Universidade de Passo Fundo
Curso de Engenharia Elétrica

Siclano de Souza

Título do Trabalho

Passo Fundo – RS
Julho de 2021

Siclano de Souza

titulo

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Engenharia Elétrica da Universidade de Passo Fundo como requisito parcial para obtenção da titulação como Engenheiro Eletricista.

Aprovado em Mês de 2021.

Comissão Examinadora

Prof. Dr.Eng. Fernando Passold
Orientador

Prof. Professor(a) da Banca 1

Professor(a) da Banca 2

Resumo

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Palavras-Chave:

Abstract

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Keywords:

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

1.1 Preambulo

Descreve uma história sobre dificuldades envolvidas em sistemas semelhantes ao que está sendo proposto.

1.2 Motivação

O que o levou a fazer este trabalho.

1.3 Justificativa

"O problema é este, por isto vou fazer deste jeito".
Justifica a "entrada"do meu sistema na "área", o que se poderia conseguir.
Qual o acréscimo que meu trabalho está causando ou espera obter nesta área.
Enfim, porque foi desenvolvido esta trabalho.

1.4 Objetivos

"Vou fazer isto", (curto e grosso - 1 parágrafo).

1.4.1 Objetivos Específicos

Seria uma "cópia"da lista de "objetivos específicos"que esta' no projeto de dissertação. Espera-se uma lista.
"É esperado para este projeto o desenvolvimento das seguintes atividades:

- Objetivo 1
- Objetivo 2
- Objetivo 3
- Objetivo 4

1.5 Organização do Trabalho

Explica-se a forma como foi organizado este documento, ou seja, explica-se sobre o que trata cada capitulo (item) da minha dissertação. Cada capítulo pode ser apresentado em sequencia. Um parágrafo para cada capítulo.

A seguir no capítulo 2 será discutido *Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer..*

Capítulo 2

Fundamentação

Apresenta a teoria que foi usada para realizar o trabalho. Pode ser apresentado de forma cronológica, citando sistemas semelhantes (vantagens e desvantagens). Ao final apresenta o "estado da arte"associado com o desenvolvimento deste projeto. Passa uma idéia para o leitor de sistemas semelhantes ou como o trabalho pode ser desenvolvido (não é este capítulo que apresenta como o trabalho foi desenvolvido; isto fica para o próximo capítulo).

Pode introduzir um parágrafo explicando como este capítulo foi organizado, a forma na qual foi dividido. Não ultrapassar muito 7 linhas.

2.1 Sistemas Similares (Revisão da Literatura)

Este é só um exemplo mostrando como incluir/realizar referências bibliográficas.

Robôs foram maquinas criadas inicialmente para realizar tarefas repetitivas, principalmente como é o caso com robôs manipuladores industriais normalmente usados na industria automobilística que passam o dia executando sempre a mesma sequencia de movimentos em operações comuns de pintura ou soldagem (SCIAVICCO; SICILIANO, 2012; SICILIANO; KHATIB, 2016a).

Mas alguns robôs deixaram de ser fixos e evoluíram para estruturas móveis ou plataformas móveis, inicialmente tele-operadas. Com o passar do tempo, e aprimoramentos associados principalmente com a área de inteligência artificial passaram a possuir certo comportamento autônomo, ou seja, que não depende de uma ação direta humana e foram sendo capazes de continuamente reagir de forma autônoma ao seu próprio entorno, desviando automaticamente de obstáculos à medida que se locomoviam, até chegar ao ponto de, usando sensores apropriados, serem capazes de mapear por conta própria o ambiente ao seu redor (entorno) e até mesmo se localizar no mesmo de forma autônoma (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011; RUSSEL; NORVIG, 2003; HERTZBERG; CHATILA, 2008; SICILIANO; KHATIB, 2016b) Neste último caso, ao procedimento de Mapeamento e Localização realizados de forma simultânea se dá o nome de SLAM (Simultaneous Localization and Mapping) (THRUN et al., 2005). Para realizar este tipo de ação, obviamente os robôs dependem do conhecimento do ambiente e diferentes tipos de sensores são adotados para detectar objetos, medir distâncias de obstáculos, inferir distâncias de fonte de referência fixas (radiofaróis), inferir distâncias já percorridas (encoders) e mapear o entorno usando sensores visuais (visão computacional ou de máquina) ou sensores de varredura laser (liDAR = laser rangefinders) (BORENSTEIN; EVERETT; FENG, 1996; THRUN et al., 2005).

