

# Controle de Robô Humanoide para a Interação com Seres Humanos

Cristiana Miranda de Farias

**Abstract**—O presente trabalho propõe um experimento em que a plataforma robótica NAO, da Aldebaran Robotics, deverá atuar em um cenário de um jogo de caça ao tesouro. Este é um cenário rico, em que há a necessidade de se lidar com diversas variáveis, como a interação com agentes externos por comandos de voz e a procura de objetos em diferentes condições de iluminação. Visando uma atuação robusta, serão utilizadas ferramentas como a biblioteca de visão computacional (OpenCV), o software de decodificação de comandos de voz por HMM (Julius), além do framework NAOqi.

## I. INTRODUÇÃO

A pesquisa em robótica e em áreas relacionadas vêm crescendo cada vez mais em diversos centros de pesquisa em todo mundo. De fato, esse desenvolvimento fica claro ao se notar que robôs de diversos tipos estão cada vez mais presentes no dia a dia da população: Tem-se grandes manipuladores industriais, pequenos robôs móveis para limpar a casa, drones capturando imagens aéreas, além de incontáveis outras aplicações. Em particular existe a classe de robôs móveis que possuem duas pernas para se locomover, estes são chamados de robôs humanoides [2].

É notável cada vez mais a presença destes robôs na mídia, seja pelas enormes inovações tecnológicas que vem sendo alcançadas, ou pelo eterno imaginário herdado de contos de ficção científica, em que homem e robô vivem lado a lado em futuros distantes. Apesar de ainda não se ter chegado nesse tempo, é possível vislumbrar nas próximas décadas um mundo em que se terão humanoides atuando como robôs de companhia, em situações de perigo, ou até mesmo realizando diversas tarefas em conjunto com a população. De fato, esses autômatos são bastante versáteis, com grandes graus de liberdade em seus movimentos e devido a sua característica de semelhança com humanos de fácil adaptação no mundo em que vivemos.

Visando se aprofundar cada vez mais em áreas correlatas à robótica, o presente trabalho procura, a partir de um experimento em que um jogo de Caça ao Tesouro[5] é implementado, se aprofundar no tema de interação entre robôs humanoides e seres humanos.

### A. Plataforma NAO

Para este projeto foi utilizada a plataforma NAO, produzida pela Aldebaran Robotics [1]. Este pequeno humanoide se popularizou pela sua aparência cativante e pela vasta gama de aplicações em que ele vem sendo utilizado, notavelmente interações com crianças autistas, interação humano-robô, controle remoto por teleoperação, além, é claro, da sua participação na categoria de plataforma padrão da RoboCup.

O NAO possui uma ampla gama de sensores e atuadores, operando em conjunto com um processador Intel Atom de 1.6 GHz e 1 GB de RAM, sistema operacional baseado na distribuição do Linux, Gentoo. Dentre seus diversos sensores o humanoide possui em seu torso dois sonares, um par com receptor e um transmissor a esquerda e outro a direita, também no torso tem-se a unidade inercial, que é composta por um giroscópio de dois eixos e um acelerômetro de três. Além destes, o NAO possui também dois sensores infravermelho nos olhos, quatro sensores de pressão em cada pé, sensores capacitivos na cabeça e em cada mão, um botão no torso, um bumper em cada pé e encoders que fornecem a posição de cada junta. Finalmente o robô também possui duas câmeras, com resolução de 640x480 pixels, uma localizada na parte frontal superior da cabeça e a outra na parte frontal inferior. Na “Fig. 1” pode-se ver uma descrição esquemática dos diversos sensores que o robô possui.

