Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica e de Computação

Controle inteligente de um robô móvel omnidirecional com tomada de decisão utilizando aprendizagem por reforço

Victor Ramon Firmo Moreira

Natal - RN

10 de junho de 2021

Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica e de Computação

Controle inteligente de um robô móvel omnidirecional com tomada de decisão utilizando aprendizagem por reforço

Victor Ramon Firmo Moreira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Elétrica e de Computação (PPGEEC) da Universidade Federal do Rio Grande do Norte como parte dos requisitos para a obtenção do título de **Mestre em Engenharia Elétrica e de Computação**, orientado pelo Prof. DSc. Wallace Moreira Bessa.

Natal - RN

10 de junho de 2021

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Moreira, Victor Ramon Firmo.
Controle inteligente de um robô móvel omnidirecional com tomada de decisão utilizando aprendizagem por reforço / Victor Ramon Firmo Moreira. - 2021.
68 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, RN, 2021.
Orientador: Prof. Dr. Wallace Moreira Bessa.

1. Controle não linear - Dissertação. 2. Linearização por realimentação - Dissertação. 3. Redes neurais artificiais - Dissertação. 4. Epsilon-Greedy - Dissertação. 5. Robotino - Dissertação. I. Bessa, Wallace Moreira. II. Título.

RN/UF/BCZM CDU 681.511.4

Elaborado por Ana Cristina Cavalcanti Tinoco - CRB-15/262

Agradecimentos

Agradeço primeiramente à minha família, meu pai José Melquizedeque, minha mãe Rita Luzeth e meu irmão Luccas Gabriel, muito obrigado por todo o apoio que me deram e por toda a paciência que tiveram comigo ao longo desse período, cada etapa cumprida até hoje tem uma enorme contribuição de vocês, amo vocês.

A minha parceira Jéssica Jales, por saber que quando estava calado, significava que estava preocupado com os prazos do trabalho. Obrigado pelas conversas, atenção, conselhos, incentivo, por passar mais de dez horas diárias me ajudando nos experimentos, e pelas incontáveis ajudas que você me deu até hoje. Gratidão.

Ao Prof. D.Sc. Wallace Moreira Bessa, pela confiança depositada em me receber como orientando desde meu trabalho de conclusão de curso da graduação, pelos ensinamentos passados e pela amizade (líder do RoboTeAM).

Aos amigos e membros do Grupo de Estudos em Robótica e Aprendizado de Máquina (RoboTeAM) pelo acolhimento, ensinamentos e apoio que me proporcionaram nesses últimos anos. Em especial, M.Sc. Eng. Gabriel Lima, Eng. B.Sc. Lucas Solano e Eng. Vitor Vale pelo enorme auxílio durante a etapa experimental do trabalho e pelas dúvidas sanadas. Os demais membros do grupo cito, em ordem alfabética: B.Sc. Aissa Cavalcante, Eng. B.Sc. Diago Xavier, Eng. B.Sc. Gabriel Baumann, B.Sc. Júlio Freire (Bira), Eng. B.Sc. João Lucas (Jota), B.Sc. Lidiane Rodrigues, B.Sc. Nicolas Firmo e Prof. D.Sc. Philippe Medeiros.

Ao meu *Sihing*, Aécio Dantas, da Escola de Kung Fu Lung Fu por estar disponível principalmente nos momentos de dificuldade me fazendo recuperar a força de vontade com seus treinos, *Kin Lay*. Minha válvula de escape não poderia ser melhor.

Ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação (PPGEEC) e a Universidade Federal do Rio Grande do Norte (UFRN) pela oportunidade da realização deste trabalho.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e ao Laboratório de Manufatura (LabMan-UFRN) por todo o suporte.

"A Evolução é a Lei da Vida, o Número é a Lei do Universo, a Unidade é a Lei de Deus."

— Pitágoras

Moreira, V. R. F. **Controle inteligente de um robô móvel omnidirecional com tomada de decisão utilizando aprendizagem por reforço**. 54 p. Dissertação de Mestrado (Programa de Pós-Graduação em Engenharia Elétrica e de Computação) - Universidade Federal do Rio Grande do Norte, Natal - RN, 10 de junho de 2021.

