



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Plataforma robótica de baixíssimo custo para robótica educacional

Rafael Vidal Aroca

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Número de ordem PPgEEC: D85
Natal, RN, Dezembro de 2012

Catálogo da Publicação na Fonte. UFRN / SISBI / Biblioteca Setorial
Especializada do Centro de Ciências Exatas e da Terra - CCET.

Aroca, Rafael Vidal.

Plataforma robótica de baixíssimo custo para robótica educacional / Rafael
Vidal Aroca - Natal, RN, 2012.

116 f. : il.

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves.

Tese (Doutorado) - Universidade Federal do Rio Grande do Norte. Centro de
Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Compu-
tação.

1. Cibernética - Operadores humanos - Tese. 2. Robótica educacional - Tese.
3 - Arquitetura de controle - Tese. 4 - Plataforma robótica - Tese. 5. Interface de
áudio - Tese. I. Gonçalves, Luiz Marcos Garcia. II. Título.


RN/UF/BSE-CCET

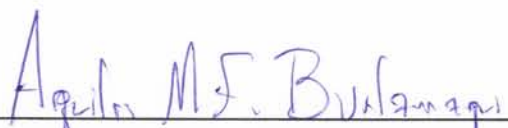
CDU 007.51


Plataforma robótica de baixíssimo custo para robótica educacional

Rafael Vidal Aroca

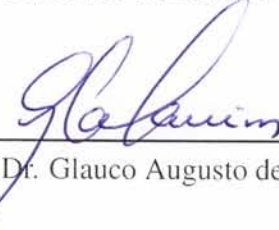
Tese de Doutorado aprovada em 17 de dezembro de 2012 pela banca examinadora composta pelos seguintes membros:


Prof. Dr. Luiz Marcos Garcia Gonçalves (orientador) DCA/UFRN


Prof. Dr. Aquiles Medeiros Filgueira Burlamaqui ECT/UFRN


Prof. Dr. Pablo Javier Alsina DCA/UFRN


Prof. Dr. Flavio Tonidandel FEI


Prof. Dr. Glauco Augusto de Paula Caurin EESC/USP

À minha mãe, Denisi.

Agradecimentos

Ao meu orientador, Prof. Luiz Marcos, pelo apoio e orientação.

Ao professor Aquiles Burlamaqui pela conversa que deu origem a este trabalho.

À minha família pelo apoio durante esta jornada, especialmente Silvia, Matheus e Lilian.

Aos colegas do laboratório NatalNet pelo apoio e amizade, especialmente Rafael B. Gomes, Renata Pitta e Renato Gardiman que ajudaram em parte deste trabalho.

Aos professores Pablo J. Alsina, Luiz Eduardo C. Leite e Diogo P. F. Pedrosa por oferecer a oportunidade de mostrarmos o N-Bot às suas turmas.

À Comunidade de software livre pelas excelentes ferramentas, destacando o Linux, Latex, SL4A e Blockly que foram usados neste trabalho.

Ao Daniel Enos e toda equipe da organização da SINEC 2012 no DCA-UFRN.

Ao CNPq pelo apoio financeiro durante este trabalho.

À Deus.

Resumo

Os robôs estão cada vez mais presentes nas mais diversas atividades da sociedade, porém ainda são considerados equipamentos caros e restritos para poucas pessoas. Neste trabalho, propomos uma arquitetura de controle e um conjunto de técnicas que possibilitam a construção de robôs de baixíssimo custo, além de diminuir a complexidade na sua montagem e programação. Um aspecto chave da arquitetura proposta é o uso de interfaces de áudio para controlar atuadores e ler sensores, permitindo assim o uso de qualquer dispositivo que produza sons como unidade de controle de um robô. Também desenvolvemos ferramentas de programação baseadas em ambientes web que permitem o uso de computadores ou telefones celulares como unidades de controle de um robô, que pode ser programado e controlado remotamente. O trabalho também contempla possíveis aplicações de tal plataforma robótica de baixíssimo custo, incluindo principalmente o seu uso educacional, que foi validado experimentalmente, por professores e alunos de várias disciplinas de graduação. Apresentamos uma análise de dados de entrevistas realizadas com os alunos, antes e após o uso da plataforma, que confirmam a sua aceitação como ferramenta de apoio ao ensino.

Palavras-chave: Robótica educacional, Robô de baixo custo, Interface de áudio, *Smartphones*

Abstract

Robots are present each time more on several areas of our society, however they are still considered expensive equipments that are restricted to few people. This work consists on the development of control techniques and architectures that make possible the construction and programming of low cost robots with low programming and building complexity. One key aspect of the proposed architecture is the use of audio interfaces to control actuators and read sensors, thus allowing the usage of any device that can produce sounds as a control unit of a robot. The work also includes the development of web based programming environments that allow the usage of computers or mobile phones as control units of the robot, which can be remotely programmed and controlled. The work also includes possible applications of such low cost robotic platform, including mainly its educational usage, which was experimentally validated by teachers and students of several graduation courses. We also present an analysis of data obtained from interviews done with the students before and after the use of our platform, which confirms its acceptance as a teaching support tool.

Keywords: Educational robotics, Low cost robot, Audio interface, Smartphones

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Símbolos e Abreviaturas	vii
1 Introdução	1
1.1 Robôs na Educação	3
1.2 Robôs de Pesquisa	5
1.3 Motivação e Objetivo	5
1.4 Contribuições	6
1.5 Organização da Tese	6
2 Embasamento Teórico	9
2.1 Robótica Educacional	9
2.1.1 Material para Robótica Educacional (o Hardware)	10
2.1.2 Softwares para Robótica Educacional	12
2.2 Robôs e <i>Smartphones</i>	13
2.2.1 O Sistema Android	13
2.2.2 Acesso à rede e Robótica nas Nuvens	14
2.2.3 Recursos de Hardware	15
2.3 Arquiteturas de Controle de Hardware	16
2.4 Comunicação via Áudio	17
2.4.1 DTMF	17
2.4.2 Análise de Sinais	18
3 Estado da Arte	21
3.1 Robótica na Educação	21
3.1.1 Robôs Educativos de Baixo Custo	22
3.1.2 Robôs como Motivadores	23
3.1.3 Um Robô por Aluno	24
3.1.4 O “ <i>10-dollar robot design challenge</i> ”	25
3.2 Ambientes de Programação de Robôs	28
3.3 Comunicação Robô-Computador	30

3.3.1	Interfaces de Dados Digitais	30
3.3.2	Interfaces Analógicas Usando Áudio	31
3.4	Resumo	33
4	Sistema Proposto	39
4.1	Hardware	39
4.1.1	Controle de Atuadores Usando Áudio	41
4.1.2	Leitura de Sensores	45
4.1.3	Controle em Malha Fechada Usando Canal de Áudio	50
4.2	Sistema baseado em Microcontrolador	51
4.3	Estrutura do Robô	56
4.4	Software	58
4.4.1	Ambiente de Programação via Web	58
5	Resultados	73
5.1	Visão Geral	73
5.1.1	Robôs Construídos e Analisados	73
5.1.2	Custos do Sistema	75
5.1.3	Repercussão	77
5.2	Aspectos Técnicos	78
5.2.1	Análise de Desempenho	78
5.3	Aspectos Educacionais	84
5.3.1	Uso em Sala de Aula	84
5.3.2	Oficina de Robótica de Baixo Custo	91
5.3.3	Possíveis Usos	92
6	Conclusão e Perspectivas	93
6.1	Contribuições	94
6.2	Trabalhos Futuros	95
	Referências bibliográficas	96
A	Questionários	111
A.1	Questionário Prévio	111
A.2	Questionário Posterior	113
A.3	Avaliação do mini-curso de robótica - SINEC 2012	115

Lista de Figuras

1.1	Visão geral do sistema	3
4.1	Visão geral da arquitetura do sistema	40
4.2	Controle de dispositivos via áudio	41
4.3	Controle de motores via áudio	42
4.4	Circuito para controle de motores DC	43
4.5	Controle de servo-motores	45
4.6	Circuito de controle de servo-motores	46
4.7	Leitura de sensores usando DTMF	47
4.8	Exemplo de conexão de sensores via áudio	48
4.9	Odometria via canal de áudio: resultados experimentais	51
4.10	Leitura de sensores via áudio	52
4.11	Leitura de sensores usando gerador de tom único	53
4.12	Circuito para leitura de sensores usando áudio	54
4.13	Sistema baseado em microcontrolador	55
4.14	Rede de microcontroladores conectados via áudio	55
4.15	Estrutura do N-Bot	57
4.16	Arquitetura de software do sistema	59
4.17	Captura de tela do ambiente de programação	61
4.18	Captura de tela do Anwide	63
4.19	Funcionamento interno do Anwide	63
4.20	Funcionamento interno do ADWN	65
4.21	Captura de tela do ambiente de programação para PC	66
4.22	Bloco usado para gerar DTMF	66
4.23	Bloco de configuração de frequência	67
4.24	Analisador de espectro via web	68
4.25	Bloco para leitura de estado de sensores	68
4.26	Programa de blocos 1	69
4.27	Programa de blocos 2	70
4.28	Programa de blocos 3	71
5.1	Robô controlado via canal de áudio	74
5.2	Dispositivos móveis controlando o robô via áudio	74
5.3	Robôs comerciais controlados pelo Anwide	75
5.4	Fotos do N-Bot	75
5.5	N-Bot construído com peças de móveis	76
5.6	Teste da taxa de leitura dos sensores	78

5.7	Desempenho da leitura de sensores via áudio	79
5.8	Desempenho para geração de tons DTMF	81
5.9	Análise de navegadores web	82
5.10	Esquema de teste dos processadores ARM	84
5.11	Ambiente de teste dos processadores ARM	85
5.12	Desempenho do servidor web	86
5.13	Desempenho em ponto flutuante	87
5.14	Interesses dos estudantes em construir um robô	88
5.15	Valores financeiros que estudantes gastariam	89
5.16	Interesses gerais dos alunos	90

Lista de Tabelas

2.1	Pares de frequências DTMF	18
3.1	Robôs de “baixo custo”	33
3.2	Interfaces entre robô e computador	35
3.3	Técnicas de conexão computador-robô	36
4.1	Dígitos DTMF e estado de motores	44
4.2	Mapeamento de estados de sensores para DTMF	49
4.3	Resolução do encoder lido via áudio	50
4.4	Protocolo de comunicação com microcontrolador	56
4.5	Parâmetros da função setFrequency	67
5.1	Custos do N-Bot	76
5.2	Redução dos custos do N-Bot	77
5.3	Comparativo entre diversos computadores de controle	77

Lista de Símbolos e Abreviaturas

A:	Amperes
ab:	<i>Apache Benchmark</i>
ADK:	<i>Android Open Accessory Development Kit</i>
ADWN:	Ambiente de Desenvolvimento Web do N-Bot
AFRON:	<i>AFrican RObotics Network</i>
AJAX:	<i>Asynchronous Javascript and XML</i>
Anwide:	<i>Android Web-based Integrated Development Environment</i>
AP:	Alta Prioridade
API:	<i>Application Programming Interface</i>
ARM:	<i>Advanced RISC Machines</i>
CI:	Circuito Integrado
CPU:	<i>Central Processing Unit</i>
DC:	<i>Direct Current</i>
DFT:	<i>Discrete Fourier Transform</i>
DIEB:	Dicionário Interativo de Educação Brasileira
DTMF:	<i>Dual Tone Multi Frequency</i>
E/S:	Entrada e Saída
FEI:	Faculdade de Engenharia Industrial
FFT:	<i>Fast Fourier Transform</i>
FM:	<i>Frequency Modulation</i>
FURG:	Fundação Universidade Federal do Rio Grande
GPS:	<i>Global Positioning System</i>

HTML: *Hypertext Markup Language*

HTTP: *Hyper Text Transfer Protocol*

Hz: Hertz

IEEE: *Institute of Electrical and Electronics Engineers*

IFES: Instituto Federal do Espírito Santo

INPI: Instituto Nacional de Propriedade Industrial

IP: *Internet Protocol*

LCD: *Liquid Crystal Display*

LDR: *Light Dependent Resistor*

LED: *Light Emitting Diode*

M2M: *Machine to Machine*

MB: *Mega Byte*

MFlops: Milhões de operações de ponto flutuante por segundo

MHz: *Mega Hertz*

MICE: *Mindstorms Internet Control Environment*

MIT: *Massachusetts Institute of Technology*

Modem: Modulador/Demodulador

PC: *Personal Computer*

PCI: *Peripheral Component Interconnect*

PDA: *Personal Digital Assistant*

PDS: Processamento Digital de Sinais

PID: *Proportional-Integral-Derivative*

PWM: *Pulse Width Modulation*

ROS: *Robot Operating System*

SA: Sensores e Atuadores

SD: *Secure Digital*

SINEC: Seminário de Informática e Engenharia de Computação

SL4A: *Scripting Layer for Android*

SLAM: *Simultaneous Localization and Mapping*

SSH: *Secure Shell*

TCP: *Transfer Control Protocol*

UCA: Um Computador por Aluno

UFES: Universidade Federal do Espírito Santo

UFMS: Universidade Federal do Mato Grosso do Sul

UFRN: Universidade Federal do Rio Grande do Norte

UFSJ: Universidade Federal de São João del-Rei

UNESP: Universidade Estadual Paulista

URA: Um Robô por Aluno

USB: *Universal Serial Bus*

V: Volts

VNC: *Virtual Network Computing*

WRE: Workshop de Robótica Educacional

WWW: *World Wide Web*

Capítulo 1

Introdução

We make no apologies for making these excursions into other fields, because the separation of fields, as we have emphasized, is merely a human convenience, and an unnatural thing. Nature is not interested in our separations, and many of the interesting phenomena bridge the gaps between fields.

Richard Feynman

A robótica encontra-se cada vez mais presente na realidade das pessoas. Enquanto robôs eram conhecidos por seu uso restrito em indústrias, hoje observa-se o seu uso frequente em escolas, em serviços gerais, e até mesmo em residências, que já possuem robôs de limpeza. Entretanto, robôs ainda são considerados por muitos como um sonho de consumo, sendo efetivamente produtos caros, principalmente devido aos custos de pesquisas envolvidas em seu desenvolvimento e materiais específicos usados em sua construção, o que ainda dificulta a sua adoção de forma mais abrangente.

Neste contexto, propomos neste trabalho uma nova abordagem para o desenvolvimento de uma plataforma robótica de baixíssimo custo que pode ser utilizada em diversas aplicações, tendo sido validada, especialmente em projetos de robótica educacional. Ao invés de utilizar microcontroladores, como é comum em robôs de baixo custo, a arquitetura da plataforma proposta possui uma interface de comunicação que utiliza tons de áudio para controlar atuadores e codificar estado de sensores. Os tons de áudio usados são sinais sonoros de frequência única ou tons do tipo *Dual Tone Multi Frequency* (DTMF), que oferecem uma forma de troca de dados via áudio robusta e amplamente usada em sistemas de telefonia. Dessa forma, qualquer dispositivo com interface de áudio pode ser usado como unidade de controle deste tipo de robô, incluindo telefones celulares, *smartphones*, *tablets* e computadores tradicionais, dentre outros dispositivos.

O robô desenvolvido, batizado de N-Bot, custa cerca de quatorze dólares quando as peças são compradas em pequena quantidade. O preço pode ser reduzido para menos de dez dólares por unidade dependendo da quantidade de componentes adquirida.

Os robôs e sistemas de automação em geral possuem tipicamente uma ou mais unidades de controle compostas por processadores embarcados, como microcontroladores ou computadores de propósito geral. Essas unidades de controle possuem interfaces para conexão com sensores e atuadores, além de interfaces de programação e monitoramento,

que tipicamente são acessadas através de um computador. Nesse cenário, é frequentemente necessária a instalação de softwares específicos e *drivers* de dispositivos em um computador para programação e controle, limitando essas operações a usuários especialistas.

Em geral, para programar um robô, é necessário usar softwares e ferramentas específicas visando desenvolver os programas, compilá-los e enviá-los ao robô para execução autônoma. Muitas dessas ferramentas são proprietárias [Gumbley & MacDonald 2005] e somente funcionam com uma determinada linguagem de programação e sistema operacional. Em alguns casos, usuários precisam ter acesso a um computador com privilégios de administração de sistema para instalar, e em outros casos, executar softwares de desenvolvimento de robótica. Outra abordagem comum consiste em utilizar um computador embarcado no robô, que inclui sistema operacional e ferramentas de desenvolvimento, de forma que o usuário possa conectar-se remotamente ao robô, desenvolver, compilar e executar programas diretamente no computador de bordo do robô.

Uma alternativa para resolver o problema descrito consiste em oferecer uma ferramenta de desenvolvimento que pode ser acessada via web, permitindo que os usuários possam editar, compilar e executar programas remotamente usando apenas um navegador web. Com essa solução, qualquer dispositivo que possua um navegador web, pode ser usado como estação de desenvolvimento, monitoramento e controle de um robô. Uma das contribuições do trabalho descrito neste texto consiste no uso dessa abordagem embarcada em um telefone móvel instalado no robô, através de uma arquitetura de software e hardware que pode ser aplicada a diferentes tipos de robôs.

Para usar um telefone celular como unidade de controle de um robô, um canal de comunicação deve ser estabelecido entre os circuitos eletrônicos do robô e do telefone. Alguns trabalhos [CellBots 2011, Ahn et al. 2009, Sa et al. 2009] usam portas seriais para este fim, e outros *bluetooth* [CellBots 2011, Santos et al. 2010], entretanto, nesses trabalhos, é necessário um telefone com essas interfaces de comunicação, bem como um microcontrolador para prover a troca de dados com o dispositivo de controle. Como alternativa, apresentamos o uso de um canal de comunicação universal e presente na grande maioria dos dispositivos móveis: o canal de áudio, que pode ser acessado através das conexões para fones de ouvido.

Dessa forma, uma das principais contribuições deste trabalho consiste no desenvolvimento de uma técnica de controle de atuadores e leitura de sensores usando canais de áudio e componentes eletrônicos de baixo custo que podem ser facilmente encontrados no mercado. Como o canal de áudio é universal, não apenas telefones celulares podem ser usados como unidade de controle, mas qualquer outro dispositivo com interface de áudio, como *tablets*, reprodutores de música MP3 e computadores tradicionais.

De fato, o uso de telefones como unidades de controle para robôs é uma tendência promissora para a robótica, como apontado recentemente pela Associação Internacional dos Engenheiros Elétricos e Eletrônicos (IEEE) [Guizzo & Deyle 2012]. Os telefones atualmente disponíveis no mercado oferecem diversos recursos diretamente úteis para robótica, destacando bibliotecas de programação com suporte a reconhecimento e síntese de voz, e vários sensores embutidos no aparelho, como acelerômetros, receptor do sistema de posicionamento global (GPS), câmera, wifi, dentre outros. Além disso, também

desenvolvemos ferramentas amigáveis de programação textual ou em blocos para tornar a programação do N-Bot intuitiva e disponível via Internet para qualquer computador.

A Figura 1.1 mostra uma visão geral da arquitetura da plataforma robótica proposta, incluindo comunicação com sensores e atuadores. Embora a figura mostre um computador, a interface de áudio permite que qualquer dispositivo capaz de produzir sons possa ser usado como unidade de controle do robô. Como mostrado na figura, uma porta do barramento serial universal (USB) pode ser usada para fornecer energia para o robô, dispensando o uso de pilhas. A saída de áudio do computador, normalmente usada para fones de ouvido, é ligada ao robô, que possui um circuito responsável por decodificar sinais sonoros de áudio gerados pela unidade de controle e realizar o controle de motores e atuadores diversos. Da mesma forma, a entrada de microfone do dispositivo de controle recebe sinais de áudio gerados pelo circuito de controle que codificam o estado de vários sensores. Teoricamente, esse sistema permite que os estados de centenas de sensores possam ser adquiridos simultaneamente.

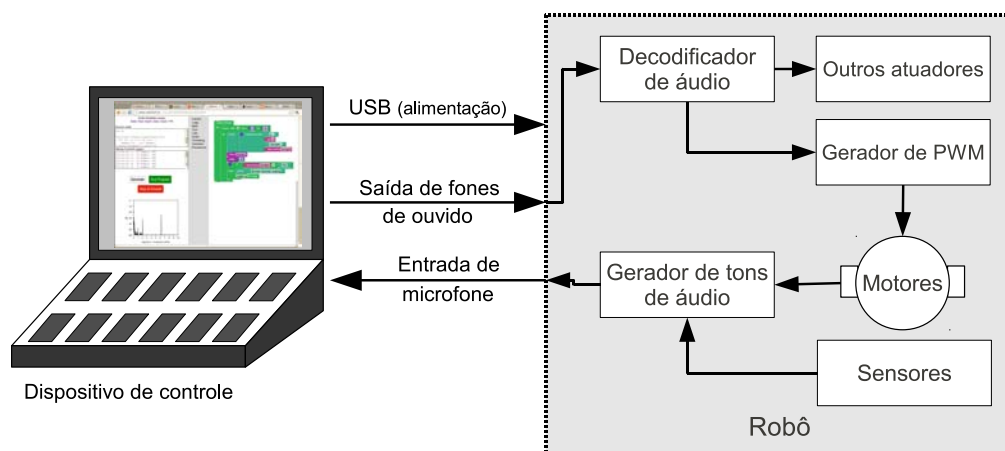


Figura 1.1: Visão geral do sistema e suas conexões entre dispositivo de controle e robô.

1.1 Robôs na Educação

O uso da robótica vem se tornando cada vez mais presente em todos os níveis da educação, incluindo o ensino fundamental, médio, graduação e pós-graduação. De fato, diversos autores demonstram que o uso de robôs como ferramentas no processo educacional proporcionam ambientes estimulantes e motivadores [Soto et al. 2006, Hamblen & Hall 2004, Alves et al. 2011b, Howard & Graham 2007, Aroca, Gomes, Tavares, Souza, Burlamaqui, Caurin & Gonçalves 2012], além de uma experiência única de aprendizado [Weinberg & Yu 2003]. Rawat & Massiha (2004) demonstram dados confirmando que o retorno dos alunos em aulas que envolvem robótica é *surpreendentemente positivo*. É importante ressaltar a robótica também como ferramenta, ou seja, que esses resultados não se aplicam somente ao ensino de robótica em si, mas também em seu uso para diver-

sas outras disciplinas, incluindo até mesmo geografia, ciências ambientais, dentre outras. Nesses casos, a grande vantagem é que o robô atua como um instrumento motivador para os alunos.

Contudo, ainda existem dois principais problemas para a adoção mais abrangente de tecnologias de robótica na educação: os custos e a dificuldade de uso. Com relação aos custos, vários autores [Alves et al. 2011b, Galvan et al. 2006, Alves et al. 2011a, Chella 2012] alertam que os custos altos de robôs podem impedir seu uso em salas de aula. Além do mais, Lumsden e Ortega-Sanchez afirmam que existe uma escassez de plataformas robóticas de baixo custo [Lumsden & Ortega-Sanchez 2010]. Esse problema é ainda mais grave no Brasil, onde muitas escolas e alunos sofrem dificuldades financeiras, não podendo adquirir facilmente robôs ou kits de robótica. De fato, alguns kits comercialmente disponíveis no Brasil chegam a custar milhares de reais.

A questão de facilidade de uso também é peça fundamental para o sucesso de projetos de robótica educacional. O uso do sistema deve ser fácil não apenas para os alunos, mas também para os professores, já que muitos educadores não possuem experiência com softwares de robótica. Nesse sentido, Dipietro et al. (2008) explicam que quando o software de um sistema computacional é intuitivo, a adoção em massa desse produto é uma consequência, citando o exemplo do *tablet* pioneiro, o *iPad*.

Mesmo com essas restrições, observamos que quando é dada a oportunidade de experimentar com robôs, estudantes frequentemente expressam interesse em possuir seus próprios robôs para explorar mais. Embora eles demonstrem interesse em investir seu tempo e dinheiro para construir robôs, poucos alunos podem gastar com esse tipo de projeto. Para melhorar esse cenário, o N-Bot pode ser usado como plataforma robótica educacional de baixo custo, e pode viabilizar que mais alunos possam construir e possuir seus próprios robôs.

Como já mencionado, o N-Bot pode ser controlado por um telefone celular. Uma das vantagens do uso de celulares, é que o Brasil possui mais de um telefone celular para cada habitante [Teleco 2012a] (de fato pouco mais que 1,3). Assim, um aluno que não possua um telefone celular poderá tentar obter um modelo usado ou antigo. Além disso, as operadoras de telefonia celular poderiam destinar aparelhos usados, e trocados por novos, para uso em aplicações de robótica educativa em escolas públicas, ajudando a reaproveitar equipamentos obsoletos.

Durante este trabalho, uma pesquisa foi conduzida com mais de duzentos alunos matriculados em diversos cursos de graduação para verificar o interesse desses alunos em robótica e em disciplinas relacionadas, destacando o resultado de que mais de 70% desses alunos disseram que gastariam seu dinheiro para construir seu próprio robô caso o custo fosse acessível e desde que ele pudesse ser usado em várias disciplinas do curso de graduação [Aroca, Gomes, Tavares, Souza, Burlamaqui, Caurin & Gonçalves 2012].

O N-Bot pode ser utilizado nos níveis fundamental, médio e superior, entretanto, até o momento somente analisamos seu uso no nível superior. Em um trabalho futuro, seu uso será estudado para outros níveis de educação, levando em conta o desenvolvimento de ambientes de programação com nível de abstração adequado para a idade dos alunos.

1.2 Robôs de Pesquisa

O N-Bot também pode facilitar experimentação e testes em algumas subáreas de pesquisa da robótica. Nessa área de pesquisa, existe ainda o mito de que a realização de investigações em robótica requerem equipamentos caros, já que a maioria dos robôs usados em pesquisa são de fato caros. Por exemplo, o robô Pioneer AT custa mais de 10 mil dólares (outros muito mais caros estão comercialmente disponíveis). Nesse contexto, o N-Bot oferece uma alternativa simples, mas de baixíssimo custo que pode ser usada para realizar pesquisas consideradas de alto nível em robótica, envolvendo decisão e execução de tarefas de mais alto nível de processamento. Notamos que a unidade de controle do N-Bot pode ser um PC ou telefone celular com alto poder de processamento, que também pode ser facilmente substituído para o aumento do desempenho ou mesmo trocado entre robôs.

Um aspecto interessante e em crescimento, é a “robótica nas nuvens” (*Cloud Robotics*) [Guizzo 2011], que consiste em manter algoritmos essenciais e simples a bordo do robô e delegar parte do processamento do mesmo para serviços disponíveis via rede (Internet). Assim, por exemplo, a unidade de controle pode se ocupar das técnicas de navegação e desvio de obstáculos enquanto que a tarefa de reconhecimento de objetos 3D pode ser delegada para serviços executados *na nuvem*. Neste cenário, mesmo um robô bastante simples e barato pode tirar proveito de técnicas avançadas, complexas e com demanda de um alto poder de processamento, disponíveis via rede como serviços remotos.

1.3 Motivação e Objetivo

A motivação básica deste trabalho é tornar o acesso à robótica mais abrangente. Para tanto, o objetivo principal deste trabalho é projetar e validar uma plataforma robótica aberta e gratuita, com recursos otimizados, permitindo construir um robô de baixíssimo custo, que possa ser utilizado amplamente, por alunos e profissionais de qualquer setor da sociedade (alunos, professores, pesquisadores, leigos).

Também como resultado de nossos desenvolvimentos, esperamos que mais pesquisadores possam fazer suas investigações em áreas relacionadas com a robótica, com a possibilidade de testar seus algoritmos de forma experimental, no mundo real, não ficando restritos às simulações. Além disso, a plataforma robótica proposta também pode ser adotada em ambientes educacionais como ferramenta auxiliar para aumentar a motivação dos alunos de vários níveis de ensino.

Convém ressaltar que nossa contribuição principal, o N-Bot (melhor descrito a seguir), foi atestada internacionalmente, tendo sido um dos vencedores do prêmio “*10-dollar robot design challenge*”. Com motivação similar à nossa, os professores Ken Goldberg da Universidade de Berkeley (EUA) e Ayorkor Korsah, da Universidade de Ashesi (Gana), criaram um desafio para o desenvolvimento de robôs educativos de dez dólares [Hurst 2012a, Korsah & Goldberg 2012a, Hurst 2012b] com a expectativa de que esses robôs de baixo custo possam ser usados na África para melhorar a motivação e interesse dos alunos do ensino fundamental e médio [Hurst 2012a]. O N-Bot participou desta competição, ficando em segundo lugar entre 28 participantes [Korsah &

Goldberg 2012a, Hurst 2012b].

1.4 Contribuições

As contribuições deste trabalho, incluindo suas publicações, são:

- A principal contribuição deste trabalho é um conceito de robô de baixo custo usando a integração de várias tecnologias. Como prova de conceito, foi construído um robô batizado de N-Bot, desenvolvido usando as técnicas mencionadas abaixo, com a possibilidade de controle via telefone, *tablet*, computador ou outros dispositivos [Aroca, Burlamaqui & Gonçalves 2012, Aroca, de Oliveira & Gonçalves 2012, Aroca, Gomes, Tavares, Souza, Burlamaqui, Caurin & Gonçalves 2012];
- Uma nova técnica para controle de atuadores usando canais de áudio e tons DTMF [Aroca, Burlamaqui & Gonçalves 2012, Aroca et al. 2011, Júnior et al. 2012];
- Uma nova técnica para leitura de estado de sensores usando canais de áudio [Aroca, Burlamaqui & Gonçalves 2012, Aroca et al. 2011];
- Um sistema de controle em malha fechada usando as técnicas mencionadas acima [Aroca, Burlamaqui & Gonçalves 2012, Aroca et al. 2011];
- Dois ambientes de programação de robôs baseados na web, disponibilizados como softwares livres [Aroca, Guardiman & Gonçalves 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c];
- Metodologia para aplicação e análise do uso do robô de baixo custo proposto em ambientes educacionais [Aroca, Gomes, Tavares, Souza, Burlamaqui, Caurin & Gonçalves 2012];
- Disponibilização à comunidade de um roteiro com instruções passo a passo para construção do N-Bot [Aroca, Pitta, Burlamaqui & Gonçalves 2012a].

1.5 Organização da Tese

Este texto está organizado em 6 Capítulos: uma melhor definição do problema (com embasamento), todos os desenvolvimentos realizados, com experimentação e resultados, e uma conclusão ao final.

O embasamento teórico necessário à compreensão do texto é apresentado no Capítulo 2, onde são discutidos aspectos já estabelecidos de robótica educacional, seus resultados e vantagens, além de equipamentos e softwares para o uso de robótica na educação. Também são apresentadas as vantagens de se utilizar *smartphones* como controladores de robôs, assim como formas de conexão entre computadores de controle e os circuitos de acionamento de atuadores e de leitura de sensores em robôs. Ainda neste Capítulo, são apresentados alguns detalhes sobre técnicas clássicas de transmissão de dados via canais de áudio, como o DTMF, e a transformada de Fourier, que é usada para análise de sinais.

Em seguida, o Capítulo 3 apresenta uma revisão do estado da arte, incluindo trabalhos que analisam as vantagens do uso de robótica na educação e o impacto no interesse e motivação dos alunos. Também são discutidos diversos trabalhos que propõem plataformas

robóticas educacionais de baixo custo, bem como os resultados do desafio “*AFRON 10-dollar robot design challenge*”. Neste Capítulo também são revisados trabalhos que apresentam plataformas de programação remota de robôs via web, interfaces de comunicação digitais, analógicas, e aquelas que usam canais de áudio entre robôs e computadores.

No Capítulo 4, a plataforma robótica proposta é descrita com detalhes, incluindo aspectos de implementação e construção. Neste Capítulo são apresentadas novas técnicas para leitura de acionamento de motores de corrente contínua via tons da áudio DTMF, bem como de servo-motores. Além das técnicas de acionamento de motores, também são apresentadas três técnicas para leituras de dados de sensores via canais de áudio, incluindo o uso de DTMF e de tons únicos associados a cada sensor. São apresentadas ainda duas plataformas de programação via web desenvolvidas para facilitar a programação de robôs de baixo custo e robôs baseados em dispositivos móveis.

O Capítulo 5 apresenta resultados experimentais do robô proposto dos pontos de vista técnico e educacional. Além do N-Bot ter um baixo custo, de cerca de US\$ 14,00, também demonstramos que seu ambiente de software, o Anwide, pode ser utilizado para controlar diversos tipos de robôs. Uma análise de desempenho também apresentada, mostra que o sistema proposto possui performance de leitura de sensores compatível com kits Lego. Adicionalmente, são apresentados resultados de questionários sobre a opinião e interesse de mais de 200 alunos em assuntos relacionados a aulas de graduação e robótica de baixo custo.

Por fim, o Capítulo 6 apresenta as conclusões do trabalho, bem como uma lista de possíveis trabalhos futuros, visando dar continuidade ao projeto aqui iniciado. Acreditamos ter obtido bons resultados, principalmente em número de publicações, que podem ser explorados ainda mais em diferentes níveis educacionais e de pesquisa. Além disso, pretende-se estabelecer parcerias para o uso do N-Bot como plataforma que possibilite maior acesso à robótica para diversos perfis de alunos.

Capítulo 2

Embasamento Teórico

Knowledge is only part of understanding. Genuine understanding comes from hands-on experience.

Dr. Seymour Papert

O principal foco deste trabalho é o desenvolvimento de uma plataforma robótica, incluindo sua arquitetura e seus métodos intrínsecos, para que se possa difundir a robótica educacional como uma ferramenta poderosa de aprendizado. Assim, para entender melhor o contexto desta tese, introduzimos conceitos e técnicas já consolidadas, visando uma melhor compreensão do aprofundamento específico do tópico de estudo, que será realizado nos capítulos seguintes deste texto. Entender melhor esta área envolve saber o que é e como se faz robótica educacional, quais as metodologias atuais, recursos (software e hardware), quais os materiais usados, incluindo os produtos atuais voltados para robótica educacional à venda no mercado, bem como soluções alternativas. Ainda, no tocante às principais contribuições, é necessário também um melhor embasamento dos aspectos técnicos e científicos de mais baixo nível, relacionados à robótica, bem como técnicas já estabelecidas e as ferramentas matemáticas que foram utilizadas ao longo do trabalho.

2.1 Robótica Educacional

O Dicionário Interativo de Educação Brasileira (DIEB) [Menezes & Santos 2002] descreve a robótica educacional e a robótica pedagógica como sinônimos. Segundo o dicionário, são termos utilizados para caracterizar ambientes de aprendizagem que reúnem materiais de sucata ou kits de montagem compostos por peças diversas, motores e sensores controláveis por computador e softwares que permitem programar de alguma forma o funcionamento dos modelos montados. Em ambientes de robótica educacional, os sujeitos constroem sistemas compostos por modelos e programas que os controlam para que eles funcionem de uma determinada forma. Ainda segundo o dicionário, o trabalho com a robótica educacional tem vários objetivos e metodologias, apontando, em geral, para que o aluno siga instruções e manuais ou crie e experimente a partir dos materiais específicos desse ambiente. Alguns objetivos para o trabalho com a robótica educacional podem se relacionar com as artes, a cibernética, o design, a física, a matemática, a motricidade

e a vida artificial. Além disso, a elaboração de sistemas robotizados incentiva reflexão sobre as implicações que os projetos podem gerar em âmbito social, cultural, político e ambiental.

De acordo com a definição formal, acima, um aspecto importante da robótica educacional é que ela não é uma abordagem com foco exclusivamente no ensino de robótica em si, já que ela introduz o robô como elemento motivador para possibilitar o aumento de interesse e reflexão em diversos outros assuntos. Isso foi confirmado por vários autores que relatam que o uso de robôs em sala de aula aumenta o interesse dos alunos, proporciona ambientes estimulantes e motivadores, além de oferecer uma experiência única de aprendizado [Soto et al. 2006, Hamblen & Hall 2004, Alves et al. 2011b, Howard & Graham 2007, Aroca, Gomes, Tavares, Souza, Burlamaqui, Caurin & Gonçalves 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c, Weinberg & Yu 2003, Rawat & Massiha 2004], e de possibilitar a integração de várias áreas do conhecimento.

Essas vantagens já foram percebidas e incorporadas em diversos países, como na Alemanha e Holanda, que já possuem robôs em 100% das escolas públicas [Gonçalves 2012, Bieniek et al. 2012]. A Inglaterra, Itália, Espanha, Estados Unidos e Canadá também estão implementando projetos similares [Gonçalves 2012], enquanto que, na América do Sul, o Peru já possui mais de três mil escolas com robôs [Gonçalves 2012]. No Brasil, esse movimento ainda não é comum e ocorre mais em escolas particulares. Sabe-se que algumas universidades, como a FEI, a UNESP, a UFMS, a FURG, a UFRN, a UFES e a UFSJ têm projetos de extensão e iniciativas relacionadas com robótica educacional. No estado da Paraíba, especificamente, a *Estação Ciência*, um ambiente criado para popularizar e desmistificar experiências científicas e tecnológicas usando linguagem acessível, disponibiliza kits de robótica que podem ser usados por visitantes e turmas de escolas, numa iniciativa pioneira. Na própria UFRN, o desenvolvimento de metodologias e técnicas para robótica educacional tem sido alvo de experimentação científica nos últimos dez anos, pelo Laboratório NatalNet (www.natalnet.br), culminando com a incubação tecnológica de uma empresa no setor (*RoboEduc* - www.roboeduc.com).

2.1.1 Material para Robótica Educacional (o Hardware)

Existe uma grande variedade de produtos para robótica educacional à venda no mercado. Segundo nossos estudos da literatura, podemos dividir esses equipamentos, de acordo com a abordagem a ser utilizada na construção do robô em si, em três categorias: o uso de kit comercial, o uso de peças e materiais eletrônicos para montagem e o uso de material alternativo (sucata).

Uso de Kit Comercial

Essa categoria refere-se a kits prontos de fábrica, sendo eles já produtos comerciais de prateleira (disponíveis em lojas ou rede especializada) que podem ser adquiridos para uso direto em ambientes educacionais. A vantagem desses produtos é que em geral eles já oferecem manuais e guias didáticos de montagem para os alunos e professores. Por outro lado, esses produtos são caros, com custos que vão desde centenas de reais até várias dezenas de milhares de reais.

Sem dúvida, o produto mais conhecido nesta área é o Lego NXT *Mindstorms* [The 2012], um kit robótico amplamente usado em escolas. Este kit contempla, além dos blocos convencionais de montagem, sensores e atuadores que são controlados por uma unidade de controle (um computador embarcado) que pode ser programada usando um software intuitivo de programação por blocos. É interessante notar que esse produto foi concebido cuidadosamente em parceria com o Prof. Seymour Papert do MIT, um pesquisador pioneiro no uso de tecnologias aplicadas à educação, tais como o computador e a robótica. O nome *Mindstorms* tem origem no livro seminal do Prof. Papert: *Mindstorms: Children, Computers and Powerful Ideas*.

Outros kits também usados são o VEX [Vex 2012] e o Fischertechnik [fis 2012], que também incluem partes mecânicas, eletrônicas e respectivos softwares de programação. Empresas brasileiras também projetaram seus kits e comercializam diversos produtos de robótica educacional. Destacam-se os kits da Modelix [Mod 2012], da PNCA [PNC 2012] e da Robokit [Rob 2012a].

Uso de Material de Montagem

Alguns projetos e escolas podem optar por não usar um kit disponível no mercado, e sim construir um robô comprando e adaptando as peças mecânicas e eletrônicas, também disponíveis no mercado. Normalmente, nesta abordagem, esse tipo de atividade não conta com manuais e guias de montagem, e requer um maior amadurecimento dos alunos, para projetar, especificar, adquirir e integrar as peças de forma adequada.

A unidade de controle desses robôs costuma ser um microcontrolador, como o PIC da Microchip, ou o AtMega da Atmel, que é o microcontrolador usado nas placas Arduino. O Arduino, por sua vez, tem seu uso cada vez mais predominante em projetos de robótica desse tipo, já que sua programação é fácil, prática, e seu custo não é muito alto. Outra plataforma adequada para montagem de robôs é o dispositivo SunSpot [Oracle 2012], que é uma unidade de controle que pode ser programada em Java, facilitando seu desenvolvimento para iniciantes em programação. Além disso, também é possível a implementação de algoritmos clássicos de robótica no SunSpot, como por exemplo o sistema de localização usando odometria convencional desenvolvido por do Nascimento et al. (2012). Nossa proposta de trabalho se encaixa melhor nesse modelo.