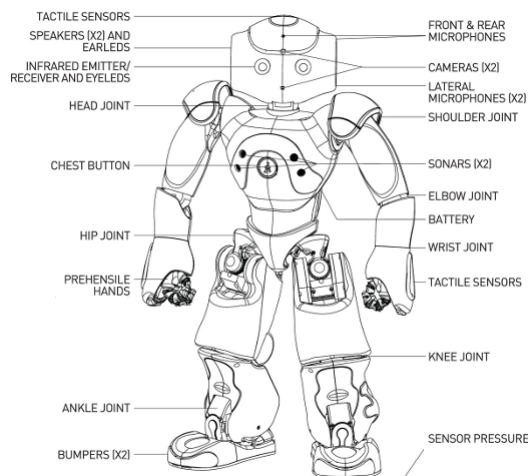


Fig. 1: Esquemático das posições dos principais sensores do NAO (fonte: <http://doc.aldebaran.com/>)

Podemos enfatizar, também, que o humanoide possui 25 graus de liberdade em seus movimentos, sendo os atuadores em suas juntas mostrados na “Fig. 2”

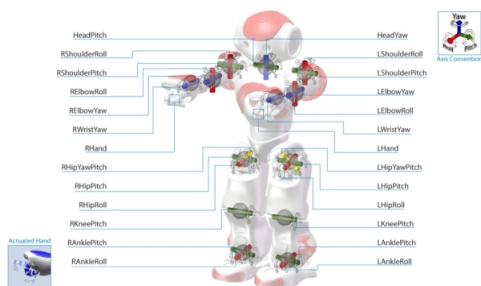


Fig. 2: Atuadores do Robô NAO (fonte: <http://doc.aldebaran.com/>)

Finalmente é importante ressaltar que tanto o acesso a memória do robô para a obtenção dos dados dos sensores ou a atuação sobre as suas juntas são feitos através de um framework disponibilizado pela Aldebaran Robotics, o NAOqi. Este framework chama os diversos módulos disponíveis através da criação de um Broker que carrega as bibliotecas referentes aos diferentes métodos que se deseja utilizar. A “Fig. ” faz referência a esse processo

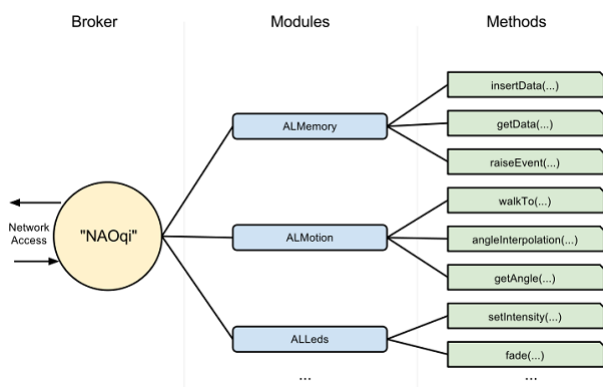


Fig. 3: Esquemático da estrutura do Sistema disponibilizado pela NAOqi (fonte: <http://doc.aldebaran.com/>)

### B. Caça ao Tesouro

Como descrito anteriormente a plataforma NAO é uma plataforma robusta, tendo como características principais a presença de inúmeros sensores e atuadores, além de sua aparência carismática, que faz bastante sucesso, principalmente com crianças. Visando utilizar todas as características presentes no robô, foi elaborado o experimento de um jogo, o Caça ao Tesouro.

O jogo se dá com o robô utilizando a câmera para procurar algum objeto em seu campo de visão, e caso esse não encontre o objeto, uma pessoa deverá interagir com o NAO dando as instruções “quente”, que significa que o robô está indo na direção certa ou “frio”, significando que o autômato está na direção errada. Ele, assim, irá se locomover por um ambiente e procurar um objeto.

Este é um jogo que pode ser usado para entreter crianças ou adultos que tenham vontade de brincar com o carismático NAO.

## II. METODOLOGIA

O projeto foi desenvolvido em quatro partes:

- Um módulo de reconhecimento de comandos de voz, feito com o software Julius. Este módulo roda em um servidor remoto e envia as mensagens via o protocolo de comunicação UDP. Nesta etapa o robô foi configurado como um cliente para o recebimento das mensagens.
- Um módulo para reconhecimento de objetos. Este é composto por um calibrador rodando remotamente e pelo módulo em si, embarcado no robô.
- Biblioteca auxiliar de atuação do robô, contendo funções para movimentação, fala e mudança de cor dos LEDs.
- Uma máquina de estados, descrevendo o comportamento do robô e integrando os diversos módulos desenvolvidos. Esta foi criada com base no objetivo proposto, de um jogo de caça ao tesouro, como mostrado na “Fig. 4”.