Robôs móveis ou plataformas móveis autônomas dependem do conhecimento do ambiente para se locomoverem e diferentes tipos de sensores são adotados para detectar objetos, medir distâncias de paredes, inferir distâncias de rádio-faróis (intensidade do sinal) e mesmo para inferir distâncias já percorridas (*encoders*). Mas sensores são sujeitos à ruídos de leitura e defeitos e de forma a melhorar valores de medição quanto ao ambiente, algum processo de fusão sensorial deve ser empregado, combinando informações de dois ou mais sensores (BORENSTEIN; EVERETT; FENG, 1996; SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011).

Robôs para agricultura são normalmente mais complexos na sua parte mecânica porque requerem mecanis- mos de tração capazes de cruzar imperfeições no terreno onde estão se locomovendo (eventualmente saltando pedras e troncos de árvores caídos no chão). Para não dizer que ainda trazem acoplado numa plataforma que é móvel uma espécie de robô manipulador na forma de uma garra ou pinça eventualmente

Característica:	RaspBerry Pi IV	NVidia Jetson Nano
Clock:	??	??
Memória RAM:	??	??
GPU:	??	??
VRAM:	(compartilhada)	(compartilhada)
Pinos I/O:	44?	50?
Conector Câmera:	Sim (tipo XX)	Sim (tipo ZZ)
Custo Médio:	R$ 300,00	R$ 500,00

Tabela 2.1: Tabela comparativa entre placa RaspBerry Pi IV e NVidia Jetson Nano.

sendo capaz de interagir com o entorno.

(AGRO, 2019) e (KOLLEWE; DAVIES, 2019) citam o exemplo de um robô recém criado pela startup inglesa Fieldwork Robotics, uma spinoff da Universidade de Plymouth, na Inglaterra, desenvolvido para selecionar e colher framboesas. Trata-se de um ótimo exemplo de integração sensorial, já que o robô é equipado com uma garra manipuladora (pinça) equipada com sensor de imagem 3D (câmera de profundidade) e algoritmos de aprendizagem de máquina e IA para analisar o grau de desenvolvimento de cada fruta. Notar que sua pinça é capaz de se aproximar gentilmente da fruta usando aprendizagem de máquina. Segundo a proposta desta startup, este maquinário será capaz de colher 25 mil frutas por dia, 10 mil a mais que um humano. O processo ainda é um pouco lento, com o robô levando pouco mais de 1 minuto por framboesa. Mas a empresa pensa em resolver este problema numa versão final com quatro braços operando simultaneamente, e assim a máquina poderá identificar o grau de desenvolvimento da fruta em menos de 10 segundos. Na China já existem robôs colhendo tomates e testes estão sendo realizados testes para colher couve-flor (STARTAGRO, 2019).

2.2 Hardwares similares

Introduz sistema similares já existentes (ou não). Pode apresentar de forma cronológica como sistemas semelhantes fora desenvolvidos ao logno da estória. Como eles foram evoluindo, que cada novo sistema trouxe de vantagens em relação aos anteriores.

2.2.1 Sensores mais usados

Aqui se apresentam variáveis (estados do sistema) que devem ser captados e se apresenta as diferentes possibilidades e tipos de sensores que podem ser empregados para captar cada tipo diferente de variável necessária para o sistema.

Aqui devem ser apresentados os diferentes tipos de sensores, características básicas dos mesmos. Algo pode ser comentado a respeito de sua "popularidade". Sugere-se apresentar um quadro/tabela com as principais características ao lado de uma figura do sensor – por exemplo, ver figura XX.

2.2.2 Kits de desenvolvimento

Aqui pode-se apresentar placas ou sistemas quase prontos, apresentando suas principais características. Uma foto com tabela resumo ao lado é o ideal. A ideia aqui é passar para o leitor uma ideia da potencialidade de certo(s) sistema(s).

Se mais de um sistema for apresentado (ou está sendo selecionado), é **muito** interessante, terminar esta seção apresentando uma **tabela comparativa final** entre diferentes opções disponíveis. Esta tabela pode ser bastante útil para ressaltar grandes diferenças de valores entre diferentes produtos ou ressaltar grande diferenças de poder de processamento, quantidade de memória disponível para programação, clock de cada sistema, etc; por exemplo, vide tabela 2.1.

