

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
DEPARTAMENTO ACADÊMICO DE MECÂNICA  
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

ALEXANDRE GUILHERME DAMM DOS SANTOS

**APLICAÇÕES DE ROBÓTICA MÓVEL EM AMBIENTES  
DOMÉSTICOS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2018

ALEXANDRE GUILHERME DAMM DOS SANTOS

## **APLICAÇÕES DE ROBÓTICA MÓVEL EM AMBIENTES DOMÉSTICOS**

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial, dos Departamentos Acadêmicos de Eletrônica e Mecânica, da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. André Schneider de Oliveira

CURITIBA

2018

## **TERMO DE APROVAÇÃO**

ALEXANDRE GUILHERME DAMM DOS SANTOS

### **APLICAÇÕES DE ROBÓTICA MÓVEL EM AMBIENTES DOMÉSTICOS**

Este Trabalho de Conclusão de Curso foi apresentado no dia 10 de dezembro de 2018, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O aluno foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Milton Luiz Polli

Coordenador de Curso

Departamento Acadêmico de Mecânica

---

Prof. Me. Sérgio Moribe

Responsável pela Atividade de Trabalho de Conclusão de Curso

Departamento Acadêmico de Eletrônica

### **BANCA EXAMINADORA**

---

Prof. Dr. João Alberto Fabro  
UTFPR

---

Prof. Me. Ronnier Frates Rohrich  
UTFPR

---

Prof. Dr. André Schneider de Oliveira

Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

Dedico este trabalho aos dois amores da minha vida. À minha amada Natasha, por acreditar em mim e me incentivar a sempre buscar o melhor em mim. Você é minha musa e sempre será minha maior inspiração. Ao Emmanuel, meu filho, meu melhor amigo. Pelos momentos mais felizes que já vivi. Sou eternamente grato à Deus por ter pessoas tão especiais na minha vida.

## AGRADECIMENTOS

À Deus, minha gratidão pela vida que possuo.

À Natasha, minha companheira de todas as vidas. Minha maior incentivadora. Sem você nada disso seria possível. Passamos dias e noites afastados, lutando pela família, e mesmo com todas as tormentas, sempre conseguimos triunfar. Me sinto honrado em ter uma pessoa tão nobre ao meu lado.

Ao Emmanuel, por entender os meus compromissos, e por vir trabalhar comigo sempre que possível, apenas para ficar ao meu lado. Nos momentos difíceis, sempre encontrei em você a alegria necessária para seguir em frente.

Aos meus pais, Lori e Everaldo, com grande carinho.

À minha irmã Fernanda, por compartilhar comigo o amor pela ciência, quando eu ainda era uma criança.

À minha sogra Carla, por sempre torcer por mim.

Ao Prof. André Schneider de Oliveira, por confiar no meu trabalho e se mostrar presente sempre que necessário.

Ao Prof. João Alberto Fabro, pelo apoio e interesse em mim, assim como pelas palavras de sabedoria e incentivo durante o evento em João Pessoa.

Ao Danilo Henrique Santos, pela amizade e ajuda durante os eventos da *At Home*. Nas etapas finais do TCC seu auxílio foi de grande valia.

Aos demais colegas de LASER, pelos bons momentos envoltos em grandes momentos de dificuldade e superação.

Meus agradecimentos sinceros à UTFPR e todos os professores que fizeram parte da minha formação, pois parte do que sou eu devo a eles.

Há mundos onde a vida nunca surgiu. Há mundos que foram reduzidos a carvão e arruinados por catástrofes cósmicas. Somos afortunados: estamos vivos; somos poderosos; o bem estar da nossa civilização e da nossa espécie está em nossas mãos. Se não nos responsabilizarmos pela Terra, quem o fará? Se não nos comprometermos pela nossa responsabilidade, quem o fará?

(SAGAN, 1980)

## RESUMO

DOS SANTOS, Alexandre G. D. **Aplicações de Robótica Móvel em Ambientes Domésticos**. 2018. 59 páginas. Trabalho de Conclusão de Curso (Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A presença de robôs no cotidiano das pessoas tem aumentado nas últimas décadas. Os robôs utilizados em ambientes domésticos vêm ganhando cada vez mais espaço, em virtude de suas inúmeras aplicações, que incluem desde tarefas mais simples, como aspirar o pó do chão, até tarefas mais complexas, como o monitoramento dos diversos ambientes da casa. Para que um robô possa atuar em ambientes domésticos, é necessário que ele apresente um conjunto de habilidades para desempenhar as suas funções. Seguindo as indicações da *RoboCup@Home*, foi realizada a adaptação de um robô Pioneer 3-AT para o cumprimento de tarefas específicas em ambientes domésticos. Para tanto, foi instalada uma estrutura na base do robô e nele acrescentados os sensores *Light Detection and Ranging* (LIDAR) e Microsoft Kinect, compondo o sistema de navegação e visão do robô, e foi implementado um manipulador robótico em sua estrutura, possibilitando a manipulação de objetos. Foi realizada a instalação de diversos pacotes que utilizam o *Robotic Operating System* (ROS) como núcleo, de forma a possibilitar a operação do robô. Com todos os componentes em funcionamento, o robô P3-AT teve condições de realizar algumas tarefas, como navegação segura e manipulação de objetos, em um ambiente totalmente doméstico.

**Palavras-chave:** Robótica Móvel. ROS. Pioneer 3-AT. Manipulador robótico. *RoboCup@Home*.

## ABSTRACT

DOS SANTOS, Alexandre G. D. **Applications of Mobile Robotics in Domestic Environments**. 2018. 59 páginas. Trabalho de Conclusão de Curso (Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The presence of robots in people's daily lives has increased in last decades. The robots used in domestic environments come increasingly gaining space, due to its numerous applications, including simpler tasks, such as from vacuuming the floor, to more complex tasks, such as monitoring of the diverse environments of home. So that a robot can perform in domestic environments, it is necessary that he presents a set of skills to carry out their functions. Following the instructions of RoboCup@Home, the adaptation of a Pioneer 3-AT robot was performed to accomplish specific tasks in domestic environments. For that, a structure was installed in the base of the robot and in it added the sensors Light Detection and Ranging (LIDAR) and Microsoft Kinect, composing the system of navigation and vision of the robot, and a robotic arm was implemented in its structure, enabling the manipulation of objects. Several packages have been installed that use the Robotic Operating System (ROS) as a core, in order to allow the operation of the robot. With all components in operation, the robot P3-AT has conditions to accomplish some tasks, such as safe navigation and manipulation of objects, in a completely domestic environment.

**Keywords:** Mobile Robotics. ROS. Pioneer 3-AT. Robotic arm. RoboCup@Home.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Pioneer 3-AT .....	21
Figura 2 - Dimensões do Pioneer 3-AT .....	22
Figura 3 - Manipulador robótico .....	23
Figura 4 - Arduino UNO .....	24
Figura 5 - <i>Shield</i> ProtoShield RC .....	24
Figura 6 - Servomotor Power HD LF-20 .....	25
Figura 7 - LM2596 .....	26
Figura 8 - <i>Wii</i> mote .....	27
Figura 9 - Sensor EAI YDLIDAR X4 .....	28
Figura 10 - Sensor Microsoft Kinect .....	29
Figura 11 - Marcadores ARTag .....	29
Figura 12 - Primeira estrutura .....	34
Figura 13 - Montagem finalizada .....	35
Figura 14 - Participação na LARC .....	39
Figura 15 - Objeto sendo manipulado .....	40
Figura 16 - Robô sendo teleoperado .....	41
Figura 17 - Robô no ponto de origem .....	42
Figura 18 - Robô no ponto de chegada .....	43
Figura 19 - Robô manipulando um objeto com ARTag .....	44

## LISTA DE QUADROS

Quadro 1 - Especificações do robô Pioneer 3-AT .....	21
Quadro 2 - Comandos de teleoperação .....	37
Quadro 3 - Comandos de manipulação .....	38

## LISTA DE TABELAS

Tabela 1 - Especificações do manipulador robótico .....	23
Tabela 2 - Pacotes utilizados .....	36

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

ARTag	<i>Augmented Reality Tag</i>
IHC	Interação Humano-Computador
LARC	<i>Latin American Robotics Competition</i>
LASER	Laboratório Avançado de Sistemas Embarcados e Robótica
LIDAR	<i>Light Detection and Ranging</i>
RGB	<i>Red Green Blue</i>
ROS	<i>Robot Operating System</i>
USB	<i>Universal Serial Bus</i>
UTFPR	Universidade Tecnológica Federal do Paraná