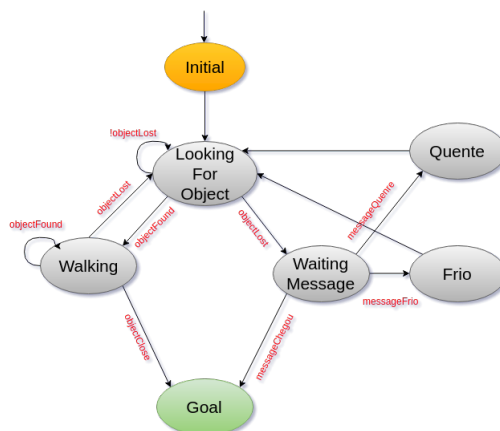


Fig. 4: Diagrama de Estados Para o Jogo de Caça ao Tesouro

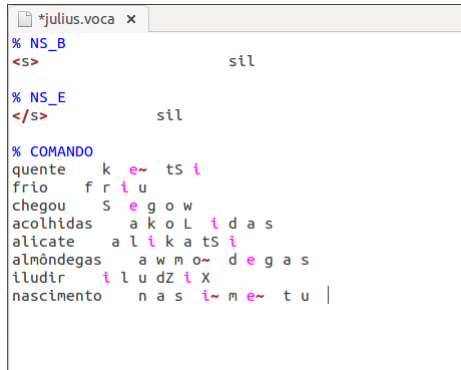
Cada etapa será descrita mais a fundo nas próximas seções.

### A. Reconhecimento de Voz

O módulo de reconhecimento de voz utilizou os resultados fornecidos pelo software Julius [6]. Inicialmente o sistema foi configurado utilizando as palavras “achou”, “quente” e “frio” como candidatas ao reconhecimento, ou seja, para qualquer som detectado, o software deveria tentar encontrar uma das três palavras e gerar um comando a partir delas.

Entretanto, como qualquer som era detectado como “achou”, “quente” ou “frio”, ruídos externos estavam gerando comandos que não deveriam ser enviados. Para contornar este problema foram adicionadas na lista de palavras candidatas ao reconhecimento cerca de cinco palavras novas

(escolhidas aleatoriamente). Desse modo os ruídos poderiam ser detectados como alguma dessas palavras, diminuindo, assim, o número de os falsos positivos. Podemos ver na “Fig 5” o arquivo usado para a configuração, em que NS\_B e NS\_E representam os silêncios antes e depois de cada palavra.



```

% NS_B
<s>          sil

% NS_E
</s>        sil

% COMANDO
quente      k e~ ts i
frio        f r i u
chegou      S e g o w
acolhidas   a k o L i d a s
alicate     a l i k a t s i
almôndegas  a w m o~ d e g a s
iludir      i l u d z i x
nascimento  n a s i~ m e~ t u |

```

Fig. 5: Palavras candidatas para reconhecimento pelo software Julius

Realizadas as configurações para o reconhecimento de voz, foi desenvolvido um programa para receber os dados vindos do Julius e os transmiti-los através de um servidor UDP. Somente os dados referentes as palavras “quente”, “frio” ou “achou” são enviados. Caso qualquer outra palavra seja identificada ela será descartada.

Quanto as configurações da rede, optou-se por utilizar um IP fixo em “10.1.0.2” no robô, de modo que não houvesse a necessidade de realizar mudanças no servidor todas as vezes que o sistema fosse reiniciado. Além disso, visando maior simplicidade, o pacote de mensagem foi definido como um char: a primeira letra da string da palavra reconhecida.

### B. Detecção de Objetos

A detecção de objetos foi feita através de técnicas de segmentação por cores e transformações morfológicas utilizando-se funções da biblioteca OpenCV [3].

