

**UNIVERSIDADE FEDERAL DE VIÇOSA  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**VINÍCIUS PACHECO BACHETI**

**DESENVOLVIMENTO DE UMA PLATAFORMA  
ENTRE O ROBÔ PIONEER 3-DX E O MATLAB**

**VIÇOSA  
2015**

**VINÍCIUS PACHECO BACHETI**

**DESENVOLVIMENTO DE UMA PLATAFORMA  
ENTRE O ROBÔ PIONEER 3-DX E O MATLAB**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 - Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Alexandre Santos Brandão

VIÇOSA  
2015



**VINÍCIUS PACHECO BACHETI**

**DESENVOLVIMENTO DE UMA PLATAFORMA ENTRE O ROBÔ  
PIONEER 3-DX E O MATLAB**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 - Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 02 de Dezembro de 2015.

**COMISSÃO EXAMINADORA**

---

Prof. Dr. Alexandre Santos Brandão - Orientador  
Universidade Federal de Viçosa

---

Prof. Dr. Tarcísio de Assunção Pizziolo - Membro  
Universidade Federal de Viçosa

---

M.Sc. André Luís de Carvalho Mendes - Membro  
Universidade Federal de Viçosa

*Dedico esta monografia ao meu pai Gilmar, minha mãe Luciane  
e aos meus irmãos Pedro e Gilmara*

# *Agradecimentos*

Quero agradecer à minha família, Gilmar, Luciane, Pedro e Gilmara por todo apoio, incentivo e amor que deram-me durante toda a vida, que foram fundamentais para a minha formação pessoal e profissional. À minha namorada, Jaquelina, por estar ao meu lado nesta reta final, dando-me forças para finalizar esta etapa com sucesso.

Ao meu orientador, Alexandre Brandão, o Timótiu, por ter me guiado ao longo de toda a graduação, além, é claro, da amizade genuína, que ficará para a vida inteira. Ainda estou te devendo algumas viagens!

Aos companheiros do NERO, que sempre me apoiaram e ajudaram quando eu precisei, e viveram momentos inesquecíveis comigo na equipe de futebol de robôs.

Aos meus amigos, por todo o companheirismo, risadas, e por estarem ao meu lado durante todos esses anos, passando por dificuldades e vitórias juntos. Não chegaria onde cheguei sem cada um de vocês, e sua amizade será o bem mais precioso que levarei de Viçosa. Nunca poderei expressar o suficiente o quanto vocês mudaram minha vida.

Dedico esta conquista a todos vocês!

*“Like all young men I set out to be a genius,  
but mercifully laughter intervened.”*

*Lawrence Durrell*

# *Resumo*

Este trabalho detalha o funcionamento de um software desenvolvido em Matlab<sup>©</sup> que faz a interface entre um usuário e o robô Pioneer 3-DX, Mobile Robots Inc. Para utilizar o programa o usuário deve ter apenas conhecimentos de Matlab<sup>©</sup>, não sendo necessário se preocupar com qualquer outro tipo de linguagem.

Este trabalho tem como objetivo o desenvolvimento de uma plataforma de testes de controladores aplicados à robótica móvel. O usuário deve informar várias posturas desejadas de um ambiente estruturado, as quais devem ser alcançadas através dos controladores implementados. Além disso, o software possibilita a geração de gráficos das variáveis de estado do robô e dos sinais de controle, assim como a exportação desses dados para arquivos de texto.

Para validação, dois controladores foram implementados e testados em simulação inicialmente e experimentalmente no software. O primeiro controlador é capaz de levar o robô unicycle Pioneer 3DX a uma posição desejada, enquanto o segundo controlador é capaz de controlar tanto a posição quanto a orientação durante a navegação.

É demonstrado no fim deste trabalho uma comparação entre os dois controladores, utilizando-se somente os gráficos gerados pelo próprio software. Conforme proposto, o programa forneceu ao usuário as ferramentas necessárias à implementação do controlador e análise dos resultados de sua navegação, de maneira simples, tendo como base programação apenas em Matlab<sup>©</sup>.

# *Abstract*

This paper details the operation of a software developed in Matlab<sup>©</sup>, responsible for the interface between an user and the robot Pioneer 3-DX, Mobile Robots Inc. To be able to use the software it is only required of the user knowledge in Matlab<sup>©</sup>, not being necessary to be concerned about any additional programming languages.

The goal is the development of an applied controllers testing platform. The user must inform a set of desired poses within a structured environment, which must be reached by the robot using the implemented controllers. Furthermore, the software displays graphs from the state variables of the robot and the control signals, besides exporting the navigation data to text files.

In order to validate the platform, two controllers were implemented and then tested initially in simulation and later experimentally on the software. The first controller is capable of taking the Pioneer 3-DX to a desired position, while the second one is capable of controlling both the position and the orientation during the navigation.

A comparison between the two controllers is demonstrated at the end of this work, using only the graphs generated by the software built. As proposed, the platform supplied the user with the necessary tools to the implementation of a controller and analysis of its navigation, being based only in Matlab<sup>©</sup> programming language.

# *Sumário*

## **Lista de Figuras**

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	O Robô Pioneer 3-DX . . . . .	13
1.2	Odometria . . . . .	14
1.3	Objetivos do Trabalho . . . . .	16
1.4	Estrutura do Trabalho . . . . .	17
<b>2</b>	<b>Modelagem e Controle</b>	<b>18</b>
2.1	Modelo Cinemático . . . . .	18
2.2	Controladores . . . . .	20
2.2.1	Controlador por Modelo Cinemático Inverso . . . . .	21
2.2.2	Controlador Chaveado de Posição com Orientação Final . . . . .	21
<b>3</b>	<b>Interface Matlab-Pioneer</b>	<b>25</b>
3.1	Mapa . . . . .	26
3.2	Conectar Pioneer . . . . .	26
3.3	Postura do Pioneer . . . . .	26
3.3.1	Nova Postura . . . . .	26
3.3.2	Método de Entrada . . . . .	26
3.3.3	Próxima Postura, Cancelar Postura e Finalizar escolha de posturas	27
3.4	Escolha dos limites . . . . .	27
3.5	Escolha do controlador . . . . .	28

3.6	Botões de Navegação e Outros . . . . .	28
3.6.1	Rodar . . . . .	28
3.6.2	Stop . . . . .	28
3.6.3	Home . . . . .	29
3.6.4	Definir Nova Home . . . . .	29
3.6.5	Ligar Sonar e Configurar Sonar . . . . .	29
3.6.6	Exibir Gráficos e Exportar Dados . . . . .	30
<b>4</b>	<b>Resultados e Discussões</b>	<b>31</b>
4.1	Comparação de Sinais de Controle para uma única postura desejada . . . . .	31
4.2	Comparação de Sinais de Controle para várias posturas desejadas . . . . .	34
4.3	Comparação de diferentes valores de ganho . . . . .	37
<b>5</b>	<b>Conclusões</b>	<b>42</b>
	<b>Referências</b>	<b>43</b>

# *Lista de Figuras*

1	Robôs terrestres com rodas. . . . .	12
2	Pioneer 3-DX produzido pela Mobile Robots Inc. . . . .	14
3	Modelo da malha de controle simplificada do Pioneer 3-DX. . . . .	14
4	“Elipses de erro”indicando o aumento da incerteza na posição . . . . .	15
5	Sistemas de coordenadas inercial e do robô. . . . .	18
6	Sistemas de coordenadas inercial e do robô com busca a uma postura determinada. . . . .	20
7	Tela principal do Software . . . . .	25
8	Conjunto de sonares do Pioneer P3-DX . . . . .	29
9	Caminho feito pelo Pioneer saindo da origem e indo até a posição $X_d = [3 \ 2]^T$ com o controlador baseado em cinemática inversa. . . . .	31
10	Evolução da posição e orientação do Pioneer com o controlador baseado em cinemática inversa. . . . .	32
11	Comparação entre as ações de controle enviadas ao Pioneer e as executadas com o controlador baseado em cinemática inversa. . . . .	33
12	Caminho feito pelo Pioneer saindo da origem e indo até a posição $X_d = [3 \ 2 - \frac{3}{4}\pi]^T$ com o controlador chaveado de posição com orientação final. . . . .	33
13	Evolução da posição e orientação do Pioneer com o controlador chaveado de posição com orientação final. . . . .	34
14	Comparação entre as ações de controle enviadas ao Pioneer e as executadas com o controlador chaveado de posição com orientação final. . . . .	35
15	Comparação entre os valores de $x$ e $y$ assumidos pelo Pioneer utilizando-se os dois controladores diferentes. . . . .	36

16	Comparação entre as orientações assumidas pelo Pioneer utilizando-se os dois controladores diferentes. . . . .	37
17	Comparação entre as Ações de controle enviadas e executadas pelo Pioneer utilizando-se os dois controladores diferentes. . . . .	38
18	Rota feita pelo Pioneer com diferentes valores de ganho. . . . .	39
19	Evolução da posição com diferentes valores de ganho. . . . .	40
20	Evolução da posição com diferentes valores de ganho. . . . .	41

# 1 *Introdução*

As mais diversas tarefas podem ser realizadas por robôs. Antes presentes apenas em indústrias, eles começaram há alguns anos a ganhar espaço em outros tipos de tarefa como limpeza doméstica ou de galpões industriais, exploração de regiões inóspitas, uso militar, uso na agricultura, entretenimento e muitos outros [1]. Outros tipos de atividade incluem a substituição do homem em tarefas repetitivas ou suscetíveis a erro [2].

Como exemplo dessas aplicações, pode-se citar o robô Groundhog desenvolvido na Universidade Carnegie Mellon com o objetivo de explorar minas abandonadas [3], oferecendo alto risco para seres humanos, e o primeiro robô desenvolvido pela NASA para exploração do planeta Marte denominado de Sojourner [4]. Em [5], é feita a análise do uso de robôs em cirurgias na cabeça e pescoço, mostrando que eles são pelo menos igualmente eficazes.

Os robôs podem ser classificados em duas categorias distintas: os robôs manipuladores e os robôs móveis. A principal diferença é a de que, mesmo tendo mobilidade em suas ações, os manipuladores estão restritos pelas suas dimensões físicas, definindo o que se conhece como espaço de trabalho de um manipulador. Em contrapartida, os robôs móveis tem mais liberdade em seus movimentos, podendo navegar em ambientes de grandes dimensões [6].

Os robôs móveis podem ser classificados de acordo com sua anatomia, ou seja, em qual ambiente eles são preparados para navegar, podendo ser terrestres, aquáticos ou aéreos. Entretanto, os princípios gerais continuam os mesmos, quer o robô seja dirigido, pilotado, navegado ou submerso. Todos os veículos autônomos devem navegar dentro de um ambiente. Para conseguir isso, o robô deve ser capaz de “ver” seu ambiente, interpretar as informações sensoriais para refinar seu conhecimento de sua posição e estrutura do ambiente e planejar uma rota de um ponto inicial até um ponto final na possível presença de obstáculos conhecidos ou desconhecidos [7].

Os robôs terrestres estão entre os mais difundidos e são classificados quanto ao atuador

que utilizam para se locomover: rodas, esteiras ou patas.

Cada escolha deve ser pesada em relação ao tipo de ambiente no qual o robô deve navegar. Rodas são mais fáceis de se implementar, mas são limitadas no que tange a transposição de obstáculos. Esteiras são melhores nesse tipo de tarefa, porém gastam muita energia com o atrito no piso. Os robôs com pata, ou pernas, superam as limitações acima e recentemente estão recebendo muita atenção da comunidade científica [8, 9], porém ainda sofrem de uma grande dificuldade no projeto de controladores para coordenação do movimento das patas.

