ciência da computação

V EPAC - Encontro Paranaense de Computação ISSN:1981-8653
18 a 20 de setembro de 2013
Unioeste – Cascavel – Paraná

CONTROLE DE ROBÔ LEGO® MINDSTORMS NXT POR COMANDOS DE VOZ UTILIZANDO MATLAB®

Adriana Postal[1], Diego Henrique Pagani[2], Gustavo Henrique Paetzold[3]; Josué Pereira de Castro[4]
[1,2,3,4] Universidade Estadual do Oeste do Paraná – Unioeste – Cascavel – Paraná
{adriana.postal, diego.pagani, gustavo.paetzold, josue.castro}@unioeste.br

Resumo
O objetivo deste trabalho é apresentar um sistema de controle por voz para o Lego® Mindstorms® NXT 2.0. A implementação do sistema foi feita em MATLAB® devido a sua robustez e a existência de várias bibliotecas com suporte a processamento de sinais. Nos testes realizados com o sistema, a taxa mínima de acerto foi de 49% e como taxa máxima de 96%, segundo a metodologia adotada, Ao final deste resumo sugere-se maiores estudos, testes e que podem ser realizados para melhorar o desempenho do sistema.
Palavras-chave: controle de robô, comandos de voz, Escala MEL.

1. Introdução
Com a tecnologia atual, a interatividade dos equipamentos eletrônicos vem crescendo bastante. Com cada vez mais poder computacional, smartphones, video-games, televisores e outros dispositivos vem apresentando ao mercado inovações como reconhecimento de movimento, de toque, ou de voz. Visando um aumento da interatividade, e até mesmo segurança, smartphones vêm apresentando softwares de reconhecimento de fala que reconhecem o usuário sem que o mesmo precise tocar na tela.

Este trabalho apresenta a implementação de um sistema de reconhecimento de comandos de voz para o Lego® Mindstorms® NXT 2.0 (LEGO, 2013). O sistema foi desenvolvido na plataforma MATLAB®, e estabelece a conexão entre computador e robô através da biblioteca RWTH-Mindstorms NXT *Toolbox* (RWTH, 2013) via Bluetooth®.

Optou-se pela implementação em MATLAB® porque o este software contém uma série de bibliotecas, como a *Toolbox Signal Analysis* que implementa todas as funções básicas de processamento de sinais, e também por ser uma linguagem interpretada, o que torna o processo de prototipação de sistemas mais rápido, além de facilitar a descoberta de erros no código, e também permite a realização de testes em tempo real através de seu *console*.

2. Sistema de Reconhecimento de Comandos de Voz
O sistema de reconhecimento de comandos de voz desenvolvido neste trabalho é dividido em módulos de treinamento e de testes. Cada módulo é responsável por realizar uma parte específica do sistema. Segue abaixo a descrição dos módulos:

Módulo de treinamento: Nesta fase, o módulo de treinamento é acionado com o objetivo de capturar múltiplos segmentos de áudio com a voz do locutor, para que estes

sejam então manualmente classificados com algum dos possíveis comandos reconhecidos pelo robô. O objetivo desta fase é produzir uma base de conhecimento para que o sistema reconheça corretamente os comandos de voz na etapa de testes.

Fase de testes: Quando iniciada, esta fase aciona o módulo de testes para que este adquira um segmento de voz do usuário, classifique-o com referência à base de conhecimento construída anteriormente e então direcione o comando ao robô.

2.1. Fase de Treinamento

O objetivo da fase de treinamento é construir a base de conhecimento que será utilizada como referência na classificação de comandos de voz na etapa de testes.

Para construir esta base são coletadas múltiplas amostras de áudio de um locutor para que estas sejam então pré-processadas e classificadas manualmente quanto ao comando que representam. Este processo é repetido para cada comando que será reconhecido futuramente. Pode ser coletada qualquer quantidade de amostras de áudio de referência para cada comando. Estes dados são então armazenados em uma matriz e posteriormente direcionados à fase de testes.

O pré-processamento aplicado pelo módulo de treinamento produz o espectro de frequências da escala MEL (STEVENS et al, 1937) de cada um dos trechos de áudio adquiridos. A escala MEL descreve o conjunto de tons que são distinguíveis pelo ouvido humano. Esta escala é construída com base na observação de que os seres humanos classificam certos grupos de frequências em um mesmo tom, ou seja, o sistema auditivo humano é incapaz de distinguir a diferença entre sons cujas frequências sejam minimamente distintas (LANNERER, 2005).

O espectro de frequências (BOUALEM, 1992) é representado por uma matriz V de dimensões MxN. Cada uma das M linhas do espectro representa um tom de áudio, enquanto as N colunas representam a dimensão de tempo, ou seja, cada coluna corresponde a um momento do trecho de áudio capturado. Um valor V(i, j) da matriz do espectro de frequências MEL corresponde ao nível de energia presente no tom “i” no momento “j” de um dado trecho de áudio.

Finalmente, o espectro de frequências MEL de cada trecho de áudio é então classificado manualmente como algum dos comandos reconhecíveis pelo robô. Por exemplo, um trecho de áudio onde o locutor disse “Ande” deverá ser classificado como um representante do comando que leva o robô andar.