Uso de Sucata

A robótica com material alternativo, também conhecida como robótica livre, consiste em reaproveitar materiais, peças e equipamentos que seriam descartados (sucatas), para a construção dos robôs. Além de reduzir os custos [da Silva & da Silva Almeida 2012], esta abordagem também é interessante, já que muitos equipamentos que seriam normalmente jogados no lixo possuem peças que podem ser usadas em um robô. Impressoras, por exemplo, possuem diversos motores, engrenagens, cabos e circuitos lógicos que podem ser reaproveitados. Assim, antes desses equipamentos serem descartados efetivamente, pode-se retirar as peças que poderão ser usadas para compor as diversas partes do robô. Convém ressaltar que o termo *sucata* tem sido questionado pela comunidade de robótica educacional, referenciando que isso pode lembrar o lixo em si, mas que esse termo

efetivamente denota o modelo em questão. Usaremos esse termo ou *robótica livre* ou ainda *robótica com materiais alternativos* indistintamente neste trabalho para indicar este modelo de construção.

Nesta linha, Thomaz et al. [de Lima Sá 2011, Thomaz et al. 2012] listam diversas abordagens e projetos de robótica livre e apresentam uma proposta para construção de robôs educativos por cerca de R\$ 100,00 usando peças recicladas de computadores e alguns componentes adicionais comprados no mercado. Botelho et Al. propõem um kit de robótica didática acessível baseado no uso de sucata e componentes de baixo custo [Botelho et al. 2012], apontando que uma das vantagens dessa abordagem é dar maior liberdade para os alunos usarem a criatividade para construir seus robôs, evitando a necessidade de seguir um guia de instruções rígido do tipo passo a passo.

2.1.2 Softwares para Robótica Educacional

Existem diversas opções de softwares para robótica educacional, que normalmente são usados para desenvolver e programar os robôs criados pelos alunos, ou em alguns casos, apenas para simular robôs no computador, dispensando o uso de um robô real. Esses softwares são normalmente específicos para determinados robôs.

Para a robótica educativa, é comum que as ferramentas de programação mencionadas ofereçam a opção de programação por blocos: onde o aluno usa o software para conectar, de forma visual, caixinhas que representam o acionamento de algum atuador, a leitura de algum sensor, ou estruturas lógicas e de repetição. Outros softwares somente suportam a programação textual tradicional, e alguns oferecem um ambiente híbrido, onde os blocos são traduzidos para código fonte, que pode ser editado.

O Lego *Mindstorms* oferece uma ferramenta de programação baseada no LabView, que permite programar graficamente o computador do robô para diversos tipos de comportamento autônomo. O programa desenvolvido é gravado na unidade de controle, que opera de forma independente. Devido a dificuldades com a língua inglesa e buscando melhor nível de abstração para trabalhar com alunos do primeiro segmento do ensino fundamental (primeiro ao quinto ano), pesquisadores do Laboratório NatalNet desenvolveram o ambiente RoboEduc [Barros 2008], que também oferece um ambiente para programação visual de robôs Lego, porém em Português, e com abstrações mais simples e didáticas para atrair crianças. O sistema criado por Barros (2008) já foi usado com sucesso em centenas de crianças com faixa etária entre 4 e 10 anos de idade.

No Brasil, a empresa XBot também apresentou recentemente [Netto et al. 2012] um ambiente de programação gráfica para seus robôs didáticos, incluindo recursos avançados de programação gráfica e a funcionalidade de captura de imagens de uma câmera. Nota-se, nesse sistema, que o programa não é transferido e executado no robô, mas sim executado no computador de controle, que envia comandos de movimentação remotos via rádio.

Outra ferramenta bastante poderosa, mas restrita para alunos do ensino superior, é o Microsoft *Robotics Studio*, uma ferramenta que permite programação textual ou gráfica, e inclui blocos com funcionalidades e algoritmos avançados para controle de robôs, incluindo captura de dados 3D de uma cena usando o sensor Kinect, também da Micro-

soft. Outra vantagem dessa ferramenta, é o fato dela ser genérica e gratuita, podendo ser utilizada com vários robôs e plataformas robóticas, incluindo o Lego *Mindstorms*.

2.2 Robôs e *Smartphones*

Uma das possíveis unidades de controle do N-Bot são os *Smartphones*, que são telefones celulares dotados de muitos recursos de software e hardware. De fato, os *smartphones* possuem alto poder computacional, além de diversos outros recursos que podem ser úteis para controle de robôs. Essa tendência foi confirmada em um artigo recente publicado na *IEEE Robotics and Automation Magazine*, apontando robôs baseados em *smartphones* como uma tendência para a robótica [Guizzo & Deyle 2012].

De fato, é comum esses telefones possuírem receptor do sistema de posicionamento global (GPS), câmera, bússola, acelerômetro, wifi, *bluetooth*, alto-falantes, microfone, sensor de luz, além de várias bibliotecas de software úteis para robótica, como reconhecimento e síntese de voz [Aroca, Burlamaqui & Gonçalves 2012, Aroca, de Oliveira & Gonçalves 2012]. Além disso, como esses aparelhos são produzidos com grande economia de escala, é mais barato comprar um telefone com todos esses recursos, do que um microcontrolador tipicamente usado em robótica, e cada um dos sensores mencionados para então integrá-los.

Dessa forma, vários projetos têm sido apresentados recentemente usando telefones celulares como unidades de controle de robôs [CellBots 2011, Ahn et al. 2009, Sa et al. 2009, Santos et al. 2010, Aroca, Burlamaqui & Gonçalves 2012]. A viabilidade também é confirmada por Santos et al. em um estudo que aponta que telefones equipados com processadores de 300MHz já são capazes de executar algoritmos complexos de robótica [Santos et al. 2010]. Como mencionado, os dispositivos móveis possuem várias características que podem ser úteis em robótica. Algumas destas possibilidades são discutidas a seguir.

Com relação ao software, os telefones atuais também possuem sistemas operacionais e ambientes com diversas facilidades para implementação de sistemas robóticos. Um desses sistemas, o Android, é apresentado a seguir.

2.2.1 O Sistema Android

O Android é um sistema operacional baseado em Linux desenvolvido pelo Google para ser usado em *smartphones* e *tablets*. Para oferecer um modelo de desenvolvimento de aplicações produtivo, os programas podem ser desenvolvidos usando a linguagem Java e facilmente testados e enviados para dispositivos com Android [Google 2012], ou para um simulador. Quando dispositivos reais são usados, os programadores podem depurar erros no próprio dispositivo via cabo USB, ou via rede. Assim, é possível depurar erros e corrigir aplicações no robô à distância facilmente. Se necessário, o programador também pode optar pelo uso da linguagem C, Python ou outras linguagens suportadas pelo Linux.

Usando a interface para programação de aplicações (API) do Android é possível aproveitar uma vasta gama de opções para desenvolvedores de pesquisas em robôs. A síntese e reconhecimento de voz, por exemplo, um recurso muito desejado em robôs, funciona

perfeitamente, enquanto em robôs tradicionais os programadores devem necessariamente instalar, configurar e ajustar bibliotecas de software para reconhecimento de voz, que geralmente não funcionam bem. Com a API do Android, é possível fazer com o que robô reconheça frases completas em várias línguas, além de falar frases completas com apenas algumas linhas de programação. Outro aspecto interessante, é que ao analisar a fala do usuário, a API do Android retorna uma lista das frases mais prováveis faladas pelo usuário, facilitando a interpretação dos comandos de voz.

A API do Android também oferece acesso facilitado à câmera e sensores de orientação. Alguns dispositivos possuem apenas acelerômetro, outros possuem acelerômetro e bússola, e outros ainda possuem um giroscópio adicional. As mesmas chamadas de funções podem ser usadas para obter dados de todos esses sensores, independentemente do dispositivo. Para tanto, é usada uma abstração que retorna a acurácia dos sensores juntamente com as medidas.

Em 2011 o Google introduziu o *Android Open Accessory Development Kit* (ADK), um sistema que permite dispositivos com Android, incluindo telefones, a se conectarem diretamente a dispositivos externos, tais como teclados, mouses, câmeras e *joysticks*. Uma limitação que poderia surgir com esse sistema, é o fato dos telefones móveis não possuírem portas USB do tipo mestre, assim nenhum acessório USB poderia ser conectado ao telefone. Para resolver esse problema, o ADK permite que acessórios USB atuem como o mestre do barramento USB, permitindo que equipamentos externos possam interagir com programas rodando no dispositivo com Android. O Google já oferece suporte oficial para microcontroladores PIC e para o Arduino.

Possível uso de Visão Computacional

Um sistema de visão artificial é essencial para qualquer robô que precisa perceber e interagir com o mundo real. De fato, Kyrki e Kragic explicam que o sucesso de uma tarefa de um robô é relacionado com o sucesso do seu sistema de visão [Kyrki & Kragic 2011].

Normalmente, várias técnicas de visão computacional são usadas em robótica, onde destacam-se a detecção e reconhecimento de objetos e a atenção robótica [Garcia et al. 2000], que é a habilidade de selecionar um tópico de interesse, a visão estéreo, a odometria visual [Scaramuzza & Fraundorfer 2011], e sistemas de localização e mapeamento simultâneos (SLAM). Desenvolvedores normalmente usam a biblioteca de visão computacional OpenCV, que foi portada para dispositivos móveis, podendo ser usada tanto no *iPhone* quanto em sistemas com Android, permitindo o desenvolvimento de sistemas de visão computacional embarcados com a mesma flexibilidade oferecida pelo OpenCV em computadores tradicionais.

2.2.2 Acesso à rede e Robótica nas Nuvens

Como os *smartphones* possuem diversos meios de acesso à rede Internet, destacando acesso via serviços GPRS/3G/4G da operadora de telefonia ou via conexão wifi, essa conectividade pode ser usada para monitorar ou enviar comandos remotamente para o robô, ou para dar acesso a serviços disponíveis na Internet através do conceito de robótica

nas nuvens. Neste conceito, é possível que robôs usem a *World Wide Web (WWW)* como fonte para busca de conhecimento e de soluções para resolver certos problemas.

Uma das primeiras demonstrações desse conceito foi feita por pesquisadores do Google. Eles apresentaram um robô controlado por um telefone com sistema Android que pode receber comandos de voz como *Encontre um bolinho*, e pesquisar nos serviços *online* do Google usando as imagens adquiridas por sua câmera. Assim, o reconhecimento de imagens é executado por um serviço externo, na nuvem, sem a necessidade da execução de nenhum algoritmo de visão no telefone [Hickman et al. 2011, Tofel 2011]. Eles também mostram que esse modelo de programação permite que os robôs acessem facilmente serviços na nuvem, como instruções de navegação até um determinado destino, tradução e reconhecimento de textos vistos em placas de sinalização ou objetos usando reconhecimento ótico de caracteres [de Molengraft et al. 2011] com a possibilidade de traduzir esses textos.

Outros trabalhos recentes dão aos robôs a capacidade de buscar e descarregar da Internet modelos 3D de móveis, aprender sobre eles e reconhece-los enquanto eles navegam em ambientes desconhecidos [Mozos et al. 2011]. Além disso, não é necessário ensinar as informações desejadas para cada robô, já que todos podem se beneficiar da mesma fonte de informações e algoritmos disponíveis na nuvem.

Finalmente, o PR2, robô mencionado na subseção anterior, possui dois computadores embarcados, cada um com um processador de 8 núcleos e 24GB de memória RAM. Assim, embora os robôs venham recebendo computadores cada vez mais poderosos, ainda é fato a dificuldade de um robô poder adquirir e armazenar grandes quantidades de conhecimento e utilizá-los para reconhecer objetos e tomar decisões em tempo real. Muitos algoritmos classificadores, por exemplo, levam vários segundos para reconhecer objetos, e alguns necessitam de uma fase de treinamento que pode levar vários dias, tornando a nuvem uma solução bastante interessante para tomada de decisões em robôs com velocidade e eficácia.

2.2.3 Recursos de Hardware

Além dessas possibilidades descritas, ressaltamos ainda vários outros recursos, e mesmo assim esta lista não é exaustiva:

- CPU: *Smartphones* possuem processadores com velocidade de 1GHz ou mais. Alguns modelos possuem processadores multi-core, e são capazes de executar tarefas complexas e computacionalmente intensivas para navegação autônoma de robôs;
- Bússola: pode ser usada para conhecer a direção de movimento do robô, melhorando algoritmos de controle;
- GPS: Pode ser usado para obter a posição do robô em ambientes externos, além de sua altitude e velocidade;
- Acelerômetro: pode ser usado para detectar mudanças de velocidade, e consequentemente se o robô bateu em algum objeto em qualquer direção. Também pode ser usado para detectar a orientação do robô e para melhorar a precisão de algoritmos de navegação visual através de fusão sensorial;

- *Bluetooth*: pode ser usado para trocar informações com robôs próximos, para localização, e para comunicação com outros dispositivos *bluetooth*. Um robô baseado em *Lego Mindstorms NXT*, por exemplo, pode ser controlado por um *smartphone* via *bluetooth*;

As características de telefones celulares apresentadas nesta seção mostram a viabilidade do uso desses dispositivos como unidade de controle de um robô de forma prática e flexível. Assim, isso justifica o uso de *smartphones* como unidades de controle de robôs ao longo desse trabalho, fato que também já foi enfatizado por Goebel et al. (2011).

2.3 Arquiteturas de Controle de Hardware

Para situar e comparar a técnica de controle de hardware usada no N-Bot, apresentamos nesta Seção uma breve descrição das formas mais comuns de conectar computadores de controle de robôs com os seus sensores e atuadores (SA).

1. SA ↔ Microcontrolador: essa é uma das abordagens mais comuns em projetos de pesquisa e na indústria. Um microcontrolador, como o PIC ou AtMega (usado no Arduino), é diretamente conectado a sensores e atuadores e os algoritmos são executados todos no microcontrolador. Esses dispositivos possuem baixo custo, mas também possuem capacidade de processamento limitada, normalmente menos de 50MHz, além de possuírem memórias de apenas alguns Kilobytes ou menos, dificultando a implementação de tarefas complexas que usam câmera, ou fazem algum tipo de mapeamento robótico [de Santana Souza & Gonçalves 2012];
2. SA ↔ Microcontrolador ↔ Computador: essa abordagem consiste em usar um microcontrolador para ler sensores e controlar atuadores. Os laços de controle principais executam em outro computador, normalmente um PC, que é conectado ao microcontrolador usando porta serial RS-232, USB, Bluetooth ou outra forma de conexão;
3. SA ↔ Placas de E/S ↔ Computador: esta técnica é tipicamente usada em robôs industriais, comerciais e em alguns robôs de pesquisa. Todos os laços de controle são executados em um computador, normalmente um PC, equipado com placas especiais de entrada e saída (E/S) para coletar dados dos sensores e controlar atuadores. Exemplos são as placas da *National Instruments* e a *ServoToGo*. Essas placas são instaladas em barramentos do computador, como o *Peripheral Component Interconnect (PCI)*, ou *Compact PCI*, e sensores podem ser ligados diretamente nessas placas. Os atuadores também são ligados nessas placas, mas normalmente amplificadores de potência também são usados, dependendo do tipo de atuador;
4. SA ↔ Microcontrolador ↔ Rede ↔ Computador: trata-se de uma arquitetura de controle distribuída onde sensores e atuadores são conectados a microcontroladores de baixo custo. Esses microcontroladores são interconectados com o computador de controle usando uma rede de comunicação. Essa abordagem já é usada em diversos tipos de robôs: o PR2, por exemplo, um robô recente e altamente flexível usa uma rede *Ethercat*, uma variação da rede *Ethernet*, para trocar dados de sensores e atuadores com o computador principal.

Todos esses métodos de conexão de sensores e atuadores baseiam-se em componentes eletrônicos digitais. No Capítulo 4, uma técnica de coleta de dados de sensores e acionamento de atuadores usando o canal de áudio é descrita. A vantagem dessa técnica é a possibilidade do uso de diversos dispositivos móveis como unidade de controle de forma intercambiável e a simplicidade dos circuitos.

2.4 Comunicação via Áudio

A troca de dados via canais de áudio é uma técnica bastante antiga e útil em inúmeras aplicações. Na década de 80, era comum utilizar fitas cassete para gravar programas de computador como uma sequência de sinais sonoros. Para executar um programa, era preciso tocar (apertar o *Play*) a fita cassete no gravador, que gerava uma sequência de sons, que eram usados pelos microcomputadores para carregar programas em suas memórias. Nessa época, antes da existência da Internet comercial, algumas universidades transmitiam, via rádio FM, programas de computador em determinadas horas do dia. Bastava sintonizar um receptor de rádio FM na rádio universitária, e conectar a saída do rádio à entrada do microcomputador, que recebia o novo software, via transmissão sem fio.

Também baseados em canais telefônicos de voz, os modems foram o principal meio de conexão entre computadores remotos durante muitos anos. Hoje em dia, variações dessas técnicas são muito utilizadas em sistemas de comunicação entre máquinas (*machine to machine, ou M2M*), onde destacam-se a comunicação entre alarmes residenciais e centrais de monitoramento, e outros equipamentos de telemetria. Como os canais de telefonia de voz possuem banda restrita, baixa confiabilidade e ruído, foram criadas várias técnicas para codificar e transmitir dados de forma segura através desses canais. Uma dessas técnicas é o tom duplo de frequência múltipla, o DTMF, que é usado como base desse trabalho.

2.4.1 DTMF

O DTMF, ou *Dual Tone Multi Frequency*, é uma técnica que codifica dezesseis caracteres diferentes através de um par de frequências sonoras. Ele foi criado na década de 50, sendo uma opção mais rápida em relação ao sistema de discagem por pulsos que hoje é obsoleto. Seu principal propósito é o de enviar os dígitos que um usuário pressionou em seu telefone para uma central de telefonia. Embora o sistema já tenha mais de 60 anos, ele ainda é amplamente usado em sistemas de telecomunicações e em praticamente todos os projetos de novos telefones [Davidson et al. 1999].

Os tons DTMF podem ser identificados ao pressionar teclas de um telefone durante uma ligação. O sistema é baseado em 16 frequências de áudio organizadas em uma matriz de dimensão quatro. A Tabela 2.1 mostra essas frequências e seus dígitos correspondentes. Assim, um dígito válido é sempre composto por um par de frequências: uma da coluna e uma da linha da tabela, que devem ser geradas simultaneamente. Por exemplo, para codificar o dígito “9”, um sinal de áudio deve conter as frequências 852 Hz e 1,477 Hz.

	1,209 Hz	1,336 Hz	1,477 Hz	1,633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Tabela 2.1: Pares de frequências DTMF e seus dígitos correspondentes. Tabela adaptada do *Audio Engineer's Reference Book*.

Como o DTMF foi projetado para trocar dados em linhas telefônicas ruidosas, o uso de um par de frequências torna o sistema eficiente e robusto a ruídos e ainda a outros sons que não caracterizam um dígito DTMF. De fato, quando o sistema DTMF foi concebido, suas frequências foram cuidadosamente escolhidas para minimizar a semelhança entre pares de tons DTMF e sons naturais [van Bosse & Devetak 2006].

Além de aplicações de comunicação para alarmes residenciais, o DTMF também é usado em rastreadores de carro e em sistemas interativos de resposta por voz, como aqueles sistemas bancários automáticos, que solicitam para o usuário discar senhas, e escolher opções através do teclado de seu telefone.

2.4.2 Análise de Sinais

Como foi mencionado no Capítulo 1, o N-Bot utiliza sinais de áudio para codificar o estado de diferentes sensores. Dessa forma, um circuito gerador de tons de áudio é utilizado no robô, e o dispositivo de controle precisa analisar o sinal de entrada para identificar frequências sonoras que contenham alguma informação sobre o estado dos sensores.

Para identificar as frequências presentes no sinal de entrada, utiliza-se a transformada discreta de Fourier (DFT), descrita pela Equação 2.1. Assim o dispositivo de controle deve amostrar continuamente o sinal de áudio, que está no domínio do tempo, e através da DFT, representá-lo no domínio da frequência.

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nm/N} \quad (2.1)$$

Na Equação 2.1, $X(m)$ é a magnitude do sinal em análise na posição m , $x(n)$ é a sequência de entrada no domínio do tempo, representando o sinal de entrada com índice n . O parâmetro N especifica o número de pontos da DFT, que também determina a resolução e o número de amostras a serem analisadas. Como a DFT requer alta demanda de processamento, implicando também em aumento no tempo, é comum implementar a transformada rápida de Fourier (FFT), assim como fizemos neste trabalho. Para mais detalhes e explicações sobre esses e outros conceitos de análise de sinais, recomendamos recorrer à literatura sobre processamento digital de sinais [Lyons 2010, Chitode 2008, Khan 2005].

Um aspecto importante a ser observado na implementação da DFT, é a taxa de amostragem usada. De acordo com o teorema fundamental da amostragem de Nyquist-Shannon [Lyons 2010], esse parâmetro deve ser pelo menos duas vezes a maior frequência que será

analisada. Assim, caso se deseje analisar sinais com tons DTMF, deve-se notar, pela Tabela 2.1, que a maior frequência de um dígito DTMF é 1,633Hz. Assim, para sinais deste tipo, a taxa de amostragem, F_s , deve ser de pelo menos 3,266Hz.

Outro aspecto que deverá ser considerado na implementação é o equilíbrio entre taxa de amostragem, número de pontos da FFT e resolução de frequência desejada. Para determinar esta relação, a Equação 2.2 é utilizada.

$$f(m) = \frac{mF_s}{N} \quad (2.2)$$

Para exemplificar, do ponto de vista teórico, a capacidade do uso de canal de áudio para transmitir o estado de diversos sensores do tipo ligado/desligado, podemos tomar como base um canal de áudio típico para transmissão de voz que vai de 20Hz a 20KHz. Pela equação 2.2, usando uma taxa de amostragem (F_s) de 40KHz e uma FFT com 2048 pontos, é possível analisar o sinal com uma resolução de aproximadamente 20Hz. Assim, tal sistema pode analisar cerca de 1000 tons de frequências diferentes.

Por segurança, pode-se estabelecer um espaçamento de 20Hz antes e depois do sinal de interesse relacionado a cada sensor, caracterizando a capacidade de identificar cerca de 300 frequências diferentes, sendo que a presença ou ausência de cada um desses tons codificará o estado de um determinado sensor. Na prática, este número é ainda menor devido a interferências e outros motivos práticos. Dependendo da forma com que o sinal dos sensores é codificado, outro motivo potencial para redução da capacidade do sistema pode surgir devido a presença de harmônicas no sinal de entrada, que são tons de duas vezes a frequência do sinal principal.

Maiores detalhes sobre a aplicação da DFT para análise do estado dos sensores representados por tons de áudio, a escolha da resolução adequada para nosso sistema de acordo com a Equação 2.2, bem como o uso do DTMF para controle de atuadores são apresentados no Capítulo 4.

Capítulo 3

Estado da Arte

If I have seen further it is by standing on the sholders of Giants.

Isaac Newton

Visto que esta tese possui uma característica multidisciplinar, fazemos neste capítulo a revisão do estado da arte de uma forma lógica, de acordo com cada aspecto relacionado com o seu tema, concentrando e contextualizando as referências de trabalhos relacionados diretamente ao tema desta tese.

Como o foco maior do nosso trabalho está no desenvolvimento da plataforma robótica em si, trabalhos sobre teorias de ensino-aprendizagem não são estudados neste Capítulo, sendo alguns apenas citados como motivação.

3.1 Robótica na Educação

As pessoas riram de Seymour Papert, nos anos 60, quando ele sugeriu a ideia das crianças usarem computadores como instrumentos de aprendizagem. Mas Papert não se deixou abater, tendo conduzido de forma brilhante suas pesquisas no Instituto de Tecnologia de Massachusetts (MIT), sendo considerado, por isso, o pai da Robótica Educacional. Segundo Blikstein, o segredo das metodologias educacionais usando tecnologias está no desenho das tecnologias em si, com interfaces simples que só expõem a matemática e a física que são adequadas ao desenvolvimento cognitivo da criança [Dalmazo 2012].

Hoje em dia, é fato amplamente conhecido que a robótica é uma solução ideal para integração curricular, oferecendo aos estudantes a possibilidade de relacionar conceitos aprendidos em várias disciplinas [Manseur 1997, Mirats Tur & Pfeiffer 2006, Alves et al. 2011a]. O incentivo ao trabalho em equipe também é naturalmente promovido [Mirats Tur & Pfeiffer 2006, Martins et al. 2012]. Além disso, a maioria das pessoas aprendem mais facilmente quando a execução de tarefas e atividades práticas estão envolvidas no processo de aprendizagem [Conrad 2005], de forma que os robôs podem ser usados como ferramentas pedagógicas que oferecem uma experiência do tipo “aprender fazendo” [Alves et al. 2011b, Ahlgren 2002].

Outra possibilidade oferecida pela robótica no ambiente educacional é a possibilidade dos alunos participarem em competições de robótica, que estimulam e despertam

o interesse desses alunos [Angonese et al. 2012, Reis et al. 2012]. De fato, Miranda & Suanno (2012) explicam que a participação em competições não motiva apenas os alunos, mas também os professores. Além disso, cursos que envolvem competições de robótica e o uso de robôs na sala de aula aumentam significativamente o número de matrículas [Alemany & Cervera 2012].

Um fato interessante é que a robótica na educação, e seus bons resultados, vêm atraindo cada vez mais o interesse das comunidades de pesquisa. Em 2012 a reconhecida revista *Transactions On Education* da IEEE lançou uma chamada de trabalhos para uma edição especial sobre robótica na educação. No Brasil, o Prof. Dr. Luiz Marcos Garcia Gonçalves, orientador deste trabalho, vem organizando repetidamente com apoio da Sociedade Brasileira de Computação (SBC), o Workshop de Robótica Educacional, que já está na sua terceira edição. O Workshop recebeu 29 submissões em 2012, incluindo algumas de fora do Brasil. No âmbito internacional, ocorreu nesse mesmo ano a terceira edição da conferência internacional de robótica na educação (RiE), que já tem data marcada para a quarta edição em 2013.

A seguir, descrevemos diversos robôs educacionais criados para propósitos educacionais.

3.1.1 Robôs Educativos de Baixo Custo

Embora os robôs sejam ferramentas educacionais excelentes, vários especialistas [Alves et al. 2011b, Galvan et al. 2006, Alves et al. 2011a, Chella 2012] alertam que os altos custos dos robôs podem impedir seu uso em salas de aula. Da mesma forma, Lumsden e Ortega-Sanchez argumentam que o mercado ainda carece de plataformas robóticas flexíveis com custo acessível [Lumsden & Ortega-Sanchez 2010]. Essa realidade motivou educadores e pesquisadores a desenvolverem vários robôs educacionais de baixo custo para o ensino [Greenwald & Kopena 2003, Galvan et al. 2006, Dinh & Inanc 2009, Hamblen & Hall 2004, Lumsden & Ortega-Sanchez 2010]. Além disso, Botelho et al. (2012), enfatizam que o baixo poder aquisitivo não deve ser um impedimento para escolas implementarem projetos de educação tecnológica, sugerindo alternativas acessíveis, como o uso de sucata para construir robôs.

Existem vários trabalhos na literatura descrevendo projetos de robôs de baixo custo para educação, sendo que alguns deles são diretamente relacionados com este trabalho. Nesse contexto, vários autores propõem robôs educacionais de baixo custo, mas na maioria dos casos, sem mencionar o custo desses dispositivos. Outros autores propõem robôs educacionais, que consideram de baixo custo, na faixa entre 200 e 1000 dólares [Greenwald & Kopena 2003, Dinh & Inanc 2009, Hamblen & Hall 2004, Lumsden & Ortega-Sanchez 2010]. Embora esses valores possam ser baixos para universidades, eles são inacessíveis para a maioria dos estudantes que desejam possuir seu próprio robô, ou seu kit de robótica, especialmente para aqueles de países em desenvolvimento, como a África ou mesmo o Brasil.

Nota-se assim que o termo *baixo custo* é relativo. No intervalo de preço mencionado também se encontra o Lego NXT Mindstorms, uma plataforma robótica amplamente usada em ambientes escolares [Mirats Tur & Pfeiffer 2006]. Uma tecnologia nacional

para ambientes educacionais é o robô Curumin, comercializado pela empresa Xbot. Embora já seja adotado em diversas escolas, ele tem um custo relativamente alto [da Silva & da Silva Almeida 2012]. Esse robô oferece uma ferramenta de programação para plataforma Windows, que permite aos alunos definirem as tarefas do robô conectando blocos graficamente [Netto et al. 2012], assim como é feito no ambiente do Lego NXT.

Piperidis et al. (2007) apresentam uma pesquisa mostrando que apenas 5% dos robôs móveis comerciais custam menos de 100 Euros. Além disso, eles também afirmam que um robô com recursos suficientes para ser usado tanto em educação quanto em pesquisa custa pelo menos 3000 Euros, concluindo que existe a necessidade de robôs móveis com melhor custo-benefício. Como solução, Piperidis et al. (2007) apresentam um robô com custo de 500 Euros. Para obter maiores reduções de custo, Hamblen e Hall sugerem o uso de peças de brinquedos para construção de robôs educativos [Hamblen & Hall 2004].

Com relação as unidades de controle, uma abordagem muito usada consiste no uso de computadores portáteis (*notebooks*, por exemplo) como unidades de controle de robôs, contudo esses computadores possuem portas de E/S limitadas, necessitando placas auxiliares para conectar sensores e atuadores [Hamblen & Hall 2004], além do fato do robô precisar ser robusto para carregar um *notebook*, ou mesmo um *netbook*. O robô LUDEBot [Pin et al. 2012] é um projeto que se encaixa nessa categoria, já que possui um Arduino para interface com sensores e atuadores e um *netbook* para execução de algoritmos de alto nível. Além disso, o LUDEBot custa menos de um quarto do valor de um Kit Lego Mindstorms, e os autores sugerem a possibilidade de substituir o *netbook* por um telefone celular ou *tablet* com Android.

Uma alternativa com menor custo é o uso de simuladores de robótica [Chella 2012], embora essa atividade possa não ser tão atrativa para os alunos. Fernandes et al. (2012) propõem uma abordagem alternativa onde os alunos usam um simulador para refletir e resolver um determinado problema. Uma vez solucionado o problema no simulador, os alunos montam um robô real para testar a solução imaginada na prática usando um kit de robótica livre.

Outra solução que vem ganhando cada vez mais atenção é o uso de *smartphones* como as principais unidades de controle de robôs [Guizzo & Deyle 2012, Santos et al. 2010, Aroca, Burlamaqui & Gonçalves 2012, Aroca, de Oliveira & Gonçalves 2012]. Como os telefones móveis possuem vários recursos e sensores diretamente úteis para robôs, seu uso como *cérebro* de um robô torna-se interessante. Além disso, como já descrito no Capítulo 2, os celulares são fabricados em larga escala, então é mais barato comprar um celular que possui acelerômetro, bússola, câmera e outros sensores do que um microcontrolador e todos esses sensores separadamente para depois integrá-los. Finalmente, *smartphones* oferecem ambientes de desenvolvimento e bibliotecas de programação que são mais fáceis de usar do que aquelas fornecidas por fabricantes de microcontroladores.

3.1.2 Robôs como Motivadores

É uma crença comum que o aprendizado torna-se melhor quando os estudantes estão motivados e engajados [Mok 2011, Galvan et al. 2006, Pink 2009]. Dessa forma, de acordo com Alves et al. (2011b), a robótica pode ser usada como um elemento mo-

tivador, e de fato Rawat & Massiha (2004) relatam que os comentários dos alunos após aulas que envolvem robótica são *extremamente positivos*. Alves et al. (2011b) também argumentam que os alunos consideram a construção de protótipos de robôs uma atividade motivadora. Além disso, os alunos demonstram um grande interesse na aplicação de conceitos aprendidos durante aulas que envolvem robôs [Galvan et al. 2006]. Adicionalmente, a motivação não se aplica apenas à alunos em cursos convencionais: Ranieri et al. (2012) apresentam o uso de robótica para melhorar a motivação e realizar atividades educacionais com crianças com deficiências, onde os autores observaram que o sistema mostrou-se atrativo para as crianças.

Avaliações de trabalhos práticos com robôs em vários cursos mostram que os estudantes consideram estas práticas de laboratório desafiadoras e estimulantes [Kramer & Anderson 2011]. Firebaugh e Piepmeier coletaram as opiniões de estudantes durante um curso sobre nano-robôs, e observaram que a maioria dos alunos avaliou o curso como *um dos melhores que eles tinham feito*. Esses estudantes também disseram que trabalhar em um laboratório e construir *alguma coisa* são pontos fortes daquele curso específico de robótica [Firebaugh & Piepmeier 2008].

Um estudo mostra que estudantes são mais motivados pela vontade de aprender do que por ganhar algum prêmio ou completar os créditos de um curso [Hernando et al. 2011]. Esse fato é consistente com as teorias descritas no livro de Pink [Pink 2009], que compila vários estudos sobre motivação. De acordo com Pink, a motivação autônoma, que tem origem no interesse individual, promove melhor compreensão conceitual, melhores notas, e melhor persistência na escola [Pink 2009]. Ele também argumenta que esse tipo de motivação promove o compromisso.

Um aspecto importante da motivação é que as tarefas não devem ser nem muito difíceis nem muito fáceis, pois caso contrário a motivação se perde [Mok 2011], ou ainda pior, as tarefas podem causar frustração [Pink 2009]. De acordo com Mok, um descompasso entre a complexidade das tarefas e as habilidades dos alunos prejudica a obtenção de engajamento dos estudantes. Acreditamos que o N-Bot seja uma ferramenta ideal para evitar esse tipo de problema. Com atividades planejadas adequadamente, ele pode ser aplicado em uma larga variedade de tópicos e dificuldades, possibilitando seu uso por estudantes novatos e seniores.

3.1.3 Um Robô por Aluno

A ideia de um *robô por aluno (URA)*, que pode ser uma extensão do projeto já existente *um computador por aluno (UCA)*, foi sugerida várias vezes pelo Prof. Dr. Aquiles Burlamaqui em comunicações pessoais realizadas no laboratório NatalNet-UFRN. Uma proposta semelhante, feita pelo Prof. James McLurkin, de uma universidade Americana, ganhou destaque da imprensa internacional no final do ano de 2011, bem como recebeu diversos prêmios pela proposta de um robô por aluno de engenharia.

McLurkin defende um curriculum acadêmico onde cada aluno deve ter seu próprio robô, permitindo assim testes e estudos de lições individuais, bem como a realização de trabalhos em equipe [Ric 2011]. Para tanto, McLurkin criou uma plataforma robótica de baixo custo (segundo o autor) chamada *R-One* que custa cerca de US\$ 200,00. O autor

também abriu uma empresa e pretende comercializar esse robô a partir de 2012, já que ele acredita que esse custo possibilite a adoção em massa desses robôs em escolas. No momento, cada estudante novato da universidade de Rice, nos EUA, recebe seu próprio robô *R-One* no primeiro dia de aula [Ric 2011].

Também com a intenção de possibilitar que cada aluno tenha seu próprio robô para fazer lições em casa, Lauwers desenvolveu o *Finhch*, um robô com custo de US\$ 99,00 já à venda no mercado Americano [Ino 2012]. Esse robô inclui sensores de temperatura, de luminosidade, de toque, acelerômetro, LEDs, câmera, microfone e alto falantes. Segundo Lauwers, o foco desse robô não é para ensino de robótica, mas para facilitar e tornar mais atrativo o aprendizado de programação de computadores (na versão atual usando Java). A intenção desta iniciativa é tornar aulas de programação mais interessantes, dispensando a necessidade dos alunos construírem seus próprios robôs [Ino 2012]. Deve-se notar que esse robô não é autônomo: ele não possui baterias e opera somente quando conectado a um PC via cabo USB. Todos algoritmos são executados no PC, que envia comandos para leitura de sensores e movimento das rodas [Lauwers & Nourbakhsh 2010].

3.1.4 O “10-dollar robot design challenge”

Durante uma viagem para Gana, o professor Ken Goldberg, da universidade de Berkeley (EUA), notou que alunos de uma universidade usavam kits Lego *Mindstorms*, e em seguida os empacotavam para enviar para outra escola, onde o kit seria usado na próxima semana. Isto o fez refletir se não seria possível reduzir os custos desse kit [Hurst 2012a].

Interessados em melhorar as condições de ensino e motivação dos alunos africanos, Goldberg e Korsah fundaram a rede africana de robótica (AFRON), uma comunidade de pesquisadores e instituições que trabalham para levar a robótica para África, especialmente robótica educativa.

De acordo com Goldberg e Korsah [Korsah & Goldberg 2012b], os fundadores da AFRON, robôs excitam pessoas de todas as idades, mas são muito caros para alunos africanos e de economias emergentes. Dessa forma, como primeiro projeto da AFRON, eles promoveram o *AFRON 10-dollar robot design challenge*, com a intenção de fomentar o desenvolvimento de plataformas robóticas abertas de baixo custo, especialmente para educação.

É interessante o fato de que o próprio Goldberg, ao criar o desafio, alegou que a meta de dez dólares é irreal, mas que eles queriam estabelecer um valor que fizesse as pessoas pensarem em soluções criativas [Hurst 2012a]. Dessa forma, Goldberg se espantou ao notar que diversos pesquisadores conseguiram projetar e construir robôs com custos próximos, ou até inferiores a dez dólares.

Para participar do desafio, não era permitido apenas propor um projeto, mas um robô real deveria ser construído, e todas suas instruções de montagem e funcionamento deveriam ser publicadas na Internet. A competição foi dividida em três categorias e teve um total de 28 participantes, sendo que os ganhadores foram divulgados em vários canais da imprensa especializada [Korsah & Goldberg 2012a, Hurst 2012b, Aroca, Pitta, Burlamaqui & Gonçalves 2012b]. A seguir, cada uma das categorias são relatadas, com uma breve descrição de cada ganhador. Detalhes adicionais de cada um dos robôs podem ser

encontrados no site da AFRON [Korsah & Goldberg 2012a].

Categoria *Tethered*

A categoria *Tethered* contempla robôs que funcionam conectados a um computador via um cabo (USB, por exemplo), de forma que a alimentação pode ser fornecida pelo computador, bem como os algoritmos e programas também são executados no computador.

1. Suckerbot (Tailândia)

O Suckerbot é um robô móvel construído a partir de uma adaptação de controles de vídeo-games e custa US\$ 8,96. A construção consiste em fazer cortes e instalar motores e sensores no próprio *joystick*, e controlá-los através de um software instalado no PC. Um aspecto que marcou seu projeto foi o uso de pirulitos para detectar colisões do robô em qualquer direção.

2. Baobot (EUA)

O Baobot é um robô móvel sem carcaça, de forma que todos componentes do robô, incluindo motores, são fixados em sua placa de circuito impresso. Ele possui um processador AtMega, sendo assim compatível com o ambiente de desenvolvimento e programas para a plataforma Arduino. O custo do Baobot é US\$ 31,06.

3. Afrobot (EUA)

O Afrobot é um robô móvel caracterizado pela construção usando caixas de papelão, onde são fixados todos os motores, sensores e demais componentes. Com custo de US\$ 9,94 (para 1000 peças), ele é controlado pela placa *Launchpad* da Texas Instruments, que é baseada no processador MSP430.

4. RoboArm (Nigéria)

O RoboArm é um braço robótico composto apenas por componentes eletrônicos analógicos e nenhuma unidade de processamento, nem conexão de dados como USB ou serial. Para controlar esse robô, seus idealizadores utilizaram a tela de um computador como canal de controle. Assim, diversos sensores de luz do tipo resistor dependente de luz (LDR) são colados com fita adesiva sobre a tela do computador, e um programa de computador desenha círculos na tela em posições coincidentes com o local onde os sensores estão colados. Variando as cores e brilho de cada círculo desenhado na tela, é possível controlar cada motor do RoboArm. Seu custo é de US\$ 50,00.