2.3 Algoritmo XX

Note, já estou escrevendo uma equação:

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \tag{2.1}$$

E posso referenciar esta equação no texto usando: podemos usar a eq. (2.1), para calcular as raízes de um polinômio de 2^a-ordem.

2.4 Softwares similares

Aqui se apresentam como sistemas similares foram desenvolvidos usando que tipo de software, linguagem de programação, *framework*, etc. Aqui são apresentadas o caso de uso de bibliotecas especias como **OpenCV** (e alguns de seus comandos com rápidos exemplos), **Matplotlib** (biblioteca para gráficos do Python, similar ao Matlab). Aqui se apresentam as bibliotecas, recursos, repositórios externos disponíveis e se passa uma idéia do que pode ser desenvolvido com cada uma destas partes/recursos. Seria ótimo apresentar testes realizados pelo próprio mostrando resultados obtidos com o uso de certas ferramentas (e de forma sútil fica claro que o aluno estava se familiarizando com alguma ferramenta/recurso específico). Códigos (completos ou longos) **não são** apresentados aqui, se for desejado disponibilizar/apresentar algum código exemplo, usar **Anexos** e referenciar que o código usado para produzir a figura YY consta no anexo ZZ.

2.4.1 Biblioteca de Software OpenCV

Para processamento de imagens é comum a adoção da biblioteca gráfica **OpenCV**. A figura 2.1 mostra um exemplo de uso.

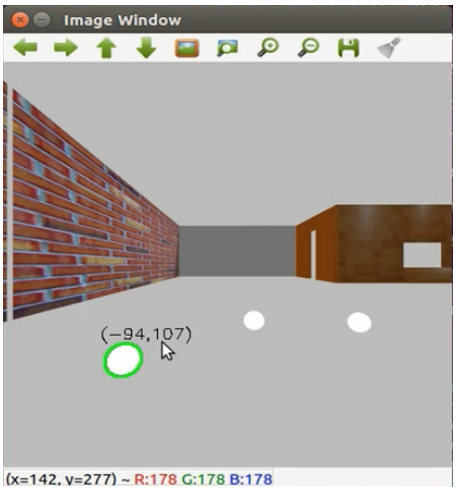


Figura 2.1: Resultado da rotina find_my_ball.py usando biblioteca OpenCv.

O código usado para gerar a figura acima é mostrado no anexo B.

Capítulo 3

Sistema Desenvolvido

3.1 Introdução

Introduz de maneira geral o sistema que foi desenvolvido. Um diagrama de blocos bem genérico sobre o sistema desenvolvido é super bem vindo.

Em seguida, as próximas seções explicam cada parte do sistema desenvolvido.

3.2 Hardware Desenvolvido

3.2.1 Sensores Adotados

3.2.2 Placa Microcontrolada/Microcontrolador Adotado

3.2.3 Hardware Final

3.3 Software Desenvolvido

3.3.1 Integração de Sistemas

3.3.2 Programação no PC/Raspberry

3.3.3 Programação na placa microcontrolada

3.3.4 Integração do Software

Estão previstas simulações iniciais usando ferramenta pública OpenAI Gym (`https://gym.openai.com`) para estudar diferentes algoritmos de aprendizagem, incluindo possíveis implementações físicas no mundo real usando o kit Lego Mindstorm NXT (BROCKMAN et al., 2016).

Capítulo 4

Resultados

4.1 Introdução

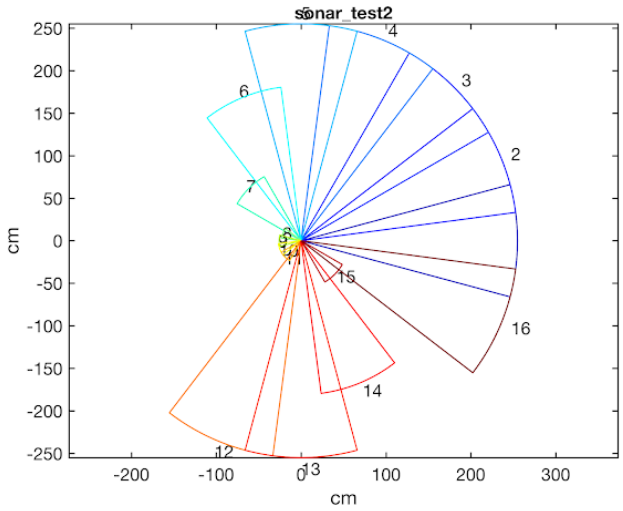
4.2 Testes do Hardware

Deve apresentar "cenários de teste"(condições/tipos) de teste previstos para item do hardware. Comprovar que diferentes subsistemas do hardware estão funcionando. No texto deve ficar claro o objetivo de cada teste (mas não é neste capítulo que se conclui que cada teste foi bem sucessivo).