Resumo

A evolução dos sistemas robóticos se tornou evidente no decorrer do tempo. Tanto pelos avanços em fabricação mecânica quanto pelos novos algoritmos utilizados, os robôs móveis têm se tornado cada vez mais independentes em suas ações. No que tange às estratégias de aprendizagem de máquina, uma atenção especial vem sendo dada aos algoritmos de aprendizagem por reforço, em virtude de suas semelhanças com o processo de aprendizado biológico. Neste trabalho propõe-se o desenvolvimento de um agente autônomo, combinando estratégias de controle inteligente com algoritmos de tomada de decisão. Para a implementação da estratégia proposta, será utilizado o robô móvel omnidirecional Robotino®. Foram realizadas simulações de atuação do robô que tem por objetivo a exploração espacial de um ambiente, sendo para isso aplicado um modelo matemático específico. Para o controle do sistema, a estratégia de Linearização por Realimentação foi combinada a um compensador baseado em Redes Neurais Artificiais para lidar com as incertezas, eventuais perturbações externas e compensar a dinâmica não modelada. O algoritmo ϵ-greedy, por sua vez, foi escolhido para capacitar o robô no processo de tomada de decisão. Os resultados da implementação experimental mostram que a estratégia de controle inteligente foi eficiente e o agente inteligente proposto foi capaz de explorar o ambiente de maneira efetiva, obtendo uma alta recompensa média.

Palavras-chave: Controle Não Linear, Linearização por Realimentação, Redes Neurais Artificiais, ϵ-greedy, Robotino®.

Moreira, V. R. F. **Controle inteligente de um robô móvel omnidirecional com tomada de decisão utilizando aprendizagem por reforço**. 54 p. Master's Thesis in Electrical and Computer Engineering - Federal University of Rio Grande do Norte,Natal - RN, June 10, 2021.

Abstract

The evolution of robotic systems has become evident over time. Due to the advances in mechanical manufacturing and the new algorithms used, mobile robots have become increasingly independent in their actions. Regarding machine learning strategies, special attention is given to reinforcement learning algorithms, because of its similarities with the biological learning process. This work proposes the development of an autonomous agent, combining intelligent control strategies with decision-making algorithms. For the implementation of the proposed strategy, the Robotino®omnidirectional mobile robot will be used. Simulations of the robot's performance were performed to explore space in an environment, for which a specific mathematical model is applied. For system control, the Linearization by Feedback strategy was combined with a compensator based on Artificial Neural Networks to deal with uncertainties, possible external disturbances disturbances and compensate for unmodeled dynamics. The ϵ-greedy algorithm, in turn, was chosen to enable the robot in the decision-making process. The results of the experimental implementation show that an intelligent control strategy was efficient and the proposed intelligent agent was able to explore the environment effectively, obtaining a high average reward.

Keywords: Nonlinear Control, Feedback Linearization, Artificial Neural Networks, ϵ-greedy, Robotino®.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACO	*Ant colony optimization* - Optimização de colônias de formigas
ANN	*Artificial Neural Network* - Redes Neurais Artificiais
API	*Aplication Programming Interface* - Interface de Programação de Aplicativos
FBL	*Feedback Linearization* - Linearização por Realimentação
IDE	*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado
IMU	*Inertial Measurement Unit* - Unidade de Medição Inercial
LED	*Light Emitting Diode)* - Diodo Emissor de Luz
LDR	*Light Dependent Resistor* - Resistor Dependente de Luz
MAB	*Multi-Armed Bandit* - Bandido de Vários Braços
PWM	*Pulse Width Modulation* - Modulação de Largura de Pulso
RBF	*Radial basis function* - Funções de Bases Radiais
RL	*Reinforcement Learning* - Aprendizagem por reforço
ROS	*Robot Operating System* - Sistema Operacional de Robôs
UCB	*Upper Confidence Bound* - Limite de Confiança Superior
USB	*Universal Serial Bus* - Porta Universal
SMD	*Sliding Mode Differentiator* - Derivador por Modos Deslizantes
VGA	*Video Graphics Array* - Matriz de Gráficos de Vídeo