Categoria *Traditional*

A categoria tradicional é composta por robôs que podem operar de forma autônoma usando baterias e sem conexão direta com um computador durante a operação. Eles podem ser conectados ao computador apenas para programação.

1. Kilobot (EUA)

O Kilobot é o resultado de trabalhos já desenvolvidos anteriormente pelo grupo de pesquisas de sistemas auto-organizáveis da universidade de Harvard (EUA). Trata-se de um robô móvel de formato circular construído sem o uso de rodas, mas apenas

três suportes fixos em configuração triangular. O robô possui 2 pequenos motores de vibração, iguais àqueles utilizados em telefones celulares, e ajustando a vibração desses dois motores, o software do Kilobot permite fazer todos tipos de movimento sobre uma superfície. Ele é alimentado por uma bateria do tipo “moeda” e controlado por um microcontrolador AtMega, portanto compatível com o ambiente e programas da plataforma Arduino. Outro aspecto interessante do Kilobot é que ele possui um transmissor e um receptor de infravermelho, que permite sua comunicação bidirecional sem fio com um computador de controle e com todos os Kilobots que estiverem nas proximidades. Quando vários Kilobots estão juntos, eles podem se auto organizar e executar várias tarefas em grupo de forma colaborativa. O custo do Kilobot é US\$ 43,00 para uma unidade, caindo para US\$ 14,00 caso sejam compradas peças para construção de mil unidades.

2. SwarmRobot (China)

O SwarmBot é um robô móvel minimalista, mas flexível. Ele possui um microcontrolador AtTiny, compatível com Arduino, uma ponte H para controle de motores, e dois motores fixados diretamente na placa de circuito, que também é a carcaça do robô. As rodas são tampas de garrafas plásticas do tipo PET, e ao invés de possuir uma terceira roda livre, os projetistas optaram por usar um LED que serve de apoio e desliza sobre o chão quando o robô se movimenta. Sua placa é modular, permitindo adicionar sensores. De acordo com os autores, ele custa US\$ 9,51 para uma unidade, e US\$ 4,89 caso sejam compradas peças para mil unidades.

3. SEG (EUA)

O SEG é um robô móvel desenvolvido pelo instituto de tecnologia de Massachusetts, MIT (EUA). Ele é composto por dois servo-motores e um microcontrolador AVR fixados à mesma placa de circuito impresso. Uma característica especial é que as rodas são dobráveis e vêm desmontadas em um kit. A montagem das rodas é feita de forma análoga a um origami e dobraduras de papel. O robô possui sensores infravermelho para desviar de obstáculos, sendo sustentado por duas rodas apenas, ficando a placa de circuito em “rotação livre” enquanto o robô se move. Seu custo é de US\$ 14,09.

4. DiscBot (EUA)

O DiscBot é um robô móvel com seu chassi construído de partes de madeira, incluindo as rodas, cortadas à laser por apenas dois dólares, de acordo com o proponente. Ele utiliza um processador AtMega, compatível com Arduino, e dois motores de passo, além de possuir um sensor de ultra-som para desviar de obstáculos. O protótipo custa US\$ 47,00, e com escala de mil unidades, cada DiscBot custaria US\$ 9.76.

Categoria *All-in-one*

A categoria *All-in-one*, ou todos em um, agrupa robôs polivalentes que podem operar tanto na categoria *Tethered* quanto *Traditional*.

1. MITBOTS (Índia)

O MITBOTS é uma cópia bastante similar ao Lego *Mindstorms*. Ele possui um conjunto de blocos de montagem que se encaixam entre si, sensor de infravermelho, de toque e um bloco “inteligente” composto por um processador ARM com tela de cristal líquido (LCD) e conexão USB para programação via PC. Ele também inclui um software de programação usando blocos. De acordo com o fabricante, o MITBOTS custa US\$ 33,00.

2. N-Bot (Brasil)

O MITBOTS é uma cópia bastante similar ao Lego *Mindstorms*. Ele possui um conjunto de blocos de montagem que se encaixam entre si, sensor de infravermelho, de toque e um bloco “inteligente” composto por um processador ARM com tela de cristal líquido (LCD) e conexão USB para programação via PC. Ele também inclui um software de programação usando blocos. De acordo com o fabricante, o MITBOTS custa US\$ 33,00. O N-Bot, proposto neste trabalho, é o robô de baixíssimo custo que foi desenvolvido especialmente para o AFRON usando as técnicas descritas ao longo deste trabalho para controle de atuadores e leitura de sensores usando canais de áudio. O N-Bot pode ser controlado por um telefone ou por um PC com Linux, Windows ou Mac, ligando diretamente cabos de áudio entre o robô e o PC. Outro aspecto interessante do N-Bot, é que ele pode ser programado de forma textual ou usando blocos, sem a necessidade de instalar nenhum software no PC, já que o ambiente de programação é baseado na web. Seu protótipo custa US\$ 14,00, e para escala de mil unidades, cada robô custaria US\$ 9,00. Melhores descrições das técnicas e metodologias do N-Bot podem ser obtidas ao longo de todo este trabalho.

3.2 Ambientes de Programação de Robôs

Uma característica da vasta maioria dos ambientes para programação de robôs, e de todos os trabalhos mencionados até o momento, é que todos eles requerem a instalação de softwares específicos e configuração de computadores. Embora esta tarefa não seja altamente complexa, ela pode ser difícil de ser realizada por professores de escolas de nível médio e fundamental sem experiência em informática. Além do mais, para instalação dos softwares, é necessário privilégios de administrador no computador, dificultando a instalação dos ambientes de programação em computadores de uso público como escolas, bibliotecas e tele-centros.

Uma solução proposta por diversos autores é disponibilizar ambientes de programação e desenvolvimento de robótica via web [Belousov et al. 2001, Garrett & Thornton 2005, Aroca, Guardiman & Gonçalves 2012, Popescu et al. 2008, Aroca, Pitta, Burlamaqui & Gonçalves 2012c], de forma que qualquer computador ou dispositivo com um navegador web possa ser usado para programar e controlar o robô, dispensando a necessidade de instalação de softwares. Com relação à programação remota de robôs, frequentemente se utilizam os termos *programação remota* ou *tele-programação*. A origem dessas técnicas ocorreu naturalmente como próximo passo após os robôs controlados remotamente, que já existem desde os anos 1940 [Goldberg & Siegart 2001].

Um dos primeiros projetos de programação remota de robôs surgiu em 2001 [Belousov et al. 2001] com o objetivo de simplificar o problema de controle de robôs. Posteriormente, Lowe e Cambrell [Lowe & Cambrell 2002b] apresentaram um sistema que permitia enviar comandos de movimento remoto via web usando *Applets* Java. Nesse sistema, a programação remota do robô era feita através de uma conexão de terminal remoto (telnet) com um servidor que ficava conectado ao robô via porta serial RS-232, sendo que através dessa conexão era possível acessar o interpretador de comandos proprietário do robô.

Mootien et al. (2006) propõem um robô educacional operado remotamente para permitir à estudantes a escrita e execução de programas a qualquer momento. A implementação desse sistema não requer nenhum desenvolvimento de software, já que uma ferramenta pronta, o *Virtual Network Computing* (VNC), foi usada em um computador conectado ao robô para dar acesso remoto à interface gráfica de programação e controle do robô.

Lopez et al. (2009) descrevem um sistema que permite acesso remoto para escrever, compilar e executar programas usando uma conexão de terminal remoto seguro (SSH). Para complementar a conexão SSH, que oferece apenas comandos em modo texto, uma página web oferece acesso à visão da câmera e um sistema de troca de mensagens entre os usuários conectados no sistema. O robô pode ser programado usando as linguagens Ada ou C.

Outro projeto que oferece a visualização de imagens da visão robô em tempo real requer a instalação de uma aplicação Java no computador do usuário, que atua como cliente remoto do robô a ser programado. Essa aplicação permite a edição e compilação dos programas de controle, que são posteriormente enviados via rede para o robô. O sistema suporta apenas scripts V+, uma linguagem proprietária e usada apenas com uma marca específica de robôs [Alifragis et al. 2008].

Marin et al. (2005) descrevem um ambiente virtual com recurso de programação remota. O sistema oferece um servidor de programação remota com uma interface de comunicação via *sockets* TCP/IP, de forma que os programas clientes podem ser escritos em qualquer linguagem que se comunique via TCP/IP com esse servidor. Assim, apenas comandos de alto nível previamente estabelecidos podem ser trocados entre o software remoto e o servidor. No mesmo trabalho, os autores comparam seu projeto com diversos outros, e argumentam que o ambiente proposto é o único com capacidade de programação remota.

Fernandez e Casals apresentam um conceito de laboratório *on line* que permite aos usuários a edição, publicação e execução de programas remotamente [Fernandez & Casals 2004]. Todo sistema funciona em um servidor web, de forma que o cliente precisa utilizar apenas um navegador web com suporte a Java para acessar o ambiente.

Um projeto de destaque é o *Mindstorms Internet Control Environment (MICE)* [Garrett & Thornton 2005], que permite programar e controlar kits Lego via Web usando uma variação da linguagem C. Outro projeto interessante, criado pelo Google e atualmente mantido por pesquisadores do MIT, é o *MIT App Inventor*, um sistema web que permite criar aplicações para telefones celulares com sistema operacional Android de forma gráfica (conectando blocos), incluindo comandos para controle de Legos Mindstorms, e em seguida gerar um pacote de instalação para ser instalado em telefones com o sistema An-

droid [MIT 2012].

O *MIT App Inventor* explora a possibilidade já mencionada no Capítulo 2 que trata do uso de um telefone celular como unidade de controle de um robô. A única dificuldade imposta por esse sistema, é que cada vez que o aplicativo de controle é modificado, é necessário gerar um novo pacote de instalação, apagar o aplicativo antigo do telefone com sistema Android, copiar o novo instalador para o telefone, e executar sua instalação.

Popescu et al. (2008) apresentam um trabalho completo, composto por um robô móvel que inclui um ambiente de programação remota via web embarcado em um servidor web localizado no robô. Os sensores e atuadores do robô são gerenciados por um microcontrolador de baixa performance que é conectado via porta serial a um *tablet* PC também instalado no robô. O PC possui conexão wifi e executa um servidor web para permitir acesso remoto ao robô, de forma que qualquer usuário que possua um computador com navegador web e suporte a Java pode acessar o sistema.

Como mencionado nesta revisão da bibliografia, vários trabalhos de programação remota de robôs já foram propostos, mas esses projetos são limitados a linguagens de programação específicas e requerem navegadores web com suporte a Java, que infelizmente não costuma estar disponível em *tablets* e *netbooks* de baixo custo, limitando o seu uso. No Capítulo 4 apresentamos uma solução que une o melhor de cada trabalho mencionado nesta revisão, removendo a necessidade de Java no navegador do cliente, oferecendo a opção de uso de diversas linguagens de programação, tanto baseadas em blocos quanto em modo textual.

3.3 Comunicação Robô-Computador

Esta Seção revisa alguns trabalhos relevantes relacionados ao uso de dispositivos móveis para controlar robôs, bem como seus canais de comunicação.

3.3.1 Interfaces de Dados Digitais

As interfaces digitais são as mais comuns e amplamente usadas atualmente. Destacamos o uso de portas seriais, USB ou tecnologias sem fio como *bluetooth* para que a unidade de controle troque dados com os circuitos de controle de um robô.

Santos et al. (2010) conectam um telefone celular aos circuitos de controle de um robô usando *bluetooth*, e implementam no telefone complexos algoritmos de navegação robótica com restrições de tempo real, tais como, filtros potenciais, filtro de partículas e filtro estendido de Kalman.

Outro exemplo do uso de *smartphones* como computador de bordo e controle de robôs encontra-se no projeto pioneiro *CellBots* [CellBots 2011], que foi o primeiro a utilizar telefones com o sistema Android para controlar robôs. Nesse projeto, o software do telefone comunica-se com um microcontrolador instalado no robô, normalmente um Arduino, via porta serial ou *bluetooth*. Para tanto, em alguns casos, é necessário desmontar os celulares para poder conectar os sinais da porta serial internamente. No caso do *bluetooth*, os custos são mais altos, pois é necessário equipar o robô com um microcontrolador e um módulo *bluetooth*, elevando os custos.

O trabalho de Hess e Rohrig [Hess & Rohrig 2009] consiste em usar telefones móveis para controlar robôs remotamente através de canais de rede TCP/IP ou *bluetooth*. Park et al. (2008) descrevem técnicas para o desenvolvimento de interfaces com o usuário usando PDAs ou *smartphones* para controlar robôs remotamente. Novamente, o controlador original do robô é mantido, e o dispositivo móvel é simplesmente usado como controle remoto via rede sem fios.

Já com relação ao uso de *smartphones* como controladores de robôs através de interfaces digitais, algumas empresas já oferecem produtos comerciais usando esse conceito. Nesta categoria, o Botiful é um robô de telepresença usando conexão *bluetooth* com telefones com o sistema Android [Honig 2012]. Ele custa US\$ 299,00 e funciona integrado ao Skype, assim, ao fazer uma chamada de vídeo para o robô via Skype, é possível visualizar a imagem remota vista pela câmera do robô, e a partir do próprio Skype, enviar comandos de movimentação para o robô. Já a empresa OLogic oferece robôs modulares que podem ser controlados por um iPhone ou um telefone com Android [Inc 2012]. Chamado de Oddwerx, tal robô também comunica-se com o telefone de controle via *bluetooth*, e já possui diversos recursos, como o controle de atuadores e leitura de sensores, compatibilidade com o sistema operacional para robôs (ROS), e com vários sistemas de robótica nas nuvens.

Uma equipe do Instituto Federal do Espírito Santo também desenvolveu um robô controlado por um *smartphone* baseado em Android [Martins et al. 2012]. O robô criado por eles utiliza uma placa chamada IOIO, que possui um microcontrolador PIC e conecta-se ao celular via porta USB. O microcontrolador atua como intermediário, e permite ao celular acionar os atuadores e ler os sensores do robô. O robô ganhou várias competições de robótica, e foi premiado como o melhor robô na categoria *Rescue B* [Martins 2012] durante a competição mundial de robótica (RoboCup), que ocorreu no México em 2012.

As formas de conexão descritas nesta Seção são de fato aquelas predominantes para conexão de computadores com sistemas robóticos, sendo assim inviável listar todos os trabalhos nessa categoria. Além disso, em geral, esses sistemas de conexão não representam novidade tecnológica ou científica.

3.3.2 Interfaces Analógicas Usando Áudio

Como a maioria dos dispositivos móveis possuem interfaces de áudio, uma alternativa interessante é o uso de um circuito dedicado para converter as suas saídas de áudio em sinal serial, compatível com protocolos seriais padronizados [Rob 2012b], como o RS-232, por exemplo. O problema é que em geral, esta abordagem somente permite comunicação unidirecional [Kranjec 2010, Rob 2012b], e além do mais, requer um microcontrolador para receber e decodificar o sinal serial e então executar alguma ação. Especificamente para o iPhone, Sasaki (2011) criou um modem de áudio, que conecta-se na saída de fone de ouvidos do iPhone, e permite a troca de dados seriais de forma bidirecional usando a interface de áudio do iPhone. Baseado neste projeto, Pascual (2012a) desenvolveu o projeto Androino [Pascual 2012a, Pascual 2012b] que oferece um modem via áudio para integrar dispositivos Android com o Arduino via interface de áudio.

Recentemente, Robinson et al. (2010) criaram o Hijack, que oferece uma melhoria

significativa em relação ao sistema mencionado anteriormente. A técnica de [Kuo et al. 2010] permite a troca de dados bidirecional via áudio entre um microcontrolador e um telefone móvel, bem como a obtenção de energia para sua alimentação usando o próprio sinal de áudio. Além de publicado, o resultado da pesquisa de Kuo et Al. hoje também é oferecido como produto comercial [SeedStudio 2011] com custo de cerca de 70 dólares. Várias aplicações já foram implementadas e demonstradas usando o Hijack, incluindo um sistema de eletrocardiograma de baixo custo usando um iPhone, um osciloscópio didático, e um sistema de acionamento de dispositivos.

Uma abordagem muito usada para controle remoto de equipamentos é através da transmissão de tons DTMF, já descritos no Capítulo 2. Assim, o uso de DTMF para controlar robôs remotamente já foi explorado em diversos trabalhos, como no sistema proposto por Patil & Henry (2008). Manikandan et al. (2010) apresentam um robô que utiliza dois telefones celulares: um instalado no robô, e outro que atua como controle remoto. Nos projetos mencionados, o robô é controlado através de tons DTMF gerados por toques nas teclas do telefone remoto. Um circuito decodificador de DTMF recebe o sinal de áudio proveniente da saída de fones de ouvidos do celular, e que fornece códigos de controle para um microcontrolador que interpreta e executa os movimentos desejados. Sai & Sivaramakrishnan (2009) usam a mesma configuração, mas em um robô com arquitetura diferente. Naskar et al. (2011) também apresentam o uso de um teclado DTMF remoto para controlar um robô militar, onde algumas teclas são usadas para acionar o disparo de armas reais. Nesse último trabalho, ao invés de usar telefones, os sinais DTMF são transmitidos através de um canal de rádio.

Ainda com relação a controle remoto usando DTMF, Ladwa et al. (2009) propõem um sistema que controla remotamente utensílios domésticos e robôs por meio de ligações telefônicas. Um trabalho similar é apresentado por Cho & Jeon (2008), onde as teclas de um telefone remoto são diretamente mapeadas para movimentos básicos de um robô, a saber: frente, ré, esquerda e direita.

No contexto do projeto CellBots [CellBots 2011], Darrell (2010a) criou um sistema capaz de controlar servo-motores com entrada PWM diretamente a partir da placa de som de computadores, sem o uso de qualquer componente eletrônico intermediário. O controle de velocidade e rotação dos motores é feito pelo ajuste da frequência sonora produzida por um software que controla a saída de som. A técnica criada por Darrel é de fato a forma mais barata de controlar motores com *smartphones*, porém ela possui um acoplamento elétrico inadequado entre os componentes do sistema e pode causar danos permanentes aos aparelhos envolvidos.

Em uma evolução desse conceito, Darrel e a equipe do CellBots criaram um circuito com acopladores ópticos [Darrell 2010b, Taylor 2011] para proteger o telefone de controle. Embora seguro, o método funciona em poucos telefones, já que requer a geração de um padrão de onda na saída de áudio que poucos dispositivos podem oferecer. Além disso, esta abordagem funciona porque os servos possuem a eletrônica de potência e acionamento embutidas no motor, assim o telefone apenas gerencia o sinal de controle. Usando a mesma técnica desenvolvida por Darrel, Carter (2011) apresenta um robô composto de dois servo-motores alimentados por pilhas e controlados diretamente pelo canal de áudio estéreo de um *iPhone*. O canal de áudio direito controla o motor direito, e o

canal esquerdo, o motor esquerdo.

Em 2011, a Romotive, uma *startup* americana criou um robô móvel chamado Romo que pode ser controlado por tons de áudio de dispositivos móveis [Romotive 2011]. O sistema somente controla motores, sem nenhum canal de retorno para leitura de sensores. De acordo com os fundadores da empresa, os sensores não seriam necessários já que todo controle do robô pode ser feito usando apenas os sensores de um *smartphone*, embora eles mesmos admitam que a falta de sensores convencionais de robótica podem dificultar a implementação dos softwares para o Romo [Seid et al. 2012].

Mais de 2000 Romos foram vendidos em menos de um ano [Seid et al. 2012], e em Outubro de 2012 a empresa iniciou o desenvolvimento da nova versão do Romo, que já está em pré-venda por US\$ 150,00 [Seid et al. 2012]. Ainda sem suporte a sensores externos, o novo Romo também deixou de suportar telefones Android, funcionando apenas com dispositivos da *Apple*. A ênfase da nova versão é um suporte de software mais completo que deve incluir telepresença remota de duas vias, visão computacional, navegação autônoma e reconhecimento de faces [Seid et al. 2012].

O uso da interface de áudio de telefones móveis para controle remoto de brinquedos também começou a ser explorada recentemente por empresas, já que além de reduzir os custos dos brinquedos, por dispensar o fornecimento de um controle remoto, também contempla a possibilidade de oferecer interfaces gráficas mais ricas para interação com o usuário. Exemplos são o carrinho de controle remoto CarBot [Siu 2012] e o um helicóptero de controle remoto [DealExtreme 2012]. Esse dois produtos são controlados remotamente via transmissão na faixa do infravermelho, e para seu controle, um pequeno transmissor é ligado à saída de fones de ouvido do celular de controle. Um software fornecido pelos fabricantes oferece telas com os comandos de controle dos veículos, gerando sinais sonoros de acordo com cada comando dado pelo usuário, que são convertidos e transmitidos via infravermelho.

3.4 Resumo

As Tabelas 3.1, 3.2 e 3.3 mostram os trabalhos relacionados citados neste capítulo de forma comparativa e resumida. Na Tabela 3.1, são listados robôs considerados de baixo custo pelos seus autores. No último item dessa tabela, a observação (*) significa que o custo foi estimado de acordo com a informação dos autores, de que o projeto deles custa 1/4 do custo de um Kit Lego *Mindstorms*.

Tabela 3.1: Tabela comparativa de robôs de “baixo custo”.

Robô	Custo (US\$)	Características	Trabalho
Suckerbot	9	Motores, seguidor de linha, sensor de toque omnidirecional, alimentação e controle via USB	[Tilley 2012]
Continua na próxima página			

Tabela 3.1 – continuação da página anterior

Robô	Custo (US\$)	Características	Trabalho
Baobot	31	Motores, seguidor de linha, compatível com Arduino, alimentação e controle via USB ou operação autônoma	[H. et al. 2012]
Afrobot	10	Microcontrolador MSP430, motores, seguidor de luz, detector de obstáculos	[Reben 2012]
RoboArm	50	Braço robótico, interface com computador via sensores de luz na tela	[O.M. 2012]
Kilobot	43	Locomoção por vibração, comunicação em grupo e com PC, microcontrolador	[Rubenstein et al. 2012]
SwarmRobot	10	Compatível com Arduino, motores, sensor de objetos e de obstáculos	[Michaelis et al. 2012]
SEG	14	Estilo <i>Segway</i> , microcontrolador, motores, rodas montadas como Origami	[Hamer et al. 2012]
DiscBot	47	Motores de passo, compatível com Arduino, sonar, estrutura de madeira cortada à laser	[Schlesinger 2012]
MITBOTS	33	Processador ARM, tela de LCD, blocos de montagem, motores	[Singh et al. 2012]
N-Bot	14	Motores, controlado por canal de áudio, compatível com celulares ou computadores	[Aroca, Pitta, Burlamaqui & Gonçalves 2012a]
Sem Nome	150	Microcontrolador, uso de odometria para localização	[do Nascimento et al. 2012]
SH-EDUC	84	Uso de sucata em conjunto com Microcontrolador PIC, tela de LCD, sonar, sensores de luz, toque e temperatura	[Fernandes et al. 2012, de Lima Sá 2011, Thomaz et al. 2012]
Sem Nome	Não informado	Uso de sucata em conjunto com microcontrolador e motores	[Botelho et al. 2012]
R-One	200	MSP430, motores, giroscópio, acelerômetro	[Ric 2011, Jobe et al. 2012]

Continua na próxima página

Tabela 3.1 – continuação da página anterior

Robô	Custo (US\$)	Características	Trabalho
Finhch	99	Motores, acelerômetro, sensores de luminosidade e toque, LEDs, câmera, microfone e alto falantes	[Ino 2012, Lauwers & Nourbakhsh 2010]
Lego Mindstorms	640	Lego RCX, motores, diversos sensores	[Galvan et al. 2006, Greenwald & Kopena 2003]
Sem Nome	> 500	Usa Lego Mindstorms + Microcontrolador + câmera + sonar. Sistema de visão rastreia objetos	[Dinh & Inanc 2009]
Sem Nome	200-2000	Reaproveita estrutura e motor de brinquedos, uso de PCs, FPGAs e microcontroladores, sensores	[Hamblen & Hall 2004]
Sem Nome	755	Microcontrolador, sensor de distancia, GPS, bússola, LCD, motores	[Lumsden & Ortega-Sanchez 2010]
LUDEBot	250 (*)	Servos, compatível com Arduino, sonar, PC opcional	[Pin et al. 2012]

Características	Trabalho
Uso de ferramentas de acesso remoto (SSH e VNC)	[Mootien et al. 2006, Lopez et al. 2009]
Programação remota via web (requer Java)	[Fernandez & Casals 2004, Alifragis et al. 2008, Belousov et al. 2001, Lowe & Cambrell 2002a]
Programação remota via web para Lego Mindstorms	[Garrett & Thornton 2005, Aroca, Guardiman & Gonçalves 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c]
Programação via web, e geração de aplicativo para controlar Lego Mindstorms via telefone com Android	[MIT 2012]
Servidor web embarcado em um robô móvel	[Popescu et al. 2008, Aroca, Guardiman & Gonçalves 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c]

Tabela 3.2: Tabela comparativa de sistemas de programação remota de robôs.

Categoria	Características	Trabalho
Bluetooth	Conexão <i>bluetooth</i> entre circuito de controle do robô e computador de controle	[Santos et al. 2010, CellBots 2011, Honig 2012, Inc 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c, Aroca, Guardiman & Gonçalves 2012]
Cabo serial ou USB	Conexão com cabo USB ou serial entre circuito de controle do robô e computador de controle	[CellBots 2011, Martins 2012]
Áudio	Conexão entre computador de controle e circuitos de acionamento de atuadores e leitura de sensores via áudio	[Rob 2012b, Kuo et al. 2010, Darrell 2010b, Darrell 2010a, Taylor 2011, Seid et al. 2012, Romotive 2011, Siu 2012, DealExtreme 2012, Aroca, Guardiman & Gonçalves 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c]
Áudio Remoto	Controle remoto via DTMF	[Hess & Rohrig 2009, Patil & Henry 2008, Manikandan et al. 2010, Sai & Sivaramakrishnan 2009, Naskar et al. 2011, Ladwa et al. 2009, Cho & Jeon 2008]

Tabela 3.3: Tabela resumo de técnicas comuns de conexão entre computador e robô.

A revisão feita neste capítulo demonstrou a importância do uso de robôs para motivar alunos, e a demanda por plataformas robóticas com custo mais acessível, que vem sendo explorada por vários pesquisadores. Também foram revisados aspectos técnicos apresentados em trabalhos recentes para construção de robôs usando computadores e dispositivos móveis como unidades de controle, e interfaces de áudio. Assim, pode-se notar pelas referências, o ressurgimento do interesse no uso de interfaces de áudio, que estavam deixando de ser usadas em detrimento do uso de sistemas digitais mais modernos. Contudo, devido a sua simplicidade, pôde-se notar o surgimento recente de diversos trabalhos e produtos que exploram o uso de interfaces de áudio para troca de dados.

O Capítulo 4 descreve maiores detalhes da técnica proposta nesse trabalho e sua implementação, onde são apresentadas várias possibilidades de circuitos de controle para o N-Bot, permitindo controlar dois motores por canal de áudio, ou quatro motores por canal de áudio estéreo em versões sem microcontrolador, bem como receber dados de diversos sensores. Nesse circuito, é possível construir uma rede de microcontroladores e efetuar controle de atuadores e leitura de sensores de forma distribuída, onde cada microcontrolador pode comandar até oito servo motores. Vale lembrar, que um Lego, por exemplo, suporta até três motores ligados diretamente a unidade de controle do NXT e até quatro sensores.

Capítulo 4

Sistema Proposto

Make things as simple as possible, but not simpler.

Albert Einstein

Este Capítulo descreve os aspectos de hardware e software da plataforma de robótica de baixíssimo custo proposta neste trabalho. A ênfase está no projeto de um circuito simples, de baixíssimo custo, que pode ser construído e replicado com peças comuns disponíveis no mercado. Com base neste circuito, é possível usar programas didáticos ou de pesquisa com alta sofisticação para controle do N-Bot.

Primeiro são apresentadas as técnicas e circuitos para leitura de sensores e controle de atuadores usando áudio, e em seguida são apresentados os softwares do sistema. Com realação aos softwares, foram desenvolvidos dois ambientes de programação independentes que são descritos neste capítulo: o *Android Web-based Integrated Development Environment* (Anwide), que permite a programação remota, via web, de robôs controlados por telefones com o sistema operacional Android. Como alternativa para programação e controle dos robôs sem o uso de telefones foi implementado o Ambiente de Desenvolvimento Web do N-Bot (ADWN).

Vale mencionar que alguns aspectos das técnicas descritas aqui foram patenteadas junto ao INPI e/ou publicadas em periódicos relevantes da literatura, o que reforça o fato de estarmos apresentando técnicas e soluções inéditas. Ao longo deste Capítulo, o termo *dispositivo de controle* refere-se a todo dispositivo com saída e/ou entrada de áudio, que pode ser usado como controlador do sistema proposto, tal como um PDA, telefone, *tablet*, computador, dentre outros.

4.1 Hardware

O robô construído (N-Bot) é uma plataforma para robótica educacional desenvolvida visando tornar a robótica mais popular no meio estudantil, entre aficionados e mesmo entre leigos no assunto. A Figura 4.1 ilustra de forma esquemática a arquitetura da plataforma proposta neste trabalho. Como descrito anteriormente, resumidamente, a plataforma usa o canal de áudio como interface de controle. Como visto na Figura 4.1,

frequências de áudio são usadas para enviar comandos aos atuadores e também para receber leituras dos sensores do robô, que são analisados por um computador de controle (ou por um telefone celular ou algum dos dispositivos citados acima). Também, pode-se usar uma página web implementada por nós, onde estudantes podem escrever os programas que irão gerar e receber áudio do robô visando executar os controles desejados.

A Figura 4.1 também mostra que o dispositivo de controle, que pode ser um computador, *tablet*, telefone, dentre outros, e ser acessado e programado remotamente via rede local ou Internet, bem como pode se aproveitar de serviços de robótica na nuvem. O robô pode ser programado tanto com o Ambiente de Desenvolvimento Web do N-Bot (ADWN), quanto pelo *Android Web-based Integrated Development Environment* (Anwide), ou qualquer outra linguagem de programação que possa gerar e capturar áudio. Assim, essa arquitetura e a plataforma desenvolvida permitem que qualquer computador possa ser usado como o *cérebro* do robô, sem a necessidade de instalar quaisquer softwares adicionais ou *driver* de dispositivo. Isso torna nossa proposta fácil de ser usada em computadores públicos e mesmo por pessoas sem muita experiência, inclusive em computadores onde os usuários não têm permissão para instalar softwares. A seguir, detalharemos cada um dos componentes do hardware.

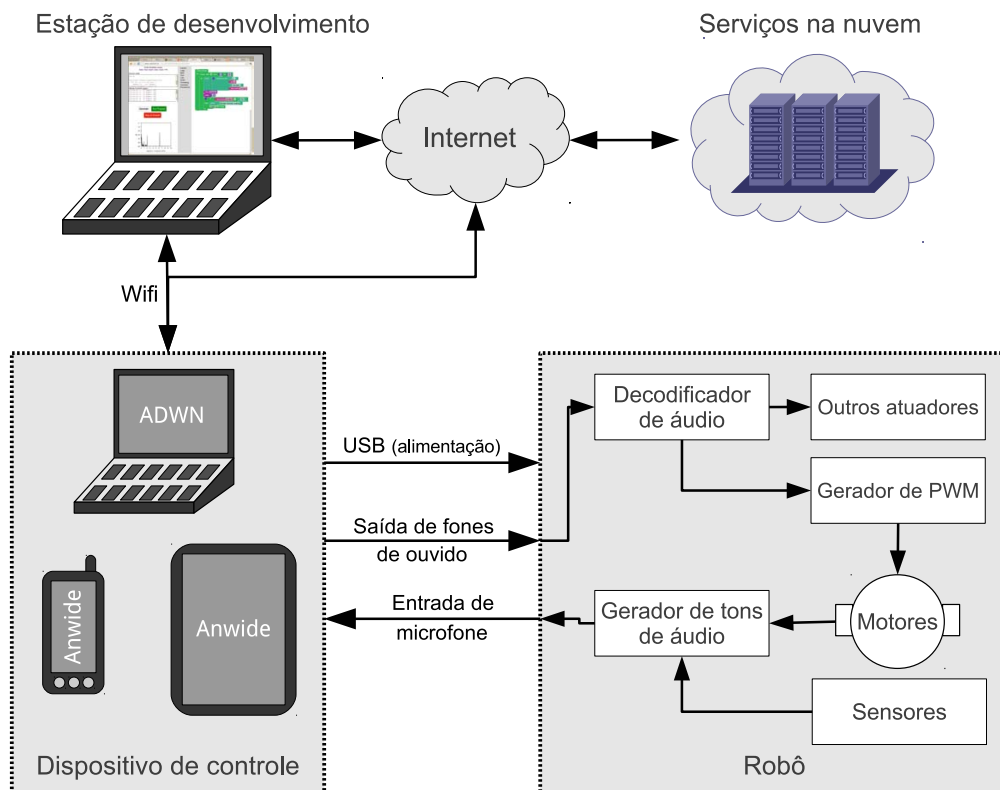


Figura 4.1: Visão geral do sistema e suas conexões entre possíveis dispositivos de controle, desenvolvimento e de controle.

4.1.1 Controle de Atuadores Usando Áudio

Para controlar atuadores, um programa instalado no dispositivo de controle gera um tom DTMF. Esse tom é decodificado por um decodificador de DTMF, como o MT8870 ou HT9170, resultando em uma saída paralela composta por 4 bits, que codificam um valor binário equivalente ao dígito DTMF presente na entrada do sistema. O estado decodificado fica presente na saída do decodificador enquanto o tom de áudio DTMF estiver presente na entrada do circuito. Esses quatro bits podem ser conectados em circuitos para controle de cargas, permitindo o controle independente e simultâneo do tipo liga/desliga de até quatro dispositivos, tais como freios eletro-mecânicos, luzes ou uma garra pneumática. A Figura 4.2 mostra o conceito básico do sistema.

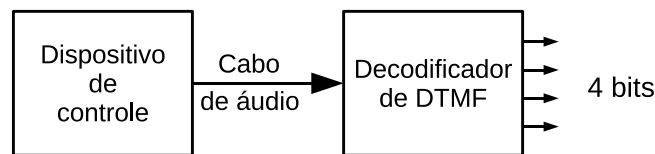


Figura 4.2: Decodificador de DTMF com quatro saídas independentes controladas pelo áudio de um dispositivo de controle.

Como os decodificadores de DTMF, em geral, já possuem um amplificador interno, na maioria dos casos não é necessário etapas de pré-amplificação do sinal de áudio de entrada, entretanto, pode existir situações e aparelhos com nível de saída muito tênues, que demandem um pré amplificador de áudio entre o dispositivo de controle e o decodificador de DTMF.

Controle de Motores de Corrente Contínua

A Figura 4.3 mostra uma aplicação da técnica descrita para controlar motores de corrente contínua (DC). Os motores são controlados por uma ponte H comercial, como a L298, que requer 2 bits de controle para cada motor. Assim, os 4 bits provenientes do decodificador de DTMF podem ser usados para controlar dois motores de forma independente. A Tabela 4.1 mostra os possíveis dígitos DTMF gerados pelo dispositivo de controle e o estado resultante dos motores, porém deve-se notar que esses estados podem variar de acordo com a ordem de conexão dos bits de saída com a entrada do decodificador de DTMF e da ponte H. Para controlar a velocidade dos motores, o dispositivo de controle pode alternar dígitos DTMF em uma determinada frequência, de forma a gerar um sinal de modulação por largura de pulso (PWM).

A Figura 4.4 mostra o diagrama elétrico completo do circuito de controle para dois motores. Neste diagrama, pode ser visto um pré-amplificador de áudio com transistores, que pode ser omitido na maioria dos casos, ou substituído por um amplificador operacional. A saída do amplificador é conectada ao decodificador de DTMF (MT8870), que gera 4 bits, sendo esses conectados à entrada da ponte H (L298), que é responsável por controlar os motores.

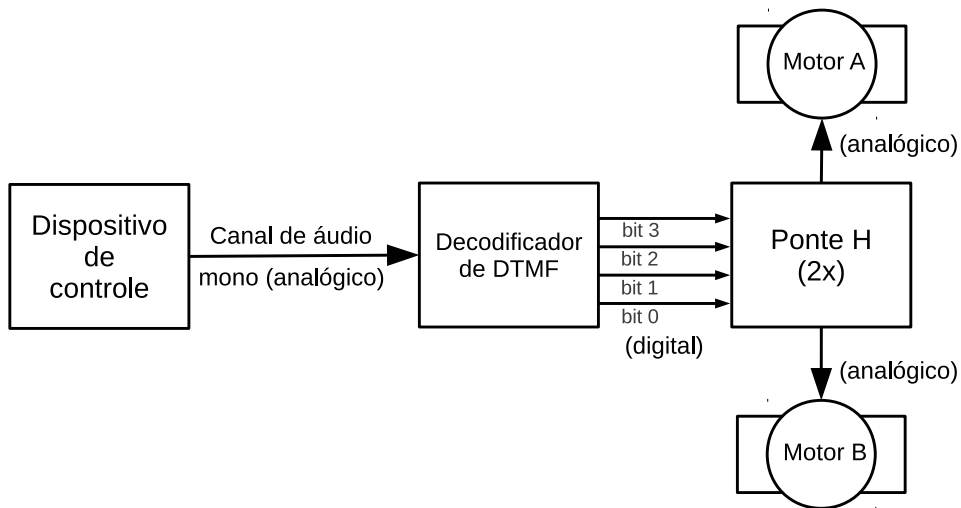


Figura 4.3: Controle de dois motores de corrente contínua usando um canal de áudio mono e tons DTMF.

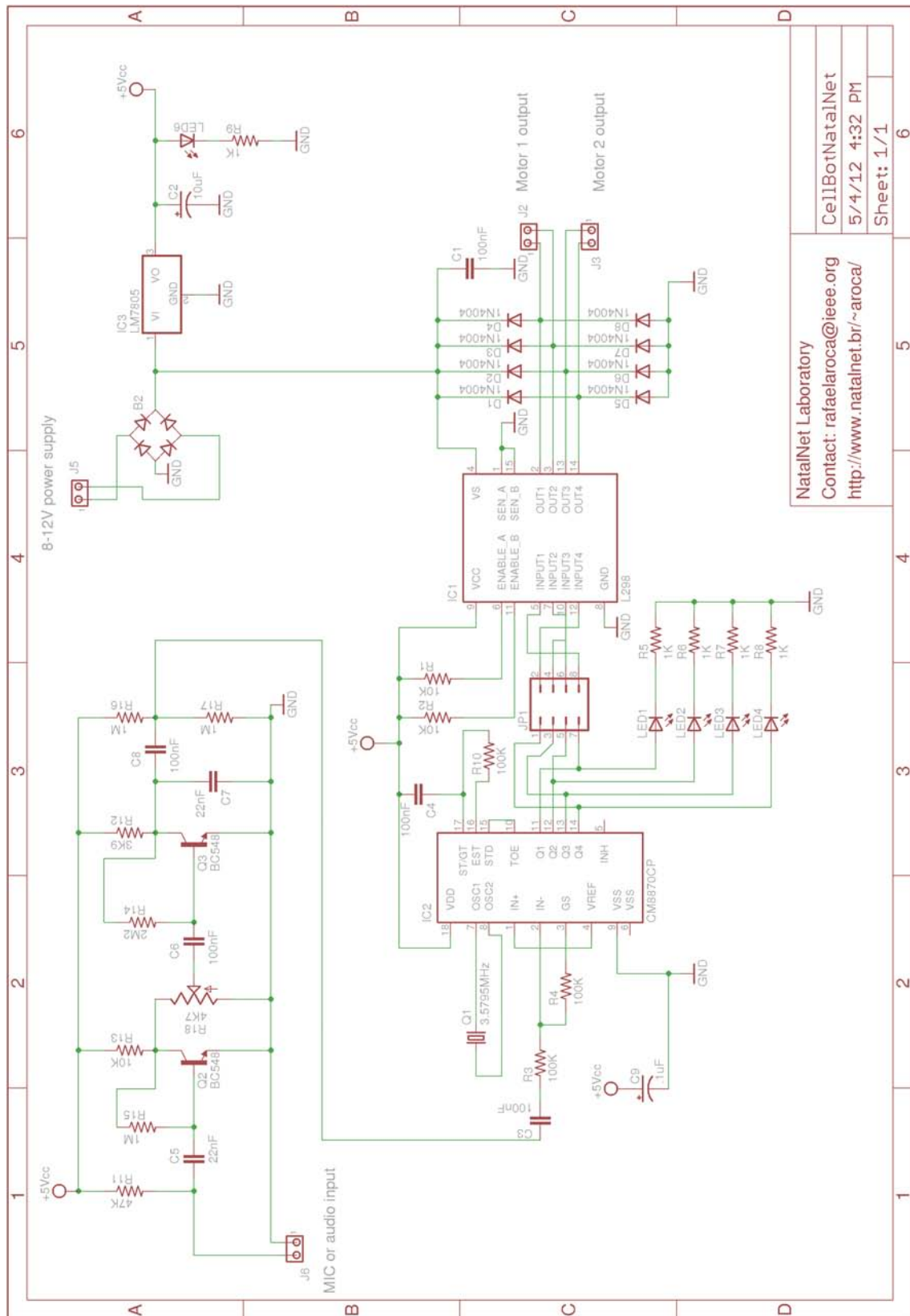
Uma das vantagens desse circuito é que ele pode controlar desde pequenos motores até aqueles de alto consumo de corrente. Contudo, deve-se escolher a ponte H adequada para cada tipo de motor. O L298, por exemplo, suporta motores que consumem até 3A de corrente. Os diagramas elétricos em formato Eagle e o desenho da placa de circuito impresso deste circuito estão disponíveis *on-line* no site do laboratório NatalNet.