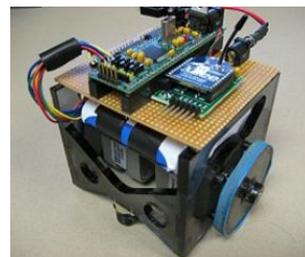
Entre os robôs terrestres de rodas destacam-se três tipos. Os robôs omnidirecionais podem executar movimentos em qualquer direção do plano XY. Frequentemente, eles são dotados de rodas especiais que permitem o movimento em direção perpendicular ao sentido de giro do robô. Os robôs do tipo *car-like*, como o nome sugere, têm a configuração semelhante a um automóvel convencional. Eles possuem um par de rodas montadas sobre o mesmo eixo, e tracionadas simultaneamente, que ficam localizadas normalmente na parte traseira do robô. Um outro par de rodas, localizado na dianteira, define a orientação do robô durante a navegação. Finalmente, os robôs do tipo unicycle, também chamados de robôs móveis de tração diferencial, possuem duas rodas montadas sobre um mesmo eixo virtual, que são acionadas independentemente permitindo que o robô rotacione sobre o próprio eixo. A Figura 1 mostra os três tipos de robô descritos acima.



(a) Robô omnidirecional.



(b) Robô car-like.



(c) Robô unicycle.

Figura 1: Robôs terrestres com rodas.

Os ambientes, por sua parte, também podem ser classificados. A classificação mais geral os divide em três tipos: ambientes estruturados, semi-estruturados e não estruturados. A diferença entre eles tem a ver com a mobilidade dos objetos no espaço de trabalho do robô.

Aqueles considerados estruturados, não apresenta mudança de postura de qualquer objeto no espaço, logo o robô conhece previamente a posição dos obstáculos, os quais devem ser estáticos. Se é assumido que um robô vai trabalhar em um ambiente fortemente

estruturado, o poder computacional utilizado para a tomada de decisões é radicalmente diminuído [10]. Já um ambiente semi-estruturado, uma certa mobilidade de seus objetos é permitida, em padrões conhecidos ou previsíveis. Escritórios e residências podem ser tomados como exemplo. Por fim, em ambientes não estruturados, não há qualquer tipo de restrição ou conhecimento de movimentos dos objetos presentes no espaço de trabalho do robô, tornando o planejamento de movimentos muito mais complexo.

Outro aspecto importante é o sistema de visão dos robôs. Diferente dos seres humanos que já nascem com olhos e a capacidade de interpretar sinais visuais, os robôs podem ser considerados cegos. Para conseguir criar um mapa do ambiente que os cerca e sua localização nele, os robôs precisam contar com algum sistema, que o auxilie em sua localização. Sonares, lasers, satélites, câmeras e odometria são alguns dos sensores mais comuns [11].

Como se pode perceber a navegação de robôs móveis depende de vários fatores, que devem ser definidos e adequados de acordo com a tarefa que o robô deve executar, levando em consideração desde os desafios propostos até os custos das soluções e sua viabilidade.

## 1.1 O Robô Pioneer 3-DX

O robô Pioneer 3-DX, produzido pela empresa Mobile Robots Inc., ilustrado na Figura 2, é um dos mais populares robôs de pesquisa no mundo. Sua versatilidade, confiabilidade e durabilidade o fizeram referência na pesquisa de robôs móveis. O Pioneer, robô do tipo terrestre unicycle, vem equipado com um planejador de trajetórias dedicado, com retro-alimentação por encoder, com sonares e é totalmente reprogramável.

O microcontrolador interno monitora a informação de posição fornecida pelos dois encoders de cada roda e gera comandos PWM, que são amplificados e aplicados independentemente nas rodas, como mostra a Figura 3. As duas malhas de controle independentes são controladas pelo mesmo algoritmo PID [12]. Além disso, circuitos adicionais, não mostrados na Figura 3, são relacionados ao conjunto de sonares.

O microcontrolador embarcado do Pioneer também fornece a posição e orientação estimada do Pioneer, estado da bateria e leitura dos sonares, além de outras informações.

Como mencionado, o Pioneer é um robô extremamente popular no mundo acadêmico. Várias pesquisas o utilizam pelos mais diversos motivos. Em [13], o Pioneer é usado na construção de um andador robótico para ajudar pessoas com pouca mobilidade nas



Figura 2: Pioneer 3-DX produzido pela Mobile Robots Inc.

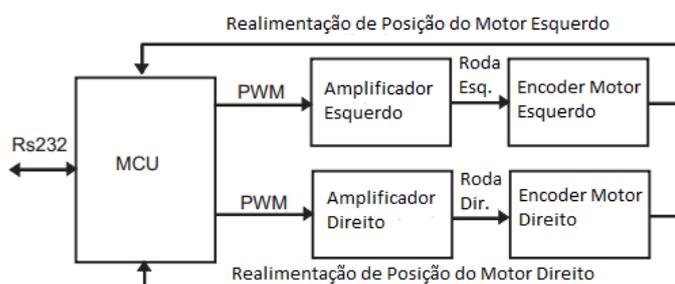


Figura 3: Modelo da malha de controle simplificada do Pioneer 3-DX.

pernas. Já [14], faz um controlador dinâmico adaptativo para seguimento de trajetória. Em [15], tem-se a teleoperação de um Pioneer utilizando a captação dos movimentos do giroscópio de um celular Android.

Daqui em diante, quando um robô for mencionado no texto, o leitor deve associá-lo ao Pioneer 3-DX.

## 1.2 Odometria

A odometria é uma técnica de posicionamento e localização relativa, que determinar a postura estimada de um veículo baseado no deslocamento de suas rodas.

Os encoders óticos, sensores mais comuns associados à odometria, funcionam baseados na transmissão e recepção de feixes de luz que cruzam um disco perfurado girando acoplado ao eixo da roda. Assim, a velocidade angular da roda é facilmente convertida

em velocidade linear. Por fim, a posição do robô é então determinada pelo deslocamento incremental das rodas.

Entretanto, por depender desta integração do deslocamento incremental, a odometria está vulnerável ao acúmulo de erros, que aumenta conforme a distância percorrida pelo robô vai ficando maior, o que impõe uma limitação para grandes trajetos [11]. A Figura 4 mostra a progressão das “Elipses de erro”, que representam a incerteza no posicionamento do robô.

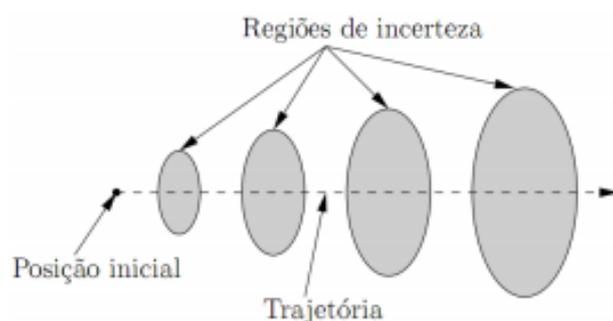


Figura 4: “Elipses de erro” indicando o aumento da incerteza na posição

Segundo [16], os erros de odometria são divididos em dois grupos. Os sistemáticos são originados pelo próprio modelo ou objeto de estudo. Diferença no diâmetro das rodas, desalinhamento dos eixos de direção, resolução finita dos sensores e taxa de amostragem limitada são todas causas sistemáticas de erro.

Os erros não-sistemáticos vêm de fatores externos, como terrenos irregulares, objetos inesperados no trajeto e deslizamento das rodas. Como são fatores imprevisíveis, é difícil estabelecer métodos de correção. Uma solução seria melhorar a estrutura física do robô, diminuindo o efeito destes elementos.

Os autores em [16] e [17] destacam alguns métodos de correção (ou minimização) desses erros. O primeiro deles é trabalhar com equipamentos melhores dentro do possível, entretanto uma melhora em qualidade indica um aumento dos custos. Uma calibração precisa também é muito importante.

O primeiro teste sugerido é chamado de caminho quadrado unidirecional, onde o robô é programado para seguir um caminho quadrado de 4x4m, obviamente dotado de quatro ângulos retos. No final, compara-se o ponto de chegada com o de partida, estimando-se o erro através do desvio medido. Uma versão mais completa desse teste envolve a realização do percurso nas duas direções, horária e anti-horária.

Outra alternativa é usar dois robôs movendo-se lado a lado, em uma distância fixa, onde um servirá de referência ao outro. A variação de distância entre eles pode então ser usada para imediatamente corrigir o desvio, pois dispendo-se de dois ou mais sinais ruins, pode-se utilizá-los em conjunto para obter um sinal melhor.

Uma outra solução comum é misturar mais de um método de posicionamento para que, em conjunto, eles possam eliminar os erros um do outro e transpor limitações individuais do método. Localização por GPS é usada em [18], e marcos visuais, que podem ser naturais ou artificiais, são usados em [19].

Como nota ao leitor, os erros de odometria serão desconsiderados neste trabalho, pois fogem ao escopo desta proposta. Uma adição futura poderia ser a inserção de métodos de correção dos erros de odometria.

## 1.3 Objetivos do Trabalho

Tendo em mente que a linguagem utilizada no Matlab<sup>®</sup> é amplamente difundida na engenharia, o objetivo geral deste trabalho é fazer uma interface entre o robô móvel terrestre Pioneer 3-DX e o usuário através de uma plataforma utilizando esta linguagem. Para tarefas mais simples, como enviar o robô a posturas desejadas, o usuário não precisará de nenhum conhecimento de linguagem de programação. Para outras tarefas, como a inserção de novos controladores, um conhecimento mínimo de Matlab<sup>®</sup> é requerido, mas ainda assim, muito mais simples que aprender a linguagem específica utilizada no Pioneer 3-DX.

Levando isto em consideração, existem também alguns objetivos específicos:

- Construir uma interface intuitiva e amigável, que seja robusta o suficiente para sofrer modificações futuras, de modo a atender tanto usuários iniciantes quanto avançados.
- Implementar dois controladores na interface de maneira que o usuário possa escolher se quer que o robô assuma uma posição final ou uma postura final.
- Dar ao usuário ferramentas necessárias para analisar o comportamento dos controladores. Elas serão em forma de gráficos, os quais irão delinear a rota feita pelo robô, assim como posições, orientações e ações de controle tanto desejadas, quanto as executadas.

## 1.4 Estrutura do Trabalho

O trabalho está estruturado em cinco capítulos. O primeiro deles apresenta uma introdução ao tema, abordando os aspectos gerais de robótica móvel, além de explicitar algumas ferramentas que serão usadas no restante do trabalho.

No segundo capítulo se encontra a apresentação de dois modelos cinemáticos para o Pioneer, sendo que cada um deles será usado em um controlador. Os controladores, por sua parte, são apresentados e justificados, sendo que o segundo demanda a prova de sua estabilidade. O terceiro capítulo compreende a apresentação da interface criada e a explicação de cada uma de suas ferramentas.

Os resultados experimentais são apresentados no quarto capítulo, através de gráficos gerados no próprio programa durante a execução de experimentos de controle de postura. Por fim, o quinto capítulo apresenta as conclusões do trabalho, assim como sugestões de implementações futuras.

## 2 Modelagem e Controle

### 2.1 Modelo Cinemático

O modelo cinemático de um robô é usado para descrever seu movimento em um ambiente. Considerando que a navegação do Pioneer 3-DX é feita em um plano horizontal. Ele possui tração diferencial, o que o permite rotacionar sobre o próprio eixo. Seu movimento total é o conjunto do movimento de suas rodas, que lhe proporcionam velocidade linear  $v$  e velocidade angular  $\omega$ . Pela Figura 5, a origem do sistema de coordenadas inercial  $o$  e a origem do sistema de coordenadas do robô  $r$  são utilizadas para definir a posição do robô. Note que a orientação da velocidade linear é coincidente com um dos eixos do robô e a velocidade angular é perpendicular ao plano de movimento do robô.

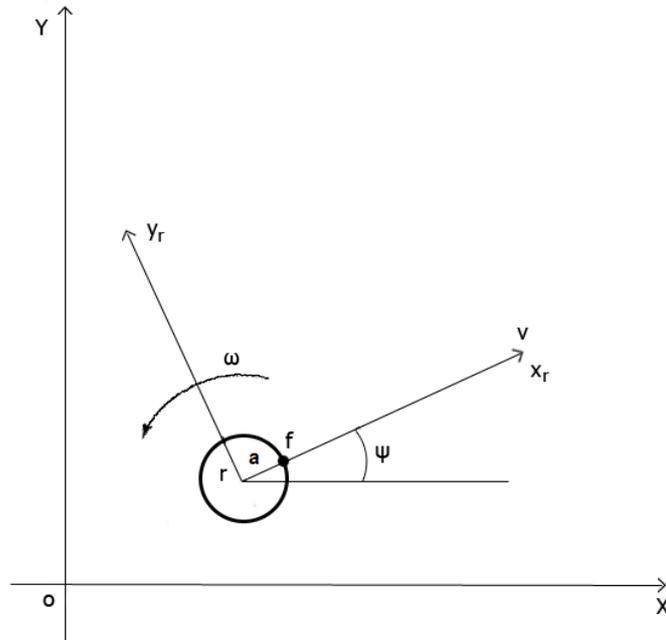


Figura 5: Sistemas de coordenadas inercial e do robô.