Para este módulo, primeiramente foi capturada uma imagem no espaço de cores HSV pela câmera inferior presente no robô. Esta câmera foi escolhida pois a câmera superior perde objetos que estão muito próximos. A captura podia ser feita por ambas as câmeras, entretanto, como o projeto também utiliza comandos de voz para guiar o NAO, não há a necessidade de encontrar um objeto que esteja muito longe.

Após obtida a imagem esta é segmentada a partir de thresholds referentes a cada canal. Os thresholds são obtidos a partir de um programa de calibração, em que se pode ajustar manualmente cada valor. A próxima etapa é realizar transformações morfológicas de erosão e dilatação para eliminar ruídos, e finalmente verificar se a área do objeto encontrado é condizente (áreas muito pequenas são consideradas ruído). Tanto a área quanto o centro do objeto são encontrados a partir do calculo do momento da imagem.

### C. Funções de Atuação na plataforma NAO

A plataforma NAO possui um framework com diversas funções para a atuação no robô, a NAOqi. Para a chamada destas funções foram criados os módulos auxiliares, debugAuxiliaryFunctions e motionAuxiliaryFunctions.

A função de Debug foi criada para a receber cores referentes a ambos os olhos e ao botão do peito, para atuar sobre os LEDs presentes nestas regiões. Além disso nesse módulo foi implementada uma função para receber uma string e fazer com que o robô fale esta string. Ambas funções foram utilizadas para debug, indicando se partes do código foram alcançadas.

A outra função é referente aos movimentos de se levantar e de andar.

### D. Máquina de Estados

Com todos os módulos funcionando, há a necessidade de juntá-los e definir um comportamento. Na “Fig. 4” pode-se ver o diagrama referente ao que foi implementado no cenário experimental do jogo de Caça ao Tesouro.

Primeiramente tem-se o estado *Initial*, aqui o robô irá simplesmente se levantar e falar que o jogo vai começar. Este estado levará diretamente ao estado *Looking for Object*.

O estado *Looking for Object* tem a função de procurar um objeto, chamando portanto o módulo de detecção. Aqui haverá um contador que toda vez que o objeto não for encontrado será acrescido de uma unidade e o robô continuará procurando. Caso o contador chegue ao valor dez, o objeto será considerado perdido, o contador será zerado e a máquina de estados irá para o estado *Waiting Message*. Caso o objeto seja encontrada o contador será zerado e o próximo estado, *Walking*, será chamado.

O estado *Walking*, chamará o módulo de movimento e fará com que o robô ande até o objeto desejado. De modo semelhante ao estado *Looking for Object*, este estado possui um contador que irá crescer até no máximo dez, indicando se o objeto realmente foi perdido ou não. Caso o objeto esteja perdido o estado de procura será chamado novamente, caso o objeto seja encontrado perto o bastante, o robô irá parar e ir para o estado *Goal*. Caso o objeto seja encontrado mas não esteja perto o bastante o robô simplesmente continuará andando.

O estado *Waiting for Message* será chamado quando nenhum objeto estiver no campo de visão do robô, e, portanto, com ele parado irá esperar um dos três comandos possíveis: “quente”, “frio” ou “chegou”. Os comandos “quente” e “frio” irão para seus estados respectivos. O comando “chegou” foi criado porque muitas vezes o objeto estava muito próximo ao robô, mas fora do seu campo de visão, então havia a necessidade de sinalizar que o robô havia chegado ao seu objetivo, levando assim ao estado *Goal*.

Os estado *Quente* significa que o robô está na direção certa mas não consegue ver o objeto, assim ele deverá andar 30cm para frente. No estado *Frio* o robô está na direção errada, então ele irá rodar 1.5 radianos. Ambos os estados irão retornar para o estado *Looking for Object*.

O estado *Goal* será chamado quando o objetivo for alcançado. O robô irá avisar que chegou no lugar desejado e seus LEDs irão piscar.