A técnica apresentada requer apenas um canal de áudio mono. Dessa forma, como a maioria dos dispositivos possuem um canal de áudio estéreo, até quatro motores podem ser controlados por um canal de áudio estéreo. Além disso, o sistema também pode ser facilmente expandido adicionando novas saídas de áudio através de placas de som USB de baixo custo, que funcionam em computadores tradicionais e dispositivos com o sistema Android.

Controle de Servo Motores

Embora o circuito descrito na subseção anterior tenha o custo baixo, decidimos buscar alternativas para reduzir ainda mais os custos. Para tanto, optou-se pelo uso de servomotores usados em aeromodelismo e outros veículos de controle remoto. Esses servos oferecem um ótimo custo-benefício, pois o próprio motor já possui uma caixa de redução, permitindo a conexão direta do eixo do motor com rodas ou braços mecânicos e também possui um circuito de controle, dispensando o uso de uma ponte H externa. Dessa forma, esses servos possuem três fios de conexão: dois para alimentação, e um para receber o sinal de controle, que é semelhante a um sinal PWM.

De acordo com a especificação dos servos, o sinal de controle deve ter um período total de 20ms. Para o motor girar no sentido anti-horário, o sinal deve ficar em nível alto por um período entre 1.7ms e 2.0ms. Para o motor girar no sentido horário, o sinal deve



NataNet Laboratory
 Contact: rafaelaroca@ieee.org
 http://www.natanelab.br/~aroca/

Figura 4.4: Diagrama elétrico do circuito para controle de dois motores DC com base em um sinal de áudio.

Digito DTMF	Palavra binária	Movimento do motor	
		Motor A	Motor B
Nenhum	00 00	Parado	Parado
D	00 00	Parado	Parado
1	00 01	Parado	Esquerda
2	00 10	Parado	Direita
3	00 11	Parado	Inválido
4	01 00	Esquerda	Parado
5	01 01	Esquerda	Esquerda
6	01 10	Esquerda	Direita
7	01 11	Esquerda	Inválido
8	10 00	Direita	Parado
9	10 01	Direita	Esquerda
0	10 10	Direita	Direita
*	10 11	Direita	Inválido
#	11 00	Inválido	Parado
A	11 01	Inválido	Esquerda
B	11 10	Inválido	Direita
C	11 11	Inválido	Inválido

Tabela 4.1: Dígitos DTMF de controle e estados dos motores.

ficar em nível alto entre 0.7ms e 1.0ms. Através de diversos experimentos, observou-se que poucos dispositivos podem gerar um sinal com essas características diretamente via software, de forma que optou-se por um circuito gerador de sinal semelhante ao PWM, já que existe uma pequena diferença entre o sinal PWM tradicional e o sinal usado pelos servos. O PWM, na verdade, é um sinal de frequência fixa, onde se ajusta o seu ciclo de trabalho (*duty cycle*), que é a relação entre o período que o sinal fica em nível alto e nível baixo. Já o sinal de controle do servo tem período fixo de 20ms com nível alto no intervalo entre 0.7ms e 2.0ms.

A Figura 4.5 mostra um diagrama de blocos do circuito proposto. Para gerar o sinal de controle, o circuito utiliza o 555, um *timer* de baixíssimo custo, que pode gerar sinais de variadas frequências de acordo com capacitores e resistores de ajuste. Assim, é necessário alterar o valor de um dos resistores conectados ao 555 para modificar o sentido de rotação e velocidade do motor. Para tanto, uma chave analógica (4066N) é utilizada. Essa chave liga ou desliga a conexão de seus terminais de acordo com níveis lógicos de controle. Nas entradas da chave analógica, são conectados dois resistores para cada motor: um para determinar a frequência que resulta em rotação no sentido horário, e outro para determinar a rotação no sentido anti-horário. O sinal DTMF de entrada é decodificado pelo MT8870, e seus bits são usados para controlar a chave analógica.

A Figura 4.6 apresenta o diagrama esquemático desse circuito. Variações no intervalo do áudio enviado para o circuito podem controlar a velocidade dos motores. Os resistores variáveis (*trimpots*) permitem ajuste fino de velocidade e direção de cada motor, e podem

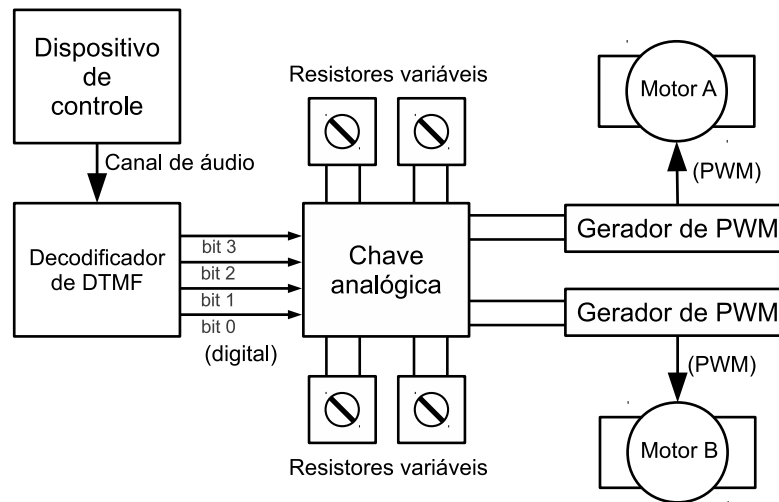


Figura 4.5: Diagrama geral de funcionamento do sistema para controle de servos usando sinal de áudio com tons DTMF.

ser substituídos por resistores convencionais após um valor de resistência adequado para cada motor ser estabelecido. Ao contrário do circuito para controle de motores DC, que possui uma tabela com movimentos pré-determinados para cada dígito DTMF, o circuito de controle de servos permite o ajuste da velocidade e sentido da rotação de cada dígito DTMF por meio dos resistores variáveis.

Assim como mencionado para o circuito anterior, esse sistema é expansível usando canais de áudio adicionais, e o *layout* da placa de circuito impresso também está disponível no site do laboratório NatalNet.

4.1.2 Leitura de Sensores

Para que o N-Bot possa reagir ao ambiente e executar tarefas de acordo com determinadas condições, torna-se interessante o uso de sensores. Mais uma vez, optou-se pelo uso de uma interface de áudio.

Leitura de Sensores Usando DTMF

A primeira solução projetada baseia-se no uso de um codificador de DTMF, um componente eletrônico convencional, amplamente usado em telefones. O codificador de DTMF recebe 4 bits de entrada, normalmente provenientes de um teclado numérico, e gera um sinal DTMF correspondente. Exemplos de codificadores de DTMF são o TCM5087, 5088, MT8880, TEA1015, dentre outros. A Figura 4.7 mostra a ideia geral de funcionamento do sistema.

Para identificar o estado de cada sensor, o software executado no dispositivo de controle executa a transformada de Fourier, que permite identificar as frequências presentes

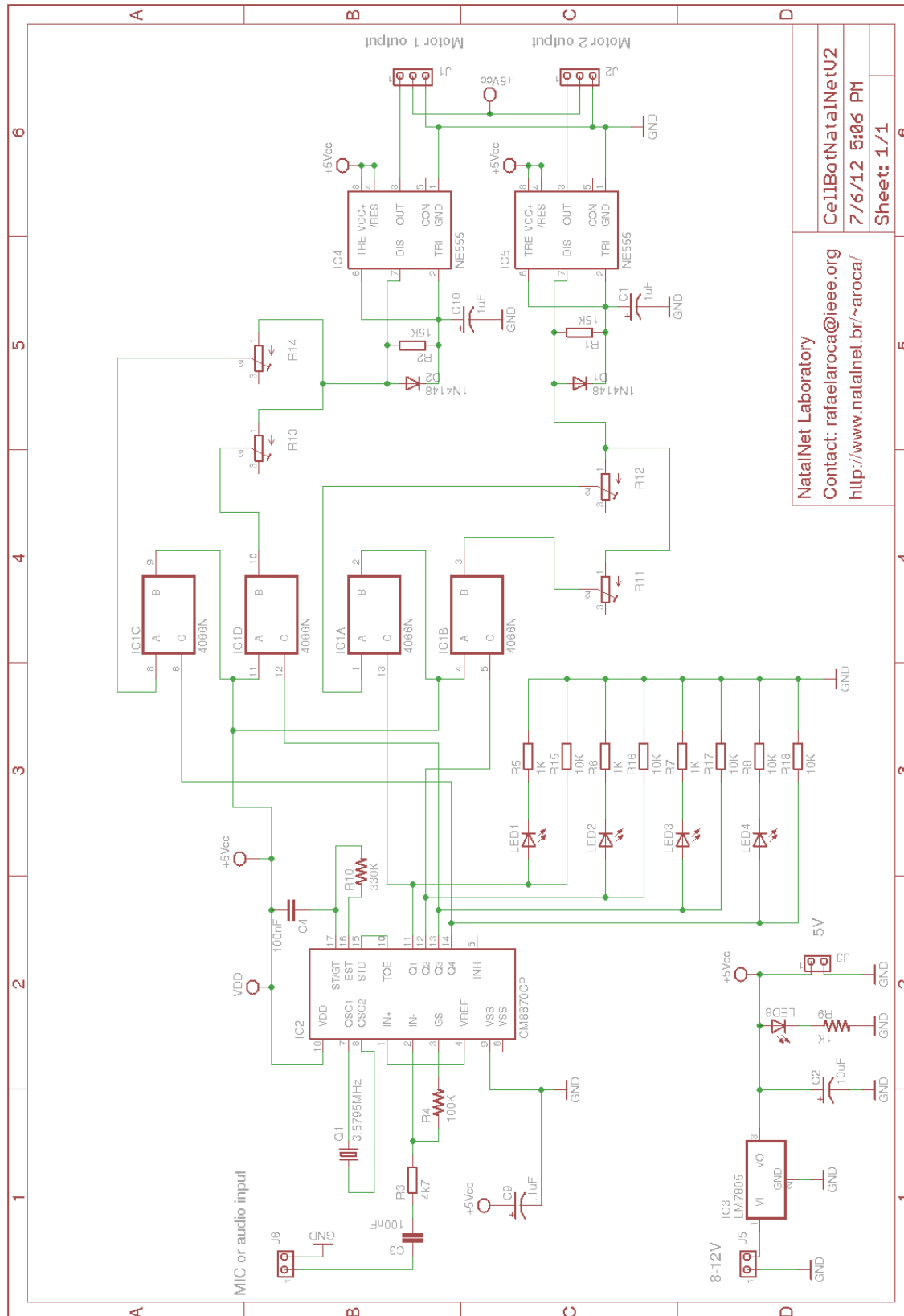


Figura 4.6: Diagrama elétrico do circuito para controle de dois servo-motores a partir de tons de áudio.

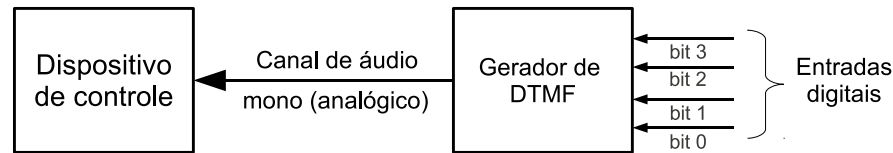


Figura 4.7: Codificação dos dados de sensores usando DTMF.

no sinal DTMF. Por questão de desempenho, optamos pela implementação da transformada rápida de Fourier (FFT), que trata-se de uma DFT calculada de forma mais eficiente. Ainda assim, para obter melhoria adicional no tempo de computação é necessário especificar valores adequados para o número de pontos da FFT (N) e a taxa de amostragem do sinal (F_s). Com base na Tabela 2.1, nota-se que a maior frequência presente em um sinal DTMF é um tom de 1,633Hz. Pelo teorema fundamental da amostragem, sabe-se que a taxa de amostragem deve ser de pelo menos duas vezes a maior frequência presente no sinal a ser analisado, resultando em $F_s \geq 3,266\text{Hz}$. Por conveniência da implementação e melhor compatibilidade, recomenda-se o uso de uma taxa de 8KHz, que pode ser executada por grande parte dos dispositivos de captura de áudio.

Com relação a N , quanto menor o número de pontos, mais rapidamente a FFT pode ser calculada, e conseqüentemente, mais dígitos DTMF podem ser reconhecidos em um intervalo de tempo, levando a uma maior taxa de leitura do estado dos sensores. Contudo, um valor muito pequeno de N reduz a resolução da DFT, causando leituras incorretas. Para determinar o menor valor adequado para N , a Equação 2.2 apresentada no Capítulo 2 é utilizada. Utilizando $F_s = 8\text{KHz}$, em busca do menor N possível, obtém-se $N = 256$, que oferece uma resolução de análise de 30Hz, um intervalo que possibilita diferenciar uma componente DTMF de outra. Esses parâmetros são consistentes com outros trabalhos [Khan 2005, Chitode 2008], que também usam a DFT para identificar tons DTMF.

A Figura 4.8 mostra um exemplo de conexão de quatro sensores ao dispositivo de controle utilizando DTMF. Nesse exemplo, é possível, inclusive, medir o deslocamento e velocidade do robô. Ao invés de usar *encoders* ópticos comerciais, optou-se por desenhar o disco do *encoder* em um computador e imprimi-lo com uma impressora convencional. Um sensor de reflexão de luz (CNY70) é usado para detectar se o disco do *encoder* está posicionado na área branca ou preta, e o software de controle contabiliza os pulsos gerados pelos *encoders* para obter o deslocamento e velocidade. Essas informações podem ser utilizadas em algoritmos clássicos de odometria e localização para determinar a posição do robô em um espaço cartesiano [Siegwart & Nourbakhsh 2004].

A Tabela 4.2 mostra uma tabela verdade com o mapeamento do estado de cada sensor para dígitos DTMF, que pode ser obtido aplicando operações básicas de lógica Booleana ao valor DTMF recebido. Por exemplo, pela Tabela 4.2, sabe-se que os sensores de toque são os bits 0 e 1. Assim, a operação lógica “E” bit a bit com a máscara 0001 filtra os dados de todos os outros sensores, resultando apenas no estado daquele sensor de toque específico. Para o outro sensor de toque, a máscara usada deve ser 0010.

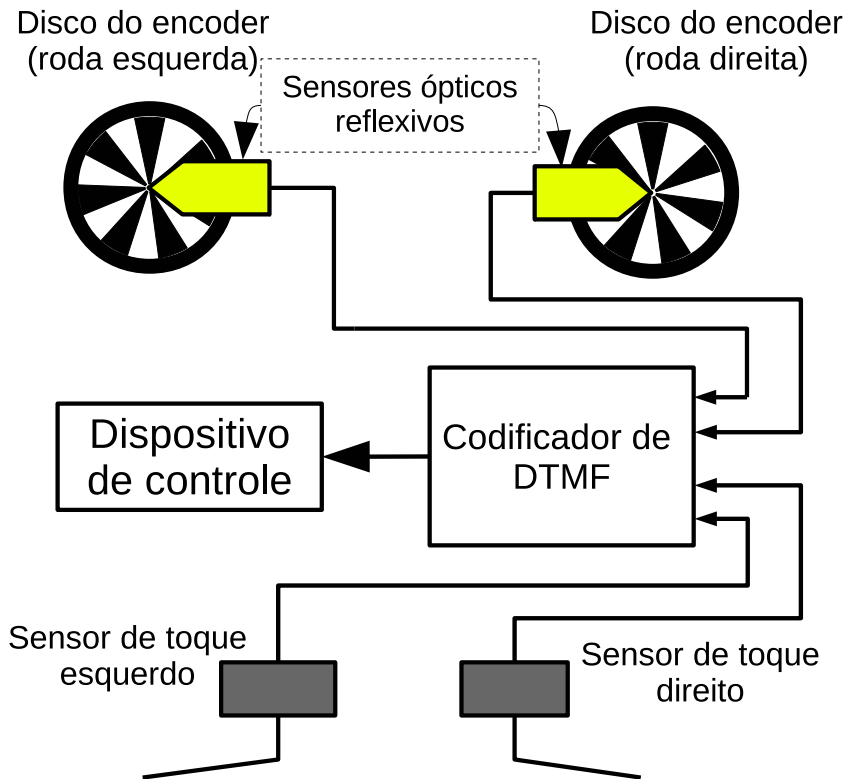


Figura 4.8: Conexão de sensores de um robô a um dispositivo de controle.

Para projetar um robô baseado na técnica de leitura de sensores apresentada, deve-se levar em consideração os limites na taxa de leitura de sensores do sistema. Para tanto, introduzimos uma relação entre dimensão das rodas e a máxima velocidade linear que pode ser medida com o sistema. Na Equação 4.1, V_{Max} é a máxima velocidade linear do robô que pode ser medida, r é o raio da roda, c é a taxa máxima de dígitos DTMF que podem ser analisados pelo sistema em um segundo, e s é a resolução do disco do *encoder*, que refere-se ao número de pulsos por revolução da roda.

$$V_{Max} = \frac{2\pi rc}{s} \quad (4.1)$$

A Tabela 4.3 mostra alguns exemplos da resolução para medida de deslocamento e a máxima velocidade que pode ser medida de acordo com a Equação 4.1.

A Figura 4.9 mostra resultados experimentais realizados no primeiro protótipo do N-Bot. As barras de erro mostram o desvio padrão da distância percorrida, enquanto a linha vermelha mostra a distância medida por um telefone celular conectado aos sensores, a linha azul mostra a distância real percorrida. Cada experimento apresentado no gráfico consiste na média de dez amostras.

De acordo com McComb G.; Predko (2006), erros de odometria são inevitáveis devido

Encoder esquerdo (bit 3)	Encoder direito (bit 2)	Sensor de toque direito (bit 1)	Sensor de toque esquerdo (bit 0)	Dígito DTMF
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	*
1	1	1	1	#

Tabela 4.2: Tabela verdade com mapeamento do estado dos sensores para dígitos DTMF.

a vários fatores, como escorregamento das rodas e pequenos erros de medida no raio da roda que se acumulam com o tempo. Eles afirmam que deslocamento entre 6 e 9 metros resultam em erros no odômetro de 15 centímetros ou mais, que é um erro percentual entre 1.6% e 2.5%. O maior erro de odometria observado durante os testes foi de 3.7% em uma distância de 74cm e 0.75% para uma distância de 0.76cm.

Leitura de Sensores Usando Tons Simples

Embora o sistema descrito acima seja robusto, ele possui a limitação de suportar apenas quatro sensores binários. Assim, uma abordagem mais simples para leitura dos sensores foi idealizada em busca de um projeto mais flexível e expansível. Essa outra técnica é descrita nesta Seção.

A Figura 4.10 ilustra o funcionamento do sistema: cada sensor é conectado a um gerador de sinal com frequência ajustável através de um *trimpot*, permitindo assim o ajuste de uma frequência diferente para cada sensor. O gerador somente gera um sinal enquanto seu sensor correspondente está acionado. As saídas de todos os geradores são misturadas, resultando em um sinal contendo todas as frequências geradas. O software de controle utiliza então a FFT para identificar qual sensor está acionado. A página web do projeto N-Bot [Aroca, Pitta, Burlamaqui & Gonçalves 2012a] contém diversos vídeos mostrando essa técnica funcionando em experimentos práticos.

Qualquer circuito gerador de sinais de áudio pode ser utilizado. Para que o custo fosse reduzido optou-se pelo uso do CI 555, que pode gerar ondas quadradas. Testes demonstra-

Resolução do encoder (pulsos por revolução) (s)	Resolução para medida de deslocamento	Velocidade máxima (V_{Max})
1	157 mm	6283 mm/s
6	26 mm	1047 mm/s
12	13 mm	571 mm/s
24	6,5 mm	261 mm/s
40	3,92 mm	157 mm/s

Tabela 4.3: Resolução do encoder e da medida de deslocamento, e máxima velocidade que pode ser medida (considerando $r = 25$ mm e $c = 40$ dígitos por segundo).

ram a estabilidade e robustez do sistema com o 555, entretanto, como uma consequência do uso de um sinal quadrado, surgem muitas harmônicas no sinal resultante (sinais sobrepostos de duas vezes frequência do sinal base). O tratamento dessas harmônicas deve ser feito pelo software de análise que é executado no dispositivo de controle.

Uma alternativa para redução de custos é o uso de um único gerador de tom para todos os sensores do sistema, como ilustrado na Figura 4.11. Nessa situação, cada sensor é conectado em série com um resistor, que determina sua frequência. Quando vários sensores são acionados simultaneamente, uma única frequência resulta da combinação das duas frequências originais dos dois sensores acionados, sendo necessário um tratamento via software dessa característica. A Figura 4.12 mostra o diagrama elétrico desse circuito.

Os sistemas descritos nesta subseção podem ser usados para capturar o estado de diversos sensores, como, por exemplo, sensores de toque, inclinação, seguidores de linha, fim de curso, botoeiras e sensores de distância ajustados para serem acionados em uma distância específica, além de outros sensores de dois estados (acionado ou não acionado).

Também é possível adaptar o sistema para leitura de sensores analógicos através de um mapeamento da tensão ou resistência dos sensores para uma faixa de frequências. O circuito da Figura 4.12 mostra essa possibilidade aplicada a leitura da posição absoluta das rodas do robô. Isso é possível, pois os servos de baixo custo utilizam resistores variáveis, conectados ao eixo dos motores, como sensores de posição, possibilitando usar a saída desses potenciômetros para determinar a posição das rodas.

Para diferenciar as frequências dos sensores analógicos e digitais, utilizam-se diversos resistores de ajuste para determinar a frequência inicial e final relacionada com um sensor analógico. Essa possibilidade já foi validada experimentalmente, porém ainda não foi explorada em maiores detalhes.

4.1.3 Controle em Malha Fechada Usando Canal de Áudio

Usando o sistema descrito, é possível fechar malhas de controle no N-Bot. Experimentos detalhados no Capítulo 5 mostram que é possível executar malhas de controle com taxa de 40Hz usando sensores externos. Também é possível usar os sensores de um telefone no laço de controle, como, por exemplo, a bússola, para controlar a direção de

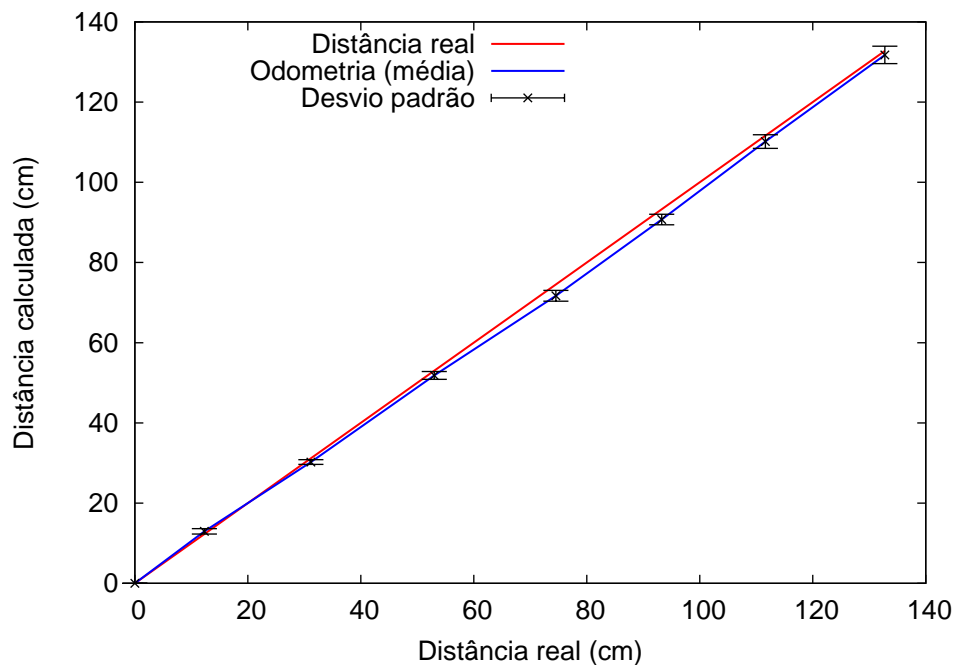


Figura 4.9: Resultados de medidas experimentais de odometria. O eixo X é a distância real percorrida e medida com uma fita métrica. O eixo Y é a distância calculada pelo dispositivo móvel usando o sistema proposto.

movimento do robô.

4.2 Sistema baseado em Microcontrolador

Embora o foco deste trabalho seja na proposta de um arquitetura que dispensa o uso de microcontroladores, essa possibilidade também foi incluída e validada no contexto de acionamento de dispositivos via canal de áudio. Lembramos que essa abordagem já foi proposta em outros trabalhos [Robinson et al. 2010, Rob 2012b], que são discutidos no Capítulo 3. Contudo, esses trabalhos somente funcionam com poucos modelos de telefones e necessitam de programas especializados para gerar sinais de áudio bastante específicos.

Nossa proposta consiste em usar tons de áudio do tipo DTMF para enviar comandos para um ou vários microcontroladores conectados em um dispositivo de controle via áudio. Como os microcontroladores possuem baixo custo e muitas opções de conexão de sensores e atuadores, é possível construir um sistema com poucos componentes e alta flexibilidade. Em um AtMega, por exemplo, o microcontrolador usado no Arduino, é possível ligar oito servo motores e controlar precisamente e com suavidade tanto a velocidade quanto a posição desses servos. Assim, pode-se delegar tarefas de baixo nível para o microcontrolador, como geração de PWM, enquanto o software de alto nível toma decisões e envia comandos via DTMF para o microcontrolador. Além disso, o microcon-

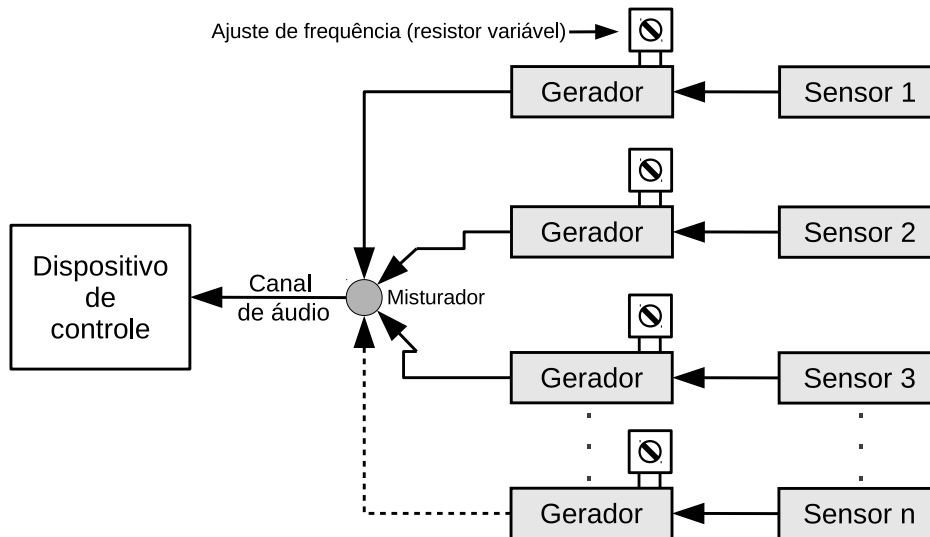


Figura 4.10: Sistema para captura do estado de diversos sensores usando um canal de áudio.

trolador também pode gerar tons de áudio para enviar informações sobre sensores para a unidade de controle, bem como respostas aos comandos recebidos via DTMF.

A Figura 4.13 mostra a visão geral do sistema. Um dispositivo móvel envia tons DTMF via canal de áudio, que é conectado a uma entrada analógica do microcontrolador. O microcontrolador, por sua vez, executa um algoritmo para identificação de tons DTMF, e com base nos tons recebidos executa alguma ação, tal como acionar algum atuador. Alguns testes foram realizados em um microcontrolador AtMega328, e por questões de desempenho, optou-se pelo uso do algoritmo de Goertzel (1985) ao invés da FFT. A vantagem deste, é que ele é mais eficiente que a FFT pelo fato de analisar apenas frequências específicas, enquanto a FFT analisa toda uma faixa de frequências. Assim, no caso de um microcontrolador, como o AtMega, que roda a 16MHz, algoritmos otimizados são importantes. Para gerar tons de áudio que codificam respostas a comandos e estado de sensores, uma saída do tipo PWM é utilizada, para gerar um sinal que se aproxima de um tom de áudio.

Uma outra vantagem do sistema baseado em microcontrolador, e que nas abordagens já apresentadas neste capítulo, sem o uso de microcontrolador, a unidade de processamento precisa gastar tempo de CPU executando uma tarefa de baixo nível, que é a geração de sinal PWM. Com o microcontrolador, a unidade de controle precisa apenas emitir os comandos de controle, enquanto o microcontrolador gera os sinais PWM de forma independente.

Também é possível expandir facilmente o sistema construindo uma rede de microcontroladores conectados via áudio, mas para tanto um protocolo de comunicação deve ser estabelecido. A Figura 4.14 mostra a arquitetura dessa rede, que pode ser composta de diferentes tipos e modelos de microcontroladores, sendo que cada um deve possuir uma implementação funcional de um decodificador de DTMF via software. Para a comunica-

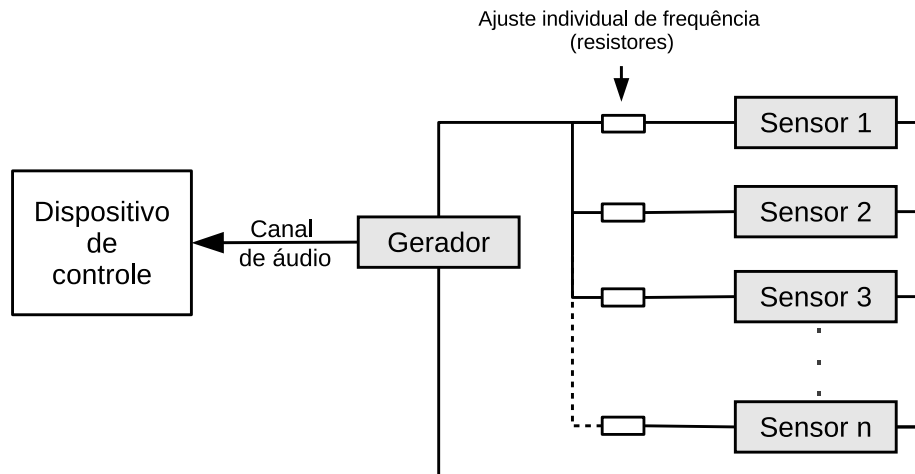


Figura 4.11: Sistema para captura do estado de diversos sensores usando um canal de áudio com um único gerador de tons.

ção bidirecional, utiliza-se um protocolo do tipo comando/resposta, onde um determinado microcontrolador somente envia dados, neste caso tons de áudio, caso uma requisição específica tenha sido enviada para seu endereço na rede. Este sistema foi validado com dois microcontroladores operando em conjunto.

O protocolo de comunicação mostrado a seguir é composto por um dígito DTMF delimitador de início de comando, o *, seguido do endereço do microcontrolador que deve receber o comando, seguido do comando a ser enviado, de seus parâmetros, e de um delimitador de fim de comando especificado pelo dígito DTMF \#. Também pode-se utilizar um dígito específico para endereço de *broadcast*, como o B, por exemplo, que identifica um comando a ser executado por todos microcontroladores da rede. A estrutura do protocolo é a seguinte:

$$* < \text{endereço} (1) > < \text{comando} (1) > < \text{parâmetros} (3) > \#$$

A Tabela 4.4 mostra alguns exemplos de comandos baseados nesse protocolo. Nesse exemplo, o sistema é composto por dois microcontroladores conectados a dois servomotores. Um dos microcontroladores tem endereço 1 e o outro tem endereço 2, assim como os servos também têm endereços 1 e 2. Ainda no exemplo, o segundo microcontrolador possui um dispositivo emissor de som, um *buzzer*, ligado na porta 3.

Com esse protocolo, o dispositivo de controle simplesmente precisa gerar a sequência DTMF que caracteriza um comando. Assim, todos softwares desenvolvidos neste projeto, bem como diversos outros, podem ser usados para enviar comandos para os microcontroladores. Novamente, usando o ADWN ou o Anwide, nenhum software ou configuração especiais precisam ser feitas no computador de desenvolvimento. Para interpretar a resposta, a unidade de controle deve executar a FFT e analisar as frequências presentes no sinal de áudio proveniente do microcontrolador.

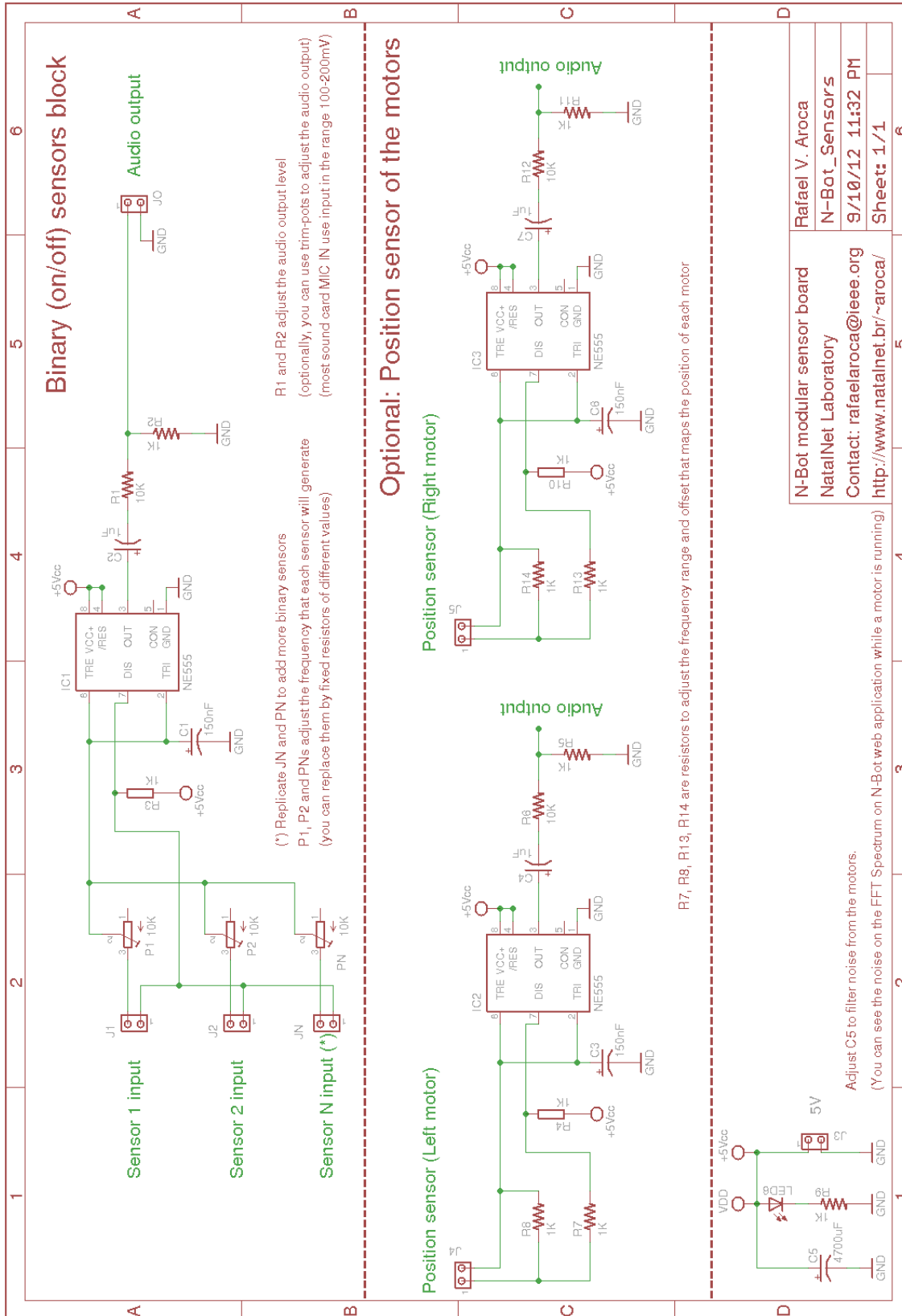


Figura 4.12: Diagrama elétrico do circuito para leitura do estado de sensores usando sinal de áudio.

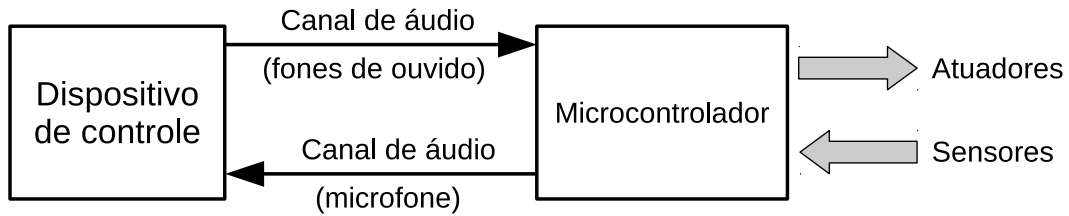


Figura 4.13: Envio de comandos para um microcontrolador via canal de áudio usando DTMF, e codificação de respostas e estados de sensores via tons de áudio únicos.

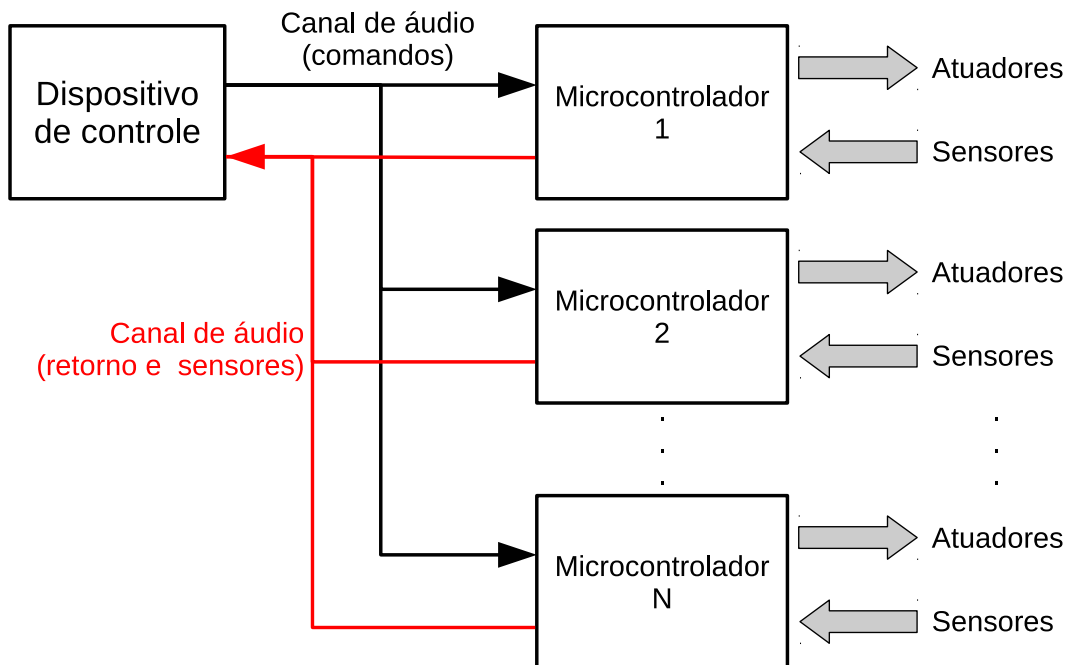


Figura 4.14: Envio de comandos para uma rede de microcontroladores via canal de áudio (linha preta) usando DTMF, e canal de retorno via áudio (linha vermelha).

