Interface de Voz para Controle de Robôs

Gabriel F. P. Araújo
Laboratório de Automação e Robótica - LARA
Universidade de Brasília - UnB
Brasília - DF - Brasil

***Resumo*—Este documento tem como objetivo relatar o trabalho realizado sobre o tema "Interface de Voz para Controle de Robôs" durante o ano de pesquisa 2014/2015. O trabalho foi dividido em três módulos, reconhecimento de fala, processamento de palavras e síntese de voz. Os módulos foram implementados usando o ROS para comunicação.**

***Index Terms*—Processamento de Linguagem Natural, Sistema de Audição Robótico, Síntese de Áudio, Reconhecimento de Fala.**

I. Introdução

A história marca o seu próprio começo quando os humanos desenvolvem a escrita. Contudo, muito antes dessa nascer, o Homo Sapiens já utilizava a fala para se comunicar. É inegável a importância da língua falada, não apenas comunicação humana, mas também para a construção de civilizações.

Portanto, a capacidade humana de reconhecer sons é essencial tanto para a comunicação quanto para a interação social. Dessa forma, robôs podem ter essa mesma capacidade usando reconhecimento sonoro. Assim, o potencial humano, de entender e responder aos estímulos sonoros provenientes do ambiente, deve ser incorporado. Como reconhecer uma variedade de sons em diversos meios, localizar quem está falando, saber executar ações em resposta.

Como ponto de partida para um trabalho inicial com robótica e reconhecimento de áudio, foi escolhido como tema controle de robôs usando fala. Pois, espera-se que, no futuro, robôs sejam usados para ajudar diariamente os humanos em ambientes variados, desde domésticos a industriais. O trabalho foi dividido em três partes, reconhecimento de fala, processamento de palavras e síntese de voz.

Este relatório tem como objetivo documentar o trabalho do autor no desenvolvimento de um sistema de reconhecimento de fala e síntese de voz para o controle de robôs. Na implementação do sistema foi usado diversas ferramentas desenvolvidas por grupos de pesquisa de universidade de várias partes do mundo. O sistema foi testado no robô Aramis do grupo de robótica móvel, AMORA, do Laboratório de Robótica e Automação, LARA, da Universidade de Brasília, e em trabalhos futuros deve ser estendido para os demais robôs do grupo.

O resto do relatório está segmentado da seguinte forma: a seção II contém as especificações de hardware, tanto do original quanto dos acessórios, em que o sistema foi implementado. A seção III apresenta as ferramentas computacionais usadas neste projeto, para que o trabalho possa ser reproduzido, e o porquê de segmentar o trabalho em módulos. A seção IV apresenta a implementação dos módulos do sistema e como eles se relacionam com o meio externo e entre si. Já a seção V apresenta e discute os resultados, VI conclusão do trabalho e VII agradecimentos finais.

II. Hardware Utilizado

O uso real de um sistema robótico tem que ser implementado e testado em robôs reais ou simulados. Neste trabalho foi utilizado um robô real comprado pelo Laboratório LARA e customizado para o uso da interface desenvolvida.

A. Plataforma Móvel

Para a implementação do sistema foi necessário um robô que tivesse um hardware em que o sistema pudesse rodar. Foi utilizado o robô diferencial de duas rodas Aramis, figura 1, da família Pioneer modelo P3-DX da Mobile Robots, foi usado para os testes, ele tem a seguinte especificação:

- Processador Intel Pentium M 1.8 GHz;
- Memória RAM de 1 GB;
- SSD de 60 GB da Kingston;
- Sistema Operacional Linux Ubuntu 12.04 LTS de 32 bits;
- ROS Hydro Medusa.

Além do Hardware original de fábrica o Aramis detém os seguintes componentes adicionais:

- Câmera digital IEEE-I394 Unibrain Fire-i400;
- Microsoft Kinect 1.0;
- Caixa de som estéreo USB Clone;
- Estrutura Metálica customizada para suporte.