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
1.1 PROBLEMA .....	15
1.2 JUSTIFICATIVA .....	16
1.3 OBJETIVOS .....	16
1.3.1 Objetivo geral .....	16
1.3.2 Objetivos específicos .....	17
<b>2 METODOLOGIA</b> .....	<b>18</b>
<b>3 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>19</b>
3.1 ROBÓTICA MÓVEL .....	19
3.2 APLICAÇÕES EM AMBIENTES DOMÉSTICOS .....	20
3.3 ROBÔ PIONEER 3-AT .....	21
3.4 MANIPULADOR ROBÓTICO .....	22
3.4.1 Arduino .....	23
3.4.2 <i>Shield</i> para Arduino .....	24
3.4.3 Servomotor .....	25
3.4.4 Regulador de tensão .....	25
3.5 TELEOPERAÇÃO .....	26
3.5.1 <i>Wii</i> mote .....	26
3.6 SISTEMA DE NAVEGAÇÃO .....	27
3.7 SISTEMA DE VISÃO .....	28
3.7.1 Sensor Microsoft Kinect .....	28
3.7.2 Reconhecimento de objetos .....	29
3.8 INTEGRAÇÃO DE TODOS OS COMPONENTES .....	30
3.8.1 ROS .....	30
3.8.1.1 <i>Filesystem Level</i> .....	30
3.8.1.2 <i>Computation Graph Level</i> .....	31
3.8.1.3 <i>Community Level</i> .....	32
3.9 Liga <i>ROBOCUP@HOME</i> .....	32
<b>4 DESENVOLVIMENTO DO TEMA</b> .....	<b>33</b>
4.1 PARTE MECÂNICA .....	33

4.1.1	Primeira estrutura .....	33
4.1.2	Segunda estrutura .....	34
4.2	PARTE ELÉTRICA .....	35
4.3	CONFIGURAÇÃO .....	36
4.3.1	Movimentação do robô .....	36
4.3.2	Teleoperação e manipulação de objetos .....	37
4.3.2.1	Teleoperação através do <i>Wiimote</i> .....	37
4.3.2.2	Manipulação de objetos através do <i>Wiimote</i> .....	37
4.3.3	Utilização do manipulador robótico .....	38
4.3.4	Sistema de navegação .....	38
4.3.5	Sistema de visão .....	39
4.4	TESTES .....	39
<b>5</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS .....</b>	<b>40</b>
5.1	MANIPULAÇÃO DE OBJETOS .....	40
5.2	TELEOPERAÇÃO .....	41
5.3	NAVEGAÇÃO AUTÔNOMA .....	41
5.4	MANIPULAÇÃO COM ARTAGS .....	43
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>45</b>
	<b>REFERÊNCIAS .....</b>	<b>47</b>
	<b>APÊNDICES .....</b>	<b>50</b>

## 1 INTRODUÇÃO

A robótica sempre despertou o interesse das pessoas, seja através de expectativas sobre suas aplicações, seja por meio de histórias retratadas por meio da ficção científica. Cada vez mais pode ser observado a utilização de robôs no dia a dia das pessoas, abandonando a ideia de que pertencem apenas aos livros de ficção científica (MEDEIROS et al., 2014).

A área da robótica cresceu de forma significativa nas últimas décadas, como resultado dos novos recursos de *hardware* e *software* desenvolvidos. Essa nova fase da robótica é definida pelo desenvolvimento de técnicas de controle inteligente de robôs. O objetivo é desenvolver robôs capazes de interagir com as pessoas e o ambiente ao seu redor de forma autônoma, ou seja, sem supervisão externa (MEDEIROS et al., 2014).

Robôs com essas características são denominados robôs autônomos e podem ser utilizados desde ambientes domésticos, realizando a limpeza do ambiente, até em ambientes hostis, onde a atuação direta do ser humano é difícil ou pode representar riscos a sua integridade física (MEDEIROS et al., 2014).

Existem robôs disponíveis no mercado que podem ser usados em tarefas no ambiente doméstico. Um exemplo deles é o robô Pioneer 3-AT, que recebendo diversas adaptações, pode atuar num ambiente doméstico e, a partir dos dados obtidos e do cenário em que está inserido, tomar decisões para cumprir suas tarefas (OLIVEIRA et al., 2017).

Atualmente, o Laboratório Avançado de Sistemas Embarcados e Robótica (LASER) trabalha com o desenvolvimento de robôs para fins domésticos. No presente trabalho será realizado a adaptação de um robô Pioneer 3-AT para o cumprimento de tarefas em ambientes domésticos.

## 1.1 PROBLEMA

Para que um robô possa atuar em ambientes domésticos, é necessário que ele apresente um conjunto de habilidades para desempenhar as suas funções. A *RoboCup@Home* elaborou uma lista de habilidades desejadas para robôs domésticos, a saber (RASCON et al., 2018):

- Navegação em ambientes dinâmicos: capacidade de navegar em ambientes sujeitos a mudanças repentinas;
- Calibração e configuração rápidas e fáceis: facilidade em configurar ou calibrar o robô em diferentes situações;
- Reconhecimento de objetos: capacidade de reconhecer objetos com características físicas diferentes;
- Manipulação de objetos: capacidade de manipular diferentes tipos de objetos;
- Detecção e reconhecimento de seres humanos: capacidade de detectar e reconhecer seres humanos;
- Interação com seres humanos: capacidade de interagir e estabelecer uma comunicação com seres humanos;
- Reconhecimento de fala: capacidade de reconhecer a fala humana e realizar sua interpretação;
- Inteligência ambiental: facilidade em coletar informações sobre o meio em que está inserido.

Portanto, para utilizar o robô Pioneer 3-AT em ambientes domésticos, será necessário realizar adaptações em sua estrutura e desenvolver algoritmos para que o robô possua as habilidades necessárias para a realização de tarefas em ambientes domésticos.



## 1.2 JUSTIFICATIVA

Com o desenvolvimento de novos recursos na tecnologia de robôs móveis, a participação de robôs no dia a dia das pessoas tem aumentado nas últimas décadas. Os robôs utilizados em ambientes domésticos vêm ganhando cada vez mais espaço, pelo fato de interagirem com pessoas e objetos ao seu redor, e em razão das inúmeras possibilidades a serem exploradas, que incluem desde tarefas mais simples, como aspirar o pó do chão, até tarefas mais complexas, como o monitoramento dos diversos ambientes da casa. Outra aplicação importante dos robôs domésticos está relacionada com a área de robótica assistiva, que tem por objetivo melhorar a qualidade de vida de indivíduos com limitações funcionais e que não possuem condições de realizar determinadas tarefas cotidianas.

## 1.3 OBJETIVOS

O trabalho consiste no desenvolvimento de aplicações de robótica móvel para ambientes domésticos.

### 1.3.1 Objetivo geral

Adaptar o robô Pioneer 3-AT para realizar a manipulação de objetos, navegar com segurança em ambientes domésticos e avaliar o seu desempenho na competição nacional da *RoboCup@Home*.

### 1.3.2 Objetivos específicos

Para atingir o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

1) Permitir que o robô Pioneer 3-AT realize a manipulação de objetos ao seu redor, através da instalação de um manipulador robótico;

2) Realizar o sistema de navegação do robô P3-AT, de forma que o robô apresente navegação segura em ambientes domésticos, ou seja, trabalhe com baixas velocidades e realize o desvio de obstáculos;

3) Elaborar algoritmos para que o robô P3-AT tenha condições de participar do seguinte teste sugerido pela *RoboCup@Home* (RASCON et al., 2018):

- *Help-me-carry*. Carregar objetos de um lugar para outro com navegação segura, conseguir acompanhar um operador e navegar em ambientes desconhecidos.

## 2 METODOLOGIA

Para alcançar os objetivos estabelecidos, foram elaboradas as seguintes etapas:

1ª etapa: Incorporar um manipulador robótico na estrutura do robô Pioneer 3-AT;

2ª etapa: Realizar a disposição dos componentes que serão utilizados pelo robô P3-AT, como Arduino UNO, Microsoft Kinect, entre outros;

3ª etapa: Implementar um mecanismo de teleoperação no robô P3-AT;

4ª etapa: Desenvolver os sistemas de navegação e visão do robô P3-AT;

5ª etapa: Elaborar algoritmos para que o robô P3-AT realize a manipulação de objetos e navegue com segurança em ambientes domésticos;

6ª etapa: Realizar testes e verificar o desempenho do robô P3-AT no evento *Latin American Robotics Competition* (LARC).

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são desenvolvidos os conceitos teóricos envolvidos neste trabalho.

#### 3.1 ROBÓTICA MÓVEL

No final da década de 1970 surgiu o termo mecatrônica, para designar a união das áreas de mecânica, eletrônica e computação. A interação das três áreas pode gerar uma série de aplicações, como robótica móvel e robôs bípedes, entre outros, portanto, dependendo da aplicação em questão, o termo mecatrônica pode ser interpretado de formas distintas (ROSÁRIO, 2005).

