



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

DISSERTAÇÃO DE MESTRADO

**PROJETO E CONSTRUÇÃO DE UM ROBÔ MÓVEL AGV / ROV
NÃO-HOLONÔMICO COM HABILIDADE
PARA NAVEGAÇÃO AUTÔNOMA DO TIPO WALL-FOLLOWING**

Por

OTAVIO ANDRE CHASE

Belém, PA – Brasil

Dezembro de 2009

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PROJETO E CONSTRUÇÃO DE UM ROBÔ MÓVEL AGV / ROV
NÃO-HOLONÔMICO COM HABILIDADE
PARA NAVEGAÇÃO AUTÔNOMA DO TIPO WALL-FOLLOWING

Por

OTAVIO ANDRE CHASE

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

Orientador: Prof. Jorge Roberto Brito de Souza, Ph. D.

Área de concentração: Sistemas de Energia, com ênfase em Sistemas de Controle.

Belém, PA – Brasil

Dezembro de 2009

C4877p Chase, Otavio Andre

Projeto e construção de um robô móvel AGV/ROV não holonômico com habilidade para navegação autônoma do tipo wall-following / Otavio Andre Chase; orientador, Jorge Roberto Brito de Souza. -2009.

Dissertação (Mestrado) – Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-graduação em Engenharia Elétrica, Belém, 2009.

1. Robótica. 2. Robôs – projeto e construção. 3. Sistemas de controle inteligente. I. Orientador. II. Título.

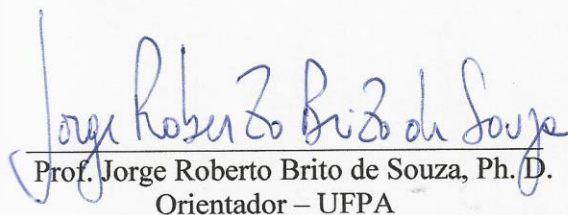
CDD 22. ed. 629.892

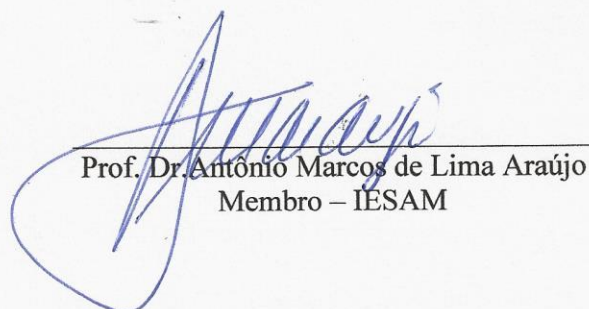
PROJETO E CONSTRUÇÃO DE UM ROBÔ MÓVEL AGV/ROV
NÃO-HOLONÔMICO COM HABILIDADE PARA
NAVEGAÇÃO AUTÔNOMA DO TIPO WALL-FOLLOWING

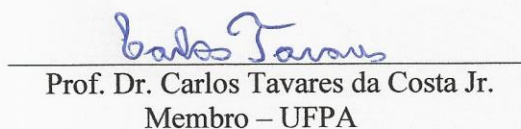
Dissertação de Mestrado submetida à avaliação da Banca Examinadora aprovada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará e julgada adequada para obtenção do título de Mestre em Engenharia Elétrica.

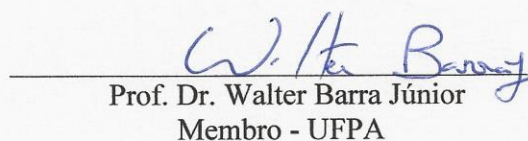
APROVADA EM 21 / 12 / 2009

BANCA EXAMINADORA:

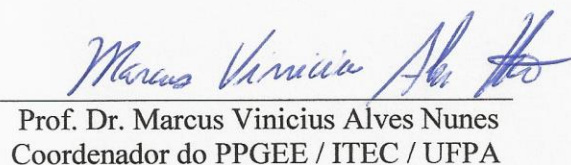

Prof. Jorge Roberto Brito de Souza, Ph. D.
Orientador – UFPA


Prof. Dr. Antônio Marcos de Lima Araújo
Membro – IESAM


Prof. Dr. Carlos Tavares da Costa Jr.
Membro – UFPA


Prof. Dr. Walter Barra Júnior
Membro - UFPA

Visto:


Prof. Dr. Marcus Vinicius Alves Nunes
Coordenador do PPGE / ITEC / UFPA

DEDICATÓRIA

Aos esforços de meu tio avô Nichollas Chase e de meu avô Octavio Chase, que sempre incentivaram a todos para a busca do conhecimento e suas aplicações para o bem da sociedade. E a Deus.

AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq, pelo apoio durante o projeto no mestrado. Um agradecimento especial a UFPA, pela estrutura, ambiente, pessoas e base de conhecimento científico e tecnológico.

Nenhum sonho vai para a frente sem dinheiro.

Wernher Von Braun

LISTA DE SIGLAS E ABREVIATURAS

Sigla	Significado	Tradução
ROV	Remotely Operated Vehicle	Veículo Operado Remotamente
AGV	Autonomous Guided Vehicle	Veículo Guiado Autonomamente
PID	Proportional Integral Derivative	Proporcional Integral Derivativo
PIC	Peripheral Interface Controller	Periférico Interface de Controle
WMR	Wheeled Mobile Robot	Robô Móvel com Rodas
GPS	Global Position System	Sistema Global de Posição
RSSF	Rede de Sensores Sem Fio	WSN - Wireless Sensor Network
PWM	Pulse Width Modulation	Modulação por Largura de Pulso
RTOS	Real Time Operational System	Sistema Operacional de Tempo Real
TTL	Transistor Transistor Logic	Lógica entre Transistores
PC	Personal Computer	Computador Pessoal
CPU	Central Processor Unit	Unidade Central de Processamento
UART	Universal Asynchronous Receiver/Transmitter.	Transmissor/Receptor Assíncrono Universal.
A/D	Analog to Digital Converter	Conversor Analógico/Digital
D/A	Digital to Analog Converter	Conversor Digital/Analógico
RAM	Random Access Memory	Memória de Acesso Randômico
ROM	Read Only Memory	Memória Somente de Leitura
IEEE	Institute of Electrical and Electronic Engineers	Instituto de Engenheiros Eletricistas e Eletrônicos
LOGBOT	Logistic Robot	Robô para Logística

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Níveis de controle de um robô móvel autônomo.	14
Figura 2.2 – Mapeamento de um ambiente no espaço de configurações.	18
Figura 2.3 – Níveis hierárquicos da arquitetura de navegação para robôs móveis	19
Figura 2.4 - Robô Móvel LOGBOT (UFPA-BRASIL, 2009).....	22
Figura 3.1 - Robô móvel LOGBOT subindo uma rampa.	23
Figura 3.2 – Ilustração sobre força e torque para o robô LOGBOT se locomover em um plano.	25
Figura 3.3 – Resposta do motor DC ao degrau de amplitude 4V.....	27
Figura 3.4 – Largura das Modulações PWM.....	31
Figura 3.4(a) - ROV Lunokhod I (URSS, 1970).....	33
Figura 3.4(b) - ROV Chico Mendes da Petrobras (Brasil, 2007).....	33
Figura 3.5 – Estrutura base em alumínio naval do robô LOGBOT.....	34
Figura 3.6 – Modelagem do robô LOGBOT feita no programa <i>Google Sketchup</i>	34
Figura 3.7 – Tração de movimento diferencial para um Robô Móvel.	35
Figura 3.8 – Projeto de um Robô Móvel pela Abordagem em Camadas	42
Figura 3.9 – Diagrama Modular do Robô LOGBOT	43
Figura 4.1 – Plano de referência global e local do robô móvel.....	46
Figura 4.2 – Movimento do tipo Tração Diferencial Não-Holonômico.....	50
Figura 4.3 – Estrutura Cinemática do robô móvel LOGBOT.	52
Figura 4.4 – Modelo Cinemático Completo do Robô Móvel LOGBOT.....	54
Figura 5.1 – Representação do robô como o ponto ‘q’ em seu espaço de configuração.....	57
Figura 5.2 – Sistema de Controle Tradicional para um Robô Móvel.....	58
Figura 5.6 – Corredor de testes do LOGBOT (PPGEE-UFPA, 2009).....	61
Figura 5.7 – Disposição do Robô no Corredor em 2 e 3 Dimensões.	62
Figura 5.8 – Configuração do robô em seu primeiro instante de movimento.	62
Figura 5.9 – Organização das variáveis e disposição do robô no Wall-Following.	63
Figura 5.9 – Geometria do Robô na problemática do Wall-Following.....	64
Figura 5.10 – Geometria Tratamento dos Erros de Distância e Orientação.....	65
Figura 5.11 – Movimento do robô com trajetória com comportamento em ganho crítico.	68
Figura 6.1 – Modelo Genérico de um Agente Inteligente de Software.....	69
Figura 6.2 – Agente Inteligente de Software Embarcado ao Robô Móvel LOGBOT.	75

Figura 6.3 – Configuração do Agente Reativo do LOGBOT.....	75
Figura 7.2 – Diagrama de um sistema embarcado monitorando um ambiente, através de um microcontrolador.	78
Figura 7.3 – Arquitetura Modular do Sistema Eletrônico do Robô.	81
Figura 7.4 – Módulo de Decisão.	82
Figura 7.5 – Módulo de Percepção.....	82
Figura 7.6 – Sensor de Ultra-Som LV Max - Sonar EZ-1.....	83
Figura 7.7 – Configuração do Vetor dos Sensores de Ultra-Som.	83
Figura 7.8 – Módulo de Ação.....	84
Figura 7.9 – Ligação da Ponte LMD18200T com o Motor DC.	84
Figura 7.10 – Arquitetura de um Sensor Inteligente.	85
Figura 7.11 – Camada criada por um RTOS entre o aplicativo e o Hardware.....	87
Figura 7.12 – Diagrama com as cinco classes de serviços básicos de um Kernel.	87
Figura 7.13 – Exemplo de Troca de Tarefas para um Agendador Preemptivo Baseado em Prioridade.....	89
Figura 7.14 – Posição e orientação de um robô móvel em seu espaço de configurações por odometria.	92
Figura 7.15 – Protocolo de comunicação do Robô Móvel LOGBOT.....	93
Figura 7.16 – Diagrama completo de comunicação entre o robô e a estação de controle (PC).	94
Figura 7.17 – Mapa do Setor Básico, que mostra onde o teste foi realizado (Campus da UFPA).....	95
Figura 7.18 – Orla da UFPA no Rio Guamá, e Robô na Base da Capela Ecumênica.....	95
Figura 8.2 (a) – Teste I: Erro e Ação de Controle.....	98
Figura 8.2 (b) – Teste I: Correção do erro de distância com a parede.	99
Figura 8.2 (c) – Teste I: Sinal de controle aplicado às velocidades (PMW) dos motores.....	99
Figura 8.3 – Correção da posição do Robô Abaixo da Referência.	100
Figura 8.4 (a) – Teste II: Erro e Ação de Controle.	100
Figura 8.4(b) – Teste II: Correção do erro de distância com a parede.	101
Figura 8.4(c) – Teste II: Sinal de controle aplicado às velocidades (PMW) dos motores. ...	101
Figura 8.5 – Correção da posição do Robô com a Problemática do Vão da Porta.....	102
Figura 8.6(a) – Teste III: Erro e Ação de Controle.	102
Figura 8.6(b) – Teste III: Correção do erro de distância com a parede no vão de uma porta.	103

Figura 8.6(c) – Teste III: Sinal de controle aplicado às velocidades (PMW) dos motores...	103
Figura 8.7 – Correção da posição do Robô com o contorno do final do corredor.	104
Figura 8.8(a) – Teste IV: Erro e Ação de Controle.....	104
Figura 8.8(b) – Teste IV: Correção do erro de distância da com a parede na mudança de parede.....	105
Figura 8.8(c) – Teste IV: Sinal de controle aplicado às velocidades (PMW) dos motores...	105

LISTA DE TABELAS

Tabela 3.1 – MKS Técnico ou MK*S.	29
Tabela 3.2 – SI – Sistema Internacional de Unidades.	29
Tabela 3.3 – Peso Total do robô LOGBOT em cargas útil.	30
Tabela 3.3 – Características de Baterias Diversas.	37
Tabela 3.4 – Características de Baterias Diversas.	38
Tabela 3.5 – Fontes, reguladores, cargas e autonomia para o módulo de alimentação.	41
Tabela 8.1 – Matriz de Aquisição de Dados do Controlador Wall-Following do LOGBOT...	96

RESUMO

Esta dissertação de mestrado apresenta o projeto e a construção de um robô móvel terrestre denominado LOGBOT, com tração de movimento do tipo diferencial – com duas rodas motoras e uma roda livre para manter a estabilidade de sua estrutura em relação à superfície. O controle do robô dispõe dos modos de telemetria e autônomo.

No modo de controle por telemetria (ROV), a comunicação do robô com a estação de controle é feita por radiofrequência a uma distância de até um quilometro em ambientes externos, e até cem metros em ambientes internos.

No modo de controle autônomo (AGV), o robô tem habilidade para navegar em ambientes internos e desconhecidos usando sempre a parede à sua esquerda como referência para a trajetória de seu movimento. A seqüência de movimentos para execução da trajetória é enviada para a estação de controle que realiza análises de desempenho do robô.

Para executar suas tarefas no modo autônomo, a programação do robô conta com um agente inteligente reativo, que detecta características do ambiente (obstáculos, final de paredes, etc.) e decide sobre qual atitude deve ser executada pelo robô, com objetivo de contornar os obstáculos e controlar a velocidade de suas rodas.

Os problemas de erro odométrico e suas correções com base no uso de informações sensoriais externas são devidamente tratados. Técnicas de controle hierárquico do robô como um todo e controle em malha fechada da velocidade das rodas do robô são usadas.

Os resultados mostraram que o robô móvel LOGBOT é capaz de navegar, com estabilidade e precisão, em ambientes internos no formato de um corredor (*wall following*).

Palavras-chave: Robótica Móvel, Controle de Robôs Não-Holonômicos, Navegação de Robôs, Plataformas Robóticas.

ABSTRACT

This dissertation aims to present the design and construction of a terrestrial mobile robot called LOGBOT with traction type differential movement - with two wheels and an impeller to maintain the stability of its structure with the surface. The robot control modes have the telemetry and autonomous.

In the control by telemetry – Remotely Operated Vehicle, communication with the robot control station operator is made by radio at a distance of one kilometer to outdoors, and one hundred meters indoors.

In autonomous control mode – Autonomous Guided Vehicle, the robot has the ability to navigate in indoor environments and unfamiliar with the use of the corridor wall as your reference trajectory to be traveled. The sequence of movements for execution of the path is sent to the control station that performs analysis of performance of the robot.

To perform these tasks in standalone mode, your program includes a reactive intelligent agent that detects behaviors of the environment and decides a course of action to be performed by the robot, aiming to avoid obstacles and to control the speed of the wheels of mobile robots.

The problem of odometer error and its correction based on the use of external sensory information were assessed. Techniques of hierarchical control of the robot as a whole and speed control closed loop of the robot's wheels were used.

The results showed that the LOGBOT mobile robot is able to navigate in indoor environments in the form of a corridor, with stability and accuracy.

Keywords: Mobile Robotics, Control of Non-holonomic Robots, Navigation Robots, Robotic Platforms.

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO	1
1.1 Propriedades de um Robô Móvel.....	5
1.2 Ambiente de Atuação de um Robô Móvel.....	6
1.3 Robôs Móveis Holonômicos e Não-Holonômicos.....	7
1.4 Escopo e Contribuições desta Dissertação	8
1.4 Organização do Conteúdo deste Trabalho.....	10
CAPÍTULO 2: ASPECTOS TEÓRICOS EM ROBÓTICA MÓVEL	12
2.1 Estratégias de Controle para Robôs Móveis.....	13
2.2 Sistemas Não-Holonômicos.....	15
2.3 Funcionalidade	16
2.4 Espaço de Configurações: O Ambiente de Atuação	18
2.5 Arquitetura de Robôs Móveis	19
2.4 Arquitetura e Missão do Robô Móvel LOGBOT	21
2.5 Conclusões.....	22
CAPÍTULO 3: PROJETO DE ROBÔS MÓVEIS DIFERENCIAIS TERRESTRES ...	23
3.1 Seleção dos Motores	23
3.1.1 Considerações sobre Força, Torque e Velocidade em Robótica Móvel	25
3.1.2 Considerações sobre Caixa de Redução, Torque e Velocidade de um Motor	29
3.1.3 Considerações sobre Sistema de Unidades e Peso dos Componentes do LOGBOT	29
3.1.4 Controle do Motor DC Através de Modulação por Largura de Pulso	30
3.2 Seleção do Material para Construção da Estrutura.....	32
3.3 Tipo do Movimento de Tração do Robô.....	35
3.4 Fonte de Energia Embarcada - Baterias	36
3.4.1 Reguladores de Tensão.....	39
3.4.2 Proteção de Circuitos, Isolamento e Redução de Ruídos	39
3.4.3 Dimensionamento das Baterias para Fornecimento de Energia ao Robô LOGBOT	40
3.5 Abordagem em Camadas do Projeto de um Robô Móvel.....	41
3.6 Conclusões.....	43
CAPÍTULO 4: MODELAGEM DE ROBÔS MÓVEIS DIFERENCIAIS	44
4.1 Modelo Cinemático de um Robô Móvel Não-Holonômico.....	45
4.1.1 Considerações físicas sobre restrições Holonômicas e Não-Holonômicas.....	50

4.2	Modelo Cinemático do Robô Móvel LOGBOT	52
4.3	Modelo Dinâmico de um Robô Móvel Não-Holonômico.....	54
CAPÍTULO 5: CONTROLE DE ROBÔS MÓVEIS		56
5.2	Controle de Robôs Móveis Não-Holonômicos.....	58
5.3	Controlador do LOGBOT no Modo Teleoperado.....	60
5.4	Controlador do LOGOBOT no Modo Autônomo - Wall-Following	61
5.4.1	Modelo Matemático do Wall-Following para um Robô Móvel Não-Holonômico.....	62
5.4.2	Geometria da Aquisição de Dados do Vetor de Sensores de Ultra-Som	64
5.4.3	Controlador Proporcional de Trajetórias para Robôs Móveis Não-Holonômicos	65
5.4.4	Algoritmo de Controle.....	66
5.5	Sintonia do Controlador Proporcional de Trajetórias Wall-Following	68
CAPÍTULO 6: CONTROLE INTELIGENTE DE ROBÔS MÓVEIS		69
6.1	Paradigmas de Programação de Agentes Inteligentes para Robôs Móveis.....	70
6.1.1	Paradigma Deliberativo	71
6.1.2	Paradigma Reativo	72
6.1.3	Paradigma Híbrido	73
6.1	Agente Inteligente do Robô Móvel LOGBOT.....	74
CAPÍTULO 7: HARDWARE EMBARCADO PARA ROBÔS.....		76
7.1	Sistemas de Computação em Automação e Robótica.....	76
7.1.1	Sistema Computacional Embarcado do Robô Móvel LOGBOT.....	77
7.2	Sistemas Embarcados em Automação e Robótica	78
7.2.1	Considerações Importantes sobre Sistemas Embarcados	79
7.3	Arquitetura Modular para Hardware de Robôs	81
7.3.1	Módulo de Decisão.....	82
7.3.2	Módulo de Percepção	82
7.3.2	Módulo de Ação	83
7.4	Sensor Inteligente e Interface de Aquisição de Dados.....	84
7.5	Sistema Operacional de Tempo Real Embarcado em Robôs	85
7.5.1	Kernel de Tempo Real.....	86
7.5.2	Aplicação do RTOS em Robótica Móvel	90
7.6	Odometria	91
7.7	Comunicação de Dados em Robótica.....	92
CAPÍTULO 8: RESULTADOS EXPERIMENTAIS.....		96
8.1	Resultados Experimentais no Modo Autônomo	96
8.1.1	Teste I – 10 cm Acima da Referência de Distância com a Parede (Erro Negativo)	98
8.1.2	Teste II – 10 cm Abaixo da Referência de Distância com a Parede (Erro Positivo)	100
8.1.3	Teste III – Mudança Brusca de Posição e Orientação: Vão de uma Porta.....	102
8.1.4	Teste IV – Variação Brusca de Posição e Orientação: Mudança de Parede no Corredor	104

8.2	Análise dos Resultados Experimentais no Modo Autônomo	106
CAPÍTULO 9: CONCLUSÕES E PERSPECTIVAS FUTURAS		107
REFERÊNCIAS BIBLIOGRÁFICAS		111
ANEXO A – O CASO ZUSE E SHANNON. QUEM PRIMEIRO “APLICOU” A ÁLGEBRA DE BOOLE.....		116

Capítulo 1: Introdução

Ainda que possa parecer surpreendente, a ancestralidade dos robôs recua alguns séculos no tempo — ao menos até o momento em que seres humanos construíram a primeira armadilha para a caça de pequenos animais. Esse invento primitivo deu início a um processo rudimentar de automação, que trouxe para a espécie humana benefícios de produtividade e liberação de tempo para a realização de outras tarefas e/ou atividades.

Desde então, e durante muito tempo, a automação foi basicamente empregada na substituição de ações repetitivas, puramente mecânicas, mas esse tipo simples de utilização foi superado e, no fim do século passado, os robôs passaram a ser empregados em tarefas mais complexas e/ou sofisticadas, como por exemplo, a realização de delicadas cirurgias no cérebro de pacientes humanos, performances musicais, soldagem de partes e pintura final de carros produzidos pela indústria automobilística [1], dentre muitos outros.

A Robótica é um importante ramo da Engenharia que se preocupa com o desenvolvimento de robôs ou dispositivos robóticos, e constitui-se numa tecnologia multidisciplinar altamente ativa que busca o desenvolvimento e integração de técnicas e algoritmos para a concepção de equipamentos. Sua meta final é o desenvolvimento de robôs que possam substituir o homem na execução de suas ações e tarefas cotidianas, sejam elas simples ou complexas.

O escritor checo Karel Čapek (1890-1938) introduziu a palavra "Robô" em sua peça de teatro "R.U.R" (*Rossum's Universal Robots*), encenada em 1921 (Praga, República Checa). O termo "robô" realmente não foi criado por Karel Čapek, mas por seu irmão Josef, outro respeitado escritor checo. O termo "Robô" vem da palavra checa "robota", que significa "trabalho forçado".

Por ser um vasto campo de estudo, a robótica é dividida em várias linhas de pesquisa, dentre as quais a robótica móvel é uma das mais ativas e desenvolvidas. O estudo da robótica

móvel surgiu durante a segunda guerra mundial nos anos de 1940, como resultado de muitos avanços técnicos em campos de pesquisa relativamente novos como computação, algoritmos e cibernética. Esses estudos tiveram como principais contribuições os trabalhos desenvolvidos na Alemanha por dois engenheiros aeronáuticos: Konrad Zuse – projetista de computadores eletromecânicos ZX, e pai da informática [2]; e Wernher Von Braun – projetista de foguetes, que após a guerra se tornou o cientista chefe do recém criado programa espacial norte americano [2]. Naquela época os robôs eram em sua maioria bombas voadoras, como os foguetes batizados de V1 e V2, que possuíam um autopiloto primitivo e sistemas de detonação automáticos. No anexo A, há uma discussão sobre as contribuições de Claude Shannon e Konrad Zuse para a informática e uso da álgebra de Boole em sistemas digitais.

Os primeiros robôs móveis foram desenvolvidos na década de 1960, com foco nos objetivos da corrida espacial/armamentista, que era impulsionada pela Guerra Fria entre Estados Unidos e União Soviética (atual Rússia), e seus respectivos aliados. Em 1961 foi lançado nos Estados Unidos, o robô móvel *Stanford Cart*, o primeiro a ser construído no formato de um veículo diferencial semelhante a um carro – ele contava com recursos de aquisição de imagens de uma câmera de vídeo para detectar e seguir uma linha branca fixada ao chão, e era conectado via cabo a um grande *mainframe* que realizava o processamento necessário para o controle de seus movimentos [3]. Em 1970 a União Soviética apresentou o robô móvel *Lunokhod 1*, que foi o primeiro a ser usado em missões espaciais, e tinha como principal objetivo explorar a superfície da lua. Isto o classificou como um *ROV (Remotely Operated Vehicle)* – veículo operado remotamente. Em seu projeto global, o *Lunokhod 1* contou com o uso da teoria de controle e computação embarcada, além de outros conceitos até então desconhecidos, como odometria, telemetria e fusão de dados de sensores, que hoje em dia constituem a base da atual robótica, cuja origem pode ser atribuída a este projeto pioneiro [4] [5].

Nas décadas de 1980 e 1990, os avanços da microeletrônica e informática possibilitaram o desenvolvimento, em meios acadêmicos, de robôs móveis autônomos, que utilizam novos recursos como computadores com poderosos processadores, linguagens de programação de médio e alto nível, sistemas avançados de comunicação, sensores e atuadores de alta precisão, e o surgimento do conceito de agentes de *software* adequados a algoritmos de controle e inteligência artificial. No ano de 1996 a NASA enviou ao espaço o robô móvel denominado de *Sojourner*, um *rover* com a missão de explorar a superfície do planeta Marte, e determinar através de aparelhos de medição, a composição das rochas e do solo marciano. Este robô foi o primeiro a contar com um sistema operacional de tempo real para controle e aquisição de imagens, sistemas *fuzzy* e neural para análise de compostos orgânicos, e microcontroladores para gerenciar cada um dos seus diversos módulos constituintes [6].

Na década de 2000 surgem os robôs móveis comerciais, e a NASA envia o segundo robô explorador ao planeta Marte denominado de *Spirit*. Estes acontecimentos tornaram a robótica móvel mundialmente conhecida, e logo surgiram incentivos para o seu emprego em aplicações terrestre, aquática e aérea. Os robôs móveis da classe *ROV* ganharam visibilidade e/ou popularidade mundial graças ao seu emprego em missões arrojadas, tais como os *ROVs* enviados ao espaço para missões de exploração no planeta Marte no ano de 2004; ou no caso do robô *Scorpio-45* – um *ROV* submarino da real marinha inglesa que atuou com sucesso no resgate dos tripulantes do submarino russo AS-28 no ano de 2005 [7]. Em geral os *ROVs* são empregados nas áreas industrial, militar e aeroespacial. Na maioria das vezes, eles têm a missão de atuar na inspeção de ambientes de difícil acesso ou que contenham materiais perigosos.

As atuais pesquisas em robótica móvel têm ênfase nos robôs de serviço [8], que podem ser usados em diferentes setores da sociedade, tais como na fabricação de produtos industriais, transporte de documentos em escritórios, limpeza e vigilância de residências,

guias de museus, transporte de mercadorias em lojas, transporte de equipamentos médicos e monitoramento de pacientes em hospitais, e etc. O robô deve atuar e executar tarefas nestes ambientes de maneira autônoma ou semi-autônoma.

No Brasil a robótica móvel tem alguns destaques. Em aplicações aéreas, tem-se o desenvolvimento do VANT (Veículo Aéreo Não Tripulado) brasileiro, através da parceria das universidades – USP, IME, ITA, UFMG, UNB, UFRGS e UFSC, e o DRVC/CTI. O VANT tem como missão monitorar fronteiras com outros países e parques ambientais. Na aplicação aquática, tem-se o robô ambiental híbrido denominado de Chico Mendes, que é desenvolvido pelo projeto *Cognitus* da Petrobrás e tem como missão realizar o monitoramento ambiental da área brasileira da região amazônica, com objetivo de coletar informações e estruturar um banco de dados sobre a região Amazônica, a ser disponibilizado para centros de excelência em pesquisas e universidades [9].

O DRVC/CTI – Divisão de Robótica e Visão Computacional do Centro de Tecnologia da Informação Renato Archer, é um centro de pesquisa mantido pelo Ministério da Ciência e Tecnologia Brasileiro – MCT, e além do VANT, conta com o projeto AURORA – um dirigível robótico autônomo monitoramento ambiental, levantamentos urbanos e agropecuários, prospecção mineral e arqueológica, inspeção de grandes estruturas (linhas de dutos, de transmissão, rodovias) e vigilância [29], e o projeto VERO – Veículo Robótico., tem como foco o desenvolvimento gradual de metodologias de navegação autônoma para veículos terrestres em ambientes externos, capacitando-os à realização de diferentes classes de aplicações [38] [39]. Na UFMG o grupo CORO – Computação e Robótica desenvolveu uma plataforma embarcada para tornar veículos comuns em autônomos [40]. Na UFSC, o departamento de automação e sistemas desenvolve robôs de serviço para soldagem de dutos, concerto de cabos das redes de distribuição de energia elétrica, e monitoramento ambiental [8] [41].

1.1 Propriedades de um Robô Móvel

A robótica móvel ocupa-se do estudo e desenvolvimento de máquinas capazes de se locomover em ambientes que em geral são não controlados, ruidosos e desconhecidos. Para atender a estes objetivos, a multidisciplinaridade faz-se necessária. Conhecimentos de mecânica fina, eletro/eletrônica e computação são integrados para dar suporte a esta nova e importante área.

As principais áreas de conhecimento envolvidas são: Engenharias – concepção mecânica e elétrica, modelagem cinemática e dinâmica, controle e estimação, eletrônica, sensores, atuadores, interferência eletromagnética, redes de comunicação, e processamento digital de sinais; Sistemas de Computação – algoritmos e estruturas de dados, sistemas operacionais, processamento paralelo, restrições de tempo real, e inteligência artificial; Matemática e Estatística – teoria das probabilidades, processos estocásticos, modelagem matemática, e curvas parametrizáveis; Humanas – filosofia, psicologia, e impacto social; Multidisciplinares – agentes de inteligência artificial, processamento da informação, visão computacional, otimização, sistemas de tempo real, e controle embarcado.

Para robótica móvel terrestre, os tópicos mais importantes são – locomoção, paradigmas de programação, planejamento de caminhos, localização, mapeamento de ambientes, e cooperação entre múltiplos robôs. Um robô móvel pode ser especificado no seu espaço de trabalho (ambiente onde atua) por seu vetor de configurações $q(t)$. A dimensão de $q(t)$ é o número de graus de liberdade (*degrees of freedom* – *DOF*) do robô e representa o número mínimo de variáveis independentes que, juntamente com a sua geometria, são necessárias para especificar completamente o robô [10]. Por exemplo, um robô móvel terrestre em forma de carro possui três graus de liberdade e pode ser especificado pela configuração $q(t) = [x(t), y(t), \theta(t)]^T$, onde x e y especificam a posição cartesiana do robô em relação a um referencial fixo $\{W\}$ no ambiente e θ especifica a sua orientação.

1.2 Ambiente de Atuação de um Robô Móvel

A estrutura e os componentes mecânicos que compõem um robô móvel devem ser definidos de acordo com as propriedades físicas do ambiente onde atuará. O robô deve perceber algumas destas propriedades de forma que possa interagir e se locomover pelo ambiente de maneira ordenada, ou seja, continuamente coletando dados deste ambiente de forma a desviar de obstáculos e realizar sua função programada pelo projetista. Esta percepção é possível, através do uso de sensores acoplados ao robô, cujos dados coletados servem a realimentação para o estado de decisão da próxima atitude do robô. Os sensores mais comuns são – posição, velocidade, distância, e visão.

Para iniciar suas atividades o robô deve estar localizado em um ponto inicial a um referencial fixo $\{W\}$ no ambiente de atuação, com o objetivo de se localizar no ambiente durante seus movimentos. O robô quando passa a se mover a partir de um ponto inicial, a sua configuração $q(t)$ chega a $\dot{q}(t) = [\dot{x}(t), \dot{y}(t), \dot{\theta}(t)]^T$.

O robô deve contar com um algoritmo de controle para tomada de decisão sobre qual atitude tomar após se localizar no ambiente de atuação, com o objetivo de navegar no ambiente para cumprir suas tarefas programadas.

Na localização os dados mais importantes são – a posição proporcional a contagem dos pulsos durante o movimento; a velocidade inversamente proporcional ao intervalo de tempo entre cada pulso; e o ângulo de orientação. Estas variáveis são mensuradas geralmente por um sensor do tipo *encoder* fixado ao eixo de cada motor, e são conhecidas como variáveis internas, pois estão relacionadas aos movimentos do robô, cujo processo de aquisição e análise destes dados é denominado de odometria.

Na navegação os dados referentes à odometria são comparados a um mapa do ambiente de atuação, que é gravado na memória do robô, e com dados coletados do ambiente como: distância frente a obstáculos ou paredes, coletados geralmente por sonares ou sensores

de distância a laser; marcas naturais ou artificiais do ambiente usadas como referência (*Landmarks*), que geralmente são linhas formadas pelas quinas de um corredor ou as lâmpadas de um teto, e são identificadas, através da aquisição de imagens de uma câmera de vídeo. Estas variáveis são conhecidas como variáveis externas do robô, pois estão relacionadas aos dados externos, coletados do ambiente de atuação do robô.

O algoritmo de controle da navegação realiza a fusão dos dados relacionados às variáveis internas e externas ao robô, com o conhecimento e planejamento do mapa, e após isto gera um controle de movimento, de forma que o robô execute a ação correta para o estado atual em que está localizado no ambiente. A fusão destes dados geralmente pode ser feita por - lógica booleana, filtros de *Kalman e Bayes*, *dead reckoning*, ou sistema *fuzzy*. O controle de movimento (tensão aplicada aos motores) que origina a ação do robô, geralmente é feito por – controlador booleano, controlador *P.*, *P.I.* ou *P.I.D.*, ou controlador *fuzzy*.

1.3 Robôs Móveis Holonômicos e Não-Holonômicos

Em robótica móvel terrestre, o modelo cinemático relaciona as velocidades dos atuadores (rodas) às velocidades de um referencial fixo no robô, geralmente com origem em seu centro de massa. O modelo cinemático mais simples para um robô móvel é dado por: $\dot{q}(t) = G(q)u(t)$, onde $\dot{q}(t)$ é a derivada de $q(t)$, e $u(t)$ é o vetor de controle, representado pelas velocidades de atuação do robô. A matriz $G(q)$ é quadrada se o vetor de controle tiver dimensão igual ao número de graus de liberdade do robô, este fato simplifica o controle do robô [10]. Porém, para a maioria dos robôs móveis a matriz $G(q)$ não é quadrada, já que o número de velocidades de atuação (sinais de controle) é menor que o número de graus de liberdade do robô. Este fato é geralmente caracterizado por restrições de movimento que não permitem o controle independente de todas as variáveis do vetor de configurações.

Por exemplo, a situação onde um carro deve ser estacionado. O carro não pode se mover lateralmente e, portanto, diversas manobras devem ser executadas para estacioná-lo, ou seja, para controlar sua configuração (x, y, θ) . Restrições deste tipo são conhecidas como restrições não-holonômicas. O efeito desta restrição é reduzir o número de velocidades independentes do robô, tornando o seu sistema de locomoção subatuado. Os robôs com restrições não-holonômicas são classificados como robôs não-holonômicos, enquanto que aqueles que não possuem estas restrições são classificados como robôs holonômicos.

Os esforços de projetos são geralmente voltados para robôs não-holonômicos, no desenvolvimento de algoritmos de controle e técnicas aperfeiçoáveis para os movimentos possíveis do robô, a fim de respeitar a restrição não-holonômica de um robô móvel diferencial terrestre dada por $\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$, que demonstra que a componente de velocidade perpendicular às rodas do robô deve ser nula [10] [11]. A sua consequência é a dependência entre as velocidades \dot{x} e \dot{y} das rodas usadas em sua locomoção.