### III. RESULTADOS

Para a análise dos resultados primeiramente foram feitos testes nos módulos de reconhecimento de voz e no reconhecimento de imagens. Os resultados estão relacionados nas seções III-A e III-B respectivamente. Após verificado que os métodos utilizados estavam funcionando, foram realizados experimentos com o sistema inteiro operando em conjunto, o que pode ser visto na seção III-C.

#### A. Reconhecimento de Voz

Para verificar os resultados obtidos referentes o reconhecimento de comandos de voz para cada uma das palavras desejadas foram feitos dois tipos de teste. Primeiramente verificou-se a taxa de acerto utilizando-se a configuração em que só as palavras “quente”, “frio” e “chegou” eram reconhecidas. Em seguida o experimento foi repetido com a configuração em que mais palavras eram reconhecidas e, se diferentes das desejadas, descartadas. Os resultados são apresentados na “Tab. I”.

TABLE I: Resultados do reconhecimento de voz

	Configuração com três palavras	Configuração com oito palavras
Quente	64%	84%
Frio	70%	72%
Chegou	98%	98%

Pode-se notar que houve uma melhora, principalmente referente ao “quente”, quando foram reconhecidas mais palavras. É interessante também ressaltar que esses testes foram feitos em um ambiente com pouco ruído, quando realizados testes em ambientes mais sonoros a quantidade de resultados errados é bem maior.

#### B. Detecção de Objetos

A detecção de objetos, pelo método implementado, sempre dependerá da calibração. Muitas vezes pequenas mudanças na luminosidade podem fazer com que um objeto previamente identificado seja perdido. Desse modo, considerando que o objeto utilizado seja composto por alguma cor que se destaque em seu meio, e que os thresholds para a segmentação sejam calibrados corretamente, ele geralmente será encontrado. Na figura “Fig. 6” pode-se ver diversos objetos sendo reconhecidos nas imagens, além de seus thresholds.

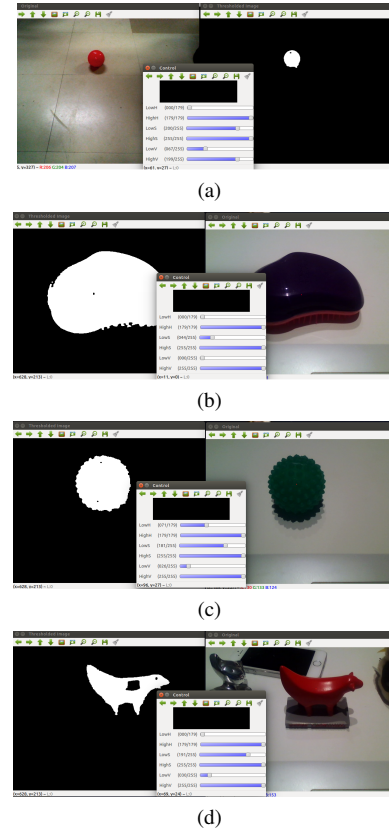


Fig. 6: Segmentação de diferentes objetos

Dentre os diversos métodos para reconhecimento de objetos, o utilizado se revelou ser a melhor opção. A “Fig. 6” mostra que itens de formatos variados podem ser facilmente reconhecidos, havendo somente a necessidade que uma rápida calibração seja feita. De fato, como se está utilizando somente a calibração por cores há um maior risco de falsos positivos, principalmente quando se tem muitos objetos de cores semelhantes no campo de visão do robô.

Entretanto, métodos como a detecção de características morfológicas, por exemplo detecção de círculos ou linhas[4], seriam específicos para um único formato de objeto, fazendo com que o módulo perdesse sua generalidade. Do mesmo modo, métodos que se utilizam de aprendizado de máquina, como o Haar Cascade[7], necessitam de grandes databases e de um longo período de treinamento para o reconhecimento de um único objeto, o que torna impraticável a ideia de utilizar um item qualquer como tesouro no cenário experimental da Caça ao Tesouro.

#### C. Experimentos de Caça ao Tesouro

Para testar o sistema por completo, no cenário de um jogo de Caça ao Tesouro, foram realizados três experimentos em que o robô deveria encontrar uma bola alaranjada. Em cada um dos testes a posição da bola e do robô foi diferente, a “Fig. 7” mostra cada uma das configurações utilizadas.