Para muitos pesquisadores, o surgimento da mecatrônica está ligado ao desenvolvimento de robôs. Os projetos em robótica conduziram o desenvolvimento de novas tecnologias, como sensores, visando a capacidade perceptiva de robôs, e atuadores, possibilitando a criação de robôs capazes de se locomover e realizar tarefas de manipulação de forma precisa e eficiente em diferentes tipos de terrenos (ROSÁRIO, 2005).

Existem diversos tipos de robôs. Os robôs móveis são robôs que possuem a capacidade de se locomover em diferentes ambientes e podem ser terrestres, aéreos ou subaquáticos. Esses robôs apresentam diferentes níveis de inteligência e autonomia, podendo ser operados remotamente, ou atuando de forma autônoma, interagindo com humanos, ou trabalhando em grupos de robôs na solução de problemas (MEDEIROS et al., 2014).

Um dos desafios da robótica móvel é desenvolver robôs com a capacidade de interagir com o ambiente e tomar decisões adequadas para que suas tarefas sejam executadas com sucesso. Para que consiga operar de forma eficiente, um robô deve ser capaz de atuar em ambientes desconhecidos e dinâmicos e de responder diante de situações inesperadas (MEDEIROS et al., 2014).

Um robô móvel que pode ser utilizado no meio acadêmico é o Pioneer 3-AT. Através desse robô é possível realizar uma série de experimentos em razão de sua versatilidade. Ao inserir sensores, o robô consegue extrair informações sobre sua posição e o ambiente a sua volta, assim como fazer o reconhecimento de pessoas e objetos. Existe ainda a possibilidade de incorporar um manipulador robótico em sua base, ou numa estrutura fixada na base, permitindo a manipulação de objetos (MOBILEROBOTS, 2006).

A integração de todos os componentes do robô P3-AT, como sensores, entre outros, pode ser realizada através do *Robot Operating System* (ROS), um *framework* que contém um conjunto de bibliotecas e ferramentas que fornecem uma camada de abstração sobre o sistema operacional Linux, atuando na padronização de mensagens em sistemas robóticos (ROS.ORG, 2018).

### 3.2 APLICAÇÕES EM AMBIENTES DOMÉSTICOS

Um ambiente doméstico consiste em ambientes interconectados, como, por exemplo, um apartamento com uma sala de estar, uma cozinha, um banheiro e um quarto. Existem paredes e portas dividindo esses ambientes. Normalmente não existem degraus ou diferenças no nível do chão do apartamento. Ao inserir um robô móvel em um ambiente doméstico, deseja-se que esse robô não tenha dificuldade em acessar qualquer ambiente do apartamento e que possa realizar suas atividades sem qualquer impedimento (RASCON et al., 2018). Para que um robô possa atuar em ambientes domésticos, é necessário que ele tenha sido fabricado para esse propósito ou, no caso de robôs sem uma finalidade específica, que este receba as devidas modificações para ser usado nesse tipo de ambiente.

### 3.3 ROBÔ PIONEER 3-AT

O robô Pioneer 3-AT pode atuar em diferentes tipos de terreno e sua base oferece diversas possibilidades, como a incorporação de sensores, ou até mesmo de uma estrutura para abrigar diferentes componentes, como *notebook*, sensores, manipuladores, entre outros. No quadro 1 são apresentadas as especificações do P3-AT.

Item	Descrição
Dimensões	508mm de comprimento x 497mm de largura x 277mm de altura
Peso	12kg
Motor	4 motores em cada roda, com os pares conectados por correias
Autonomia	2 a 4 horas, ao utilizar 3 baterias de 12V e 7,2Ah
<i>Inputs</i>	32 digitais, 8 analógicas
<i>Outputs</i>	8 digitais, 3 portas seriais
Velocidade	Velocidade linear: 0,7m/s, Velocidade de rotação: 140°/s

Quadro 1 - Especificações do robô Pioneer 3-AT

Fonte: Génération Robots (2018)

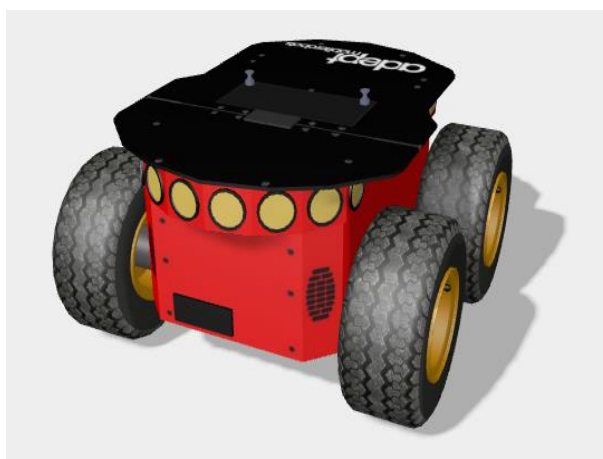


Figura 1 - Pioneer 3-AT

Fonte: Webots User Guide (2018)

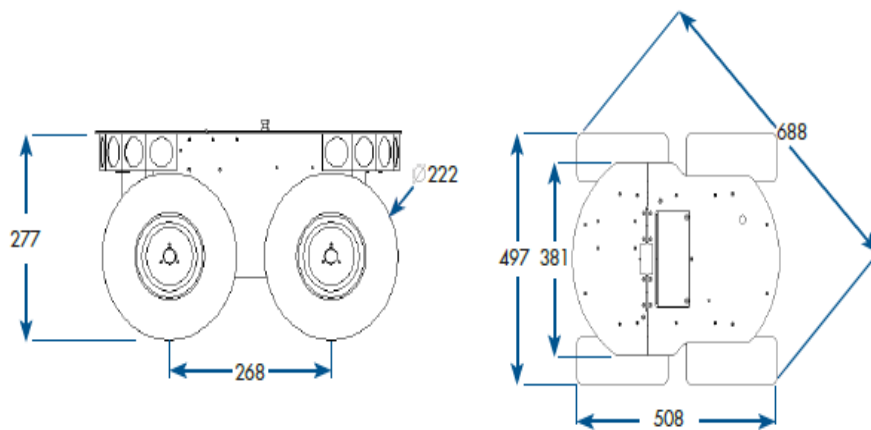


Figura 2 - Dimensões do Pioneer 3-AT

Fonte: Adept (2011)

### 3.4 MANIPULADOR ROBÓTICO

Os manipuladores são constituídos por corpos rígidos denominados segmentos, conectados uns aos outros através de juntas. O número de graus de liberdade de um manipulador é o número de variáveis independentes utilizado para movimentação espacial. Na extremidade livre do conjunto de segmentos que compõe o manipulador está o *end-effector*, responsável pela realização de tarefas específicas, como a manipulação de objetos (CRAIG, 2005).

A fim de realizar a manipulação de objetos, foi incorporado ao robô P3-AT um manipulador comercial com 6 graus de liberdade, e que faz uso de 6 servomotores. O manipulador é controlado por um Arduino UNO e utiliza um *end-effector* do tipo garra. Seguem na tabela 1 as especificações do manipulador robótico.

Tabela 1 - Especificações do manipulador robótico

Item	Comprimento (mm)
Alcance máximo	460
Abertura da garra	121

Fonte: Autoria própria.

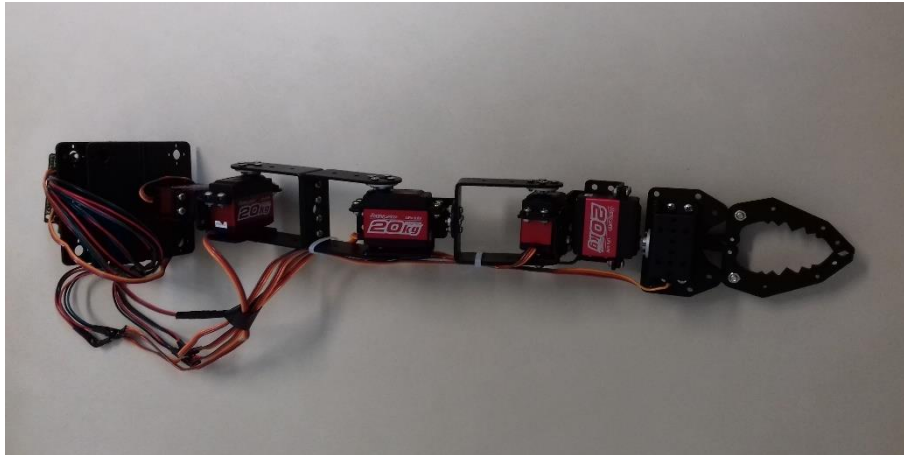


Figura 3 - Manipulador robótico

Fonte: Autoria própria.

### 3.4.1 Arduino

Para controlar os valores dos ângulos das juntas, foi utilizado o Arduino UNO. Um Arduino é uma placa de microcontrolador que permite a ligação com um computador através de uma conexão *Universal Serial Bus* (USB). Além disso, contém outros terminais que permitem a conexão com dispositivos externos, como servomotores, sensores, entre outros (MONK, 2013).