B. Kinect

O Kinect é um sensor de movimentos para vídeo games desenvolvido para o Xbox 360 e o Xbox One, em uma parceria da Microsoft e Prime Sense. O Kinect permite que o jogador possa interagir com o jogo sem tem em mãos um controle, tem cerca de 23 cm de comprimento e 4 recursos principais:

- Câmera RGB;
- Sensor de profundidade;
- Conjunto de quatro Microfones;

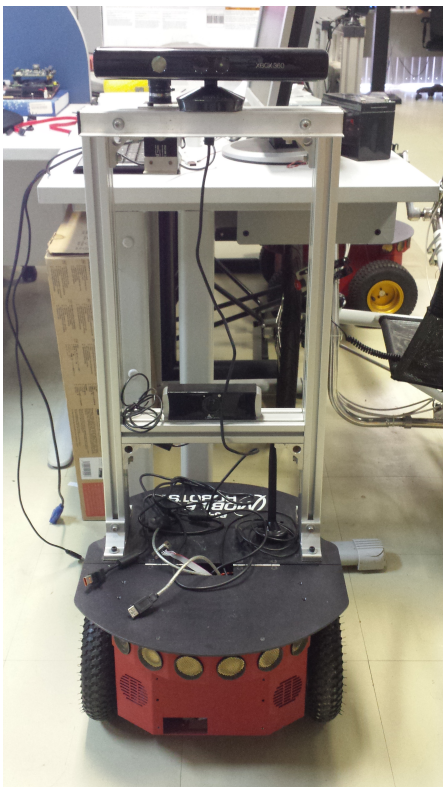

Figura 1: Robô Móvel P3-DX (Aramis)

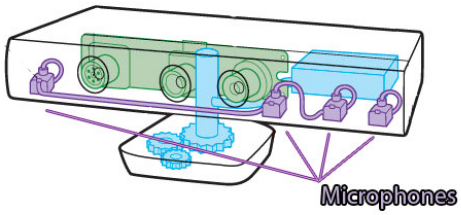

Figura 2: Esquema do arranjo espacial do microfones do Kinect

- Conjunto de motores para mover as câmeras.

Foi lançado em 2010 em uma feira de jogos eletrônicos, a E3. O Kinect foi bastante aclamado pelos jogadores e também pela comunidade científica, pois ele possui um dos melhores sensores de profundidade do mercado. A figura 2 mostra a disposição espacial dos microfones do Kinect, na imagem é possível ver que três deles são bastante próximos e são todos colineares.

III. Software Utilizado

Uma pessoa reconhece sons em vários ambientes onde sons de muitas alturas são ouvidos, processa eles para se comunicar com pessoas e para aproveitar TV, música ou filmes. Um sistema de audição robótico que detém tal funcionalidade/nível de reconhecimento precisa processar áudios de vários ambientes e de vários níveis, isso não pode ser definido facilmente, similar em visão computacional.

Assim, como o OpenCV, software de processamento de imagens open source, é um agregado de módulos de processamento, um sistema de audição robótico precisa consistir de um agregado que inclui o minimo de funções requiridas. A discussão sobre quais funções devem ser unidas para completar esse sistema pode chegar a temas desde técnicos, a necessidade de implementação de certos módulos, a filosóficos, a necessidade de imitar o ser humano na comunicação. Então, o autor decidiu dividir o trabalho em três partes seguindo o tema, para controlar um robô por meio de voz existe a necessidade de reconhecer os comandos falados ao robô (1), processar o comando (2) - decidir o que ser feito - e devolver uma resposta ao comandante (3), a tarefa foi, está ou será executada.

A. *HARK Kinect*

HARK[1] [1] é uma biblioteca de processamento de sinais, tal como o OpenCV, otimizada para áudio, desenvolvida pela Universidade de Kyoto. Os desenvolvedores do HARK seguem a seguinte filosofia:

- Provisão de funções processamento de sinais para áudio;
- Ser flexível à forma do robô;
- Suporte a sistemas multi canais A/D;
- Processamento em Tempo Real.