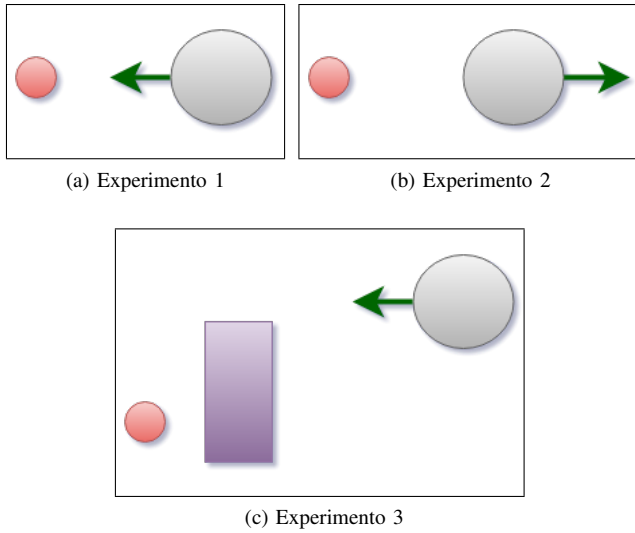


Fig. 7: Posição da bola (vermelho), robô (cinza) e obstáculo (roxo) para cada experimento.

1) *Experimento I:* Nesse experimento foi colocada a bola em frente ao NAO. Aqui não era desejado que o robô recebesse orientações por comando de voz, ele deveria simplesmente detectar o objeto e andar em sua direção. Como discutido anteriormente, quando o objeto se encontrava muito perto ele saía do campo de visão do robô e acabava não sendo detectado, portanto houve a necessidade de se reconhecer um comando de voz que indicava que o objetivo foi alcançado. A “Fig. 8” mostra o robô em relação a bola, assim como indicado previamente no esquema da “Fig. 7a”.

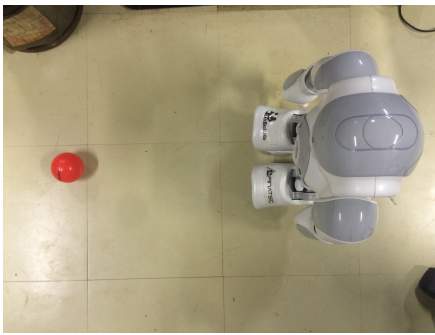


Fig. 8: Configuração do experimento 1

Para verificar seu funcionamento, esse experimento foi repetido quatro vezes. Sendo os resultados mostrados na “Tab. II”.

TABLE II: Resultados do experimento I

	Teste 1	Teste 2	Teste 3	Teste 4	Total	Total(%)
Número de comandos de voz	1	1	1	1	4	-
Comandos de voz corretos	1	1	1	1	4	100%
Comandos de voz incorretos	0	0	0	0	0	0%
Número de vezes em que a bola foi detectada	1	1	1	1	4	-
Bola detectada corretamente	1	1	1	1	4	100%
Bola detectada incorretamente	0	0	0	0	0	0%
Objetivo alcançado	1	1	1	1	4	100%

Este foi um teste simples, em que se desejava somente verificar se o sistema funcionava. Os resultados foram, portanto, bastante positivos, o reconhecimento de voz não falhou e, tendo em vista que a calibração havia sido feita momentos antes dos testes, a bola foi detectada 100% das vezes.

2) *Experimento II:* Este experimento foi um pouco mais complexo que o anterior. Aqui o robô se encontrava de costas para bola e houve a necessidade de se usar os comandos de voz para ele se reorientar de modo a encontrar o objeto. A “Fig. 9” mostra a configuração utilizada.