Considerando que postura do robô é representada por  $X = [x \ y \ \psi]^T$ , onde  $x$  e  $y$  representam as coordenadas do centro de rotação do robô, referente ao eixo inercial, e  $\psi$  representa o ângulo de orientação, tomando como base também a referência inercial. O sistema de equações que descreve este movimento é dado por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & 0 \\ \sin \psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (2.1)$$

onde  $v$ ,  $\omega$  e  $\psi$  podem ser vistos na Figura 5.

Entretanto, muitas vezes controlar o centro de rotação do robô não é a estratégia mais interessante, sendo melhor controlar a frente do mesmo, como mostra o ponto  $f$  na Figura 5. Considerando que o ponto de controle  $f$  está a uma distância  $a$  do seu centro de rotação, seu movimento é descrito por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (2.2)$$

A Figura 6 mostra a situação onde o robô deve encontrar um alvo, lá assumindo uma postura previamente definida. A tarefa é então a navegação até o ponto alvo  $g$  onde, dependendo da aplicação, também se deseja que o veículo tenha uma orientação predefinida ao chegar. Estes controladores serão tratados na seção subsequente.

Torna-se interessante para certos controladores a representação do modelo cinemático em coordenadas polares. A Equação (2.1) reescrita com a mudança de coordenadas é dada por

$$\begin{bmatrix} \dot{\rho} \\ \dot{\beta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\cos(\beta - \psi) & 0 \\ \frac{\sin(\beta - \psi)}{\rho} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (2.3)$$

onde  $\rho$  representa o erro de posição, definido pela distância do robô até o ponto  $g$  e  $\beta$  é definido por

$$\beta = \tan^{-1} \frac{y_d - y_r}{x_d - x_r}. \quad (2.4)$$

Considerando que  $\beta$  é a soma da orientação do robô em relação ao referencial inercial ( $\psi$ ) e um ângulo  $\alpha$ , tem-se que

$$\beta = \psi + \alpha \Rightarrow \alpha = \beta - \psi \quad (2.5)$$

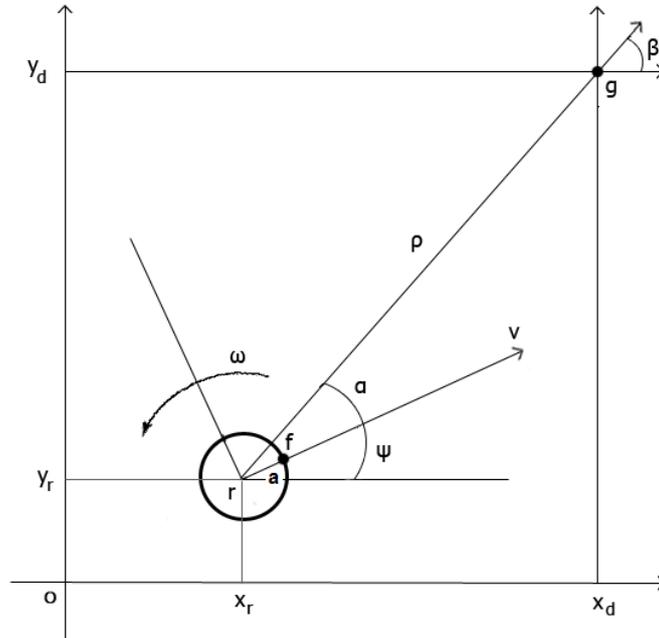


Figura 6: Sistemas de coordenadas inercial e do robô com busca a uma postura determinada.

Define-se  $\alpha$  então como o erro entre a orientação do robô em relação ao referencial inercial  $o$  e a orientação do robô em relação a  $g$ . Fazendo a derivada temporal de (2.5), tem-se que

$$\dot{\alpha} = \dot{\beta} - \dot{\psi}. \quad (2.6)$$

Substituindo (2.5) e (2.6) em (2.3), chega-se a

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos \alpha & 0 \\ \frac{\sin \alpha}{\rho} & -1 \\ \frac{\rho}{\sin \alpha} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (2.7)$$

É importante observar que não se deve ordenar que o robô chegue no alvo, pois, para que isto ocorra, o erro de posição  $\rho$  deve ser nulo, o que causaria uma indeterminação nas equações das derivadas de  $\alpha$  e  $\beta$ . Isto pode ser observado na Equação (2.7).

## 2.2 Controladores

Nesta seção serão apresentados dois controladores. O primeiro será um controlador de posição baseado no modelo cinemático inverso, e o segundo um controlador chaveado

de posição com orientação final, cuja estabilidade é demonstrada no sentido de Lyapunov.

### 2.2.1 Controlador por Modelo Cinemático Inverso

Este controlador de posição sem orientação final é simples de ser aplicado. Entretanto, para que a inversa do modelo cinemático possa ser definida, este modelo não pode ser aquele definido na equação (2.1) onde o ponto de controle é o centro do robô. Deste modo, para este controlador utiliza-se o modelo cinemático descrito em (2.2), ignorando na proposta do controlador, a orientação do robôs (a qual será tangente à sua movimentação). Desta forma, a Equação (2.2) é escrita como

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (2.8)$$

o que leva a

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \end{bmatrix}^{-1} \begin{bmatrix} \nu_x \\ \nu_y \end{bmatrix}, \quad (2.9)$$

onde já tem-se diretamente a velocidade linear e a velocidade angular definidas. Assumindo,

$$\begin{bmatrix} \nu_x \\ \nu_y \end{bmatrix} = \begin{bmatrix} \dot{x}_d + k_{x1}\tilde{x} \\ \dot{y}_d + k_{y1}\tilde{y} \end{bmatrix}, \quad (2.10)$$

onde  $\tilde{x} = x_d - x$  e  $\tilde{y} = y_d - y$  são os erros de posição, e  $k_i$  são os ganhos positivos. Em malha fechada, tem-se

$$\begin{bmatrix} \dot{\tilde{x}} + k_{x1}\tilde{x} \\ \dot{\tilde{y}} + k_{y1}\tilde{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.11)$$

Logo, os erros de posição tendem a zero, quando o tempo tende a infinito, indicando a estabilidade assintótica do sistema.

### 2.2.2 Controlador Chaveado de Posição com Orientação Final

Este controlador tem por objetivo a navegação do robô com a orientação final definida. Entretanto, este controlador será baseado no modelo cinemático em coordenadas polares e, como foi discutido, não se pode ordenar que o robô chegue ao seu objetivo, tendo que se admitir uma tolerância como critério de parada. Isto, porém, pode fazer com que a orientação final seja diferente da desejada. Para resolver este problema utiliza-se o chaveamento de controladores.

Neste caso, o sistema é composto inicialmente de um controlador de posição final com

orientação final e outro apenas de orientação final. No tipo de situação em que há um chaveamento de controlador, segundo [20], é correto afirmar que o sistema será estável uma vez que se tenha uma função de Lyapunov comum para ambos os controladores, devido à extensão direta do teorema da estabilidade de Lyapunov.

Define-se para o controlador as variáveis de estado  $\Pi = [\rho \ \alpha \ \tilde{\beta}]^T$ , onde  $\tilde{\beta}$  é o erro de orientação final do robô, sendo definido por

$$\tilde{\beta} = \beta_d - \beta, \quad (2.12)$$

e  $\beta_d$  é a orientação final desejada. Tomando-se a derivada temporal de (2.12) tem-se

$$\dot{\tilde{\beta}} = -\dot{\beta}, \quad (2.13)$$

por se tratar de um controle de posição (derivadas iguais a zero).

Substituindo (2.13) em (2.7), chega-se em novas equações de estado em coordenadas polares:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\tilde{\beta}} \end{bmatrix} = \begin{bmatrix} -\cos \alpha & 0 \\ \frac{\sin \alpha}{\rho} & -1 \\ -\sin \alpha & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (2.14)$$

Escolhe-se, então, para a função candidata de Lyapunov

$$V(\rho, \alpha, \tilde{\beta}) = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2 + \frac{1}{2}\tilde{\beta}^2, \quad (2.15)$$

radialmente não limitada e crescente, com o objetivo de verificar a estabilidade assintótica global do sistema.

Derivando-se a função candidata de Lyapunov e substituindo as equações de estado presentes em (2.14), obtêm-se

$$\begin{aligned} \dot{V}(\rho, \alpha, \tilde{\beta}) &= \rho\dot{\rho} + \alpha\dot{\alpha} + \tilde{\beta}\dot{\tilde{\beta}} \\ &= -\rho v \cos \alpha + \alpha \left( -\omega + v \frac{\sin \alpha}{\rho} \right) - \tilde{\beta} v \frac{\sin \alpha}{\rho} \end{aligned} \quad (2.16)$$

a qual deve ser definida negativa. Separando os termos e fazendo

$$\begin{aligned} \dot{V}_1(\rho, \alpha, \tilde{\beta}) &= -\rho v \cos \alpha \\ \dot{V}_2(\rho, \alpha, \tilde{\beta}) &= \alpha \left( -\omega + v \frac{\sin \alpha}{\rho} \right) - \tilde{\beta} v \frac{\sin \alpha}{\rho} \end{aligned} \quad (2.17)$$

e garantindo que  $\dot{V}_1 \leq 0$  e  $\dot{V}_2 \leq 0$ , pode-se concluir que  $\dot{V} \leq 0$ .

Sendo  $\dot{V}_1(\rho, \alpha, \tilde{\beta}) = -\rho v \cos \alpha$  e assumindo que

$$v = v_{max} \tanh \rho \cos \alpha \text{ para } v_{max} > 0, \quad (2.18)$$

resulta em

$$\dot{V}_1 = -v_{max} \rho \tanh \rho \cos^2 \alpha \leq 0. \quad (2.19)$$

Repetindo a análise anterior para  $\dot{V}_2$  e assumindo

$$\omega = k_\alpha \alpha + k_\beta \frac{\tilde{\beta}^2}{\alpha} + v \frac{\sin \alpha}{\rho} \left( 1 - \frac{\tilde{\beta}}{\alpha} \right) \text{ para } k_\alpha, k_\beta > 0 \quad (2.20)$$

tem-se que

$$\dot{V}_2 = -k_{alpha} \alpha^2 - k_{beta}^2 \leq 0. \quad (2.21)$$

Assim, como  $\dot{V}_1 \leq 0$  e  $\dot{V}_2 \leq 0$ , corrobora-se a hipótese de que

$$\dot{V}(\rho, \alpha, \tilde{\beta}) = -v_{max} \rho \tanh \rho \cos^2 \alpha - k_{alpha} \alpha^2 - k_{beta}^2 \leq 0. \quad (2.22)$$

Mais ainda,  $\dot{V}(\rho, \alpha, \tilde{\beta}) < 0$  resultando em um sistema assintoticamente estável.

Substituindo (2.18) e (2.20) em (2.14), as equações em malha fechada do sistema se tornam

$$\begin{cases} \dot{\rho} = -v_{max} \tanh \rho \cos^2 \alpha \\ \dot{\alpha} = -k_\alpha \alpha^2 - k_\beta^2 - v_{max} \tilde{\beta} \frac{(\tanh \rho)}{\rho} \frac{\sin \alpha}{\alpha} \cos \alpha \\ \dot{\tilde{\beta}} = -v_{max} \tilde{\beta} \frac{(\tanh \rho)}{\rho} \frac{\sin \alpha}{\alpha} \cos \alpha \end{cases} \quad (2.23)$$

e pode-se afirmar que

$$\begin{bmatrix} \rho \\ \alpha \\ \tilde{\beta} \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

de forma assintótica. Além disso, como

$$\tilde{\beta} = \beta_d - \beta \rightarrow 0 \Rightarrow \beta \rightarrow \beta_d$$

conforme desejado. Porém, conforme mencionado, não se pode ordenar que o robô chegue na posição desejada, já que  $\rho = 0$  introduziria indeterminações no sistema. Define-se então uma região que indica que o robô já chegou na posição desejada. Isto, porém, não garante que a orientação é a desejada, havendo nesse instante o chaveamento de controladores.