1.4 Escopo e Contribuições desta Dissertação

Um dos principais objetivos no projeto de um robô móvel é que este execute suas tarefas de maneira autônoma, e que dependa cada vez menos de auxílio humano para o seu controle. Os robôs móveis da atualidade são controlados remotamente por telemetria, e estão em fase de adaptação para também terem controle autônomo, para torná-los robôs de serviço.

Isto é uma realidade, devido o forte investimento de governos e grandes empresas do setor em pesquisas e desenvolvimento de robôs móveis autônomos, ou de controle híbrido, ou seja, podem ser tanto controlados remotamente por um operador, como também controlados autonomamente por seu sistema inteligente. Estes tipos de controle e os conceitos abordados nas seções anteriores constituem os focos principais desta dissertação, onde eles são abordados de maneiras específicas, as quais são brevemente descritas a seguir.

Esta dissertação propõe o uso da abordagem em camadas, que define os requisitos e passos para o projeto global do robô móvel diferencial não-holonômico LOGBOT, cujo processo de desenvolvimento é dividido pelas camadas – mecânica, eletrônica, e computacional. O LOGBOT foi totalmente desenvolvido na Universidade Federal do Pará, através do Programa de Pós-Graduação em Engenharia Elétrica, e seus componentes foram adquiridos no mercado local e nacional.

O modelo cinemático de um robô móvel diferencial não-holonômico é descrito em detalhes até a sua configuração no espaço de estados, assim como os requisitos gerados pela restrição não-holonômica para o projeto do algoritmo de controle dos movimentos que o robô pode exercer. A dinâmica do robô móvel tem como principal objetivo garantir que o robô siga uma trajetória pré-especificada.

Em robótica móvel a trajetória é uma seqüência contínua de configurações parametrizadas pelo tempo. Esta dissertação propõe o uso do algoritmo Wall-Following com correção de posição e orientação do robô com relação a sua distância com a parede. Este algoritmo de controle tem como principal característica a parede do ambiente de atuação como a referência de trajetória que o robô deve seguir, e que deve se manter a uma distância definida (*set-point*) da parede.

É desenvolvido um dispositivo para aquisição e tratamento de informações de um vetor de sonares para realizar a varredura do ambiente e assim seguir uma parede e detectar obstáculos, desta forma o robô pode navegar no ambiente que o circunda, onde executa suas tarefas. A fusão dos dados do vetor de sonares é testada por técnicas e combinação sensorial.

Porém os sensores têm taxas de erros que são acumulativos, e que somadas a perturbações do ambiente ou até mesmo da própria estrutura do robô móvel podem gerar erros de detecção de obstáculos, e por conseqüência erros de navegação, e isto impossibilita o robô de executar suas tarefas de maneira correta.

Esta dissertação propõe o uso de um controlador proporcional de trajetórias para Wall-Following para o robô móvel não-holonômico LOGBOT. O controlador tem capacidade para suavizar os movimentos de desvio de obstáculos e troca de parede em um corredor.

O desempenho do robô no modo autônomo para execução de suas tarefas é testado no corredor do prédio da engenharia elétrica da Universidade Federal do Pará. No modo teleoperado o robô é testado no parque de manejo florestal da Universidade Federal Rural da Amazônia. O *LOGBOT* é um projeto de baixo custo, se comparado aos robôs importados com as mesmas funções, e que pode ser construído de acordo com a realidade do cenário nacional.

1.4 Organização do Conteúdo deste Trabalho

Esta dissertação encontra-se organizada na forma descrita a seguir.

O Capítulo 2 apresenta os conceitos sobre robôs móveis como: uso de sensores para base de informação em suas informações, e outros aspectos como: anatomia, tipo de controle e funcionalidade de robôs móveis.

O capítulo 3 apresenta os conceitos e cálculos necessários para a escolha dos motores válidos para o robô, a requisitos necessários para a construção da estrutura (*frame*) do robô. O capítulo também apresenta os cálculos necessários para os requisitos da fonte de energia embarcada a ser usada no robô, e quais tipos de bateria podem ser usados. A compreensão e identificação da frequência do motor, e da frequência de modulação por largura de pulso para controle do motor são mostradas.

O Capítulo 4 contém o modelo cinemático para um robô móvel diferencial não-holonômico, descrevendo principalmente os graus de liberdade do robô e do controlador para robôs com restrições não-holonômicas.

O Capítulo 5 descreve o controle de robôs móveis, arquiteturas controle hierárquico, o método de navegação *Wall-Following* baseado em sensores de ultra-som, e o projeto de um controlador proporcional de trajetórias para o robô móvel não-holonômico LOGBOT.

O Capítulo 6 apresenta o projeto e o uso de agentes inteligentes de software para uso como gerenciadores de tarefas que o robô pode executar perante a um determinado comportamento do mundo onde evolui – Percepção/Ação e Condição/Ação

O Capítulo 7 apresenta o projeto global do hardware embarcado do robô móvel LOGBOT, e mostra a importância do sistema computacional e suas interfaces, assim como o uso de sistemas embarcados e arquiteturas modulares no desenvolvimento de hardware em automação e robótica.

O Capítulo 8 apresenta os resultados experimentais desta dissertação, e considerações dos controladores e métodos usados para navegação autônoma.

Finalmente, o capítulo 9 apresenta as conclusões desta dissertação, juntamente com as perspectivas futuras sobre o projeto LOGBOT e novos trabalhos.

Capítulo 2: Aspectos Teóricos em Robótica Móvel

Os robôs móveis com rodas (WMR – *Wheeled Mobile Robots*) autônomos têm sido usados para realizar uma grande variedade de tarefas, permitindo aplicações que vão desde as associadas à substituição do homem em ambientes hostis até a exploração da mata virgem na floresta Amazônica.

Robôs móveis podem ser classificados segundo o meio em que evoluem: subaquáticos, terrestres ou aéreos. O desenvolvimento de capacidade de operação autônoma para esses robôs constitui importante e ativo tema de pesquisa em todo o mundo, dada a ampla gama de aplicações, como por exemplo: sensoriamento remoto, estudos e monitoração ambiental, agricultura, inspeção, vigilância, atuação em áreas de risco, etc.

Robôs móveis autônomos têm como base a combinação de informações sensoriais proprioceptivas (i.e., sobre o próprio robô) e externoceptivas (i.e., sobre o ambiente onde o robô evolui). Sensores comumente usados são: GPS, odométricos, centrais inerciais, magnetômetros, altímetros, câmeras, sonares, lasers, radares. As informações proprioceptivas e externoceptivas, devidamente interpretadas, permitem tanto a localização do robô no ambiente que o circunda (em termos de posição e orientação), quanto à percepção desse próprio ambiente (em termos, por exemplo, de referências geo-localizadas e suas características, obstáculos, caminhos livres, etc.). Os conceitos sobre localização e navegação são fundamentais no planejamento de caminhos de um robô móvel autônomo:

- Localização: Permite que robôs móveis autônomos ou operados remotamente determinem suas posições no ambiente onde atua.
- Navegação: Permite que robôs móveis possam evoluir autonomamente no ambiente que o circunda, contornando obstáculos e paredes com objetivo de chegar ao local programado. Existem dois tipos de navegação:

- Baseada em Comportamento: Os robôs usam sensores (geralmente exteroceptivos) para reagir ao estado atual do ambiente, o que dispensa o conhecimento prévio do ambiente onde atua. O agente é reativo e baseado em regras percepção-ação.
 - Baseada em Mapas: Os robôs tentam se localizar coletando dados de sensores (geralmente proprioceptivos) para que possam atualizar sua posição a respeito de um mapa do ambiente. O agente apresenta uma pré-concepção do ambiente onde o robô atua.
- Odometria: A função de odometria permite obter a posição real do robô a cada instante de tempo, através de sensores odométricos como encoders de movimento, ou contadores de passo.

2.1 Estratégias de Controle para Robôs Móveis

Segundo o tipo de controle, os robôs podem ser classificados em três categorias: Teleoperados, Semi-Autônomos e Autônomos. Nos robôs teleoperados, o operador comanda todos os movimentos do robô. Nos Semi-Autônomos, o operador indica o macro comando a ser executado e o robô o faz sozinho. Nos autônomos, o robô realiza suas tarefas autonomamente, tomando decisões próprias e baseando-se nos dados obtidos do ambiente que o circunda. Em geral, os sistemas de controle de robôs móveis autônomos possuem dois níveis principais como mostrados na figura 2.1 [16]. O nível superior, ou *planejamento de trajetórias* é responsável pela escolha de um melhor caminho a ser seguido pelo robô. O planejamento de trajetórias pode ser resumido da seguinte maneira – “dado um robô móvel com localização e orientação iniciais, uma posição e orientação alvo e um conjunto de obstáculos localizados no espaço, encontre um caminho contínuo para o robô do estado inicial

até o objetivo sem que haja colisão com os obstáculos ao longo do percurso. A trajetória gerada indica além das posições e orientações do robô no espaço, os perfis de velocidades e acelerações com que o robô deve se deslocar ao percorrer estas trajetórias”. Este problema aparenta ser simples, porém não está completamente solucionado. O problema torna-se ainda mais complicado quando o ambiente é não controlado e dinâmico, ou seja, os obstáculos e alvo se movem.

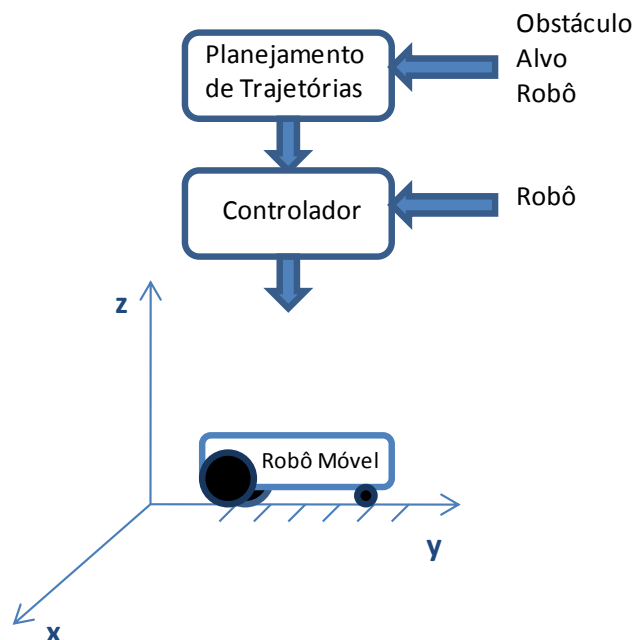


Figura 2.1 – Níveis de controle de um robô móvel autônomo.

Na figura 2.1, o nível inferior, ou controlador é responsável por garantir que o robô siga a trajetória escolhida pelo nível superior. O planejamento de trajetórias pode ser visto como um gerador de referências para o controlador. Por isso, muitas vezes, o controlador é considerado como um nível de rastreamento de trajetórias [16]. Exceto em alguns casos, onde os dois níveis são calculados na mesma taxa, o controlador tem em geral um período de amostragem menor que o nível de planejamento de trajetórias. A maior dificuldade desde controle em robôs móveis autônomos está no fato de os mesmos serem, em geral sistemas multivariáveis, não-holonômicos e não-lineares [10]. A característica multivariável é evidente

quando se sabe que os robôs são acionados por dois ou mais atuadores e têm como saída uma posição e orientação no espaço ou no plano.

2.2 Sistemas Não-Holonômicos

Uma definição formal para a característica não-holonômica é: “Um sistema é chamado não-holonômico, quando a velocidade de um sistema mecânico satisfaz uma condição de igualdade que não pode ser escrita como uma condição equivalente de posição” [17]. O termo holonômico é atribuído a Hertz [18], e significa “universal”, “integral”, “integrável” (literalmente: *holo* = o todo, conjunto, totalidade. – *nomia* = lei), portanto, sistemas não-holonômicos podem ser interpretados como sistema não integráveis.

A abordagem matemática a este tipo de problema é realizada através de ferramentas da geometria diferencial, e o desenvolvimento da teoria iniciou-se há mais de 150 anos, baseado numa série de artigos clássicos sobre a mecânica não-holonômica de matemáticos e físicos tais como: *Hertz, Voss, Hölder, Routh, Voltera*, dentre outros. Logo se definem como não holonômicos sistemas com dimensão finita onde algum tipo de restrição é imposta a um ou mais estados do sistema. Estas limitações podem ser provocadas pela conservação do momento angular, condições impostas pela impossibilidade de deslocar em uma ou mais direções, como resultado da imposição de restrições durante o projeto do sistema de controle, pelo fato de o sistema não ter atuadores em todas as direções do espaço do problema, e em várias outras situações. Existem três classes de sistemas onde restrições não-holonômicas aparecem [17]:

→ Restrição de não-deslize: A condição de não-deslizamento ou de rolamento puro significa que a velocidade linear no ponto de contato é zero. Esta restrição é não-integrável, isto é, não redutível a uma restrição de posição, e, portanto é não-holonômica.

- Conservação do momento angular.
- Sistemas mecânicos subatuados: Sistemas nos quais a dimensão do espaço de configurações excede o espaço das entradas de controle.

2.2.1 Modelo de Robôs Não-Holonômicos

A maioria das estratégias propostas para controle de robôs autônomos, tanto no nível de planejamento como no controlador propriamente dito, utilizam-se de algum conhecimento do processo, ou seja, da cinemática ou dinâmica do robô [16]. Em CAMPION, et al.[24], o artigo aborda os modelos cinemáticos e dinâmicos de robôs móveis não-holonômicos, e são feitas várias observações sobre as diversas estruturas de robôs móveis com rodas. Apesar disso, não se têm encontrado muitas publicações referentes à modelagem, principalmente relacionados à dinâmica de robôs móveis.

Métodos de identificação de modelos, baseados na observação experimental do comportamento dos robôs, são, entretanto, pouco difundidos na prática [25]. Em PEREIRA [16] é usado modelos autoregressivos na predição do movimento de objetos móveis para facilitar o planejamento de trajetórias, cuja aplicação foi usada na predição do movimento de robôs para o futebol de robôs.

2.3 **Funcionalidade**

Ao agrupar robôs segundo sua funcionalidade, identifica-se quatro grupos: robôs industriais, robôs de serviço, robôs de campo e robôs pessoais. Contudo, há uma sobreposição entre os três primeiros, devido à diferença dos ambientes onde atuam e a necessidade de maior autonomia [8].

Robôs industriais são usados em linhas de produção. Estes robôs recebem tarefas determinadas a priori na forma de uma seqüência explícita de ações e executam este programa automaticamente. O ambiente é completamente estruturado e ajustado, para a execução da

tarefa, tendo o robô o conhecimento exato da sua posição e da posição dos objetos. Em geral, robôs industriais são plataformas móveis usadas para tarefas pesadas, como transporte de materiais e produtos finais em sistemas de manufatura.

Os robôs de serviço industrial possuem as características de um robô de serviço, porém atuam em um ambiente completamente estruturado.

Os robôs de serviço de campo atuam em ambientes externos que podem ser previamente modelados ou não. Em geral, caso haja um modelo, este é precário e há a necessidade do processamento sensorial para complementar o modelo existente. São usados na realização de tarefas agrícola e para navegação em auto-estradas. Os robôs de campo trabalham em ambientes não estruturados, pouco conhecidos e geralmente perigosos. Suas principais atividades são a exploração (espacial, de cavernas, vulcões), a mineração e a limpeza de acidentes nucleares.

Os robôs pessoais são aqueles vendidos em prateleiras, que não desenvolvem tarefas específicas, mas interagem com os humanos e aprendem a localizar-se no ambiente.

Os robôs pessoais e robôs agrícolas são uma linha de pesquisa impulsionada na segunda metade dos anos 2000, e que vem ganhando força devido o aumento da produção e a redução do custo dos dispositivos que compõem estas tecnologias.

Os robôs de entretenimento são uma linha de pesquisa impulsionada no início da década de 2000, no desenvolvimento do futebol de robôs, robôs bio inspirados, e robôs de combate para atuar em torneios denominados de guerra de robôs.

Em robótica agrícola existem desafios novos para a década de 2010, como o desenvolvimento de estruturas reconfiguráveis conforme a evolução em um ambiente cuja estrutura, pois o veículo robô está sujeito a inclinações, escorregamentos, efeitos dinâmicos da interação veículo-terreno, etc. Além disto, outro desafio é a integração de um robô móvel com a infra-estrutura de uma RSSF - rede de sensores sem fio, para agricultura de precisão.

2.4 Espaço de Configurações: O Ambiente de Atuação

O espaço de configurações, C , é o conjunto de todas as possíveis configurações q de um robô no ambiente que o circunda, logo denominado também de ambiente de atuação. Assim, um robô pode ser representado por um ponto em seu espaço de configurações, cuja dimensão é igual ao seu número de graus de liberdade.

No espaço de configurações, obstáculos são mapeados com um conjunto de configurações proibidas, C_{obs} . Fisicamente, uma configuração pertencente à C_{obs} indica uma colisão do robô com um obstáculo no espaço de trabalho (ambiente de atuação). A Figura 2.2 mostra o mapeamento de um ambiente no espaço de configurações a partir da figura 2.2(a), basta deslizar o robô ao redor de cada um dos obstáculos. A origem do robô traçará as bordas de C_{obs} . Matematicamente, esta operação pode ser expressa pela soma de Minkowski [10]. O espaço fora da região hachurada é o espaço de configurações livre, C_{free} . Caso o robô pudesse rotacionar seu espaço de configurações seria tridimensional, (x, y, θ) . Para cada orientação do robô o espaço seria semelhante ao mostrado na figura 2.2(b).

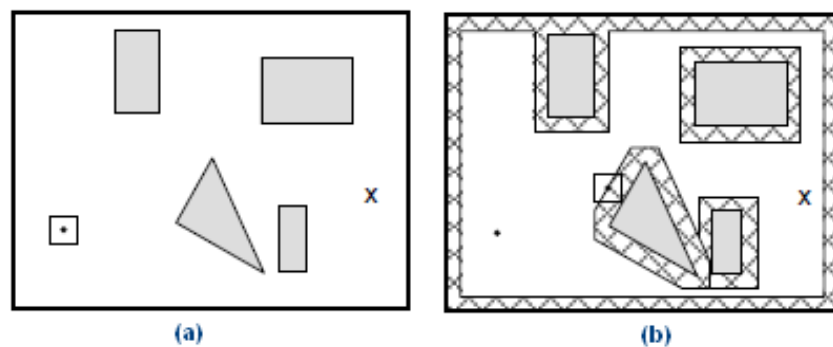


Figura 2.2 – Mapeamento de um ambiente no espaço de configurações.

Uma vez construído o espaço de configurações, o problema de planejamento de caminhos corresponde a encontrar uma seqüência de pontos entre a configuração inicial q_o e a configuração final q_d que não intercepte o espaço proibido representado por C_{obs} . Este princípio será usado nas próximas seções. Todas as técnicas mostradas a seguir encontrarão

caminhos no espaço de configurações, cuja construção deve ser feita *a priori*. Para que o robô siga um determinado caminho, este deve ser transformado em uma trajetória que, além de especificar as posições, especifica também as velocidades e acelerações do robô. As trajetórias são calculadas ajustando-se funções multidimensionais $f(q,t)$ que passam pelos pontos do caminho em intervalos de tempo predeterminados.

2.5 Arquitetura de Robôs Móveis

Um robô móvel é composto por uma arquitetura de hardware e software que é responsável por garantir a execução correta de sua navegação para execução de uma determinada tarefa. Esta arquitetura para navegação de robôs móveis pode ser dividida em níveis hierárquicos, como visto na figura 2.3.

Cada um dos níveis apresentados na figura 2.3 constitui áreas de pesquisa que podem ser tratadas individualmente dentro do estudo da robótica. A seguir os níveis serão brevemente descritos:

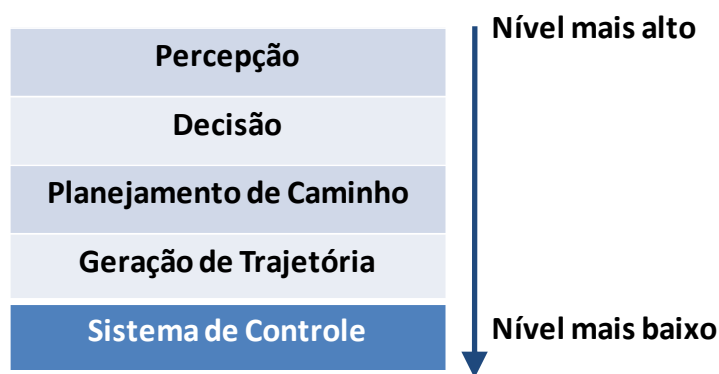


Figura 2.3 – Níveis hierárquicos da arquitetura de navegação para robôs móveis

O nível Percepção representa a interface do robô com seu ambiente de trabalho. Este nível é composto por sensores e algoritmos para tratamento das informações coletadas. As informações são tratadas de forma a ter algum significado para o sistema robótico e para a tarefa que este necessita executar. Em seguida as informações são passadas para o nível de

decisão. As informações extraídas do ambiente e do estado do robô são, normalmente, a localização do robô no ambiente, a localização de obstáculos e informações sobre a velocidade do robô.

O nível de Decisão é responsável pela inteligência e tomada de decisões, de acordo com as informações obtidas através do nível de percepção. Estas decisões variam com o tipo de tarefa que se deseja realizar. Normalmente são utilizadas, neste nível, árvores de decisão ou técnicas de inteligência artificial.

No nível de Planejamento de Caminho é feito o planejamento do melhor caminho que leva o robô de sua localização original a uma determinada localização desejada, levando em consideração os obstáculos e as dimensões do robô. Define-se como localização ou configuração do robô (postura ou pose) a soma das informações de posição e orientação. Há várias técnicas que podem ser usadas neste nível, como campos de potencial, diagramas de Voronoi, decomposição exata em células convexas, dentre outras.

O nível de Geração de Trajetória é responsável por determinar as restrições temporais para a execução de um determinado caminho, gerando, desta forma, uma trajetória. A geração de trajetórias impõe as velocidades e acelerações que o robô deve adotar em cada instante de tempo para que a execução do caminho seja realizada no período especificado. Uma das possíveis técnicas para geração de trajetórias é através do uso de polinômios de interpolação.

O nível Sistemas de Controle é responsável por garantir a execução das trajetórias através da geração de sinais de comando para os atuadores. Desta forma, este nível garante que as velocidades e acelerações impostas pelo nível de geração de trajetória sejam executadas de forma correta. Normalmente os sistemas de controle para robôs móveis são sistemas baseados em realimentação do estado do robô, isto é, informações a respeito da velocidade e localização do robô, que foram geradas pelo nível de percepção.

Os sinais de controle gerados pelo sistema de controle são comandos para os acionamentos dos atuadores do robô (motores), que podem ser de origem eletrônica, pneumática, mecânica, dentre outros. Os atuadores também podem ser de diversos tipos, por exemplo, motores elétricos, propulsores a jato, ou motores de combustão. Os atuadores são conectados às partes móveis do robô que possibilitam a locomoção, como rodas, hélices, pás de uma turbina, discos de inércia, esteiras ou patas.

2.4 Arquitetura e Missão do Robô Móvel LOGBOT

O robô no modo teleoperado pode ser controlado pelo operador a uma distância de até 1 km em áreas abertas, e de até 100 metros em ambientes fechados. No modo autônomo usado o controlador baseado em regras *Wall-Following* que é um algoritmo que usa a parede lateral e paralela em relação ao robô, como referência no seguimento de sua trajetória, através da realimentação de sensores do tipo sonares, cujo erro da distância de referência da parede e orientação do robô são usados no controlador, cujas saídas são aplicadas aos atuadores que são dois motores de corrente contínua [23] [35]-[37]. Os detalhes sobre o modelo do controle no modo teleoperado e autônomo é abordado no capítulo 4 desta dissertação. O robô móvel LOGBOT desenvolvido nesta dissertação têm as seguintes características:

- Anatomia: Terrestre com rodas;
- Atuador: Dois Motores de Corrente Contínua;
- Mecanismo de Locomoção: Rodas;
- Sistema Mecânico: Não-holonômico;
- Tipo de Controle: Teleoperado, Semi-Autônomo e Autônomo;
- Funcionalidade: Acadêmica, Inspeção e Monitoramento Ambiental.

Os motores são do tipo corrente contínua com torque de 44 kgf.cm e velocidade angular de 115 RPM. A base estrutural é confeccionada em alumínio naval com espessura de 4mm. A bateria de chumbo é valvulada tem tensão nominal de 12V com corrente de 9A/h

para fornecer energia aos motores. As bateria de metal Hidreto tem tensão nominal de 12V com corrente de 2.3A/h para fornecer energia aos circuitos eletrônicos. As barras de ferro de 1/8 polegadas são usadas para sustentar as placas de acrílico que suportam os circuitos eletrônicos. Na figura 2.4 é visto o robô móvel LOGBOT.



Figura 2.4 - Robô Móvel LOGBOT (UFPA-BRASIL, 2009).

2.5 Conclusões

Neste capítulo foram descritos conceitos importantes sobre o estudo da robótica móvel como: propriedades de sensores externoceptivos e proprioceptivos; modelos de localização em ambientes externos e internos; navegação baseada em comportamento e navegação baseada em mapa. Pela descrição dos níveis hierárquicos da arquitetura de navegação é possível notar o nível interdisciplinar que envolve o projeto e construção de robôs móveis autônomos. Conforme descrito pela figura 2.2 o nível sistemas de controle é o objeto de estudo desta dissertação e será abordado em mais detalhes nos próximos capítulos.

Capítulo 3: Projeto de Robôs Móveis Diferenciais Terrestres

O projeto de um robô móvel tem como etapa inicial, a definição de itens vitais:

1. Motores que suportem a tarefa designada para o robô;
2. Estrutura que suporte as particularidades físicas do ambiente de atuação;
3. Tipo da Tração de Movimento do Robô;
4. Fonte de Energia Embarcada.

A estrutura deve ser resistente o suficiente para suportar os fatores físicos do ambiente de atuação do robô, a fim de não comprometer sua estrutura com avarias do material que o compõe. E os motores devem ter força e velocidade suficientes para que suporte a carga do robô em seus movimentos, de forma a executar manobras precisas, tudo isto com o objetivo de evitar maior fadiga mecânica dos motores.

3.1 Seleção dos Motores

Em robótica móvel é comum o uso de motores de corrente contínua (Direct Current Motor). A comutação de um motor CC é feita por um circuito eletrônico de controle denominado de ponte-H. O dimensionamento e a configuração de um motor CC se baseiam no peso do robô móvel e nas tarefas que realiza [12], conforme mostra a figura 3.1.

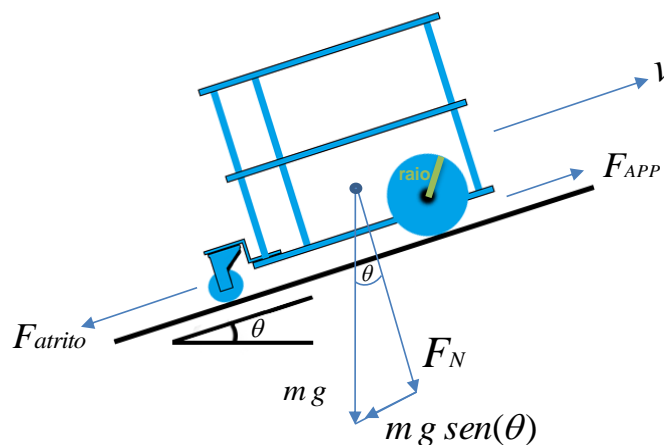


Figura 3.1 - Robô móvel LOGBOT subindo uma rampa.

Ao considerar que o robô móvel LOGBOT mostrado na figura 3.2 tem duas rodas acionadas independentemente por dois motores CC, pesa 9.4kg e é projetado para subir rampas de até 15° de inclinação, com velocidade máxima de 0.42m/s (70% PWM), deve-se calcular a potência necessária do motor CC, através da equação 3.1.

$$P_m = F_{aplicada} \times v = (F_{atrito} + F_p) = m \times g(\mu \cos(\theta) + \text{sen}(\theta)) \times v , \quad (3.1)$$

considerando o coeficiente de atrito $\mu = 0.5$ (borracha da roda com o piso), obtém-se:

$$P_m = 9,4 \times 9,8 \times (0,5 \times \cos(15^\circ) + \text{sen}(15^\circ)) \times 0,42 = 10,46 \text{ Watts}$$

Para super-dimensionar os motores, de modo a compensar a incerteza associada ao valor do coeficiente de atrito e outros fatores não considerados, multiplica-se o resultado obtido por 3, resultando em $P_m = 31,38 \text{ Watts}$. Uma vez que são usados dois motores iguais, cada um com a seguinte especificação:

- Fabricante: Symphony Robotics Brasil;
- Modelo: MR-3;
- Tensão Nominal de Operação: 12 VDC (Linear);
- Corrente de Operação sem Carga: 0.5 A;
- Corrente de Operação Máxima com Carga: 1.8 A;
- Corrente de Stall (Travado): 4 A;
- Velocidade Angular Máxima sem Carga: 115 RPM;
- Velocidade Angular Máxima com Carga: 78 RPM;
- Torque de saída: 44,0 kgf.cm = 0,44 kgf.m = 431,0 N.cm = 610,5 oz-in;
- Peso: 1.5 kg;
- Dimensões máximas: 180 mm (comprimento) e 60 mm (diâmetro).

Cada motor tem potência de 21,6 Watts. Logo, o sistema motor do robô como um todo suporta o consumo de até 43,2 Watts, isto valida configuração do corpo-rígido do robô para as tarefas definidas em seu projeto.

3.1.1 Considerações sobre Força, Torque e Velocidade em Robótica Móvel

A força é requerida para que o robô saia do estado de eminência e entre em movimento como visto na figura 3.2, para subir rampas e andar em terrenos com muitos desníveis, ou para alcançar a velocidade máxima. É comum o uso do termo torque para se referir à força do robô, porém são grandezas diferentes e que estão relacionadas. No geral as informações sobre a configuração do motor, como suas constantes, força, torque, velocidade e aceleração são importantes.

A força que este motor pode exercer de modo a movimentar um mecanismo não depende apenas da potência, mas do lugar em que ela é aplicada a partir do ponto central de seu eixo, logo o uso de outra grandeza que independe da distância do centro do eixo se faz necessário, observando que normalmente este eixo vai ser usado justamente para acionar outros mecanismos. Esta outra grandeza é o torque, que consiste em:

$$\text{Torque} = \text{Força} \times \text{Distância} \quad (3.2)$$

Se um pequeno motor tem um torque de 10 N.cm, significa que a uma distância de 1 cm do eixo ele pode exercer uma força de 10N, por exemplo. Este mesmo motor irá produzir uma força de 2N se uma roda de 5 cm de diâmetro for acoplada diretamente ao seu eixo. Em algumas literaturas o termo momento torçor é usado ao invés de torque.

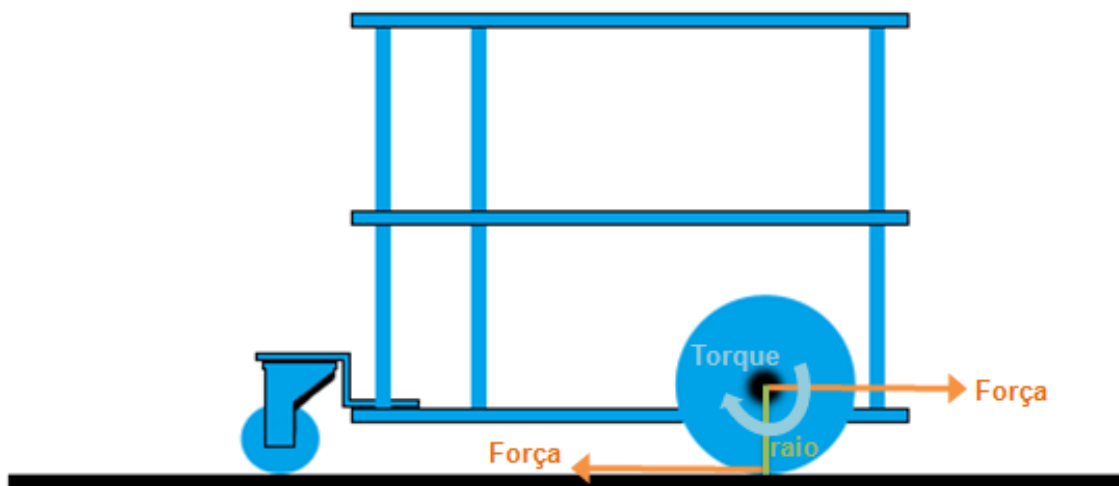


Figura 3.2 – Ilustração sobre força e torque para o robô LOGBOT se locomover em um plano.

No robô móvel LOGBOT o mecanismo acoplado no eixo de cada motor são rodas constituídas em material de polipropileno com revestimento de borracha automotiva (fabricante: *Colson Wheel, USA*), o que a torna leve e altamente resistente a impactos de flexão ou fadiga. Cada roda tem um diâmetro de 14.8 cm, logo por ser o mecanismo acoplado ao motor, a distância usada na equação 3.1 é:

$$Distância = raio da roda \quad (3.3)$$

O próximo passo é calcular a força na periferia da roda de acordo com o seu raio [cm] e torque do motor [N.cm]:

$$Força = \frac{Torque}{Distância} \quad (3.4)$$

$$Força = \frac{431}{7,4} \Rightarrow Força = 58,24N$$

Pelo princípio fundamental da dinâmica calcula-se a aceleração do robô [m/s^2], através da força [N] do motor, e da massa do robô [kg]:

$$Força = massa \times aceleração \quad (3.5)$$

$$aceleração = \frac{Força}{massa} = \frac{58,24}{9,4} \Rightarrow aceleração = 6,19m/s^2$$

Na especificação do motor é comum a informação da velocidade angular ω_n em rotações por minuto – RPM, porém para efeitos de cálculo se usa rotações por segundo – RPS [Hz]. O motor sem carga tem velocidade de 115 RPM, cuja conversão para RPS é:

$$Rotações\ por\ Segundo = \frac{N_{RPM}}{60} \quad (3.6)$$

Com o motor funcionando sem carga:

$$Rotações\ por\ Segundo = \frac{115}{60} \Rightarrow RPS_{SEMC} = 1,91Hz$$

Com o motor funcionando com carga:

$$\text{Rotações por Segundo} = \frac{78}{60} \Rightarrow RPS_{COMC} = 1,3Hz$$

O calculo da velocidade linear máxima do robô [m/s]:

$$V_{robô} = RPS_{COMC} \times 2 \times \pi \times \text{Raio da Roda} \quad (3.7)$$

$$V_{LOGBOT} = 1,3 \times 2 \times \pi \times 0,074 \Rightarrow V_{LOGBOT} = 0,60m/s$$

Para efeitos de projeto do controlador é importante saber em quanto tempo o sistema motor do robô demora em alcançar a velocidade linear máxima. Isto pode ser calculado pela função horária da velocidade escalar [12]:

$$V_{robô} = \text{Velocidade inicial} + \text{Aceleração} \times \text{Tempo} \quad (3.8)$$

$$0,60 = 0 + 6,19 \times \text{Tempo} \Rightarrow \text{Tempo} = \frac{0,60}{6,19} \Rightarrow \text{Tempo} = 0,096\text{segundos}$$

O robô leva aproximadamente 0,1 segundos (100 milissegundos) para alcançar a sua velocidade máxima. Uma identificação experimental em primeira ordem do motor DC para controle de velocidade é realizada para comparar os dados calculados para o robô. No ensaio é aplicado um degrau de ganho DC com amplitude de 4 Volts no motor, e os dados são recolhidos por um osciloscópio digital, cujo gráfico é mostrado na figura 3.3.