HARK possui drivers que facilitam o uso de hardwares presentes no mercado, como o Kinect. Neste trabalho foi usado um componente do HARK, o HARK Kinect, um driver que auxilia o uso dos microfones do Kinect. Os microfones são montados como uma placa de captura de áudio com quatro canais.

B. *PocketSphinx*

O Sphinx[2] é uma biblioteca de reconhecimento de áudio desenvolvida pela Universidade Carnegie Mellon. O processo de reconhecimento usado pelo CMU Sphinx tem duas partes:

- **Split** Separa cada áudio em partes menores, por silêncio.
- **Matching** Cada áudio é equiparado com todas as possíveis palavras.

Começa separando o sinal de áudio em falas menores, então tenta-se reconhecer o que é dito em cada fala. Para isso, todas as possíveis palavras são comparadas com o áudio, assim melhor combinação é escolhida.

Na fase de Matching é usado o conceito de modelo matemático da fala, aqui apenas modelo. Um modelo descreve matematicamente alguns atributos da linguagem falada. O modelo

[1] http://winnie.kuis.kyoto-u.ac.jp/HARK/
[2] http://cmusphinx.sourceforge.net/

usado pelo Sphinx é HMM, Hidden Markov Model, é um modelo genérico que descreve uma comunicação, este modelo descreve uma sequencia de estados que muda de um para o outro com uma certa probabilidade. O modelo da língua inglesa provido pelo CMU Sphinx foi usado neste trabalho.

C. Festival

Festival[3][2] é uma biblioteca desenvolvida pela Universidade de Edimburgo para síntese de áudio.

Festival é uma biblioteca C++, multilíngue que provê a síntese de falas em inglês, britânico e americano, espanhol e galês. Implementa funções de alto nível, simples e práticas, para serem acopladas em outros códigos e sistemas. A biblioteca Festival também oferece uma interface cliente-servidor que permite que outros programas acessem as funcionalidades dessa, inclusive programas não escritos em C++. A língua escolhida para reprodução foi o inglês, para documentação da biblioteca veja [3].

D. ROS

ROS[4] [4], é uma abreviação em inglês para *Robot Operating System*, é um *framework* flexível para o desenvolvimento de *software* para robôs. É uma coleção de ferramentas, bibliotecas e convenções que ajudam a simplificar o trabalho de criação de um complexo e robusto comportamento robótico por meio de uma variedade de plataformas.

O ROS foi projetado para ser o mais distribuído e modular possível, usando um sistema de pacotes que são compilados separadamente, assim a inserção de novos pacotes é bastante simples. A infraestrutura do ROS contem uma interface para comunicação entre processos, que tem as facilidades de:

- publicar e assinar anonimamente tópicos de mensagens;
- gravação e reprodução das mensagens;
- pedir e responder chamadas de procedimentos remotos;
- sistema de parâmetro distribuído.

O projeto foi desenvolvido usando a versão Hydro Medusa do ROS, porém o projeto já foi atualizado para a versão Indigo Igloo que é a última versão mais estável do ROS, ambas as versões são open source, portanto a sua utlização é ampla e gratuita. O código do projeto está disponível no repositório do LARA[5] e seu uso e mmodificação também é gratuito.

IV. Implementação do Sistema

O sistema foi separado em três parte que foram implementadas usando as ferramentas apresentadas. O primeiro módulo a ser implementado foi o de reconhecimento de voz, depois o processamento das palavras e por fim a síntese de áudio. Pode-se notar que como o Kinect foi reconhecido como uma placa de áudio, foi possível usar os funções do sistema operacional para adquirir o sinal dos microfones.