Sequência de dígitos	Endereço	Comando	Parâmetro	Descrição
*11045#	1	1	045	Move para a posição 45 graus o servo 1 ligado ao microcontrolador 1
*12180#	1	2	180	Move para a posição 180 graus o servo 2 ligado ao microcontrolador 1
*21045#	2	1	045	Move para a posição 45 graus o servo 1 ligado ao microcontrolador 2
*23010#	2	3	010	Aciona um sinal sonoro por 10ms no microcontrolador 2

Tabela 4.4: Exemplo de comandos usando o protocolo de comunicação com DTMF entre dispositivo gerador de som e microcontrolador.

4.3 Estrutura do Robô

A estrutura física do N-Bot pode ser composta de diversos materiais, incluindo papelão, entretanto, como sugerido por Hamblen & Hall (2004), o uso de peças de brinquedos para construção de robôs educativos oferece um ótimo custo/benefício. Uma das vantagens observadas é a facilidade de obtenção desses brinquedos a baixo custo, e a maleabilidade de seus componentes, facilitando adaptações usando apenas ferramentas simples, como alicate e tesoura.

Da mesma forma que no caso das partes de brinquedos, a abordagem de robótica livre também pode ser usada através do reaproveitamento de cabos, motores, sensores e outras peças de computadores e impressoras antigas. Uma característica dessa abordagem é a necessidade de improvisação por parte dos alunos e a falta de padronização, já que cada robô é composto por peças que seu construtor obteve, dificultando a implementação de projetos onde os alunos constroem robôs padronizados e similares baseados em um guia ou manual.

Uma outra técnica interessante que utilizamos ao longo desta pesquisa é o uso de peças metálicas para montagem de móveis de madeira como peças para a construção da estrutura do robô. Essas peças são parafusos, fusos, porcas, e chapas metálicas para fixação no formato de “L”. Além de serem muito baratas, o uso dessas peças permite a construção padronizada de diversos robôs, já que todos alunos terão as mesmas peças à disposição, e um guia de montagem poderá ser baseado nestas peças.

A Figura 4.15 mostra um exemplo de configuração do N-Bot usando o chassi plástico e as rodas de carrinhos de brinquedo.

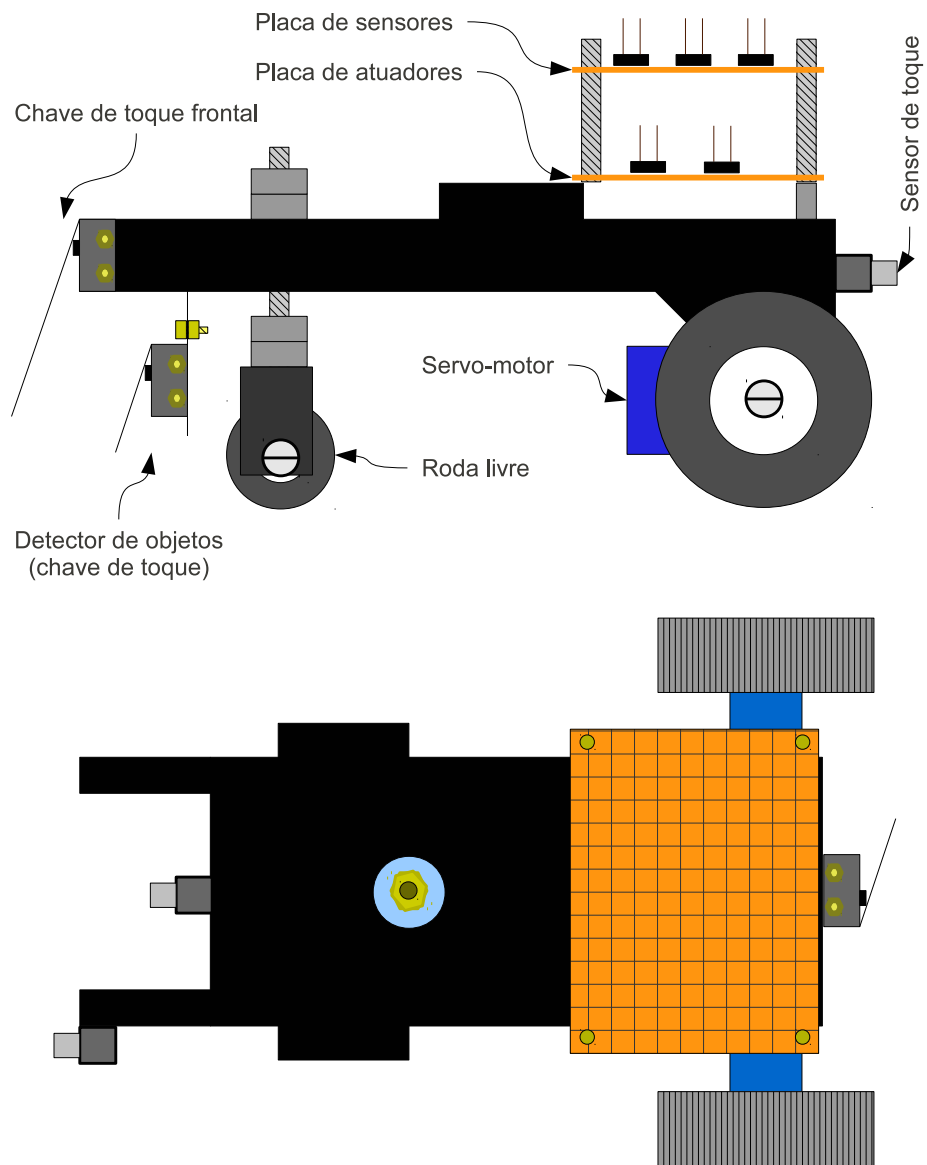


Figura 4.15: Exemplo de estrutura do N-Bot usando partes de brinquedos.

4.4 Software

Esta Seção descreve a arquitetura de software do N-Bot, e as soluções *open source* desenvolvidas tanto para dispositivos móveis quanto para computadores tradicionais. Nos dois casos, o foco é oferecer uma plataforma de desenvolvimento baseada na *web*, que dispense a instalação e configuração de qualquer programa, podendo também ser acessada por qualquer computador com um navegador web.

4.4.1 Ambiente de Programação via Web

Embarcado em um Dispositivo Móvel

O objetivo do sistema embarcado em um dispositivo móvel (telefone ou *tablet*), é permitir que esse dispositivo seja instalado fisicamente no robô para atuar como sua unidade de controle. Os programas de controle rodam nesse dispositivo, mas a sua programação é feita remotamente via web. A Figura 4.16 mostra arquitetura desse sistema.

Um dos principais componentes exibidos na Figura 4.16 é o servidor web. Ele é executado no dispositivo móvel e recebe conexões de qualquer dispositivo conectado na mesma rede. Qualquer dispositivo que disponha de um navegador web pode acessar e interagir com o sistema sem a necessidade de plugins especiais, como Java ou Flash, possibilitando assim o acesso de clientes com recursos mais restritos. A página web oferece opções para editar, salvar ou executar programas no robô, que podem usar tanto os sensores externos quanto os sensores do telefone. Notamos que além da interface com o robô usando áudio, esse ambiente também pode ser usado para programar robôs que usam outras interfaces de comunicação, como o Lego Mindstorms, que se conecta ao telefone via *bluetooth*.

Para tornar a implementação possível, foi utilizada a camada de scripts para Android (SL4A) [Jordan & Greyling 2011, Ferrill 2011], que permite a execução de diversas linguagens de script em dispositivos com o sistema Android. Além disso, também oferece abstração para os recursos do telefone e do sistema Android. Atualmente, as linguagens suportadas são o Python, Perl, Ruby, Lua, BeanShell, JavaScript, Tcl e Scripts Shell, mas o SL4A oferece a possibilidade de adicionar outras linguagens, assim o robô pode ser programado em qualquer uma dessas linguagens, ou mesmo em novas linguagens que venham a ser adicionadas ao SL4A.

Os programas editados ou criados pelo usuário são salvos na memória interna do dispositivo móvel, preferencialmente um cartão de memória do tipo SD. Ao clicar no botão de execução, alguns scripts preparam e iniciam a execução do programa do usuário. A entrada e saída padrão de todos programas são automaticamente redirecionadas para arquivos de *log* que podem ser inspecionados posteriormente. Além disso, a saída padrão também é exibida e atualizada automaticamente no navegador web do usuário, permitindo acompanhar e diagnosticar a execução do programa.

Devido à flexibilidade da plataforma Android, tarefas complicadas podem ser implementadas com pouco esforço de programação. A Listagem 1 mostra um exemplo de leitura de sensores do telefone, especificamente da bússola. Após obter a leitura do sensor, o sistema usa o sintetizador de voz para falar uma frase com os dados do sensor.

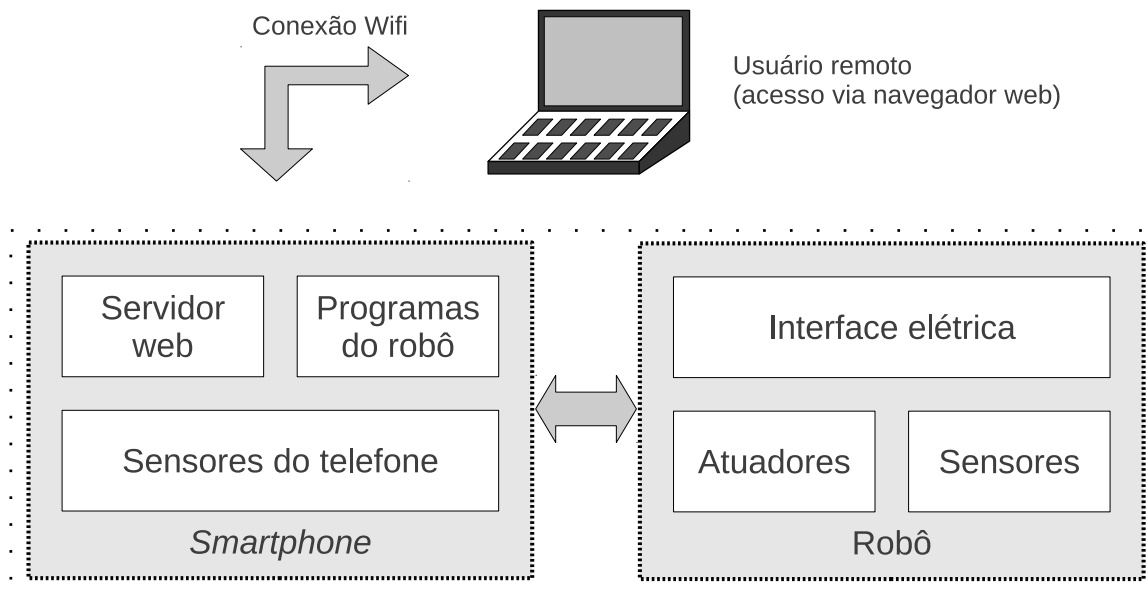


Figura 4.16: Arquitetura do sistema que possibilita programação remota.

Notamos que esse trecho de código refere-se ao programa completo, e não um fragmento de código.

Listagem 1: Realiza a leitura da bússola do telefone e utiliza o sintetizador de voz para falar esta leitura.

```
import android, time
droid = android.Android()

droid.startSensing()
time.sleep(1)
results = droid.readSensors().result
R = results['zMag'];

droid.ttsSpeak("My heading is " + str(R) + " degrees")
```

Na Listagem 2, a câmera de um telefone é usada para reconhecer etiquetas de código de barras, que poderiam ser usadas como marcadores de posições conhecidas para localização de robôs, como já sugerido em outro trabalho [Sousa et al. 2009].

Listagem 2: Reconhece uma etiqueta de código de barras com a câmera e fala “I’m home” se o código de barras for 7891219026195.

```
import android, time
droid = android.Android()

code = droid.scanBarcode().result

if code['extras']['SCAN_RESULT'] == "7891219026195":
    droid.ttsSpeak("I'm home")
else:
    droid.ttsSpeak("Unknow landmark")
```

A Listagem 3 aguarda um comando de voz e então move um robô baseado em Lego *Mindstorms* NXT caso o comando de voz seja “Mover”. A Listagem 4 tem o mesmo funcionamento apresentado na Listagem 3, mas ao invés de controlar um robô Lego via *bluetooth*, controla o N-Bot via áudio.

Listagem 3: Conecta a um robô Lego e executa um movimento quando o usuário falar “*Move*”, e por fim fala a distância medida com o sensor de ultra-som do Lego.

```
import android, time
import nxt.locator
from nxt.motor import *

droid = android.Android()

def move(s)
    m_left = Motor(s, PORT_B)
    m_left.update(100, 3600)
    m_right = Motor(s, PORT_C)
    m_right.update(100, 3600)

sock = nxt.locator.find_one_brick()

if sock:
    voiceCmd = droid.recognizeSpeech().result;

    if voiceCmd == "move":
        move(sock.connect())
    else:
        droid.ttsSpeak("Command not understood")
else:
    print 'Lego NXT unavailable'

D = UltrasonicSensor(sock, PORT_4).get_sample()
droid.ttsSpeak("Ultrasound distance is " + str(D))

sock.close()
```

Listagem 4: Gera tons de áudio DTMF para o N-Bot se mover quando o usuário falar “*Move*”.

```
import android, time

droid = android.Android()

voiceCmd = droid.recognizeSpeech().result;

if voiceCmd == "move":
    droid.generateDtmfTones("5", 500)
else:
    droid.ttsSpeak("Command not understood")
```

Também é possível utilizar facilmente a câmera, ou outros recursos do telefone. Para obter sua localização geográfica, por exemplo, basta utilizar o GPS através da função `droid.getLastKnownLocation()`.

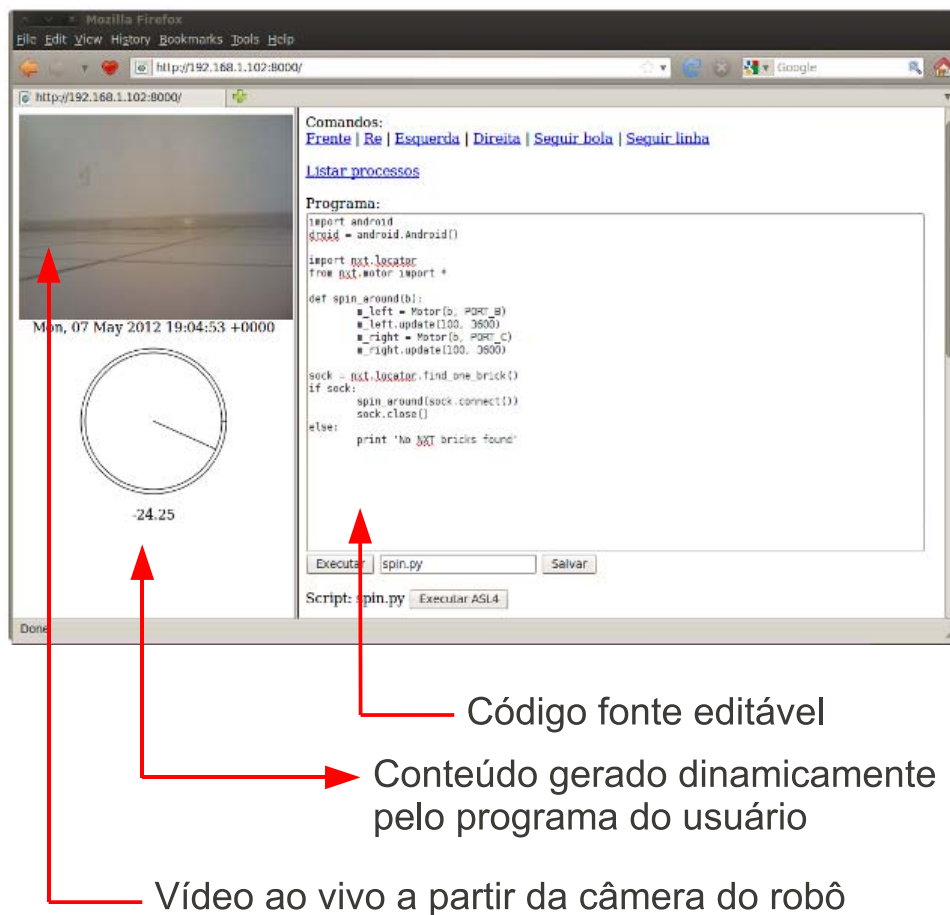


Figura 4.17: Captura de tela de um navegador web acessando o ambiente de programação via web.

A Figura 4.17 mostra um exemplo desse sistema em funcionamento, incluindo a funcionalidade de visualizar ao vivo, via web, a cena vista pela câmera do dispositivo móvel instalado no robô. Abaixo da cena da câmera, também é possível ver um conteúdo gráfico gerado pelo programa de usuário, que, nesse caso, é um ponteiro representando a direção de movimento detectada pela bússola.

Após a implementação dessa versão do sistema, decidimos desenvolver uma versão melhorada e mais acessível para usuários leigos através de programação por blocos ou textual. O novo ambiente de programação foi batizado como *Anwide: Android web based integrated development environment*, e disponibilizado na Internet como software livre, de forma que tanto o software quanto o seu código fonte (*open source*), podem ser baixados livremente a partir do site *web* `http://code.google.com/p/anwide/`.

Após ser instalado e executado no telefone, o Anwide mostra o IP (endereço do dispositivo na rede) do celular onde ele está instalado. Para possuir esse endereço, o celular deve estar conectado na Internet, em uma rede local ou operando no modo roteador (AndroidAP). Em seguida, o usuário deve usar um navegador *web*, em qualquer computador conectado à mesma rede, ou no próprio telefone e acessar o IP exibido na tela. Ao acessar

o sistema, o usuário terá acesso ao aplicativo *Web* mostrado na Figura 4.18. Lembramos que o Anwide é executado completamente no telefone do usuário, dispensando conexão com Internet ou o uso de qualquer computador auxiliar.

Como pode ser visto na Figura 4.18, a área do lado direito permite construir um programa apenas encaixando blocos. Quando o programa está pronto, o usuário clica em um botão que transforma os blocos em um programa na linguagem Python, e o executa no dispositivo móvel através do SL4A. Embora nossos testes tenham sido feitos em Python, outras linguagens são suportadas. Se o usuário desejar, ele pode editar, também via *Web*, diretamente o código fonte Python e executá-lo, permitindo alto grau de personalização e a possibilidade de programação avançada. Além do SL4A, o Anwide também utiliza porções de código do Blockly, que é uma linguagem de programação por blocos desenvolvida por engenheiros do Google. Outro projeto que utiliza o Blockly para programação gráfica do celular é *MIT App Inventor* e o *Scratch* [Resnick 2007], também do MIT.

Optou-se pela inclusão do ambiente de programação usando blocos baseado no *Blockly* já que a maioria dos sistemas para robótica educativa possuem a opção de programação por blocos. Além disso, a forma de conectar os blocos do *Blockly*, utilizado em nosso projeto, é similar àquela utilizada pelo *Scratch* do MIT, que não requer conhecimentos prévios de programação, sendo adequada inclusive para crianças. A programação é feita usando blocos com formas específicas, de forma que apenas blocos compatíveis e complementares podem ser encaixados.

A Figura 4.19 mostra o esquema geral de funcionamento do Anwide. Primeiro o usuário acessa a página web através de um navegador, e digita seu programa em linguagem Python (1), por exemplo. O usuário também pode desenhar seu programa usando a programação por blocos que serão automaticamente convertidos para um programa em linguagem Python (1). A página web carregada no navegador do cliente (1) comunica-se com o servidor web embarcado no dispositivo móvel (2) via protocolo HTTP, e envia o programa para ser armazenado (3) na memória do dispositivo. Em seguida o programa (*script*) em Python é executado usando o SL4A (4). Antes da execução, nosso sistema manipula os descritores de arquivo do programa do usuário para que a saída padrão seja gravada em um arquivo de log (5), que é continuamente monitorado pelo servidor web (6) de forma que o usuário possa visualizar em tempo real (7) a saída padrão e erros da execução de seu programa. Este retorno ocorre através do sistema *Asynchronous Javascript and XML* (AJAX), que permite a atualização constante de dados na página web sem a necessidade de recarregá-la.

Embora totalmente funcional, o Anwide é uma prova de conceito. Apesar de o termos desenvolvido para a plataforma Android, ele pode ser portado para utilização em outras plataformas de dispositivos móveis. O Anwide foi testado em diversos dispositivos móveis com Android, mas fizemos os testes principais no telefone HTC G1, um dos primeiros telefones com Android, com processador ARM de 528MHz, 192MB de memória e o sistema Android 2.1. Como os dispositivos mais novos são mais poderosos, fizemos esforços para o sistema executar nesse celular, para assim poder funcionar facilmente nos outros dispositivos.

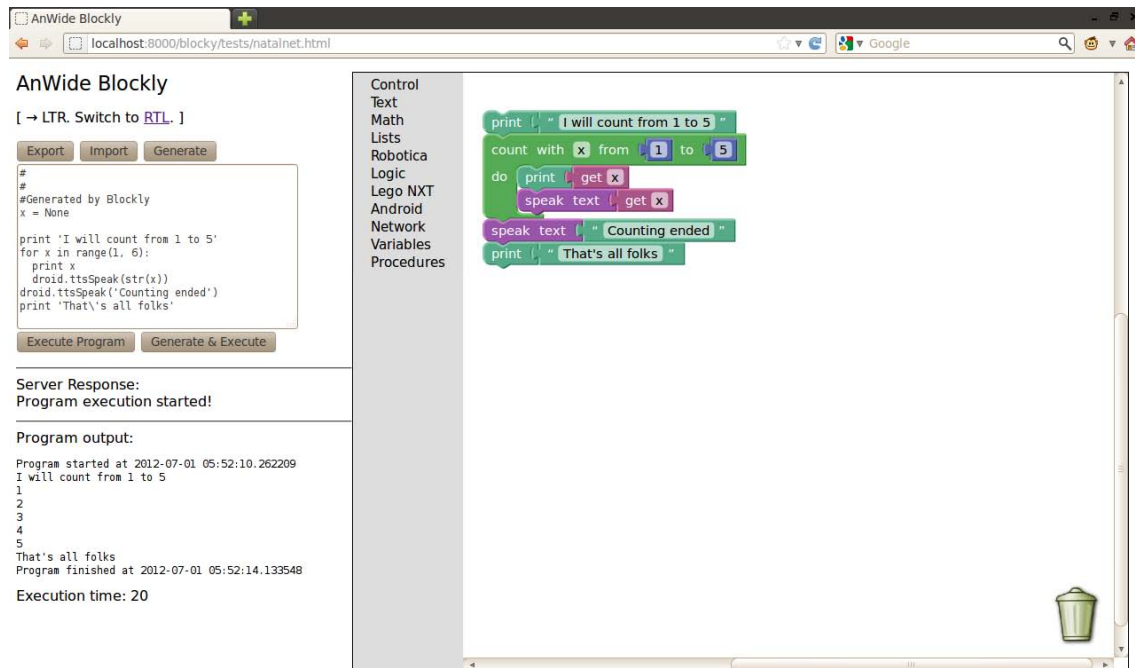


Figura 4.18: Captura da tela do sistema Web embarcado no celular, o Anwide. No lado direito o programa pode ser feito conectando os blocos. No lado esquerdo, o programa é traduzido para código fonte Python, e na parte inferior esquerda o resultado da execução do programa é mostrada.

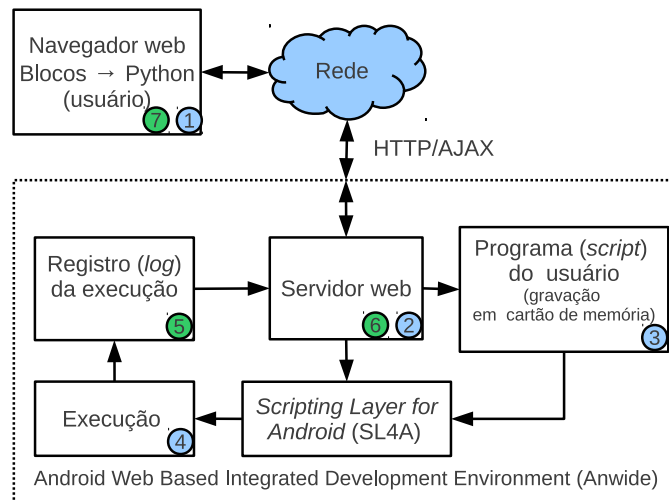


Figura 4.19: Esquema interno de funcionamento do Anwide e a interação de seus subsistemas. Os passos de execução, em ordem, são mostrados em círculos azuis, enquanto os círculos verdes mostram o fluxo para retorno de informações.

Programação Usando um Computador

Durante as demonstrações e experimentos com alunos para validar o N-Bot, notou-se que embora o uso de um telefone celular como unidade de controle ofereça uma solução poderosa e flexível, nem todos os alunos dispunham de celulares com capacidade de execução do Anwide. Para solucionar esse problema, optamos pela construção de um ambiente de desenvolvimento para o N-Bot baseado na web, o ADWN, sem o uso de um telefone, com a intenção de oferecer uma plataforma simples e intuitiva para iniciantes, porém poderosa e flexível para usuários experientes.

Além disso, para permitir o uso em computadores públicos, de bibliotecas e telecentros, o projeto do ambiente de programação exige que ele dispense a necessidade de instalação de qualquer software no computador de desenvolvimento.

A solução encontrada foi o desenvolvimento de uma aplicação web. Assim, para programar o N-Bot, o usuário somente precisa acessar uma página web que oferece a opção de programação em blocos, textual ou híbrida. Outra característica importante desse ambiente de programação, é que trata-se de um sistema web que não depende de servidor: como toda implementação usa JavaScript e tecnologias que são executadas no navegador, é possível fazer o download da página web de programação, e usar o sistema sem a necessidade de nenhum tipo de conexão com a Internet ou rede.

Por exemplo, um estudante pode fazer o download do ambiente de programação e gravar em um CD-ROM ou em um *pen drive*, e utilizá-lo em qualquer computador que tenha um navegador web, mesmo que desconectado da Internet. Como não é necessária a instalação de softwares, o sistema pode ser usado até em computadores com políticas restritivas de uso.

Para criar programas, basta ligar os blocos de uma palheta disponível em um menu lateral. Ao clicar no botão executar, os blocos são traduzidos para a linguagem JavaScript, o código fonte é exibido para o usuário, e o programa executado. O código gerado pode ser editado manualmente e executado. A execução ocorre no próprio navegador web, que possui um interpretador JavaScript embutido. Para interagir com o robô, esta página web gera tons de áudio, que são decodificados pelos circuitos do robô. Da mesma forma, para ler sensores, a página web executa uma transformada de Fourier para capturar o áudio da entrada de microfone do computador e decodificar o estado de cada sensor. Como o canal de comunicação é baseado em áudio, o sistema pode não funcionar se o nível de volume do computador estiver muito baixo.

A intenção inicial era desenvolver o sistema utilizando apenas HTML5, tornando o sistema mais portátil para ser executado em diversos dispositivos, mas infelizmente, o HTML5 ainda não suporta capturar dados do microfone do computador. Assim, para desenvolver a versão atual do sistema foi necessário desenvolver um programa em linguagem Flash (*Action Script*) que fica embarcado na página web do sistema de desenvolvimento. Esse programa captura o áudio do microfone do sistema, e executa a FFT para obter suas frequências. Ele também gera os tons DTMF e comunica-se com o programa principal, na página web via interface de chamadas entre objetos HTML do JavaScript.

A Figura 4.20 mostra o esquema interno de funcionamento do ADWN. Primeiro o usuário acessa a página web do ADWN (1) e cria seu programa conectando blocos (2), que em seguida são traduzidos para um programa em JavaScript (3). Se desejar, o usuário

pode pular o passo (2) e digitar diretamente o programa em JavaScript (3). Em seguida, o programa em JavaScript é executado (4), e durante a execução, a porção de código do usuário troca dados com um módulo em Flash (ActionScript) que gera tons DTMF e realiza a FFT no sinal de entrada do microfone do computador. Durante a execução do programa, o ambiente também monitora sensores e a saída do programa do usuário, exibindo, para o usuário, e atualizando continuamente estas informações.

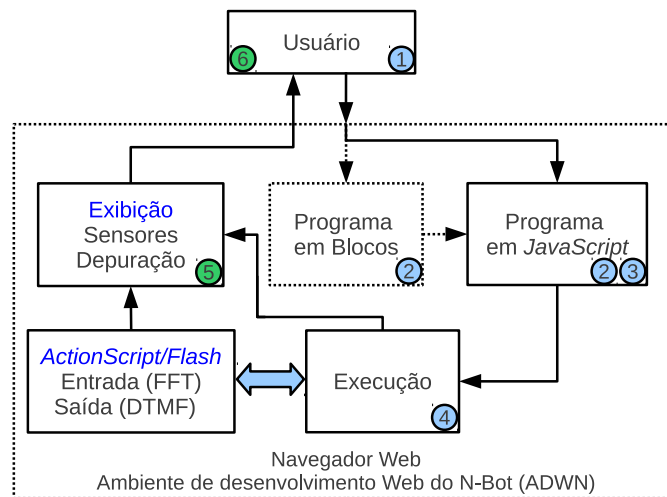


Figura 4.20: Esquema interno de funcionamento do ADWN e interação de seus subsistemas. Os círculos verdes representam fluxo de dados que exibem informações para o usuário.

Outro recurso interessante que implementamos no sistema é o suporte à programação multi-tarefa, possibilitando ao usuário criar vários grupos de blocos que executam de forma simultânea e independente. Para facilitar tarefas de depuração, também criamos uma área dinâmica que mostra o resultado da saída do programa quando o código executa uma chamada para uma função de impressão. Essa função pode ser usada com a chamada a `debug()` ou através do bloco `print2`.

A Figura 4.21 mostra uma captura de tela do sistema de programação web em funcionamento. Note que ele é similar ao Anwide, utilizado no telefone celular. No lado direito da tela, é possível implementar o programa usando blocos. Na parte esquerda superior é possível ver e editar o código do programa em forma textual. Abaixo do código, pode-se ver as mensagens impressas pelo programa seguido de um analisador de espectro, que mostra em tempo real as componentes de frequência encontradas no sinal de áudio de entrada.

O ambiente de programação também tem código aberto e está disponível *on line* no endereço <http://www.natalnet.br/~aroca/N-Bot/html/beta/natalnet/>. Ao acessar o sistema, um programa de exemplo irá ser carregado automaticamente. A página oficial do N-Bot [Aroca, Pitta, Burlamaqui & Gonçalves 2012a] possui informações adicionais e vídeos do sistema em funcionamento.

Para controlar atuadores do N-Bot, o sistema gera tons DTMF. A função textual para

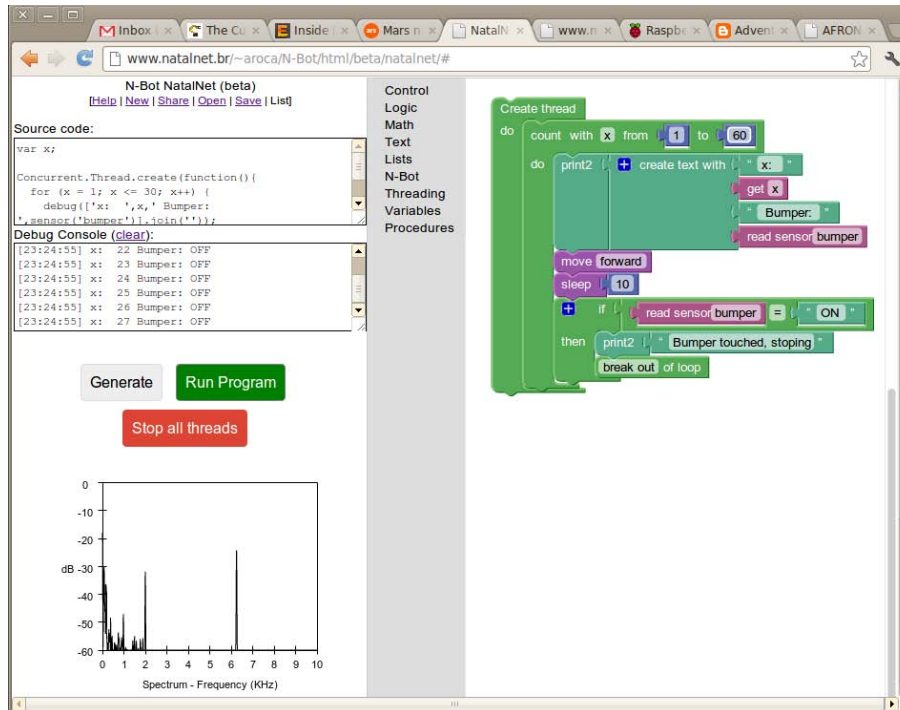


Figura 4.21: Captura de tela do ambiente de programação via web rodando em navegador web de um computador tradicional.

gerar tons DTMF é $DTMF_R(d, r)$, onde d é o dígito DTMF a ser gerado e r o número de vezes a repetir esse tom. Para programação em blocos, basta usar o bloco *Play DTMF*, que será mapeado para a função já mencionada. A Figura 4.22 mostra o bloco usado para gerar tons DTMF. Também é possível usar um bloco com maior nível de abstração, do tipo *Mover para frente*, que é mapeado para a função $DTMF_R(d, r)$.



Figura 4.22: Bloco usado para gerar um tom DTMF.

Para obter o estado de sensores, o sistema oferece uma função (ou bloco) para configurar a frequência associada a cada sensor. Um exemplo de uso desta função é `setFrequency(7, 1, 500, -10, 2)`; Os parâmetros desse exemplo são detalhados na Tabela 4.5.

Ao invés de usar a função textual `setFrequency()`, é possível usar o bloco correspondente a essa função, como mostrado na Figura 4.23.

Na implementação atual, a taxa de amostragem do sinal é de 22KHz e o número de pontos da FFT é 2048. Usando a Equação 2.2 sabe-se que esses parâmetros resultam em uma resolução de análise do sinal de 10Hz. Usando um espaçamento de 10Hz entre o sinal de cada sensor, esse sistema permitiria, teoricamente, o uso de até 500 sensores conectados simultaneamente via canal de áudio. Na prática, é claro, esse valor não pode

Parâmetro	Exemplo	Função
1	7	Número do sensor
2	1	Liga (1) ou desliga (0) a leitura deste sensor
3	500	Identificação da frequência associada a este sensor, entre 0 e 935. Multiplique por 10.7 para determinar a frequência. Para escolher 3.6KHz, por exemplo, usar 345 neste parâmetro
4	-10	Limiar de amplitude do sinal de áudio para considerar que o sensor foi acionado
5	2	Duração do tom de áudio para considerar que o estado do sensor mudou (múltiplo de 50 milissegundos)

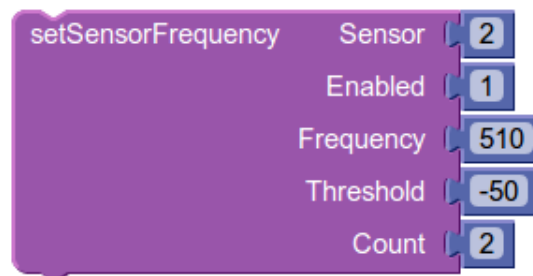
Tabela 4.5: Parâmetros usados na função `setFrequency`.

Figura 4.23: Bloco para configurar a frequência associada a um sensor.

ser obtido devido a fatores como interferência e distorção do sinal. Além disso, intervalos maiores podem ser usados para mapear uma faixa de frequências para o sinal de sensores analógicos.

A Figura 4.24 mostra exemplos do analisador de espectro em tempo real que exibe o sinal no domínio da frequência na tela do ambiente de desenvolvimento. A primeira imagem mostra o analisador de espectro após três execuções do comando `setFrequency`: cada uma configurou a amplitude e frequência de um determinado sensor, que fica representado na tela do analisador de espectro por um círculo sem preenchimento. A imagem do lado direito mostra um sinal associado a um sensor presente na entrada de áudio. Nessa situação, o círculo sem preenchimento fica de cor azul, representando que aquele sensor está acionado. Nesse momento, o sistema também notifica e atualiza o estado dos sensores para uso no programa do usuário. As frequências configuradas são 2KHz, 5.5KHz e 7KHz.

Para que os programas dos usuários possam realizar a leitura dos sensores em um programa em modo texto, o ambiente provê a função `sensor()`, que recebe como parâmetro o número do sensor, e retorna 0 ou 1 de acordo com o estado do sensor. A Figura 4.25 mostra o bloco equivalente para leitura de sensores.

O sistema foi testado nos navegadores Mozilla Firefox e Google Chrome e funcionam tanto no Windows quanto no Linux. O sistema ainda não foi ajustado para funcionar no

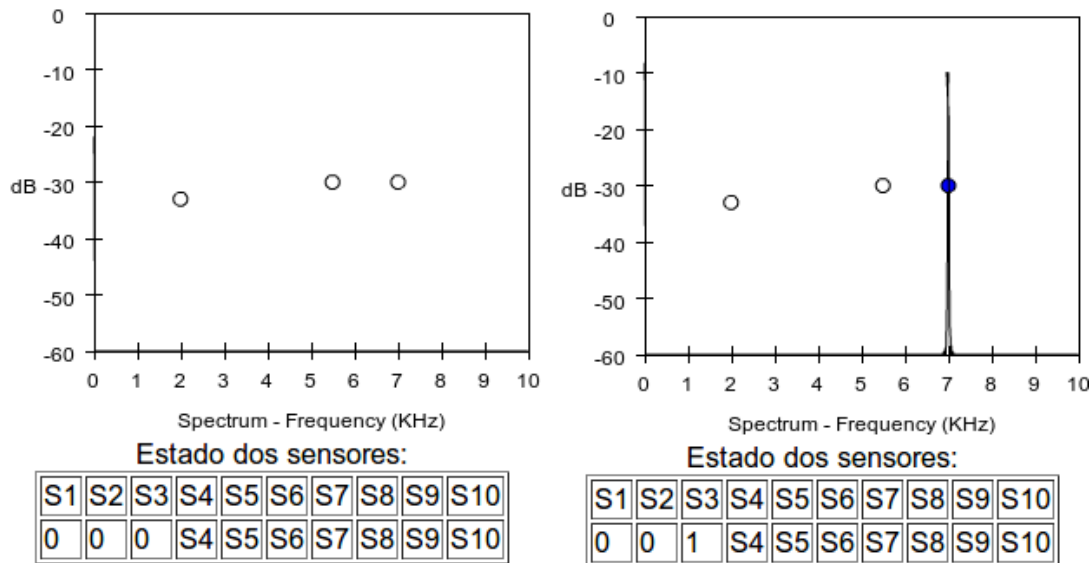


Figura 4.24: Vista do analisador de espectro no ambiente web. No lado esquerdo o analisador foi configurado para identificar o estado de três sensores, e no lado direito, o analisador detectou um sensor ativo.

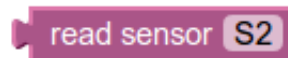


Figura 4.25: Bloco para leitura do estado de um sensor conectado via canal de áudio.