Figura 4 - Arduino Uno

Fonte: Autoria própria.

### 3.4.2 *Shield* para Arduino

Para usar os 6 servomotores do manipulador robótico, foi utilizado o *shield* ProtoShield RC da Robocore. Um *shield* é uma placa que pode ser conectada na parte superior do Arduino, expandindo suas capacidades (ARDUINO, 2018).

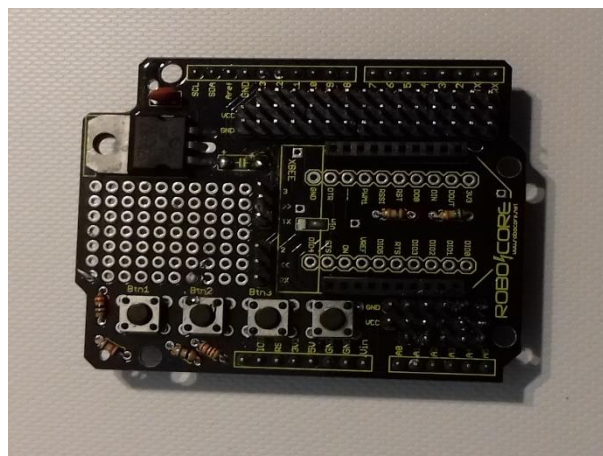


Figura 5 - *Shield* ProtoShield RC

Fonte: Autoria própria.

### 3.4.3 Servomotor

O servomotor é o atuador responsável pelo movimento da junta do manipulador. Para movimentar as juntas do manipulador foi utilizado um conjunto de 6 servomotores do modelo Power HD LF-20, com torque máximo de 20,0kg·cm, quando submetidos a uma tensão de 6,6V, permitindo ao manipulador sustentar objetos similares a uma lata vazia de refrigerante.



Figura 6 - Servomotor Power HD LF-20

Fonte: Autoria própria.

### 3.4.4 Regulador de tensão

O regulador de tensão LM2596 foi utilizado para reduzir a tensão de 12V da bateria para 6,6V, e mantê-la estável, com o objetivo de alimentar os servomotores e garantir o maior torque possível.

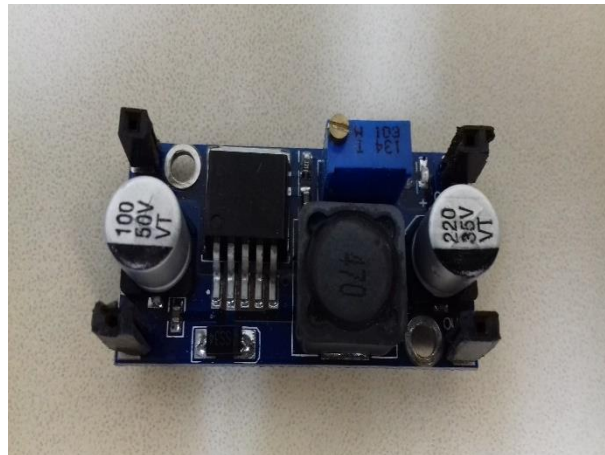


Figura 7 - LM2596

Fonte: Aatoria própria.

### 3.5 TELEOPERAÇÃO

Para realizar testes em robôs, como teste de navegação segura, por exemplo, é necessário que o operador do robô implemente um mecanismo de teleoperação. Na teleoperação, o controle do robô é realizado remotamente ou próximo da área de atuação do robô, tendo como referência as informações recebidas pelos sensores ou pela câmera do próprio robô (MEDEIROS et al., 2014).

A fim de realizar testes com o robô P3-AT, foi implementado um mecanismo de teleoperação utilizando o controle *Wii Remote*, popularmente chamado de *Wiimote*.

#### 3.5.1 *Wiimote*

O *Wiimote* é o controle principal do *Console Nintendo Wii* e será empregado para realizar a teleoperação do robô P3-AT. O ROS usa *Bluetooth* para se comunicar com o dispositivo *Wiimote*, obtendo dados dos botões, do

estado dos LEDs, da câmera de infravermelho, do acelerômetro, do giroscópio e do estado da bateria.



Figura 8 - *Wiimote*

Fonte: Autoria própria.

### 3.6 SISTEMA DE NAVEGAÇÃO

Para realizar a navegação, será usado um sensor *Light Detection and Ranging* (LIDAR). LIDAR é uma tecnologia de sensoriamento remoto que emite feixes de luz e mede o tempo que leva para as reflexões serem detectadas pelo sensor. Os dados são usados para calcular os valores das distâncias até objetos próximos. As coordenadas tridimensionais dos objetos ao redor do sensor são calculadas a partir da diferença de tempo entre o pulso de laser sendo emitido e retornado (NOAA, 2012).

O sensor LIDAR detecta o ambiente ao seu redor com poucas lacunas, e se objetos próximos estão em movimento ou repouso, por essa razão o LIDAR é ideal para tarefas de detecção de objetos e proteção contra colisões. Foi utilizado o sensor EAI YDLIDAR X4 para realizar a navegação do robô P3-AT.



Figura 9 - Sensor EAI YDLIDAR X4

Fonte: Autoria própria.

### 3.7 SISTEMA DE VISÃO

Para implementar um sistema de visão no robô P3-AT, possibilitando a criação de mapas de navegação e realização de teleoperação remota, foi utilizado o sensor Microsoft Kinect.

#### 3.7.1 Sensor Microsoft Kinect

O sensor Microsoft Kinect apresenta as seguintes funcionalidades (MICROSOFT DOCS, 2018):

- Sensor de profundidade: Permite a visualização do ambiente ao seu redor, com alcance mínimo de 800mm e máximo de 4000mm;
- Câmera *Red Green Blue* (RGB): Permite o reconhecimento facial de pessoas que estão na frente da câmera.



Figura 10 - Sensor Microsoft Kinect

Fonte: A autoria própria.

### 3.7.2 Reconhecimento de objetos

Para facilitar o reconhecimento de objetos distintos através do sensor Microsoft Kinect, foi atribuído um *Augmented Reality Tag* (ARTag) a cada objeto. Um ARTag é um sistema de marcadores fiduciais que consiste em padrões que são construídos no ambiente e detectados por câmeras digitais utilizando um algoritmo de detecção específico. Ele é útil para a navegação de robôs, onde a posição relativa entre a câmera e o objeto é necessária. Os marcadores ARTag têm taxas de erro muito baixas, em consequência da sua baixa taxa de falsos positivos e imunidade à variação de iluminação (FIALA, 2005).

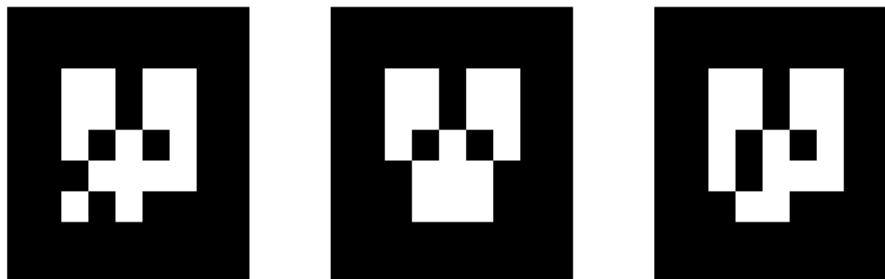


Figura 11 - Marcadores ARTag

Fonte: Saito (2016)

### 3.8 INTEGRAÇÃO DE TODOS OS COMPONENTES

Existem diversos sistemas que permitem a criação de aplicações para robôs, como Player, YARP, Orocos, CARMEN, Orca, MOOS, Microsoft Robotics Studio e ROS (DATTALO, 2018). Para realizar a integração de todos os dispositivos do robô P3-AT, será utilizado o ROS. Até o momento, o ROS só pode ser utilizado em sistemas GNU/Linux.

#### 3.8.1 ROS

O ROS é um *framework* de código aberto para aplicações em robótica. A função do ROS é simplificar a criação de aplicações complexas para robôs. O ROS é composto por três partes: *Filesystem Level*, *Computation Graph Level* e *Community Level*.

##### 3.8.1.1 *Filesystem Level*

*Filesystem Level* é o nível onde estão localizados os recursos do ROS. É composto por *packages*, *metapackages*, *package manifests*, *message types* e *service types* (JOSEPH, 2015).

- *Package*: ou pacote, é a unidade básica utilizada pelo ROS. Um pacote pode conter processos, dependências e arquivos de configuração;
- *Metapackage*: também chamado de *stack*, representa um conjunto de pacotes relacionados;
- *Package Manifests*: ou manifesto, fornece informações sobre um pacote;

- *Message type*: define a estrutura de dados para mensagens enviadas pelo ROS;
- *Service type*: define a estrutura de dados para solicitar e responder serviços pelo ROS.