[3]http://www.cstr.ed.ac.uk/projects/festival/
[4]http://ros.org/
[5]https://github.com/lara-unb

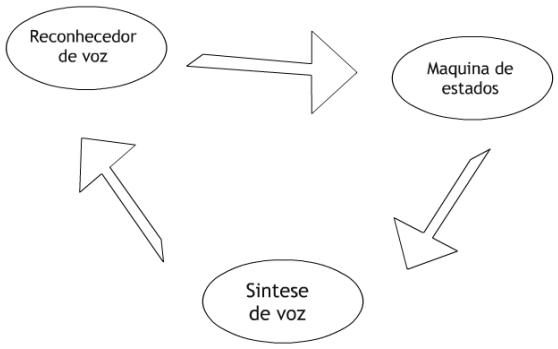

Figura 3: Grafo da Comunicação dos Módulos.

A. Decodificador de Fala

Para o reconhecimento de áudio foi implementado uma classe em C++ para decodificar o sinal proveniente dos microfones. Essa classe, ver o algoritmo 1, verifica continuamente os bufferes dos microfones procurando algum sinal que não seja ruído. Quando um sinal é reconhecido como uma fala, o áudio é guardado em um buffer auxiliar até que um silêncio com duração de um segundo seja capturado, isso marca o fim de uma fala. Após o silêncio, o buffer contendo o áudio falado é decodificado, ao fim da busca, o texto reconhecido é mandado para o próximo módulo. Para a busca ser realizada de forma mais eficiente, o estado de busca foi restringido, a apenas as palavras usadas como comandos previamente definidos.

Algorithm 1 Decodificador

```
loop
    espera o começo da próxima fala;
    decodifica a fala até um silêncio de 1 segundo ser observado;
    enviar resultado;
end loop
```

B. Processamento das Palavras

Para o processamento dos comandos foi pensado em algo que pudesse ser reusado e fácil de adicionar novos comandos. Por isso, foi escolhido uma máquina de estados como modelo para o processamento dos comandos. O esquemático da máquina implementada pode ser visto na figura 4.

1) Init: Estado inicial de operação, todos os módulos, modelos são inicializados e carregados.

2) Ack: Do inglês acknowledge, o estado onde o robô espera por algum comando falado.

3) Follow Me: Estado em que o robô reconhece visualmente quem está falando e o segue.

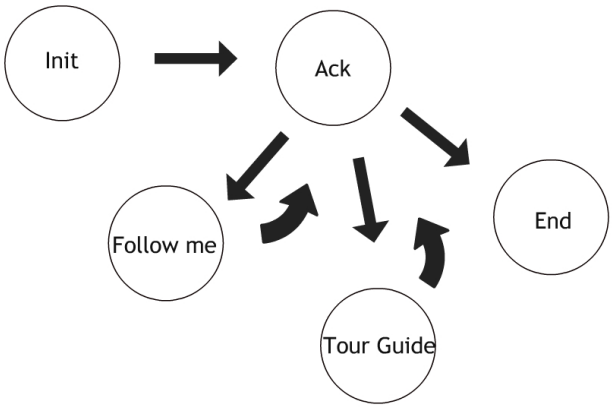

Figura 4: Máquina de Estados Finitos.

4) Tour Guide: Estado guia, o robô vai para um lugar predeterminado no laboratório onde começa a fazer um *tour* pelo laboratório, explicando as partes e locais do laboratório e quais pesquisas são desenvolvidas nelas.

5) End: Este estado descarrega e desativa todos os módulos previamente carregados.

O robô começa no estado Init, onde todos os módulos do robô são inicializados. A partir disso, o estado muda para Ack, o robô espera por algum comando, se alguma fala for reconhecida, o sistema tenta encontrar o comando compatível. Se esse comando existir, então uma resposta é enviada para o módulo de sítese onde é sintetizado o áudio e reproduzido pelas caixas de som. Os estados Follow Me e Tour Guide são módulos a parte do sistema, a máquina envia uma chamada de rotina onde o módulo Follow Me assumirá controle do robô, ao final o estado volta a ser Ack. O mesmo para o estado Tour Guide. O estado End finaliza o sistema e o robô.

Os rotinas dos estados Follow Me e Tour Guide são projetos futuros do grupo AMORA.