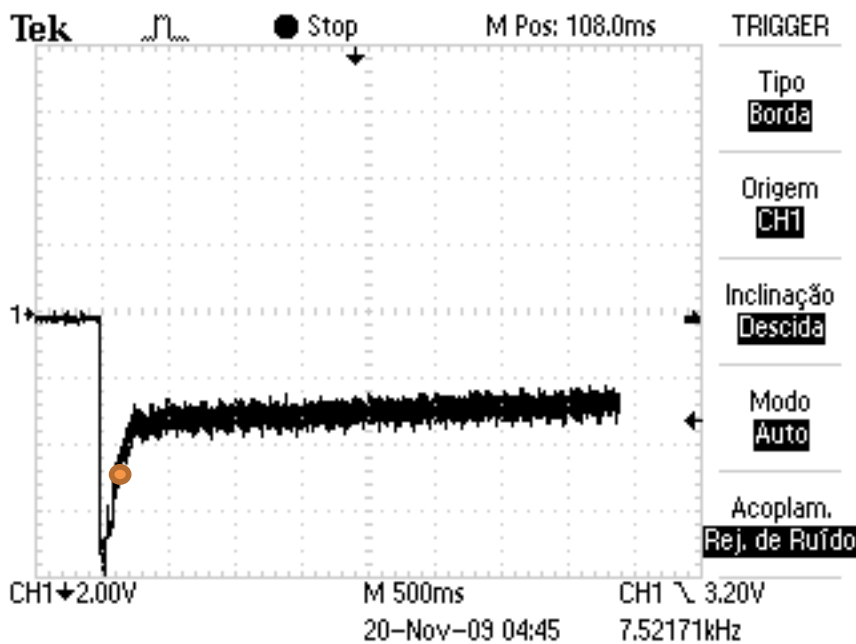


Figura 3.3 – Resposta do motor DC ao degrau de amplitude 4V.

A função de transferência simplificada do motor DC é apresentada [52]:

$$\frac{W(s)}{E(s)} = \frac{K}{R_a J s + R_a f + K^2} \quad (3.9)$$

Sendo:

$E(s)$: a tensão de armadura no plano s ;

R_a : a resistência de armadura;

K : a constante de armadura;

$W(s)$: a velocidade angular do eixo do motor no plano s ;

f : o coeficiente de atrito que atua sobre o eixo;

J : a inércia em relação ao eixo do motor e de sua carga.

Na equação 3.9, aplicando-se $s = 0$, tem-se o ganho do motor em regime permanente que é mostrada na equação 3.10. Pode-se escrever a função de transferência do motor como na equação 3.11. Onde τ é o tempo que a saída leva para atingir 66,6% do valor em regime permanente, como mostra a marcação na figura 3.3:

$$G = \frac{K}{R_a f + K^2} \quad (3.10)$$

$$F(s) = \frac{G}{\tau s + 1} \quad (3.11)$$

A constante de tempo do motor é aproximadamente 0,1 segundos e o ganho do motor é de aproximadamente 0,8, que é obtido da divisão entre o valor da amplitude de degrau, 4V, pelo valor de tensão em regime, 3.2V, e é apresentada na equação 3.12.

$$F(s) = \frac{0,8}{0,1s + 1} \quad (3.12)$$

O ensaio em ambos os motores tem resultados aproximados. O controle de velocidade é feito em malha aberta por PWM. Logo, a identificação da função de transferência serve como documentação para o projeto de um controlador de velocidade em malha fechada.

3.1.2 Considerações sobre Caixa de Redução, Torque e Velocidade de um Motor

A caixa de redução apenas "converte os valores" de torque e velocidade de um motor. A velocidade é inversamente proporcional ao torque. A notação usada para representar a relação entre velocidade e torque é:

$$\text{Velocidade} : \text{Torque} \quad (3.13)$$

Uma caixa com relação 1:10 – divide a velocidade do motor por 10 e multiplica o torque do motor por 10. Uma caixa com relação 10:1 – multiplica a velocidade por 10 e divide o torque por 10, por exemplo.

3.1.3 Considerações sobre Sistema de Unidades e Peso dos Componentes do LOGBOT

O sistema de unidades MKS técnico representa as grandezas usadas em robótica móvel e é descrito na tabela 3.1. A tabela 3.2 apresenta o Sistema Internacional de unidades.

Tabela 3.1 – MKS Técnico ou MK*S.

Grandeza	Unidade	Notação	Dimensão
Comprimento	metro	<i>m</i>	<i>L</i>
Força	quilograma-força	<i>kgf</i>	<i>F</i>
Tempo	segundo	<i>s</i>	<i>T</i>
Temperatura	Kelvin	<i>K</i>	θ

Tabela 3.2 – SI – Sistema Internacional de Unidades.

Grandeza	Unidade	Notação	Dimensão
Comprimento	metro	<i>m</i>	<i>L</i>
Massa	quilograma	<i>kg</i>	<i>M</i>
Tempo	segundo	<i>s</i>	<i>T</i>
Força	Newton	<i>N</i>	MLT^2
Energia	Joule	<i>J</i>	ML^2T^{-2}
Potência	Watt	<i>W</i>	ML^2T^{-3}
Pressão	Pascal	<i>Pa</i>	$ML^{-1}T^{-2}$
Frequência	Hertz	<i>Hz</i>	T^{-1}

A tabela 3.3 mostra o peso de cada item que compõe o robô LOGBOT, onde é notável que os motores e a bateria de chumbo-ácido são responsáveis por 57% do peso do corpo rígido do robô.

Tabela 3.3 – Peso Total do robô LOGBOT em cargas útil.

Dispositivo	Peso Unitário(kg)	Quantidade	Peso Final (Kg)	%
Motores	1.5 kg	2	3,0 kg	32,00
Folha Base de Alumínio 4mm	2,0 kg	1	2,0 kg	21,23
Barras de Ferro 1/8 (Frame)	0.25 kg	4	1,0 kg	10.62
Placas de Acrílico	0.15 kg	2	0.3 kg	3.21
Bateria Chumbo-Ácido 12V / 9Ah	2.4 kg	1	2.4 kg	25.51
Bateria NiMH 12V / 2.3Ah	0.2 kg	1	0.2 kg	2.12
Placas com Circuitos Eletrônicos	0.25 kg	2	0.5 kg	5.31
Peso Total do Robô LOGBOT			9.4 kg	100%

3.1.4 Controle do Motor DC Através de Modulação por Largura de Pulso

A modulação por largura de pulso (PWM – Pulse Width Modulation) é uma técnica comum para controle de velocidade, capaz de contornar o problema de baixo desempenho de partida de um motor, que tende a saltar, quase imediatamente, de uma posição estacionária para mais que a metade da velocidade nominal ao ser alimentado.

Em SUNADA, [13] mostra que - uma analogia a ser feita com o PWM é com o andar de bicicleta, na qual se deve pedalar (exercer energia) e então descansar (relaxar) usando-se a inércia para movimentar-se a frente. À medida que a velocidade é reduzida (devido à resistência do vento, ao atrito, a forma da estrada) pedala-se para acelerar e então se descansa outra vez. O “ciclo de trabalho” e a relação do tempo de pedalada pelo tempo total (pedalada + tempo de descanso). Um “ciclo de trabalho” de 100% significa que pedalasse todo o tempo, e um “ciclo de trabalho” de 50%, que se pedala somente a metade do tempo.

O PWM para o controle da velocidade do motor funciona de maneira muito similar. Em vez de fornecer uma tensão variável ao motor, e fornecido um valor fixo de tensão, que o

faz girar imediatamente. A tensão é então removida e o motor “descansa”. Continuando este ciclo de ligar/desligar a tensão com um “ciclo de trabalho” variado, a velocidade do motor pode ser controlada. A figura 3.4 mostra três sinais diferentes de PWM.

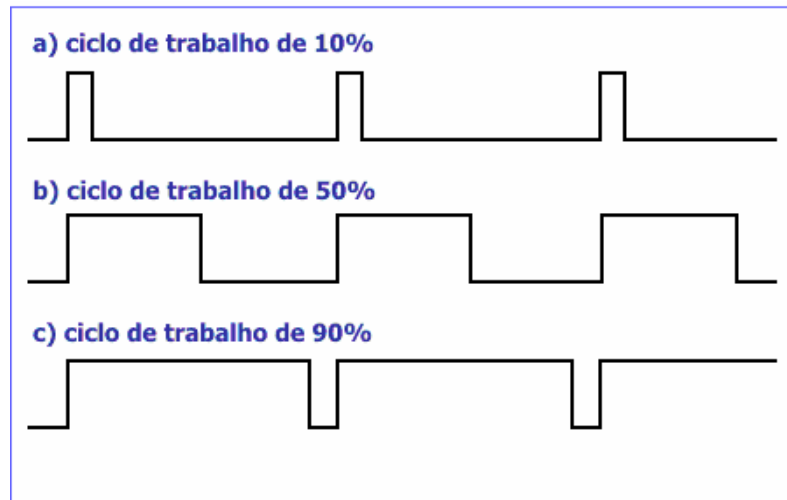


Figura 3.4 – Largura das Modulações PWM.

A figura 3.4(a) mostra uma saída PWM com um “ciclo de trabalho” de 10%. Isto é, o sinal é ligado para 10% do período e desligado os outros 90%. As figuras 3.4(b) e 3.4(c) mostram saídas de PWM em “ciclos de trabalho” de 50% e de 90%, respectivamente. Estas três saídas de PWM codificam três valores diferentes de sinal analógico, em 10%, em 50%, e em 90% da potencia total. Por exemplo, com o sinal a 12V por 50% do tempo, a tensão media é de 6V, então o motor gira a metade da sua velocidade máxima.

As melhores condições de chaveamento são quando a frequência de comutação é muito mais elevada do que a dinâmica do motor. O motor deveria “pensar” que esta sendo alimentado por uma tensão C.C. verdadeira, ou seja, deve-se ligar e desligar o motor, rápido o suficiente, para que ele não “perceba” as mudanças. Devido as diferentes referencias existentes na bibliografia, a frequência deve ser pelo menos 5 vezes mais elevada que a velocidade de rotação do motor como mostra.

$$f_{PWM} \geq 5 \times f_{MOTOR} \quad (3.14)$$

Os motores do LOGBOT têm velocidade angular de 115 RPM, e conforme a equação 3.6, a frequência destes motores é de 1.91Hz. A frequência PWM deve ser mais elevada que $5 \times 1.91 \text{ Hz} = 9.5 \text{ Hz}$. Em uma abordagem teórica a frequência de comutação deve ser muito mais elevada que $\frac{1}{T_a}$, onde $T_a = \frac{L}{R}$ é a constante de tempo elétrica do motor, L é a indutância e R é a resistência interna do motor. Para o motor D.C. do LOGBOT (21.6W, 12V), $L = 52 \text{ mH}$ e $R = 0.5$ a frequência do motor é vista na equação 3.15. Isto significa, outra vez, que a frequência PWM deve ser muito mais elevada do que a frequência. Geralmente uma frequência ao redor de 10 kHz a 20 kHz é usada. Isso evita também o ruído do motor dentro da escala auditiva. A frequência PWM identificada para os motores do LOGBOT é mostrada na equação 3.16.

$$\frac{1}{T_a} = \frac{R}{L} = \frac{0,5}{0,052} = 9,61\text{Hz} \quad (3.15)$$

$$f_{PWM\text{LOGBOT}} = 12\text{kHz} \quad (3.16)$$

3.2 Seleção do Material para Construção da Estrutura

Em robótica móvel o material que compõe a estrutura deve ser escolhido de acordo com a aplicação, de maneira que proteja a parte interna do robô, e que suporte as particularidades físicas do ambiente de atuação. Como exemplo, um robô que tenha missão de atuar em um ambiente extremamente quente ou frio, não deve ter maior parte da estrutura confeccionada com metais de baixo calor específico, como alumínio e ferro, cuja temperatura varia de forma rápida, conforme o comportamento térmico do ambiente.

Os robôs empregados em missões espaciais como o *Lunokhod* visto na figura 2.4(a), tem sua estrutura constituída em fibra de carbono, para que suporte variações térmicas e pressões de outros corpos oriundos do espaço [5]. O robô ambiental híbrido desenvolvido pela Petrobrás, e visto na figura 3.4(b) tem sua estrutura constituída em fibra de vidro, com

grandes rodas de baixa pressão, que o permitem andar em virtualmente qualquer terreno, inclusive subir em dutos metálicos.

Em robôs móveis usados no meio acadêmico e em aplicações terrestres, o acrílico e o alumínio são os materiais comumente usados para confeccionar uma estrutura. O alumínio é um metal forte, leve, resistente a corrosões, e de baixo custo. A manipulação do alumínio para construir a estrutura é fácil, e requer ferramentas comuns. Isto possibilita construir estruturas robustas e com formatos visualmente agradáveis. A resistência a ferrugem faz com que o alumínio tenha longa vida útil.

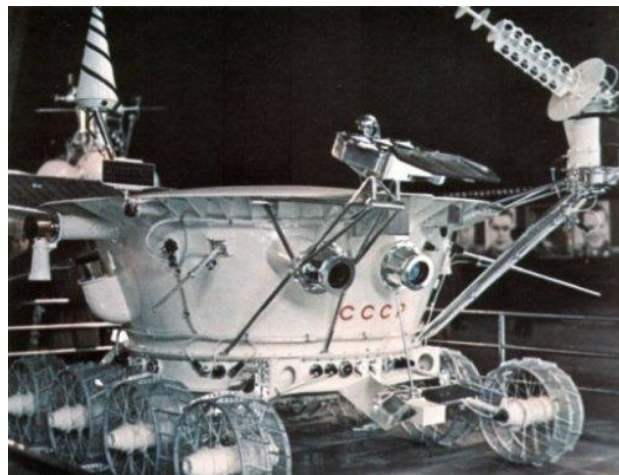


Figura 3.4(a) - ROV Lunokhod I (URSS, 1970)



Figura 3.4(b) - ROV Chico Mendes da Petrobras (Brasil, 2007)

A estrutura base do LOGBOT é constituída em alumínio naval, ou alumínio 5052, como conhecido na indústria naval [12]. Este alumínio é de alta resistência a corrosões, e

escolhido devido às particularidades físicas da região amazônica por apresentar alta umidade e temperatura. Estes fatores diminuem drasticamente a vida útil de um metal, através do processo de corrosão. A espessura da folha de alumínio em 4 mm suporta pressões de até 50Kg sem sofrer avarias. A estrutura foi construída em uma oficina de metais e embarcações, como visto na figura 3.5.



Figura 3.5 – Estrutura base em alumínio naval do robô LOGBOT

A estrutura se mostrou bastante resistente e robusta para a fixação dos motores na lateral frontal, e da roda livre (i.e. *Caster Wheel*) na parte central dianteira. A estrutura tem dimensão de 30cm x 40cm, e as hastes para fixação dos motores tem dimensão de 10cm x 15cm. A modelagem da estrutura foi feita no programa *Google Sketchup*, vista na figura 3.6.

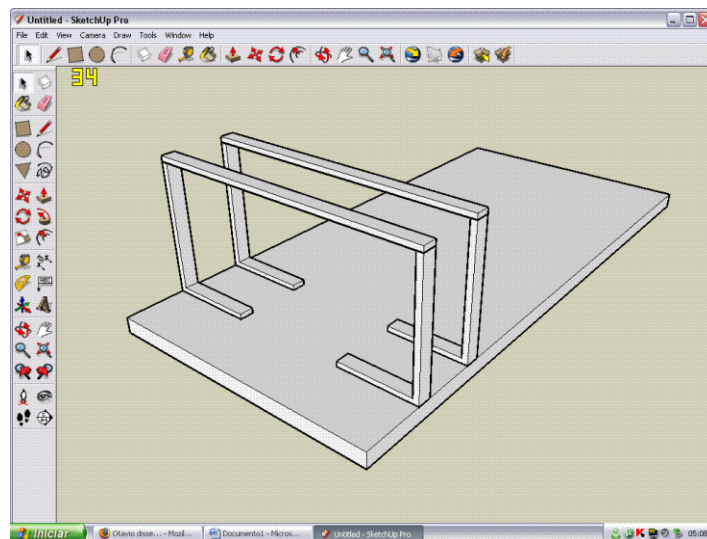


Figura 3.6 – Modelagem do robô LOGBOT feita no programa *Google Sketchup*.

3.3 Tipo do Movimento de Tração do Robô

O modelo e a concepção mecânica de um robô móvel são definidos a partir do modo de tração que o robô usa para se locomover [15]. O modo de tração omnidirecional é comumente usado para atuar em ambientes desconhecidos, como terrenos hostis e missões espaciais. O custo e complexidade para construção de um robô de tração omnidirecional são altos, e este modo de tração não é abordado neste trabalho.

O modo de tração diferencial é comumente usado para atuar em ambientes conhecidos ou semi-conhecidos, como terrenos planos com ou sem inclinações e com obstáculos fixos ou móveis. O custo e complexidade para construção de um robô de tração diferencial são baixos, e este modo de tração é o usado no estudo deste trabalho.

No geral um robô móvel de tração diferencial é composto por uma base estrutural, dois motores (um para cada lateral) com eixos fixos em rodas, denominadas de rodas de tração (*fixed wheels*), e uma roda livre, denominada como roda de suporte da estrutura (*castor wheel*). A disposição das rodas é da seguinte maneira: As rodas de tração estão fixas nas laterais dianteiras do robô, e a roda de suporte está fixa na parte central traseira do robô. O modo de tração diferencial é visto na figura 3.7 [15].

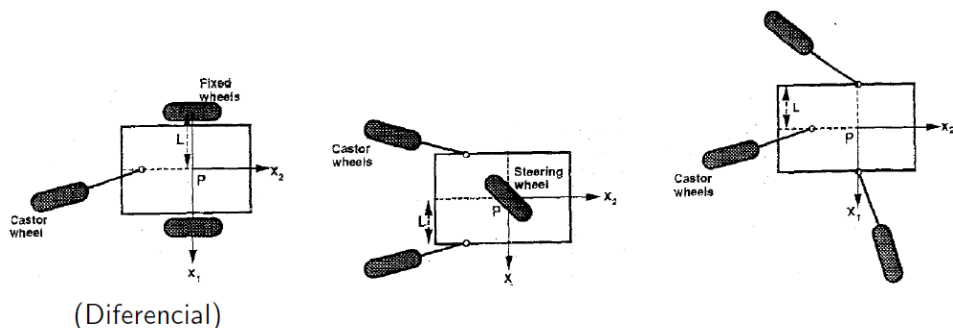


Figura 3.7 – Tração de movimento diferencial para um Robô Móvel.

Os robôs de tração omnidirecional são sistemas do tipo holonômico. E os robôs de tração diferencial são sistemas do tipo não holonômico. Estas propriedades são relacionadas à cinemática e ao controle de um robô móvel.

3.4 Fonte de Energia Embarcada - Baterias

Uma bateria é um dispositivo capaz de armazenar energia química e liberá-la na forma de energia elétrica. As características de uma bateria são: possibilidade de recarga, densidade de energia (máxima quantidade de energia por unidade de massa ou volume), quantidade de energia armazenada na célula (geralmente em A.h, ou mA.h), tensão de operação (em função dos elementos químicos usados), resistência interna, taxa de descarga (varia com a resistência interna), “shelf life”, ou seja, a medida de quão rapidamente a bateria perde a carga sem ser usada, e dependência de temperatura [12]. A tabela 3.4 apresenta algumas características das baterias de: chumbo-ácido, Lítio-Ion, Lítio-Ion Polímero, níquel-cádmio e Hidreto de Níquel-Metal [12].

A análise de todas essas características é importante para a escolha do tipo e o dimensionamento da bateria a ser usada no projeto. Dos principais tipos de baterias comerciais adequadas para esta aplicação (Níquel – Cádmio, Hidreto de Níquel Metal e Chumbo Ácido), as de NiCd (Níquel – Cádmio), são indicadas para a maioria dos usos devido a sua diversidade de tamanhos, sua baixa resistência interna, sua selagem (não havendo risco de derramamento de material contaminante no interior do robô) e são relativamente baratas em relação as NiMH (que apresentam melhores características), e mais leves do que as de chumbo-ácido. Em contrapartida, estas baterias têm restrições técnicas para operação em grupo, apresentando uma baixa densidade de energia (comparável com as de chumbo ácido). Além disto, as baterias de NiCd possuem um “efeito de memória”, isto é, se forem recarregadas sem estarem completamente descarregadas, a nova carga não será total. Por exemplo, se uma bateria com 50% da carga for recarregada, a nova carga adquirida será de 50%. Por possuir baixa resistência interna, as baterias de NiCd possuem valores elevados de corrente de curto circuito (corrente de descarga), levando em determinadas situações a causar danos ao isolamento da bateria e até incêndio.

Tabela 3.3 – Características de Baterias Diversas.

Tipo de Bateria	NiCd	NiMh	Li-Ion	Li-Íon Polímero	Chumbo-Ácido
Densidade de Energia (Wh/kg)	45-80	60-120	110-160	100-130	30-50
Resistência Interna (miliOhm)	100-200 Pack 6V *(1)	200-300 Pack 6V *(1)	150-250 Pack 7,2V *(1)	200-300 Pack 7,2V *(1)	<100 Pack 12V *(1)
Ciclo de Vida (80% da capacidade inicial)	1500 *(2)	500-1000 *(2)(3)	500-1000 *(3)	300-500	200-300 *(2)
Tempo para Carga Rápida	1 hora	2 a 4 hs	2 a 4 hs	2 a 4 hs	8 a 16 hs
Tolerância para Sobrecarga	Moderada	Baixa	Muito Baixa	Baixa	Alta
Auto-Descarga Mensal (na temperatura ambiente)	20% *(4)	30% *(4)	10% *(5)	10% *(5)	5%
Tensão da Célula	1,25V *(6)	1,25V *(6)	3,6V	3,6V	2V
Corrente de Carga - Pico	20C	5C	>2C	>2C	5C - *(7)
- Melhor Resultado	1C	0,5C	1C	1C	0.2C
Temperatura de Operação (somente descarga) *(8)	-40 a 60 °C	-20 a 60 °C	-20 a 60 °C	0 a 60 °C	-20 a 60 °C
Manutenção	30 a 60 dias	60 a 90 dias	Não é necessário	Não é necessário	3 a 6 meses *(9)
Comparação de Custo Pack 7,2V – U.S.A. *(10)	\$50	\$60	\$100	\$100	\$25
Custo por ciclos *(11)	\$0,04	\$0,12	\$0,14	\$0,29	\$0,10
Usada comercialmente desde	1950	1990	1991	1999	1970

(01) A resistência interna de uma bateria varia com a capacidade da célula, tipo de proteção e número de células. Os circuitos de proteção para Li-Íon e Li-Íon Polímero adiciona 100 mili Ohms de resistência.

- (02) O ciclo de vida é baseado na consideração que a bateria recebe o ciclo adequado de manutenção. A falha na aplicação de ciclos profundos de descarga pode reduzir a vida útil por três vezes.
- (03) O ciclo de vida útil é baseado na profundidade da descarga. Descargas curtas permitem ciclos de vida mais longos.
- (04) A descarga é imediatamente maior após a carga. A bateria de NiCd descarrega aproximadamente 10% nas primeiras 24 horas e após descarrega 10% cada 30 dias. A autodescarga aumenta com a elevação da temperatura.
- (05) Circuitos internos de proteção tipicamente consomem 3% da energia armazenada por mês.
- (06) 1,25V é a tensão de célula sem carga. 1,2V é a tensão mais comum.
- (07) Capaz de altas correntes pulsadas.
- (08) Aplicada apenas à descarga; a temperatura de carga é mais restrita.
- (09) A manutenção pode ser na forma de carga de equalização ou pico.
- (10) Custo das baterias para aplicações portáteis.
- (11) Derivado do preço da bateria dividido pelo número de ciclos.

Uma boa característica da bateria de lítio é a de manter praticamente constante o valor da tensão à medida que diminui sua capacidade (carga) com o tempo. No caso das baterias Alcalinas, Chumbo-Ácido e NiCd, o valor da tensão é mais afetado em relação ao estado da carga, diminuindo significativamente com o tempo.

Tabela 3.4 – Características de Baterias Diversas.

Tipo de Bateria	Tempo de Retenção de Carga	Curva de Descarga
Lítio	Alto tempo de retenção de carga em vazio (\pm 10anos)	Plana
NiMH	Médio tempo de retenção em vazio	Inclinada
Chumbo-Ácido	Médio tempo de retenção em vazio	Inclinada
NiCd	Médio tempo de retenção em vazio	Inclinada

3.4.1 Reguladores de Tensão

Os regulares de tensão são dispositivos que servem para fornecer uma tensão constante para o circuito, mesmo que a tensão de entrada varie sobre uma ampla faixa, compensando o efeito da redução da tensão fornecida pela bateria à medida que esta se descarrega [12].

Um regulador linear é um dispositivo constituído normalmente por 3 terminais: alimentação (cujo valor deve ser superior em alguns volts a tensão de saída desejada), terra e tensão de saída. Alimentando o regulador linear com uma tensão dentro da faixa de operação, a tensão de saída permanecerá sempre constante. Um regulador linear como o LM78S05, mantém a tensão de saída em 5V para tensões de entrada dentro da faixa de 7 a 35V, e drena correntes de até 2A. Se a tensão de entrada for menor que 7V, a tensão de saída não se mantém regulada. Um outro regulador, o LM2940CT-5.0 opera com tensões de entrada na faixa entre 5.5V e 26V, mantendo a saída regulada em 5V. Sua vantagem em relação ao LM7805 é a necessidade de precisar de apenas um acréscimo de 0,5V na tensão de entrada em relação a saída para operar satisfatoriamente. Esta característica acarreta uma menor dissipação de potência pelo regulador.

3.4.2 Proteção de Circuitos, Isolamento e Redução de Ruídos

Uma vez que os motores e determinados sensores (ultra-som) produzem muita interferência, torna-se necessário no projeto do robô o isolamento da parte lógica em relação aos sensores e circuitos de potência.

Os motores usados nos robôs móveis causam picos de tensão quando passam por uma seção do comutador. Para evitar esse efeito indesejável, uma solução é a colocação da bateria entre o motor e a parte lógica, bastando somente regular a tensão para alimentar os circuitos lógicos. Caso isto não seja suficiente para evitar problemas de funcionamento no circuito, outra solução é a colocação de um capacitor em paralelo com o circuito lógico e um diodo

entre a bateria e este circuito. Outra medida, considerada mais sofisticada e eficiente é a utilização de fontes de alimentação distintas. Uma última alternativa é a utilização de um isolador ótico, o que permitiria um verdadeiro isolamento entre o circuito lógico e os motores. Referências de chips comerciais para esta finalidade são: 4N25, 4N28 e 4N30.

3.4.3 Dimensionamento das Baterias para Fornecimento de Energia ao Robô LOGBOT

Os picos de tensão, gerados pelos MOSFETS ou pontes do tipo monolítico para chaveamento de corrente para os motores, provocam ruídos interferentes no módulo de controle e comunicação por radio frequência – RF. Como forma de se evitar estas interferências, foi adotada a configuração com fontes de alimentação independentes.

Para alimentação dos motores de corrente contínua, optou-se pela bateria do tipo Chumbo-Ácido, 12VDC/7A.h, por ter baixa resistência interna e alta corrente de descarga. Um dos requisitos do planejamento da trajetória é evitar mudanças de velocidade em degrau que possam gerar um aumento súbito da demanda de corrente do motor e perda de precisão do percurso devido a patinação das rodas. Em caso de travamento do eixo a bateria deverá ser capaz de fornecer altas correntes em pequenos intervalos de tempo.

Para alimentação do módulo de eletrônica vista no capítulo 7 desta dissertação, optou-se pelo uso de uma bateria de Níquel Metal-Hidreto – NiMH de 12V/2.3A.h, por resistir a um número maior de cargas/descargas em sua vida útil do que as de NiCd, e por ter tempo de recarga inferior e maior resistência a variações de temperatura.

Com base nos dados anteriores e supondo que o robô pesa 9,4Kg e desenvolve uma velocidade constante de 0,4 m/s, considerando a gravidade em 9,8m/s e o coeficiente de atrito μ é de 0,5. A potência instantânea exigida é:

$$P_i = \mu \times m \times g \times v = 0,5 \times 9,4 \times 9,8 \times 0,4 = 18,424 \text{Watts} \quad (3.17)$$

Onde:

$$Watts = \frac{Joules}{seg} = \frac{Volts \times Coulombs}{seg} = \frac{Volts \times Amps \times seg}{seg} = Volts \times Amps \quad (3.18)$$

Entretanto se tem:

$$Volts \times Amps \times horas = Watts \times horas \quad (3.19)$$

A expectativa de vida da bateria é dada por:

$$T_{bateria} = \frac{Volts \times AmperHora}{Potência} \quad (3.20)$$

A expectativa de vida para a bateria de Chumbo-Ácido do robô LOGBOT é:

$$T_{bateria} = \frac{12 \times 7}{18,242} = 4,6 \text{ horas}$$

Existem imprecisões nesta análise porque não é desejável que o robô se movimente em velocidade constante durante todo o tempo de vida da bateria e a tensão da bateria tende a diminuir com o tempo. A tabela 3.5 apresenta as fontes, os reguladores e os valores estimados de carga e sua autonomia.

Tabela 3.5 – Fontes, reguladores, cargas e autonomia para o módulo de alimentação.

Fonte de Energia	Regulador	Carga	Autonomia
Bateria 12VDC/7Ah Chumbo-Ácido	-	Dois motores DC (1.8A cada)	276 minutos
Bateria 12VDC/2.3Ah NiMH	LM78S05 5VDC	Módulo RF (60mA), Microcontrolador (25mA), Vetor de Sonares (40mA), Módulo Driver Ponte-H (200mA)	420 minutos

3.5 Abordagem em Camadas do Projeto de um Robô Móvel

O desenvolvimento de um robô móvel envolve diversas tecnologias, logo isto requer um plano de integração destas tecnologias, de maneira que não haja inconsistências no funcionamento do sistema que compõe o robô. Uma abordagem em camadas deste planejamento foi desenvolvida para simplificar e documentar os passos de montagem e

integração das tecnologias que compõe o robô. O projeto foi dividido em camadas, conforme mostrado na Figura 3.8.

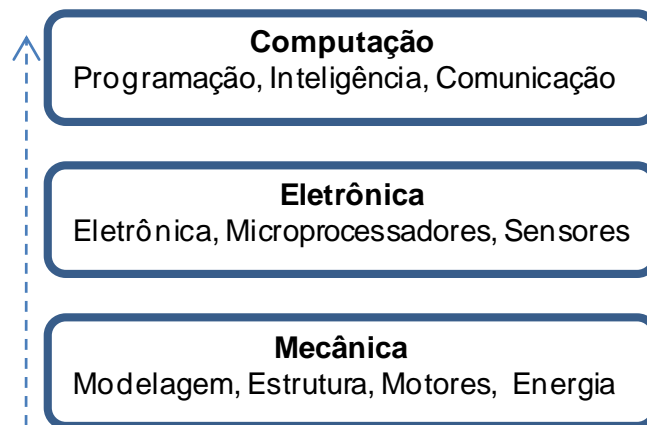


Figura 3.8 – Projeto de um Robô Móvel pela Abordagem em Camadas

O desenvolvimento baseado em camadas facilita o raciocínio e a organização do projeto. As camadas podem ser desenvolvidas em paralelo para depois haver a integração dos sistemas mecânico, eletrônico e computacional, que é feita na ordem indicada pela seta.

Na camada mecânica, a estrutura mecânica do LOGBOT é feita de alumínio naval e possui dimensões de 30cm x 40cm. Ele conta com duas rodas independentes, à esquerda e direita de sua parte dianteira, e uma terceira roda de sustentação na parte central traseira. Possui dois motores DC de 115RPM, com tensão de 12V e torque de 44Kgf.cm, Os motores são alimentados por uma bateria DC recarregável. Uma segunda bateria é usada para alimentar exclusivamente a camada eletrônica, que assim fica protegida da interferência de eventuais ruídos gerados pelos motores que movimentam o robô.

Nas camadas de eletrônica e computação, são desenvolvidos os módulos de sensoriamento/controle (aquisição de dados dos sensores), e de acionamento dos motores. É nesta camada que ocorre a integração de todos os módulos que compõe o sistema, através do computador embarcado, que é um Microcontrolador. Na figura 3.9 é visto o diagrama básico da integração das tecnologias que compõe o robô LOGBOT.

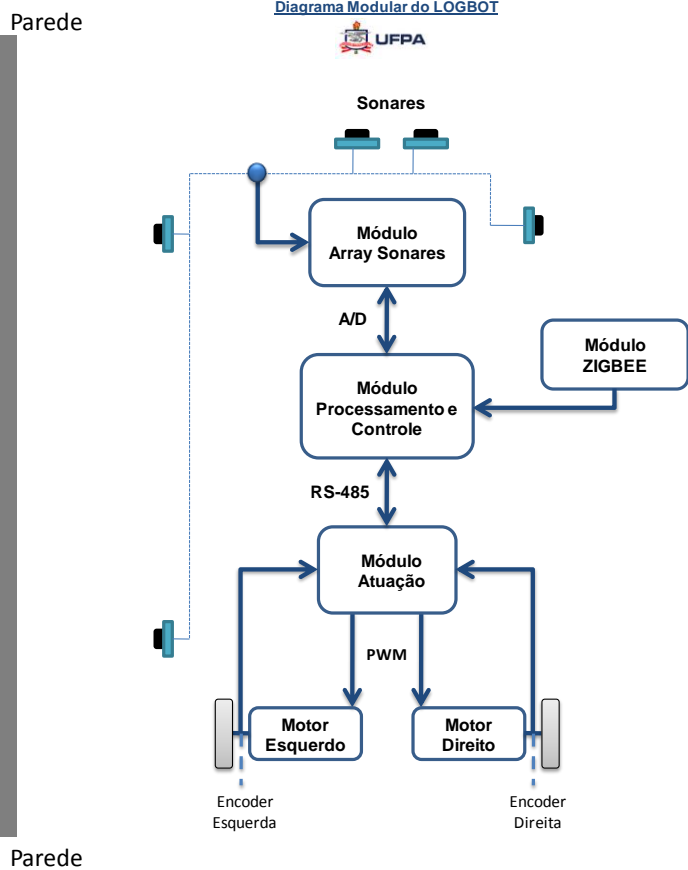


Figura 3.9 – Diagrama Modular do Robô LOGBOT

3.6 Conclusões

Neste capítulo foram apresentados os conceitos vitais para o projeto de um robô móvel diferencial, cálculos para validação dos motores a serem usados, a escolha do material que constitui a base estrutural, a classificação do tipo de movimento do robô, e seleção da fonte de energia embarcada, assim como seu tempo de consumo.

No capítulo também foram abordados as considerações sobre a aceleração, força e momento de um robô móvel diferencial terrestre, cujo modelo cinemático é denominado de não-holonômico. E a abordagem em camadas como metodologia usada no projeto e organização dos módulos que compõem o sistema do robô móvel LOGBOT.

Todos esses conceitos são de grande utilidade no projeto de robôs móveis diferenciais, e serão recorrentemente usados nos próximos capítulos desta dissertação.

Capítulo 4: Modelagem de Robôs Móveis Diferenciais

Os robôs móveis podem ser classificados de diversas maneiras que incluem a forma de locomoção, os tipos de tarefas executadas e o seu grau de autonomia. Os robôs móveis são classificados principalmente pela forma como se locomovem em seu ambiente.

