$$\frac{\partial T}{\partial \xi} = {}^0 \dot{\xi}_c^T {}^c R_0^T M(\beta_o) K_v(\xi) + V^T(\beta_o) K_v(\xi) \dot{\beta}_o$$

Expressão (277):

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_o} \right) &= {}^0 \ddot{\xi}_c^T {}^c R_0^T V(\beta_o) + {}^0 \dot{\xi}_c^T \frac{d}{dt} ({}^c R_0^T) V(\beta_o) + {}^0 \dot{\xi}_c^T {}^c R_0^T \frac{d}{dt} (V(\beta_o)) \\ &+ \ddot{\beta}_o^T I_\beta \end{aligned}$$

Expressão (285):

$$\frac{\partial T}{\partial \beta_o} = \frac{1}{2} {}^0 \dot{\xi}_c^T {}^c R_0^T G_\beta(f_M) + {}^0 \dot{\xi}_c^T {}^c R_0^T G_\beta(f_V)$$

Expressão (287):

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\varphi}} \right) = \ddot{\varphi}^T I_{\varphi}$$

Expressão (288):

$$\frac{\partial T}{\partial \varphi} = 0$$

APÊNDICE D DERIVADAS DE SEGUNDA ORDEM DAS VARIÁVEIS ${}^0\xi_c$, β_o E φ

No apêndice C, foram obtidas as derivadas de segunda ordem ${}^0\ddot{\xi}_c$, $\ddot{\beta}_o$ e $\ddot{\varphi}$. E conforme (48), (55) e (56), pode-se escrever:

$$\begin{aligned} {}^0\ddot{\xi}_c &= \frac{d}{dt} {}^0\dot{\xi}_c = \frac{d}{dt} ({}^0R_c \Sigma(\beta_c) \eta) \\ \ddot{\beta}_o &= \frac{d}{dt} \dot{\beta}_o = \frac{d}{dt} (D(\beta_o) \Sigma(\beta_c) \eta) \\ \ddot{\varphi} &= \frac{d}{dt} \dot{\varphi} = \frac{d}{dt} (E(\beta_c, \beta_o) \Sigma(\beta_c) \eta) \end{aligned}$$

Para seguir com o desenvolvimento das derivadas de segunda ordem, algumas definições referentes as derivadas de primeira ordem são apresentadas. A derivada de primeira ordem do termo $\Sigma(\beta_c)$ é definida como uma função da variável β_c e de sua derivada de primeira ordem $\dot{\beta}_c = \zeta$

$$\frac{d(\Sigma(\beta_c))}{dt} = f_s(\beta_c, \dot{\beta}_c) = f_s(\beta_c, \zeta) \quad (289)$$

Para a derivada do termo $D(\beta_o)$, também define-se uma função dependente das variáveis β_c e $\dot{\beta}_c$, sendo assim, tem-se que:

$$\begin{aligned} \frac{dD(\beta_o)}{dt} &= \frac{d(-C_{2o}^{-1} C_{1o}(\beta_o))}{dt} = -C_{2o}^{-1} \frac{d(C_{1o}(\beta_o))}{dt} \\ &= -C_{2o}^{-1} \frac{d}{dt} \begin{bmatrix} C_{1o_1}(\beta_{o_1}) \\ \vdots \\ C_{1o_{no}}(\beta_{o_{no}}) \end{bmatrix} = -C_{2o}^{-1} \begin{bmatrix} \frac{\partial C_{1o_1}(\beta_{o_1})}{\partial \beta_{o_1}} \dot{\beta}_{o_1} \\ \vdots \\ \frac{\partial C_{1o_{no}}(\beta_{o_{no}})}{\partial \beta_{o_{no}}} \dot{\beta}_{o_{no}} \end{bmatrix} \end{aligned}$$

desta forma pode-se definir a função $f_d(\beta_c, \zeta)$

$$f_d(\beta_c, \zeta) = f_d(\beta_c, \dot{\beta}_c) = -C_{2o}^{-1} \begin{bmatrix} \frac{\partial C_{1o_1}(\beta_{o_1})}{\partial \beta_{o_1}} \dot{\beta}_{o_1} \\ \vdots \\ \frac{\partial C_{1o_{no}}(\beta_{o_{no}})}{\partial \beta_{o_{no}}} \dot{\beta}_{o_{no}} \end{bmatrix} \quad (290)$$