Lista de símbolos

$\mathbf{A}$	vetor com a quantidade de ações disponíveis
A_i	ação selecionada
$\mathbf{B}$	matriz de inércia
$\hat{\mathbf{B}}$	estimativa de $\mathbf{B}$
$\mathcal{C}$	constante
$\mathbf{c}_i$	vetor dos centros das funções de ativação
$\mathbf{d}$	vetor de incertezas associadas ao sistema
$\hat{\mathbf{d}}$	estimativa para $\mathbf{d}$
$\hat{\mathbf{d}}^*$	estimativa ótima
$\mathbf{f}$	vetor que incorpora os efeitos centrífugos de Coriolis e da força peso
$\hat{\mathbf{f}}$	estimativa de $\mathbf{f}$
$f_1,\ f_2$	funções propostas para a avaliar o comportamento do derivador
g	aceleração gravitacional
h_r	altura do robô
IR_i	sensores infravermelhos
L	função de Lagrange
$\mathcal{L}$	equação de Euler - Lagrange
l	distância das rodas até o centro geométrico do robô
$\mathbf{M}$	$\mathbf{T}_r^\top \mathbf{M} \mathbf{T}_r$
$\mathcal{M}$	$\text{diag}\{m_r,\ m_r,\ m_r R_r^2/2\}$
M_i	motores do robô
m_r	massa do robô
Q	vetor de forças generalizadas
Q_n	média de recompensas

Q_{n+1}	média de recompensas otimizada
q	conjunto de coordenadas generalizadas
q^*	valor ideal
$\mathbf{R}$	vetor de recompensas
R_i	recompensa obtida
R_r	raio do robô
$\mathbf{R}_o^\top$	matriz de transformação
r	raio das rodas
SL	sensor anti colisão
$\mathbf{s}$	medida do erro combinado
$\mathbf{T}$	vetor de episódios
T	energia cinética
t	tempo
t_i	episódio
U	energia potencial
$\mathbf{u}$	sinal de controle
V	função candidata de Lyapunov
$\mathbf{v}$	vetor de velocidade translacional
$\mathbf{W}$	matriz dos pesos
$\mathcal{W}$	região de convergência da rede neural
$\mathbf{w}_i$	vetor dos pesos
$\mathbf{x}$	vetor de estados do sistema
$\tilde{\mathbf{x}}$	erro de rastreamento
$\mathbf{x}_d$	vetor de estados desejados
x_f	sinal filtrado
x_0	posição inicial

$\hat{x}$	estado estimado
X, Y	sistema de coordenadas inercial
x, y	posição do robô no sistema de coordenadas inercial
x_r, y_r	sistema de coordenadas móveis do robô
α_0, α_1	constantes do filtro de primeira ordem
β	metade do ângulo entre as rodas
$\Delta\mathbf{B}$	incertezas associadas à $\mathbf{B}$
$\Delta\mathbf{f}$	incertezas associadas à $\mathbf{f}$
δ	variável limitante da zona morta nos motores
ϵ	probabilidade de exploração do ϵ-greedy
$\boldsymbol{\varepsilon}$	erro mínimo de aproximação
η_i	constante estritamente positiva
γ	constante do derivador
$\mathbf{\Lambda}$	matriz diagonal com entradas positivas λ_i
λ	parâmetro de convergência do controlador
μ_i	limite superior desejado para os $\|\|\mathbf{w}_i\|\|$
$\boldsymbol{\omega}$	vetor de velocidade rotacional
$\varphi_1, \varphi_2, \varphi_3$	velocidade angular nas rodas do robô
φ	orientação do robô no sistemas de coordenadas inercial
$\mathbf{\Psi}$	matriz que incorpora as funções de ativação
$\boldsymbol{\psi}_i$	vetor que incorpora as funções de ativação
$\boldsymbol{\sigma}_i$	vetor das larguras das funções de ativação
$\boldsymbol{\tau}$	torque
$\mathbf{\Theta}$	matriz de inércia
ϑ	constante do derivador
ξ_i	limite superior relacionado ao erro de aproximação
ζ	ruído branco

Sumário

1 Introdução

O estudo da interação social e individual dos animais com o meio físico, ecológico e social é conhecido como etologia, e baseia-se na premissa de que o comportamento animal emergiu de adaptações evolutivas (ANDERSEN et al., 2015). Porém, para se compreender um comportamento, é necessário avaliar sua causa, como se desenvolve e beneficia um organismo, e como ele evoluiu. No campo da etologia, Tinbergen (1963) postulou quatro questionamentos básicos para compreender o comportamento animal, dentre os quais abordam: *i)* como o comportamento evoluiu e se desenvolveu ao longo da linhagem ancestral das espécies, também conhecido como filogênese; *ii)* a necessidade de um determinado comportamento e sua contribuição para a sobrevivência ou reprodução do indivíduo, conhecido como funcionalidade ou valor adaptativo; *iii)* quais eventos e fenômenos desencadeiam o comportamento, chamado de mecanismo ou motivação; e, por fim, *iv)* a ontogênese, que visa compreender como o comportamento surge e se desenvolve ao longo de vida do indivíduo.