Um grande número de características influencia a escolha do sistema de locomoção a ser utilizado em uma tarefa. Entre elas pode-se citar: manobrabilidade, controlabilidade, estabilidade, eficiência e tração [10]. O sistema de locomoção de um robô móvel pode ser representado por modelos cinemáticos e dinâmicos. Apesar de o modelo dinâmico ser importante para o controle de alguns tipos de robôs, principalmente os robôs aéreos e aquáticos, este capítulo apresentará somente sua modelagem cinemática. A justificativa para este fato é que grande parte dos sistemas de controle de movimento é baseado apenas na cinemática dos robôs.

A definição dos conceitos do modelo cinemático e dinâmico de um robô móvel se faz importante para a compreensão dos próximos tópicos abordados nesta dissertação:

- **Locomoção:** É o processo que causa o movimento do robô móvel. Inicialmente sai do estado de eminência e entra em movimento, através de forças aplicadas ao robô (motores).
- **Modelo cinemático:** É o estudo da matemática do movimento, sem considerar as forças que causam o movimento. Lida com as relações geométricas que governam o sistema (Compõem, controlam...). Lida com a relação entre os parâmetros de controle e o comportamento do sistema no espaço de estados.
- **Modelo dinâmico:** É o estudo do movimento que o robô executa. As forças aplicadas são modeladas, como energia, massa, torques e velocidades associadas aos movimentos (motores).

4.1 Modelo Cinemático de um Robô Móvel Não-Holonômico

Um robô móvel pode ser especificado em seu ambiente de atuação (espaço de trabalho) por seu vetor de configurações $q(t)$. A dimensão de $q(t)$ é o número de graus de liberdade (*degrees of freedom – DOF*) do robô e representa o número mínimo de variáveis independentes que, juntamente com a sua geometria, são necessárias para especificar completamente o robô.

Por exemplo, um robô terrestre em forma de carro possui dois graus de liberdade – posição e orientação, e pode ser especificado pela configuração $q(t) = [x(t), y(t), \theta(t)]^T$, onde x e y especificam a posição cartesiana do robô em relação ao um referencial fixo $\{W\}$ no ambiente e θ especifica a sua orientação, como mostra a equação 4.1.

$$q = \begin{bmatrix} \text{Posição} \\ \text{-----} \\ \text{Orientação} \end{bmatrix} \Rightarrow q = \begin{bmatrix} x(t) \\ y(t) \\ \text{---} \\ \theta(t) \end{bmatrix} \quad (4.1)$$

Em robótica móvel, o modelo cinemático relaciona as velocidades dos atuadores (rodas, hélices, turbinas) às velocidades de um referencial fixo no robô, geralmente com origem em seu centro de massa. A posição do robô pode ser encontrada pela integração de suas velocidades, como mostra as equações 4.7 e 4.8. Em geral, a cinemática inversa de robôs móveis é muito difícil de ser calculada globalmente [10]. Técnicas locais serão tratadas na próxima seção. O modelo cinemático mais simples para um robô móvel é dado por:

$$\dot{q}(t) = G(q)u(t) \quad (4.2)$$

Onde $\dot{q}(t)$ é a derivada de $q(t)$, e $u(t)$ é o vetor de controle, cujas componentes são as velocidades de atuação do robô. A matriz $G(t)$ é quadrada quando o vetor de controle possui dimensão igual ao número de graus de liberdade do robô. Porém, para a grande maioria dos

robôs móveis, a matriz $G(q)$ não é quadrada, já que o número de velocidades de atuação é menor que o número de graus de liberdade do sistema.

Isto geralmente é caracterizado por restrições de movimento que não permitem o controle independente de todas as variáveis do vetor de configurações. Desta forma o robô não pode se mover lateralmente, e assim diversas manobras devem ser executadas para aproximá-lo de uma parede ou movimentá-lo em um ambiente estreito, isto é, para controlar sua configuração (x, y, θ) . Restrições deste tipo são conhecidas como restrições não-holonômicas.

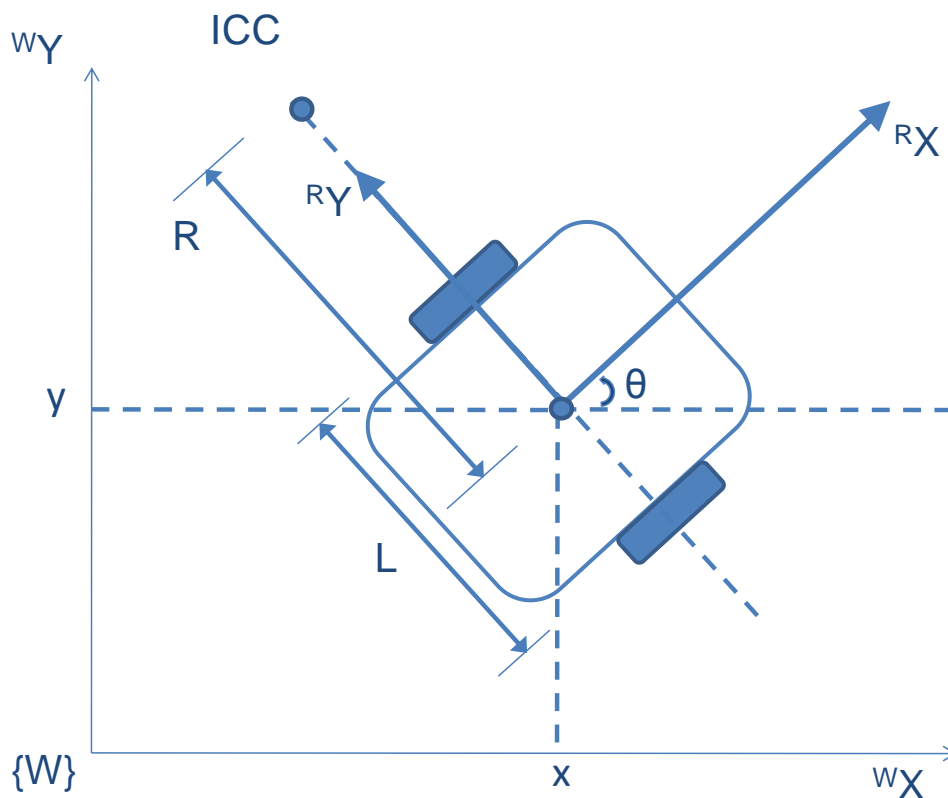


Figura 4.1 – Plano de referência global e local do robô móvel.

O efeito deste tipo de restrição é reduzir o número de velocidades independentes do robô, o que o torna um sistema subatuado [10]. Os robôs com restrições não holonômicas são chamados de robôs não holonômicos. Os robôs que não possuem estas restrições são sistemas (plenamente) atuados e são conhecidos como robôs holonômicos.

O robô móvel diferencial em seu projeto é constituído de duas rodas montadas em um eixo imaginário e controladas por motores independentes, conforme mostra o do seu modelo cinemático na figura 4.1.

O par $\{W_x, W_y\}$ representa o *frame* de coordenadas do ambiente de atuação do robô, $\{W\}$, e $\{R_x, R_y\}$ representa o *frame* de coordenadas locais (próprias) do robô, $\{R\}$ [3].

O modelo cinemático deve relacionar a velocidade das rodas com as velocidades do referencial fixo em seu centro de massa. As rodas do robô só se movem na direção em que estão apontadas, e possuem velocidade zero na direção perpendicular a esta. Desta forma não ocorre derrapagem lateral.

Para que esta condição seja satisfeita, deve existir um ponto em torno do qual cada uma das rodas do robô segue um caminho circular, mesmo que este ponto esteja no infinito (caminho retilíneo). Este ponto é chamado de centro instantâneo de curvatura, ICC (*Instantaneous Center of Curvature*). O ICC sempre fica localizado no ponto do eixo comum das rodas laterais, definido por:

$$ICC = (x - R \sin(\theta), y + R \cos(\theta)) \quad (4.3)$$

Como o robô é um corpo rígido, a roda direita e esquerda devem se mover simultaneamente ao redor do ICC com a mesma velocidade angular ω , que é dada por:

$$\begin{cases} \omega = \frac{v_r}{R+L/2} = \frac{r \cdot \omega_r}{R+L/2} \\ \omega = \frac{v_l}{R-L/2} = \frac{r \cdot \omega_l}{R-L/2} \end{cases} \quad (4.4)$$

Onde v_r e v_l são as velocidades lineares da roda direita e esquerda, respectivamente, ω_r e ω_l são as correspondentes velocidades angulares, r é o raio das rodas, L é a distância entre os seus centros e R é o raio de curvatura instantâneo do caminho do robô móvel.

Resolvendo-se para R e ω as equações em 4.4, e dado que a velocidade linear V do centro de massa do robô é o produto de R e ω , obtém-se:

$$\begin{cases} R = \frac{L}{2} \cdot \frac{(\omega_l + \omega_r)}{(\omega_r - \omega_l)} \\ \omega = \frac{r(\omega_r - \omega_l)}{L} \\ V = \frac{r(\omega_r + \omega_l)}{L} \end{cases} \quad (4.5)$$

Decompondo-se a velocidade linear em suas componentes $x(t)$, $y(t)$ e a velocidade angular do robô $\omega(t)$, onde $\omega(t) = \dot{\theta}(t)$, chega-se às seguintes equações:

$$\begin{cases} \dot{x}(t) = V(t) \cos(\theta(t)) \\ \dot{y}(t) = V(t) \text{sen}(\theta(t)) \\ \dot{\theta}(t) = \omega(t). \end{cases} \quad (4.6)$$

Para os controladores usados neste trabalho, através do uso de microcontroladores, obrigam um sistema discreto na sua formulação. Conseqüentemente é necessária uma discretização da equação 4.6. A equação 4.7 mostra a configuração atual do robô no tempo contínuo, e a equação 4.8 mostra a configuração obtida pela aproximação de Euler do modelo em tempo contínuo.

$$\begin{cases} x(t) = \int_{t_0}^t v(t) \cos[\theta(t)] \\ y(t) = \int_{t_0}^t v(t) \text{sen}[\theta(t)] \\ \theta(t) = \int_{t_0}^t \omega(t) dt \end{cases} \quad (4.7)$$

$$\begin{cases} x(k+1) = x(k) + v(k)T \cos \theta(k) \\ y(k+1) = y(k) + v(k)T \operatorname{sen} \theta(k) \\ \theta(k+1) = \theta(k) + \omega(k)T, \end{cases} \quad (4.8)$$

onde T é o período de amostragem. Substituindo V e ω na equação 4.6, chega-se ao seguinte modelo cinemático em espaço de estados para um robô móvel diferencial terrestre:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} r \cos \theta(t)/2 & r \cos \theta(t)/2 \\ r \operatorname{sen} \theta(t)/2 & r \operatorname{sen} \theta(t)/2 \\ r/L & -r/L \end{bmatrix} \begin{bmatrix} \omega_r(t) \\ \omega_l(t) \end{bmatrix}. \quad (4.9)$$

Apesar de o modelo cinemático completo ser dado em função das velocidades angulares das rodas, muitas vezes parte das equações são “encapsuladas”, e o modelo é simplificado, como mostra a equação 4.10:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \operatorname{sen} \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \quad (4.10)$$

Isso é comum quando os controladores de mais baixo nível do robô se encarregam do cálculo de V e ω em função de ω_r e ω_l . Nas equações 4.6 e 4.9 é descrito a posição e a orientação do sistema de coordenadas do robô em relação ao sistema de coordenadas inercial. Observa-se que neste modelo os ângulos de orientação e rotação da roda orientável não-centrada (roda livre) não são considerados. Especificamente, o sistema é estável no sentido de *Lyapunov* e controlável, entretanto um ponto qualquer não pode ser estabilizado por uma realimentação de estados suave e invariante no tempo [16]. O modelo obtido da sua linearização em torno do estado não é controlável e o sistema não pode ser linearizado por realimentação de estados. É possível, entretanto, obter uma linearização entrada-saída se a saída y considerada for $[x \ \theta]^T$, apenas.

Nas equações 4.9 e 4.10 observa-se que o número de entradas de controle independentes é menor que o número de graus de liberdade devido à existência de uma restrição não-holonômica, que é representada por:

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0 \quad (4.11)$$

A interpretação física da equação 4.11 é que a componente de velocidade perpendicular do robô móvel deve ser nula, e a consequência disso é a dependência entre as velocidades \dot{x} e \dot{y} . A restrição não-holonômica de robôs móveis terrestres é consequência direta da forma em que as rodas são usadas em sua locomoção, como mostra a figura 4.2.



Figura 4.2 – Movimento do tipo Tração Diferencial Não-Holonômico.

Na figura 4.2 é visto o direcionamento diferencial das rodas, através das setas laranja. O robô está a 25 cm de distância da parede a sua esquerda, e não pode se deslocar diretamente na perpendicular devido a sua restrição não-holonômica, isto exige que ele realize balisas até aproximar-se da parede, como um carro que está estacionando em uma garagem.

4.1.1 Considerações físicas sobre restrições Holonômicas e Não-Holonômicas

O termo “restrição” pode estar relacionado a sistemas dinâmicos de diversas formas, notadamente com controle ou modelagem. Relacionado a controle normalmente denota limites nos estados e nas entradas que devem ser satisfeitos ou para o bom funcionamento do mesmo ou devido a peculiaridades do sistema real que não constam na modelagem. Pode também ser de interesse que seja obtida uma lei de controle que garanta determinadas medidas de desempenho e possivelmente estabilidade ou rejeição a distúrbios. Uma lei de controle pode ser obtida considerando conjuntos invariantes de tal modo que se tem certeza de que os requisitos de desempenho e as restrições serão atendidos para instantes de tempo positivos [20]. Tal área é comumente chamada controle sob restrições. A obtenção de leis de controle com base em conjuntos invariantes não é assunto deste trabalho. Porém, com relação à modelagem, assunto deste capítulo, o termo denota a relação entre as coordenadas que o descrevem. Dividem-se em restrições holonômicas e não-holonômicas.

As restrições holonômicas são equações expressas em termos das coordenadas e tempo diretamente, mas incluem diferenciais de tais variáveis. São tais que não podem ser integradas de modo a se obter um conjunto de coordenadas generalizadas independentes. Tem-se que o número de graus de liberdade para sistemas que contém restrições não-holonômicas no seu modelo – dito então sistemas Não-Holonômicos – é inferior ao número de coordenadas generalizadas dependentes [20]. Um exemplo clássico de sistema não-holonômico é o modelo simplificado de uma moeda rolando sem deslizamento no plano. Tal modelo, por ignorar a ação da gravidade e o ângulo entre o plano da moeda e o plano sobre o qual ela rola é o mesmo que descreve uma roda que gira em torno de um eixo horizontal [24], o qual serve de base para a modelagem de robôs móveis dotados de rodas.

A obtenção de leis de controle para sistemas Não-Holonômicos é mais complexa que para sistemas holonômicos. A explicação é o reduzido número de graus de liberdade em relação ao número de estados deste tipo de sistema. Mais especificamente, as condições de

Brockett [26] expressam a impossibilidade de, para um sistema deste tipo e sem deriva, se estabilizar assintoticamente um dado ponto de equilíbrio qualquer através de uma realimentação de estados contínua e invariante no tempo.

4.2 Modelo Cinemático do Robô Móvel LOGBOT

O robô móvel LOGBOT desenvolvido neste trabalho é um robô móvel no formato de um veículo com rodas conforme visto no capítulo 3.

Na figura 4.3 é visto a estrutura cinemática aplicada ao LOGBOT. A estrutura física do robô é feita de alumínio naval com revestimento pneumático, e suas dimensões são de 40cm x 50cm. Possui duas rodas independentes, à esquerda e direita, e uma roda de sustentação traseira tornando-o equilibrado durante seus movimentos.

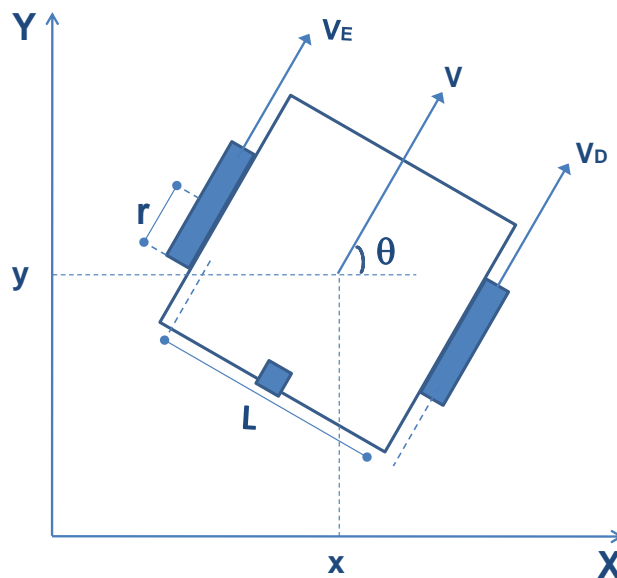


Figura 4.3 – Estrutura Cinemática do robô móvel LOGBOT.

O robô é um sistema com três graus de liberdade (3-DOF) cujas saídas são representadas pela sua posição (x, y) e orientação θ , porém o controle é realizado em dois graus de liberdade (2-DOF) devido à restrição não-holonômica de movimento. Considera-se ainda que:

$$V_1 = r\omega_1 \quad \text{e} \quad V_2 = r\omega_2 \quad (4.12)$$

onde V_i é a velocidade linear da roda i , ω_i é a velocidade angular do respectivo eixo e r é o raio das rodas. Sabe-se ainda que:

$$V = \frac{V_1 + V_2}{2} \quad (4.13)$$

$$\omega = \frac{V_1 - V_2}{L} \quad (4.14)$$

onde V é a velocidade linear do centro do robô, ω é a velocidade angular do robô e L é a distância entre as rodas. Além disso, a velocidade linear V pode ser decomposta ao longo dos eixos X e Y respectivamente como:

$$V_x = V \cos(\theta) \quad \text{e} \quad V_y = V \sin(\theta), \quad (4.15)$$

onde θ é o ângulo mostrado na figura 4.3. Então, um modelo cinemático para o robô LOGBOT pode ser escrito como:

$$\dot{x} = V_x \quad (4.16)$$

$$\dot{y} = V_y$$

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

Substituindo as equações para V_x , V_y e $\dot{\theta} = \omega$ em (4.16) tem-se que:

$$\dot{x} = \frac{V_1 + V_2}{2} \cos(\theta)$$

$$\dot{y} = \frac{V_1 + V_2}{2} \sin(\theta) \quad (4.17)$$

$$\dot{\theta} = \frac{V_1 - V_2}{L}$$

A restrição não-holonômica do sistema Pode ser representada pela seguinte equação:

$$\dot{x} \text{sen}(\theta) - \dot{y} \text{cos}(\theta) = 0 \quad (4.18)$$

A equação 4.18 especifica que a direção tangente a qualquer caminho possível para o robô e um limite de curvatura para este caminho. O modelo completo do robô móvel LOGBOT é mostrado na figura 4.4.

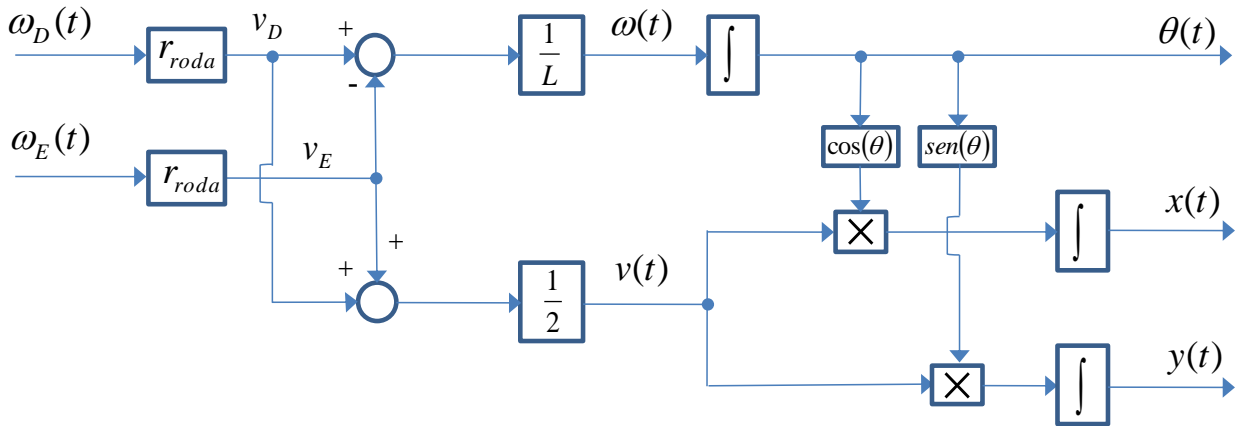


Figura 4.4 – Modelo Cinemático Completo do Robô Móvel LOGBOT.

Na figura 4.4 observa-se no modelo cinemático completo do robô móvel LOGBOT a forte dependência entre os três graus de liberdade do robô causada pela restrição não-holonômica, cujo controlador tem ação somente em dois graus de liberdade do robô, devido este ter somente dois atuadores (motores DC).

4.3 Modelo Dinâmico de um Robô Móvel Não-Holonômico

Uma grande classe de sistemas mecânicos Não-Holonômicos é descrita pela seguinte forma de equações dinâmicas baseadas em *Euler Lagrange* [27]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau + J^T(q)\lambda$$

Enquanto que a restrição não-holonômica é dada por:

$$J(q)\dot{q} = 0$$

Onde q é um vetor n dimensional das variáveis de configuração, $M(q)$ é uma matriz simétrica e positiva $n \times n$, $C(q, \dot{q})$ mostra o vetor n de torque centrípeto e de *Coriolis*, $G(q)$ é

o vetor n de torques gravitacionais, $B(q)$ é a matriz de transformação $n \times r$ ($r < n$), τ é o vetor r dimensional de entradas, e λ os multiplicadores de *Lagrange* das forças de restrição. De acordo com a figura 3.1 a estrutura de um robô móvel diferencial é constituída por dois motores de corrente contínua idênticos, que são os atuadores da roda esquerda e direita (na dianteira), enquanto que a roda traseira serve somente para suporte, a fim de manter a plataforma estável. O vetor de postura do robô é definido como $q = (x, y, \theta)^T$, onde x e y são as coordenadas do ponto ICC – centro do eixo das rodas, e θ é o ângulo de orientação do robô móvel diferencial no sistema de coordenadas inerciais (ambiente de atuação). Uma forma de representar as equações dinâmicas do robô móvel de acordo com a equação (3.9) é assumir que $G(q)$ e $C(q, \dot{q})$ possuem valor zero.

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau^1 \\ \tau^2 \end{bmatrix} + \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \lambda$$

Onde τ_1 e τ_2 são os torques do motor esquerdo e motor direito, m e I representam a massa e a inércia do robô respectivamente. R é o raio das rodas, e L é a distância entre as rodas com os motores.

Capítulo 5: Controle de Robôs Móveis

Os controladores para robôs móveis geralmente têm objetivo de garantir que o robô siga uma trajetória pré-especificada. Neste sentido, a trajetória é uma sequência contínua de configurações parametrizadas pelo tempo, conforme visto na figura 5.1. O controlador deve alcançar determinadas especificações de desempenho e características desejáveis para a movimentação do robô, e deve ser projetado para executar corretamente a sequência planejada de movimentos na presença de qualquer erro.

As estratégias de controle são baseadas nas forças de tração que provocam o movimento, assim como a manutenção do percurso no caminho desejado, sem a ocorrência de desvio ou derrapagem. O controle do movimento da tração tem algumas características desejáveis como:

1. Manter a aceleração e a desaceleração a mais rápida e precisa possível;
2. Manter a precisão no percurso do caminho desejado;
3. Manter a estabilidade do robô durante suas manobras;
4. Prevenir o robô da ocorrência de derrapagem ou escorregamento.

Os controladores do tipo multivariável com uma realimentação de estados suave não estabiliza um sistema com restrições não-holonômicas. Entretanto, é provado que se o sistema for controlado por partes a estabilidade pode ser alcançada [16] [28]. Este tipo de controlador consiste no projeto de sistemas de controle independentes para algumas variáveis. Neste capítulo será mostrado o projeto de um controlador para robôs autônomos não-holonômicos baseado neste princípio. Além disso, transformações de variáveis serão feitas para facilitar o controle, tornando simples o projeto dos controladores. Assume-se que as trajetórias a serem seguidas pelo robô são pré-especificadas por um nível de planejamento executado em malha aberta.

Para um robô holonômico mais simples, onde $G(q)$ é a matriz identidade, onde:

$$\dot{q}(t) = u(t) \quad (5.1)$$

Nesse caso, aplicando-se a lei de controle $u(t) = r(t)$, onde,

$$r(t) = \dot{q}^*(t) + K_p(q^*(t) - q(t)) \quad (5.2)$$

$q^*(t)$ Representa a trajetória a ser seguida e K_p é uma matriz diagonal positiva, é possível garantir que a configuração do robô convergirá assintoticamente para a trajetória especificada, independentemente das condições iniciais. Para os robôs totalmente atuados com diferentes modelos cinemáticos, as variáveis de controle na equação (5.1) podem ser calculadas diretamente por $u(t) = G^{-1}(q)r(t)$.

Para um robô não-holonômico, onde G não pode ser invertido, nem todos os graus de liberdade do robô podem ser controlados diretamente. Pode-se mostrar que não existe nenhum controlador suave e invariante no tempo capaz de estabilizar um robô subatuado em uma configuração específica. Entretanto, o controle de um sistema não-holonômico para seguir trajetórias predeterminadas é possível por meio da linearização do modelo em torno do seu estado atual [16].

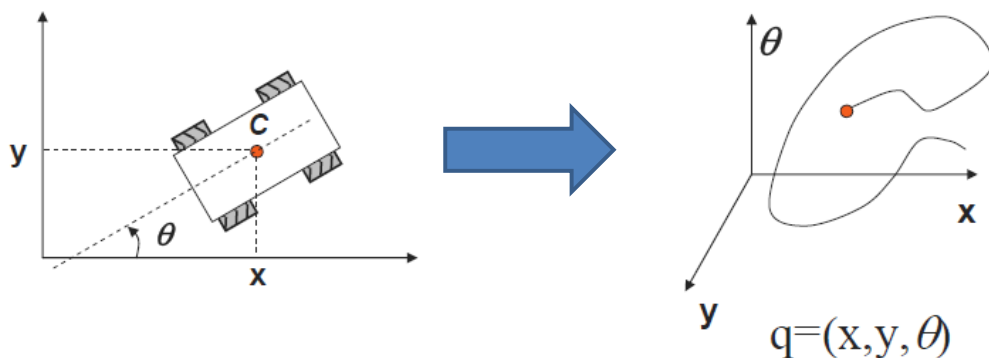


Figura 5.1 – Representação do robô como o ponto ‘q’ em seu espaço de configuração.

Neste capítulo será abordado o uso de controladores para robôs móveis, com ênfase no método experimental de *Kuipers* para identificação de parâmetros do controlador [25]. E o uso de sua combinação com agentes de *software* como forma de aumentar a inteligência do robô e aumentar seu desempenho nas tarefas no desvio de obstáculos e localização.

5.2 Controle de Robôs Móveis Não-Holonômicos

Os sistemas de controle de trajetórias tradicionalmente usados para controlar robôs móveis podem ser representados por um diagrama de blocos semelhante ao da figura 5.2. Esta estratégia de controle conhecida como “cascata”, onde a saída de um controlador é a referência para outro [16] [23].

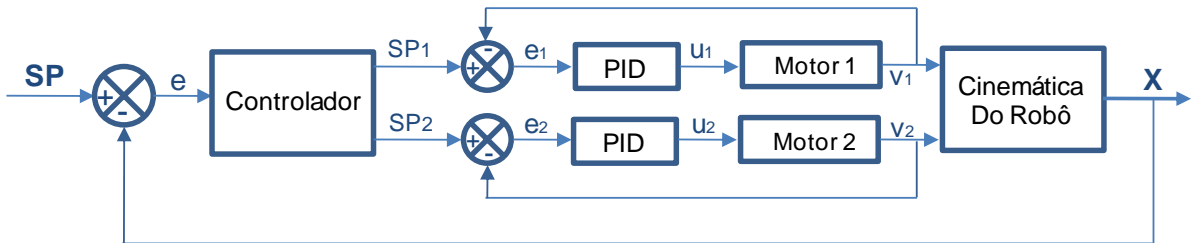


Figura 5.2 – Sistema de Controle Tradicional para um Robô Móvel.

Os controladores internos (PID – Proporcional Integral Derivativo) são chamados de “secundários” ou “escravos” e o externo é conhecido como “primário” ou “mestre”. De uma maneira geral, este tipo de estratégia é usada para melhorar o desempenho do sistema por meio de uma rápida correção dos distúrbios na malha interna. O aumento do desempenho pode ser muito grande se o tempo morto da fonte de distúrbios for pequeno em relação às variáveis controladas pelos controladores internos e as constantes de tempo dos controladores secundários forem pequenas em relação à do controlador primário, considerando-os assim reguladores. Este tipo de estratégia largamente utilizada em veículos autônomos, onde os controladores escravos são geralmente construídos em Hardware através de microcontroladores digitais que aumentam a velocidade de processamento [16] [29]. Se as condições forem cumpridas, os controladores internos podem ser desconsiderados durante o projeto do bloco “controlador”, pois a sua dinâmica tende a ser desprezível em relação à dinâmica total do robô.

Geralmente, o controle de trajetórias para robôs autônomos consiste em fazer o robô seguir referências de posição e orientação, (x, y, θ) , e velocidades, (V, ω) . Em alguns casos,

algumas destas variáveis, como V e ω podem ser consideradas livres. Assim, o controlador deve ser projetado como uma função que gere um par de sinais de referência (*setpoints*), $SP1$ e $SP2$, que minimize o vetor de erros $e = [e_x, e_y, e_\theta, e_\omega]^T$. Como o sistema possui restrições não-holonômicas e não pode ser controlador por um único controlador multivariável, uma boa alternativa é a mudança de variáveis e o uso de controladores independentes para cada nova variável ou para um conjunto destas variáveis. Um controlador como este pode ser visto na figura 5.3. Nesta figura, houve uma mudança de variáveis no sentido de que as variáveis x e y não são diretamente controladas e há o aparecimento da variável “distância”, que pode ser a distância entre o alvo e o robô. Desta forma a distância poderia, por exemplo, substituir as variáveis e_x e e_y no vetor de erros. No robô móvel LOGBOT, o erro é orientado a distância lateral do robô com a parede.

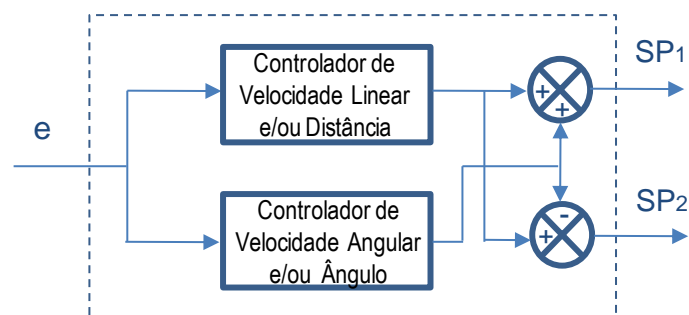


Figura 5.3 – Organização interna do Controlador Global

O controlador da figura 5.3 supõe que as “variáveis lineares” posição e velocidade linear podem ser controladas de forma independente das “variáveis angulares”, orientação e velocidade angular. Além disso, é considerado que cada uma dessas variáveis pode ser controlada por um único controlador monovariável, já que a única saída de cada um dos controladores é usada como parte de duas variáveis manipuladas. O projeto de cada controlador pode, então, ser feito de forma independente, considerando que o sistema contém apenas uma variável manipulada que é a soma das variáveis reais, V_1 e V_2 . Assim, pelas equações 4.13 e 4.14, considera-se que as novas variáveis manipuladas são proporcionais a

V e ω . A Equação (4.13) explica também o sinal negativo da saída SP_2 . Desta forma, o problema de controle foi simplificado e pode ser resumido como a determinação de uma função que minimize o erro de uma única variável medida a partir de uma única variável manipulada. Apesar de este ser um problema relativamente simples, conhecido e facilmente solucionado por controladores lineares do tipo P, PI, PID [30]-[34], a grande dificuldade do projeto está no fato que o robô não possui os controladores locais¹ de velocidade, como será visto nas próximas seções. Os controladores PID de velocidade são chamados locais porque, em geral, estes estão localizados no robô próximos às rodas.

5.3 Controlador do LOGBOT no Modo Teleoperado

No modo teleoperado o operador através da estação de controle envia um conjunto de comandos denominados de macro, e configuração (x, y, θ) do robô como mostra a figura 5.5.

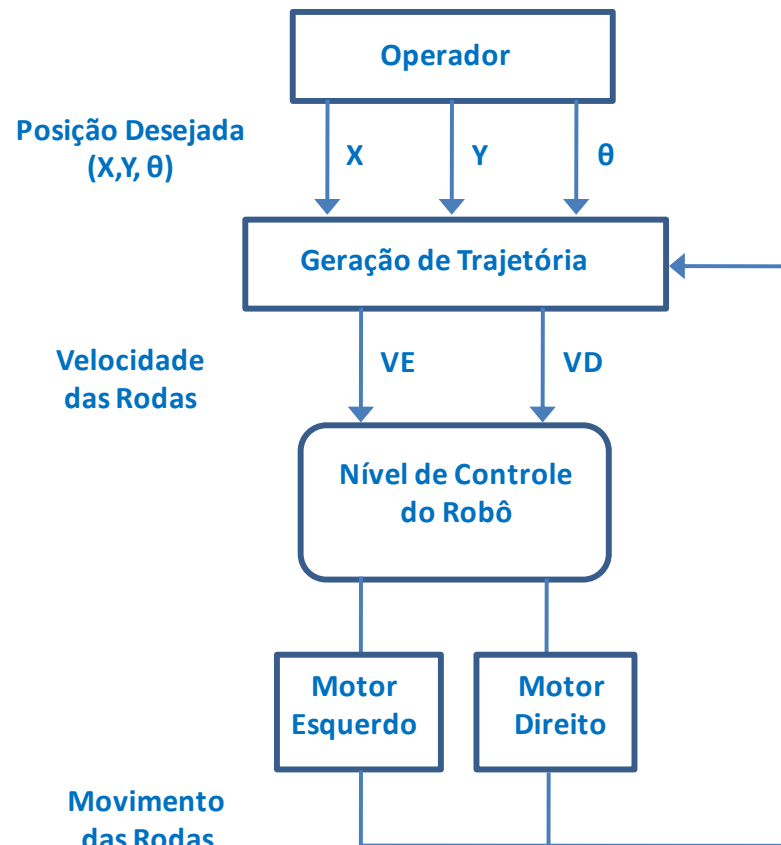


Figura 5.6 – Corrtrole teleoperado de movimento do robô móvel LOGBOT

5.4 Controlador do LOGOBOT no Modo Autônomo - Wall-Following

O Controle *Wall-Following* (Seguimento de parede) é o modo de uso do controlador usado no LOGBOT, e é uma técnica usada para navegação autônoma em estruturas conhecidas e ambientes indoor desconhecidos [23]. O Wall-Following basicamente usa a parede, ou contornos de um objeto como um marco natural do ambiente (*Landmark*) onde atua como base de informação para gerar um novo estado de trajetória a ser executada. Para isto o robô usa sensores de distância do tipo sonares ou lasers para analisar o ambiente e manter o robô a uma distância de referência da parede. A saída destes sensores não tem ligação direta com o controle de trajetória, e servem assim como base de informação com relação à parede. O robô deve manter-se a uma distância específica da parede, cujo valor é definido como a referência do controlador. O algoritmo Wall-Following usado quando a proximidade de obstáculos consiste basicamente em:

1. Seguir em frente enquanto tiver uma parede à esquerda (direita);
2. Manter-se a uma distância de referência da parede à esquerda (direita);
3. Quando houver uma aproximação frontal com a parede virar para a direita (esquerda);
4. Se não houver mais parede perto do lado esquerdo, virar para a esquerda (direita).