### 3.8.1.2 *Computation Graph Level*

*Computation Graph Level* é o nível onde a rede de processos do ROS é executada. É composto por *nodes*, *Master*, *Parameter Server*, *messages*, *topics*, *services* e *bags* (JOSEPH, 2015).

- *Nodes*: ou nós, são processos que executam ações. Um robô compreende diversos nós, como, por exemplo, um nó controlando os motores das rodas;
- *Master*: ou Mestre, permite que os nós se encontrem, troquem mensagens e chamem serviços;
- *Parameter Server*: ou Servidor de Parâmetros, permite que os dados sejam armazenados em um local central;
- *Messages*: ou mensagens. Os nós se comunicam através de mensagens. Uma mensagem é uma estrutura de dados;
- *Topics*: ou tópicos, são os nomes utilizados para identificar o conteúdo de uma mensagem. Um nó interessado em um determinado tipo de dado assinará o tópico apropriado. Pode haver vários *publishers* e *subscribers* para um único tópico, e um único nó pode publicar e assinar vários tópicos;
- *Services*: ou serviços. Nos casos onde o modelo *publish-subscribe* não é indicado, é necessário utilizar o modelo *request-response*, realizado através de serviços;
- *Bags*: são um formato para salvar e reproduzir dados de mensagens do ROS.



### 3.8.1.3 *Community Level*

*Community Level* é o nível onde ocorre o compartilhamento de arquivos e informações entre os usuários do ROS. É composto por distribuições, repositórios, sites e blogs (JOSEPH, 2015).

### 3.9 Liga *RoboCup@Home*

A *RoboCup@Home* é uma competição que tem como objetivo o desenvolvimento de robôs que venham a desempenhar tarefas em ambientes domésticos, sendo a maior competição internacional de robôs de serviço autônomos e faz parte da iniciativa *RoboCup*. Para executar as tarefas propostas, são desenvolvidos, dentre vários campos da robótica, a interação entre homem e robô, a navegação e mapeamento em ambientes dinâmicos e a visão computacional (RASCON et al., 2018)

## 4 DESENVOLVIMENTO DO TEMA

Seguindo as etapas estabelecidas no capítulo 2, foram desenvolvidas as atividades a seguir.

### 4.1 PARTE MECÂNICA

Para instalar todos os componentes no robô Pioneer 3-AT, foi necessário incorporar duas estruturas à base do robô.

#### 4.1.1 Primeira estrutura

A primeira estrutura apresenta dois níveis, sendo que o primeiro nível está a 145mm da base do robô e o segundo nível a 305mm. Essa estrutura foi usada para fixar o sensor LIDAR e servir de base para segunda estrutura. Para a sua construção, foram utilizados duas chapas de alumínio com dimensões de 300mm x 300mm, feitos 4 furos próximos de suas extremidades, sendo atravessadas por barras roscadas de 7mm de diâmetro. As chapas são sustentadas por barras redondas vazadas presas com porcas sextavadas normais e autotravantes. O resultado pode ser visto na figura 12.



Figura 12 - Primeira estrutura

Fonte: Aatoria própria.

#### 4.1.2 Segunda estrutura

A segunda estrutura é uma eletrocalha perfurada em alumínio com 120mm de largura, 120mm de comprimento e 690mm de altura e foi fixada na primeira estrutura com cantoneiras, parafusos e porcas sextavadas. Na parte interna foi realizada a divisão da estrutura através de 3 prateleiras de madeira, fixadas por parafusos.

Na parte superior foi acomodado o sensor Microsoft Kinect, já que o sistema de visão do robô precisa de uma posição privilegiada. Na parte frontal foi alocado um tablet, responsável pela Interação Humano-Computador (IHC), que vai ser utilizado em estudos futuros. Na lateral esquerda foi instalado o manipulador robótico, de forma a simular um braço humano. Na parte interna, na prateleira do centro, foram acomodados os componentes Arduino UNO, *shield* ProtoShield RC e LM2596. Na prateleira inferior, foram acomodadas as caixas de som para complementar o sistema de IHC.



Figura 13 - Montagem finalizada

Fonte: Autoria própria.

## 4.2 PARTE ELÉTRICA

A alimentação do robô P3-AT vem de duas baterias de 12V e 7,2Ah. As duas baterias também alimentam o sensor Microsoft Kinect e o manipulador robótico. O Kinect opera sob a tensão de 12V, e por consumir a carga da bateria rapidamente, foi instalado um interruptor para mantê-lo desligado quando fora de operação. O manipulador atua sob a tensão de 6,6V, portanto, foi necessário utilizar um LM2596 para reduzir a tensão de 12V para a tensão desejada. O Arduino Uno e o sensor LIDAR são alimentados pelas portas USB do *notebook*. No Apêndice A pode ser visualizado o diagrama elétrico utilizado no robô P3-AT.

### 4.3 CONFIGURAÇÃO

Para realizar a comunicação com o robô Pioneer 3-AT, é necessário usar o ROS. No trabalho foi utilizada a distribuição *Kinetic*. A tabela 2 mostra todos os pacotes utilizados no trabalho.

Tabela 2 - Pacotes utilizados

<b>Função</b>	<b>Pacote</b>
Movimentação do robô	<i>RosAria</i>
Teleoperação do robô	<i>Wiiote</i>
Manipulação de objetos	<i>rosserial_arduino</i>
Sistema de navegação	<i>Gmapping</i> <i>hector_mapping</i> <i>navigation</i>
Sistema de visão	<i>libfreenect</i> <i>freenect_camera</i> <i>ar_track_alvar</i>

Fonte: Autoria própria.

Os procedimentos utilizados para instalar os pacotes pode ser visualizado no Apêndice B.

#### 4.3.1. Movimentação do robô

Para movimentar o robô P3-AT em diferentes ambientes, é necessário utilizar o *pacote RosAria*. Os dados relativos à base do robô, como, por exemplo, nível de tensão da bateria, controle de velocidade e aceleração, entre outros, podem ser obtidos ou manipulados através do *RosAria* (HEDGES, 2018).

### 4.3.2 Teleoperação e manipulação de objetos

O pacote *wiimote* permite a utilização do controle *Wiimote* para realização de teleoperação e manipulação de objetos com o robô P3-AT. O ROS utiliza *Bluetooth* para se comunicar com o dispositivo *Wiimote*.

#### 4.3.2.1 Teleoperação através do *Wiimote*

Para realizar a teleoperação através do *Wiimote*, foram configurados os botões descritos no quadro 2.

<b>Botão</b>	<b>Descrição</b>
<i>Up</i>	Avança
<i>Down</i>	Recua
<i>Right</i>	Rotaciona para direita
<i>Left</i>	Rotaciona para esquerda

Quadro 2 - Comandos de teleoperação

Fonte: A autoria própria.

O programa utilizado para realizar a teleoperação com *Wiimote* pode ser visualizado no Apêndice C.

#### 4.3.2.2 Manipulação de objetos através do *wiimote*

Para realizar a manipulação de objetos através do *Wiimote*, foram configurados os botões descritos no quadro 3.

<b>Botão</b>	<b>Descrição</b>
1	Aumenta o ângulo do servomotor 4
2	Diminui o ângulo do servomotor 4
+	Aumenta o ângulo do servomotor 6
-	Diminui o ângulo do servomotor 6

Quadro 3 - Comandos de manipulação

Fonte: Autoria própria.

O programa utilizado para realizar a manipulação de objetos com *Wiimote* pode ser visualizado no Apêndice C.

#### 4.3.3 Utilização do manipulador robótico

Para utilizar o manipulador robótico, é necessário integrar o Arduino com o ROS, através do pacote *rosserial\_arduino* (HASAN, 2018). O programa utilizado para movimentar o manipulador robótico pode ser visualizado no Apêndice D.

#### 4.3.4 Sistema de navegação

Para utilizar o sistema de navegação do robô P3-AT, foi necessário instalar e configurar o sensor LIDAR, assim como realizar a instalação dos pacotes *gmapping*, *hector\_mapping* e *navigation*. Assim, foi possível trabalhar com detecção de objetos e mapas de navegação.

#### 4.3.5 Sistema de visão

Para acessar a câmera do sensor Microsoft Kinect, foi preciso utilizar os pacotes *libfreenect* e *freenect\_camera* (KHANDELWAL, 2017). Para o reconhecimento de objetos, foi utilizado o pacote *ar\_track\_alvar*.

#### 4.4 TESTES

Para realizar os testes no robô P3-AT, foram desenvolvidos *launchers*, arquivos no formato XML, para permitir a execução de diversos nós ao mesmo tempo e realizar a configuração automática de portas utilizadas pelos dispositivos. Os *launchers* contêm todos os executáveis necessários para operar os seguintes itens: manipulação de objetos e navegação segura.



Figura 14 - Participação na LARC

Fonte: Autoria própria.



## 5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

A seguir são apresentados os resultados obtidos.