Internet Explorer da Microsoft.

Outro recurso interessante disponível no ambiente de desenvolvimento implementado é o recurso de salvar e carregar programas feitos com blocos em um arquivo XML. Um trabalho em desenvolvimento é um repositório *on-line* para troca de projetos e programas de forma interativa. Além disso, tanto esse ambiente baseado apenas na web quanto o Anwide possuem arquivos XML compatíveis entre si, possibilitando que um programa feito no Anwide possa ser usado no ADWN, e vice-versa.

Como o uso do ambiente web requer que o robô esteja conectado via cabo de áudio ao computador, sugerimos uma possível solução de baixo custo para dar mobilidade ao robô. Para tanto, deve-se alimentar o robô com baterias, ao invés de cabo USB, e utilizar um par de transmissor/receptor de rádio FM. Assim, o computador de controle irá transmitir os sons como uma estação de rádio FM, e o robô, usando um pequeno rádio portátil, irá receber os tons DTMF, decodificá-los e realizar os movimentos solicitados. Essa abordagem é viável apenas para envio de comando, já que para recepção de dados de sensores seria necessário um rádio de duas vias, como um *walk-talkie*.

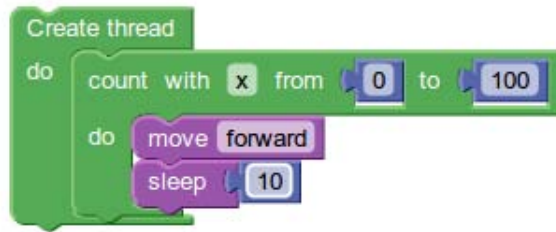


Figura 4.26: Exemplo de programa feito com blocos para o N-Bot se mover para frente por alguns segundos.

Exemplos de Programação com Blocos

Para ilustrar a implementação de programas por blocos no sistema proposto, esta Seção apresenta três exemplos de programas desenvolvidos no ambiente web do N-Bot.

A Figura 4.26 mostra um exemplo que faz com que o N-Bot mova-se para frente durante alguns segundos. A Figura 4.27 mostra um programa que configura um sensor e faz com o N-Bot movimente-se para frente até que o sensor de toque frontal (*bumper*) seja acionado, fazendo com o que o robô pare. A Figura 4.28 mostra um programa que utiliza três sensores: quando o robô tem uma colisão frontal, ele passa a andar para trás. Quando ele tem uma colisão traseira, ele passa a andar para frente, e quando um outro sensor de toque detecta um objeto, o robô para.

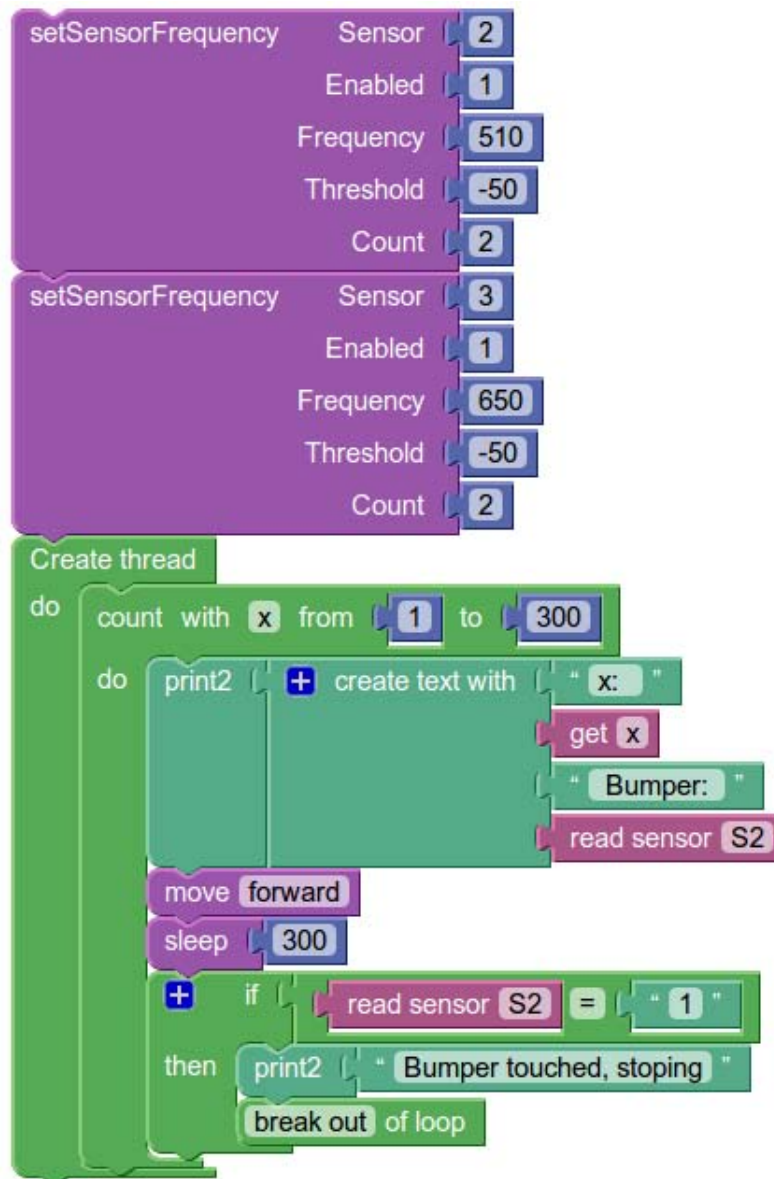


Figura 4.27: Exemplo de programa feito com blocos para o N-Bot andar para frente até o sensor de toque frontal ser acionado.

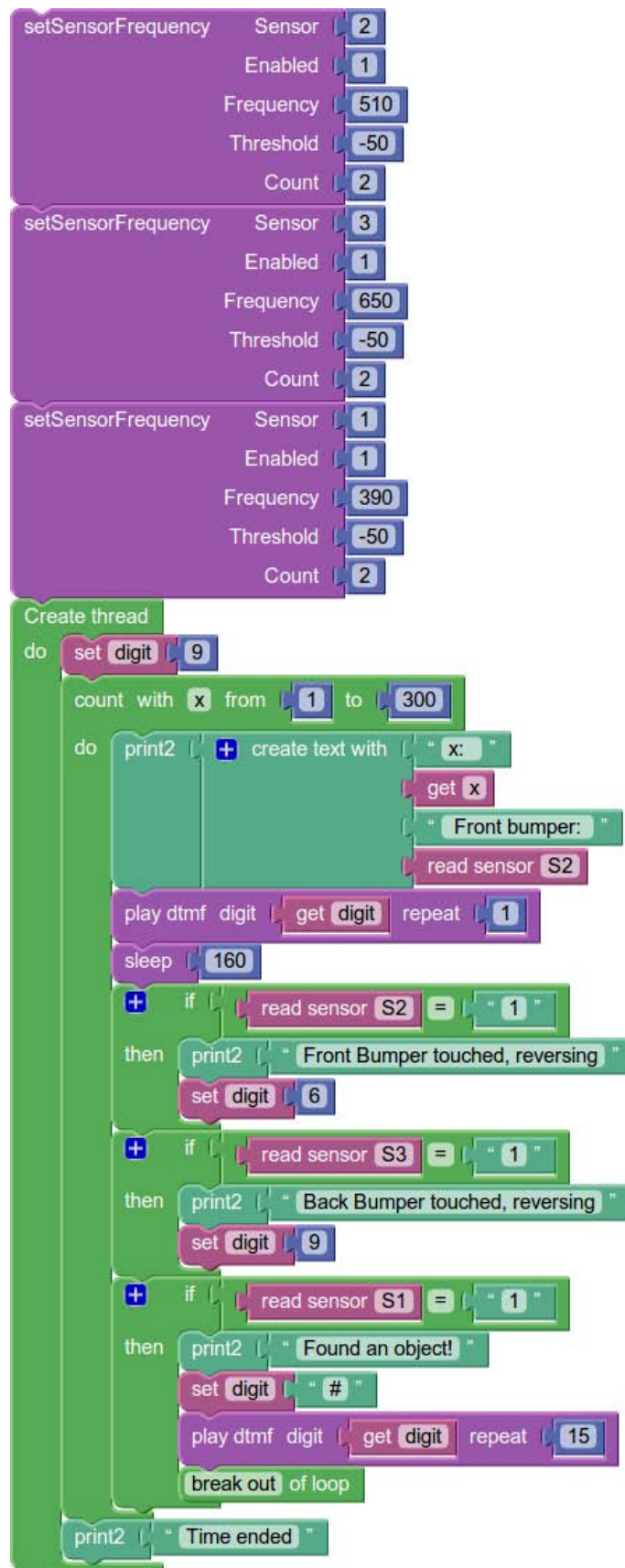


Figura 4.28: Exemplo de programa com três sensores de toque (ver texto para detalhes).

Capítulo 5

Resultados

As invenções são, sobretudo, o resultado de um trabalho teimoso.

Santos Dumont

Esquematizamos neste Capítulo a apresentação dos resultados obtidos neste trabalho em três seções. Primeiro, na Seção 5.1, são apresentados resultados gerais obtidos, tais como os robôs construídos e analisados, os custos e a repercussão do nosso projeto. A seguir, na Seção 5.2 são discutidos detalhes técnicos, como a performance do sistema e de processadores ARM, tipicamente usados em dispositivos móveis com Android. Finalmente, na seção 5.3, são apresentados resultados de uma pesquisa de interesse e opinião realizada com dezenas de estudantes durante o decorrer dos trabalhos.

5.1 Visão Geral

5.1.1 Robôs Construídos e Analisados

A Figura 5.1 mostra uma foto com detalhes do primeiro protótipo construído. Nesse modelo, o telefone executa o Anwide e atua como controlador do robô utilizando a interface de áudio para leitura de sensores e acionamento de atuadores. Também pode-se ver um apontador laser, que funciona em conjunto com a câmera do celular para medir a distância até obstáculos à frente do robô. Maiores detalhes e experimentos desse sistema podem ser encontrados em nosso trabalho disponível na literatura [Aroca, Burlamaqui & Gonçalves 2012].

A Figura 5.2 mostra um *tablet* e um *netbook* funcionando como controladores do robô. É interessante notar a portabilidade do sistema, já que o mesmo programa foi copiado do telefone móvel para esses dispositivos, e executado sem modificações. Tanto o *smartphone* da Figura 5.1 quanto o *tablet* utilizam processadores ARM, enquanto o *netbook* utiliza um processador x86.

A Figura 5.3 mostra dois robôs controlados por um telefone celular executando o Anwide, que permitiu acesso remoto via rede *wifi* à esses robôs, bem como programar e enviar comandos remotos para eles, mostrando a flexibilidade do Anwide para operar com diferentes tipos de robôs.

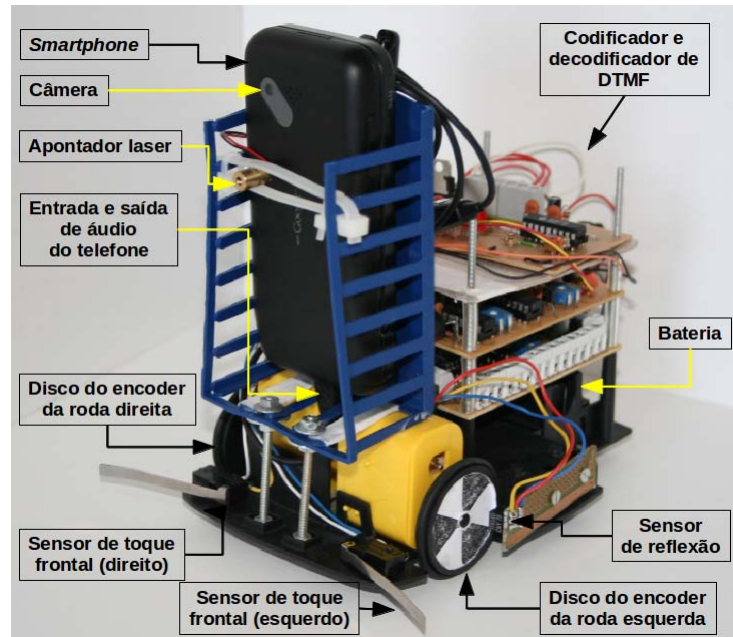


Figura 5.1: Robô projetado e construído para ser controlado via interface de áudio de um telefone celular.

A Figura 5.4 mostra a última versão do N-Bot, que foi construída com foco em redução de custos. Mais uma vez, o telefone móvel atua como o controlador desse robô, através da interface de áudio, sendo a programação feita remotamente pela interface web do Anwide. Para possuir autonomia enquanto controlado pelo celular, uma bateria é usada na parte inferior do N-Bot, dispensando o uso da alimentação via porta USB.

A Figura 5.5 mostra uma foto do N-Bot construído com microcontrolador e peças metálicas utilizadas na construção de móveis de madeira, destacando as chapas de fixação em “L”, parafusos e porcas.

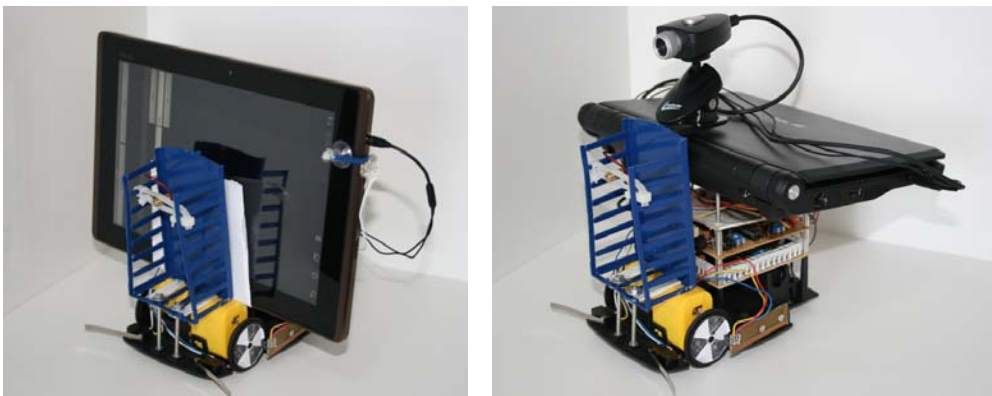


Figura 5.2: Tablet e netbook operando como controladores do robô via interface de áudio.



Figura 5.3: Dois robôs comerciais controlados pelo Anwide: um Lego Mindstorms e um Pioneer P3AT. Note o telefone fixado na parte superior dos dois robôs.

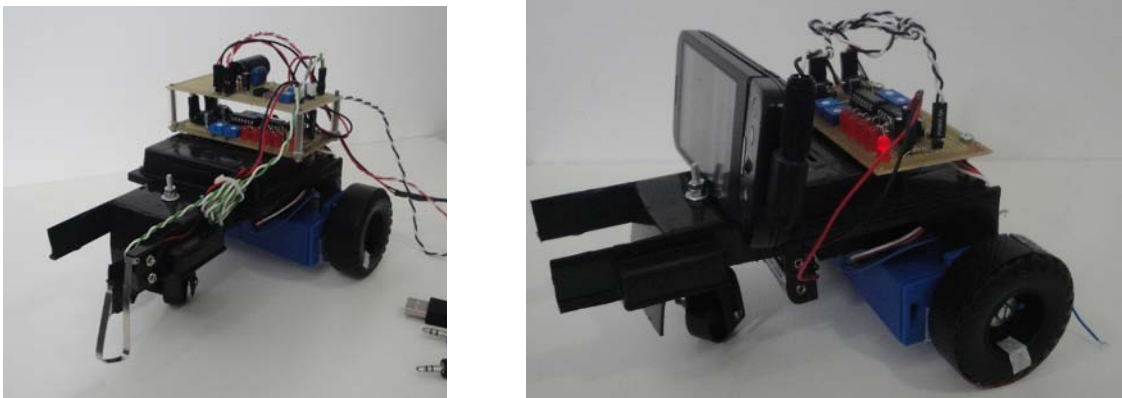


Figura 5.4: N-Bot com os cabos para conexão de áudio e alimentação e N-Bot controlado por um telefone rodando Anwide.

5.1.2 Custos do Sistema

A Tabela 5.1 mostra um resumo dos custos para construção do N-Bot considerando a compra das peças no Brasil com todos impostos inclusos. Esses valores não incluem a estrutura mecânica, já que um brinquedo antigo foi reaproveitado. Os custos apresentados incluem dois motores e dois sensores, e são relativos ao robô exibido na Figura 5.4. A Tabela 5.1 também mostra de forma comparativa o valor das peças na China (compradas em pequena quantidade).

A Tabela 5.2 mostra uma estimativa de economia(s) de escala se os componentes do N-Bot fossem adquiridos em grande quantidade para construção de kits educativos.

Como mostrado na Tabela 5.1, o N-Bot possui um baixo custo, entretanto é conveniente comparar seu custo com outras soluções. Como discutido no Capítulo 2, os robôs educativos são tipicamente construídos com microcontroladores, como o PIC, o MSP430 ou o AtMega, usado no Arduino. Esses sistemas são práticos e de baixo custo, porém limitados em capacidade computacional, além de necessitarem diversos sensores exter-

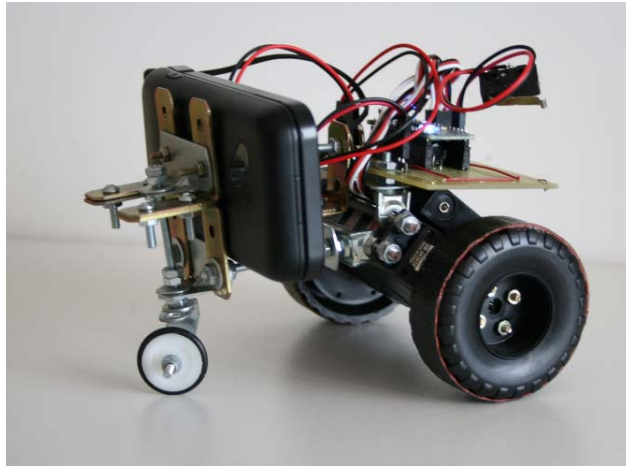


Figura 5.5: Versão do N-Bot construído com microcontrolador e peças metálicas usadas em móveis.

Descrição	Valor (R\$) (Brasil)	Valor (US\$) (China)
Circuitos integrados	8,00	1,5
Componentes diversos	4,00	2,9
Placa de circuito impresso (2)	8,00	0,82
Motores (2)	24,00	7,88
Sensores (2)	4,00	0,70
Parafusos e porcas	1,00	0,30
TOTAL	R\$ 49,00	US\$ 14,10

Tabela 5.1: Custos para aquisição das partes necessárias para construção do N-Bot no Brasil e na China. Valores referentes a Outubro de 2012.

nos, e de um nível de conhecimento intermediário de eletrônica e computação para serem programados. Ainda assim, caso se opte por construir um N-Bot com microcontrolador, conforme descrito na Seção 4.2, estimamos que seu custo será cerca de 10% inferior à versão com o decodificador de DTMF, já que o número de componentes é reduzido para apenas um microcontrolador, dois capacitores e quatro resistores.

Além disso, quando se deseja construir robôs que devem executar algoritmos mais sofisticados, como sistemas e visão computacional, normalmente utiliza-se um *notebook* ou *netbook* como unidade de controle. A Tabela 5.3 apresenta uma breve comparação de cada uma dessas abordagens. Tanto no caso do robô com PC quanto com microcontrolador, diversos sensores adicionais devem ser comprados, conectados, e esforço adicional de desenvolvimento deve ser empregado para esses sensores poderem ser utilizados.

Número de kits	Redução de custo	Valor (R\$) (Brasil)	Valor (US\$) (China)
1	0%	49,00	14,10
100	30%	34,30	9,87
1000	37%	30,87	8,88
10000	70%	14,70	4,23

Tabela 5.2: Estimativa de redução dos custos do N-Bot de acordo com a escala de compra.

Recurso	Netbook	Telefone (*)	Microcontrolador
Velocidade da CPU / Memória	1GHz/1GB	500MHz/512MB	40MHz/8KB
Duração da bateria	8-12h	10-20h	20h (\$25)
Câmera	VGA	2x6 MegaPixel	+\$100 + trabalho
Receptor GPS	+\$200	Embutido	+\$200 + trabalho
Acelerômetro	+\$70	Embutido	+\$70 + trabalho
Wifi / Bluetooth	Embutido	Embutido	+\$250 + trabalho
Reconhecimento de voz	Médio	Ótimo	+ trabalho
Bússola	+\$70	Embutido	+\$70 + trabalho
Microfone / alto falantes	Embutido	Embutido	+\$20 + trabalho
Visão computacional	OpenCV	OpenCV	+ trabalho
Linguagens de programação	Muitas	Muitas	Assembly, C
Controle de motores	+\$13	+\$13	Embutido
Tamanho	Médio	Pequeno	Pequeno
Preço (R\$)	700 (1053)	350 (363)	100 (835)

Tabela 5.3: Tabela comparativa de unidades de controle para robôs. O preço final é aquele da unidade de controle, e o valor entre parênteses é o preço da unidade de controle somado ao valor dos sensores. (*) Preço de um telefone celular com sistema operacional Android no mercado brasileiro em Jun/2012 (com os recursos mencionados).

5.1.3 Repercussão

Todas as instruções de montagem e funcionamento do N-Bot estão disponíveis no endereço *web* do laboratório NatalNet [Aroca, Pitta, Burlamaqui & Gonçalves 2012a]. Após a divulgação dos resultados do concurso *10-dollar robot design challenge*, o site recebeu mais de 6500 visualizações de páginas em cerca de 40 dias. A metade desses acessos é proveniente do Brasil, e o restante distribuído entre 83 países, sendo os primeiros da lista os EUA, o Canadá, a Índia e a Alemanha. Além disso, o N-Bot foi incluído em artigos de diversas revistas internacionais reconhecidas, como a *Wired* e a *Popular Mechanics*. O N-Bot também foi convidado, e enviado para exibição no museu de tecnologia da Universidade de Berkeley nos EUA.

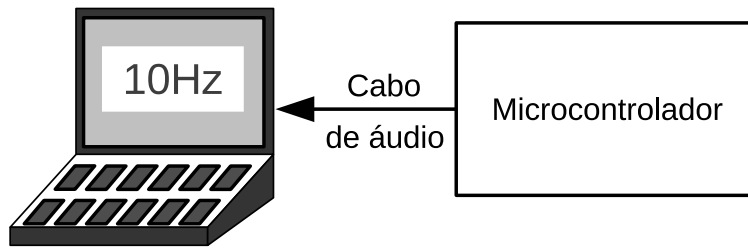


Figura 5.6: Sistema para medição da taxa máxima de leitura de sensores.

5.2 Aspectos Técnicos

Medidas de desempenho foram definidas e realizadas, visando verificação dos aspectos relacionados com a tecnologia resultante do projeto tais como taxas de leitura, execução em tempo real, entre outras.

5.2.1 Análise de Desempenho

Leitura dos Sensores

Foram realizados diversos experimentos para verificar a taxa máxima de leitura do estado dos sensores via canal de áudio no telefone celular com processador ARM. Com o objetivo de medir os piores tempos de execução, optou-se pela medição de desempenho de reconhecimento de dígitos DTMF por segundo, já que o reconhecimento de tons únicos é mais eficiente.

A figura 5.6 ilustra um sistema usado especificamente para testar a leitura de sensores. Um microcontrolador foi programado para gerar tons DTMF ou tons únicos de áudio em diferentes intervalos de tempo. Sua saída foi conectada aos dispositivos em teste, e os tons gerados com diferentes intervalos. Os testes foram realizados várias vezes até que se observou perdas de dados.

Pelos testes feitos, no pior caso, o tempo de computação da FFT levou 17ms, resultando em um limite teórico máximo de 58.8 FFTs por segundo. A Figura 5.7 mostra os resultados experimentais do sistema rodando em três dispositivos diferentes, que são: um telefone celular HTC G1 com processador ARM de 528MHz, um *tablet* com processador ARM de 1GHz e um *netbook* com processador x86 de 1GHz. Todos os testes foram executados na plataforma Android.

A Figura 5.7 mostra que mesmo o dispositivo com menor poder computacional pode manter uma taxa de reconhecimento de 40 dígitos DTMF por segundo sem perder nenhum dígito transmitido. Existem várias causas para a perda de pacotes apresentada no gráfico que se inicia em 40Hz. Uma das causas são as diferentes características do hardware de captura de áudio de cada dispositivo. Outra causa é relacionada com o escalonador de tarefas do sistema operacional Android (consequentemente do Linux), que pode não apresentar um bom determinismo quando a carga de processamento da CPU é elevada.

Como referência comparativa, fizemos alguns testes de desempenho usando um kit

Lego Mindstorms NXT. Quando conectado a um computador ou *smartphone* via *bluetooth*, a taxa máxima de leitura de sensores é de 20Hz, porém se vários sensores são lidos, esta taxa é dividida pelo número de sensores. Por exemplo, um software que faz a leitura de dois *encoders* e de dois sensores de toque obtém uma taxa de leitura aproximada de 5Hz por sensor. Já quando o *brick* do NXT está conectado a um computador usando cabo USB, a taxa máxima de leitura de sensores sobe para 166Hz. Dessa forma, se forem usados dois *encoders* e dois sensores de toque, então a taxa de leitura de cada sensor será de 41.5Hz, uma performance comparável ao do sistema proposto neste trabalho, que é de 40Hz para cada sensor.

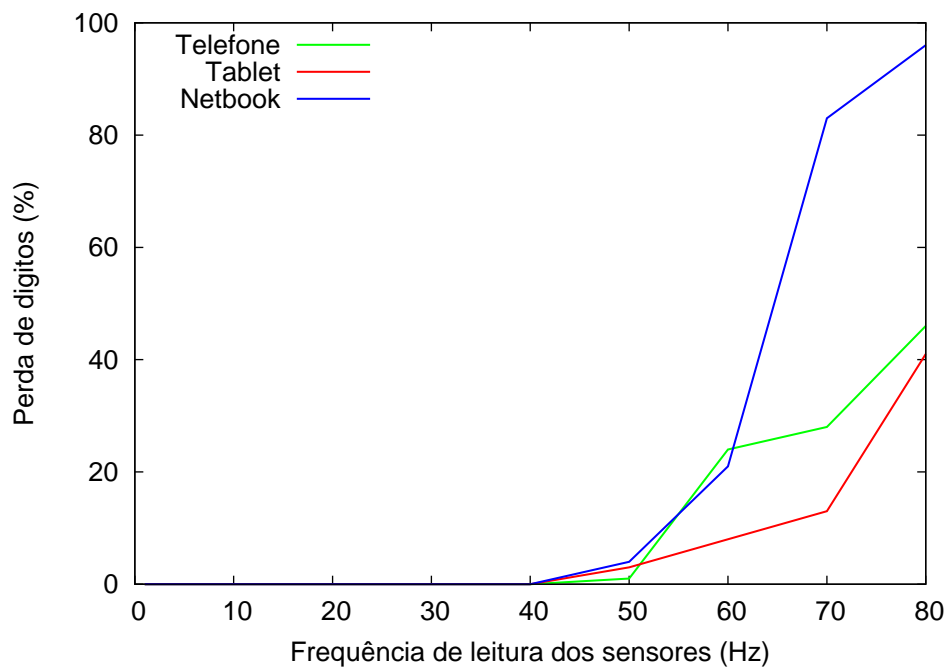


Figura 5.7: Desempenho do sistema de reconhecimento de dígitos DTMF em diversos dispositivos.

Como demonstrado, nos sistemas baseados em Android, é possível ler o estado de sensores a 40Hz, e assim fechar malhas de controle. Assim, um dos exemplos implementados permite ao N-Bot receber um comando de movimentação e realizar determinado deslocamento enquanto monitora seus sensores para desativar os motores quando a posição solicitada é atingida.

Para o ambiente de programação sem o uso do celular, onde as FFT são calculadas diretamente no navegador web do usuário, a performance obtida em um PC com sistema operacional Linux e processador de 1.6GHz, foi de cerca de 5ms, que oferece um desempenho bastante superior àquela obtida no sistema que é executado nos dispositivos móveis com Android, contudo não foi possível obter taxas de leitura de sensores superiores a 4.9Hz. Para identificar o motivo desta baixa performance, várias partes do sistema foram testadas de forma isolada e notou-se que existe uma limitação na interface de chamadas de função entre elementos JavaScript de um navegador web e elementos em

Flash/ActionScript. Enquanto o programa em JavaScript por si só pode executar até 40 mil instruções por segundo, o programa em Flash pode executar as FFTs de leitura de sensores em 5ms. Mesmo assim, pela limitação na interface entre estes dois elementos, a performance caiu para 4.9Hz. Esta limitação pode ser melhorada em trabalhos futuros através de implementações do sistema com novas tecnologias.

Temporização e Geração dos Tons DTMF

Para ajustar a velocidade de movimento dos motores é interessante que o sistema possa gerar um sinal de controle com uma determinada frequência. Isso pode ser feito utilizando laços que alternam entre a geração de tons DTMF e a espera por um intervalo de tempo. A função de espera (*delay*) está disponível e pode ser usada facilmente nos programas para dispositivos baseados em Android. Contudo, no ambiente web que dispensa o uso do telefone, foi necessário adotar uma abordagem diferenciada.

No ADWN, o programa de controle do robô é executado em JavaScript. Entretanto, essa linguagem possui uma filosofia de programação baseada no uso de eventos assíncronos, e não possui funções de espera. Ao tentar implementar uma função desse tipo, observamos que uma *espera ocupada* deixa todo o navegador bloqueado e sem resposta até que a espera ocupada acabe, resultando em uma solução indesejável pela sua natureza bloqueante.

A melhor solução encontrada foi adaptar e integrar um *framework* para execução de *threads* em JavaScript [Maki 2007, Maki & Iwasaki 2007] ao nosso sistema, resultando em um ambiente com possibilidades de programação paralela virtual onde blocos de código são executados de forma paralela, mas sem o conhecimento do escalonador de tarefas do sistema operacional nem o mapeamento para diversos processadores. Esse *framework* divide um código a ser executado em várias partes, e agenda a execução de cada uma dessas partes usando a função `setTimeout` da linguagem JavaScript. Também adicionamos uma função que interrompe a execução durante um intervalo de tempo solicitado. Para tanto, essa função também usa a chamada *setTimeout* para agendar a continuidade da execução do programa.

Com essa solução, diversos trechos de código podem ser implementados tanto de forma textual quanto em blocos, e esses serão executados em paralelo de acordo o sistema de agendamento. Para validar o funcionamento do sistema, alguns testes foram feitos em um computador com processador dual core AMD E-450 nos sistemas Linux e Windows. O ambiente foi testado nos navegadores Mozilla Firefox 16.0.1 e Google Chrome 22.0.1229.94.

A Figura 5.8 mostra um esquema do ambiente usado para realizar os testes no sistema. Um computador foi preparado para gerar sequências de dígitos DTMF com vários intervalos de tempo. Este sinal é conectado, via cabo de áudio, ao decodificador de DTMF em teste, e a saída deste decodificador é conectada a um analisador lógico que também é conectado ao computador. Dessa forma, o computador pode gerar sequências de dígitos DTMF em diferentes intervalos e monitorar quanto tempo leva para cada dígito ser decodificado, bem como a taxa máxima de decodificação obtida pelo conversor em testes.

Como mencionado, pretende-se gerar uma onda de forma quadrada que possibilite controlar a velocidade dos motores. Para tanto, é desejável que o sistema possua compor-

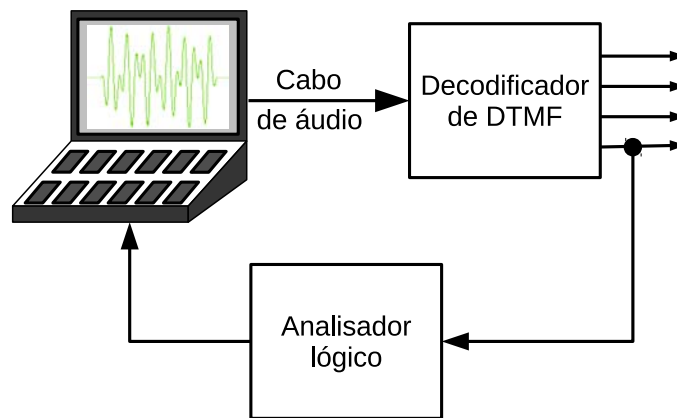


Figura 5.8: Ambiente de testes para análise desempenho da temporização para geração de tons DTMF.

tamento de tempo real, exibindo determinismo ao gerar o sinal. Para realizar os testes, um programa em blocos foi feito com o objetivo de gerar um sinal de áudio através de tons DTMF. Foram testadas diversas frequências, e foi observado que os dois navegadores, nos mesmos sistemas operacionais podem gerar sinais DTMF que alternam em até cerca de 6Hz.

A Figura 5.9 mostra os resultados dos testes em diversas situações. Assim, o programa em blocos foi ajustado para gerar frequências de 1, 2, 3, 4, 5 e 6Hz, e o osciloscópio analisou se o sinal gerado estava na frequência esperada. Tanto o Firefox quanto o Chrome foram analisados em uma situação normal, e com o sistema operacional configurado para dar alta prioridade (AP) a essas tarefas, o que, teoricamente, facilitaria o cumprimento de rotinas de temporização com maior precisão.

Também foram feitos outros experimentos, fora do ambiente web ADWN, e observou-se que todos os dispositivos testados podem gerar sinais de áudio de até 11KHz com grande estabilidade e precisão. O que reduz consideravelmente a frequência final nos pulsos obtidos na saída do circuito de controle de motores são a especificação de sinais do tipo DTMF e o conversor de DTMF MT8870. Como o DTMF foi projetado para oferecer robustez em linhas de transmissão ruidosas, sua especificação determina que um dígito DTMF válido deve ser composto do sinal DTMF de duração mínima de 45ms seguido por um silêncio *inter-dígito* de 40ms, totalizando 85ms [Durda 1995].

Mais uma vez notou-se que para o ADWN, a limitação na temporização para gerar tons DTMF está relacionada com a interface entre JavaScript e ActionScript, já que, assim como na FFT, a geração e captura de áudio é feita pelo módulo em ActionScript que se comunica com o módulo em JavaScript, onde existe um limite de velocidade entre estes dois subsistemas do ADWN.

Com 85ms por dígito, a máxima frequência que pode ser gerada é de 11.76Hz. Assim, o limite na implementação e experimentos exibidos na Figura 5.9 também é devido ao tempo de propagação do sinal no decodificador de DTMF. Entretanto, observamos que é possível utilizar tons DTMF com temporizações abaixo do padrão e obter sinais de até 50Hz na saída do decodificador de DTMF. Esse resultado foi obtido usando a ferramenta

dtmf dial [Nahshon 1998], configurada para gerar uma sequência DTMF com tempo de 10ms por dígito e de 10ms no intervalo *inter-dígito*. Para que o MT8870 suporte esses tempos, é necessário mudar seu circuito, substituindo o resistor de ajuste de $300K\Omega$ por um de $1K\Omega$.

A Figura 5.9 mostra que em todas situações analisadas, o desvio padrão da frequência gerada não foi maior que 1Hz. No geral, o pior caso observado teve um desvio de 7% entre a saída real e a frequência configurada no software.

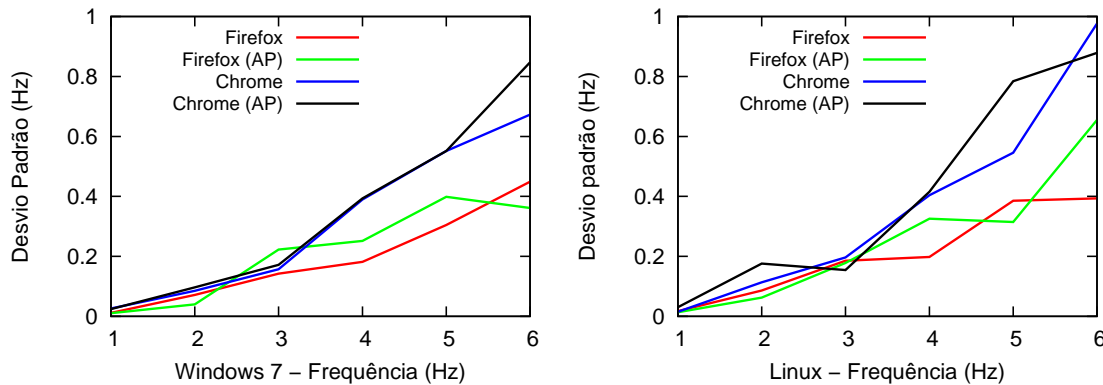


Figura 5.9: Desempenho dos navegadores Mozilla Firefox e Google Chrome para alternar a geração de tons DTMF tanto no Windows quanto no Linux (AP = Alta Prioridade).

Durante os testes, foram observados alguns fatos interessantes: primeiro, ao abrir novas abas no navegador, o sinal gerado torna-se irregular e sua frequência vai diminuindo até atingir 0Hz enquanto o usuário visualiza outra aba. Ao voltar para a aba do N-Bot, a frequência especificada volta ao normal. Esse comportamento foi observado tanto no Linux quanto no Windows em ambos os navegadores. Constatamos assim que enquanto se visualiza outras abas (páginas), scripts JavaScript em execução em qualquer outra aba são negligenciados pelo navegador.

Nota-se que o navegador continua executando o JavaScript de outras abas, mas sem determinismo ou previsibilidade. Assim, se durante a geração de um sinal de 7Hz, o usuário seleciona outra aba do navegador, imediatamente a frequência é reduzida, de forma indeterminada, para um valor entre 0Hz e 1Hz. Esse comportamento é aceitável no Firefox, onde todas as páginas, programas e scripts rodam no contexto de um único processo do sistema operacional. Já o Google Chrome mapeia cada aba aberta para um processo do sistema operacional, por isso, se esperava maior determinismo e confiabilidade usando o Chrome. Em todos os casos, contudo, uma nova janela do navegador não interfere na geração de frequência e performance do ambiente do N-Bot, permitindo que se navegue na Internet enquanto o N-Bot é controlado pelo mesmo navegador em uma janela dedicada. Finalmente, vale lembrar que esse comportamento repetiu-se em testes feitos com diferentes prioridades do sistema operacional, até a mais alta do sistema (AP).

Notamos que os resultados apresentados não comprometem o funcionamento do sistema, e melhorias podem ser obtidas com modificações de software, eletrônicas ou mecânicas.

Com relação ao sistema baseado em microcontrolador e descrito na Seção 4.2, alguns testes também foram realizados usando um microcontrolador AtMega328, e foi obtida uma taxa de envio de comandos de 10Hz, porém não foi feita nenhuma otimização na implementação. Acreditamos que essa taxa seja adequada, já que os comandos só precisam ser enviados para mudar estados de atuadores, como por exemplo: iniciar o movimento de um motor, mudar sua velocidade, ou parar. Além disso, o sistema tem a vantagem de conectar o microcontrolador ao dispositivo de controle com baixo custo e simplicidade, via áudio, dispensando etapas de configuração e pareamento, como no caso de conexões *bluetooth*, por exemplo.

Análise de Desempenho de Processadores ARM

Como um dos focos desse trabalho é o uso de dispositivos móveis que possuem processadores ARM para controle de robôs, optou-se pela execução de um estudo sistemático sobre o desempenho de processadores ARM em tarefas relacionadas ao Anwide e ao N-Bot, que são: a execução de um servidor web, e cálculos de ponto flutuante, já que a FFT requer um extenso número de multiplicações em ponto flutuante. Este trabalho, também é uma de nossas contribuições à comunidade e foi publicado em veículo relevante [Aroca & Gonçalves 2012].

Um mito entre os desenvolvedores é que processadores de sistemas embarcados podem não ter desempenho aceitável para execução de um servidor web ou para execução de tarefas com demanda de alto poder de processamento. Outro aspecto importante em robótica é a economia de energia, já que os robôs são alimentados por bateria. O estudo realizado inclui a comparação de desempenho absoluta e da eficiência energética entre processadores ARM e processadores x86, que equipam os PCs tradicionais. Essas duas plataformas são bastante comuns em computadores de controle de robôs.