O robô pode tanto seguir a parede a sua esquerda quanto à parede a sua direita, por isso as informações entre parênteses. A figura 5.6 mostra o corredor de testes do robô móvel LOGBOT no prédio de engenharia elétrica da Universidade Federal do Pará.



Figura 5.6 – Corredor de testes do LOGBOT (PGEE-UFPA, 2009)

A figura 5.7 mostra o modelo e a disposição configuração do robô móvel que se move ao longo do corredor com velocidade V e orientação θ .

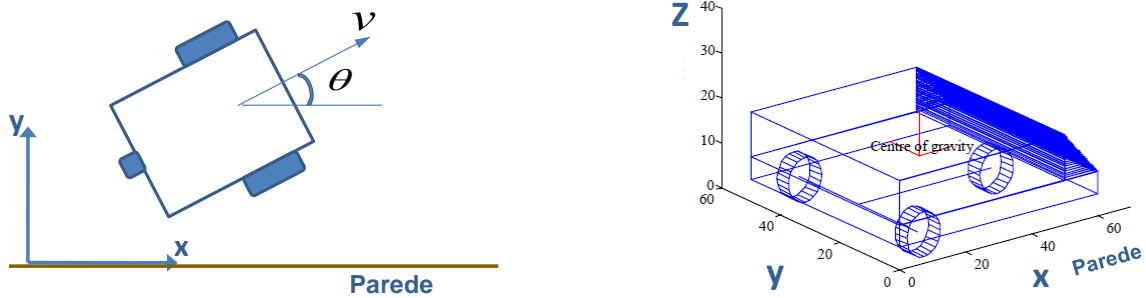


Figura 5.7 – Disposição do Robô no Corredor em 2 e 3 Dimensões.

5.4.1 Modelo Matemático do Wall-Following para um Robô Móvel Não-Holonômico

Na figura 5.8 é considerado que o robô está a uma distância diferente de zero do ponto global P fixo em um plano cartesiano fixo.

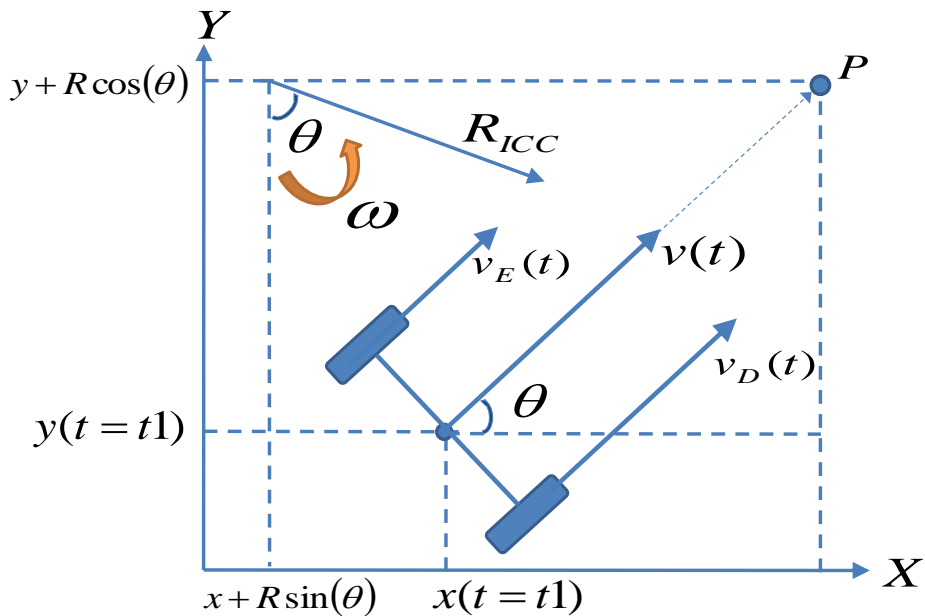


Figura 5.8 – Configuração do robô em seu primeiro instante de movimento.

O robô se desloca em direção a P e é governado pela combinação da ação da velocidade angular ω e da velocidade linear $v(t)$, que é a direção a partir do eixo do robô. No conjunto de equações cinemáticas que envolvem a sua orientação θ e sua posição cartesiana $[x \ y]^T$ do robô são descritos.

Na figura 5.9 considera-se o caso onde o robô navega no corredor. As variáveis de estado deste modo são definidas em relação ao corredor como \hat{Y} e $\hat{\theta}$, onde \hat{Y} representa a distância atual do robô com a parede, e $\hat{\theta}$ o desvio angular atual com relação ao eixo do corredor. É importante a compreensão das notações:

Trajétórias e configuração:

p : trajetória desejada

* p : trajetória real

ω : velocidade angular

\dot{x}, \dot{y} : configuração do robô em movimento

Variáveis Controladas:

$\hat{\theta}$: orientação atual do robô com a parede

\hat{Y} : distância atual do robô com a parede

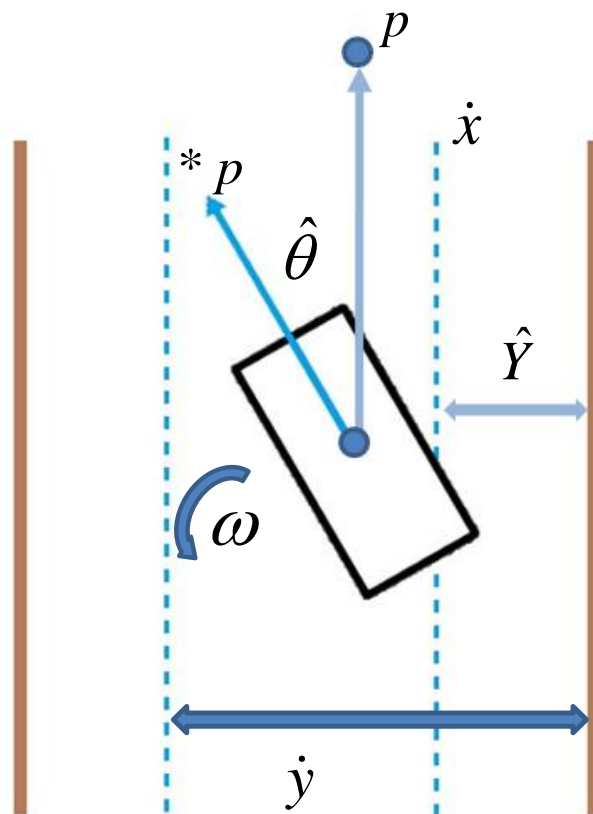


Figura 5.9 – Organização das variáveis e disposição do robô no Wall-Following.

5.4.2 Geometria da Aquisição de Dados do Vetor de Sensores de Ultra-Som

Nesta dissertação, o controlador Wall-Following do robô móvel LOGBOT é considerado com o uso de um vetor de sensores de distância do tipo ultra-som. O LOGBOT deve seguir a parede e manter uma distância de referência de 40 centímetros. Os sensores do tipo ultra-som medem com precisão distâncias relativamente pequenas, comumente no intervalo de 15 a 300 centímetros e estas leituras são medidas perpendiculares a parede, conforme visto na figura 5.9.

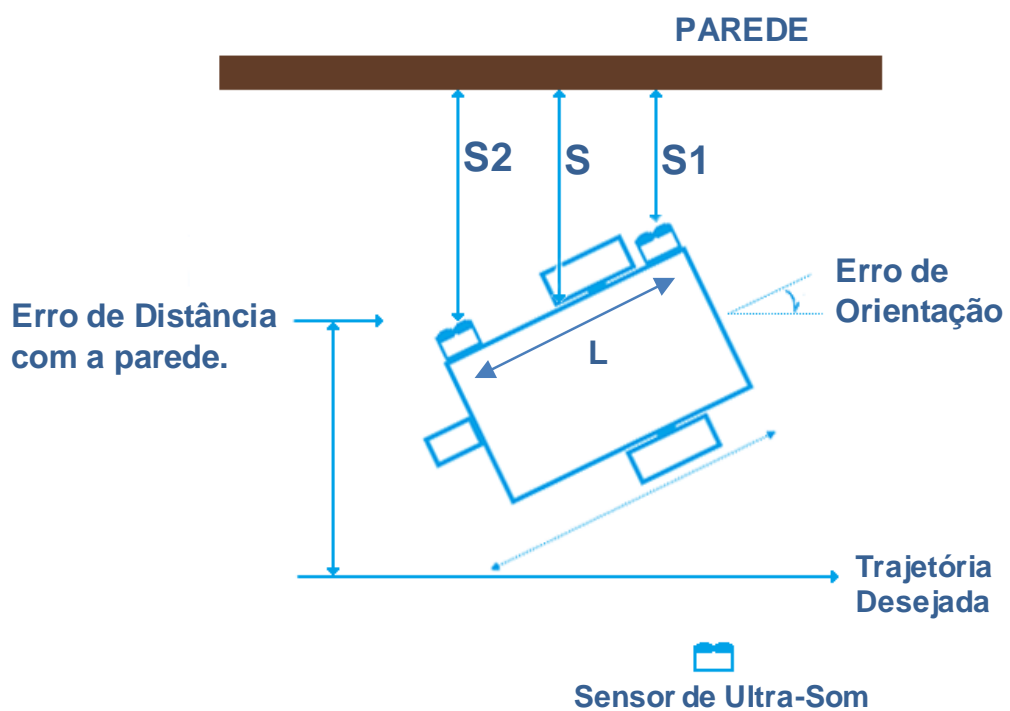


Figura 5.9 – Geometria do Robô na problemática do Wall-Following.

Onde S_1 e S_2 é a leitura atual dos sonares, e L a distância entre o sonar S_1 e o sonar S_2 . A grande desvantagem do sensor de ultra-som é a fraca resolução lateral de sua leitura, que tem um largo feixe de leitura e ocasiona um espalhamento que gera inconsistências na leitura. S é a distância do centro de massa do robô com a parede.

A figura 5.10 mostra o tratamento do erro de posição – que é a distancia definida com a parede, e do erro de orientação – que é a diferença angular entre o eixo do robô e a distância da parede.

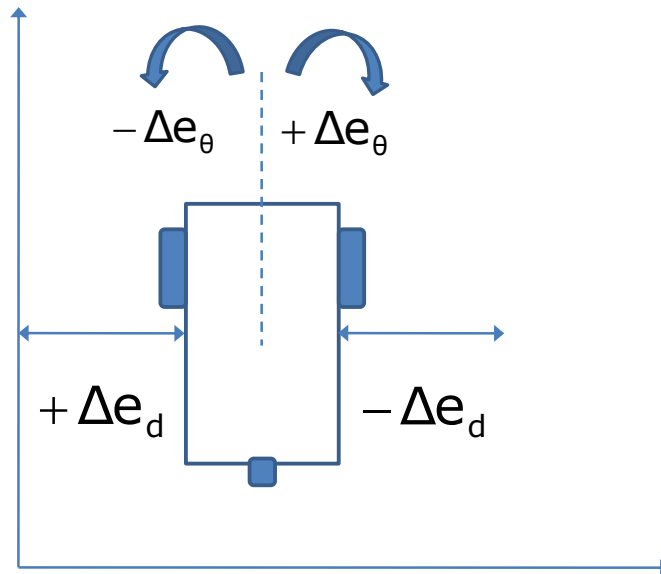


Figura 5.10 – Geometria Tratamento dos Erros de Distância e Orientação.

A posição do robô quando está a uma distância abaixo da referência o erro de posição é negativo, e quando uma distância acima da referência, o erro de posição é positivo. O erro de orientação do robô é negativo quando a parte dianteira estiver mais próxima a parede que a parte traseira, e positiva quando a parte traseira estiver mais próxima que a parte dianteira.

5.4.3 Controlador Proporcional de Trajetórias para Robôs Móveis Não-Holonômicos

O objetivo do controlador proporcional de trajetórias é desenvolver as ações de controle $\omega(t)$ e $v(t)$ tais que os erros de controle para posição e orientação do robô sejam reduzidos assintoticamente a zero, enquanto mantém as ações de controle dentro de seus limites específicos. No robô móvel LOGBOT é usado um controlador proporcional segundo a seguinte lei de controle:

$$U = K_D \times E_D + K_\theta \times \hat{\theta}, \quad (5.3)$$

onde U é o sinal de controle, K_D e K_θ são os ganhos proporcionais de distância e orientação respectivamente, E_D é o erro calculado pela diferença das distâncias da posição de referência

com a parede Y_{ref} com a leitura da posição atual \hat{Y}_D , e $\hat{\theta}$ é o estado atual da orientação do robô com relação ao eixo de x . O sinal de controle é usado nas equações que regem o controle do robô:

$$\begin{aligned} VRE(t) &= \omega_E \times r - U \\ VRD(t) &= \omega_D \times r + U \end{aligned} \quad (5.4)$$

onde $VRE(t)$ e $VRD(t)$ são as velocidades lineares, das rodas esquerda e direita respectivamente enviadas na forma de sinais modulados por largura de pulso – PWM (Pulse Width Modulation) para cada motor acoplado as rodas, $v(t)$ a velocidade linear do robô com um todo, U o sinal de controle obtido pela equação, ω_E e ω_D as velocidades angulares dos motores esquerdo e direito, e r o raio da roda acoplada ao eixo de cada motor.

5.4.4 Algoritmo de Controle

O algoritmo de controle é implementado no microcontrolador do robô móvel LOGBOT, e as tarefas de todo o procedimento são vistas a seguir:

1. Medir a distância entre o robô e a parede

$$Y_D = 0.5 \times (S1 + S2)$$

2. Medir o Erro (Referência em: 40 cm):

$$E_D = Y_{Dref} - \hat{Y}_D$$

3. Zona Morta (38 cm e 42 cm):

$$Se |E_D| = 2, \text{ Então } E_D = 0$$

4. Mede a orientação em graus radianos:

$$\theta_{rad} = \arcsen\left(\frac{(S1 - S2)}{L}\right), \text{ se } S1 \geq S2$$

$$\theta_{rad} = \arcsen\left(\frac{(S2 - S1)}{L}\right), \text{ se } S2 \geq S1$$

5. Converter para graus:

$$\hat{\theta} = \frac{\theta_{rad} \times 180}{\pi}$$

6. Lei de Controle:

$$U = K_D \times E_D + K_\theta \times \hat{\theta}$$

7. Velocidade da Roda Esquerda

$$VRE = VELBOT + U$$

8. Velocidade da Roda Direita

$$VRD = VELBOT - U$$

Na primeira tarefa a leitura dos sensores de ultra-som mede a distância entre o centro de massa do robô com a parede (coordenada x), onde $S1$ e $S2$ são as variáveis correspondentes as leituras dos sonares, e L a distância entre $S1$ e $S2$ como mostrado na figura 5.9; \hat{Y}_D é a distancia atual do centro de massa do robô com a parede. Na segunda tarefa é feito a medição do erro de distância com a parede. Na terceira tarefa é configurada uma zona morta para diminuir a margem oscilatória da ação de controle. Na quarta tarefa o ângulo de orientação do robô com relação ao eixo de x é extraído em graus radianos e tratado conforme visto na figura 5.10, assim é possível mensurar a distância do robô com a parede e sua orientação, através somente do uso de sonares. Na quinta tarefa a orientação em graus radianos é convertida para graus. Na sexta tarefa é processada a lei de controle Wall-Following para gerar a ação de controle U . Nas tarefas sete e oito, a ação de controle é usada nas equações que calculam os respectivos sinais PWM para cada motor.

O algoritmo de controle é processado por finitas iterações enquanto o robô estiver no modo autônomo e em execução por um agente inteligente de software que governa toda a programação do microcontrolador usado no robô móvel LOGBOT. Mais detalhes do agente inteligente de software para robôs móveis são abordados no capítulo 6.

5.5 Sintonia do Controlador Proporcional de Trajetórias Wall-Following

Os ganhos K_D e K_θ podem ser sintonizados de maneira experimental, através do método de *Kuipers* para obtenção de parâmetros em controladores Wall-Following para robótica móvel [25]. Primeiro deve ser encontrado o K_D , através de um ensaio experimental com o robô móvel a uma distância pré-definida da parede e seguir os passos:

- O robô deve ser programado para se manter a uma distância pré-definida da parede.
- O ganho K_D deve ser testado entre 0.1 e 0.6.
- A sintonia do ganho só deve ser validada, quando o robô se manter em ganho crítico, sem colidir com a parede, conforme mostra a figura 5.11.
- Quando K_D estiver sintonizado, então o ganho K_θ deve ser extraído na forma:

$$K_\theta = 2 \times \sqrt{K_D} \quad (5.5)$$

No ensaio realizado com o robô móvel LOGBOT, os ganhos obtiveram os seguintes valores – $K_D = 0.4$ e $K_\theta = 1.26$. De acordo com o método de *Kuipers*, o K_θ deve ser maior que K_D . Isto faz com que este controlador proporcional para robôs móveis tenha o comportamento de um controlador do tipo proporcional derivativo. Com os ganhos obtidos pelo método de sintonia de *Kuipers* o robô móvel LOGBOT tem a seguinte lei de controle:

$$U = 0.4 \times E_D + 1.26 \times \hat{\theta} \quad (5.6)$$

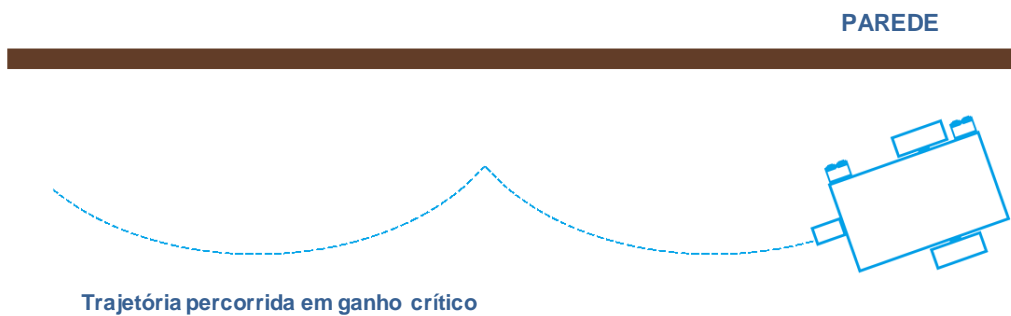


Figura 5.11 – Movimento do robô com trajetória com comportamento em ganho crítico.

Capítulo 6: Controle Inteligente de Robôs Móveis

As técnicas convencionais de controle referem-se, geralmente, a teorias e métodos desenvolvidos para controlar sistemas dinâmicos desenvolvidos para controlar sistemas dinâmicos cujos comportamentos são descritos por equações diferenciais ou a diferenças. Devido às crescentes demandas tecnológicas de sistemas de alta complexidade que, em muitos casos, não podem ser representados na forma das equações mencionadas, novas abordagens de controle vêm sendo desenvolvidas [14].

Entre estas novas abordagens destaca-se a intitulada controle inteligente. Um controlador inteligente é caracterizado, principalmente, por um alto grau de autonomia em sua operação. As metodologias de controle inteligente combinam diversas áreas, associadas não só a teoria de controle, como também à engenharia da computação (sistemas inteligentes), à estatística e à pesquisa operacional (técnicas de otimização). Este capítulo aborda à programação de robôs móveis, através do uso de agentes inteligentes de software. A figura 6.1 mostra a estrutura de um agente inteligente de software.

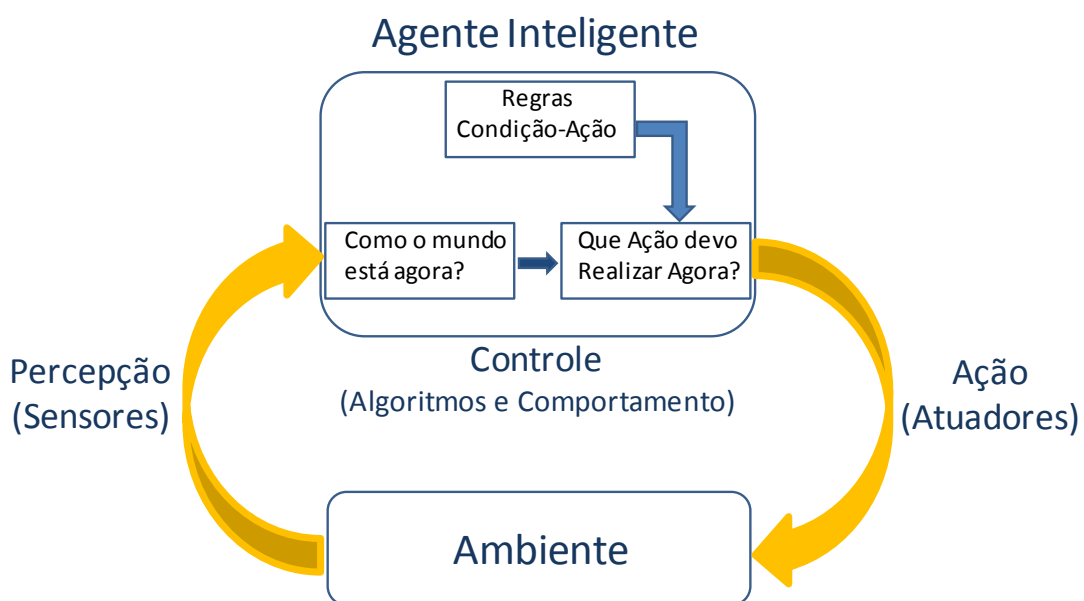


Figura 6.1 – Modelo Genérico de um Agente Inteligente de Software

Um agente inteligente é uma entidade computacional, em geral um software, que é capaz de interagir com o ambiente de modo autônomo e flexível conforme os objetivos para os quais foi projetado. Um agente se situa em um ambiente, no sentido de que recebe sinais de seus sensores e executa ações. Um agente é autônomo se pode agir sem a intervenção direta de operadores ou de outros agentes. Um agente é proativo se, ao invés de apenas reagir aos estímulos vindos do ambiente, tem iniciativa própria para realizar ações que estão no sentido de alcançar os seus objetivos.

Os agentes podem apresentar capacidade de adaptação ou de aprendizado, ou seja, podem mudar o seu comportamento em função da seqüência de experiências anteriores. Os agentes podem ser sociais. Significa que podem cooperar, quando apropriado, com outras entidades do ambiente, de forma a melhor resolver seus problemas. Os agentes podem ser móveis se possuem a habilidade de se transportar de uma máquina para outra.

O uso de agentes inteligentes para controle de robôs móveis é uma área de estudo recente, e é herdada da cibernética [44] [47]. Através de um agente inteligente é possível realizar a fusão sensorial de informações lidas dos sensores, e desta forma comparar a saída da fusão sensorial com possíveis comportamentos a qual estiver mais adequada (condição), a fim de selecionar uma regra específica (ação) para tratar tal comportamento. Com esta base de dados formada (fusão sensorial e detecção de comportamento) é possível ativar uma estratégia de controle, cuja saída é aplicada aos atuadores.

6.1 Paradigmas de Programação de Agentes Inteligentes para Robôs Móveis

Para trabalhar de forma autônoma, os robôs móveis devem ser programados adequadamente. Um programa basicamente processa as informações obtidas pelos diversos sensores e envia comandos para os atuadores. A maneira pela qual esse processamento é feito diferencia os paradigmas de programação de robôs.

Existem três paradigmas principais para a programação de robôs móveis: (i) Deliberativo, (ii) Reativo, e (iii) Híbrido. Alguns autores costumam chamar esses paradigmas de “Arquiteturas de Software” ou, simplesmente, de “Arquiteturas Robóticas” [3]. O termo “Paradigmas de Programação” é mais genérico e evita possíveis confusões com a arquitetura de hardware do robô ou com arquiteturas de software específicas construídas com base em alguns destes paradigmas. Os três principais paradigmas são discutidos a seguir [10] [43].

6.1.1 Paradigma Deliberativo

O paradigma deliberativo foi o primeiro a ser utilizado na programação de robôs móveis e herda suas características dos sistemas de planejamento utilizados pela inteligência artificial clássica. Também chamado de *Paradigma Hierárquico* ou *perceber-planejar-agir*, o paradigma deliberativo organiza a execução de forma seqüencial [47]. Primeiramente, o robô obtém uma série de informações sensoriais e constrói uma representação ou modelo do ambiente. O próximo passo consiste em executar um planejamento sobre essa representação de forma a obter uma seqüência de ações. Por fim, as ações são executadas e o processo começa novamente. Portanto, o paradigma deliberativo depende fortemente da construção de uma representação do mundo, sobre a qual o programa irá deliberar de forma a chegar à seqüência de ações desejada.

O paradigma deliberativo sofre de dois problemas principais. O primeiro deles está relacionado com a necessidade de se construir uma representação adequada do mundo. Em geral, robôs móveis trabalham em ambientes dinâmicos, parcialmente observáveis, e as informações sobre esses ambientes são obtidas por meio de sensores sujeitos a erros. Com isso, é difícil manter um modelo preciso do mundo e realizar o planejamento adequadamente. Outro problema está relacionado com o tempo gasto pelo processo seqüencial de “perceber-planejar-agir”. Em geral, a atualização do modelo com as informações sensoriais e o

planejamento das ações são processos computacionalmente caros que podem comprometer o desempenho do robô, principalmente em ambientes dinâmicos.

6.1.2 Paradigma Reativo

As limitações do paradigma deliberativo fizeram surgir, em meados da década de 1980, o paradigma reativo para programação de robôs [23]. No paradigma reativo, ao invés da implementação hierárquica e seqüencial do modelo perceber-planejar-agir, a programação dos robôs é feita através de uma série de módulos que executam em paralelo e conectam mais diretamente os processos de percepção e ação. Em geral, esses módulos recebem o nome de comportamentos (*behaviors*).

Os comportamentos funcionam como os blocos básicos na construção dos programas. No paradigma reativo, a programação dos robôs é feita combinando-se esses comportamentos de forma a obter as ações desejadas. Por exemplo, um robô cuja missão seja procurar e coletar itens possui vários comportamentos básicos, tais como “mover aleatoriamente”, “desviar de obstáculos”, “buscar item”, etc. Além da ligação forte entre percepção e ação, os comportamentos têm outras características importantes. Primeiramente, eles devem ser simples e executar de forma rápida, ou seja, reagir rápido aos dados obtidos pelos sensores. Além disso, os comportamentos devem ser autocontidos e independentes, de forma que eles possam ser reutilizados facilmente em diferentes programas. Por fim, eles devem se basear apenas em informações locais e não construir ou depender de uma representação do mundo. De fato, Rodney Brooks, um dos principais advogados desse paradigma, argumenta que a inteligência não depende de representações complexas do mundo e que o “próprio mundo é o seu melhor modelo” [10].

Um ponto importante no paradigma reativo é como combinar os diversos comportamentos. Brooks [10] propôs uma das arquiteturas seminais do paradigma reativo chamada de Arquitetura de Subsunção (*Subsumption Architecture*). Nela, os comportamentos

são organizados em camadas de acordo com as suas competências e comportamentos de nível mais baixo ou mesmo inibi-los. Outra arquitetura, chamada de *Motor Schemas* considera que a ação a ser tomada pelo robô (nesse caso um vetor de movimento) é um somatório das ações geradas por diferentes comportamentos.

Uma das principais críticas ao paradigma reativo é que ele dificulta a realização de tarefas cognitivas de mais alto nível. Muitas vezes não se consegue prever exatamente qual será a ação resultante da combinação dos vários comportamentos. Por isso, esse comportamento resultante é em geral chamado de comportamento emergente. Enquanto os defensores do paradigma reativo dizem que isso é natural e não traz problemas, os críticos, principalmente os mais teóricos, argumentam que é impossível obter provas formais de que os robôs irão executar do modo desejado.

6.1.3 Paradigma Híbrido

Apesar do grande sucesso do paradigma reativo na programação de robôs móveis, a dificuldade em se prever os comportamentos emergentes fez com que o processo de definir e combinar os comportamentos básicos se tornasse “mais uma arte do que uma ciência”. Além disso, a eliminação por completo de uma representação do mundo e da possibilidade de planejamento sobre ela dificultou a programação de certas tarefas. Por exemplo, como fazer um robô percorrer o menos caminho em um ambiente com obstáculos sem armazenar informações do ambiente e raciocinar sobre elas?

O paradigma híbrido tenta combinar os benefícios dos dois paradigmas anteriores. A idéia principal é fazer uma decomposição, retirando do laço principal de percepção e ação os módulos responsáveis pela deliberação, ou seja, pela representação do mundo e pelo planejamento. Dessa forma, os processos deliberativos e reativos funcionam separadamente, com velocidades de execução diferentes, e são interligados por módulos responsáveis por seqüenciar as ações propostas pela camada deliberativa junto à camada reativa.

A forma mais utilizada para a implementação do paradigma híbrido é a construção de uma arquitetura de três camadas. A primeira camada, chamada de *Camada Reativa*, é responsável pelo controle de baixo nível do robô. Ela é composta por um ou mais módulos que implementam comportamentos (reativo, deliberativo...) como os descritos na seção anterior, ou seja, comportamentos simples que ligam diretamente sensores e atuadores e executam de forma extremamente rápida.

A segunda camada é a *Camada Executiva* ou *Seqüenciador*. Essa camada faz a ligação entre a camada deliberativa (descrita a seguir) e a camada reativa e tem duas responsabilidades principais: (i) decidir que comportamentos da camada reativa devem ser ativados a partir dos planos definidos pela camada deliberativa e (ii) montar uma representação do ambiente que possa ser usada por essa camada a partir das informações sensoriais obtidas.

Por fim, a *Camada Deliberativa* ou *Planejador* é responsável pela elaboração de planos complexos com base nos modelos construídos do ambiente. Normalmente, essa camada usa algoritmos computacionalmente caros não podendo, portanto ser executados nos níveis mais baixos da hierarquia. Dentre os algoritmos executados pela camada deliberativa estão algoritmos específicos para o planejamento de caminhos.

6.1 Agente Inteligente do Robô Móvel LOGBOT

Um agente inteligente baseado em comportamentos foi desenvolvido para ser embarcado ao robô móvel LOGBOT para executar a escolha de tarefas que o robô pode processar durante sua navegação autônoma. O agente tem a seguinte configuração:

- Paradigma reativo baseado na detecção de comportamentos do corredor;
- Ao perceber um comportamento, ativa uma tarefa para a ação do robô;
- É programado em uma linha de processamento (Thread) do Sistema Operacional de Tempo Real embarcado no microcontrolador do robô.

A figura 6.2 mostra a ação do agente embarcado no robô.



Figura 6.2 – Agente Inteligente de Software Embarcado ao Robô Móvel LOGBOT.

A função deste agente é monitorar através do vetor de sonares o comportamento em qual o robô se encontra que podem ser – um obstáculo a ser desviado, voltar para sua referência dentro do espaço de configuração X-Y, seguir parede pelo controlador proporcional de trajetórias, ou tratar a problemática da porta que causa maior erro de posição e orientação. Os comportamentos e tarefas que o LOGBOT pode executar são mostrados na figura 6.3.

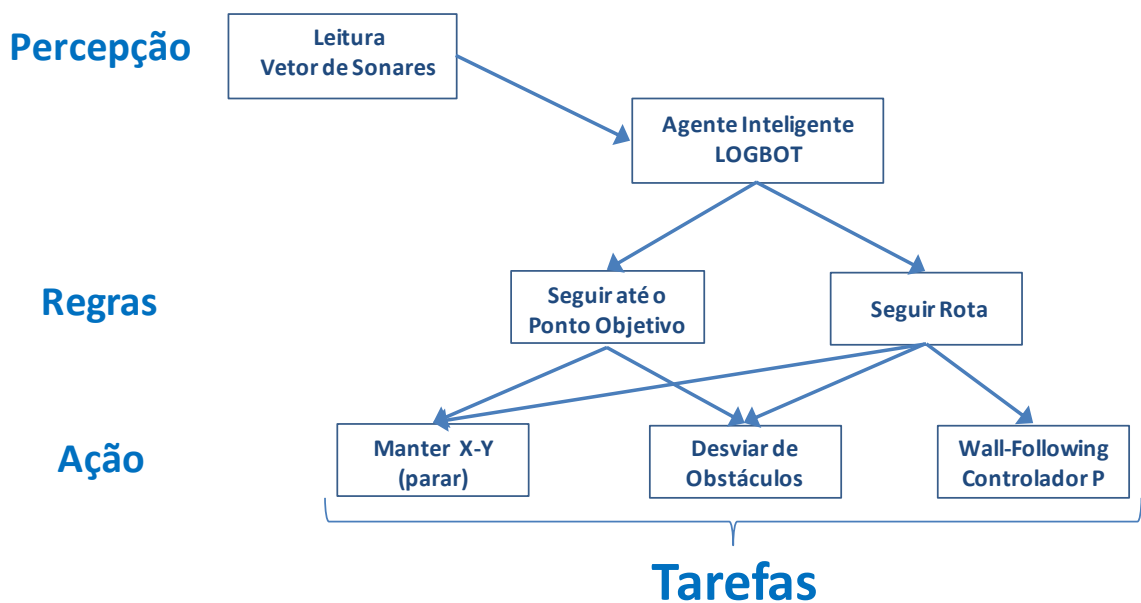


Figura 6.3 – Configuração do Agente Reativo do LOGBOT.

Capítulo 7: Hardware Embarcado para Robôs

Neste capítulo são abordados aspectos importantes sobre arquiteturas de sensores, hardware e firmware para sistemas robóticos e soluções dedicadas. A importância dos sistemas embarcados e sensores inteligentes, através do uso de microcontroladores. E também o uso de sistemas de comunicação sem fio para robótica. As tecnologias descritas neste capítulo foram integralmente usadas na construção do robô móvel LOGBOT.

7.1 Sistemas de Computação em Automação e Robótica

A arquitetura computacional típica de um robô é descrita na figura 7.1. É imprescindível que esta arquitetura tenha as interfaces de – Percepção, através de portas com conversor A/D; Ação, através de portas TTL, PWM e D/A; Comunicação, através de portas com registradores de comunicação serial, paralela, rede *ethernet*, etc.