### 5.1 MANIPULAÇÃO DE OBJETOS

Utilizando o *Wii mote*, foi realizado a operação do manipulador robótico. Para objetos localizados na frente do robô, a uma altura aproximada de 927mm, e que não ultrapassem o peso de 0,1kg, foi possível realizar a sua manipulação.

Dos 6 servomotores que compõe o manipulador, foi realizado o controle dos ângulos de 2 servomotores através do *Wii mote*. Os demais servomotores trabalharam com ângulos fixos e determinados pelo programa que pode ser visto no Anexo D. Trabalhando com acréscimos ou decréscimos de 10 graus, os objetos ficaram firmes durante a manipulação, como pode ser observado na figura 14.



Figura 15 - Objeto sendo manipulado

Fonte: Autoria própria.

## 5.2 TELEOPERAÇÃO

Utilizando o *Wiimote*, foi possível realizar a teleoperação do robô. A velocidade do robô foi considerada segura, com deslocamentos aproximados de 50mm. Com o uso do sensor LIDAR não ocorrem colisões, permitindo a utilização do robô em ambientes domésticos.



Figura 16 - Robô sendo teleoperado

Fonte: Autoria própria.

## 5.3 NAVEGAÇÃO AUTÔNOMA

Por meio de teleoperação, foi possível mapear o laboratório LASER. Através do mapa construído, foi possível mandar o robô de um ponto A para um ponto B de forma autônoma. Com o sucesso obtido nesse teste, uma parte da tarefa *Help-me-carry* proposta pela *RoboCup@Home* pôde ser executada.

Nessa tarefa, é necessário carregar objetos de um lugar ao outro, simulando a situação em que um robô está ajudando um indivíduo a carregar os objetos do bagageiro de um carro para a dispensa localizada na cozinha, por exemplo.



Figura 17 - Robô no ponto de origem

Fonte: Autoria própria.

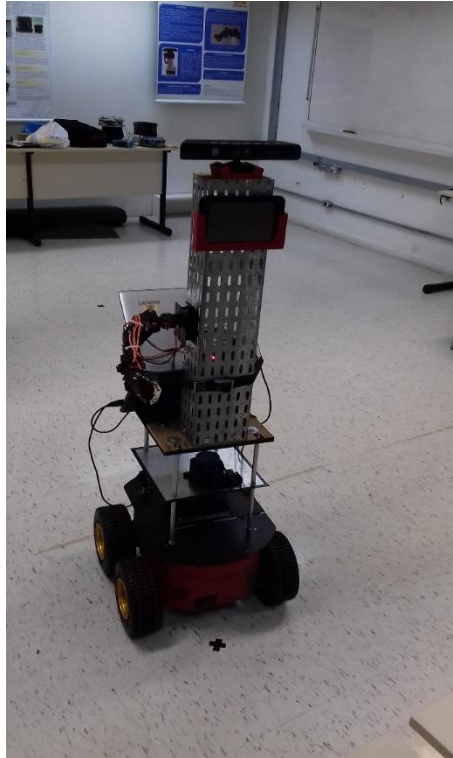


Figura 18 - Robô no ponto de chegada

Fonte: Autoria própria.

#### 5.4 MANIPULAÇÃO COM ARTAGS

Através do sistema de ARTag, foi possível realizar a manipulação de um objeto em situações especiais (objeto na frente do robô, a uma distância de 800mm do sensor Kinect e numa altura aproximada de 927mm). Nesse caso específico, o robô foi programado para se aproximar do objeto e realizar sua manipulação. O teste foi bem-sucedido.

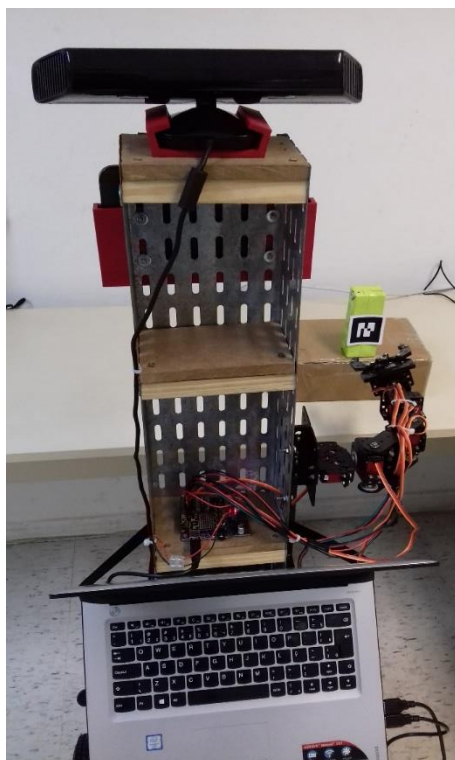


Figura 19 - Robô manipulando um objeto com ARTag

Fonte: Autoria própria.

## 6 CONSIDERAÇÕES FINAIS

Após realizar as adaptações necessárias na estrutura do robô Pioneer 3-AT, desenvolver um mecanismo de teleoperação, implementar os sistemas de navegação e visão no robô, e utilizar algoritmos para otimização de tarefas, visando a manipulação de objetos e navegação segura, o robô P3-AT teve a capacidade de cumprir uma parte da tarefa *Help-me-carry* proposta pela *RoboCup@Home*.

O robô P3-AT consegue manipular objetos em dois casos distintos. No primeiro caso, foi possível realizar a manipulação de objetos próximos ao robô através do *Wiimote*, desde que estejam a uma altura aproximada de 927mm. No segundo caso, através do reconhecimento de uma ARTag, o robô se desloca até o objeto e realiza sua manipulação. Nesse caso, existem limitações provenientes da resolução em que o Microsoft Kinect trabalha, pois em alguns testes o robô não foi capaz de reconhecer a ARTag e não realizou nenhuma operação.

O robô P3-AT se movimenta com velocidade de 50mm/s e consegue realizar o desvio de objetos com altura igual ou superior a 450mm. Portanto, em um ambiente doméstico realístico, o robô é capaz de detectar adultos e evitar colisões. No caso de crianças pequenas, que estão na fase de engatinhar, o robô ainda não tem condições de detectá-las, ou seja, nesse caso sua navegação não é indicada.

Os sistemas de navegação e visão são capazes de criar mapas bidimensionais e tridimensionais, mas existe a necessidade de se implementar um acelerômetro e um giroscópio para que sejam criados mapas com maior qualidade e menos falhas.

Uma questão a ser trabalhada envolve a trajetória desenvolvida pelo robô P3-AT quando navega de forma autônoma. Ao mandar o robô se deslocar de um ponto A para um ponto B, onde A e B representam os vértices opostos de um quadrado, o robô desenvolveu trajetórias diferentes. Em alguns testes desenvolveu uma trajetória retilínea, que é o resultado desejado, mas em outras, uma trajetória curvilínea.

Na situação presente, o robô P3-AT está apto a ser utilizado em ambientes domésticos, pois oferece segurança ao seu operador e possui a capacidade de realizar algumas tarefas, como manipulação de objetos e navegação segura. Existe apenas a restrição de não conseguir detectar objetos ou crianças com altura inferior a 450mm.

Trabalhar com robôs é desafiante e exige resiliência por parte do estudante. Muitos algoritmos, ao serem simulados com modelos virtuais, se mostram adequados e prontos para serem executados por modelos reais, mas quando são implementados em robôs, apresentam resultados inesperados. Muitas vezes a solução encontrada para melhorar os resultados é diferente da solução desenvolvida anteriormente.

A área da robótica está em expansão e apresenta diversos problemas a serem resolvidos, ou soluções a serem aperfeiçoadas. É uma área bastante promissora e atraente para quem busca desafios e se entusiasma com a busca diária por soluções cada vez melhores para os problemas existentes.

## REFERÊNCIAS

ADEPT. 2011. **Pioneer 3-AT**. Disponível em: <<https://www.generationrobots.com/media/Pioneer3AT-P3AT-RevA-datasheet.pdf>>. Acesso em: 29 set. 2018.

ARDUINO. **Arduino Shields**. Disponível em: <<https://www.arduino.cc/en/Main/arduinoShields>>. Acesso em: 29 set. 2018.

CRAIG, John J. **Introduction to Robotics: Mechanics and Control**. 3. ed. Prentice Hall. 2005.

DATTALO, Amanda. 2018. **Introduction**. Disponível em: <<http://wiki.ros.org/ROS/Introduction>>. Acesso em: 29 set. 2018.

DHOOD. 2017. **Ubuntu install of ROS Kinetic**. Disponível em: <<http://wiki.ros.org/kinetic/Installation/Ubuntu>>. Acesso em: 29 set. 2018.

FIALA, Mark. 2005. **ARTag, a fiducial marker system using digital techniques**. Disponível em: <<https://inside.mines.edu/~whoff/courses/EENG512/lectures/other/ARTag.pdf>>. Acesso em: 14 out. 2018.