Considerando que  $\rho \approx 0$  e  $\alpha \approx 0$  na região de chegada, pode-se reescrever a candidata

de Lyapunov apresentada na Equação (2.15) como

$$V(\tilde{\beta}) = \frac{1}{2}\tilde{\beta}^2. \quad (2.24)$$

Sabendo que a candidata de Lyapunov deste controlador está contida em (2.15), garante-se a estabilidade para sistemas chaveados.

Derivando-se (2.24) no tempo, tem-se

$$\dot{V}(\tilde{\beta}) = \tilde{\beta}\dot{\tilde{\beta}}. \quad (2.25)$$

De (2.5) pode-se escrever

$$0 = \beta - \psi \Rightarrow \beta = \psi, \quad (2.26)$$

e

$$\tilde{\beta} = \tilde{\psi} = \psi_d - \psi. \quad (2.27)$$

Fazendo a derivada temporal de (2.27) tem-se que

$$\dot{\tilde{\beta}} = -\dot{\psi}. \quad (2.28)$$

Considerando que  $\psi$  é a orientação do robô, pode-se afirmar

$$\dot{\psi} = \omega. \quad (2.29)$$

sendo  $\omega$  a velocidade angular do robô. Substituindo (2.27), (2.28) e (2.29) em (2.25), obtêm-se

$$\dot{V}(\tilde{\beta}) = -\tilde{\psi}\omega. \quad (2.30)$$

Assumindo que

$$\omega = \omega_{max} \tanh k_\omega \psi \quad (2.31)$$

e substituindo (2.31) em (2.30), obtêm-se

$$\dot{V}(\tilde{\beta}) = -\tilde{\psi}\omega_{max} \tanh k_\omega \psi \leq 0, \text{ para } \omega_{max}, k_\omega > 0 \quad (2.32)$$

Logo, é correto afirmar que  $\dot{V}(\tilde{\beta}) < 0$ , ou seja, é definida negativa, confirmando a estabilidade do controlador.

Para uma explicação mais detalhada sobre a demonstração de estabilidade no sentido de Lyapunov consulte [6].

### 3 Interface Matlab-Pioneer

Uma interface entre o usuário e o robô Pioneer P3-DX, Mobile Robots Inc foi implementada utilizando o software Matlab<sup>®</sup>. Em suma, a interface comunica com o software ARIA<sup>®</sup>, da Mobile Robots Inc., o qual faz a comunicação direta com o robô. A tela principal do programa é mostrada na Figura 7 e será detalhada no decorrer deste capítulo.

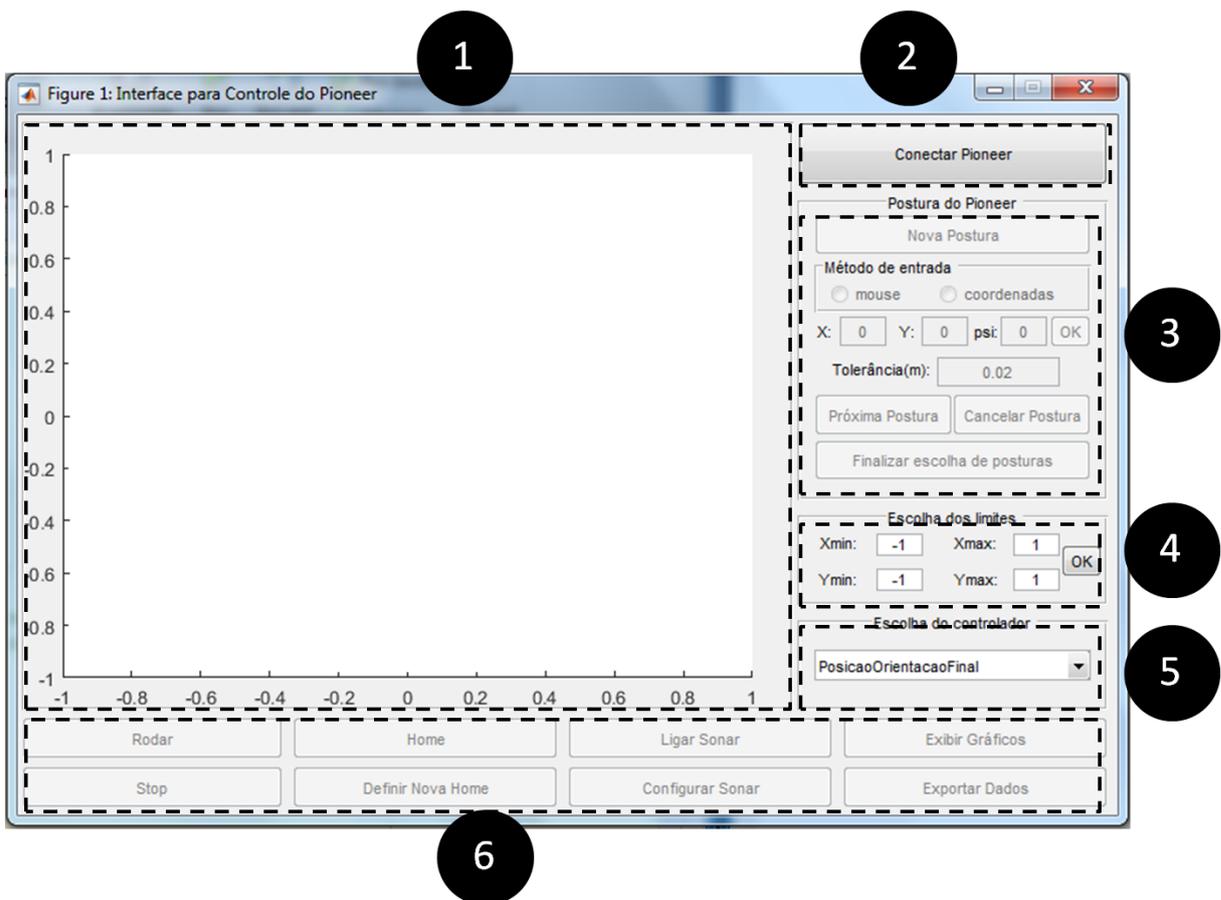


Figura 7: Tela principal do Software

## 3.1 Mapa

O item 1 da Figura 7 mostra o mapa onde o Pioneer navegará. Lá estarão representadas todas as posturas desejadas e suas respectivas regiões de chegada, assim como a orientação desejada, caso o controlador contemple tal possibilidade. O mapa também apresenta um rastro, mostrando toda a trajetória que o Pioneer executou. Este rastro, assim como todos os pontos, podem ser retirados do mapa com comandos que serão apresentados em seções seguintes.

## 3.2 Conectar Pioneer

Este botão, representado pelo item 2 da Figura 7, é o primeiro botão a ser utilizado pelo usuário. Ele faz a comunicação com o software ARIA<sup>©</sup> e então este estabelece a conexão com o Pioneer. Além disso, ao pressionar este botão as outras funções do programa são liberadas.

## 3.3 Postura do Pioneer

O conjunto de botões representados na Figura 7 pelo item 3 é responsável pela escolha das posturas, às quais o Pioneer deve atender, ou seja, aqui se definem as posições e orientações desejadas.

### 3.3.1 Nova Postura

Este botão inicia o processo de definição de um novo ponto desejado. Cada ponto contém como informação a matriz  $X_d = [x_d \ y_d \ \psi_d]^T$  que terá seus valores definidos pelo usuário. Além disso, quando um conjunto de pontos é apagado (como será explicado em seções posteriores), este botão muda de nome para ‘Nova Escolha de Posturas’ e, ao ser pressionado, prepara o programa para a entrada de um novo conjunto.

### 3.3.2 Método de Entrada

Neste menu, o usuário encontra duas opções para o método de entrada dos valores desejados de postura: mouse ou coordenadas. Se o usuário preferir utilizar o mouse para entrar com os valores, ele deverá clicar duas vezes no mapa. O primeiro clique definirá

os valores de  $x_d$  e  $y_d$ , enquanto que o ângulo entre o primeiro clique e o segundo definirá o valor de  $\psi_d$ . Caso o usuário escolha o método coordenadas, ele deve então entrar com os valores de  $x_d$ ,  $y_d$  e  $\psi_d$  em seus respectivos lugares, logo abaixo do quadro de métodos de entrada, e então pressionar o botão ‘OK’. É importante lembrar que  $x$  e  $y$  estão em metros e  $\psi$  em graus, com valores de  $0^\circ$  a  $360^\circ$ , que serão automaticamente traduzidos em radianos pelo programa. Nos dois métodos, a tolerância pode ser definida no retângulo logo abaixo e, como indicado, ela está também em metros.

Uma última consideração que deve ser feita é a de que o usuário não pode escolher posições que estejam fora dos limites do mapa. Caso haja a tentativa de se fazer isso, um aviso será dado e o processo de escolha iniciará novamente, até o usuário escolher um valor válido. Caso o usuário queira escolher uma posição fora do mapa, ele deverá alterar os limites deste, como será explicado na próxima seção.

### 3.3.3 Próxima Postura, Cancelar Postura e Finalizar escolha de posturas

Após entrar com os valores da seção anterior o usuário tem três opções. A primeira delas, o botão intitulado ‘Próximo Postura’, faz com que o processo de método de entrada e escolha de postura comece novamente, adicionando uma nova postura desejada à lista de posturas desejadas do robô. A segunda delas, ‘Cancelar Postura’, também faz com o processo de escolha de postura comece novamente, porém, a última postura adicionada será apagada. A última opção encerra o processo de escolha de posturas desejadas, libera o acesso aos botões de navegação do Pioneer e passa pra ele a lista de posturas que ele deve assumir na ordem que elas foram escolhidas pelo usuário.

## 3.4 Escolha dos limites

O item 4 da Figura 7 mostra os controles dos limites do mapa. Nele o usuário é livre para entrar os valores mínimos e máximos que serão mostrados no mapa. Ao terminar a escolha o usuário deve apertar o botão ‘OK’. Além disso, assim como o usuário não pode escolher uma postura fora dos limites do mapa, ele também não pode escolher limites que excluam uma postura previamente definida. Como exemplo, caso o usuário tenha escolhido uma postura  $X_d = [3 \ 2 \ \frac{\pi}{2}]^T$ , ele não pode escolher 2 como valor máximo de  $x$ , pois isto excluiria  $X_d$  do mapa.

## 3.5 Escolha do controlador

No item 5 mostrado na Figura 7, o usuário poderá escolher o controlador que ele deseja para o Pioneer. Atualmente, há a escolha entre dois controladores, sendo um apenas de controle de posição e outro de controle de posição com orientação final, ambos detalhados no capítulo anterior.

## 3.6 Botões de Navegação e Outros

O último conjunto de botões, o item 6 da Figura 7, cobre várias funções, sendo a mais importante delas, a função que habilita a navegação do Pioneer. Estas funções serão explicitadas uma a uma nas subseções seguintes.

### 3.6.1 Rodar

Este botão deve ser habilitado após o término da escolha pelo usuário de uma ou mais posturas desejadas. Ele permite o início do movimento do Pioneer. Após o início, este botão muda de nome para ‘Parar’ e, ao ser pressionado, faz com que o pioneer interrompa sua navegação, sem perder sua lista de postura desejadas e mudando o nome do botão novamente para ‘Rodar’, fazendo o ciclo se reiniciar.

Outro elemento importante do programa é sua compatibilidade com um joystick. Caso um joystick esteja conectado corretamente, ao ser habilitado durante a navegação do robô, os sinais de controle são enviados por ele. Caso contrário, o robô é guiado pelo controlador implementado até o próximo ponto desejado, independentemente de onde parou.

Caso o robô esteja parado (botão ‘Parar’ selecionado), o joystick funcionará diretamente, sem a necessidade de habilitação prévia.