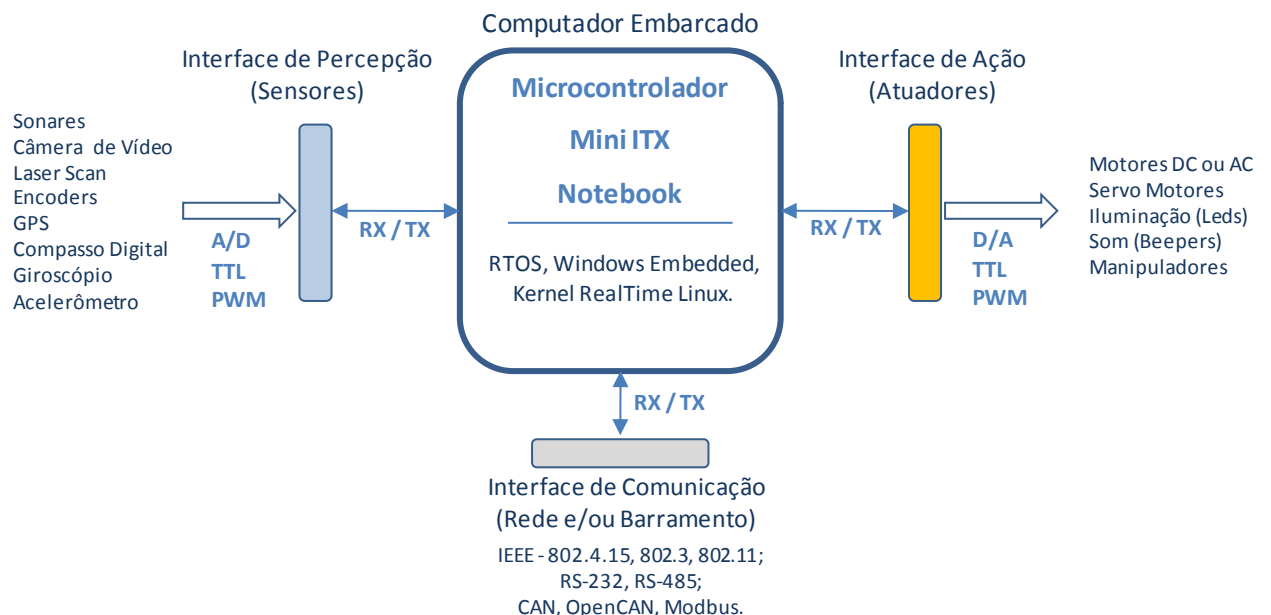


Figura 7.1 – Arquitetura Computacional Típica de um Robô

Um microcontrolador é usado como o computador embarcado do robô móvel LOGBOT. Um hardware microcontrolado é classificado de sistema embarcado, pois é usado em aplicações específicas onde não é viável o uso de uma arquitetura PC padrão [42].

Os sistemas embarcados apresentam baixo custo e consumo de energia elétrica. O programa gravado em um microcontrolador é denominado de *firmware*. Em robótica, os microcontroladores comumente desempenham as funções de perceber o ambiente, através de sensores, e de controlar os motores, através do gerenciamento dos sinais de entrada e saída programados no algoritmo de controle do robô.

A grande vantagem de se usar um microcontrolador ao invés de um computador comum está no baixo custo e nas dimensões físicas, pois o peso de um notebook exige motores e estrutura mais robustos, o que aumenta significativamente o custo do projeto global de um robô. Porém o microcontrolador tem seus periféricos como memória (do tipo *RAM*, *E2PROM* e *FLASH*) e processador são limitados, se comparados a um computador pessoal. Isto faz com que as soluções de desenvolvimento de *firmware* para robótica sejam aperfeiçoáveis e que não sobrecarreguem os periféricos do microcontrolador.

7.1.1 Sistema Computacional Embarcado do Robô Móvel LOGBOT

No robô móvel LOGBOT é usado no modulo de decisão um microcontrolador PIC18F458 fabricado pela empresa *Microchip*TM que conta com uma CPU de 8-bits, e sua organização usa a arquitetura do tipo *Havard*, cuja comunicação com a memória de programa onde o firmware é executado tem palavra com tamanho de 14-bits, e a comunicação com a memória RAM tem a palavra com tamanho igual ao do processador (8-bits). Este microcontrolador possui oito portas de conversão A/D, que são usadas para leitura dos sensores de ultra-som do robô e para leitura do sensor de temperatura LM35; duas saídas PWM (*Pulse Width Modulation*) são usadas para controle de potencia dos motores; um canal de comunicação USART RS-232 usado para comunicação Zigbee 802.15.4 com o computador remoto; um canal para barramento I2C que se comunica com o relógio de tempo real interno, e com o sensor SHT75 de temperatura e umidade; É neste microcontrolador onde todo o algoritmo de controle e agente inteligente de software do LOGBOT é processado.

7.2 Sistemas Embarcados em Automação e Robótica

A partir dos anos de 1990, mais de 90% dos microprocessadores fabricados mundialmente são destinados a máquinas que usualmente não são chamadas de computadores [43]. Dentre alguns destes dispositivos estão aparelhos celulares, fornos microondas, automóveis, aparelhos de *DVD* e *PALMs*. O que diferencia este conjunto de dispositivos de um computador “convencional” (PC – Desktop, Notebook), conhecido por todos é o seu projeto baseado em um conjunto dedicado e especialista constituído por Hardware, Software e Periféricos – um Sistema Embarcado [42].

Um Sistema Embarcado (*Embedded System*) pela sua natureza especialista pode ter inúmeras aplicações. Pode ser usado no controle de freios de um veículo automóvel, em que esse sistema deve gerenciar certos periféricos de controle como um sensor. Em outra aplicação, um sistema embarcado através de suas funções de aquisição de dados, captura informações dos sensores de temperatura e umidade, e envia estes dados a um display ou para um computador via comunicação serial. Em sistemas ciber-físicos e robótica, os microcontroladores são massivamente usados na função de computadores embarcados, ou como gerenciadores das funções específicas de um módulo [44]. A figura 7.2 mostra o uso do microcontrolador na percepção e condicionamento de um ambiente fechado [45].

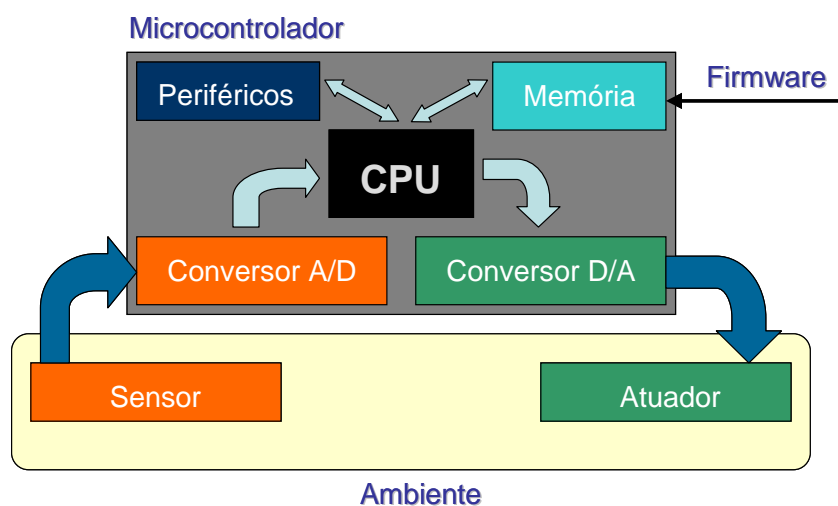


Figura 7.2 – Diagrama de um sistema embarcado monitorando um ambiente, através de um microcontrolador.

7.2.1 Considerações Importantes sobre Sistemas Embarcados

Colocar capacidade computacional dentro de um circuito integrado, equipamento ou sistema: São definições para um sistema embarcado. É importante ressaltar que um sistema como este deve ser mais do que um simples computador, ou seja, é um sistema completo e independente, mas desenvolvido para realizar apenas uma determinada tarefa.

Um sistema é classificado como embarcado quando este é dedicado a uma única tarefa e interage continuamente com o ambiente a sua volta por meio de sensores e atuadores [44]. Por exigir uma interação contínua com o ambiente, este tipo de sistema requer do projetista um conhecimento em programação, sistemas digitais, noções de controle de processos, sistemas de tempo real, tecnologias de aquisição de dados (conversão analógico/digital e sensores) e de atuadores (conversão digital/analógico, acionamento eletromecânico e PWM), e cuidados especiais na eficiência de estruturação do projeto e do código produzido.

O termo “embarcado” (do inglês *Embedded*) vem pelo fato de que estes sistemas são projetados geralmente para serem independentes de uma fonte de energia fixa como uma tomada ou gerador. As principais características de classificação deste sistema são a sua capacidade computacional e a sua independência de operação. Outros aspectos relevantes dependem dos tipos de sistemas, modos de funcionamento e itens desejados em aplicações embarcadas.

Todo sistema embarcado é composto por uma unidade de processamento como um microcontrolador, que é um circuito integrado, e fixado a uma placa de circuito impresso. Este possui uma capacidade de processamento de informações proveniente de um *software* que é executado internamente, e assim pode se afirmar que o *software* está embarcado na memória de programação do microcontrolador. Logo todo *software* embarcado pode ser denominado de *firmware* [47].

O sistema embarcado geralmente é uma solução formada de microcontrolador e *software (firmware)* dedicado e específico para desempenhar as funções operacionais de um equipamento/produto para o qual foi projetado e desenvolvido. Outros fatores importantes que ajudam a classificar um sistema como embarcado são:

- Dimensões físicas: desde a fase inicial do projeto têm que haver atenção ao tamanho e peso do sistema em desenvolvimento, sempre os menores possíveis. Com a crescente miniaturização dos equipamentos eletroeletrônicos, o fator tamanho e peso são decisivos na locomoção do sistema.
- Consumo de energia: Quanto maior for a autonomia do sistema e menor for sua necessidade de recarga, troca de sistema de alimentação ou baixo consumo elétrico, mais competitivo será o produto. Usar baterias, pilhas ou uma alimentação regular dentro de normas e legislações para o painel industrial em indústrias.
- Durabilidade: Muitos sistemas embarcados são projetados para trabalhar em ambientes com condições adversas (vibrações, calor, poeira, variações na tensão de alimentação, interferências eletromagnéticas, raios, umidade, corrosão, etc.) É necessário que o sistema resista ao máximo a todas estas interferências. Logo é importante haver um estudo sobre a forma de revestimento do circuito para isolamento em sua compatibilidade eletromagnética.

Na escolha do microcontrolador é importante observar se os recursos oferecidos suportam o objetivo do projeto. Logo é importante a leitura de seu manual denominado de *datasheet* do microcontrolador.

7.3 Arquitetura Modular para Hardware de Robôs

No projeto da camada eletrônica do robô (sistema eletrônico), se recomenda o uso da arquitetura modular. Este tipo de arquitetura permite que os circuitos sejam projetados e desenvolvidos separadamente, com suas entradas e/ou saídas de sinais disponíveis para serem interligados. O raciocínio de como separar o desenvolvimento do circuito em módulos deve levar em conta a função que a eletrônica irá executar. Para o controle dos motores é recomendável que somente o circuito do tipo ponte de chaveamento e potência do motor seja organizado em uma placa de circuito impresso ou *proto-on-board*, e que tenha uma interface de entrada para receber sinais do circuito de controle, e uma interface de saída para aplicação dos sinais elétricos aos motores, conforme visto na figura 7.3.

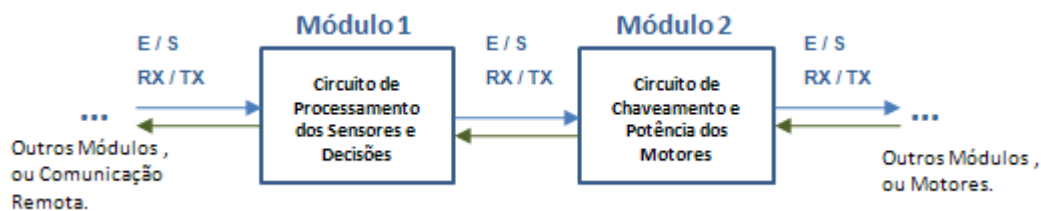


Figura 7.3 – Arquitetura Modular do Sistema Eletrônico do Robô.

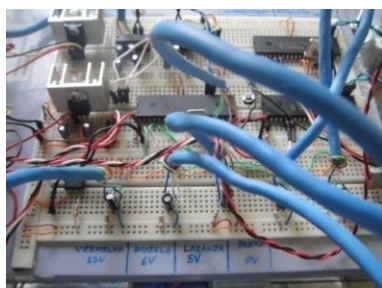
Os módulos podem ser integrados através dos sinais elétricos de entrada e saída, ou por recepção e transmissão em comunicação de dados, através de barramentos do tipo serial RS-232, RS485, CAN (*Controller Area Network*), Ethernet, entre outros.

As vantagens de se trabalhar com arquitetura modular são muitas, entre as principais – diminui a complexidade do projeto; a análise de problemas é mais eficaz e em menor tempo; facilidade de manutenção, pois basta trocar o módulo que apresentar problemas; flexibilidade; e o sistema passa a ser incremental devido ser possível a adição de mais módulos e assim aumentar as funções que o robô pode executar.

O projeto do robô móvel LOGBOT é organizado em três módulos que compõem toda a sua eletrônica:

7.3.1 Módulo de Decisão

O módulo de decisão realiza a integração de toda a informação obtida pelos outros módulos e processa um algoritmo com regras específicas para tomadas de decisões e qual atitude/tarefa o robô deve executar. É composto pelo microcontrolador PIC18F458 e pelo módulo de comunicação por rádio frequência Zigbee, conforme mostra a figura 7.4.



Processa o Agente Inteligente.

Módulo de Decisão

**Microcontrolador
PIC18F458**

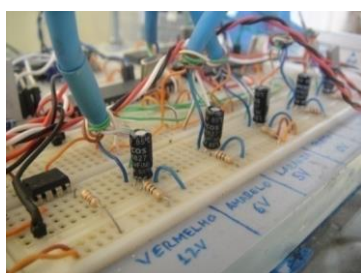


Comunicação RF ZigBee 2.4GHz.

Figura 7.4 – Módulo de Decisão.

7.3.2 Módulo de Percepção

O módulo de percepção realiza a aquisição e tratamento de dados de sensores do tipo: ultra-som para distância, temperatura e umidade do ar, conforme mostra a figura 7.5.



Temperatura e Umidade

Módulo de Percepção

**Gerência e Leitura dos
Sensores Inteligentes
PIC12F675**



Vetor de Sonares

Figura 7.5 – Módulo de Percepção.

O chaveamento da leitura do vetor de sensores de ultra-som usa um microcontrolador PIC12F675 deve emite pulsos com duração de 200 microssegundos em um período de 50 milissegundos. O dispositivo *EZ-1 Maxsonar* visto na figura 7.6 é o sensor de distância do tipo ultra-som usado no robô, e apresentou estabilidade em sua leitura, cuja saída analógica do sensor é proporcional a 10 milivolts por unidade de polegada detectada em seu raio de

leitura para detecção de parede ou obstáculos a uma distância de até 6 metros. O chaveamento da leitura de cada sonar do vetor se faz necessário devido à interferência infinita que a proximidade entre os sonares gera. Com esta configuração a geração das saídas correspondentes as distâncias é simultânea e ocorre em períodos de 0,5 segundos. A figura 7.7 mostra a configuração do vetor de sonares em uso no LOGBOT.

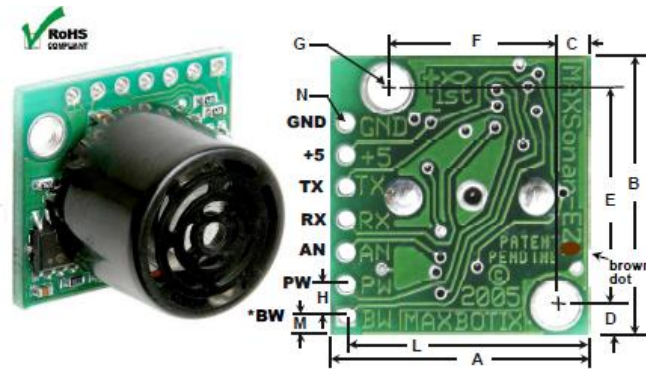


Figura 7.6 – Sensor de Ultra-Som LV Max - Sonar EZ-1.

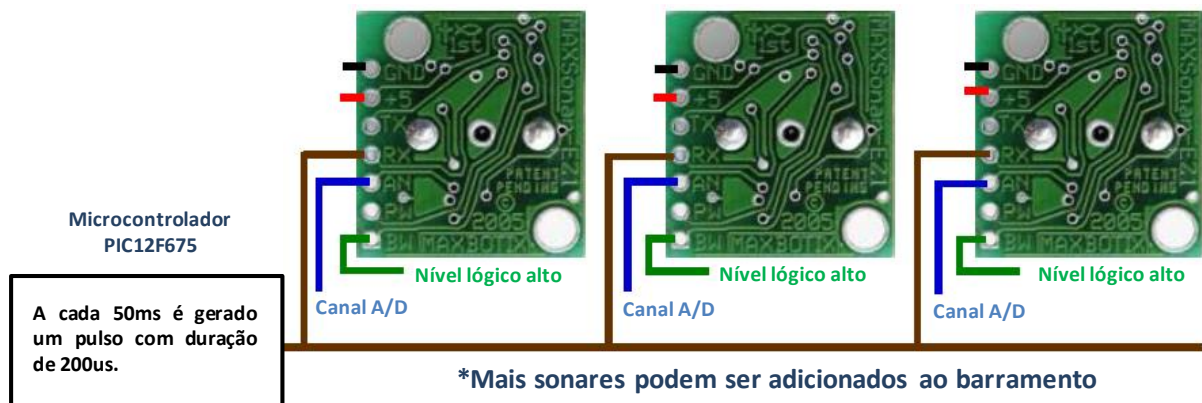
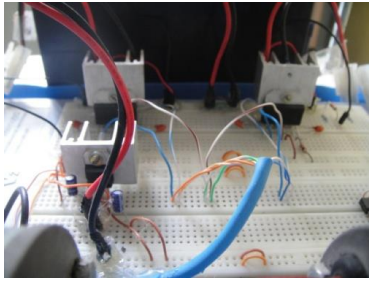


Figura 7.7 – Configuração do Vetor dos Sensores de Ultra-Som.

7.3.2 Módulo de Ação

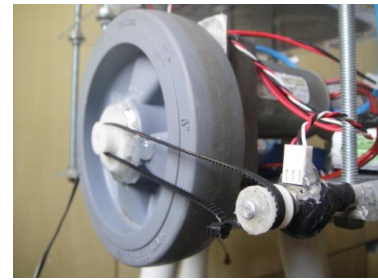
O módulo de comunicação realiza a emissão dos sinais elétricos modulados por PWM para acionamento e chaveamento e potência dos motores. Suas pontes de controle têm uma proteção contra a sobrecarga de corrente e param seu funcionamento caso detectem tal comportamento. A figura 7.8 mostra o sistema motor do módulo de ação.



Ponte LMD18200T

Módulo de Ação

Acionamento e Controle de Potência



Motor DC com Roda

Figura 7.8 – Módulo de Ação.

Neste módulo não há o uso de microcontrolador e sim de dispositivos cuja função é chavear e controlar a potencia dos motores conforme os sinais de controle emitidos pelo microcontrolador. Estes dispositivos são comumente chamados de ponte-H (H-Bridge), pois sua organização juntamente com o motor lembra a letra “H”, e são usadas duas pontes do integrado LMD18200T da Analog Devices que suporta motores com tensão de alimentação de até 55 volts, e consumo de corrente de até 3 ampéres, como mostra a figura 7.9.

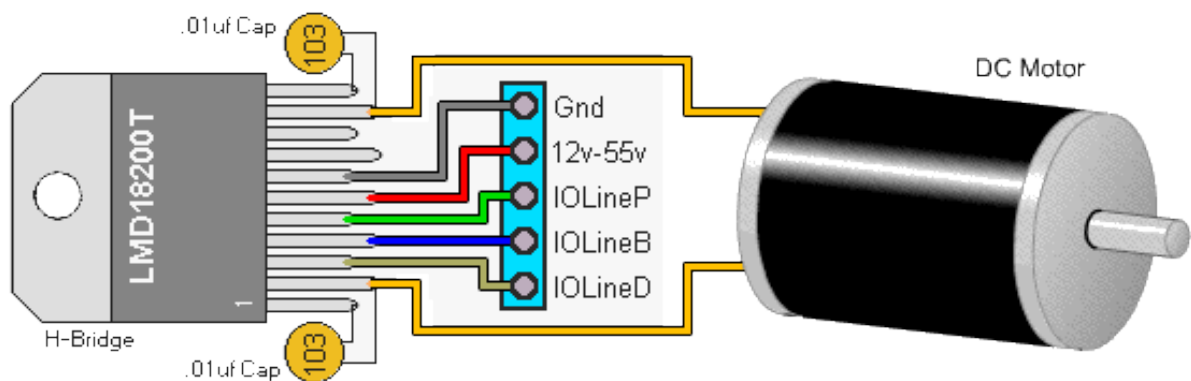


Figura 7.9 – Ligação da Ponte LMD18200T com o Motor DC.

7.4 Sensor Inteligente e Interface de Aquisição de Dados

No modo teleoperado o robô móvel LOGBOT realiza a leitura de sensores de temperatura e umidade para realizar análise ambiental, e para que os dados dos sensores possam ser adquiridos e transmitidos remotamente com alta confiabilidade, se viu necessário o uso de um padrão de aquisição de dados que se tornam possíveis estes requisitos. Com o

uso da norma IEEE1451 para aquisição de dados de sensores em rede [50] [51], um sensor inteligente é desenvolvido para aquisição de temperatura e umidade do ar visto na figura 7.10.

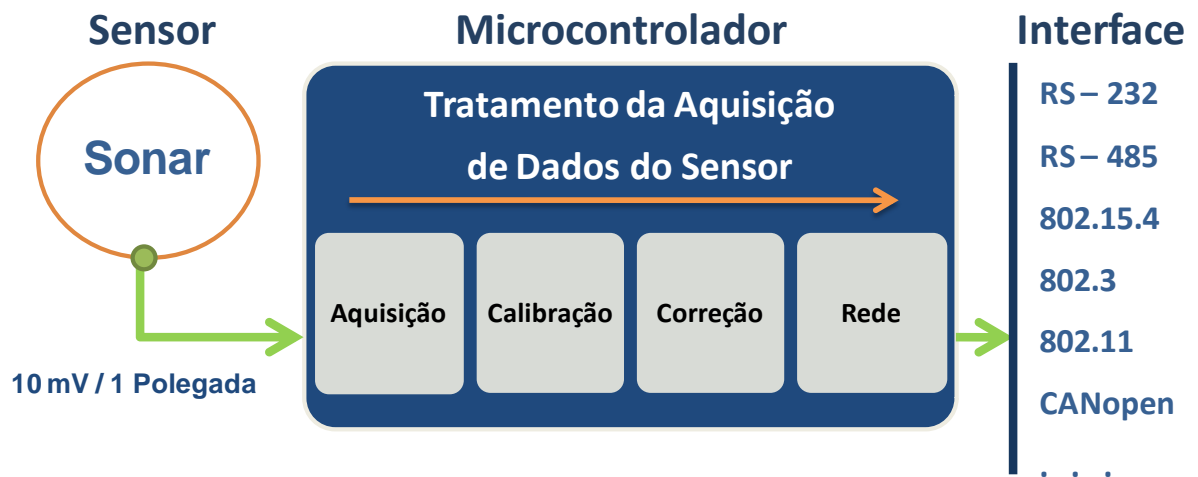


Figura 7.10 – Arquitetura de um Sensor Inteligente.

A integração entre sensor e microcontrolador em um único dispositivo é denominada de sensor inteligente [50]. A vantagem do sensor inteligente é que o microcontrolador realiza a aquisição, calibração, e correção do sinal. E assim transmite o dado da leitura com a escala correta e de alta confiabilidade para o barramento de comunicação. Com um sensor inteligente é possível ligar qualquer tipo de sensor, com qualquer tipo de rede.

7.5 Sistema Operacional de Tempo Real Embarcado em Robôs

Para que um robô possa executar suas tarefas e seguir trajetórias, ele deve realizar a leitura de vários sensores, e realizar a análise dos dados coletados, através do algoritmo de controle para que finalmente aplique a tensão nos motores, o que se refere à atitude que deve executar. Porém, por navegar em um ambiente dinâmico o robô se depara com obstáculos móveis e fixos e com outras situações que irão exigir dos movimentos do robô em um tempo de resposta mínimo, caso contrário sua demora no tempo de processamento e inércia da estrutura farão com que o robô colida com um obstáculo, ou que não realize sua tarefa com o desempenho esperado em seu projeto.

Isto caracteriza o sistema como crítico, e exige o uso de ferramentas que mantenham o processamento do algoritmo de controle em tempo real, logo se faz necessário o uso de um sistema operacional de tempo real denominado genericamente de RTOS (*Real Time Operational System*). Um sistema embarcado com uso de RTOS pode ser também definido por – sistema embarcado de tempo real [48].

Sistemas de tempo real em aplicações embarcadas operam em ambientes restritos com limitadas capacidades de processamento e de memória. Esses sistemas muitas vezes são projetados para fornecerem seus serviços dentro de tempos precisamente definidos. O fato de trabalhar com restrições de quantidade de memória, velocidade e tempos de execução é exatamente o que define a necessidade de uso de sistemas operacionais real-time em aplicações embarcadas. Sistemas de controle críticos industriais também se valem do determinismo de sistemas operacionais de tempo real devido a sua capacidade de resposta em tempos precisos. Além de executarem as tarefas de controle e processamento de informações, eles possuem a característica de que suas respostas ao ambiente devem ser dadas em um tempo hábil o suficiente para que o sistema não entre em um estado de inconsistências em seu processamento, que possam ocasionar em erros.

Um sistema de tempo real é, portanto, o software que gerencia os recursos computacionais do microcontrolador (ou computador embarcado), com o objetivo de garantir que todos os eventos sejam atendidos dentro de suas restrições de tempo, e gerenciados da forma mais eficiente possível. O software responsável pelo gerenciamento dos recursos computacionais também é chamado de *Kernel* (ou núcleo) do Sistema de Tempo Real.

7.5.1 Kernel de Tempo Real

O Kernel de um sistema operacional, seja real-time ou não, consiste em seu núcleo que tem como função gerenciar os recursos do hardware e permitir que as aplicações tenham acesso a este de forma concorrente e segura, sem que um processo atrapalhe a execução de

outro. O Kernel cria uma camada de abstração entre o hardware e as aplicações de forma que cada uma acredite que possui todos os recursos do computador à sua inteira disposição. Os principais recursos que o Kernel gerencia são o tempo de uso da CPU, acesso à memória do sistema, controle de uso dos dispositivos de entrada e saída e disponibilização, aos processos executados no espaço de usuário do sistema (*user space*), de uma interface padronizada de atendimento de seus serviços. O Kernel real-time fornece uma camada de abstração que “esconde” dos aplicativos os detalhes de hardware do processador, ou processadores em sistemas multiprocessados ou em processadores multi-core, como mostra a figura 7.11.

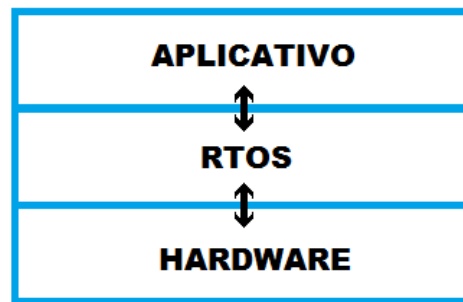


Figura 7.11 – Camada criada por um RTOS entre o aplicativo e o Hardware.

Dentro dessa camada de abstração, o Kernel irá fornecer de três a cinco tipos básicos de serviços, conforme mostra a figura 7.12, que serão disponibilizados para os aplicativos rodando sobre ele.

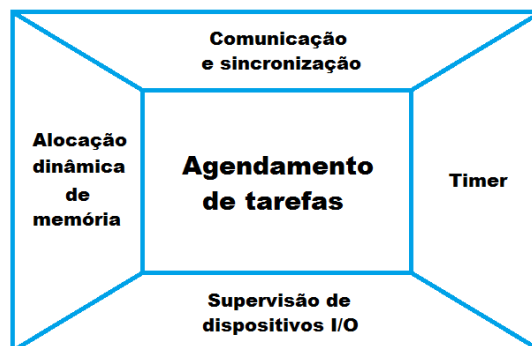


Figura 7.12 – Diagrama com as cinco classes de serviços básicos de um Kernel.

Esses serviços são: gerenciamento de tarefas (*task management*), comunicação e sincronização entre tarefas (*intertask communication and synchronization*), serviço de timers, alocação dinâmica de memória (*dynamic memory allocation*) e supervisão de dispositivos de

entrada e saída (*device I/O supervisor*). Dentre os serviços listados os dois últimos nem sempre estarão disponíveis.

O gerenciamento de tarefas consiste no “coração” de um sistema real-time e representa seu serviço mais básico e fundamental. Esse tipo de serviço permite que os desenvolvedores de aplicativos projetem seus softwares como uma soma de “pedaços” de código, de forma que cada um possa ter seu próprio motivo, seu próprio objetivo e, muitas vezes, seu próprio prazo de execução. Cada um desses pedaços de código é chamado de tarefa (task). O serviço de gerenciamento de tarefas permite iniciar e controlar a execução de cada tarefa, assim como determinar o nível de prioridade associado a cada uma delas. O principal serviço aqui será o agendamento de cada tarefa durante a operação do sistema. O agendador de tarefas irá controlar a execução dos aplicativos de forma previsível, em relação aos tempos de execução, e priorizada, em relação às tarefas executadas.

A utilização de um sistema operacional multitarefa permite simplificar o projeto do código da aplicação. A capacidade multitarefa junto com recursos de comunicação entre tarefas possibilitam que uma aplicação complexa possa ser quebrada em inúmeras atividades menores e mais simples. Essa quebra em diversas atividades permite que partes do software possam ser testadas individualmente além de facilitar a reutilização de código. Todo o trabalho de temporização e seqüenciamento passa a ser responsabilidade do sistema operacional e não da aplicação. Um processador simples somente pode executar uma tarefa por vez, mas com a utilização de chaveamento entre tarefas, irá parecer que várias tarefas estão sendo executadas ao mesmo tempo. Quando o processador utilizado possuir vários núcleos, múltiplas tarefas poderão, de fato, serem executadas concorrentemente.

A política de agendamento de tarefas é determinada por um algoritmo que tem como função determinar qual tarefa deve ser executada a qualquer momento. Um sistema que não seja real-time irá alocar um tempo “razoável” de processador para cada tarefa. Esse

comportamento não é admissível para sistemas real-time. A maioria dos sistemas operacionais real-time utiliza um esquema de agendamento conhecido como agendamento preemptivo baseado em prioridade. Cada tarefa necessária para uma aplicação terá sua prioridade determinada de forma que as tarefas que necessitem resposta mais rápida possuam prioridades mais altas. Respostas rápidas do sistema serão possíveis devido à natureza preemptiva do agendamento. Isso significa que o agendador poderá interromper qualquer tarefa em execução para que outra de maior prioridade seja executada.

Por exemplo, se uma tarefa de menor prioridade estiver em execução e uma outra de maior prioridade ficar pronta para execução, o sistema de agendamento preemptivo baseado em prioridade irá agir de forma a permitir que a tarefa de baixa prioridade termine a execução da instrução assembler que estiver sendo executada, mas não permitirá a execução de uma linha completa de código em alto nível. O sistema irá parar a execução da tarefa de baixa prioridade e iniciar a execução do código de mais alta prioridade. Após o término da tarefa de mais alta prioridade, o agendador irá viabilizar a continuação da tarefa de baixa prioridade. É claro que esse processo irá se repetir sempre que uma atividade de maior prioridade requisitar sua execução. A de mais baixa prioridade somente será executada quando todas de prioridade superior forem completadas. Esse comportamento é visto na figura 7.13.

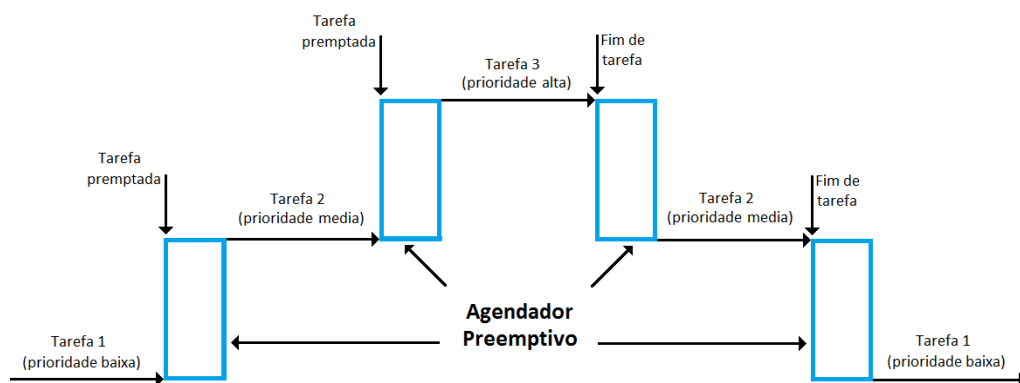


Figura 7.13 – Exemplo de Troca de Tarefas para um Agendador Preemptivo Baseado em Prioridade.

Sempre que o sistema for alternar de uma tarefa para outra de maior prioridade, ele deverá observar as seguintes etapas:

- verificar se a tarefa em execução deverá continuar sendo executada, ou não;
- verificar se a tarefa em execução deverá continuar sendo executada, ou não;
- salvar todas as condições necessárias para que a tarefa interrompida possa continuar sua execução posteriormente;
- preparar o ambiente para a realização da próxima tarefa;
- permitir o início da execução da tarefa.

Essa seqüência é conhecida como *task switching*, ou troca de tarefas. Na avaliação de um sistema operacional real-time é importante conhecer o tempo necessário para a execução do task switching. Um sistema operacional de uso geral somente pode executar o task switching em períodos de tempo múltiplos de um valor pré-determinado, no sistema operacional Microsoft Windows 1/64 s ou 1/128 segundos em alguns casos específicos, e para o sistema operacional Linux até 1/1000 segundos, o que poderá gerar atrasos inaceitáveis para sistemas embarcados real-time.

7.5.2 Aplicação do RTOS em Robótica Móvel

O robô recebe estímulos através da leitura de seus sensores que são denominados de eventos, que podem ser internos ou externos, realizam o processamento e produzem uma saída. Um evento com restrição de tempo possui um tempo limite (máximo) de processamento para que sua saída seja gerada.

Um exemplo é que o robô esteja seguindo sua rota a uma velocidade próxima da máxima, e que em dado instante passe uma pessoa em sua linha de percepção pelos sonares, o que exige que o robô pare ou desvie da pessoa em tempo hábil. Outro exemplo é um sistema de radar aeroespacial, que recebe informações de posicionamento das aeronaves, para que possíveis colisões sejam detectadas e evitadas. Se o sistema não tratar essas entradas em suas restrições de tempo, poderá causar uma tragédia. Estes exemplos representam situações

críticas que demandam de sistemas de tempo altamente restritivo, que são chamados de – *Hard Real Time Systems*. O robô deve executar em tempo hábil as tarefas (*threads*):

- Leitura do vetor de sensores de ultra-som;
- Processamento do agente inteligente de software;
- Acionamento e controle de potencia dos motores.

Onde as tarefas de recepção de dados e transmissão de dados entre o robô e sua estação de controle no modo autônomo têm seus tempos condicionados. A tarefa de recepção de dados tem prioridade e execução de 100ms a cada 100ms, enquanto que a tarefa de transmissão de dados (por parte do robô) tem menor prioridade e execução de 100ms a cada 1min. Tarefas são pequenos trechos de programa com responsabilidades específicas e bem definidas. Cada tarefa possui um conjunto de atributos, dentre eles a prioridade da tarefa, que deve ser atribuída de acordo com sua importância. É a partir desta prioridade que se tenta garantir que eventos possam ser executados de forma eficiente. As bibliotecas *RTOS* em teste são o *FreeRTOS* e o *CCS RTOS*.

7.6 Odometria

A odometria é um dos métodos mais usados para estimar a posição de um robô. A tarefa de odometria deve proporcionar uma boa precisão em tempo hábil, e está relacionada ao comportamento interno do robô, através do sensoriamento e integração da informação de maneira incremental do movimento ao longo do tempo, o qual envolve uma inevitável acumulação de erros.