Um ambiente de testes foi preparado para realizar os testes de performance dos processadores ARM. Este ambiente pode ser visto na Figura 5.10, e consiste no uso de uma placa com processador ARM conectada um computador que gera dados de teste e coleta dados de performance e uso de energia, além de um multímetro digital com saída serial RS-232 conectada diretamente ao PC, que coleta todos dados a cada um segundo. A Figura 5.11 mostra uma foto da placa PandaBoard, com um processador ARM Cortex-A9 durante os testes. Cada dado apresentado nos gráficos de desempenho à seguir consistem em uma média de 30 valores coletados durante os testes. Além disso, os testes de servidor web foram realizados utilizando a ferramenta *Apache Benchmark* (ab).

A Figura 5.12 mostra a análise de desempenho da execução do servidor web Apache, um dos mais tradicionais, nas plataformas analisadas. Nota-se claramente que os processadores ARM oferecem uma vantagem em relação ao número de requisições atendidas pelo gasto de energia. Além disso, o desempenho absoluto é totalmente aceitável no contexto do Anwide, que requer apenas alguns acessos HTTP por segundo.

A Figura 5.13 mostra o resultado da análise de performance para execução de cálculos em ponto flutuante, que foi feita utilizando a ferramenta *linpack*. Embora os processadores x86 ofereçam desempenho absoluto maior, nota-se que os processadores ARM também podem realizar alguns milhões de operações de ponto flutuante por segundo, um valor aceitável para a execução das FFTs necessárias no sistema Anwide.

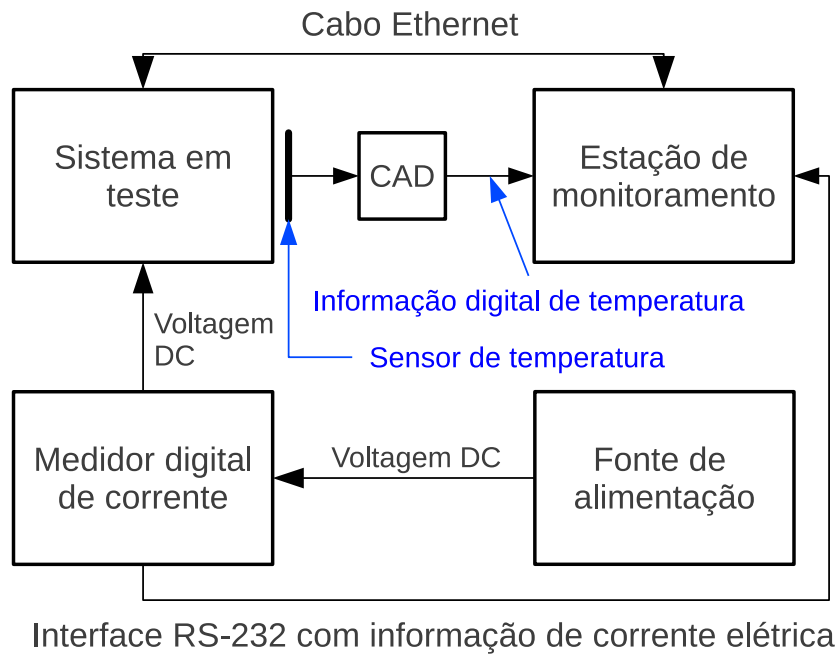


Figura 5.10: Esquema de teste dos processadores ARM.

5.3 Aspectos Educacionais

Apesar de não ser o foco principal da nossa proposta, procuramos sempre nos guiar por aspectos educacionais, relatados nesta Seção. Foram realizadas pesquisas com alunos no contexto de várias disciplinas do ensino superior e feitas análises qualitativas e quantitativas, bem como categorias foram estabelecidas visando prover melhor verificação de viabilidade e de confiabilidade para nossa proposta.

5.3.1 Uso em Sala de Aula

Pesquisas foram realizadas com diversos alunos sobre seus interesses em robótica antes e depois de conhecer a possibilidade de construir um robô didático de baixíssimo custo. Para tanto, uma aula foi ministrada apresentando aos alunos a ideia do *CellBot*: um robô controlado por um celular.

Materiais e Métodos

Primeiro, um questionário anônimo foi aplicado em todos os alunos que participaram da pesquisa para levantar suas concepções e interesses em robótica, bem como para obter informações sobre seus telefones celulares. Algumas semanas depois, eles assistiram a uma aula especial sobre *Como construir seu próprio robô de baixo custo usando um celular como seu cérebro*. Desde então, as interações e retornos dos alunos passaram a ser documentadas e posteriormente, um questionário foi aplicado para verificar se houve



Figura 5.11: Foto com o ambiente de teste de performance dos processadores ARM.

alguma mudança na opinião desses alunos, ou em seus níveis de motivação. Com idade média de 21 anos, 204 alunos matriculados em cursos de graduação distintos de quatro universidades brasileiras responderam à pesquisa. Os questionários aplicados encontram-se no apêndice A.

Questionário Prévio

Os resultados do questionário prévio mostram que 41.5% de todos os alunos preferem usar um kit didático disponível apenas em laboratório da universidade ao invés de comprar ou construir seu próprio robô. Dos 58.5% restantes, 78% preferem construir seu próprio robô, e 22% comprar um kit pronto para usar. Além disso, 23% desses alunos disseram que não gastariam nenhum dinheiro em um robô ou kit didático, 13% gastariam 25 dólares ou menos, 27% gastariam até 50 dólares, 18% gastariam até 125 dólares e 19% gastariam 125 dólares ou mais.

Apenas 1% dos estudantes não possuem telefone celular. Dos que o possuem, 27% compraram seus telefones nos últimos seis meses, 32% no último ano, 21% nos últimos dois anos e 19% nos últimos quatro anos. Com relação aos recursos técnicos dos telefones desses alunos, 68% possuem câmeras, 32% possuem receptores de GPS, 27% possuem acelerômetro, 47% possuem conexão wifi, 68% possuem *bluetooth* e 94% possuem conexão para fones de ouvido externos, que podem ser usados para conectar o N-Bot.

Questionário Posterior

O questionário posterior foi aplicado para verificar se alguma opinião dos alunos mudou após eles assistirem a aula de robótica de baixo custo usando celulares. Da amostra

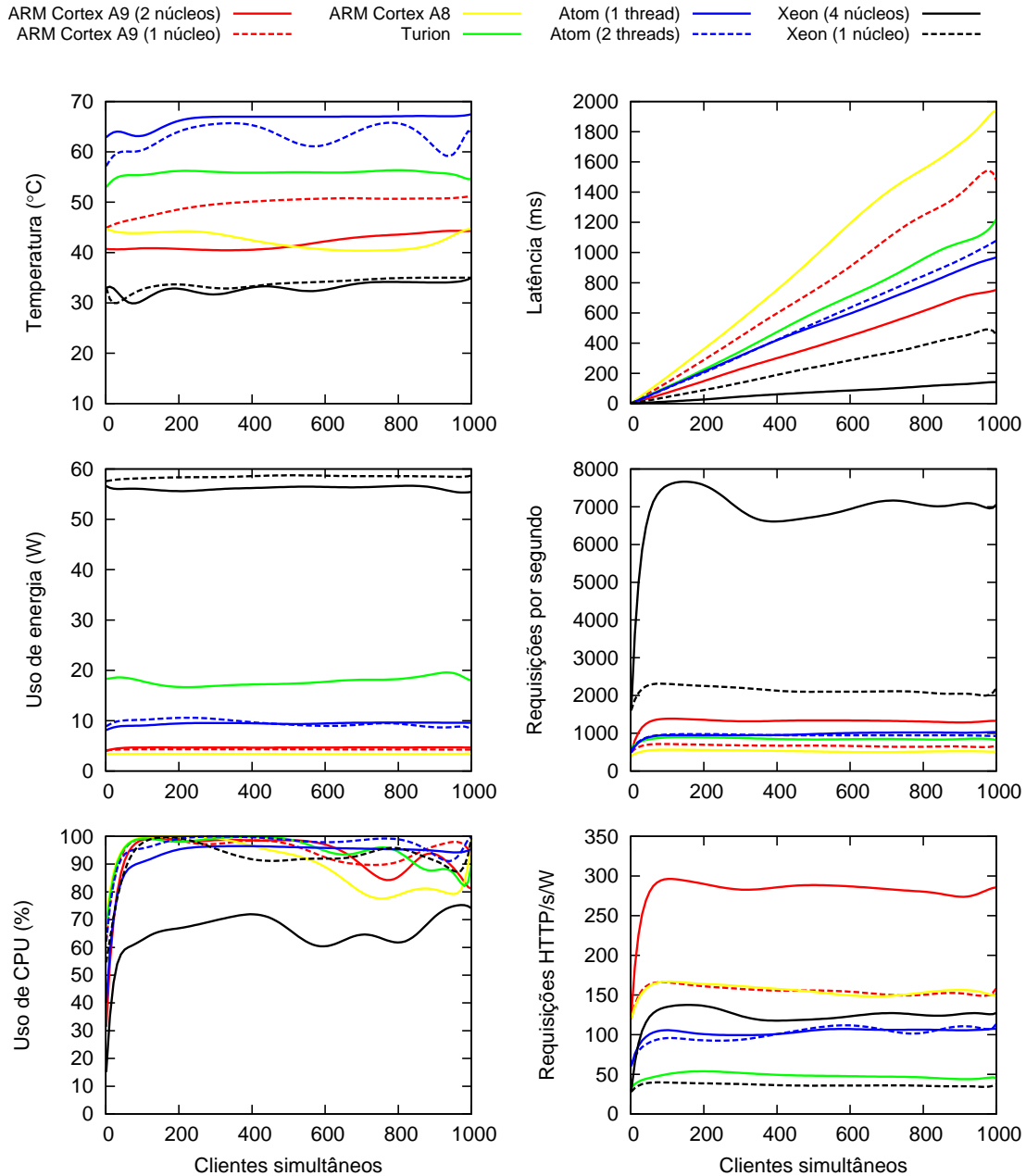


Figura 5.12: Resultados da performance absoluta e energética da execução de um servidor web Apache em várias plataformas mediante diversas condições de uso.

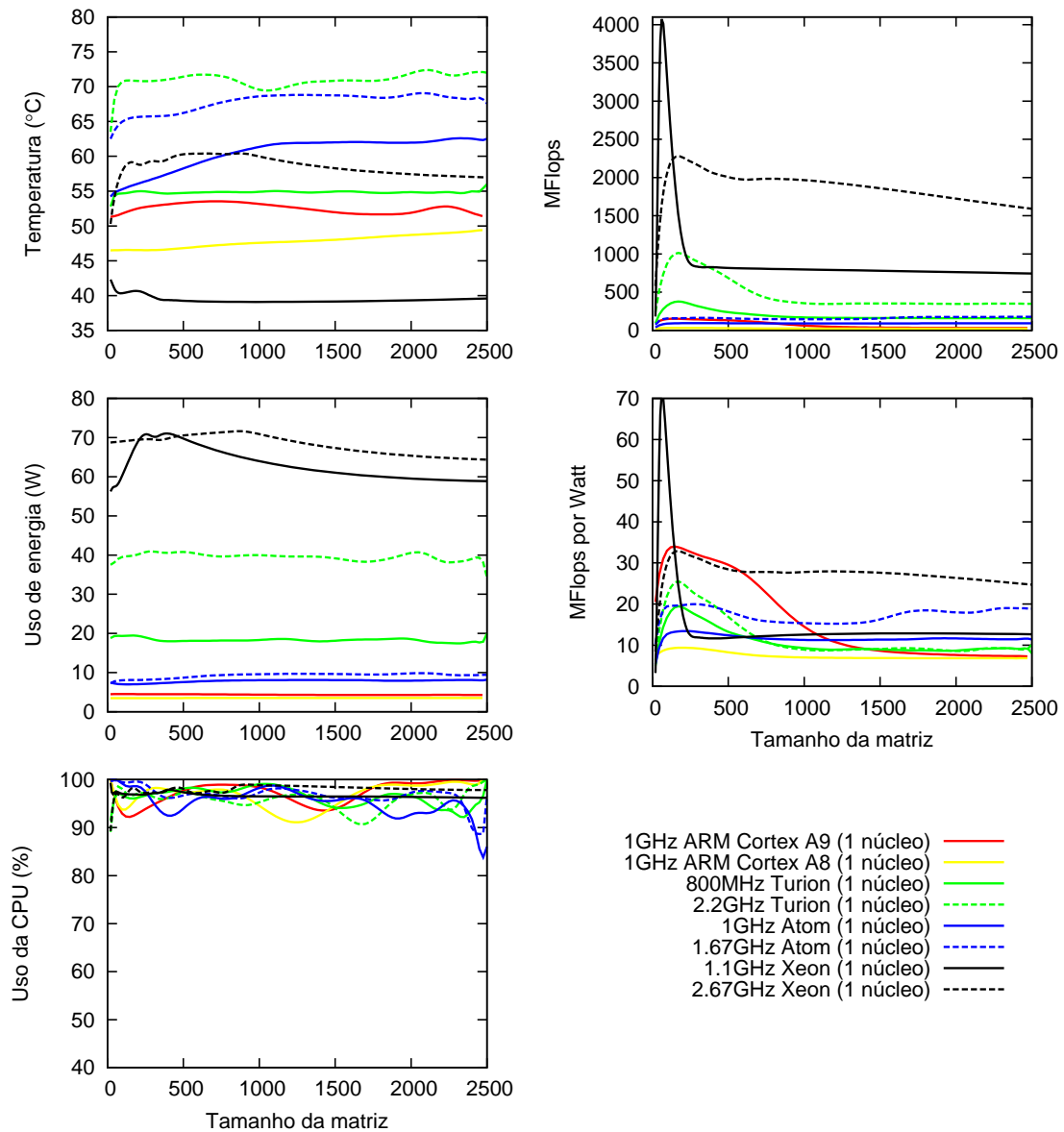


Figura 5.13: Resultados da análise de performance de ponto flutuante em diversos cenários de uso.

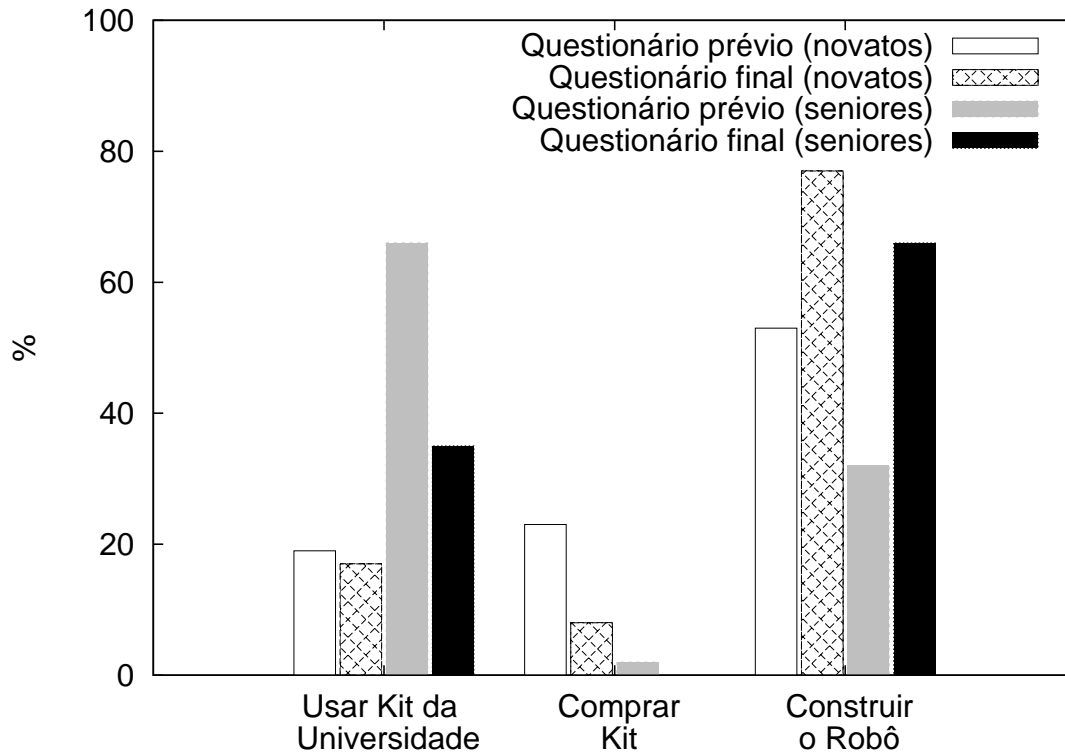


Figura 5.14: Interesse dos estudantes em construir um robô ou adquirir um kit de prateleira.

de 204 alunos, 83% disseram que seria mais interessante o aprendizado de vários assuntos usando um *CellBot* (para essa população, a estimativa da média está entre 76% e 89% com nível de confiança de 95%).

Enquanto na pesquisa prévia, 58.5% dos alunos disseram que preferem possuir seu próprio robô, este número aumentou para 71% no questionário posterior, mostrando que o *CellBot* aumentou o interesse geral dos alunos (para a população, a estimativa da média está entre 64% e 77% com nível de confiança de 95%).

A Figura 5.14 mostra o nível de interesse dos estudantes em construir seus próprios robôs antes e depois da aula. Os resultados são apresentados em duas categorias: novatos, que são alunos que entraram na universidade em 2010 ou 2011; e seniores, que são alunos que entraram na universidade em 2007. A Figura mostra que os alunos novatos tiveram opiniões divididas, provavelmente porque eles não sabiam quais opções estavam disponíveis, enquanto alunos seniores estavam mais inclinados a usar kits já disponíveis na universidade. A Figura 5.14 também mostra que depois de conhecer a possibilidade dos *CellBots*, muitos estudantes dos dois grupos migraram para a opção de construir seus próprios robôs. Uma mudança interessante foi observada em uma turma de Ciência da Computação, onde o interesse dos alunos em construir robôs subiu de 0% para 59%.

Como mostrado na Figura 5.15, depois da aula de robótica, a quantia de alunos dispostos a investir seu dinheiro para construir robôs subiu. É interessante notar que o pico ocorre em um valor próximo àquele do robô de baixo custo descrito para os alunos.

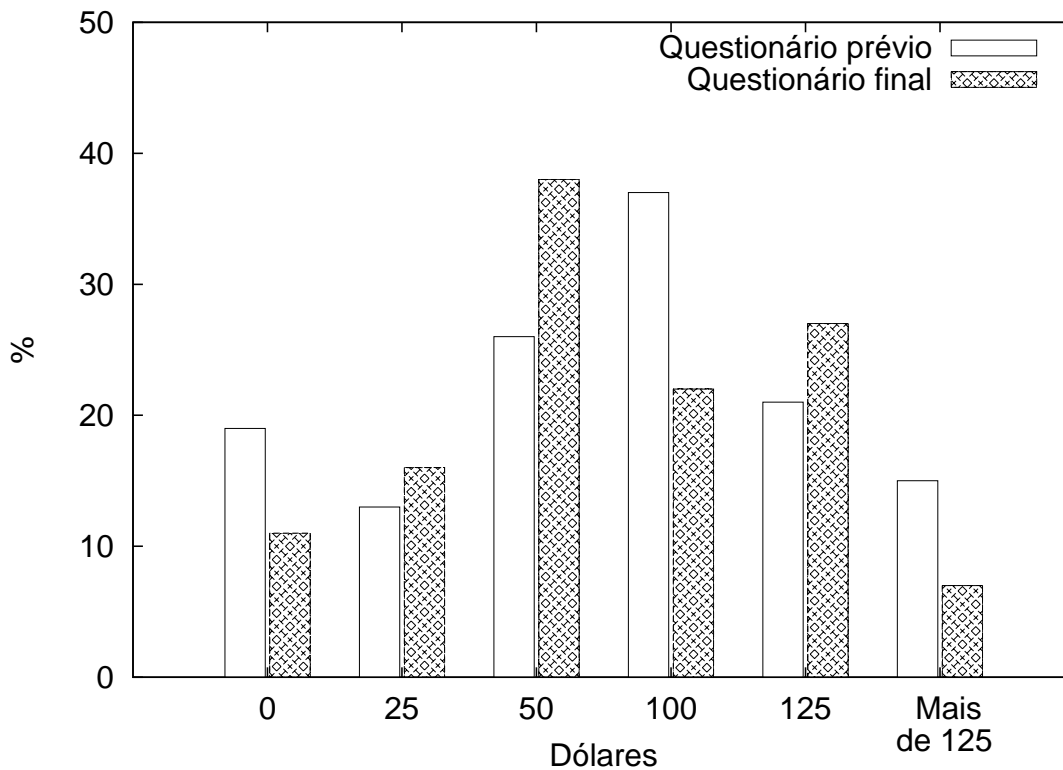


Figura 5.15: Distribuição de quanto os alunos estão dispostos em gastar para construir seus próprios robôs (valores em dólares americanos).

Os questionários também investigaram os interesses dos alunos em cursos de graduação antes e depois da aula sobre *CellBots*. Com exceção do curso de processamento digital de sinais (PDS), nenhuma diferença significativa foi observada entre o interesse dos alunos em cursar certas disciplinas antes e depois da aula. No caso do curso de processamento digital de sinais, o interesse dos alunos novatos subiu de 83% para 95%. Claramente esta mudança ocorreu devido a aula ter enfatizado a importância das técnicas de PDS para construção do *CellBot*.

A Figura 5.16 mostra o interesse dos alunos em vários aspectos de um robô de baixo custo. Como o questionário foi aplicado várias semanas após a aula, alguns alunos já estavam montando seus robôs (cerca de 8%). Como mostrado na Figura 5.16, a maioria dos alunos interessou-se pelo projeto e pretende construir seu robô. Aparentemente, estudantes seniores também estavam interessados no projeto, porém menos interessados em construir seus robôs, já que eles estão se aproximando de sua graduação e procurando outros desafios e oportunidades após a graduação. Além disso, para esses alunos, existem menos conceitos para aprender e desafios para vencer na construção do robô. A Figura 5.16 também mostra os assuntos que mais interessam aos alunos sobre aspectos de desenvolvimento do robô.

É interessante observar as mudanças que ocorreram na opinião dos estudantes após uma única aula de cerca de 40 minutos descrevendo as opções de robótica de baixo custo. Acreditamos que a motivação e interesse poderiam ser ainda maiores se robôs de baixo

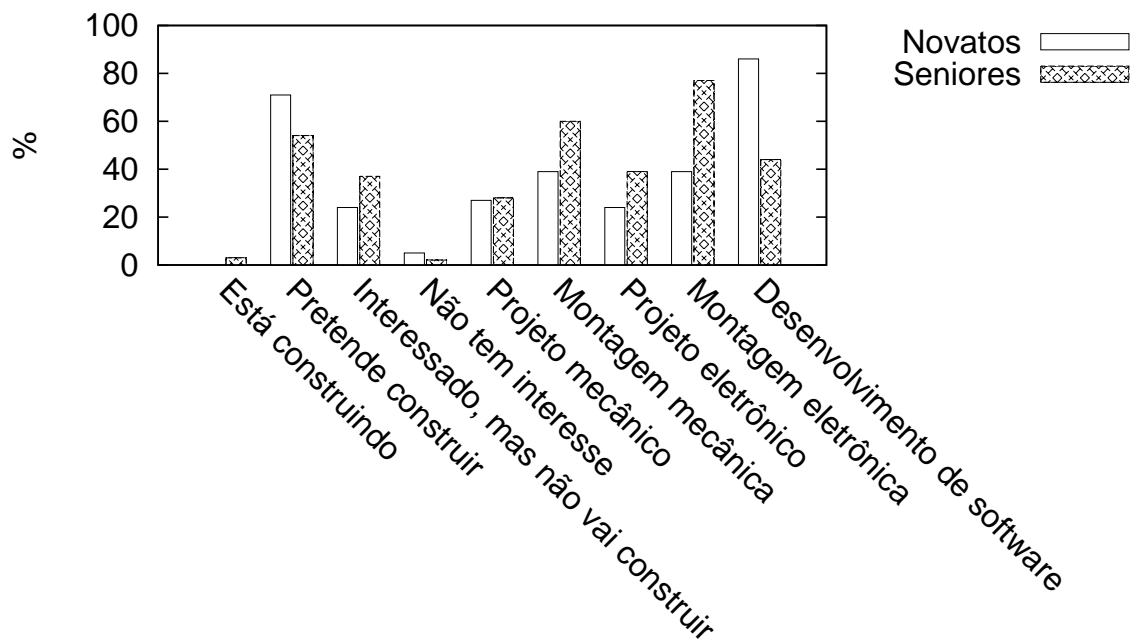


Figura 5.16: Interesses relatados pelos alunos sobre o *CellBot*.

custo que podem ser levados para casa fossem usados mais amplamente como ferramenta para aprender e testar conceitos de diversos cursos.

Os robôs de baixo custo, como o *CellBot* e o N-Bot, podem ser aplicados em diversos cursos, incluindo disciplinas que não são diretamente relacionadas com robótica, como por exemplo, programação web para implementar sistemas de controle de robôs via Internet, cálculo para utilizar os conceitos e definições básicos de controle em malha fechada, como o PID (Proporcional-Integral-Derivativo), Álgebra Linear para construir matrizes de transformação homogêneas para calcular a cinemática do robô, dentre muitos outros.

Análise Qualitativa

O último questionário deu aos alunos a oportunidade de deixar comentários escritos, alguns dos quais são replicados aqui. Além disso, alguns comentários e desdobramentos daquela única aula foram surpreendentes: por exemplo, em uma das universidades, os alunos pediram para que um curso regular de robótica passasse a ser oferecido no currículo de Ciência de Computação deles (aquela universidade não tinha nenhuma disciplina de robótica até então), sendo que a coordenação gostou da ideia e criou o novo curso.

Um resultado recompensador foi observado com estudantes que decidiram construir seus robôs. Por exemplo, um grupo de estudantes construiu um robô funcional em cerca de duas horas, sendo que essa foi a primeira vez que eles usaram um ferro de solda.

Alguns estudantes solicitaram que cursos ou minicursos de *CellBots* fossem ministrados. Um dos alunos deixou o seguinte comentário: *o conceito de robótica deveria ser adicionado a cursos de graduação, visto que esse tópico causa grande curiosidade e interesse*. Outro estudante comentou: *essas iniciativas deveriam ser publicadas e divulgadas*

mais amplamente porque é muito interessante e essencial para alunos de graduação.

Diversos alunos afirmaram que a robótica deveria ser mais estudada nas universidades, e que o *CellBot* é uma boa opção por ser fácil de montar e poder ser usado em disciplinas para aprender técnicas básicas de construção de robôs.

5.3.2 Oficina de Robótica de Baixo Custo

Mais recentemente, tivemos a oportunidade de ministrar dois minicursos de robótica de baixo custo. Um deles, no formato de uma oficina, ocorreu durante o X SINEC (Seminário de Informática e Engenharia de Computação), que foi realizado na UFRN. Nessa oficina, o valor da taxa de inscrição no evento foi usado para comprar peças e montar kits que foram fornecidos como brinde a cada aluno da oficina, permitindo que eles construíssem seus robôs durante o evento. Após a oficina, algumas perguntas foram feitas para avaliar a qualidade do minicurso. Essas perguntas encontram-se no apêndice A.

A oficina teve duração de 4,5 horas, sendo que os primeiros 60 minutos foram dedicados a uma apresentação do N-Bot e dicas de montagem. O restante do tempo foi usado para montagem dos robôs, sendo que além do palestrante, mais quatro monitores estavam disponíveis para ajudar e tirar dúvidas de montagem. Para economizar tempo, as placas de circuito impresso foram parcialmente montadas, restando alguns componentes para os alunos soldarem durante a oficina.

Cerca de metade dos 25 alunos que participaram da oficina conseguiu terminar a montagem do robô (sem a fixação das rodas) durante a oficina. Todos participantes foram convidados a responder um breve questionário sobre a oficina, sendo que doze alunos responderam esse questionário.

Entre os doze alunos que responderam a pesquisa, 83% afirmaram que não achavam possível construir um robô por R\$50,00; Com relação ao tempo do minicurso, 75% dos alunos acharam que a duração foi curta. Além disso, 92% dos alunos disseram que recomendariam o minicurso para outro colega.

Sobre a finalização da montagem, 42% dos alunos nunca tinham montado algum circuito anteriormente e 75% dos alunos não conseguiram terminar a montagem durante o minicurso. É interessante o fato de que entre os participantes que já tinham montado algum circuito antes, 50% terminaram a montagem do robô, e os outros 50% montaram 76% do robô (em média). Já, dos alunos que nunca tinham montado um circuito antes, apenas 14% terminaram a montagem, mas afirmaram que completaram 73% da montagem (média). Esses dados indicam, que de fato, como apontado pelos alunos, a duração do minicurso foi muito curta.

Ainda nessa pesquisa, 100% dos alunos acreditam que seria interessante usar o N-Bot em aulas de graduação; 50% dos alunos consideraram o minicurso *Muito bom*, 33% consideraram *Bom* e 17% *Regular* (nenhum aluno achou que foi *Ruim* ou *Muito ruim*).

A maior crítica dos alunos foi sobre o tempo do minicurso, e o tempo gasto na soldagem dos componentes, deixando pouco, ou nenhum tempo para os outros testes do robô.

5.3.3 Possíveis Usos

Este trabalho apresenta várias soluções de hardware e software para uso em ambientes educacionais, sendo que notamos que algumas soluções são mais adequadas para determinados níveis de ensino, e assim propomos indicações de uso das diferentes técnicas apresentadas para diferentes ambientes de ensino.

A versão com microcontrolador, por exemplo, é uma alternativa interessante para alunos de universidades, já que é comum estes alunos já possuírem noções de programação, bem como de microcontroladores. Notamos, porém, que em alguns cursos técnicos de eletrônica ou eletro-técnica, os alunos só aprendem noções de microcontroladores e programação no final do curso, dificultando o uso e construção do N-Bot. Dessa forma, a opção sem microcontrolador torna-se interessante como uma etapa inicial, já que não requer do aluno qualquer conhecimento de programação ou microcontroladores, mas apenas a capacidade de construir o circuito e o robô em si. Esta versão do robô também pode ser útil para calouros de universidades que ainda não tenham noção de programação nem de microcontroladores.

Nesta versão, ainda, o aluno pode escolher entre um circuito para controlar motores DC ou servos, possibilitando reaproveitamento de motores, brinquedos, ou mesmo o uso de material novo.

Com ambas as versões é possível explorar conteúdos didáticos de forma abrangente, como redes de computadores, sistemas operacionais, controle, robótica, álgebra linear, programação de computadores, algoritmos, programação web, visão computacional, eletrônica analógica, eletrônica digital, processamento de sinais, dentre outras.

Finalmente, o software Anwide pode ser usado de forma complementar a outras plataformas robóticas já presentes em determinada instituição para facilitar a sua programação e exploração pelos alunos, como já foi mostrado neste capítulo o exemplo de controle de um Lego Mindstorms pelo Anwide. Esta abordagem une o melhor de dois mundos, já que oferece a flexibilidade e facilidade de programação remota do Anwide unido aos sensores e atuadores dos kits Lego NXT.

Capítulo 6

Conclusão e Perspectivas

Stay hungry, stay foolish.

The Whole Earth Catalog

Formalmente, podemos concluir que a tese que foi defendida neste texto é a de que é possível desenvolver robôs de baixíssimo custo com material comercialmente disponível (conseguimos propor e desenvolver um robô cujo custo é menor que 14 dólares). Suplementamos esse robô com uma arquitetura de controle via canal de áudio, bem como com ambientes de programação simplificados visando que ele seja acessível ao público mais diverso, incluindo alunos desde o ensino fundamental e médio, até o ensino superior, bem como, possivelmente, o público leigo.

Para demonstrar esta tese, propomos neste trabalho uma abordagem totalmente nova, incluindo arquitetura de hardware e software, além de uma nova técnica de controle, que pode ser empregada no desenvolvimento de robôs de baixíssimo custo. A técnica de controle proposta é simples, porém universal, permitindo a construção de circuitos de acionamento de atuadores e leitura de sensores em robôs utilizando canais de áudio. Com isso, o ciclo de controle pode ser fechado e sistemas de controle operando em malha fechada, essenciais para desenvolvimento de pesquisas e outras atividades em robótica, podem ser desenvolvidas sobre nossa proposta.

Na arquitetura proposta, diversos tipos de dispositivos podem ser usados como unidade de controle da plataforma apresentada, permitindo o controle em malha fechada. Além disso, a unidade de controle pode ser facilmente trocada, já que basta conectar um cabo de áudio, sem a necessidade de ajustes, configurações ou a instalação de *drivers* de dispositivos. O desempenho do sistema, como um todo, incluindo o tempo de cada ciclo de controle, permite execução de tarefas em tempo real. A arquitetura mostrou-se eficaz quando implementada no sistema Android, mas funciona também em outros sistemas operacionais convencionais (Windows/Linux) e usando diversas linguagens de programação.

Entre as várias aplicações do sistema proposto, destaca-se seu uso para construção de um robô educacional de baixíssimo custo, o N-Bot. No contexto do N-Bot, ainda como uma contribuição nossa, este trabalho também contempla a proposta e implementação de dois ambientes de programação de robôs via web, dispensando a necessidade de instalação de qualquer software na estação de desenvolvimento. Um dos ambientes é baseado

em um servidor web que executa em um dispositivo móvel, como um *tablet* ou um telefone celular, e o outro consiste de uma página web que pode ser acessada a partir de qualquer computador que possua um navegador web.

Também como uma contribuição do trabalho desenvolvido, incluímos uma verificação empírica, com a aplicação e avaliação do uso das técnicas descritas aqui junto a diversos alunos, mostrando que o uso de robôs, especialmente de baixíssimo custo, aumenta o interesse e motivação dos alunos em participar de atividades de construção, projeto e desenvolvimento de software para robótica. Esses fatores podem ser usados para aumentar o interesse dos alunos em participarem de aulas e em estudar conceitos apresentados no curso de graduação, que sem um contexto ou aplicação, podem parecer menos interessantes.

Outro aspecto interessante observado experimentalmente é que a maioria dos alunos que participaram do estudo se mostraram interessados em construir seus próprios robôs, e após saber da possibilidade de construir robôs com baixo custo, eles disseram estar dispostos a gastar o próprio dinheiro para investir em tal ferramenta didática. Também foi observado que pelo menos metade dos alunos possuem telefones celulares com recursos e sensores sofisticados o suficiente para construir robôs com alto poder computacional e sensorial. Além disso, 71% dos alunos ficaram interessados em possuir seus próprios robôs, e 83% dos alunos disseram que o uso de um robô de baixo custo tornaria mais fácil o estudo de várias disciplinas.

Como citado no texto, para construção artesanal, em pequena escala, o custo do N-Bot é de cerca de US\$ 14,00, mas podendo ser reduzido para menos de dez dólares por unidade caso se adquira os componentes em escala maior.

Outro aspecto que consideramos importante, é o reaproveitamento de telefones celulares como base de uma plataforma robótica, especialmente no Brasil, que em Maio de 2012 tinha aproximadamente 255 milhões de celulares [Teleco 2012a] (1,3 celulares para cada habitante [Teleco 2012a]). Além disso, o telefone celular, é o segundo equipamento eletrônico mais presente na vida de nossa população. Assim, uma pesquisa feita em 2010, que incluiu áreas rurais, mostrou que 95% dos lares possuem televisão, seguido de 83% que possuem celulares. Já os lares com microcomputador são apenas 38% [Teleco 2012b]. Além disso, outra pesquisa mostra que 77% das famílias que ganham menos de 10 salários mínimos possuem celulares. Para as pessoas que ganham mais que 10 salários, este número é superior a 97% [Teleco 2011].

6.1 Contribuições

As contribuições principais da tese são a arquitetura de hardware e software [Aroca, Burlamaqui & Gonçalves 2012] e a nova técnica de controle, usando áudio, que podem ser empregadas no desenvolvimento de robôs de baixíssimo custo. Além disso, podemos citar também como contribuição a interface de software, incluindo os ambientes de controle e programação do robô [Aroca, Guardiman & Gonçalves 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c], desenvolvidos, que usam um novo paradigma de programação de robôs via web. Esses ambientes podem ser implementados usando celular, *tablet*, computador, entre outros dispositivos.

Ainda entre as contribuições resultantes deste trabalho, ressaltamos o desenvolvimento do N-Bot, que foi premiado em concurso internacional relevante para o desenvolvimento de robôs educativos de baixíssimo custo, o *AFRON 10-dollar robot design challenge*. O N-Bot tem o projeto e software abertos e gratuitamente disponíveis no sítio web do laboratório NatalNet [Aroca, Pitta, Burlamaqui & Gonçalves 2012a]. Como meio de atestar e disseminar nossas contribuições científicas à comunidade, diversos artigos foram publicados em assuntos relacionados ao N-Bot, sendo três dos trabalhos em reconhecidas revistas de primeira linha [Aroca, Burlamaqui & Gonçalves 2012, Aroca, Gomes, Tavares, Souza, Burlamaqui, Caurin & Gonçalves 2012, Aroca & Gonçalves 2012]. Tivemos também uma patente sobre controle em malha fechada de dispositivos mecânicos usando canais de áudio depositada junto ao INPI [Aroca et al. 2011], bem como vários outros trabalhos relacionados ao tema da nossa tese [Aroca, Guardiman & Gonçalves 2012, Aroca, de Oliveira & Gonçalves 2012, Júnior et al. 2012, Aroca, Pitta, Burlamaqui & Gonçalves 2012c].

6.2 Trabalhos Futuros

Existem muitos trabalhos planejados para dar continuidade ao N-Bot, dentre os quais destacamos alguns aqui.

Para o ensino médio e fundamental, pretende-se desenvolver e adaptar os ambientes de programação do N-Bot e do robô em si, se necessário, para aplicação com crianças e adolescentes. Neste caso, a ideia principal é que a construção do robô deve ser facilitada, de modo que trabalhos manuais com ferro de solda e cortes com alicates serão evitados de serem executados pelas crianças. Além disso, espera-se estabelecer parcerias com escolas e operadoras de telefonia celular, que poderiam doar celulares usados para os alunos construírem seus robôs.

Também vemos várias possibilidades a serem exploradas no campo da pesquisa, tal como a implementação de sistemas de localização e mapeamento (SLAM) e de odometria visual para oferecer aos usuários do N-Bot uma forma simples de executar tarefas de deslocamento entre dois pontos de um mapa. Pode-se ainda explorar a integração com serviços de robótica nas nuvens para o robô utilizar recursos externos e algoritmos complexos de reconhecimento de padrões, além da possibilidade de integrar o N-Bot com o ROS, possibilitando o desenvolvimento de pesquisas em robótica usando uma plataforma de baixo custo.

Outros trabalhos futuros podem envolver diversas melhorias e otimizações do sistema proposto neste trabalho, tal como otimizações mecânicas, eletrônicas e de software para controle mais suave e preciso dos motores. Também podem ser implementados algoritmos de filtragem e processamento de sinais para filtrar harmônicas e ruídos do sinal de áudio e estimar com precisão a posição das rodas do robô. Finalmente, realizar outras análises dos limites do sistema par controle em malha fechada, bem como a resposta do sistema em diferentes frequências.