C. Síntetizador de áudio

Devido a implementação da biblioteca Festival, o módulo de síntese é o mais simples dos três. Recebe um texto qualquer, do Processamento de Palavras, e envia para a Festival que cuida da síntese e da reprodução. Enquanto o áudio é falado o módulo de reconhecimento é travado, para que este não reconheça o que está sendo falado pelo próprio robô. A implementação do sintetizado oferece suporte para que outros módulos possam enviar mensagens para serem faladas, por exemplo o módulo Tour Guide pode enviar as falas que devem ser faladas durante a apresentação do laboratório.

Os módulos conversam entre si por meio do ROS, mensagens são mandadas de um módulo a outro indicando ações que devem ser executadas e estados em que estão.

V. Resultados

Os módulos foram testados separadamente e depois o sistema foi testado como um todo. O primeiro foi o de reconhecimento de voz, houve dois testes, usando o modelo completo da língua inglesa, com todas as palavras, e outro usando o modelo parcial, apenas as palavras que apareciam nos comandos definidos. Foi possível perceber que existe bastante erro quando o modelo completo da língua inglesa é utilizado, devido ao espaço de busca. Já o segundo experimento teve resultados bastante bons o reconhecedor foi capaz de reconhecer a maioria dos comandos falados pelo autor.

Os testes dos módulos de processamento e de síntese são bastante simples, apenas verificaram a implementação destes. O Máquina de Estados funcionou como deveria, mudando os estados e avisando os outros módulos sobre esses estados. O sintetizador de voz também funcionou como esperado, qualquer palavra ou frase enviada é reproduzida nas caixas de som do robô e o som reproduzido é audível.

Por fim, o sistema como um todo foi testado, os erros de reconhecimento continuaram aparecendo, porém os resultados continuaram bons. A comunicação entre os módulos funcionou como implementado. As falas que o robô reproduziu não foram reconhecidas devido à comunicação entre o módulo de reconhecimento e de reprodução.

VI. Conclusão

Um robô capaz de entender seres humanos no nível de fala traz a possibilidade de pessoas leigas interagirem com esses robôs em tarefas, por exemplo, domésticas. Para isso ocorrer, sistemas de audição robóticos devem ser desenvolvidos, técnicas de reconhecimento e síntese devem ser aprimoradas. Os módulos implementados mostram o que há hoje na linha de frente dessas áreas de pesquisa. Demonstra também a capacidade de isso acontecer em um futuro próximo. Os experimentos obtidos mostraram a eficiência das técnicas e das bibliotecas aqui utilizadas. Como trabalhos futuros existe a possibilidade de estender esse sistema para algo mais flexível que apenas comandos. Pode usar os microfones do Kinect para localizar espacialmente onde, em relação ao robô, o falante está. Desenvolver as rotinas dos estados Follow Me e Tour Guide. Criar novos estados, até mesmo *easter eggs* como “self destruction”.

VII. Agradecimentos

O autor agradece a todos primeiramente a todos os integrantes do LARA, especialmente aos do grupo AMORA, pois estes ajudaram e encorajaram o trabalho, com dicas e suporte. À George Brindeiro, que deu a ideia inicial do tema, incentivou o autor e o ajudou em todos os momentos cruciais deste trabalho.

REFERÊNCIAS

[1] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "Design and implementation of robot audition system 'hark' - open source software for listening to three simultaneous speakers," *Advanced Robotics*, vol. 24, no. 5-6, pp. 739–761, 2010. [Online]. Available: http://dx.doi.org/10.1163/016918610X493561

[2] P. A. Taylor, A. Black, and R. Caley, "The architecture of the festival speech synthesis system," in *The Third ESCA Workshop in Speech Synthesis*, Jenolan Caves, Australia, 1998, pp. 147–151.

[3] A. W. Black and P. A. Taylor, "The Festival Speech Synthesis System: System documentation," Human Communciation Research Centre, University of Edinburgh, Scotland, UK, Tech. Rep. HCRC/TR-83, 1997, avaliable at http://www.cstr.ed.ac.uk/projects/festival.html.

[4] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.