GÉNÉRATION ROBOTS. **Pioneer 3-AT mobile robot**. Disponível em: <<https://www.generationrobots.com/en/402397-robot-mobile-pioneer-3-at.html>>. Acesso em: 30 set. 2018.

HASAN, Khan S. B. 2018. **Arduino IDE Setup**. Disponível em: <[http://wiki.ros.org/rosserial\\_arduino/Tutorials/Arduino IDE Setup](http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino%20IDE%20Setup)>. Acesso em: 30 set. 2018.

HEDGES, Reed. 2018. **ROSARIA**. Disponível em: <<http://wiki.ros.org/ROSARIA>>. Acesso em: 06 out. 2018.

HEDGES, Reed. 2018. **How to use ROSARIA**. Disponível em: <[http://wiki.ros.org/ROSARIA/Tutorials/How to use ROSARIA](http://wiki.ros.org/ROSARIA/Tutorials/How%20to%20use%20ROSARIA)>. Acesso em: 06 out. 2018.



JOSEPH, Lentin. **Mastering ROS for Robotics Programming**. Packt Publishing, 2015.

KHANDELWAL, Piyush. 2017. **libfreenect**. Disponível em: <<http://wiki.ros.org/libfreenect>>. Acesso em: 14 out. 2018.

MARGUEDAS. 2017. **Installing and Configuring Your ROS Environment**. Disponível em: <<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>>. Acesso em: 07 out. 2018.

MEDEIROS, Adelardo A. D. et al. **Robótica Móvel**. Rio de Janeiro: LTC, 2014.

MICROSOFT DOCS. **Kinect Sensor**. Disponível em: <[https://docs.microsoft.com/en-us/previous-versions/microsoft-robotics/hh438998\(v%3dmsdn.10\)](https://docs.microsoft.com/en-us/previous-versions/microsoft-robotics/hh438998(v%3dmsdn.10))>. Acesso em: 07 out. 2018.

MOBILEROBOTS. 2006. **Pioneer 3 Operations Manual with MobileRobots Exclusive Advanced Robot Control & Operations Software**. Disponível em: <[https://www.inf.ufrgs.br/~prestes/Courses/Robotics/manual\\_pioneer.pdf](https://www.inf.ufrgs.br/~prestes/Courses/Robotics/manual_pioneer.pdf)>. Acesso em: 14 out. 2018.

MONK, Simon. **Programação com Arduino: começando com Sketches**. Porto Alegre: Bookman. 2013.

NOAA. 2012. **Lidar 101: An Introduction to Lidar Technology, Data, and Applications**. Disponível em: <<https://coast.noaa.gov/data/digitalcoast/pdf/lidar-101.pdf>>. Acesso em: 13 out. 2018.

OLIVEIRA, André S. et al. 2017. **Uso do robô Pioneer LX para tarefas domésticas**. Disponível em: <<http://sistemaolimpo.org/midias/uploads/d2d59df8411f9bf3af00bb222d6316dd.pdf>>. Acesso em: 13 out. 2018.

RASCON, Caleb et al. **Robocup@Home 2018: Rule and regulations**. Disponível em: <[http://www.robocupathome.org/rules/2018\\_rulebook.pdf](http://www.robocupathome.org/rules/2018_rulebook.pdf)>. Acesso em: 14 out. 2018.

ROSÁRIO, João M. **Princípios de mecatrônica**. São Paulo: Prentice-Hall, 2005.

ROS.ORG. **About ROS**. Disponível em: <<http://www.ros.org/about-ros/>>. Acesso em: 14 out. 2018.

SAGAN, Carl. **Cosmos**. Gradiva, 2009.

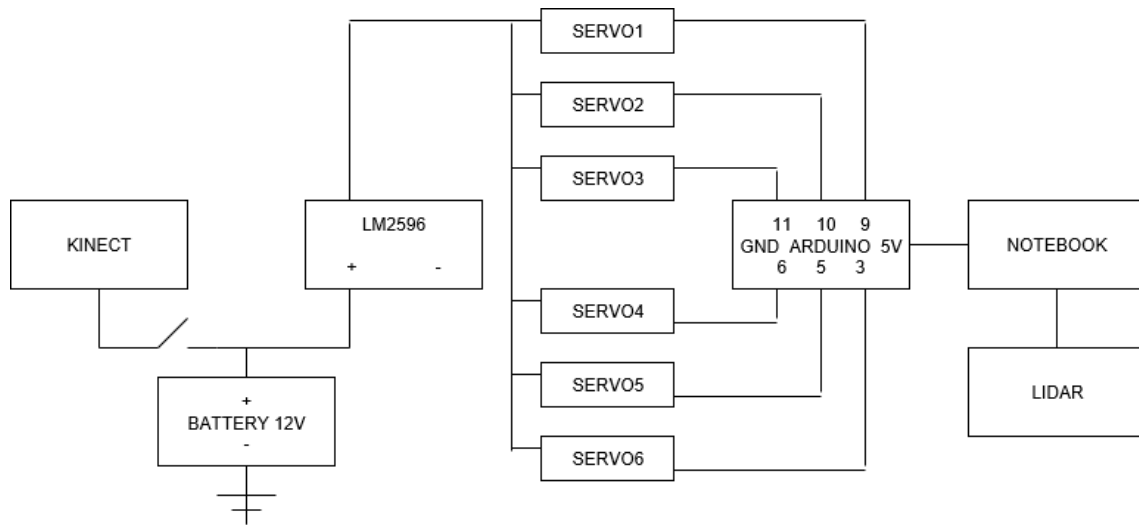
SAITO, Isaac. **ar\_track\_alvar**. 2016. Disponível em: <[http://wiki.ros.org/ar\\_track\\_alvar](http://wiki.ros.org/ar_track_alvar)>. Acesso em: 14 out. 2018.

WEBOTS USER GUIDE. **Adept's Pioneer 3-AT**. Disponível em: <<https://cyberbotics.com/doc/guide/pioneer-3at>>. Acesso em: 14 out. 2018.

WISE, Melonee. **Starting Wiimote Node**. 2011. Disponível em: <<http://wiki.ros.org/wiimote/Tutorials/StartingWiimoteNode>>. Acesso em: 14 out. 2018.

YDLIDAR. **YDLIDAR X4 USER MANUAL**. Disponível em: <<http://www.ydlidar.com/download/4>>. Acesso em: 06 out. 2018.

## APÊNDICE A - DIAGRAMA ELÉTRICO



## APÊNDICE B - CONFIGURAÇÃO

### INSTALAÇÃO DO ROS KINETIC

Para instalar o ROS *Kinetic*, foram utilizados os tutoriais obtidos em (DHOOD, 2017) e (MARGUEDAS, 2017). Em um terminal, foram executados os seguintes comandos:

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc)
main" > /etc/apt/sources.list.d/ros-latest.list'
$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
421C365BD9FF1F717815A3895523BAEED01FA116
$ sudo apt-get update
$ sudo apt-get install ros-kinetic-desktop-full
$ sudo rosdep init
$ rosdep update
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
$ sudo apt-get install python-rosinstall python-rosinstall-generator python-
wstool build-essential
```

Para finalizar a instalação do ROS *Kinetic*, foi criado um *catkin workspace*:

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
$ source devel/setup.bash
$ echo $ROS_PACKAGE_PATH
/home/youruser/catkin_ws/src:/opt/ros/kinetic/share
```

### INSTALAÇÃO DO PACOTE ROSARIA

Para instalar o pacote *RosAria*, foram executados os seguintes comandos no terminal (HEDGES, 2018).

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/amor-ros-pkg/rosaria.git
$ sudo dpkg -i libaria_2.9.4+ubuntu16_amd64.deb
$ cd /usr/local/Aria
$ make clean
$ make -j4
```

## INSTALAÇÃO DO PACOTE WIIMOTE

Para instalar o pacote *wiimote*, foram executados os seguintes comandos no terminal (WISE, 2011).

```
$ cd ~/catkin_ws/src
$ rosdep install wiimote
$ rosmake wiimote
$ sudo apt-get install libcwiid-dev
```

Em um terminal, foi executado o *roscore*, para utilização do ROS e do comando *roslaunch*:

```
$ roscore
```

Em seguida, numa nova aba, foram executados os seguintes comandos:

```
$ cd ~/catkin_ws/src
$ sudo apt-get install ros-kinetic-rosserial-arduino
$ sudo apt-get install ros-kinetic-rosserial
$ git clone https://github.com/ros-drivers/rosserial.git
$ cd ~/sketchbook/libraries
$ rm -rf ros_lib
$ roslaunch rosserial_arduino make_libraries.py .
```

## INSTALAÇÃO E CONFIGURAÇÃO DO SENSOR LIDAR

Para utilizar o sensor LIDAR, foram executados os seguintes comandos no terminal (YDLIDAR, 2018).

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/EAIBOT/ydlidar
$ cd ydlidar/startup
$ sudo chmod 0777 *
$ cd launch
$ gedit lidar.launch
```

Por fim, foi necessário alterar, no arquivo *lidar.launch*, o parâmetro “*port*” para “*/dev/ttyUSB0*” e o parâmetro “*baudrate*” para “*128000*”.