A filogênese e funcionalidade do comportamento apresentam valores mais evolutivos, os quais consideram o surgimento e desenvolvimento ancestral da tomada de decisão ideal. No entanto, a seleção natural atua como um processo de seleção otimizado, gerando escolhas que se aproximam da ótima. Em síntese, a evolução do comportamento ocorre em uma população de indivíduos e se manifesta por alterações genéticas provenientes de mutações, recombinações e seleção natural que influenciam a tomada de decisão (FUTUYMA, 2005).

Por outro lado, os eventos motivacionais como busca por alimento e reprodução, prioridades básicas dos animas, influenciam mais diretamente a tomada de decisão (TINBERGEN, 1963). Esses eventos exigem um processo de tomada de decisão bem estruturado, mesmo que de forma não racional, para perpetuação do indivíduo e da espécie (RYER; OLLA, 1995). As capacidades desenvolvidas através de experiências, como aprender a andar e falar, representam eventos ontogênicos, e, assim como os motivacionais, compreendem eventos de aprendizado individual (ENQUIST; GIRLANDA, 2005), os quais não requerem sobrevivência diferencial dos indivíduos, e sim a existência de uma diversidade de experiências por períodos que produzam, com maior frequência, bons resultados (SKINNER, 1938).

Dentre as formas de aprendizado e estratégias de tomada de decisão, se destaca o aprendizado por reforço, o qual toma por base as interações e observações do indivíduo com ambiente. Segundo Rieskamp e Otto (2006) os indivíduos podem aprender diferentes processos de decisão com base no retorno dos resultados recebidos, e a partir daí, podem adaptar as ações ou o próprio processo de decisão às contingências de reforço no ambiente

(STEVENS, 2008).

O aprendizado por reforço pode influenciar diretamente a ação (aprender um comportamento) ou o processo de decisão (aprender uma estratégia). Estudos comportamentais em animais descreveram esses processos de aquisição de comportamentos direcionados a objetivos através dos conceitos de recompensa e punição. Uma recompensa reforça a ação que causa sua entrega, ao passo que uma punição pode ser representada por um sinal negativo de recompensa que reforça a necessidade de se evitar uma ação (THORNDIKE, 1898).

Os estudos sobre aprendizado e comportamento animal incentivaram os pesquisadores de inteligência artificial a encontrar algoritmos computacionais que possibilitassem a programação de agentes através do fornecimento de recompensas e punições, sem a necessidade de especificar como uma tarefa deve ser realizada. Nesse aspecto, a aprendizagem por reforço (RL, do inglês *Reinforcement Learning*) é uma abordagem que permite que o agente programado aprenda por conta própria, interagindo com o ambiente, buscando alcançar da maneira mais eficiente a realização de uma tarefa previamente determinada (HAYKIN, 2001; SUTTON; BARTO, 2018). Dentre as formas de aprendizado de máquina, o aprendizado por reforço é o que mais se aproxima do aprendizado animal, e muitos dos principais algoritmos de aprendizagem por reforço foram originalmente inspirados em sistemas de aprendizado biológico (SUTTON; BARTO, 2018).

Uma das ferramentas tecnológicas utilizadas para estudos comportamentais de aprendizagem por reforço são os robôs móveis, caracterizados como dispositivos autônomos capazes de se movimentar e interagir com o ambiente em que estão inseridos. Esta categoria de robô tem sido bastante explorado em ambientes científicos, e industriais, já que além de executarem tarefas com grande versatilidade e utilização de estratégias de controle inteligente, podem também apresentar a capacidade de predição, adaptação e aprendizagem (BESSA et al., 2017).

A robótica móvel representa uma área com importância crescente no desenvolvimento de sistemas autônomos. Tem como um dos objetivos a construção de máquinas com capacidades e habilidades semelhantes às humanas, como robôs humanoides e veículos autônomos. Além disso, compreende uma maneira eficiente para desenvolvimento e testagem de algorítimos em etapas preliminares da produção de máquinas de grande porte, sendo uma maneira mais barata e acessível para avaliação dos modelos antes da finalização. De maneira geral, independentemente dos mecanismos usados para movimentação de um robô móvel, ou dos métodos usados para avaliação do ambiente, os processos de tomada de decisão das máquinas são importantes na etapa de produção, sendo um dos principais desafios no desenvolvimento de veículos autônomos (DUDEK; JENKIN, 2010).