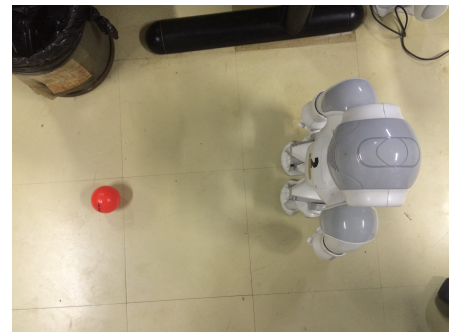


Fig. 9: Configuração do experimento 2

Mais uma vez, foram realizados quatro verificações para verificar o funcionamento do sistema, entretanto, em cada uma o trajeto resultante foi um pouco diferente, na “Fig. 10” pode-se ver um esboço de cada um dos trajetos percorridos.



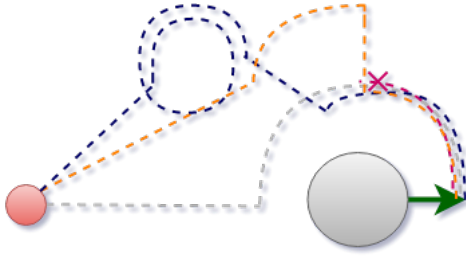


Fig. 10: Esboço dos trajetos percorridos pelo robô. Rosa: Teste 1; Azul: Teste 2; Cinza: Teste 3; Laranja: Teste 4

Além disso tem-se a taxa de acerto referente a cada um dos módulos, vistos na “tab. III”.

TABLE III: Resultados do experimento II

	Teste 1	Teste 2	Teste 3	Teste 4	Total	Total(%)
Número de comandos de voz	1	11	4	5	21	-
Comandos de voz corretos	0	9	4	5	18	85,7%
Comandos de voz incorretos	1	2	0	0	3	14,3%
Número de vezes em que a bola foi detectada	0	1	1	2	4	-
Bola detectada corretamente	0	1	1	1	4	80%
Bola detectada incorretamente	0	0	0	1	1	20%
Objetivo alcançado	0	1	1	1	3	75%

Este experimento é mais complexo, e visto que mais comandos são chamados há uma taxa de erro maior. A medida que erros ocorrem o robô geralmente começa a se afastar do seu objetivo, aumentando assim ainda mais o risco de algum falso positivo ocorrer em relação ao objeto procurado, ou de algum comando de voz ser identificado de modo errado. Nota-se, na “Fig. 10” que o trajeto em azul, que é o mais longo, foi o que resultou em uma maior taxa de erro.

É interessante notar, também, o problema de erros ocorrendo quando falsos positivos referentes ao comando “chegou” são detectados. Na “Fig. 10” isso ocorre no trajeto referente ao teste 1, em rosa. Quando a mensagem “chegou” é recebida o teste é terminado, não havendo a possibilidade do robô tentar corrigir sua trajetória e eventualmente alcançar seu objetivo.

Apesar dos problemas gerados por falsos positivos as taxas de acerto ainda são grandes, o que mostra que é possível guiar o robô até que ele consiga encontrar o objeto desejado.

3) *Experimento III:* O último experimento é referente a “Fig. 7c”, em que uma bola está escondida atrás de um

objeto e se encontra fora do campo de visão do robô. A “Fig. 11” mostra a configuração utilizada.



Fig. 11: Configuração do experimento 3

Como no caso do experimento anterior, foram esboçadas as rotas que o robô percorreu em quatro testes realizados, mostrados na “fig 12”.

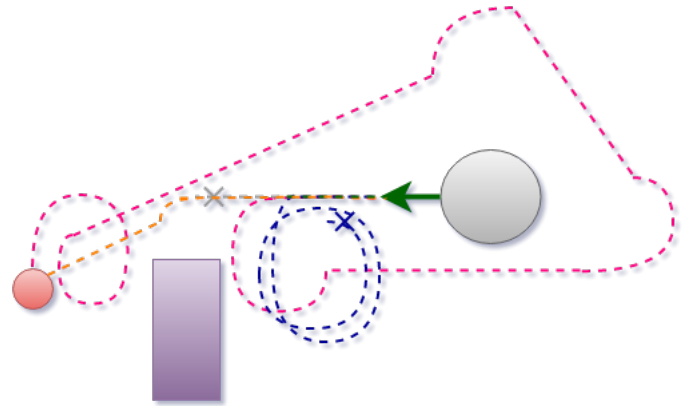


Fig. 12: Esboço dos trajetos percorridos pelo robô. Rosa: Teste 1; Azul: Teste 2; Cinza: Teste 3; Laranja: Teste 4

Foi feita também uma análise dos resultados mostrados, como visto na “Tab. IV”.