4.2.1 Teste do Sensor de Ultrassom

A figura 4.1 mostra o resultado obtido com uma varredura de 360^o graus usando sensor de ultrassom nativo do Lego NXT, quando acoplado ao servo-motor que foi programado para efetuar sucessivas rotações incrementais de 15^o. O código adotado para gerar o gráfico pode ser visto no anexo A.

(a) Estrutura física implementada.



(b) Resultado obtido (gráfico polar).

Figura 4.1: Teste do sensor de ultrassom.

4.3 Testes de Software

Apresentar "cenários de teste"da parte do software. Podem ser apresentadas as telas e forma do usuário interagir com o sistema desenvolvido.

Pode apresentar testes de comunicação com diferentes itens do hardware para comprovar efetividade em troca de dados com sensores. Aqui podem ser apresentados dados "brutos"de sensores (sem filtragem) até para mostrar presença de ruído (e sua magnitude).

Aqui podem ser apresentados gráficos adquiridos do osciloscópio caracterizando sincronização com itens do hardware. Por exemplo: comprovar períodos de amostragem de algoritmos de controle, ou períodos e sinais envolvidos para captura de dados de certos sensores mais específicos (e especiais).

4.3.1 Teste de troca de mensagens

4.4 Teste do Sistema Completo

Aqui o aluno apresenta o sistema como um todo funcionando.

Capítulo 5

Conclusão e Discussões

Apresentados na forma de diferentes parágrados, algo do tipo (*mas não apresentar na fomra de lista numerada como aparece à seguir*):

1. "Consegui fazer... UFA". Melhor, se explica o que foi feito
2. "Funcionou, mas poderia funcionar melhor".
   O que faltou fazer, o que poderia ser adicionado, melhorado, incluído, ser melhor estudado no meu trabalho, como ele está.
   Lembrar que muitas pessoas (banca principalmente só lê o Resumo, Capitulo da Introdução e Capitulo final da Conclusão, uns poucos lêem o resto).
   Posso citar aqui o que outros vem desenvolvendo parecido na minha área, citar referencias bibliográficas ou outros centros que desenvolvem sistemas parecidos.
3. Vantagens do meu tipo de sistema.
   O que ele trouxe de contribuição para esta área. As vantagens dos outros sistemas.
4. Comenta-se o que o curso (colegas) vem desenvolvendo nesta área.
   Que se pode continuar o sistema desenvolvido originalmente como trabalhos para outras áreas, detalhando um pouco outras abordagens que poderiam ter sido adotadas para este trabalho, mas que não foram (não é necessário justificar).
   Mencionar que outross algoritmos mais eficientes ou outras abordagens podem ser exploradas.
5. Sugestões finais.
   Aqui voce "justifica"porque não resolveu certos problemas: tempo, fugia do escopo do trabalho, etc...

Bibliografia

AGRO, S. **Reino Unido terá primeiro robô colhedor de framboesas**. 28 mai. 2019. Disponível em: <http://www.startagro.agr.br/reino-unido-tera-primeiro-robo-colhedor-de-framboesas/>. Acesso em: 24 mai. 2020.

BORENSTEIN, J.; EVERETT, H. R.; FENG, L. **"Where Am I?"Sensors and Methods for Mobile Robot Positioning**. [S.l.], 1996.

BROCKMAN, G. et al. **OpenAI Gym**. cite arxiv:1606.01540: [s.n.], 2016. Disponível em: <https://www.bibsonomy.org/bibtex/2cdc8f927d6c8657ea82951a09e34161a/achakraborty>. Acesso em: 24 mai. 2020.

HERTZBERG, J.; CHATILA, R. AI Reasoning Methods for Robotics. In: SICILIANO, B.; KHATIB, O. (Ed.). **Handbook of Robotics**. NJ, USA: Springer Berlin Heidelberg, 2008. p. 207–223.