Para a derivada do termo $(E(\beta_c, \beta_o))$, deve-se observar que conforme (38), cada uma de suas linhas depende apenas de β_c ou de β_o e assim pode-se escrever

$$\begin{aligned} \frac{d}{dt}(E(\beta_c, \beta_o)) &= \frac{d}{dt}(-J_2^{-1}J_1(\beta_c, \beta_o)) = -J_2^{-1}\frac{d}{dt}(J_1(\beta_c, \beta_o)) = \\ -J_2^{-1}\frac{d}{dt}\left(\begin{bmatrix} J_{1f} \\ J_{1c}(\beta_c) \\ J_{1o}(\beta_o) \\ J_{1u} \end{bmatrix}\right) &= -J_2^{-1}\frac{d}{dt}\begin{bmatrix} J_{1f_1} \\ \vdots \\ J_{1f_{n_f}} \\ J_{1c_1}(\beta_{c_1}) \\ \vdots \\ J_{1c_{n_c}}(\beta_{c_{n_c}}) \\ J_{1o_1}(\beta_{o_1}) \\ \vdots \\ J_{1o_{n_o}}(\beta_{o_{n_o}}) \\ J_{1u_1} \\ \vdots \\ J_{1u_{n_u}} \end{bmatrix} = -J_2^{-1}\begin{bmatrix} 0_{n_f \times 3} \\ \frac{\partial J_{1c_1}(\beta_{c_1})}{\partial(\beta_{c_1})}\dot{\beta}_{c_1} \\ \vdots \\ \frac{\partial J_{1c_{n_c}}(\beta_{c_{n_c}})}{\partial(\beta_{c_{n_c}})}\dot{\beta}_{c_{n_c}} \\ \frac{\partial J_{1o_1}(\beta_{o_1})}{\partial(\beta_{o_1})}\dot{\beta}_{o_1} \\ \vdots \\ \frac{\partial J_{1o_{n_o}}(\beta_{o_{n_o}})}{\partial(\beta_{o_{n_o}})}\dot{\beta}_{o_{n_o}} \\ 0_{n_u \times 3} \end{bmatrix} \end{aligned}$$

e define-se então a função $f_e(\beta_c, \zeta, \beta_o, \dot{\beta}_o)$

$$f_e(\beta_c, \zeta, \beta_o, \dot{\beta}_o) = -J_2^{-1}\begin{bmatrix} 0_{n_f \times 3} \\ \frac{\partial J_{1c_1}(\beta_{c_1})}{\partial(\beta_{c_1})}\dot{\beta}_{c_1} \\ \vdots \\ \frac{\partial J_{1c_{n_c}}(\beta_{c_{n_c}})}{\partial(\beta_{c_{n_c}})}\dot{\beta}_{c_{n_c}} \\ \frac{\partial J_{1o_1}(\beta_{o_1})}{\partial(\beta_{o_1})}\dot{\beta}_{o_1} \\ \vdots \\ \frac{\partial J_{1o_{n_o}}(\beta_{o_{n_o}})}{\partial(\beta_{o_{n_o}})}\dot{\beta}_{o_{n_o}} \\ 0_{n_u \times 3} \end{bmatrix} \quad (291)$$

Agora, utilizando (289), (290) e (291) tem-se para ${}^0\ddot{\xi}_c$, $\ddot{\beta}_o$ e $\ddot{\varphi}$

$${}^0\ddot{\xi}_c = {}^cR_0^T Q^T \dot{\theta}_c \Sigma(\beta_c) \eta + {}^cR_0^T f_s(\beta_c, \zeta) \eta + {}^cR_0^T \Sigma(\beta_c) \dot{\eta} \quad (292)$$

$$\ddot{\beta}_o = f_d(\beta_c, \zeta) \Sigma(\beta_c) \eta + D(\beta_o) f_s(\beta_c, \zeta) \eta + D(\beta_o) \Sigma(\beta_c) \dot{\eta} \quad (293)$$

$$\ddot{\varphi} = f_e(\beta_c, \zeta, \beta_o, \dot{\beta}_o) \Sigma(\beta_c) \eta + E(\beta_c, \beta_o) f_s(\beta_c, \zeta) \eta + E(\beta_c, \beta_o) \Sigma(\beta_c) \dot{\eta} \quad (294)$$