### 3.6.2 Stop

Este botão é habilitado apenas quando o robô está parado ou já concluiu todo o seu movimento. Quando o usuário pressiona este botão, toda lista de posturas desejadas é apagada, assim como os pontos no mapa e o rastro do robô, como explicado na Seção 3.1. O botão ‘Stop’ também habilita o botão ‘Nova Postura’ sob o nome de ‘Nova escolha de posturas’ como explicado na subseção 3.3.1.

### 3.6.3 Home

Este botão, que somente é habilitado quando o robô está parado ou já concluiu todo o seu movimento, faz com que o Pioneer vá de sua postura atual para a postura origem (início dos experimentos) ou para a postura definida pelo botão ‘Definir Nova Home’ (explicado na subseção 3.6.4). Em outras palavras, ao se pressionar esse botão ‘Home’, a nova postura desejada é estabelecida como  $X_d = [0 \ 0 \ 0]^T$ .

### 3.6.4 Definir Nova Home

Habilitado apenas quando o robô está parado ou já concluiu todo o seu movimento, este botão define a posição atual do Pioneer como sua ‘Home’, ou seja, ele estabelece  $X = [0 \ 0 \ 0]^T$ , zerando a odometria do Pioneer, e conseqüentemente todo o mapa será relativo a essa nova ‘Home’.

O usuário deve ter cuidado ao utilizar tal comando, afinal ele muda todo o sistema de referências para a nova posição e orientação, fazendo com que as posturas desejadas mudem de lugar de acordo com a nova referência.

### 3.6.5 Ligar Sonar e Configurar Sonar

O Pioneer P3-DX conta com oito sensores de sonar posicionados na sua parte da frente, como a Figura 8 ilustra.

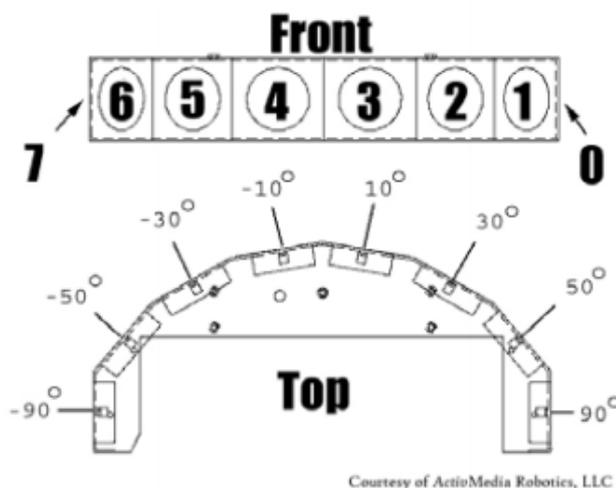


Figura 8: Conjunto de sonares do Pioneer P3-DX

Quando o usuário pressiona este botão, 8 linhas representando a leitura do sonar do Pioneer são desenhadas no mapa. O sonar conta com uma distância de até 5 metros, entretanto, para mapas menores esta distância pode ser grande demais, tornando o mapa muito carregado. Para isso, utilizando o botão ‘Configurar Sonar’, uma caixa de diálogo abrirá e o usuário poderá escolher uma saturação para a leitura do sonar de outro valor, desde que este esteja abaixo de 5 metros.

### 3.6.6 Exibir Gráficos e Exportar Dados

Este botão é habilitado após o fim da navegação do Pioneer e permite a exibição dos gráficos que comparam os sinais de controle enviados e os sinais de controle executados pelo robô. Para acesso aos dados fora do Matlab<sup>®</sup>, o usuário pode clicar no botão ‘Exportar Dados’. Isto abrirá uma caixa de diálogo para inserção do nome do arquivo e será gerada uma planilha em excel contendo todas as posturas desejadas e realizadas e os sinais de controle enviados e executados.

## 4 Resultados e Discussões

### 4.1 Comparação de Sinais de Controle para uma única postura desejada

Neste experimento, o Pioneer saiu da origem até uma posição desejada  $X_d = [3 \ 2]^T$ . As distâncias estão em metros, a tolerância usada foi de 50 mm e a orientação não é controlada. A Figura 9 mostra o caminho que o Pioneer percorreu. Observe que o controlador orienta o Pioneer para a posição desejada e, em seguida, realiza praticamente uma linha reta até o destino.

Já a Figura 10 mostra a evolução dos valores de  $x$ ,  $y$  e  $\psi$  com o tempo. As linhas tracejadas indicam o valor desejado. Note que, conforme sugerido pela rota, a orientação do Pioneer muda apenas no início e então permanece constante pelo restante do movi-

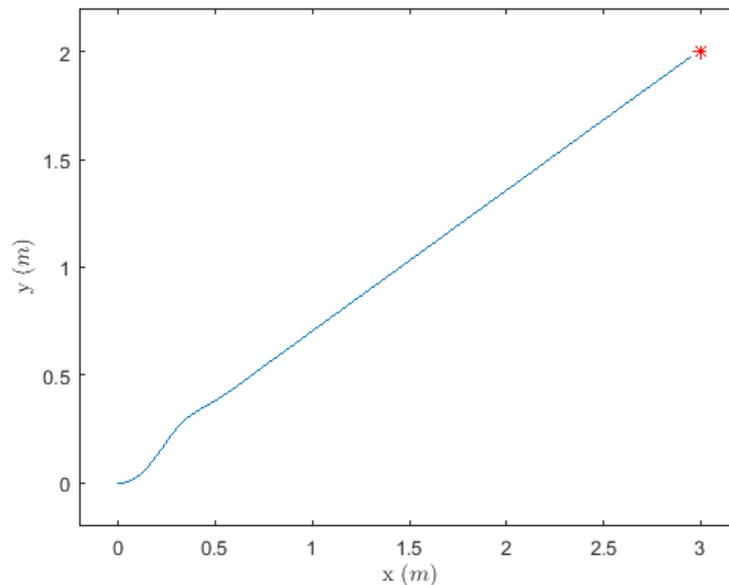


Figura 9: Caminho feito pelo Pioneer saindo da origem e indo até a posição  $X_d = [3 \ 2]^T$  com o controlador baseado em cinemática inversa.

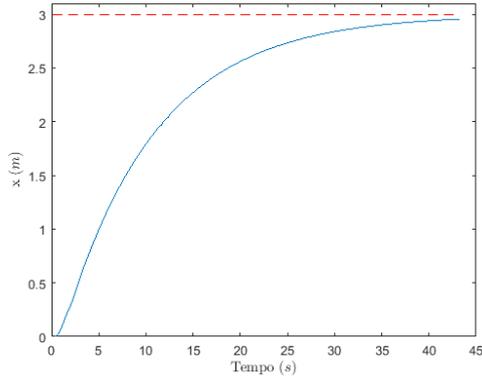
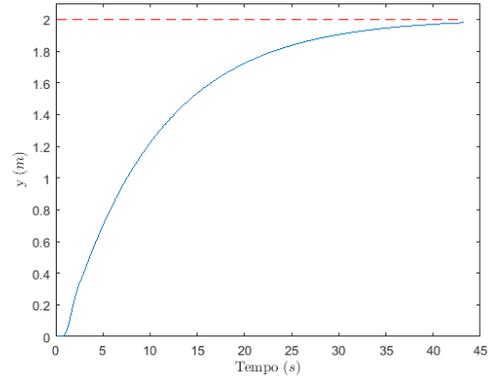
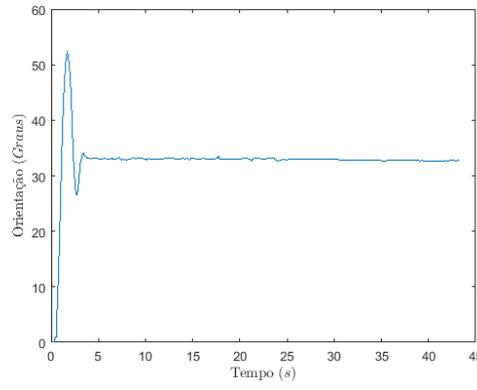
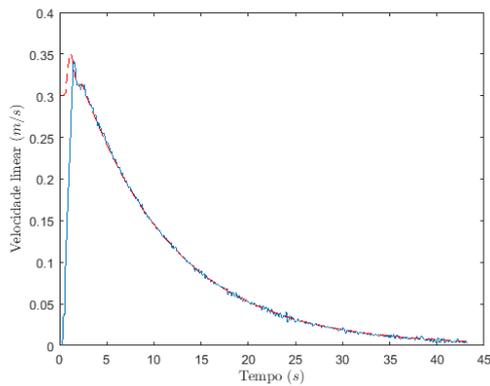
(a) Evolução da posição do Pioneer em  $x$ .(b) Evolução da posição do Pioneer em  $y$ .(c) Evolução da posição do Pioneer em  $x$ .

Figura 10: Evolução da posição e orientação do Pioneer com o controlador baseado em cinemática inversa.

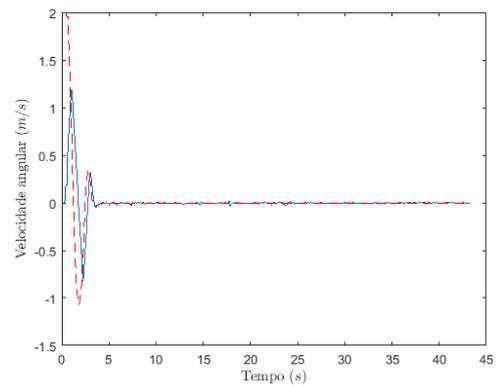
mento. Os valores de  $x$  e  $y$  se comportam como um sistema sobre-amortecido, com um erro final introduzido pela tolerância usada.

A Figura 11 mostra a comparação entre as ações de controle enviadas e executadas pelo Pioneer. As ações de controle desejadas são as linhas tracejadas, enquanto aquelas executadas pelo Pioneer são as linhas contínuas. Observe que como não há comutação de controladores, não há descontinuidades presentes, e o robô seguiu satisfatoriamente as ações de controle, sendo a exceção o início, devido à sua dinâmica, uma vez que o controlador é projetado de acordo com seu modelo cinemático. A velocidade angular, Figura 11(b), está em realce apenas no início do experimento, tendendo rapidamente à zero, afinal a trajetória realizada é aproximadamente retilínea, não requerendo, portanto, alteração angular.

Para o controlador chaveado de posição com orientação final as constantes foram definidas como se seguem:  $v_{max} = 0.35 \text{ m/s}$ ,  $\omega_{max} = 0.35 \text{ rad/s}$ ,  $k_{\alpha} = 0.44$ ,  $k_{\beta} = 10^{-4}$  e  $k_{\omega} = 1.5$ , sendo  $k_{\alpha}$ ,  $k_{\beta}$  e  $k_{\omega}$  os ganhos deste controlador. A postura final desejada foi



(a) Velocidades lineares enviadas e executadas pelo Pioneer.



(b) Velocidades angulares enviadas e executadas pelo Pioneer.

Figura 11: Comparação entre as ações de controle enviadas ao Pioneer e as executadas com o controlador baseado em cinemática inversa.

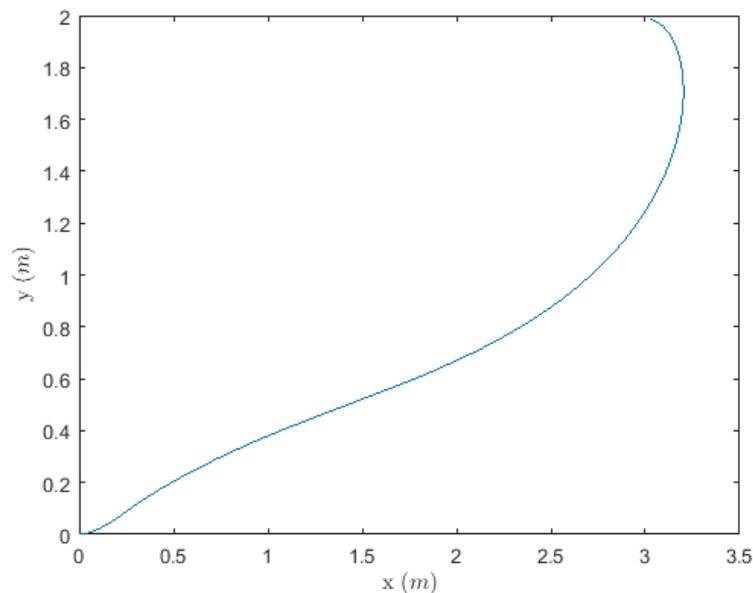


Figura 12: Caminho feito pelo Pioneer saindo da origem e indo até a posição  $X_d = [3 \ 2 \ -\frac{3}{4}\pi]^T$  com o controlador chaveado de posição com orientação final.