KOLLEWE, J.; DAVIES, R. **Robocrop: World's First Raspberry-Picking Robot Set to Work | Technology | The Guardian**. 2019. Disponível em: <https://www.theguardian.com/technology/2019/may/26/world-first-fruit-picking-robot-set-to-work-artificial-intelligence-farming>. Acesso em: 23 mai. 2020.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence, a Modern Approach**. 2. ed. [S.l.]: Prentice Hall, 2003.

SCIAVICCO, L.; SICILIANO, B. **Modelling and Control of Robot Manipulators**. [S.l.]: Springer Science & Business Media, 2012.

SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics**. [S.l.]: Springer, 2016a.

SICILIANO, B.; KHATIB, O. (Ed.). **Springer Handbook of Robotics**. 2. ed. Berlin: Springer, 2016b. (Springer Handbooks). ISBN 978-3-319-32550-7. DOI: 10.1007/978-3-540-30301-5.

SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. **Introduction to Autonomous Mobile Robots**. [S.l.]: MIT press, 2011.

STARTAGRO. Mundo Agtech: Edição de 15 de Abril | Mundo AGTECH, 2019. Disponível em: <http://www.startagro.agr.br/mundo-agtech-edicao-de-15-de-abril/>. Acesso em: 12 jul. 2019.

THRUN, S. et al. **Probabilistic Robotics**. [S.l.]: MIT Press, 2005. (Intelligent Robotics and Autonomous Agents Series). ISBN 978-0-262-20162-9. Disponível em: <https://books.google.com.br/books?id=k_yOQgAACAAJ>.

Referências

AGRO, S. **Reino Unido terá primeiro robô colhedor de framboesas**. 28 mai. 2019. Disponível em: <http://www.startagro.agr.br/reino-unido-tera-primeiro-robo-colhedor-de-framboesas/>. Acesso em: 24 mai. 2020.

BORENSTEIN, J.; EVERETT, H. R.; FENG, L. **"Where Am I?"Sensors and Methods for Mobile Robot Positioning**. [S.l.], 1996.

BROCKMAN, G. et al. **OpenAI Gym**. cite arxiv:1606.01540: [s.n.], 2016. Disponível em: <https://www.bibsonomy.org/bibtex/2cdc8f927d6c8657ea82951a09e34161a/achakraborty>. Acesso em: 24 mai. 2020.

HERTZBERG, J.; CHATILA, R. AI Reasoning Methods for Robotics. In: SICILIANO, B.; KHATIB, O. (Ed.). **Handbook of Robotics**. NJ, USA: Springer Berlin Heidelberg, 2008. p. 207–223.

KOLLEWE, J.; DAVIES, R. **Robocrop: World's First Raspberry-Picking Robot Set to Work | Technology | The Guardian**. 2019. Disponível em: <https://www.theguardian.com/technology/2019/may/26/world-first-fruit-picking-robot-set-to-work-artificial-intelligence-farming>. Acesso em: 23 mai. 2020.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence, a Modern Approach**. 2. ed. [S.l.]: Prentice Hall, 2003.

SCIAVICCO, L.; SICILIANO, B. **Modelling and Control of Robot Manipulators**. [S.l.]: Springer Science & Business Media, 2012.

SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics**. [S.l.]: Springer, 2016a.

SICILIANO, B.; KHATIB, O. (Ed.). **Springer Handbook of Robotics**. 2. ed. Berlin: Springer, 2016b. (Springer Handbooks). ISBN 978-3-319-32550-7. DOI: 10.1007/978-3-540-30301-5.

SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. **Introduction to Autonomous Mobile Robots**. [S.l.]: MIT press, 2011.

STARTAGRO. Mundo Agtech: Edição de 15 de Abril | Mundo AGTECH, 2019. Disponível em: <http://www.startagro.agr.br/mundo-agtech-edicao-de-15-de-abril/>. Acesso em: 12 jul. 2019.

THRUN, S. et al. **Probabilistic Robotics**. [S.l.]: MIT Press, 2005. (Intelligent Robotics and Autonomous Agents Series). ISBN 978-0-262-20162-9. Disponível em: <https://books.google.com.br/books?id=k_yOQgAACAAJ>.