APÊNDICE E DESENHOS DO ROBÔ MÓVEL TWIL

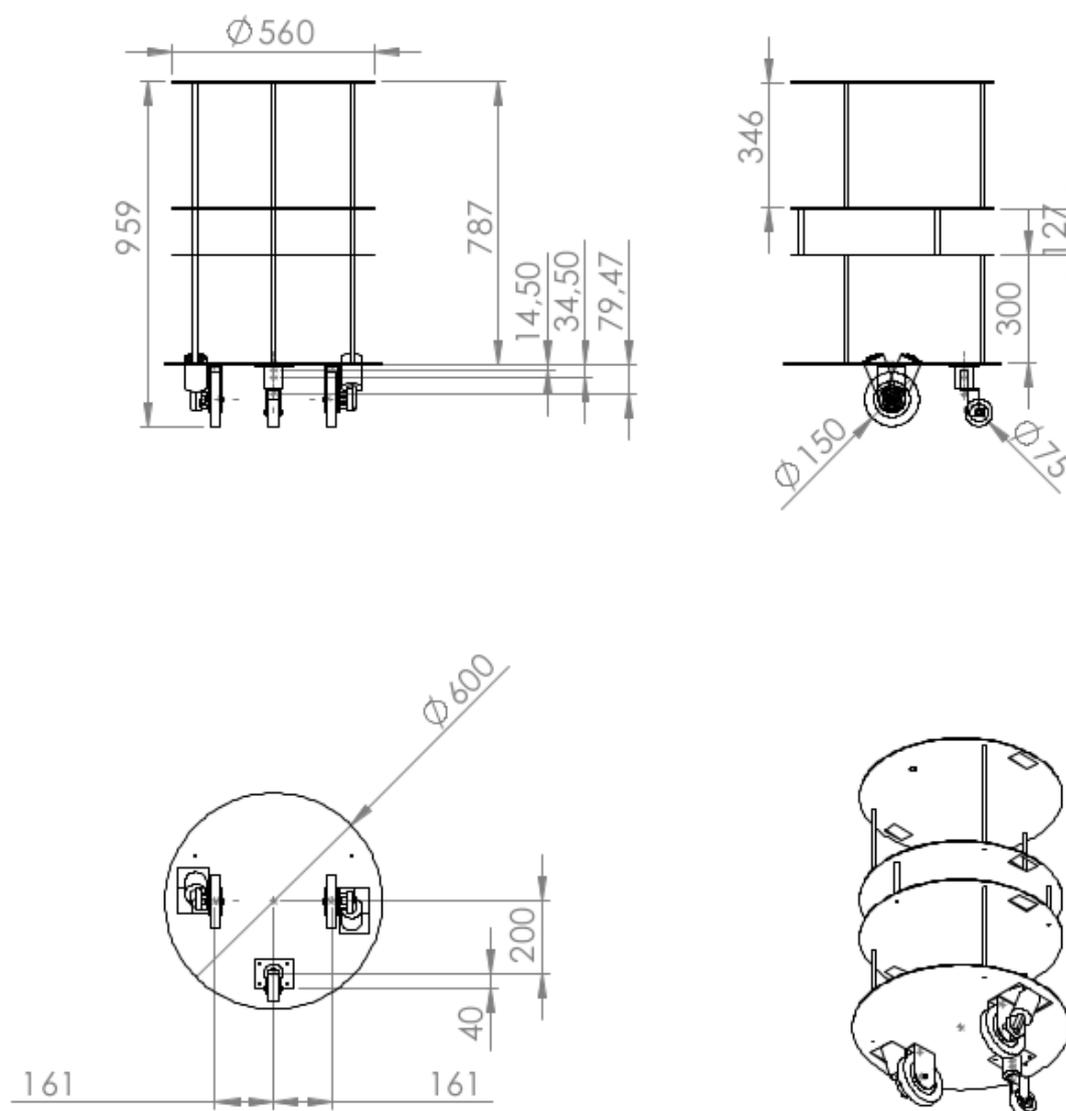


Figura 76: Estrutura do Robô Móvel TWIL.

APÊNDICE F MATRIZ DE ROTAÇÃO QUE RELACIONA DOIS PONTOS

Seja P_n um ponto obtido a partir da rotação de um ponto P em torno de um eixo arbitrário. Deseja-se determinar R_n tal que

$$P_n = R_n P \quad (295)$$

Inicialmente, nota-se que R_n não altera o módulo dos vetores que representam os pontos, portanto é conveniente continuar o desenvolvimento com vetores unitários:

$$P_1 = \frac{P}{|P|} \quad (296)$$

$$P_2 = \frac{P_n}{|P_n|} \quad (297)$$

A partir dos vetores unitários, o cosseno e o seno do ângulo de rotação entre eles pode ser obtido por:

$$\cos(\phi) = P_1 \cdot P_2 = P_1^T P_2 \quad (298)$$

$$\sin(\phi) = |P_1 \times P_2| \quad (299)$$

onde \cdot e \times representam respectivamente os produtos escalar e vetorial entre os dois vetores. Por outro lado, o eixo em torno do qual é feita a rotação pode ser obtido por:

$$r = \frac{1}{\sin(\phi)} (P_1 \times P_2) = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \quad (300)$$

A partir do seno e cosseno do ângulo de rotação e do eixo de rotação é possível obter a matriz de rotação, que tem a forma (FU; GONZALES; LEE, 1987):

$$R_n = \begin{bmatrix} r_x^2 \text{vers}(\phi) + \cos(\phi) & r_x r_y \text{vers}(\phi) - r_z \sin(\phi) & r_x r_z \text{vers}(\phi) + r_y \sin(\phi) \\ r_x r_y \text{vers}(\phi) + r_z \sin(\phi) & r_y^2 \text{vers}(\phi) + \cos(\phi) & r_y r_z \text{vers}(\phi) - r_x \sin(\phi) \\ r_x r_z \text{vers}(\phi) - r_y \sin(\phi) & r_y r_z \text{vers}(\phi) + r_x \sin(\phi) & r_z^2 \text{vers}(\phi) + \cos(\phi) \end{bmatrix} \quad (301)$$

onde $\text{vers}(\phi) = 1 - \cos(\phi)$