A acumulação de erros de orientação do robô móvel causa grandes erros na estimação da posição, os quais vão aumentando proporcionalmente com a distância percorrida pelo robô. Apesar destas limitações, a odometria é uma parte importante do sistema de navegação de um robô, e que deve se usar com medidas de posicionamento absolutas para proporcionar uma estimativa de posição mais confiável. É comum o uso de encoders incrementais e

absolutos que são fixos ao eixo do robô e medem sua velocidade angular, assim como sua posição. O uso de recursos em localização robótica pode ser usado para corrigir os erros na odometria. E existem três maneiras de se realizar a localização de um robô:

- Localização com Medidas Relativas: Com odometria e navegação inercial.
- Localização com Medidas Absolutas: Uso de reconhecimento de marcos artificiais ou naturais do ambiente (*landmarks*).
- Localização Híbrida: É o uso em conjunto das medidas relativas e absolutas.

Com isto é possível determinar a localização do robô móvel, através da observação e integração consecutiva do movimento das rodas, e assim estimar sua posição no espaço de configuração conforme visto na figura 7.14.

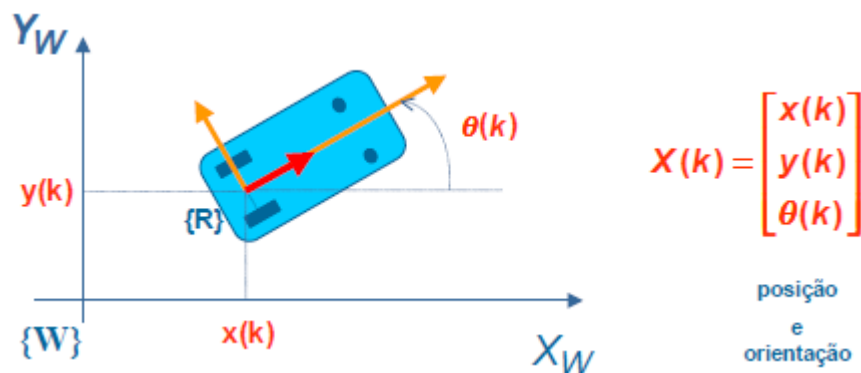


Figura 7.14 – Posição e orientação de um robô móvel em seu espaço de configurações por odometria.

O robô móvel LOGBOT não conta com qualquer mecanismo para realizar odometria, e por conseqüência a tarefa de localização no ambiente em que evolui. Suas funções são restritas a navegação autônoma em corredores.

7.7 Comunicação de Dados em Robótica

O sistema de comunicação de um robô deve ser preferencialmente sem fio, e, além disto, deve ser um canal confiável de comunicação, que possa promover uma alta taxa de dados em seu fluxo. O LOGBOT conta com um sistema de comunicação padrão IEEE 802.15.4, mais conhecido por *Zigbee*.

A tecnologia ZigBee permite comunicações robustas e opera na frequência ISM (*Industrial, Scientific and Medical*), sendo na Europa de 868 MHz (1 canal), 915 MHz (10 canais) nos Estados Unidos e 2,4 GHz (16 canais) em outras partes do mundo, e não requerem licença para funcionamento [50]. As Redes *ZigBee* oferecem uma excelente imunidade contra interferências, e a capacidade de hospedar milhares de dispositivos numa Rede (mais que 65.000), com taxas de transferências de dados variando entre 20Kbps a 250Kbps. O Protocolo ZigBee é destinado a aplicações industriais, portanto, o fator velocidade não é crítico numa implementação ZigBee.

Os módulos de comunicação *RF* padrão *ZigBee* foram criados para economizar ao máximo energia. Com isso, é possível criar dispositivos sensores remotos alimentados com pilhas ou baterias comuns, que durarão meses ou mesmo anos sem precisarem ser substituídas. Isso porque, os módulos *ZigBee* quando não estão transmitindo/recebendo dados, entram num estado de dormência ou em "*Sleep*", consumindo o mínimo de energia. O protocolo de comunicação desenvolvido para o robô LOGBOT é descrito na figura 7.15.

Envio dos dados para a Estação de Controle (Robô -> PC)

\$ST00&T35&H90*

Envio da String de Controle para o Robô (PC -> Robô)

#MR&X&V*

Figura 7.15 – Protocolo de comunicação do Robô Móvel LOGBOT.

No sentido de comunicação do robô para o PC, a string de comunicação tem seu seguinte significado em cadeia:

→ \$ - Início de Transmissão de sentido Robô/PC

→ **ST00** – Modulo de sensoriamento STIM de endereço 00.

- **&** - Entrada de uma subcadeia de caracteres para identificação.
- **T35** – Sensor de Temperatura com leitura de 35°C.
- **&** - Entrada de uma subcadeia de caracteres para identificação;
- **H90** – Sensor de Umidade com leitura de 90%.
- ***** - Fim da cadeia principal da String.

No sentido de comunicação do PC para o robô:

- **#** - Início de Transmissão de sentido PC/Robô.
- **MR** ou **ME** – Motor esquerdo ou direito a ser ativado.
- **&** - Entrada de uma subcadeia de caracteres para identificação.
- **X (H ou A)** – Sentido de direção que o motor deverá executar: horário ou anti-horário.
- **&** - Entrada de uma subcadeia de caracteres para identificação.
- **V** – Velocidade do motor: 0 é a mínima, e 255 é a máxima.
- ***** - Fim da cadeia principal de String.

A figura 7.16 mostra o funcionamento da comunicação entre o operador robô.

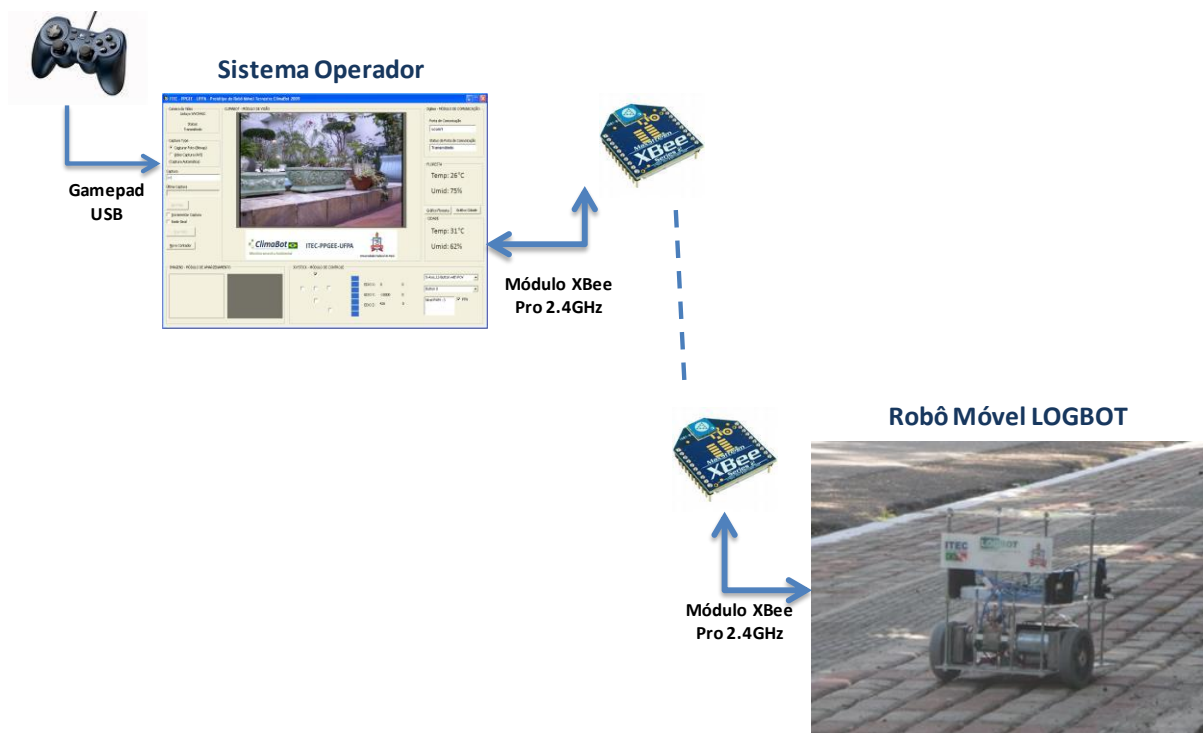


Figura 7.16 – Diagrama completo de comunicação entre o robô e a estação de controle (PC).

A comunicação se apresentou bastante consistente e confiável, e em área fechada (corredor do ITEC) conseguiu realizar comunicação com vários obstáculos em até 60 metros de distância, e em área aberta da reitoria até a orla do rio Guamá, a comunicação se apresentou estável e não houve problemas relacionados a inconsistências de dados em até 200 metros de distância.



Figura 7.17 – Mapa do Setor Básico, que mostra onde o teste foi realizado (Campus da UFPA).



Figura 7.18 – Orla da UFPA no Rio Guamá, e Robô na Base da Capela Ecumênica.

Capítulo 8: Resultados Experimentais

Este capítulo apresenta os resultados obtidos com o robô no modo autônomo com o controlador proporcional de trajetórias descrito no capítulo 5 desta dissertação.

8.1 Resultados Experimentais no Modo Autônomo

Os resultados no Modo Autônomo para Navegação Wall-Following foram obtidos pela lei de controle:

$$U = 0.4 \times E_D + 1.26 \times \hat{\theta} \quad (8.1)$$

A sintonia do Controlador proporcional de trajetórias é vista no capítulo 5 desta dissertação. A referência de distância a qual o LOGBOT deve manter-se da parede esquerda é de 40 cm. Ao evoluir no ambiente autonomamente o robô envia uma matriz de aquisição de dados do controlador com as informações de – distância atual com a parede, erro de referência, orientação atual, ação de controle, velocidade PWM da roda esquerda, e velocidade PWM da roda direita. A análise é off-line, ou seja, após o robô ter realizado todos os procedimentos. Uma amostra da matriz de aquisição de dados é apresentada na tabela 8.1.

Tabela 8.1 – Matriz de Aquisição de Dados do Controlador Wall-Following do LOGBOT.

Y_D	<i>Erro</i>	θ	U	VRE_{PWM}	VRD_{PWM}
25.44	14.55	-2.54	5.93	35.93	24.06
23.58	16.41	5.08	20.0	50.0	10.0
24.82	15.17	0.0	10.61	40.61	19.38
24.82	15.17	-15.43	-15.2	14.79	45.2
25.44	14.55	-12.81	-11.25	18.74	41.25
26.69	13.3	-7.64	-3.48	26.51	33.48
27.31	12.68	-5.08	0.36	30.36	29.63
27.31	12.68	0.0	8.88	38.88	21.11
27.31	12.68	0.0	8.88	38.88	21.11
27.93	12.06	-2.54	4.19	34.19	25.8
...

Para teste do controlador proporcional de trajetórias Wall-Following do robô móvel LOGBOT são organizados testes para problemas de navegação Wall-Following, cujo algoritmo de controle deve ser capaz de vencer. Os ensaios para os testes são organizados da seguinte forma:

- Teste I: Posição do robô 10 cm acima da distância de referência em 40 cm. O robô neste teste deve seguir uma trajetória, de modo a aproximar-se da parede até o instante em que chegar a uma distância de 40 cm, a partir deste momento o robô mantém sua trajetória reta a distância de referência com a parede.
- Teste II: Posição do robô 10 cm abaixo da distância de referência em 40 cm. O robô neste teste deve seguir uma trajetória, de modo a distanciar-se da parede até o instante em que chegar a uma distância de 40 cm, a partir deste momento o robô mantém sua trajetória reta a distância de referência com a parede.
- Teste III: Posição do robô dentro da distância de referência em 40 cm, mas que deve reagir a mudanças bruscas de posição e orientação, como o vão de uma porta fechada. O robô neste teste deve seguir um conjunto de trajetórias, de modo a vencer as bruscas mudanças do vão de uma porta, sem que perca a distância de referência com a parede.
- Teste IV: Posição do robô dentro da distância de referência em 40 cm, mas que deve reagir a mudanças bruscas de posição e orientação, como a mudança de parede no corredor. O robô neste teste deve seguir um conjunto de trajetórias, de modo a vencer as bruscas mudanças da mudança de parede no corredor, sem que perca a distância de referência com a parede.

Todos os testes são realizados a uma velocidade de $0,18m/s$ ou $30\% PWM$.

8.1.1 Teste I – 10 cm Acima da Referência de Distância com a Parede (Erro Negativo)

O robô inicia a sua navegação no ambiente de atuação posicionado 10 centímetros acima da referência de distância com a parede conforme mostra a figura 8.1. Ao passo que o robô navega e evolui no corredor, o controlador deve agir de forma que a velocidade de tração diferencial das rodas direita e esquerda faça com que o robô se aproxime da parede.

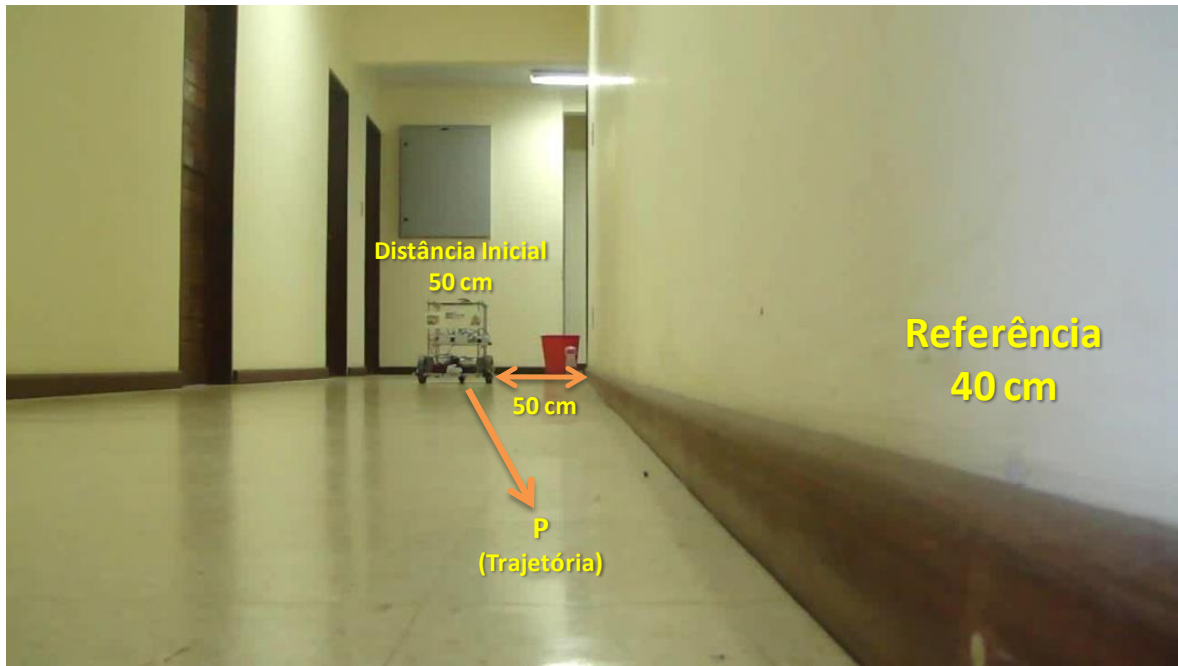


Figura 8.1 – Correção da posição do Robô Acima da Referência.

O controlador reduz o erro à zero, e entra em regime permanente mantendo o robô na distância de referência com a parede no seguimento de sua trajetória conforme mostra a figura 8.2(a).

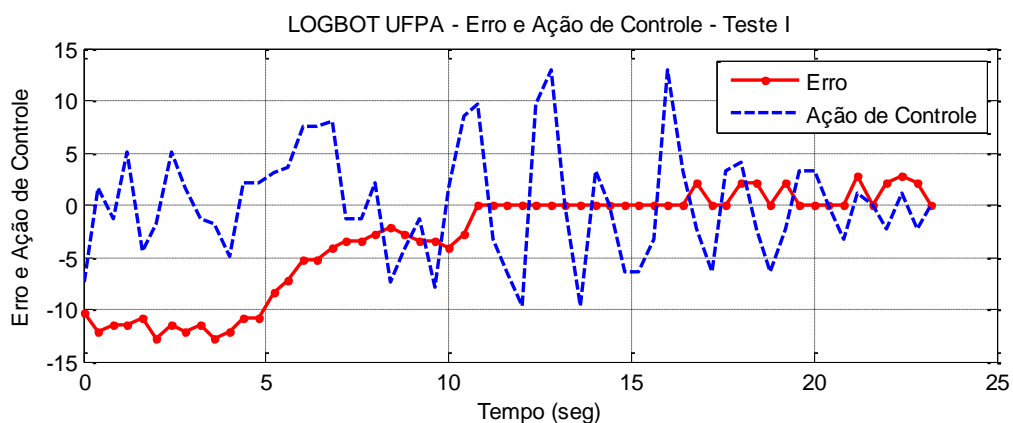


Figura 8.2 (a) – Teste I: Erro e Ação de Controle.

A figura 8.2(b) mostra a redução do erro a partir da distância atual do robô com a parede, e que entre 7 e 10 segundos há uma inconsistência de leitura dos sonares, o erro passa a ser corrigido a partir de 4 segundos e alcança o regime permanente a partir de 12 segundos, onde é notável a ação da zona morta até 17 segundos.

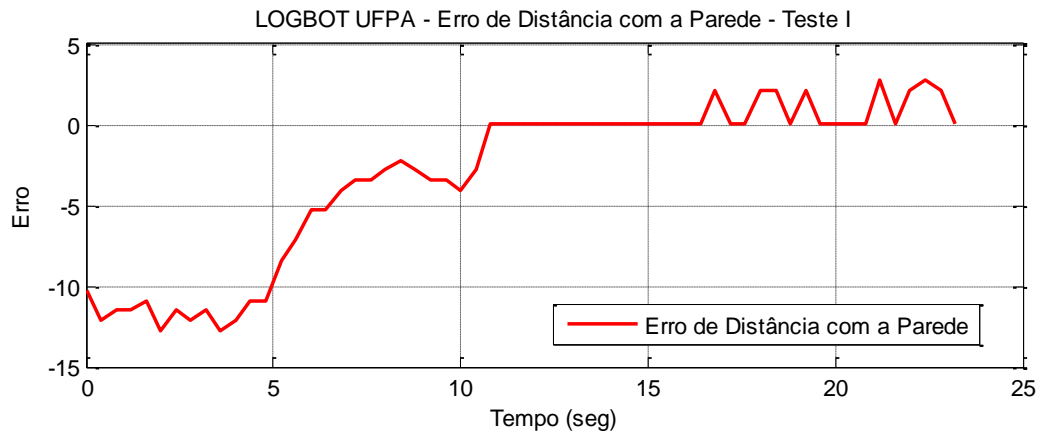


Figura 8.2 (b) – Teste I: Correção do erro de distância com a parede.

A figura 8.2(c) mostra a ação diferencial entre os motores na tarefa para o robô aproximar-se da parede, o motor direito recebe um sinal PWM de aproximadamente 45%, enquanto que o motor esquerdo recebe um sinal PWM de aproximadamente 15% entre os segundos entre 4 e 7 segundos. Logo é notável a ação diferencial, cuja roda direita tem velocidade maior que a roda esquerda, isto faz com que o robô se aproxime da parede.

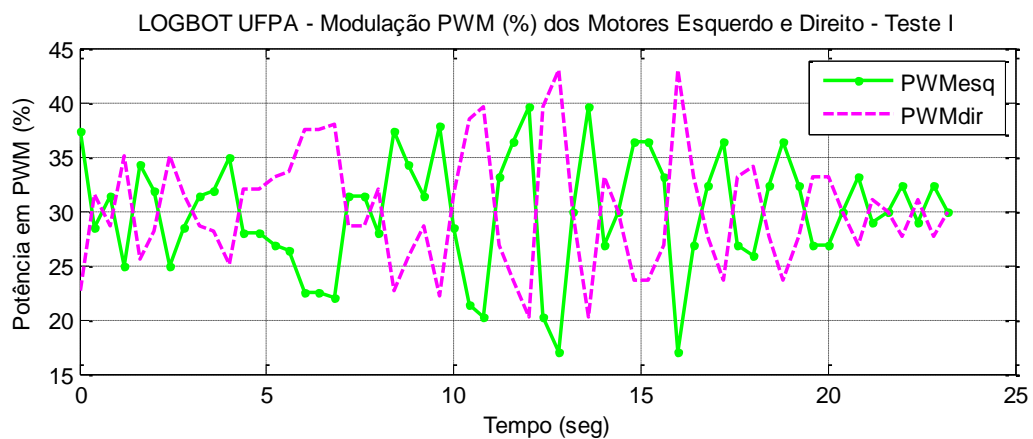


Figura 8.2 (c) – Teste I: Sinal de controle aplicado às velocidades (PMW) dos motores.

8.1.2 Teste II – 10 cm Abaixo da Referência de Distância com a Parede (Erro Positivo)

O robô inicia a sua navegação no ambiente de atuação posicionado 10 centímetros abaixo da referência de distância com a parede conforme mostra a figura 8.3. Ao passo que o robô navega e evolui no corredor, o controlador deve agir de forma que a velocidade de tração diferencial das rodas direita e esquerda faça com que o robô se afaste da parede.

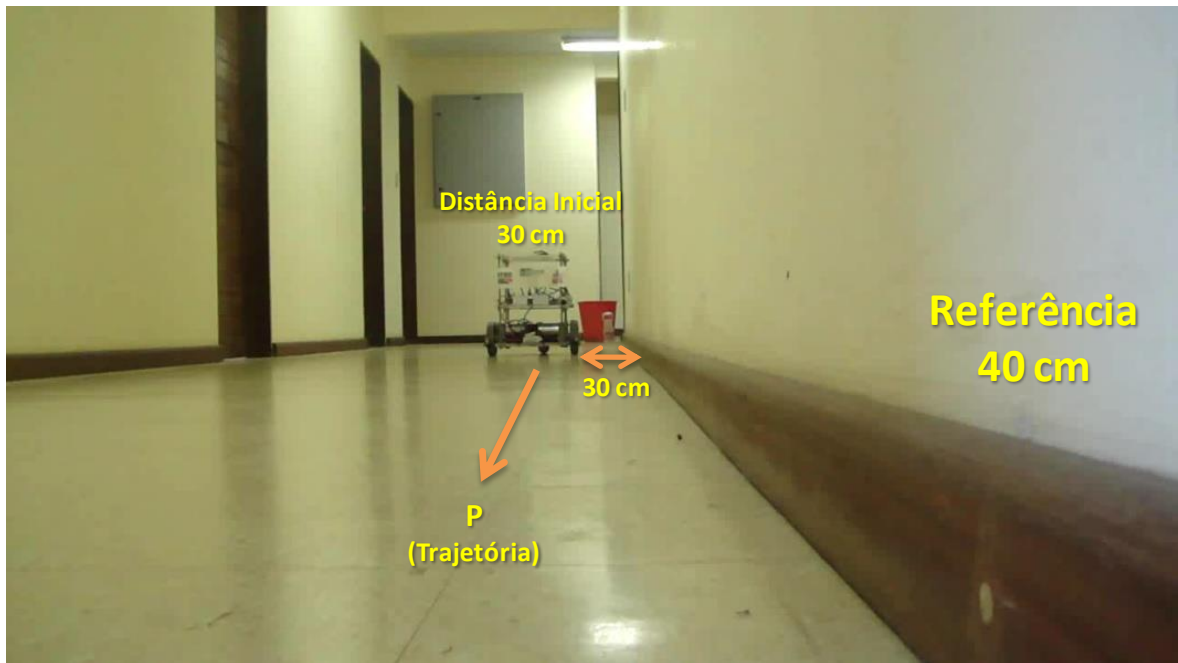


Figura 8.3 – Correção da posição do Robô Abaixo da Referência.

O controlador chega próximo de reduzir o erro à zero, e oscilando próximo da distância de referência, e mantendo o robô próximo da distância de referência com a parede no seguimento de sua trajetória conforme mostra a figura 8.4(a).

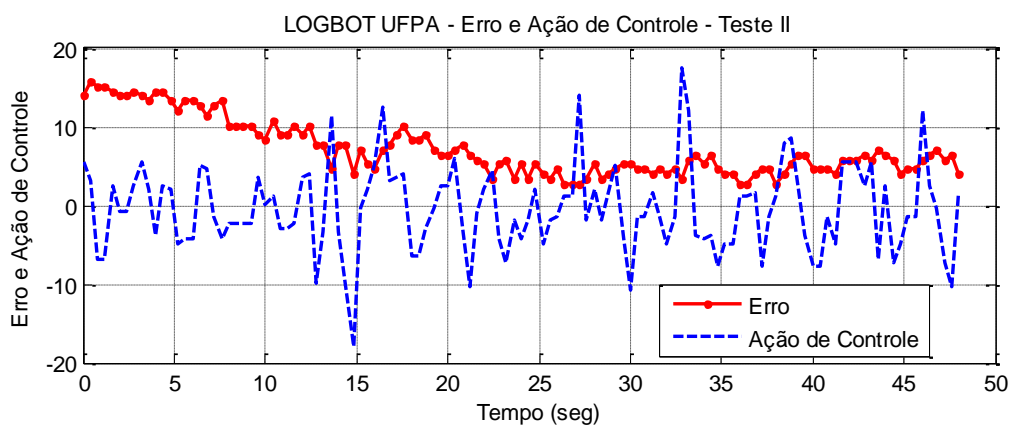


Figura 8.4 (a) – Teste II: Erro e Ação de Controle.

A figura 8.4(b) mostra a redução do erro a partir da distância atual do robô com a parede, e que entre 4 e 14 segundos, o erro passa a ser corrigido, sendo que a partir de 22 segundos alcança o regime oscilatório, cuja a distância tem variações próximas da referência.

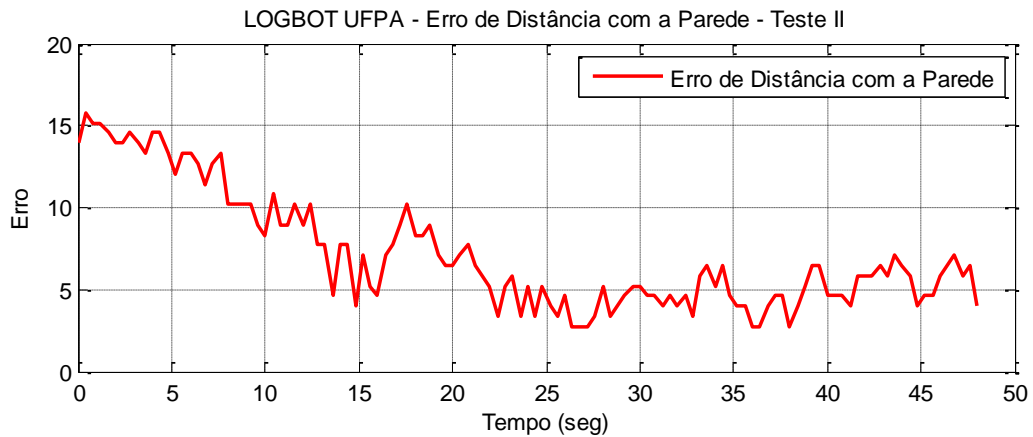


Figura 8.4(b) – Teste II: Correção do erro de distância com a parede.

A figura 8.4(c) mostra a ação diferencial entre os motores na tarefa para o robô aproximar-se da parede, o motor direito recebe um sinal PWM de aproximadamente 10%, enquanto que o motor esquerdo recebe um sinal PWM de aproximadamente 50% entre 14 e 16 segundos devido a inconsistência de leitura dos sonares. Mas a partir de 26 segundos torna-se estável. Logo é notável a ação diferencial, cuja roda esquerda tem velocidade maior que a roda direita, isto faz com que o robô se afaste da parede.

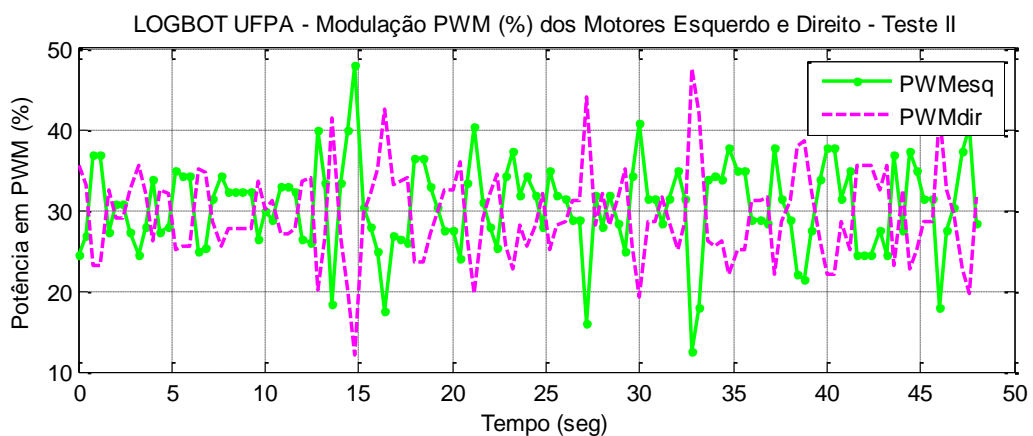


Figura 8.4(c) – Teste II: Sinal de controle aplicado às velocidades (PMW) dos motores.

8.1.3 Teste III – Mudança Brusca de Posição e Orientação: Vão de uma Porta

O robô inicia a navegação no ambiente de atuação posicionado dentro da margem de referência de distância com a parede conforme mostra a figura 8.5. Ao passo que o robô navega e evolui no corredor, o controlador deve agir de forma que a velocidade de tração diferencial das rodas direita e esquerda faça com que o robô contorne o vão de uma porta.



Figura 8.5 – Correção da posição do Robô com a Problemática do Vão da Porta.

O controlador reduziu o erro à zero, entrando em regime permanente mantendo o robô na distância de referência com a parede no seguimento de sua trajetória conforme mostra a figura 8.6(a).

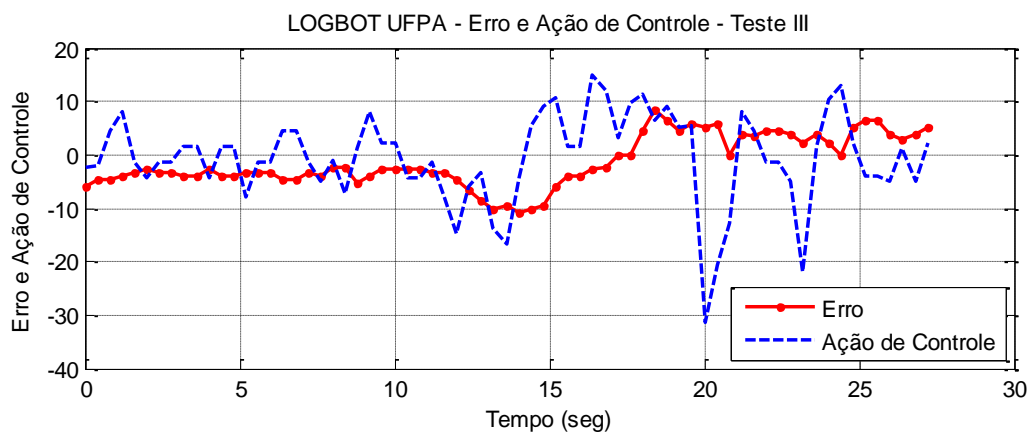


Figura 8.6(a) – Teste III: Erro e Ação de Controle.

A figura 8.6(b) mostra a redução do erro a partir da distância atual do robô com a parede, e que entre 13 e 20 segundos há a brusca variação de distância da parede devido ao vão da porta. O erro passa a ser corrigido a partir de 4 segundos e contorna o vão da porta de maneira suave e sem colisões com a porta ou a parede.

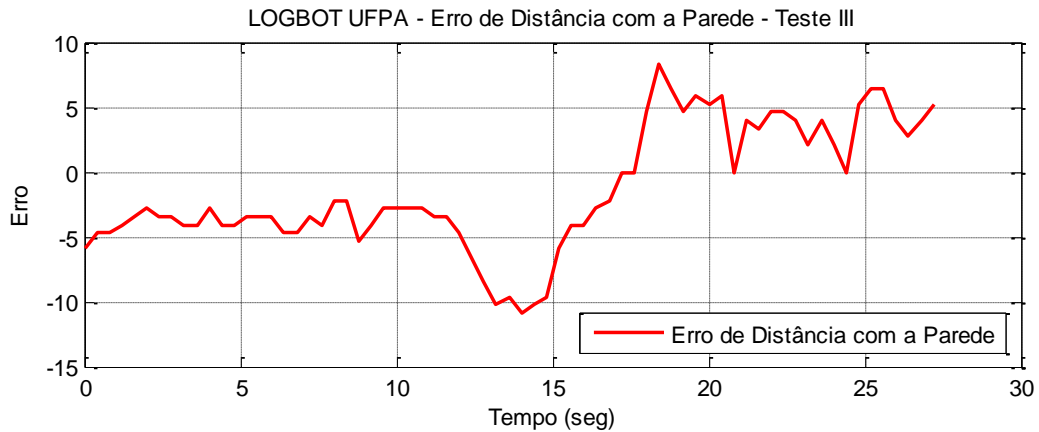


Figura 8.6(b) – Teste III: Correção do erro de distância com a parede no vão de uma porta.

A figura 8.6(c) mostra a ação diferencial entre os motores na tarefa para o robô aproximar-se da parede, o motor direito recebe um sinal PWM de aproximadamente 60%, enquanto que o motor esquerdo recebe um sinal PWM de aproximadamente 10% entre os segundos 19 e 22. Logo é notável a ação diferencial, que mostra várias comutações de velocidades em que uma roda tem velocidade maior que a outra por breves instantes de tempo, de forma a se adaptar rapidamente as bruscas mudanças causadas pelo vão de uma porta.

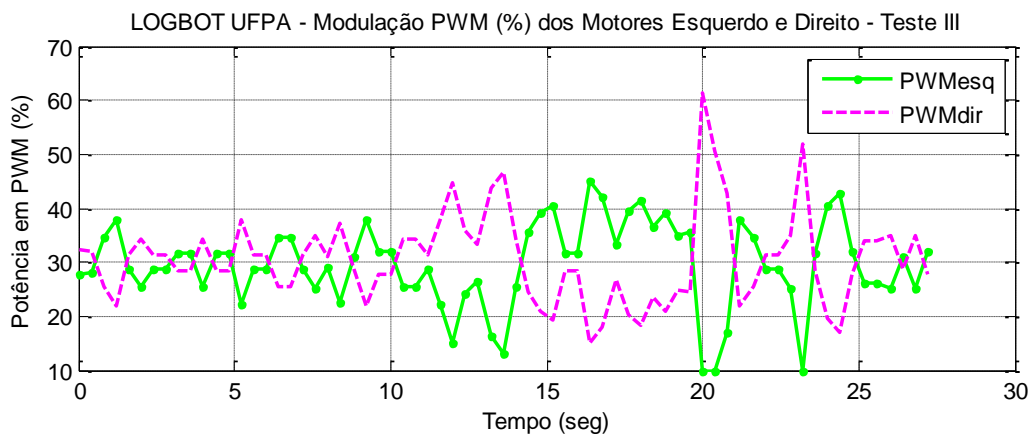


Figura 8.6(c) – Teste III: Sinal de controle aplicado às velocidades (PMW) dos motores.

8.1.4 Teste IV – Variação Brusca de Posição e Orientação: Mudança de Parede no Corredor

Neste teste o robô teve de contornar o corredor. O robô iniciou sua navegação no ambiente de atuação dentro da margem de referência de distância com a parede, e deve agir de forma que realize a mudança de parede conforme mostra a figura 8.7.