TABLE IV: Resultados do experimento III

	Teste 1	Teste 2	Teste 3	Teste 4	Total	Total(%)
Número de comandos de voz	13	12	3	5	33	-
Comandos de voz corretos	11	9	1	5	26	78,8%
Comandos de voz incorretos	2	3	2	0	7	21,2%
Número de vezes em que a bola foi detectada	2	0	0	1	3	-
Bola detectada corretamente	1	0	0	1	2	66,6%
Bola detectada incorretamente	1	0	0	0	1	33,3%
Objetivo alcançado	1	0	0	1	2	50%

Estes resultados mostram que as taxas de erro foram maiores que nos experimentos anteriores, o que é esperado, tendo em vista que a complexidade desse experimento é maior. Nota-se que devido a detecção incorreta do comando de parada, somente 50% dos testes atingiram o objetivo. Entretanto, mesmo quando se tem trajetos compridos, como o ocorrido no teste 2 (rosa), é possível fazer com que o robô se reorienta e chegue ao seu objetivo. Além disso, quando não são detectados falsos positivos, é possível guiar o robô através de poucos comandos até o objeto ser detectado.

#### IV. CONCLUSÃO E TRABALHOS FUTUROS

O objetivo deste projeto era a implementação de um sistema em que a plataforma robótica NAO pudesse utilizar suas diversas funcionalidades para interagir com uma pessoa e realizar diferentes ações. Desse modo, utilizou-se o experimento de um jogo de Caça ao Tesouro.

Como resultado, foram obtidos dados referentes aos módulos de reconhecimento de voz, detecção de objeto e do sistema funcionando como um todo no experimento proposto. O que se verificou, ao final, foram taxas de acerto altas em todos os testes feitos.

Verificou-se também que para todos os experimentos propostos o robô conseguiu chegar em seu objetivo pelo menos uma vez, o que mostrou que o jogo de Caça ao Tesouro pode ser implementado a partir das técnicas propostas neste projeto. Entretanto, pode-se ver que algumas falhas, como a detecção errada de comandos de voz, foram problemáticas nos testes feitos, fazendo com que o robô muitas vezes chegasse no seu estado final sem atingir seu objetivo. Como continuidade deste projeto poderão ser implementados algoritmos preditivos, de modo que o número de falsos positivos diminua.

Para trabalhos futuros é proposto que mais testes sejam feitos, garantindo, assim, resultados estatísticos mais confiáveis. Além disso experimentos mais complexos poderão ser feitos, e através deles a demanda por novas soluções e

pela integração de novos módulos com diferentes funcionalidades podem surgir.

#### REFERENCES

- [1] Aldebaran Robotics. online, 2015. [Online; acessado em 05/Fev/2016].
- [2] Sven Behnke. Humanoid robots - from fiction to reality? *KI*, 22(4):5–9, 2008.
- [3] G. Bradski. The OpenCV Library, 2000.
- [4] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972.
- [5] O. Stasse, T. Foissotte, D. Larlus, A. Kheddar, and K. Yokoi. Treasure hunting for humanoid robots. In *Workshop on Cognitive Humanoids Vision, 8th IEEE-RAS International Conference on Humanoid Robots*, 2009.
- [6] E. G. Tsardoulidis, A. L. Symeonidis, and P. A. Mitkas. An automatic speech detection architecture for social robot oral interaction. In *Proceedings of the Audio Mostly 2015 on Interaction With Sound*, AM '15, pages 33:1–33:8, New York, NY, USA, 2015. ACM.
- [7] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511–518. IEEE Computer Society, 2001.