$X_d = [3 \ 2 \ -\frac{3}{4}\pi]^T$ . O caminho feito pelo Pioneer é apresentado na Figura 12. Nota-se que o caminho feito é totalmente diferente do caso anterior, mesmo a posição final sendo a mesma, pois agora há a preocupação com a orientação final. A tolerância usada foi de 20 mm.

A evolução do posição e orientação do Pioneer pode ser vista na Figura 13. Note a presença de um *overshot* em  $x$ , que acontece devido à necessidade do robô de passar de seu ponto, para que possa chegar com a orientação desejada. Os valores de  $\psi$  também foram bastante diferentes, pois agora a trajetória descrita não é mais retilínea.

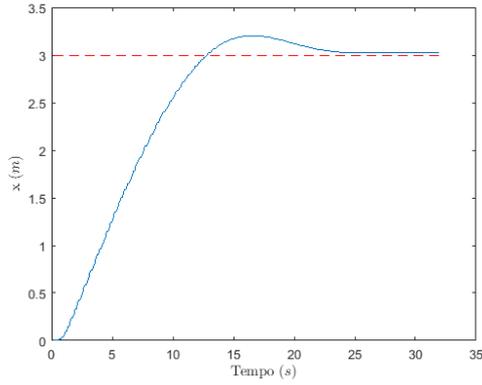
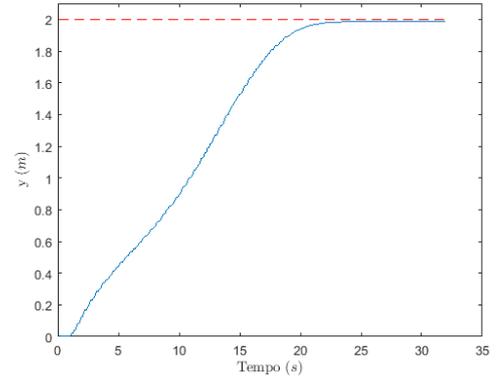
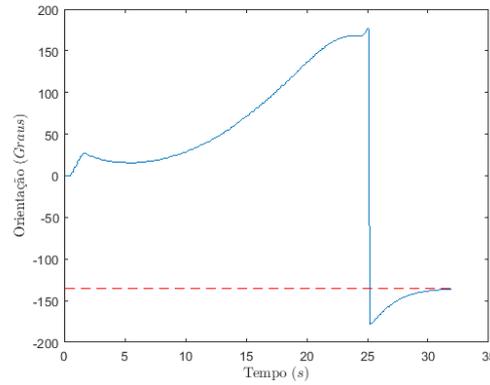
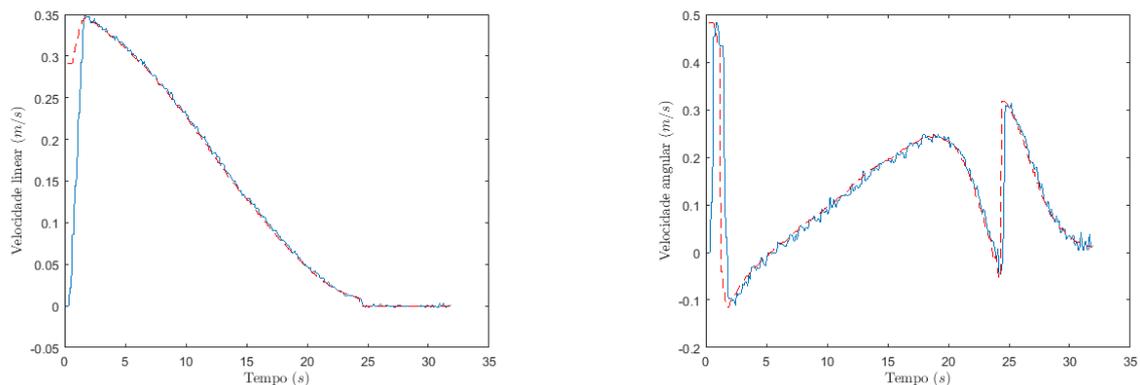
(a) Evolução da posição do Pioneer em  $x$ .(b) Evolução da posição do Pioneer em  $y$ .(c) Evolução da posição do Pioneer em  $x$ .

Figura 13: Evolução da posição e orientação do Pioneer com o controlador chaveado de posição com orientação final.

As ações de controle, apresentadas na Figura 14, também são completamente diferentes, daquelas apresentadas na Figura 11(b). Além disso, neste caso pode-se observar uma descontinuidade no final, entre 24 e 25 segundos, devido à comutação de controladores. Nota-se também a convergência assintótica à zero no final do movimento, como visto nas Figuras 14(a) e 14(b). Novamente, devido à sua dinâmica, o Pioneer inicialmente demora a assumir a ação de controle enviada, mas a partir de poucos instantes ele segue as ações enviadas satisfatoriamente.

## 4.2 Comparação de Sinais de Controle para várias posturas desejadas

Em um segundo experimento passou-se quatro posturas desejadas ao Pioneer,  $X_d = [3.2 \ 2 \ -\frac{3}{4}\pi]^T$ ,  $X_d = [1.6 \ 0.8 \ \frac{\pi}{2}]^T$ ,  $X_d = [3.2 \ 0 \ 0]^T$  e  $X_d = [0 \ 0 \ \pi]^T$  e observou-se o caminho tomado e as ações de controle enviadas e executadas. As Figuras 15(a) e 15(b) mostras



(a) Velocidades lineares enviadas e executadas pelo Pioneer.

(b) Velocidades angulares enviadas e executadas pelo Pioneer.

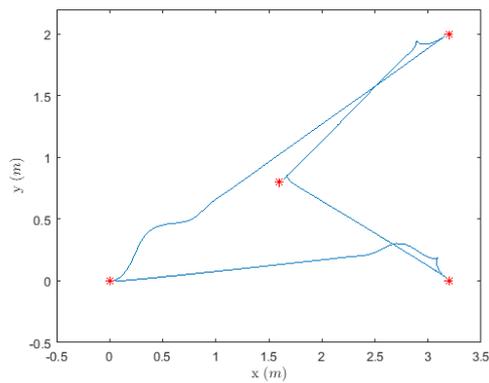
Figura 14: Comparação entre as ações de controle enviadas ao Pioneer e as executadas com o controlador chaveado de posição com orientação final.

as duas rotas tomadas pelo Pioneer. Novamente, elas são completamente distintas entre si.

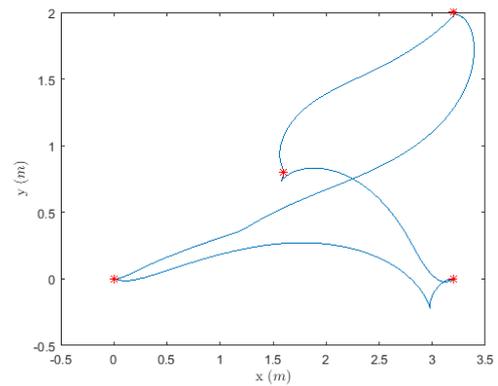
Na Figura 15 são mostrados os valores de  $x$  e  $y$  assumidos pelo Pioneer, colocando novamente os dois controladores em contraste. Na primeira coluna, Figuras 15(c) e 15(e), tem-se o controlador baseado em cinemática inversa, onde se percebe a característica sobre-amortecida do mesmo. A exceção é alguns valores, que acontecem logo após robô mudar de postura desejada. Isto é explicado, pois o robô executa uma pequena trajetória em marcha-ré dependendo da próxima postura desejada, para melhor se orientar.

Na segunda coluna, Figuras 15(d) e 15(f), tem-se o controlador chaveado com orientação final. Diferente do outro controlador, há nesse um *overshot*, como mencionado no experimento anterior. Também pode-se notar que os valores de  $x$  e  $y$  ficam constantes por algum tempo, ao atingirem cada posição desejada. Isto acontece pois o Pioneer muda o controlador após atingir a posição alvo, e o segundo controlador atua apenas na orientação. Percebe-se também que o controlador chaveado foi mais rápido que o baseado em cinemática inversa.

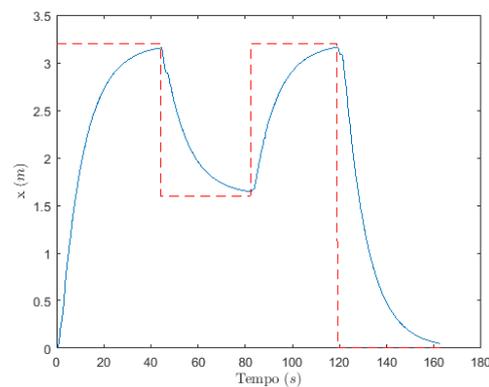
A evolução de orientação para cada controlador é mostrada na Figura 16. Observe que para a Figura 16(a) não existe indicação de orientação desejada, já que não é possível com este controlador controlar a mesma. Além disso, o controlador baseado em cinemática inversa mostra sua característica de mudar rapidamente para uma orientação apontando para a postura desejada e então deixar a orientação constante. Já a orientação do controlador chaveado mostra uma característica muito mais mutável, além de corretamente tender à orientação desejada final.



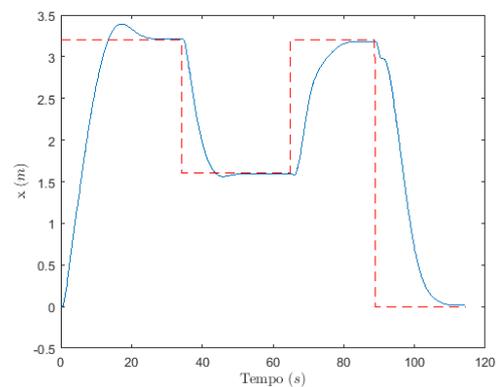
(a) Rota feita pelo controlador baseado em cinemática inversa.



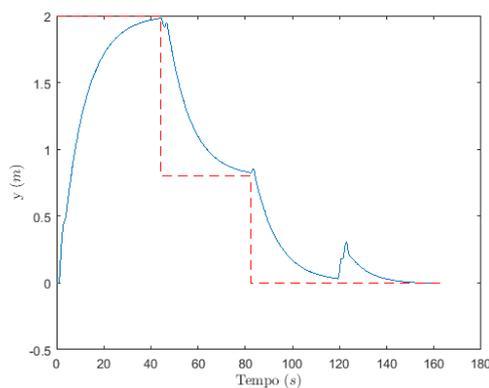
(b) Rota feita pelo controlador chaveado de posição com orientação final.



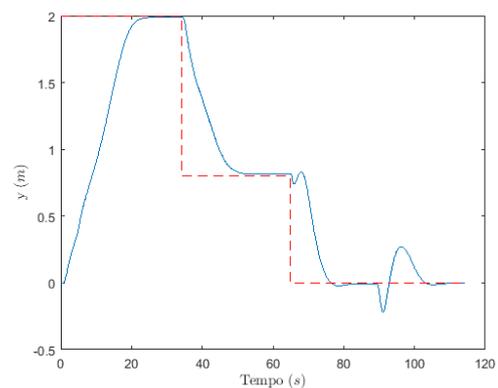
(c) Evolução da posição do Pioneer em  $x$  com controlador baseado em cinemática inversa.



(d) Evolução da posição do Pioneer em  $x$  com controlador chaveado de posição com orientação final.



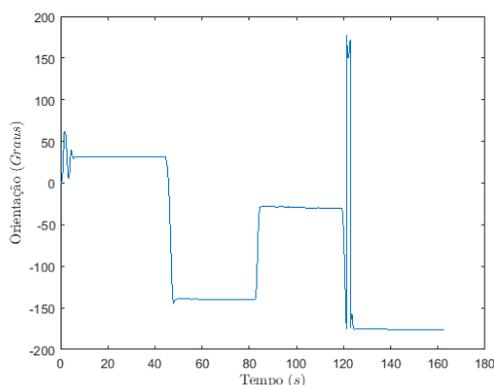
(e) Evolução da posição do Pioneer em  $y$  com controlador baseado em cinemática inversa.