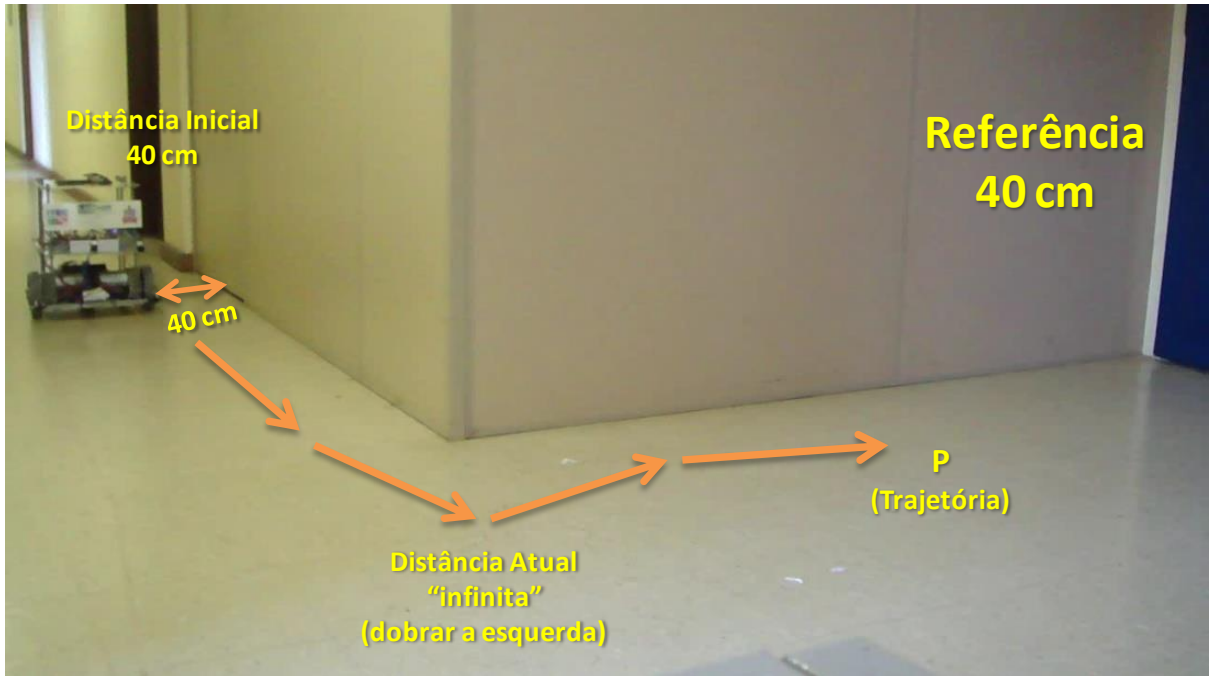


Figura 8.7 – Correção da posição do Robô com o contorno do final do corredor.

O controlador reduziu o erro à zero, e entra em regime permanente mantendo o robô na distância de referência com a parede na mudança de parede, e no seguimento de sua trajetória conforme mostra a figura 8.8(a).

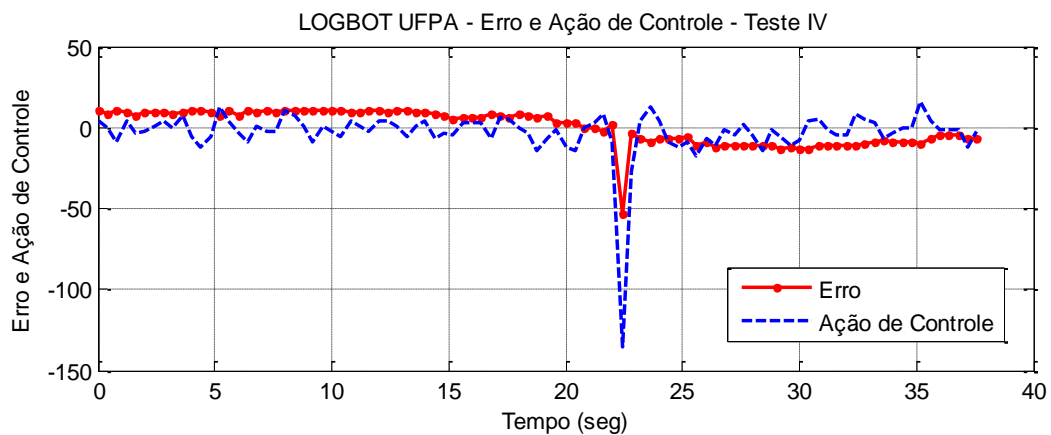


Figura 8.8(a) – Teste IV: Erro e Ação de Controle.

A figura 8.8(b) mostra a redução do erro a partir da distância atual do robô com a parede, e que entre 22 e 23 segundos há a brusca variação causada pela mudança de parede que gera uma “distância infinita” conforme descrita na figura 8.7. Após superar a mudança de parede o erro passa a ser corrigido a partir de 24 segundos e oscila próximo à distância de referência.

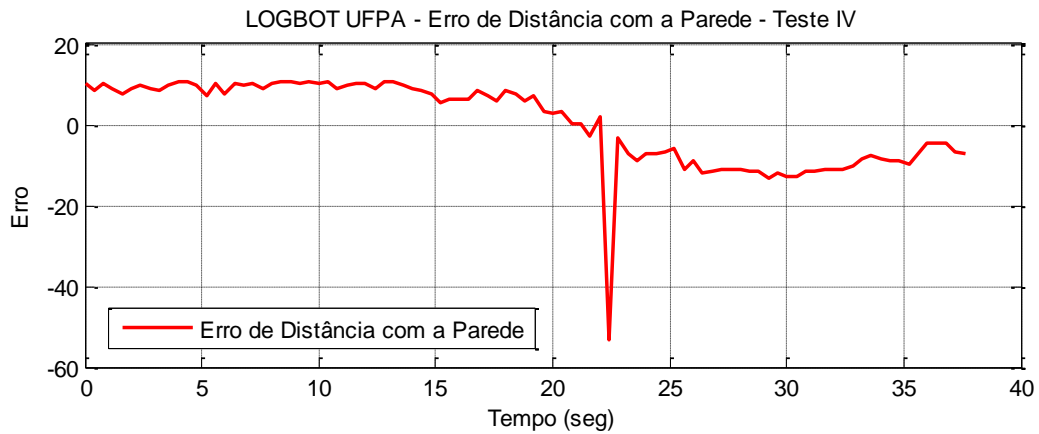


Figura 8.8(b) – Teste IV: Correção do erro de distância da com a parede na mudança de parede.

A figura 8.8(c) mostra a ação diferencial entre os motores na tarefa para o robô aproximar-se da parede, o motor direito recebe um sinal PWM de aproximadamente 100%, enquanto que o motor esquerdo recebe um sinal PWM de aproximadamente 15% entre os segundos 22 e 23, que é quando ocorre a mudança de parede. Logo é notável a ação diferencial, cuja roda direita tem velocidade máxima, e a roda esquerda velocidade próxima de mínima, isto faz com que o robô dobre a esquerda sem movimentos bruscos na mudança de parede do corredor onde atua.

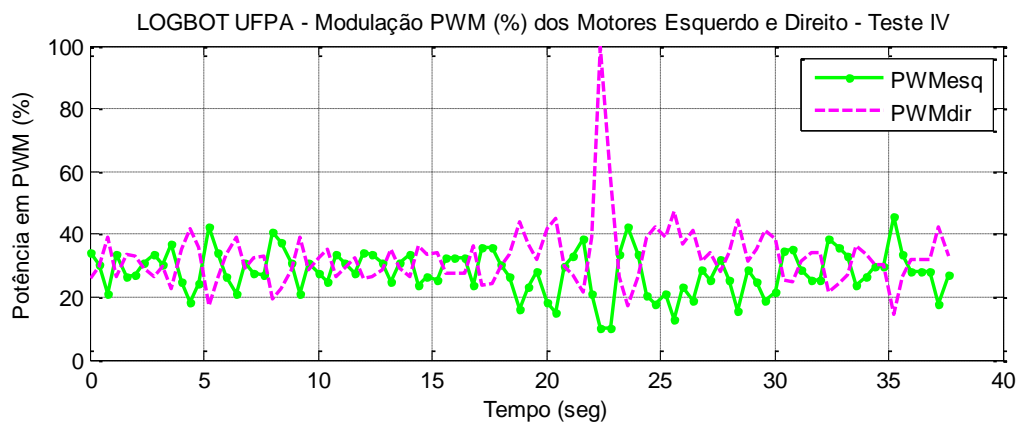


Figura 8.8(c) – Teste IV: Sinal de controle aplicado às velocidades (PMW) dos motores.

8.2 Análise dos Resultados Experimentais no Modo Autônomo

Para as tarefas – corrigir os erros de distância da parede, detectar a problemática da porta, e contornar o final corredor. O robô apresentou resultados satisfatórios e em tempo hábil de processamento, através do uso do controlador de posição e orientação de robôs móveis descrito no capítulo 5.

Em Wall-Following comportamentos como obstáculos encostados a parede, vãos de portas e mudanças de corredor devem ser superados e depois corrigidos, mesmo com brusca variação da medida da configuração do robô, de forma que a ação de controle não seja alta o suficiente de forma que o robô não se perca da referência e fique girando a procura de uma parede, e a ação de controle não seja baixo o suficiente de forma que o robô colida com a parede. Caso a porta esteja aberta, o robô tratará este comportamento como uma mudança de parede.

Capítulo 9: Conclusões e Perspectivas Futuras

O principal objetivo desta dissertação se concentra no projeto e construção do protótipo de um robô móvel autônomo e operado remotamente, para uso como uma plataforma didática e de desenvolvimento nas linhas de pesquisa – robótica móvel, automação, algoritmos de programação, controle e inteligência artificial. O projeto global do robô é de baixo custo, cuja aquisição de todos os seus componentes foi feita com fornecedores nacionais. A plataforma denominada de LOGBOT – **LOGÍSTICA ROBOT** é robusta, e suas funções apresentam bom desempenho, comparável ao de plataformas comerciais.

A presente dissertação mostra a validação dos cálculos teóricos e a identificação de restrições físicas e cinemáticas da estrutura do robô móvel LOGBOT. Estas restrições são fundamentais para o planejamento e modelo do controlador de trajetórias. O cálculo de consumo para fontes de energia embarcadas e cuidados no uso de baterias de diversos tipos é também mostrado e validado para uso no robô.

O projeto é um arcabouço de técnicas, teorias e métodos que permitem integrar eficientemente esquemas de fusão sensorial, mecanismos de controle de atenção e técnicas de redução de informação para sistemas de percepção robótica, visando a sua aplicação em missões de monitoração, navegação e inspeção.

No modo de funcionamento teleoperado – o robô realiza a aquisição remota de dados das variáveis ambientais de temperatura e umidade do ar, cujos testes foram realizados na orla do rio Guamá na Universidade Federal do Pará, e no Parque Florestal da Universidade Federal Rural da Amazônia.

No modo de funcionamento autônomo – o robô conta com um vetor de sensores de ultra-som para auxiliá-lo na tarefa de navegação autônoma em ambientes fechados, através do algoritmo de navegação Wall-Following.

A primeira fase do projeto busca desenvolver uma metodologia para projeto e construção de um robô móvel, que consiste na escolha de – materiais para confecção da estrutura, motores com torque e velocidade robustos, e o projeto e integração do hardware embarcado. A segunda fase do projeto busca desenvolver métodos para o controle de robôs móveis em dois modos de operação – O modo *ROV (Remotely Operated Vehicle)* o robô funciona como um veículo operado remotamente através de telemetria, cuja ação de controle é realizada pelo ser humano que através de uma estação de controle, que envia os comandos de atitude para o robô móvel LOGBOT. No modo *AGV (Autonomous Guided Vehicle)* o robô é autônomo no em seu controle e navega de maneira autônoma no ambiente que o circunda, de forma que desvie de obstáculos e não se perca.

Os estudos também mostram que é possível realizar o controle do robô móvel, através de controladores lineares em cascata (organização mestre/escravo) conforme visto no Capítulo 5, mesmo que seja um sistema multivariável, não integrável, não-holonômico, e não linear. O controle do sistema por partes alcançou estabilidade satisfatória, cujos erros de posição e orientação são reduzidos assintoticamente a zero.

A sintonia do controlador proporcional de trajetórias do robô como um todo é realizada através do método experimental de *Kuipers* para uso no modo autônomo. Os ganhos do sistema mesmo sintonizados experimentalmente foram escolhidos de forma a não aumentar o tempo morto no acionamento dos motores, e inércia da estrutura do robô, o que diminui drasticamente a estabilidade e o desempenho da ação de controle.

Os estudos realizados durante a elaboração da segunda fase do projeto LOGBOT indicaram que a melhor forma de controlar um robô é através do encapsulamento das técnicas de controle em um agente inteligente software conforme mostra o capítulo 6. Cada técnica inserida o agente é tratada como uma tarefa e/ou ferramenta que dependendo do comportamento detectado pode ser ativado pelo agente, e assim o robô passa a adquirir maior

precisão em sua navegação e localização no ambiente. O Agente desta forma funciona como um escalonador de tarefas, e em tempo hábil conseguiu selecionar a melhor tarefa a ser executada durante sua evolução no mundo que o circunda.

É notável que a questão da modularidade seja intrínseca ao nível do hardware, do controlador proporcional de trajetórias, e do agente inteligente, ou seja, em todas as camadas há uma divisão em módulos que podem ser ativados e desativados conforme o comportamento do robô em seu ambiente de atuação. As vantagens disto é que o sistema tem arquitetura aberta, flexível, incremental e de fácil manutenção.

A principal missão do projeto LOGBOT foi cumprida e visa o desenvolvimento de um robô móvel que fosse controlado remotamente por telemetria e que navegasse autonomamente em ambientes fechados. O software para controle autônomo (firmware) é embarcado em um microcontrolador com núcleo de 8-bits com frequência de processamento em 4MHZ. O motivo para se trabalhar com esta limitação é desenvolver algoritmos de baixo custo computacional e com alto desempenho para as várias tarefas que um robô móvel pode executar. Desta forma a missão de se desenvolver um arcabouço de hardware e software para robôs móveis de baixo custo é cumprida. Os algoritmos são validados para serem executados em plataformas computacionais mais robustas.

Nos testes de desempenho de suas funções o robô móvel LOGBOT navegou de maneira autônoma por pelo menos dez minutos no corredor de testes nos corredores do prédio da engenharia elétrica na Universidade Federal do Pará, e contornou obstáculos e realizou mudanças de parede ao decorrer dos corredores.

O robô também cumpriu as tarefas iniciais de navegar no interior do parque experimental de manejo florestal da Universidade Federal do Pará – UFPA, situado na periferia de Belém e próximo do campus da UFPA. Durante a navegação o LOGBOT coletou dados ambientais referentes à temperatura e à umidade relativa do ar.

No estágio atual, o projeto LOGBOT encontra-se na fase de prototipagem e constitui um projeto de baixo custo, cuja ênfase didática é a aplicação de tecnologias para solução e prevenção de problemas do meio ambiente, principalmente em regiões como a Amazônia, que devido suas particularidades biológicas demanda de sistemas exclusivos para o seu monitoramento ambiental. Em futuro próximo pretende-se incorporar ao robô LOGBOT outros sensores, tais como anemômetros, detectores de poluição do ar, medidores de pressão atmosférica, que integrados ao sistema de aquisição de dados do robô. O projeto tem as seguintes perspectivas futuras:

- Construção de uma estrutura robusta;
- Aquisição de hardware e sensores mais robustos;
- Uso de algoritmos mais avançados para navegação e localização robótica;
- Validar o robô como uma plataforma didática em robótica móvel;

Esses tópicos são relevantes e constituem possibilidades para temas para trabalhos de conclusão de curso e dissertações de mestrado. O projeto *LOGBOT* foi concluído a contexto, e suas perspectivas são promissoras com relação a aplicações na educação e meio ambiente, e principalmente no desenvolvimento de tecnologia nacional. Com isso é necessário a elaboração e a execução de outros projetos de aperfeiçoamento ou de desenvolvimento de aplicações específicas.

O projeto LOGBOT teve boa aceitação na apresentação de seu artigo “Robô de Inspeção para Aquisição de Dados Ambientais na Região com o Uso da Norma IEEE1451” no *IX SBAI* em Brasília-DF, por grupos de pesquisa com experiência em desenvolvimento da robótica no Brasil.

Referências Bibliográficas

Referências de todos os capítulos desta dissertação.

- [1] Gates, B., “Humans and Robots – Future Perspectives”. Scientific American Magazine, n°.25, PP. 3-13, USA, 2007.
- [2] Cornwell, J., “Hitler’s Scientists – Science, War and the Devil’s Pact”. Penguin Books Ltd, UK, 2003.
- [3] Siegwart, R. & Nourbakhsh, I., “Introduction to Autonomous Mobile Robots”. A Bradford Book The MIT Press, EUA, 2004.
- [4] Enciclopédia Barsa Ano 1971, “Exploração e Tecnologias Espaciais”. Capítulo 12, Brasil, 1972.
- [5] Nemiroff, R. & Bonnell, J., “Lunokhod: Moon Robot”. Página-eletrônica: <<http://apod.nasa.gov/apod/ap990109.html>>. Capturado em Set. 2009.
- [6] Brown, D. & Webster, G., “Mars Exploration Rovers”. Página-eletrônica: <http://www.nasa.gov/mission_pages/mer>. Capturado em Set. 2009.
- [7] Soares, M., “Final Feliz na Rússia”. Revista Mecatrônica Fácil N°25, Editora: Nova Saber, Brasil, 2005.
- [8] Pieri, E., “Curso de Robótica Móvel”. Livro da disciplina Robótica Móvel para graduação e pós-graduação, EDUFSC, Brasil, 2002.
- [9] Stein, P. S. e Reis, N. R. S., “Sistema integrado de navegação para o Robô Ambiental Híbrido na Floresta Amazônica”. Anais do XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, Brasil, 2007.
- [10] Pereira, G. & Chaimowicz, L., “Capítulo 13 - Robôs Móveis”. Enciclopédia de Automática – Controle & Automação, Vol. 3, Brasil, 2008.
- [11] Vieira, F. C., “Controle Dinâmico de Robôs Móveis com Acionamento Diferencial”. Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte - UFRN, Brasil, 2005.
- [12] Filho, T.F.B, “Oficina de Robótica – Mestrado e Doutorado”, Editora: EDUFES, Edição: 1º, Brasil, 2007.

- [13] Sunada, M. M., “Acionamento de um Posicionador Linear de Ultraprecisão Empregando uma redução Harmonic Drive com Controle de Velocidade” Dissertação de Mestrado, Universidade Federal de Santa Catarina – UFSC, Brasil, 2007.
- [14] Silva, A. P. A., “Capítulo 8 – Sistemas Inteligentes na Automática”. Enciclopédia de Automática – Controle & Automação, Vol. 3, Brasil, 2008.
- [15] Borges, G. A., “Introdução à Robótica Móvel”. Apresentação da I Semana da Mecajun, Brasília-DF, Brasil, 2005.
- [16] Pereira, G. A. S., “Identificação e Controle de Micro-Robôs Móveis”. Dissertação de Mestrado, Universidade Federal de Minas Gerais – UFMG, Brasil, 2000.
- [17] Wen, J. T., “Control of nonholonomic systems”. In Levine, W., editor, Control Handbook. CRC Press, EUA, 1995.
- [18] Figueiredo, L. C. & Jota, F. G., “Introdução ao controle de sistemas não-holonômicos”. Revista SBA Controle & Automação, vol.15, n.3, PP.243-268. ISSN 0103-1759, Brasil, 2004.
- [19] Samson, C., “Control of chained systems application to path following and time-varying point-stabilization of mobile robots”. IEEE Transactions on Automatic Control, 40(1):64-77, EUA, 1995.
- [20] Alves, J. A. V. & Fetter, W. L., “Controle em Tempo Real de Robôs Móveis Não-Holonômicos”. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul – UFRGS, Brasil, 2007.
- [21] Luenberger, D.G., “An introduction to observers”. IEEE Transactions on Automatic Control, 10(6):596-602.
- [22] Lizarralde, F. & Wen, J. T., “Attitude Control without angular velocity measurement: A passivity approach”. IEEE Transactions on Robotics and Automatic Control, 41:468-472, EUA, 1996.
- [23] Turenout, P. V. & G., Schelven, L. J., “Wall-Following Control of a Mobile Robot”. Proceedings of the 1992 IEEE International Conference on Robotics and Automation – ICRA, Nice, France, May 1992.
- [24] Campion, G., Bastin, G., D’Andréa-Novel, B., “Structural Properties and Classification of Kinematic and Dynamical Models of Wheeled Mobile Robots”.

- IEEE Transactions on Robotics and Automation, v.12, n.1, p.47-62, New York, EUA, 1996.
- [25] Kuipers, B., “Robot Control – Intelligent Robotics”. CS 395T Notes, University of Texas at Austin - UTEXAS, EUA, 2000.
- [26] Brockett, R. W., “New Directions in Applied Mathematics”. Springer-Verlag, New York, EUA, 1982.
- [27] Gholipour, A. & Yazdanpanah, M.J., “Dynamic Tracking Control of Nonholonomic Mobile Robot with Model Reference Adaptation for Uncertain Parameters”. University of Tehran – IRAN. European Control Conference – ECC, University of Cambridge, UK, 2003.
- [28] Medeiros, A. A. D., Alsina, P. J., Vieira, F. C., “Dynamic Stabilization of a Two-Wheeled Differentially Driven Nonholonomic Mobile Robot”. Simpósio Brasileiro de Automação Inteligente – SBAI. SP, Brasil, 2003.
- [29] Ramos, J. J. G., “Contribuição ao Desenvolvimento de Dirigíveis Robóticos”. Tese de Doutorado, Universidade Federal de Santa Catarina – UFSC, 2002.
- [30] Ogata, K., “Engenharia de Controle Moderno”. 4ªEd., Editora: Prentice Hall, 2003.
- [31] Bazanella, A. S., “Sistemas de Controle – Princípios e Métodos de Projeto”. 1ªEd., Editora: EDUFRGS, Brasil, 2005.
- [32] Campos, M. C. M. & Teixeira, H. G., “Controles Típicos – de Equipamentos e Processos Industriais”. 1ªEd., Editora: Blucher, Brasil, 2008.
- [33] Dorf, R. C., “Modern Control Systems”. 11ªEd., Prentice-Hall, EUA, 2008.
- [34] Pagano, D. J., “Teoria de Sistemas Amostrados e Controle Digital”. 1ªEd., Editora: EDUFSC, Brasil, 2004.
- [35] Chung, T. L., Bui, T. H., Kim, S. B., Oh, M. S., “Wall-Following Control of a Two-Wheeled Mobile Robot”. KSME International Journal, Vol.18 No.8, pp.1288~1296, Korea, 2004.
- [36] Borges, G. A., Deep, G. S., Lima, A. M. N., “Design of an Output Feedback Trajectory Controller for an Automated Guided Vehicle”. Anais do XIII Congresso Brasileiro de Automática – XIII CBA. Florianópolis-SC, Brasil, 2000.

- [37] Borges, G. A., Deep, G. S., Lima, A. M. N., “Controladores Cinemáticos de Trajetória para Robôs Móveis com Tração Diferencial”. Anais do VI Simpósio Brasileiro de Automação Inteligente – VI SBAI. Bauru-SP, Brasil, 2003.
- [38] Freire, E. O., “Controle de Robôs Móveis por Fusão de Sinais de Controle Usando Filtro de Informação Descentralizado”. Tese de Doutorado, UFES, Brasil, 2002.
- [39] Bueno, S. S., Mirisola, L. G. B., Ramos, J. J. G., “Uma Plataforma para Pesquisas e Desenvolvimento em Robótica Terrestre de Exterior”. Anais do IX Simpósio Brasileiro de Automação Inteligente – IX SBAI. Brasília-DF, Brasil, 2009.
- [40] Freitas, E. J. R., & Pereira, G. A. S., “Desenvolvimento de Automação Embarcada para um Robô Móvel Baseado em um Carro de Passeio”. Anais do IX Simpósio Brasileiro de Automação Inteligente – IX SBAI. Brasília-DF, Brasil, 2009.
- [41] Raposo, E. P., Stemmer, M. R., Pieri, E. R., Castelan, E. B., Barasuol, V., “Um Robô de Serviço Aplicado à Limpeza de Isoladores Elétricos de Sistemas de Distribuição de Energia”. Anais do IX Simpósio Brasileiro de Automação Inteligente – IX SBAI. Brasília-DF, Brasil, 2009.
- [42] Chase, O. A., “Sistema Computacional para Controle e Aquisição de Imagens de um Submarino da Classe ROV”. Trabalho de Conclusão de Curso – Instituto de Estudos Superiores da Amazônia, Brasil, 2007.
- [43] Reis, C., “Sistemas Operacionais para Sistemas Embarcados”. Livro, 1º Ed. Editora: EDUFBA, Brasil, 2004.
- [44] Litz, L., “From Ambient Intelligence to Cyber-Physical Systems”. Livro, 1º Ed, Institute of Automatic Control, Technische Universität Kaiserslautern, Alemanha, 2009.
- [45] Chase, O. A., Almeida, F. J., “Sistemas Embarcados”. Mídia Eletrônica. Página na internet: < www.sabajovem.org/chase >, capturado em 10/11/2007.
- [46] Ball, S., “Embedded Microprocessor Systems: Real World Design”, 3ºEd. Editora: MCPros, EUA, 2005.
- [47] Russel, S.J., Norvig, P., “Artificial Intelligence”. 4ºEd. Editora: Prentice-Hall, EUA, 2004.

- [48] Nascimento Jr., C. L., Yoneyama, T., “Inteligência Artificial em Controle e Automação”. 1º Ed. Editora: Edgard Blücher, Brasil, 2000.
- [49] Simões, M. G., Shaw, I. S., “Controle e Modelagem Fuzzy”. 2º Ed, Editora: Edgard Blücher, Brasil, 2007.
- [50] Viegas, V. M. R., “Projeto e Implementação de um Sistema de Sensores Inteligentes Baseado na Norma IEEE1451”, Dissertação de mestrado em Engenharia Eletrotécnica e de Computadores, Universidade Técnica de Lisboa, 241f., Portugal, 2004.
- [51] MOSCOSO, M. N. A., BEZERRA, U. H., SENA, A. S., GARCEZ, J. N., SOUZA, K. T., LEMOS, R. S., SILVA, R. R., “Sistema Distribuído de Monitoramento da Qualidade de Tensão em Redes de Distribuição de Energia Elétrica In: XV Seminário Nacional de Distribuição de Energia Elétrica”, Anais do SENDI’02, v. 1, p. 75-80, Porto Alegre-RS, Brasil, 2002.
- [52] Souza, C. P., Filho, J. T. C., “Controle por Computador – Desenvolvendo Sistema de Aquisição de Dados para PC”, Editora: EDUFMA, Edição: 1º, Brasil, 2001.

ANEXO A – O caso Zuse e Shannon. Quem primeiro “APLICOU” a álgebra de Boole.

“Os Computadores Alemães”

Do livro: “Hitler’s Scientists – Science, War and the Devil’s Pact”. Cornwell, J., Penguin Books Ltd, UK, 2003.

Dera-se início na Alemanha aos computadores binários com a obra de um engenheiro alemão, Konrad Zuse, mas suas invenções jamais foram usadas na criptoanálise. Quando ainda estudante, Zuse levava a famosa máquina calculadora de Babbage, do século XIX, ao seu próximo estágio lógico.

A partir de 1932, com vinte e dois anos e trabalhando sozinho, ele usou uma representação binária, ligado-desligado, zero - um por meio de buracos perfurados em papel, e depois pinos que podiam ser travados. Como engenheiro na indústria aeronáutica, emprego que largou, para desespero dos pais, tomou consciência do “tremendo número de monótonos cálculos necessários para o projeto de estruturas estáticas ou aerodinâmicas”. Assim, ia projetar e construir máquinas apropriadas para solucionar automaticamente tais problemas. Ninguém naquela época comentou, “sabia a diferença entre hardware [máquinas e equipamentos] e software [programas de computador]”.

Em 1933, um fabricante de máquinas de calcular dissera-lhe pelo telefone que a tecnologia do “computador” já se esgotara, e que nada de novo havia a fazer. O fabricante ainda assim foi à oficina de Zuse, onde o jovem o converteu demonstrando o princípio dos cálculos de código digitais, ou binários.

Depois disso, Zuse construiu sua Z1 e Z2, que eram modelos de teste de uma máquina de relé eletromecânica. Em 1939, foi convocado para o exercito, o que levou o fabricante a escrever uma carta ao major dele, informando-o que o jovem inventor devia ser liberado para concluir um trabalho que poderia ser útil para a indústria aeronáutica. O oficial comandante de Zuse leu a carta e disse: “Eu não compreendo isso. O avião alemão é o melhor do mundo. Eu não sei mais o que calcular”.

Seis meses depois, Zuse foi liberado do serviço militar, para trabalhar como engenheiro na indústria aeronáutica. Mas perseverou parte do tempo em seu projeto de computador, criando em 1941 o Z3, que tinha uma memória eletromecânica composta de relés, assim como uma unidade eletromecanicamente aritmética. O Z3 também tem sido descrito como a primeira máquina de calcular operacional controlada por programa. A proposta de Zuse, de um computador com base em válvulas, porém, foi rejeitado, devido ao que se via como o iminente fim da guerra. Enquanto isso, ele começou outro computador, que chamou de Z4, que sobreviveu a guerra após ser transferido de um lugar para outro, a fim de escapar dos bombardeios aliados. Zuse pode reivindicar o reconhecimento de haver feito o primeiro modelo de computador controlado por programa, mas não foi empregado como uma tecnologia importante na época da guerra.

“George Boole e Claude Shannon”

Escrito por: Ricardo K. L. Ferreira, Estudante de Ciência da Computação – Mackenzie, SP, 2006

George Boole nasceu em Lincoln - Inglaterra em 2 de Novembro de 1815, filho de um sapateiro pobre. A sua formação base na escola primária da National Society foi muito rudimentar. Autodidata, fundou aos 20 anos de idade a sua própria escola e dedicou-se ao estudo da Matemática.

Em 1840 publicou o seu primeiro trabalho original e em 1844 foi condecorado com a medalha de ouro da Royal Society pelo seu trabalho sobre cálculo de operadores. Em 1847 publica um volume sob o título *The Mathematical Analysis of Logic* em que introduz os conceitos de lógica simbólica demonstrando que a lógica podia ser representada por equações algébricas.

Este trabalho é fundamental para a construção e programação dos computadores eletrônicos iniciada cerca de 100 anos mais tarde. Na Álgebra de Boole existem apenas três

operadores E, OU e NÃO (AND, OR, NOT). Estas três funções são as únicas operações necessárias para efetuar comparações ou as quatro operações aritméticas base.

Em 1937, cerca de 80 anos após a morte de Boole, Claude Shannon, então estudante no MIT - Boston, USA - estabeleceu a relação entre a Álgebra de Boole e os circuitos eletrônicos transferindo os dois estados lógicos (SIM e NÃO) para diferentes diferenças de potencial no circuito.

Atualmente todos os computadores usam a Álgebra de Boole materializada em microchips que contêm milhares de interruptores miniaturizados combinados em portas (*gates*) lógicos que produzem os resultados das operações utilizando uma linguagem binária.

*“Conclusões sobre o primeiro cientista a usar a álgebra de Boole
em sistemas computacionais – Konrad Zuse”*

Escrito por: Otavio Andre Chase

Konrad Zuse empregou o uso da álgebra booleana para a sua principal aplicação na atualidade, que são os sistemas de computação, e isto cinco anos antes de Claude Shannon usá-las na aplicação para interruptores elétricos, tema de seu mestrado. Zuse é considerado o pai da informática, porém Shannon é quem leva o reconhecimento por ter iniciado a área de sistemas digitais e por consequência os sistemas de computação.

As nacionalidades e a tensão mundial provocada pelo início da segunda guerra mundial acabaram por ofuscar para o mundo a criatividade e prática de Zuse, que por vários momentos fora interrompido devido a trabalhar em áreas da engenharia de menos interesse para poder manter-se na então Alemanha ainda limitada pelo pacto de Varsóvia, e a obrigatoriedade de servir ao exército nazista. Konrad Zuse desenvolveu o primeiro sistema de computação digital programável do mundo em sua casa, em meio à crise alemã e ao pré-guerra.

O objetivo aqui não é desmerecer o importante trabalho de Shannon, mas sim de apresentar também o valioso trabalho de Zuse. O trabalho de Shannon só veio a ser usado em sistemas digitais de computação durante os anos de 1950, enquanto que as calculadoras de base eletrônica de Zuse já eram usadas em 1938 pelo exercito alemão para cálculos estruturais de suas aeronaves. A seguir um resumo da biografia de Zuse, através de seus projetos.

Konrad Zuse
Engenheiro alemão

22-6-1910, Berlim
18-12-1995, Hünfeld

Konrad Zuse foi o primeiro a desenvolver máquinas de cálculo controladas automaticamente. E percebeu rapidamente que um dos aspectos mais onerosos ao se fazerem longos cálculos com dispositivos mecânicos era guardar os resultados intermediários para depois utilizá-los nos lugares apropriados nos passos seguintes.

Em 1934, depois de várias idéias e tentativas, Zuse chegou à conclusão que um calculador automático somente necessitaria de três unidades básicas: uma controladora, uma memória e um dispositivo de cálculo para a aritmética. Ele desenvolveu o seu Z1, em 1936, um computador construído inteiramente com peças mecânicas e que usava uma fita de película cinematográfica para as instruções que controlavam a máquina.

Em 1938, antes mesmo de terminar o Z1, um aluno de Zuse, Helmut Schreyer, construiu uma parte do Z1 usando válvulas. Em função da situação de pré-guerra, Zuse teve de abandonar essa linha de desenvolvimento - seriam necessárias 1000 válvulas, o que era impossível naquele momento - e continuou o Z2 usando tecnologia baseada em relés. Estes dois primeiros modelos eram somente para teste: "tinham todas as características do computador posterior, mas não trabalhavam satisfatoriamente. O Z3 foi terminado em 1941 e foi o primeiro modelo totalmente operacional".

O Z3, como a maioria das máquinas dessa primeira geração, usava dois mecanismos separados para as funções aritméticas e tinha uma unidade especial para conversão de números na notação decimal para a binária.

Em termos de velocidade podia ser comparado ao MARK I, discutido mais à frente, que foi terminado dois anos após o Z3. O Z3 executava três a quatro adições por segundo e multiplicava dois números em quatro ou cinco segundos. Nunca chegou a ser usado para grandes problemas em função de possuir uma memória de tamanho limitado. Foi destruído, junto com a casa de Zuse, por um bombardeio em 1944.

O Z4 começou a ser desenvolvido quase que simultaneamente ao final do trabalho do Z3. Era essencialmente a mesma máquina, com maior capacidade de memória e mais rápida. Por causa do avanço das tropas aliadas, o trabalho do Z4 foi interrompido quase ao seu final e a máquina ficou escondida em uma pequena cidade da Bavária chamada Hinterstein. Em 1950, na Suíça, Zuse reconstruiu o seu Z4, e fundou uma empresa de computadores, absorvida depois pela Siemens.

As máquinas de Zuse tiveram pouco impacto no desenvolvimento geral da Computação pelo absoluto desconhecimento delas até um pouco depois da guerra. Em 1970, Zuse escreveu uma autobiografia, *O Computador: a Obra da Minha Vida* (1970).

Referência:

Rogers, Harold, *Pequena Biografia de Konrad Zuse*,

disponível na Internet: <http://bang.lanl.gov/video/sunedu/computer/zuse.html>