## INSTALAÇÃO DOS PACOTES DO SISTEMA DE NAVEGAÇÃO

Para utilizar o sistema de navegação do robô, foram executados os seguintes comandos no terminal.

```
$ cd ~/catkin_ws/src  
$ sudo apt-get install ros-kinetic-gmapping  
$ sudo apt-get install ros-kinetic-hector-mapping  
$ sudo apt-get install ros-kinetic-navigation*
```

## INSTALAÇÃO DOS PACOTES DO SISTEMA DE VISÃO

Para utilizar o sistema de visão do robô, foram executados os seguintes comandos no terminal (KHANDELWAL, 2017).

```
$ cd ~/catkin_ws/src  
$ git clone https://github.com/ros-drivers/libfreenect.git  
$ git clone https://github.com/ros-drivers/freenect_stack.git
```

Para trabalhar com reconhecimento de ARTag, foi necessário instalar o pacote `ar_track_alvar`.

```
$ cd ~/catkin_ws/src  
$ sudo apt-get install ros-kinetic-ar-track-alvar
```

## APÊNDICE C – TELEOPERAÇÃO E MANIPULAÇÃO COM WIIMOTE

```
/* Programa: Teleoperação e Manipulação com Wiimote
Autor: Alexandre Guilherme Damm dos Santos */

#include <iostream>
#include <string.h>
#include "geometry_msgs/Twist.h"
#include "ros/ros.h"
#include "sensor_msgs/Joy.h"
#include "std_msgs/Float32.h"
#include "std_msgs/Int32.h"
#include "std_msgs/UInt16.h"
#include "std_msgs/String.h"

using namespace std;

int button_right;
int button_left;
int button_up;
int button_down;

int button_one;
int button_two;
int button_plus;
int button_minus;

float axes[2];
float vLinear = 0.1;
float vAngular = 0.1;

int angle4 = 10;
int angle6 = 1;

void PoseCallback(const sensor_msgs::Joy::ConstPtr& msg){

    button_right = msg->buttons[6];
    button_left = msg->buttons[7];
    button_up = msg->buttons[8];
    button_down = msg->buttons[9];

    button_one = msg->buttons[0];
    button_two = msg->buttons[1];
    button_plus = msg->buttons[4];
    button_minus = msg->buttons[5];

    axes[0] = msg->axes[0];
    axes[1] = msg->axes[1];
```

```

}

int main(int argc, char **argv) {

    ros::init(argc, argv, "teleoperacao");

    ros::NodeHandle n;

    ros::Subscriber Wii_sub = n.subscribe("/joy", 100, PoseCallback); // 1 ->
    1000

    ros::Publisher P3AT_pub =
    n.advertise<geometry_msgs::Twist>("RosAria/cmd_vel", 1000);

    ros::Publisher RArm_pub4 = n.advertise<std_msgs::UInt16>("servo4",
    1000);

    ros::Publisher RArm_pub6 = n.advertise<std_msgs::UInt16>("servo6",
    1000);

    ros::Rate loop_rate(10);

    geometry_msgs::Twist msg;

    std_msgs::UInt16 ang4;

    std_msgs::UInt16 ang6;

    while(ros::ok()){

        msg.linear.x = axes[1]/20;
        msg.angular.z = -axes[0]/20;

        if (button_right == 1){
            ROS_INFO("P3AT RIGHT");
            msg.linear.x = 0.00;
            msg.angular.z = vAngular;
            P3AT_pub.publish(msg);}

        if (button_left == 1){
            ROS_INFO("P3AT LEFT");
            msg.linear.x = 0;
            msg.angular.z = -vAngular;
            P3AT_pub.publish(msg);}

        if (button_up == 1){
            ROS_INFO("P3AT UP");
            msg.linear.x = vLinear;
            msg.angular.z = 0;
            P3AT_pub.publish(msg);}
    }
}

```



```
if (button_down == 1){
  ROS_INFO("P3AT DOWN");
  msg.linear.x = -vLinear;
  msg.angular.z = 0;
  P3AT_pub.publish(msg);}

if (button_one == 1){
  if (ang4.data != 180){
    ROS_INFO("SERVO 4 INCREASE ANGLE");
    ang4.data = ang4.data + angle4;
    RArm_pub4.publish(ang4);}
}

if (button_two == 1){
  if (ang6.data != 0){
    ROS_INFO("SERVO 4 DECREASE ANGLE");
    ang4.data = ang4.data - angle4;
    RArm_pub4.publish(ang4);}
}

if (button_plus == 1){
  if (ang6.data != 180){
    ROS_INFO("SERVO 6 INCREASE ANGLE");
    ang6.data = ang6.data + angle6;
    RArm_pub6.publish(ang6);}
}

if (button_minus == 1){
  if (ang6.data != 0){
    ROS_INFO("SERVO 6 DECREASE ANGLE");
    ang6.data = ang6.data - angle6;
    RArm_pub6.publish(ang6);}
}

loop_rate.sleep();

ros::spinOnce();

}

return 0;

}
```

## APÊNDICE D - MANIPULADOR ROBÓTICO

```

/* Programa: Controle dos ângulos do manipulador robótico
Autor: Alexandre Guilherme Damm dos Santos */

#if defined(ARDUINO) && ARDUINO >= 100
  #include "Arduino.h"
#else
  #include <WProgram.h>
#endif

#include <VarSpeedServo.h>
#include <ros.h>
#include <std_msgs/UInt16.h>

ros::NodeHandle nh;

VarSpeedServo servo1; // Base
VarSpeedServo servo2;
VarSpeedServo servo3;
VarSpeedServo servo4;
VarSpeedServo servo5;
VarSpeedServo servo6; // Gripper

int i;
int aux=0;

/* void servo1_cb( const std_msgs::UInt16& cmd_msg){
  servo1.slowmove(cmd_msg.data, 10);} // Set servo angle, should be from 0-
180

void servo2_cb( const std_msgs::UInt16& cmd_msg){
  servo2.slowmove(cmd_msg.data, 10);} // Set servo angle, should be from 0-
180

void servo3_cb( const std_msgs::UInt16& cmd_msg){
  servo3.slowmove(cmd_msg.data, 10);} // Set servo angle, should be from 0-
180 */

void servo4_cb( const std_msgs::UInt16& cmd_msg){
  servo4.slowmove(cmd_msg.data, 10);} // Set servo angle, should be from 0-
180

/* void servo5_cb( const std_msgs::UInt16& cmd_msg){
  servo5.slowmove(cmd_msg.data, 10);} // Set servo angle, should be from 0-
180 */

void servo6_cb( const std_msgs::UInt16& cmd_msg){

```

```

servo6.slowmove(cmd_msg.data, 10);} // Set servo angle, should be from 0-180}

/* ros::Subscriber<std_msgs::UInt16> sub1("servo1", servo1_cb);
ros::Subscriber<std_msgs::UInt16> sub2("servo2", servo2_cb);
ros::Subscriber<std_msgs::UInt16> sub3("servo3", servo3_cb); */
ros::Subscriber<std_msgs::UInt16> sub4("servo4", servo4_cb);
/* ros::Subscriber<std_msgs::UInt16> sub5("servo5", servo5_cb); */
ros::Subscriber<std_msgs::UInt16> sub6("servo6", servo6_cb);

void setup(){

  nh.initNode();
/* nh.subscribe(sub1);
nh.subscribe(sub2);
nh.subscribe(sub3); */
nh.subscribe(sub4);
/* nh.subscribe(sub5); */
nh.subscribe(sub6);

  servo1.attach(11); // Attach it to pin 11
  servo2.attach(10); // Attach it to pin 10
  servo3.attach(9); // Attach it to pin 9
  servo4.attach(6); // Attach it to pin 6
  servo5.attach(5); // Attach it to pin 5
  servo6.attach(3); // Attach it to pin 3

}

void loop(){

  if (aux == 0){
    for (i=0; i<6; i++){
      if(i=1){
        servo1.slowmove(90, 10); // Set servo angle
        delay(2000);}
      if(i=2){
        servo2.slowmove(30, 10); // Set servo angle
        delay(2000);}
      if(i=3){
        servo3.slowmove(150, 10); // Set servo angle
        delay(2000);}
      if(i=4){
        servo4.slowmove(90, 10); // Set servo angle
        delay(2000);}
    }
  }
}

```

```
if(i=5){
  servo5.slowmove(0, 10); // Set servo angle
  delay(2000);}
if(i=6){
  servo6.slowmove(0, 10); // Set servo angle
  delay(2000);
  aux=1;}
}
}

nh.spinOnce();
delay(1);

}
```