2.2 Fase de testes

A fase de testes permite que o usuário interaja com o robô através de comandos de voz. Nesta fase o usuário deve primeiramente fornecer ao módulo de testes uma amostra de sua fala que represente algum dos comandos suportados pelo robô. É importante que o usuário forneça uma amostra de fala tão similar quanto possível às amostras coletadas na fase de treinamento.

Em seguida, a amostra de áudio coletada passa pela mesma forma de pré-processamento empregada na fase de treinamento, que produz o espectro de frequências

MEL da mesma. O próximo passo da fase de testes é decidir a que comando a amostra de áudio de testes se refere. A classificação da amostra de testes é feita através do algoritmo de Levenshtein: o comando ao qual a amostra se refere é o comando referente à entrada da base de conhecimento com a qual a amostra de testes sofre menos alterações (SAMWORTH, 2012).

O comando referente à amostra de áudio de testes é então direcionado ao robô através de uma conexão via *Bluetooth*® (BLUETOOTH, 2013) entre o robô Lego® e o computador de onde o sistema de reconhecimento de voz está sendo executado.

3. Configuração dos Experimentos

No objetivo de descobrir quão preciso é o sistema de reconhecimento de comandos de voz proposto neste trabalho, conduziu-se alguns experimentos envolvendo dois locutores do sexo masculino, com três comandos: "Pare", "Ande" e "Gire". Para cada locutor, foi elaborada uma bases de dados com 10 amostras de cada comando, totalizando 2 bases (Base A do Locutor 1 e Base B do Locutor 2). Para realizar os testes, com cada um dos 2 locutores, definiu-se que cada um deverá amostrar 33 exemplos de voz para cada comando e com isso realizou-se a análise, comparando a base de dados e as amostras para verificar a eficiência deste método.

Na captura dos comandos, definiu-se que a captura de som seria realizada em apenas um canal (Mono), com 16 bits e com 11000 Hz.

4. Resultados obtidos

A tabela 1 mostra os resultados obtidos no experimento, no qual percebe-se que o Locutor 1 obteve melhores taxas de acerto, se comparados ao Locutor 2. Pode-se considerar as taxas do Locutor 1 como sendo ótimas. Ao juntar ambas as bases, não houve melhora na taxa de acertos, porém a do Locutor 2 aumentou em 3 pontos.

	Base A	Base B	Base A+B
Locutor 1	96%	85%	96%
Locutor 2	49%	72%	75%
Média geral	72,5%	78,5%	86%

Tabela 1 – Taxa de acerto do sistema de reconhecimento de voz

A diferença que ocorre entre os locutores (quando testadas com sua própria base) pode ter ocorrido devido a mudança de microfone, qualidade do som obtido pelo mesmo, ruídos externos, diferenças entre os tons de vozes coletados na amostra e na base e o sistema privilegiar determinadas características da voz, mas necessita de mais testes para confirmar.

5. Considerações Finais

Os testes realizados são promissores, mas o sistema ainda necessita de mais ajustes e testes. O sistema de classificação utilizando o algoritmo de Levenshtein terá resultados bons se a

amostra de voz for semelhante a do banco de dados, que poderia tornar o sistema incapaz de reconhecer comandos de uma voz analiticamente diferente das cadastrada. Outro ponto negativo do algoritmo, é que quanto maior a base, maior o tempo necessário para o algoritmo executar. Outro fator que deve ser revisto, é a da captura de som, padronizada para a geração da base de dados e nos testes. Analisar as diferenças do som capturado com diferentes equipamentos se tornaria necessário caso exista necessidade de tornar o sistema mais genérico.

Referências bibliográficas

BLUETOOTH SIG, Inc. **What is Bluetooth technology**. Disponível em: http://www.bluetooth.com/Pages/what-is-bluetooth-technology.aspx. Acessado em: 14 de Julho de 2013.

BOUALEM, B. Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals. Proceedings of the IEEE. Volume 4. p. 520-538. 1992.

LANNERER, B. An **Introduction to Speech Recognition**, 2005. Disponível em: <http://www.speech-recognition.de>. Acesso em: 27 jul. 2013.

LEGO MINDSTORMS®. **What is NXT**. Disponível em: http://mindstorms.lego.com/en-us/whatisnxt/default.aspx. Acessado em 14 de Julho de 2013.

RWTH Aachen University. **RWTH - Mindstorms NXT Toolbox for MATLAB**. Disponível em: http://www.mindstorms.rwth-aachen.de/. Acessado em 14 de Julho de 2013.

SAMWORTH, R.J. **Optimal weighted nearest neighbour classifiers**. Annals of Statistics 2012, Volume 40. p. 2733-2763. 2012.

STEVENS, S.S.; VOLKMANN, J.; NEWMAN, E.B. **A Scale for the Measurement of the Psychological Magnitude Pitc**h. Harvard University, Cambridge, Massachusetts. Acoust. Soc. Am. Volume 8. p. 185-190. 1937.