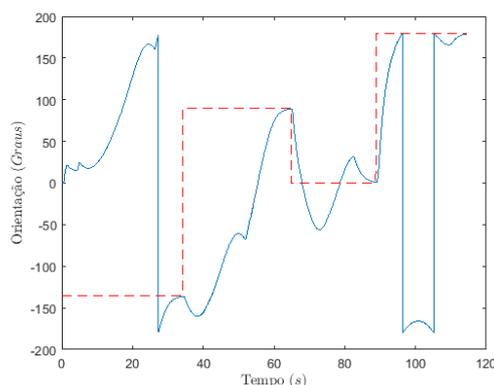
(f) Evolução da posição do Pioneer em  $y$  com controlador chaveado de posição com orientação final.

Figura 15: Comparação entre os valores de  $x$  e  $y$  assumidos pelo Pioneer utilizando-se os dois controladores diferentes.

Talvez a característica mais interessante deste experimento é a visualização das descontinuidades presentes nas ações de controle no momento que o robô chega à cada postura desejada e passa então a se encaminhar para a próxima. Isto pode ser percebido na Figura



(a) Evolução da orientação do Pioneer com controlador baseado em cinemática inversa.



(b) Evolução da orientação do Pioneer com controlador chaveado de posição com orientação final.

Figura 16: Comparação entre as orientações assumidas pelo Pioneer utilizando-se os dois controladores diferentes.

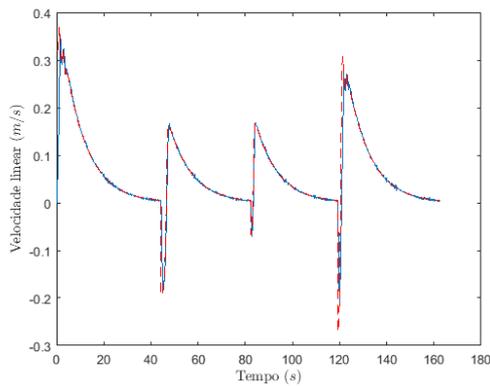
17. Para o caso do controlador baseado em cinemática inversa, Figuras 17(a) e 17(c), a descontinuidade é observada em torno de 45, 85 e 120 segundos. Já para o controlador com orientação final, Figuras 17(b) e 17(d), a descontinuidade é observada nas comutações de controladores, mas também em torno de 35, 65 e 90 segundos particularmente, onde ocorre a troca de objetivo.

Também é interessante comentar sobre os valores negativos de velocidade linear que acontecem nas trocas de objetivo, quando o novo objetivo está atrás do robô. Isto ocorre devido ao controlador que, buscando a orientação necessária para se chegar ao ponto, faz com que o Pioneer execute o movimento de marcha-ré.

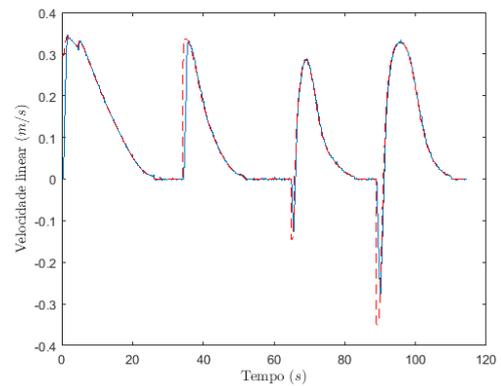
### 4.3 Comparação de diferentes valores de ganho

O objetivo deste experimento foi comparar diferentes valores de ganho para entender melhor sua influência no sistema. O teste será composto de um único ponto desejado, idêntico ao da seção 4.1,  $X_d = [3 \ 2]^T$ . O controlador utilizado será aquele baseado em cinemática inversa. O ganho deste, na seção 4.1 era de 0.1 para  $x$  e  $y$ . Os novos valores testados serão de 0.02 e 0.3 para  $x$  e  $y$ . As figuras da seção anterior referentes ao ganho de 0.1 serão repetidas para facilitar a comparação.

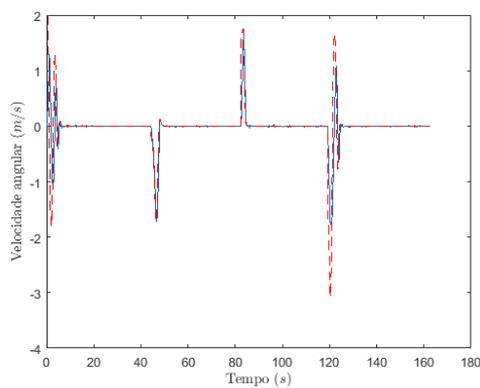
Os ganhos influenciam no movimento do Pioneer e o tempo de navegação. Se eles forem muito baixos, o tempo de navegação pode ficar demasiadamente longo, porém se eles forem aumentados excessivamente o movimento pode ficar oscilatório e pode ocorrer a saturação dos atuadores do robô.



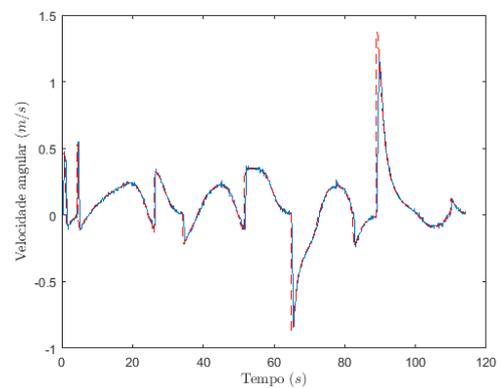
(a) Velocidades lineares enviadas e executadas: controlador baseado em cinemática inversa.



(b) Velocidades lineares enviadas e executadas: controlador chaveado com orientação final.



(c) Velocidades angulares enviadas e executadas: controlador baseado em cinemática inversa.



(d) Velocidades angulares enviadas e executadas: controlador chaveado com orientação final.

Figura 17: Comparação entre as Ações de controle enviadas e executadas pelo Pioneer utilizando-se os dois controladores diferentes.

A primeira comparação a ser feita é a rota seguida pelo Pioneer. As rotas das Figuras 18(a) e 18(b) são parecidas, sendo que a primeira é apenas um pouco menos retilínea no início. Já a rota apresentada na Figura 18(c) é muito menos suave que as outras. A razão disso ficará mais clara quando forem apresentados os valores das ações de controle enviadas ao Pioneer, mas é suficiente falar que claramente esta rota gasta mais energia e é menos ótima que as outras.

A seguir, na Figura 19, comparam-se os valores de  $x$  e  $y$  para cada valor de ganho.

Para os ganhos de valor médio e baixo percebe-se claramente a característica sobre-amortecida e eles são novamente muito parecidos em sua forma de curva. Já para o ganho de valor alto percebe-se claramente uma distorção dos deslocamentos ao longo dos eixos. Isso novamente se deve às ações de controle, muito bruscas. Entretanto, nesse conjunto de gráficos pode-se observar uma diferença importantíssima nas três situações, o tempo de trajetória. Enquanto o tempo com o ganho médio fica em pouco menos de 45 segundos

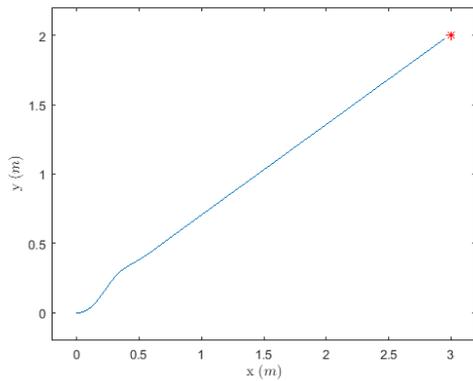
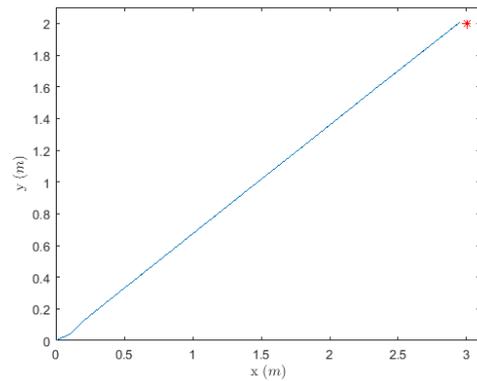
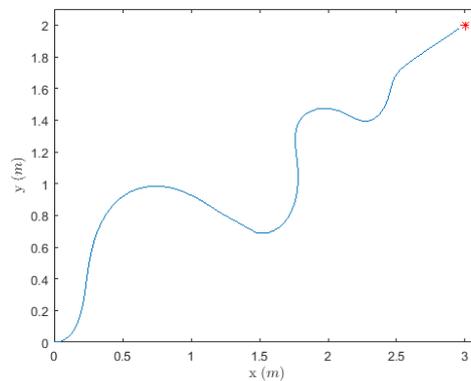
(a) Rota feita pelo Pioneer com ganho de 0.1 em  $x$  e  $y$ .(b) Rota feita pelo Pioneer com ganho de 0.02 em  $x$  e  $y$ .(c) Rota feita pelo Pioneer com ganho de 0.3 em  $x$  e  $y$ .

Figura 18: Rota feita pelo Pioneer com diferentes valores de ganho.

e o tempo para o ganho alto fica em pouco menos de 20, o tempo final para o ganho baixo fica quase 250 segundos.

Isso quer dizer que, ao se diminuir o ganho, ganha-se em estabilidade, mas sofre-se uma perda grande no tempo. Enquanto a trajetória feita utilizando-se o ganho alto foi sub-ótima, o tempo de conclusão foi muito menor se comparado aos outros.

Finalmente, é feita a comparação entre as ações de controle enviadas ao Pioneer. Aqui, enquanto as ações de controle do caso médio são sensatas, é possível ver onde os ganhos muito baixos (ou muito altos) são ineficientes. Vê-se claramente na Figura 20(c) que, enquanto a curva tem forma similar, os valores desejados de velocidade linear são muito menores, fazendo inclusive com que o ruído dos sensores a bordo do Pioneer fiquem muito mais claros. Estes valores baixos de velocidade se traduzem em um tempo muito grande para a chegada ao ponto desejado.

Nas Figuras 20(e) e 20(f) vê-se o oposto. Valores de velocidade linear e angular

desejados muito altos e mudanças muito bruscas causam a instabilidade do movimento, impedindo inclusive que o robô alcance os valores desejados antes deles serem completamente mudados. Isto ocorre pois os ganhos altos fazem uma correção muito grande do erro e, devido à dinâmica do robô, este passa da orientação que aponta para a posição desejada, o que muda o sinal do erro bruscamente causando uma nova correção e, conseqüentemente, uma nova mudança brusca do erro.