Apêndice A

Rotinas Matlab

Segue listagem do *script* adotado para gerar o gráfico da figura 4.1 (pág. 21):

```
% teste_sonar2
% Fernando Passold 21/09/2018
%   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16
d=[255 255 255 255 255 182  87  23  26  27  27 255 255 181  56 255];
u=length(d); % caso o tamanho do vetor varie
step=2*pi/u;
angle=0;
range=15; % leque de abertura;
range_rad=(pi*range)/180;
step_r=pi/180; % passo para gerar o plot -15^o ? +15^o
%%%
step_color=64/u; % colormap<=>cmap:64x3
color=step_color;
cmap=colormap(jet); % winter
for index=1:u
  cont=2;
  range_sup=angle+range_rad;
  range_inf=angle-range_rad;
  x(1)=0; y(1)=0;
  for theta=range_inf:step_r:range_sup
    x(cont)=d(index)*cos(theta);
    y(cont)=d(index)*sin(theta);
    cont=cont+1;
  end
  msg=num2str(index);
  x_msg=d(index)*1.05*cos(angle);
  y_msg=d(index)*1.05*sin(angle);
  text(x_msg, y_msg, msg);
  x(cont)=0; y(cont)=0;
  plot(x,y,'Color',cmap(color,1:3));
  if index==1
    hold on; % para pr?ximos gr?ficos
  end
  angle=angle+step;
  color=color+step_color;
end
hold off
axis equal
title('sonar\_test2')
```

Apêndice B

Rotinas Python/OpenCV

A listagem abaixo ilustra o código usado pelo robô para localizar uma bola branca à sua frente usando biblioteca OpenCV (ver figura 2.1, pág. 17):

```
#!/usr/bin/env python

import rospy
import cv2
import numpy as np

from cv_bridge import CvBridge, CvBridgeError
from sensor_msgs.msgs import Image

class FindBall:
  def __init__(self):
    self.bridge = CvBridge()
    self.image_subcriber = rospy.Subscriber('/my_robot/camera1/rgb/image_raw', Image, self.image_callback)

  def image_callback(self, image_data):
    try:
      cv_image = self.bridge.imgmsg_to_cv2(image_data, 'bgr8')
      cv_image = cv2.resize(cv_image, dsize=(0,0), fx=0.5, fy=0.5)
    except CvBridgeError as e:
      rospy.logerror(e)

    mask = self.img_create_mask(cv_image)
        ball_coontour = self.find_contours(mask)
        centroid = self.center_of_contour(ball_coontour)
    self.img_cv_show(cv_image, ball_coontour, centroid)

    def center_of_contour(selft, ball_coontour):
        M = cv2.moments(contour)
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])
        return [cX, cY]

    def find_contours(self, mask):
        im2, contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        rospy.loginfo('Detected %s contours\n', len(contours))
        #rospy.loginfo('First Contour has %s points \n', len(contours[0]))
        #print(hierarchy)
        contourAreas = [cv2.contourArea(contour) for contour in contours]
        max_index = contourAreas.index(max(contourAreas))
        return contours[max_index]

  def img_create_mask(self, image_data,
            lower_range=np.array([220, 220, 200]),
            upper_range=np.array([255, 255, 255])):
    return cv2.inRange(image_data, lower_range, upper_range)

  def img_cv_show(self, image_data, contour=[], centroid=[]):
        if (contour != []  and centroid != [])
            cv2.drawContours(image_data,
                             contours=[contour],
                             contourIdx=1,
                             color=(0,255,0),
                             thickness=2)
            offCenterX = str(centroid[0] - image_data.shape[0]/2)
            offCenterY = str(image_data.shape[1] - centroid[1])
```

```
            offCenter = "(" + offCenterX + "," + offCenterY + ")"

            cv2.putText(image_data,
                        text=offCenter,
                        org=(centroid[0]-20, centroid[1]-20),
                        fontFace=cv2.FONT_HERSHEY_SIMPLEX,
                        fontScale=0.5,
                        color=(0,0,0))
    cv2.imshow('Image Window', image_data)
    cv2.waiKey(3)

if __name__ == '__main__':
  rospy.init_node('find_ball_node')
  rospy.loginfo('NODE CREATED')
  fb = FindBall()

  try:
    rospy:spin()
  expect KeyboardInterrupt:
    print('Shutting Down')

  cv2.destoryAllWindows()
```