Referências Bibliográficas

- Ahlgren, D.J. (2002), Meeting educational objectives and outcomes through robotics education, *em* 'Automation Congress, 2002 Proceedings of the 5th Biannual World', Vol. 14, pp. 395 – 404.
- Ahn, Ho Seok, In-Kyu Sa & Jin Young Choi (2009), 'Pda-based mobile robot system with remote monitoring for home environment', *Consumer Electronics, IEEE Transactions on* **55**(3), 1487 –1495.
- Aleman, Jaime & Enric Cervera (2012), Appealing robots as a means to increase enrollment rates: a case study, *em* 'Anais da 3a. conferência Internacional de Robótica na Educação. 13-15 de Setembro. Praga, República Tcheca.', pp. 15–19.
- Alifragis, M., A. Mantelos & C.S. Tzafestas (2008), Web-based remote and virtual programming console of the v+ robotic system, *em* 'Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE', pp. 3551 –3555.
- Alves, S, Humberto Ferasoli Filho, Rene Pegoraro, M. Caldeira, J. Rosário & W. Yonezawa (2011a), 'Educational environment for robotic applications in engineering', *Research and Education in Robotics-EUROBOT 2011* pp. 17–28.
- Alves, S., Humberto Ferasoli Filho, Rene Pegoraro, M. Caldeira, J. Rosario & W. Yonezawa (2011b), Proposal of educational environments with mobile robots, *em* 'IEEE Conference on Robotics, Automation and Mechatronics (RAM)', pp. 264–269.
- Angonese, Alberto Torres, Paulo Fernando Ferreira Rosa & Stefano Henrique Rodrigues (2012), Projeto de integração engenharia-escola para competições de robótica, *em* 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Aroca, Rafael V., Antônio Péricles de Oliveira & Luiz Marcos Garcia Gonçalves (2012), Towards smarter robots with smartphones, *em* 'Robocontrol: Proceedings of 5th workshop in applied robotics and automation', pp. 1–6.
- Aroca, Rafael V., Aquiles F. Burlamaqui & Luiz M. G. Gonçalves (2011), 'Sistema para controle em malha fechada de equipamentos mecatrônicos por meio do canal de áudio de dispositivos móveis', Patente Brasileira número PI-1107288-1. Data de prioridade: 30 de Novembro de 2011.

- Aroca, Rafael V., Aquiles F. Burlamaqui & Luiz Marcos Garcia Gonçalves (2012), 'Method for reading sensors and controlling actuators using audio interfaces of mobile devices', *Sensors* **12**(2), 1572–1593.
URL: <http://www.mdpi.com/1424-8220/12/2/1572/>
- Aroca, Rafael V. & Luiz Marcos Garcia Gonçalves (2012), 'Towards green data centers: A comparison of x86 and arm architectures power efficiency', *J. Parallel Distrib. Comput.* **72**(12), 1770–1780.
URL: <http://dx.doi.org/10.1016/j.jpdc.2012.08.005>
- Aroca, Rafael V., Rafael Gomes, Dalton Tavares, Anderson Souza, Aquiles Burlamaqui, Glauco Caurin & Luiz Gonçalves (2012), 'Increasing students' interest with low-cost cellbots', *Education, IEEE Transactions on* **PP**(99), 1.
- Aroca, Rafael V., Renata Pitta, Aquiles Burlamaqui & Luiz Marcos Garcia Gonçalves (2012a), 'N-bot: A low cost educational robot', On-Line. Acesso em Out/2012. Disponível em: <http://www.natalnet.br/~aroca/afron/>.
- Aroca, Rafael V., Renata Pitta, Aquiles Burlamaqui & Luiz Marcos Garcia Gonçalves (2012b), 'N-bot on the media', On-Line. Acesso em Out/2012. Disponível em: <http://www.natalnet.br/~aroca/afron/media.html>.
- Aroca, Rafael V., Renata Pitta, Aquiles Burlamaqui & Luiz Marcos Garcia Gonçalves (2012c), Um robô por aluno: uma realidade possível, em 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Aroca, Rafael V., Renato Q. Guardiman & Luiz Marcos Garcia Gonçalves (2012), Web-based robot programming environment and control architecture, em 'Latin American Robotics Symposium / Simpósio Brasileiro de Robótica'.
- Barros, Renata Pitta (2008), 'Roboeduc- uma ferramenta para programação de robôs lego', On-Line. Acesso em Out/2012. Disponível em: http://www.dca.ufrn.br/~renata/Artigos/Monografia_%20Renata.pdf.
- Belousov, Igor R., Ryad Chellali & Gordon Clapworthy (2001), Virtual reality tools for internet robotics, em 'ICRA', IEEE, pp. 1878–1883.
- Bieniek, Grégori Betiato, Luana Tortelli, Mirian Cátia Zarpelon, Anibal Lopes Guedes, Rodrigo Saballa de Carvalho & Fernanda Lopes Guedes (2012), Robótica como alternativa nos processos educativos da educação infantil e dos anos iniciais do ensino fundamental, em 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Botelho, Silvia S. C., Lilian G. Braz & Ricardo N. Rodrigues (2012), Exploring creativity and sociability with an accessible educational robotic kit, em 'Anais da 3a. conferência Internacional de Robótica na Educação. 13-15 de Setembro. Praga, República Tcheca.', pp. 55–60.

- Carter, John (2011), 'Robophone', On-Line. Jelly Robotics. Acesso em Mar/2012. Disponível em: <http://www.jellyrobotics.com/2011/12/17/robophone/>.
- CellBots (2011), 'Cellbots: Using cellphones as robotic control platforms', On-Line. Access date: March 09 2012. URL: <http://www.cellbots.com>.
- Chella, Marco Túlio (2012), Simrobô: simulador para robótica com propósito educacional, em 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Chitode, J.S. (2008), *Digital Signal Processing*, 2nd^a edição, Technical Publications Pune: Pune, India.
- Cho, Yun Chan & Jae Wook Jeon (2008), Remote robot control system based on dtmf of mobile phone, em 'Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on', pp. 1441 –1446.
- Conrad, J.M. (2005), 'Stiquito for robotics and embedded systems education', *Computer* **38**(6), 77–81.
- da Silva, Alexandre J. B. & Eliana da Silva Almeida (2012), Integração de múltiplas plataformas robóticas no ensino fundamental e médio, em 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Dalmazo, Luiza (2012), 'A mente brilhante de paulo blikstein', On-Line. Revista Exame de 25/Jan/2012. Edição 1008. Acesso em Nov/2012. Disponível em: <http://exame.abril.com.br/revista-exame/edicoes/1008/noticias/a-mente-brilhante-de-paulo-blikstein>.
- Darrell, Dammit (2010a), 'Exciting discovery, servo control using only audio', On-Line. Publicado em Mai/2010. Grupo de discussão CellBots. Acesso em Nov/2012. Disponível em: <https://groups.google.com/forum/?fromgroups=#!msg/cellbots/2BgBjZzBTkE/3cKU0sKmJLsJ>.
- Darrell, Dammit (2010b), 'How the trrstan kit works', On-Line. Acesso em Nov/2012. Disponível em: <http://www.cellbots.com/uncategorized/how-the-trrstan-kit-works>.
- Davidson, D.M., D.C. Yates, B. Marks & F. Rumsey (1999), *Distribution of audio signals*. In Audio Engineer's Reference Book, 2nd^a edição, Focal Press: Woburn, MA, USA.
- de Lima Sá, Sarah Thomaz (2011), 'H-educ: Um hardware de baixo custo para a robótica educacional', Trabalho de Conclusão de Curso. On-Line. Acesso em Out/2012. Disponível em: http://www.dca.ufrn.br/~renata/Artigos/Monografia_Sarah.pdf.

- de Molengraft, René, van, Michael Beetz & Toshio Fukuda (2011), 'A special issue toward a www for robots [from the guest editors]', *Robotics Automation Magazine, IEEE* **18**(2), 20.
- de Santana Souza, Anderson Abner & Luiz Marcos Garcia Gonçalves (2012), 2.5-dimensional grid mapping from stereo vision for robotic navigation, *em* 'Latin American Robotics Symposium / Simpósio Brasileiro de Robótica'.
- DealExtreme (2012), 'iphone / android controlled rechargeable 3.5-ch r/c helicopter w/ 300kp camera - black (tf slot)', On-Line. Publicado em Abr/2012. Acesso em Nov/2012. Disponível em: <http://dx.com/p/iphone-android-controlled-rechargeable-3-5-ch-r-c-helicopter-w-300kp-camera-black-tf-slot-1&p=2&m=3&r=2&k=1&t=1&s=139540&u=129184>.
- Dinh, Huan & T. Inanc (2009), Low cost mobile robotics experiment with camera and sonar sensors, *em* 'American Control Conference, 2009. ACC '09.', pp. 3793 –3798.
- Dipietro, L., A.M. Sabatini & P. Dario (2008), 'A survey of glove-based systems and their applications', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **38**(4), 461 –482.
- do Nascimento, Cynthia Vasconcellos, Luiz Marcos G. Gonçalves & Anderson A. S. Souza (2012), Closing the loop: Towards a robotic system based on sun spot architecture, *em* 'Robocontrol: Proceedings of 5th workshop in applied robotics and automation', pp. 1–7.
- Durda, Frank (1995), 'Dual tone multi-frequency (touch-tone) reference', On-Line. Publicado em 1995. Acesso em Nov/2012. Disponível em: <http://nemesiis.lonestar.org/reference/telecom/signaling/dtmf.html>.
- Fernandes, Carla C., Sarah T. Sá & Luiz Marcos G. Gonçalves (2012), Uma nova abordagem em robótica educacional utilizando simuladores e kits de robótica livre, *em* 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Fernandez, J. & A. Casals (2004), Open laboratory for robotics education, *em* 'Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on', Vol. 2, pp. 1837 – 1842 Vol.2.
- Ferrill, Paul (2011), *Pro Android Python with SL4A*, Apress.
- Firebaugh, S.L. & J.A. Piepmeier (2008), 'The robocup nanogram league: An opportunity for problem-based undergraduate education in microsystems', *Education, IEEE Transactions on* **51**(3), 394 –399.
- fis (2012), 'fischertechnik gmbh', On-Line. Acesso em Dez/2012. Disponível em: <http://www.fischertechnik.de/en/Home.aspx>.

- Galvan, S., D. Botturi, A. Castellani & P. Fiorini (2006), Innovative robotics teaching using lego sets, *em* 'Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on', pp. 721 –726.
- Garcia, L.-M., A.A.F. Oliveira, R.A. Grupen, D.S. Wheeler & A.H. Fagg (2000), 'Tracing patterns and attention: humanoid robot cognition', *Intelligent Systems and their Applications, IEEE* **15**(4), 70 –77.
- Garrett, Aaron & David Thornton (2005), A web-based programming environment for lego mindstorms robots, *em* 'Proceedings of the 43rd annual Southeast regional conference - Volume 2', ACM-SE 43, ACM, New York, NY, USA, pp. 349–350.
URL: <http://doi.acm.org/10.1145/1167253.1167333>
- Goebel, Stephan, Ruben Jubeh, Simon-Lennert Raesch & Albert Zuendorf (2011), Using the android platform to control robots, *em* 'Anais da 2a. conferência Internacional de Robótica na Educação. Setembro. Viena, Austria', pp. 135–142.
URL: http://www.innoc.at/fileadmin/user_upload/_temp_/RiE/Proceedings/65.pdf
- Goertzel, G. (1985), 'An algorithm for the evaluation of finite trigonometric series', *Am. Math. Monthly* **65**, 34–35.
- Goldberg, K & R Siegwart (2001), *Beyond Webcams: An Introduction to Online Robots*, MIT Press.
- Gonçalves, Luiz M. G. (2012), 'Educational robotics: History and challenges', On-Line. Acesso em Out/2012. Disponível em: <http://www.natalnet.br/~lmarcos/Presentations/EducationalRoboticsLARS.pdf>.
- Google (2012), 'Android', On-Line. Acesso em Nov/2012. Disponível em: <http://www.android.com/>.
- Greenwald, L. & J. Kopena (2003), 'Mobile robot labs', *Robotics Automation Magazine, IEEE* **10**(2), 25 – 32.
- Guizzo, E. (2011), 'Robots with their heads in the clouds', *Spectrum, IEEE* **48**(3), 16 –18.
- Guizzo, E. & T. Deyle (2012), 'Robotics trends for 2012 [the future is robots]', *Robotics Automation Magazine, IEEE* **19**(1), 119 –123.
- Gumbley, L. & B.A. MacDonald (2005), Development of an integrated robotic programming environment, *em* 'Proceedings of the 2005 Australasian Conference on Robotics and Automation, ACRA 2005'. cited By (since 1996) 0.
- H., Shawn, Carlos M., Bryan S. & Eric C. (2012), 'Baobot', On-Line. Acesso em Nov/2012. Disponível em: <http://baobot.org/>.

- Hamblen, J.O. & T.S. Hall (2004), Engaging undergraduate students with robotic design projects, *em* 'Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA)', pp. 140–145.
- Hamer, Tyler, Alexander E. Siy, Kojo Welbeck, Cagdas D. Onal & Daniela Rus (2012), 'Seg: Mit's origami inspired segway robot', On-Line. Acesso em Nov/2012. Disponível em: http://people.csail.mit.edu/cagdas/tmpAFRON/SEG/SEG_MITs_Origami_Inspired_Submission.html.
- Hernando, M., R. Galan, I. Navarro & D. Rodriguez-Losada (2011), 'Ten years of cyber-tech: The educational benefits of bullfighting robotics', *Education, IEEE Transactions on* **54**(4), 569–575.
- Hess, D. & C. Rohrig (2009), Remote controlling of technical systems using mobile devices, *em* 'Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2009. IDAACS 2009. IEEE International Workshop on', pp. 625–628.
- Hickman, Ryan, Damon Kohler, Ken Conley & Brian Gerkey (2011), 'Google i/o 2011: Cloud robotics', On-Line. Access date: March 09 2012. URL: <http://www.youtube.com/watch?v=FxXBUp-4800>.
- Honig, Zach (2012), 'Insert coin: Botiful telepresence robot for android', On-Line. Publicado em Jul/2012. Acesso em Nov/2012. Disponível em: <http://www.engadget.com/2012/07/27/insert-coin-botiful-telepresence-robot/>.
- Howard, A. & E. Graham (2007), To encourage and excite the next generation of engineers through human-robot interaction projects for space exploration, *em* 'American Society for Engineering Education Annual Conference, Hawaii, June 2007, Hawaii, June'.
- Hurst, Nathan (2012a), 'Can a \$10 robot save african education?', *Revista Wired* (Versão On-Line). Acesso em Out/2012. Disponível em: <http://www.wired.com/design/2012/07/10-dollar-robot-challenge/>.
- Hurst, Nathan (2012b), 'These \$10 robots will change robotics education', *Revista Wired* (Versão On-Line). Acesso em Out/2012. Disponível em: <http://www.wired.com/design/2012/09/afron-winners>.
- Inc, Ologic (2012), 'Oddwerx - autonomous smartphone robots', On-Line. Publicado em Mar/2012. Acesso em Nov/2012. Disponível em: <http://www.oddwerx.com/>.
- Ino (2012), 'Cientistas propõem um robô por aluno', *Redação do Site Inovação Tecnológica*. On-Line. Disponível em: <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=um- robo-poraluno>. Acesso em Jul/2012.
- Jobe, Ebrima, James McLurkin & Chutima Boonthum-Denecke (2012), R-one swarm robot: Developing the accelerometer and gyroscope, *em* 'Florida Artificial Intelligence Research Society Conference'.

- Jordan, Lucas & Pieter Greyling (2011), *Practical Android Projects*, Apress.
- Júnior, Andouglas S., Rafael V. Aroca, Aquiles F. Burlamaqui & Luiz Marcos Garcia Gonçalves (2012), 'Os dispositivos móveis no processamento de robôs de baixo custo', *HOLOS - ISSN 1807-1600* **1**(0).
URL: <http://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/811>
- Khan, A.A. (2005), *Digital Signal Processing Fundamentals*, 1stª edição, Da Vinci Engineering Press: Hingham, MA, USA.
- Korsah, Ayorkor & Ken Goldberg (2012a), 'The afron "10 dollar robot" design challenge', On-Line. Acesso em Out/2012. Disponível em: http://robotics-africa.org/design_challenge.html.
- Korsah, Ayorkor & Ken Goldberg (2012b), 'Announcing: The afron "10 dollar robot" design challenge', On-Line. Acesso em Out/2012. Disponível em: http://robotics-africa.org/design_challenge_files/AFRONDesignChallenge.pdf.
- Kramer, K.A. & M. Anderson (2011), A pic18 robot-centered microcontroller systems laboratory, em 'Frontiers in Education Conference (FIE)', pp. T2G-1-T2G-6.
- Kranjec, Jernej (2010), 'Demodulator updated', On-Line. Acesso em Dez/2012. Disponível em: <http://perceptumvollo.blogspot.com.br/2010/01/demodulator-updated.html>.
- Kuo, Ye-Sheng, Sonal Verma, Thomas Schmid & Prabal Dutta (2010), Hijacking power and bandwidth from the mobile phone's audio interface, em 'Proceedings of the First ACM Symposium on Computing for Development', ACM DEV '10, ACM, New York, NY, USA, pp. 24:1-24:10.
URL: <http://doi.acm.org/10.1145/1926180.1926210>
- Kyrki, V. & D. Kragic (2011), 'Computer and robot vision [tc spotlight]', *Robotics Automation Magazine, IEEE* **18**(2), 121-122.
- Ladwa, T.M., S.M. Ladwa, R.S. Kaarthik, A.R. Dhara & N. Dalei (2009), Control of remote domestic system using dtmf, em 'Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2009 International Conference on', pp. 1-6.
- Lauwers, Tom & Illah Nourbakhsh (2010), Designing the finch: Creating a robot aligned to computer science concepts, em 'AAAI Symposium on Educational Advances in Artificial Intelligence'.
URL: <https://aaai.org/ocs/index.php/EAAI/EAAI10/paper/view/1849>
- Lopez, D., R. Cedazo, F.M. Sanchez & J.M. Sebastian (2009), 'Ciclope robot: Web-based system to remote program an embedded real-time system', *Industrial Electronics, IEEE Transactions on* **56**(12), 4791-4797.

- Lowe, G. & A. Cambrell (2002a), Web system for control of mechatronic devices, *em* 'Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on', Vol. 3, pp. 1464 – 1469 vol.3.
- Lowe, Gordon S. & Ashley Cambrell (2002b), Web system for control of mechatronic devices, *em* 'ICARCV', IEEE, pp. 1464–1469.
- Lumsden, J. & C. Ortega-Sanchez (2010), Modular autonomous robotics platform for educational use, *em* 'TENCON 2010 - 2010 IEEE Region 10 Conference', pp. 1577 –1582.
- Lyons, R.G. (2010), *Understanding Digital Signal Processing*, Prentice Hall.
URL: <http://books.google.com.br/books?id=UBU7Y2tpwWUC>
- Maki, Daisuke (2007), 'Concurrent.thread', On-Line. Publicado em 2007. Acesso em Nov/2012. Disponível em: http://sourceforge.net/apps/mediawiki/jstthread/index.php?title=Main_Page.
- Maki, Daisuke & Hideya Iwasaki (2007), 'Javascript multithread framework for asynchronous processing', *IPJS Transactions on Programming* **48**(12), 1–18.
- Manikandan, D., P. Pareek & P. Ramesh (2010), Cell phone operated robot, *em* 'Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010 International Conference on', pp. 183 –184.
- Manseur, R. (1997), Development of an undergraduate robotics course, *em* 'Frontiers in Education Conference, 1997. 27th Annual Conference. 'Teaching and Learning in an Era of Change'. Proceedings.', Vol. 2, pp. 610 –612 vol.2.
- Marin, R., P.J. Sanz, P. Nebot & R. Wirz (2005), 'A multimodal interface to control a robot arm via the web: a case study on remote programming', *Industrial Electronics, IEEE Transactions on* **52**(6), 1506 – 1520.
- Martins, Felipe N., Hudson C. Oliveira & Gabriela F. Oliveira (2012), Robótica como meio de promoção da interdisciplinaridade no ensino profissionalizante, *em* 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Martins, Felipe Nascimento (2012), 'Prêmio de melhor robô do mundo é de alunos do ifes!', On-Line. Publicado em Jul/2012. Acesso em Nov/2012. Disponível em: <http://nera.sr.ifes.edu.br/?p=143>.
- McComb G.; Predko, M. (2006), *The Robot Builder's Bonanza*, McGraw-Hill: New York, NY, USA, 2006.
- Menezes, Ebenezer Takuno de & Thais Helena dos. Santos (2002), 'Robótica educacional (verbete)', Dicionário Interativo da Educação Brasileira - EducaBrasil. São Paulo: Midiamix Editora, On-Line. Acesso em Out/2012. Disponível em: <http://www.educabrasil.com.br/eb/dic/dicionario.asp?id=49>.

- Michaelis, Lutz, Edward Jiang, Sun Wei Ze, Leo Yan & Spencer Featherston (2012), 'Xinchejian shanghai hackerspace afron 10 competition submission', On-Line. Acesso em Nov/2012. Disponível em: <http://wiki.xinchejian.com/>.
- Miranda, Juliano R. & Marilza Vanessa R. Suanno (2012), Robótica na escola: ferramenta pedagógica inovadora, em 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Mirats Tur, J.M. & C.F. Pfeiffer (2006), 'Mobile robot design in education', *Robotics Automation Magazine, IEEE* **13**(1), 69 – 75.
- MIT (2012), 'Explore mit app inventor', On-Line. Disponível em: <http://explore.appinventor.mit.edu/>. Acesso em Jul/2012.
- Mod (2012), 'Modelix robotics open source', On-Line. Acesso em Dez/2012. Disponível em: <http://www.modelix.cc/en/>.
- Mok, H. N. (2011), 'Student usage patterns and perceptions for differentiated lab exercises in an undergraduate programming course', *Education, IEEE Transactions on PP*(99), 1.
- Mootien, Saravanen, Robert T. F. Ah King & Harry C. S. Rughooputh (2006), 'A web-based interface for the Gryphon robot', *INTERNATIONAL JOURNAL OF ELECTRICAL ENGINEERING EDUCATION* **43**(2), 134–149.
- Mozos, O.M., Z.-C. Marton & M. Beetz (2011), 'Furniture models learned from the www', *Robotics Automation Magazine, IEEE* **18**(2), 22 –32.
- Nahshon, Itai (1998), 'dtmfdial - dtmf tone generator', On-Line. Desenvolvido em 1998. Acesso em Nov/2012. Disponível em: <http://linux.about.com/cs/linux101/g/dtmfdial.htm>
- Naskar, S., S. Das, A.K. Seth & A. Nath (2011), Application of radio frequency controlled intelligent military robot in defense, em 'Communication Systems and Network Technologies (CSNT), 2011 International Conference on', pp. 396 –401.
- Netto, Antonio Valerio, Felipe Miranda, Wesley Silva & Yan Freitas (2012), A block programming interface for educational mobile robots, em 'Robocontrol: Proceedings of 5th workshop in applied robotics and automation'.
- O.M., Adebola Simeon (2012), 'A locally sourced robotic arm', On-Line. Acesso em Nov/2012. Disponível em: <http://roboarmblog.wordpress.com/>.
- Oracle (2012), 'Sun spot world: Sun small programmable object technology', On-Line. Acesso em Nov/2012. Disponível em: <http://www.sunspotworld.com/>.

- Park, Jong Hyun, Tae Houn Song, Ji Hwan Park & Jae Wook Jeon (2008), Usability analysis of a pda-based user interface for mobile robot teleoperation, *em* 'Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on', pp. 1487–1491.
- Pascual, Alberto (2012a), 'androino: Connect your mobile to the real world objects', On-Line. Acesso em Dez/2012. Disponível em: <http://androino.blogspot.com.br/>.
- Pascual, Alberto (2012b), 'Androino terminal', On-Line. Acesso em Dez/2012. Disponível em: <http://code.google.com/p/androino/wiki/AndroinoTerminal>.
- Patil, B.C. & R. Henry (2008), Dual functional reconfigurable mobile robot, *em* 'TEN-CON 2008 - 2008 IEEE Region 10 Conference', pp. 1–5.
- Pin, Luiza B., Débora F. de Oliveira & Eduardo M. A. Amaral (2012), Ludebot: Uma plataforma com servomotores e sensoriamento utilizando a placa arduino e controle externo, *em* 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Pink, Daniel H. (2009), *Drive: the surprising truth about what motivates us*, Riverhead Books, New York.
- Piperidis, S., L. Doitsidis, C. Anastasopoulos & N.C. Tsourveloudis (2007), A low cost modular robot vehicle design for research and education, *em* 'Control Automation, 2007. MED '07. Mediterranean Conference on', pp. 1–6.
- PNC (2012), 'Pnca robótica e eletrônica', On-Line. Acesso em Dez/2012. Disponível em: <http://www.pnca.com.br/>.
- Popescu, D., D. Selisteanu, I. Dinulescu & L.C. Popescu (2008), Web based telematics application for robotics, *em* 'Computing in the Global Information Technology, 2008. ICCGI '08. The Third International Multi-Conference on', pp. 19–24.
- Ranieri, Caetano, Silas Alves, Humberto Ferasoli Filho, Marco Caldeira & Renê Pegoraro (2012), An environment endowed with a behavior-based control architecture to allow physically disabled children to control mobile robots, *em* 'Robocontrol: Proceedings of 5th workshop in applied robotics and automation', pp. 1–6.
- Rawat, K.S. & G.H. Massiha (2004), A hands-on laboratory based approach to undergraduate robotics education, *em* 'Proceedings of IEEE International Conference on Robotics and Automation (ICRA)', pp. 1370–1374 Vol.2.
- Reben, Alexander (2012), 'Proposal for a low-cost robot for education in africa', On-Line. Acesso em Nov/2012. Disponível em: https://docs.google.com/document/pub?id=1wW-45Lohr6KfgCN23bJejrtB462AAtDRKvCuM4a_gE8.

- Reis, Gabriela L., Luis F. F. Souza, Felipe C. T. Carvalho, Marcos A. A. Junior, Erivelton G. Nepomuceno, Márcio F. S. Barroso & Eduardo B. Pereira (2012), As competições universitárias e a carreira profissional do aluno de graduação: Um estudo de caso sobre a equipe uairobots-sek, *em* 'Anais do Workshop de Robótica Educacional (Latin American Robotics Symposium / Simpósio Brasileiro de Robótica). Fortaleza. ISSN 978-85-7669-261-4'.
- Resnick, Mitchel (2007), 'Scratch: imagine, program, share', On-Line. Acesso em Dez/2012. Disponível em: <http://scratch.mit.edu/>.
- Ric (2011), 'All for one, 'r-one' for all', On-Line. Disponível em: <http://engineering.rice.edu/NewsContent.aspx?id=3491>. Acesso em Jul/2012.
- Rob (2012a), 'Robokit - kit de robótica educativa', On-Line. Acesso em Dez/2012. Disponível em: <http://www.imply.com.br/pgs-inform/prtedu-robokit.htm>.
- Rob (2012b), 'Serial on android using the audio port', On-Line. Acesso em Nov/2012. Disponível em: http://robots-everywhere.com/re_wiki/index.php?title=Serial_on_Android_using_the_audio_port.
- Robinson, Andrew, Brad Campbell, Ye-Sheng Kuo, Sonal Verma, Thomas Schmid & Prabal Dutta (2010), 'Project hijack', On-Line. Acesso em Out/2012. Disponível em: <http://web.eecs.umich.edu/~prabal/projects/hijack/>.
- Romotive (2011), 'Romotive', On-Line. Acesso em Out/2012. Disponível em: <http://www.romotive.com>.
- Rubenstein, Michael, Christian Ahler, & Radhika Nagpal (2012), 'Kilobot entry for afron 10 dollar robot challenge', On-Line. Acesso em Nov/2012. Disponível em: <https://sites.google.com/site/afron10dollarrobot/>.
- Sa, In-Kyu, Ho Seok Ahn, Kwang Moo Yi & Jin Young Choi (2009), Implementation of home automation system using a pda based mobile robot, *em* 'Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on', pp. 1761 –1766.
- Sai, K.V.S. & R. Sivaramakrishnan (2009), Design and fabrication of holonomic motion robot using dtmf control tones, *em* 'Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009. 2009 International Conference on', pp. 1 –4.
- Santos, André C., Luís Tarrataca & João M. Cardoso (2010), 'The feasibility of navigation algorithms on smartphones using j2me', *Mob. Netw. Appl.* **15**(6), 819–830.
- Sasaki, Yusuke (2011), 'Audio jack modem for iphone', On-Line. Acesso em Dez/2012. Disponível em: <http://code.google.com/p/arms22/wiki/SoftModemBreakoutBoard>.
- Scaramuzza, D. & F. Fraundorfer (2011), 'Visual odometry [tutorial]', *Robotics Automation Magazine, IEEE* **18**(4), 80 –92.

- Schlesinger, Joseph (2012), 'Afron submission: Discbot', On-Line. Acesso em Nov/2012. Disponível em: <http://arcbotics.com/afron-submission/>.
- SeedStudio (2011), 'Hijack development pack', On-Line. Acesso em Jul/2012. Disponível em: <http://www.seedstudio.com/depot/hijack-development-pack-p-865.html?cPath=174>.
- Seid, Peter, Keller Rinaudo & Phu Nguyen (2012), 'Romo - the smartphone robot for everyone', On-Line. Publicado em Out/2012. Acesso em Nov/2012. Disponível em: <http://www.kickstarter.com/projects/peterseid/romo-the-smartphone-robot-for-everyone>.
- Siegwart, R. & R.S.I.R. Nourbakhsh (2004), *Introduction to Autonomous Mobile Robots*, Intelligent Robots and Autonomous Agents Series, Mit Press.
URL: http://books.google.com.br/books?id=gUbQ9_weg88C
- Singh, Karanpreet, Amitesh Ranjan & Pawan Saini (2012), 'Mitbots educational proposal for afron \$10 robot design challenge', On-Line. Acesso em Nov/2012. Disponível em: http://mitbots.com/proposal_3/.
- Siu, Jason (2012), 'Carbot remote controlled cars work off your smartphone', On-Line. Publicado em Ago/2012. Acesso em Nov/2012. Disponível em: <http://www.autoguide.com/auto-news/2012/08/carbot-remote-controlled-cars-work-off-your-smartphone-video.html>.
- Soto, Alvaro, Pablo Espinace & Ruben Mitnik (2006), A mobile robotics course for undergraduate students in computer science, em '3rd IEEE Latin American Robotics Symposium (LARS)', pp. 187-192.
- Sousa, Armando, Catarina Santiago, Paulo Malheiros, Paulo Costa & António Paulo Moreira (2009), Using barcodes for robotic landmarks, em 'Fourteenth Portuguese Conference on Artificial Intelligence', Aveiro (Portugal).
- Taylor, Darrell (2011), 'Super-simple audio cellbot', Make Projects. On-Line. Access date: March 09 2012. URL: <http://makeprojects.com/Project/Super-Simple-Audio-Cellbot/864/1>.
- Teleco (2011), 'Estatísticas de domicílios brasileiros (ibge - pnad)', On-Line. Disponível em: <http://www.teleco.com.br/pnad.asp>. Acesso em Jul/2012.
- Teleco (2012a), 'Estatísticas de celulares no brasil', On-Line. Disponível em: <http://www.teleco.com.br/ncel.asp>. Acesso em Jul/2012.
- Teleco (2012b), 'Estatísticas do brasil - geral', On-Line. Disponível em: <http://www.teleco.com.br/estatis.asp>. Acesso em Jul/2012.
- The (2012), 'Lego.com mindstorms', On-Line. Acesso em Dez/2012. Disponível em: <http://mindstorms.lego.com/en-us/Default.aspx>.

- Thomaz, Sarah, Carla Fernandes, Lucas Cavalcante & Luiz Gonçalves (2012), H-educ: A low cost platform for educational robotics, *em* 'Robocontrol: Proceedings of 5th workshop in applied robotics and automation', pp. 1–6.
- Tilley, Tom (2012), 'Suckerbot', On-Line. Acesso em Nov/2012. Disponível em: <http://www.tomtilley.net/projects/suckerbot/>.
- Tofel, Kevin C. (2011), 'Robots rise with phone sensors, android and the cloud', GigaOM Mobile Technology News. On-Line. Access date: March 09 2012. URL: <http://gigaom.com/mobile/google-android-robots-rosjava-smartphones/>.
- van Bosse, J.G. & F.U. Devetak (2006), Signaling in Telecommunication Networks, *2nd ed.*; *Wiley Series in Telecommunications and Signal Processing*, 2nd^a edição, Wiley-Interscience: Hoboken, NJ, USA.
- Vex (2012), 'Vex robotics', On-Line. Acesso em Dez/2012. Disponível em: <http://www.vexrobotics.com/vex>.
- Weinberg, J.B. & Xudong Yu (2003), 'Robotics in education: Low-cost platforms for teaching integrated systems', *IEEE Robot. Automat. Mag.* **10**(2), 4–6.

Apêndice A

Questionários

A.1 Questionário Prévio

Prezado aluno, estamos conduzindo uma breve pesquisa sobre conhecimentos e interesses em robótica e engenharia. Agradecemos muito sua ajuda em preencher este questionário.

1. Em qual universidade você estuda?
2. Qual o campus da universidade que você estuda?
3. Qual curso de graduação você está fazendo?
4. Este questionário está sendo respondido no contexto de qual disciplina?
5. Qual sua idade?
6. Qual o seu sexo?
 - (a) Masculino
 - (b) Feminino
7. Em qual ano você começou a graduação que esta fazendo atualmente?
8. Você já desenvolveu software para dispositivos móveis (celulares, tablets, etc)?
 - (a) Sim
 - (b) Não
9. Você sabe o que é ensinado na disciplina Processamento Digital de Sinais (PDS)?
 - (a) Sim
 - (b) Não
10. Você cursaria a disciplina Processamento Digital de Sinais?
 - (a) Sim
 - (b) Não
11. Você sabe o que é ensinado na disciplina visão computacional?
 - (a) Sim
 - (b) Não
12. Você cursaria a disciplina visão computacional?
 - (a) Sim

- (b) Não
13. Você sabe como funciona um servidor Web?
- (a) Sim
(b) Não
14. Você sabe o que é ensinado na disciplina redes de computadores?
- (a) Sim
(b) Não
15. Você cursaria redes de computadores?
- (a) Sim
(b) Não
16. Para fazer testes de robótica nas disciplinas, você prefere usar um kit de robótica da universidade, comprar um robô pronto, ou construir seu próprio robô?
(No caso do kit da universidade, após o uso, ele permanece na universidade. Nos outros casos, você fica com o robô)
- (a) Usar kit da universidade
(b) Comprar robô pronto
(c) Construir meu próprio robô
17. No caso de comprar um robô pronto ou construir seu próprio robô, quanto você gastaria do seu próprio dinheiro?
- (a) Não gastaria
(b) Gastaria até R\$ 50,00
(c) Gastaria até R\$ 100,00
(d) Gastaria até R\$ 250,00
(e) Gastaria até R\$ 500,00
(f) Gastaria até R\$ 1000,00
(g) Gastaria mais de R\$ 1000,00
18. Você sabe o que é e para que servem encoders, drivers de potência e acelerômetros?
- (a) Sim para todos
(b) Sim, mas apenas sobre um deles
(c) Sim, mas apenas sobre dois deles
19. Você conhece programação orientada a objetos?
- (a) Sim, e já usei
(b) Já ouvi falar
(c) Não conheço
20. Você tem celular? Quando foi comprado?
- (a) Não tenho
(b) Sim, comprado a menos de 6 meses
(c) Sim, comprado a menos de 1 ano
(d) Sim, comprado a menos de 2 anos

- (e) Sim, comprado a menos de 4 anos
21. Quais recursos seu celular tem?
- (a) Câmera
 - (b) GPS
 - (c) Acelerômetro
 - (d) Wifi
 - (e) Bluetooth
 - (f) Conexão para fones de ouvido externos
22. Quais disciplinas você acha importantes para aprender robótica?
- (a) Processamento digital de sinais
 - (b) Controle
 - (c) Introdução a robótica
 - (d) Álgebra linear
 - (e) Visão computacional
 - (f) Redes neurais
 - (g) Estatística
 - (h) Processos estocásticos
 - (i) Cálculo
 - (j) Programação orientada a objetos
 - (k) Programação web
 - (l) Redes de computadores
 - (m) Sistemas distribuídos
 - (n) Sistemas operacionais
 - (o) Sistemas de tempo real

A.2 Questionário Posterior

Importante: Prezado aluno, por favor, somente responda este questionário se você assistiu à apresentação ou vídeo “Construa seu próprio robô”, que apresentou uma forma de construir sistemas robóticos poderosos por menos de R\$ 50,00 usando um celular como computador de controle. Agradecemos muito sua ajuda em preencher este questionário.

1. Em qual universidade/campus você estuda?
2. Qual curso de graduação você está fazendo?
3. Este questionário está sendo respondido em qual disciplina?
(Questões de 4 a 9: marque apenas uma alternativa)
4. Você cursaria uma disciplina sobre Processamento Digital de Sinais?
 - (a) Sim
 - (b) Não
 - (c) Já cursei
5. Você cursaria uma disciplina sobre visão computacional?
 - (a) Sim

- (b) Não
(c) Já cursei
6. Para fazer testes de robótica nas disciplinas, você prefere usar um kit de robótica da universidade, comprar um robô pronto, ou construir seu próprio robô?
(No caso do kit da universidade, após o uso, ele permanece na universidade. Nos outros casos, você fica com o robô)
- (a) Usar kit da universidade
(b) Comprar robô pronto
(c) Construir meu próprio robô
7. No caso de comprar um robô pronto ou construir seu próprio robô, quanto você gastaria do seu próprio dinheiro?
- (a) Não gastaria
(b) Gastaria até R\$ 50,00
(c) Gastaria até R\$ 100,00
(d) Gastaria até R\$ 250,00
(e) Gastaria até R\$ 500,00
(f) Gastaria até R\$ 1000,00
(g) Gastaria mais de R\$ 1000,00
8. Com relação as opções de “CellBots” de baixo custo apresentadas, você se interessou pelo projeto?
- (a) Sim, e já montei um
(b) Sim, e estou montando um
(c) Sim, e pretendo montar um futuramente
(d) Sim, mas não pretendo montar um CellBot
(e) Não
9. Você acha que uma disciplina fica mais interessante com o uso do CellBot como uma forma de testar os conceitos ensinados nesta disciplina?
- (a) Sim
(b) Não
- (Questões 10 e 11: Marque todas alternativas que se aplicam)
10. Qual(is) aspecto(s) de um “CellBot” você acha mais interessante(s) para trabalhar?
- Projeto e melhorias da parte mecânica
 - Construção e montagem da parte mecânica (chassis, rodas, motores)
 - Projeto e melhorias da parte eletrônica
 - Construção e montagem da parte eletrônica (placas, cabos, circuitos, sensores)
 - Desenvolvimento de softwares e novos sistemas de controle
11. Quais disciplinas você acha importantes para aprender robótica?
- Processamento digital de sinais
 - Controle
 - Introdução a robótica

- Álgebra linear
 - Visão computacional
 - Redes neurais
 - Estatística
 - Processos estocásticos
 - Cálculo
 - Programação orientada a objetos
 - Programação web
 - Redes de computadores
 - Sistemas distribuídos
 - Sistemas operacionais
 - Sistemas de tempo real
12. Comentário final (opcional)
(Se julgar necessário, deixe um comentário, crítica ou sugestão sobre CellBots e suas possibilidades)

A.3 Avaliação do mini-curso de robótica - SINEC 2012

Queremos melhorar o mini-curso de robótica de baixo custo para as próximas edições. Agradecemos sua colaboração em responder as perguntas desta breve pesquisa. Muito obrigado.

1. Antes de fazer o mini-curso, você achava possível construir um robô com menos de R\$ 50,00?
 - (a) Sim
 - (b) Não
2. Sobre a duração do mini-curso, você achou que foi:
 - (a) Longa
 - (b) Adequada
 - (c) Curta
3. Se o mini curso fosse realizado novamente, você recomendaria para algum colega?
 - (a) Sim
 - (b) Não
4. Você já havia montado algum circuito eletrônico antes?
 - (a) Sim
 - (b) Não
5. Você conseguiu terminar a montagem de seu robô (placa, e movimento dos motores)?
 - (a) Sim
 - (b) Não

6. Se não terminou o seu robô, qual a porcentagem de finalização do mesmo em que você considera ter chegado?
(Exemplo: fiz 70% do robô)
7. Você acha que seria interessante usar este robô durante aulas de graduação?
 - (a) Sim
 - (b) Não
8. Qual sua avaliação geral do mini-curso?
 - (a) Muito Bom
 - (b) Bom
 - (c) Regular
 - (d) Ruim
 - (e) Muito Ruim
9. Você está fazendo graduação em qual curso?
(Exemplos: ECT, Engenharia da Computação, Ciência da Computação, Engenharia Elétrica, etc)
10. Caso você queira deixar algum comentário ou crítica, fique à vontade para digitar na seguinte caixa de texto:
(espaço para escrita livre)