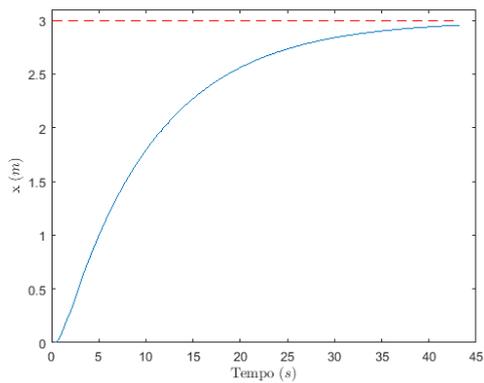
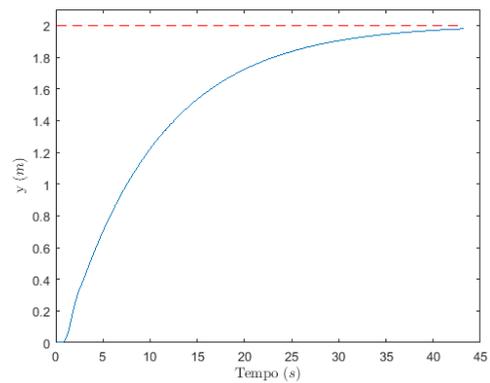
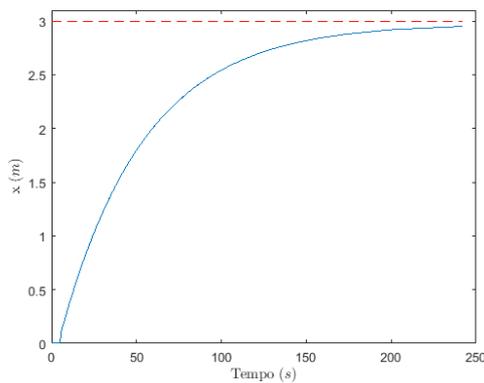
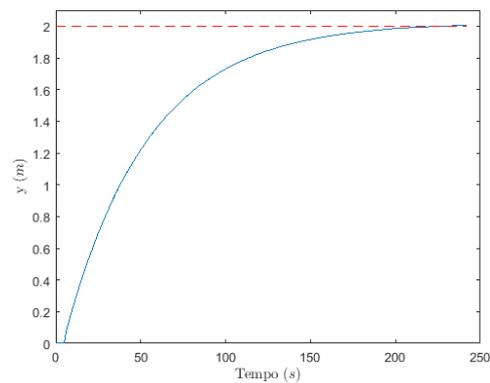
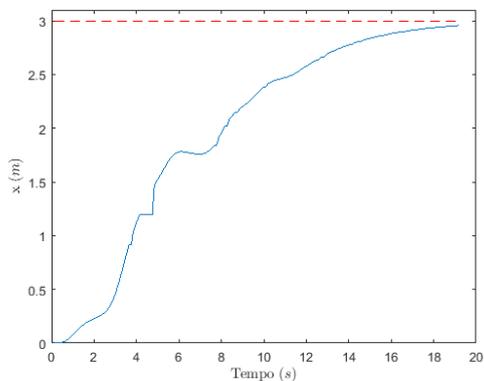
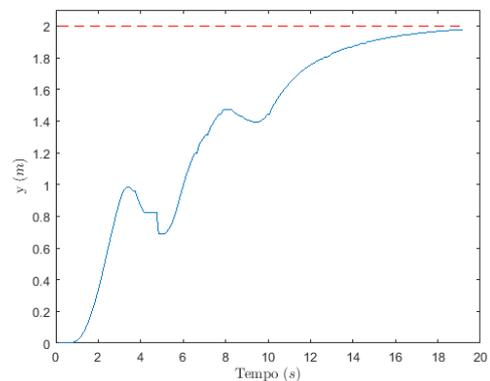
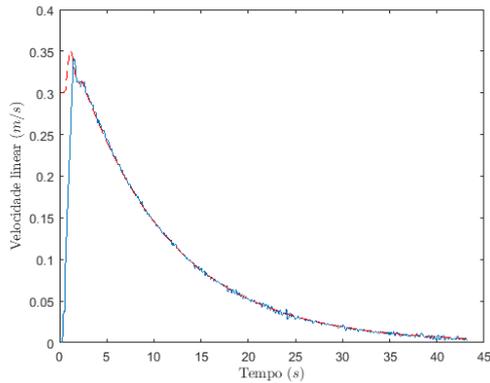
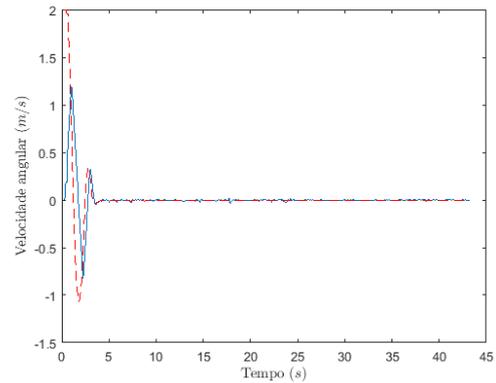
(a) Valores de  $x$  com ganho de 0.1.(b) Valores de  $y$  com ganho de 0.1.(c) Valores de  $x$  com ganho de 0.02.(d) Valores de  $y$  com ganho de 0.02.(e) Valores de  $x$  com ganho de 0.3.(f) Valores de  $y$  com ganho de 0.3.

Figura 19: Evolução da posição com diferentes valores de ganho.

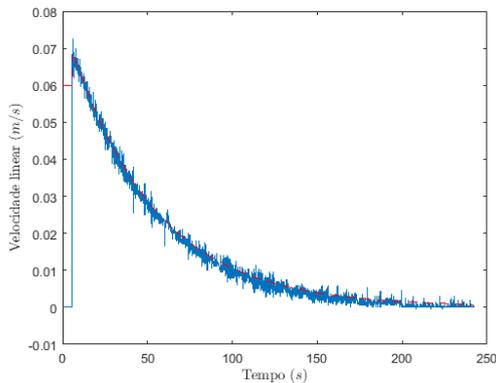
É importante lembrar que todos estes experimentos e todos os gráficos gerados foram feitos utilizando-se somente a plataforma criada para interface entre o Matlab<sup>®</sup> e o Pioneer. Estes testes, além de levarem às conclusões apresentadas, também mostram a versatilidade e o poder que a plataforma criada tem quanto ferramenta de pesquisa.



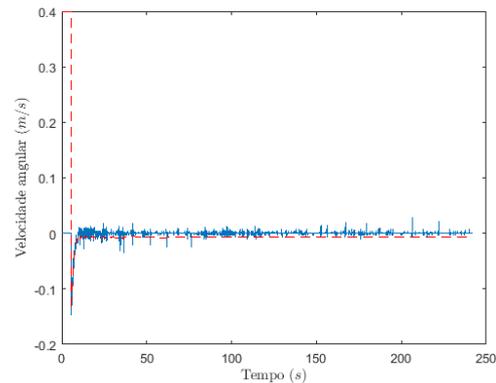
(a) Valores de velocidade linear com ganho de 0.1.



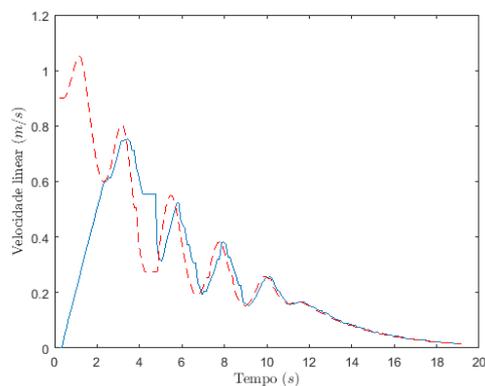
(b) Valores de velocidade angular com ganho de 0.1.



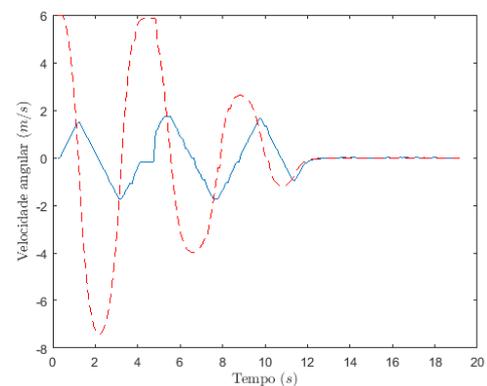
(c) Valores de velocidade linear com ganho de 0.02.



(d) Valores de velocidade angular com ganho de 0.02.



(e) Valores de velocidade linear com ganho de 0.3.



(f) Valores de velocidade angular com ganho de 0.3.

Figura 20: Evolução da posição com diferentes valores de ganho.

## 5 *Conclusões*

A interface construída e detalhada ao longo deste trabalho apresentou facilidade de uso e robustez. Ela permite um usuário iniciante fazer experimentos com o Pioneer sem conhecimento algum de programação ou controle de robôs móveis. Ela ainda permite que usuários mais avançados a modifiquem, acrescentando novos controladores, mudando estilo e conteúdo de gráficos, e inserindo novas ferramentas. Ela é simples de ser entendida, pois o fluxo de informações segue a descrição apresentada no Capítulo 3.

Além disso, ela já vem com dois controladores completamente prontos, que permitem o usuário, tanto iniciante quanto o avançado, já em seu primeiro uso passar qualquer número de posturas desejadas ao Pioneer, que as alcançará obedecendo o controlador escolhido. A declaração dos ganhos dos controladores também já está explícita, para que qualquer usuário intermediário possa alterá-los, a fim de determinar seu efeito no sistema.

Finalmente, a plataforma apresenta os resultados relevantes da navegação para o usuário, permitindo que este analise mais profundamente o sistema, detectando efeitos de alteração de parâmetros, diagnosticando erros no projeto de futuros controladores, ou apenas mostrando resultados, para que o usuário os utilize da forma que o convir.

Como projetos futuros, tem-se o desenvolvimento de novos controladores e planejadores de trajetória, para que o Pioneer possa desviar de obstáculos, escolher trajetórias ótimas, reduzir o consumo energético e tomar decisões acerca de seu ambiente. Mais ainda, pode-se alterar o software de maneira que este consiga controlar múltiplos robôs, possibilitando assim uma nova gama de testes e experimento, como a formação utilizando estratégia centralizadas e descentralizadas baseadas em consenso de robôs móveis terrestres.

# Referências

- 1 FERREIRA, A. *Desvio Tangencial de Obstáculos para um Robô Móvel Navegando em Ambientes Semi-Estruturados*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2004.
- 2 SILVA, A. L. V. da. *Arquitetura compacta para projeto de robôs móveis visando aplicações multipropósitos*. Tese (Doutorado) — Universidade de São Paulo, 2008.
- 3 THRUN, S. et al. A system for volumetric robotic mapping of abandoned mines. In: IEEE. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*. [S.l.], 2003. v. 3, p. 4270–4275.
- 4 MORRISON, J.; NGUYEN, T. On-board software for the mars pathfinder microrover. In: *Proceedings of the Second IAA International Conference on Low-Cost Planetary Missions*. [S.l.: s.n.], 1996.
- 5 BURTON, J.; WONG, R.; PADHYA, T. Robotic-assisted surgery in the head and neck. *Cancer Control*, v. 22, n. 3, 2015.
- 6 BRANDÃO, A. S. *Controle Descentralizado com Desvio de Obstáculos para uma Formação Líder-Seguidor de Robôs Móveis*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2008.
- 7 LOZANO-PEREZ, T.; COX, I. J.; WILFONG, G. T. *Autonomous robot vehicles*. [S.l.]: Springer Science & Business Media, 2012.
- 8 TODD, D. J. *Walking machines: an introduction to legged robots*. [S.l.]: Springer Science & Business Media, 2013.
- 9 SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011.
- 10 SCIAVICCO, L.; SICILIANO, B. *Modelling and control of robot manipulators*. [S.l.]: Springer Science & Business Media, 2012.
- 11 RABELO, M. F. S. *Navegação de um Robô Móvel do Tipo Uniciclo em Terrenos Não-Planares*. Dissertação (Mestrado) — Universidade Federal de Viçosa, 2011.
- 12 SUSNEA, I. et al. Real-time, embedded fuzzy control of the pioneer 3-dx robot for path following. In: WSEAS. *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. [S.l.], 2008.
- 13 VALADÃO, C. et al. *Adaptação de andador convencional para reabilitação e assistência a pessoas com restrições motoras*. [S.l.]: Julho, 2014.

- 14 MARTINS, F. N. et al. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, Elsevier, v. 16, n. 11, p. 1354–1363, 2008.
- 15 PINHEIRO, E. G.; ALBERTO, T. C. Teleoperando robôs pioneer utilizando android. *arXiv preprint arXiv:1501.02475*, 2015.
- 16 BORENSTEIN, J. et al. Where am i? sensors and methods for mobile robot positioning. *University of Michigan, Citeseer*, v. 119, n. 120, p. 27, 1996.
- 17 BORENSTEIN, J.; FENG, L. Measurement and correction of systematic odometry errors in mobile robots. *Robotics and Automation, IEEE Transactions on, IEEE*, v. 12, n. 6, p. 869–880, 1996.
- 18 MARTINS, R. J. et al. Localização em robótica terrestre: Fusão entre odometria por múltiplos encoders e gps. *X Simpósio Brasileiro de Automação Inteligente, Sao Joao del-Rei, Brasil*, 2011.
- 19 BEZERRA, C. G. *Localização de um robô móvel usando odometria e marcos naturais*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2004.
- 20 VIDYASAGAR, M. *Nonlinear systems analysis*. [S.l.]: Siam, 2002.