Algumas abordagens de aprendizado utilizando RL foram aplicadas a robôs móveis, como o desvio de obstáculos e adaptação a mudanças no ambiente sem intervenção humana

(ER; DENG, 2005), a análise de microcélulas de mergulho para o controle e estabilização de profundidade subaquática (BRINKMANN et al., 2018) ou o controle para rastreamento de trajetória (JARDINE et al., 2019; Gao et al., 2021). Kim et al. (2020) aplicaram técnicas de aprendizagem de máquina para o rastreamento de trajetória de um robô flexível e os resultados mostraram que quando submetido a abordagem de aprendizagem por reforço, o robô necessitou de menos treinamento para cumprir o objetivo (KIM; KIM; PARK, 2019; KIM et al., 2020).

Além das estratégias de controle, ferramentas matemáticas baseadas no comportamento animal são aplicadas a robôs móveis, com intuito de compreender os padrões nela encontrados e posteriormente aplicá-los na resolução de problemas, desenvolvimento de novas tecnologias e/ou aperfeiçoamento de sistemas já existentes (RESHAMWALA; P, 2013). Algumas ferramentas ou algoritmos bioinspirados já foram desenvolvidos e utilizados para estudar o comportamento de abelhas (SEELEY; BUHRMAN, 1999), formigas (KAFSI et al., 2016) e peixes (ANDERSEN et al., 2015) e auxiliam não apenas na compreensão sobre esses organismos, mas na construção de tecnologias bioinspiradas.

1.1 Objetivos

Considerando a importância científico-tecnológica dos robôs móveis associados a algoritmos bioinspirados e controle de sistemas mecânicos, este trabalho tem como objetivo geral desenvolvimento de um agente autônomo, combinando estratégias de controle inteligente com algoritmos de tomada de decisão. Para a implementação do objetivo apresentado, será utilizado Robotino® (robô móvel omnidirecional desenvolvido pela Festo®). A missão do Robotino® é maximizar a média de recompensas obtidas a partir da escolha dentre as possíveis opções com probabilidades de recompensas distintas.

Especificamente, este trabalho propõe aplicar a estratégia linearização por realimentação (FBL, do inglês *feedback linearization*), associada a um compensador baseado em redes neurais artificiais (RNA) para efetuar o controle do robô móvel. Ferramenta utilizada para o controle de sistemas não lineares e redução de incertezas associadas ao sistema (FERNANDES et al., 2012; SANTOS; BESSA, 2019). Este trabalho também objetiva utilizar um algorítimo para a tomada de decisão, o ϵ-greedy, que equilibra o fator de exploração ou prospecção do ambiente com base em um parâmetro de probabilidade ϵ buscando maximizar a recompensa média ao longo dos episódios.

A fim de facilitar a compreensão acerca do desenvolvimento do objetivo geral do trabalho, foi subdividido em em cinco objetivos específicos: *i)* modelagem matemática do sistema a ser utilizado; *ii)* o projeto de um controlador inteligente; *iii)* a implementação da estratégia de tomada de decisão; *iv)* simulação computacional integrando o controlador inteligente com o algoritmo de tomada de decisão; e *v)* validação experimental do sistema

autônomo com a realização de experimentos no robô móvel Robotino®.

1.2 Organização do trabalho

O trabalho foi estruturado em seis capítulos, buscando facilitar a apresentação dos principais assuntos abordados ao longo do texto. Os capítulos 1 e 6 representam a introdução e conclusão geral do trabalho, respectivamente. Os capítulos 2 3, 4 e 5 abordam os principais modelos utilizados e resultados obtidos no trabalho, incluindo seções de introdução, metodologia, resultados e discussões cada.

O capítulo 1 apresenta a motivação inicial e situa o leitor sobre a relevância do trabalho. Neste capítulo é feita uma revisão sobre estudos sobre comportamento animal e introduz os primeiros conceitos de aprendizagem por reforço, robôs móveis, estratégias de controle inteligente e algoritmos bioinspirados.

O capítulo 2 aborda as principais categorias de sensores e atuadores utilizados em robôs móveis. Ainda neste capítulo, são apresentados os resultados da modelagem cinética e dinâmica de um robô móvel omnidirecional utilizado no trabalho. Estes resultados servirão de base para realização de etapas posteriores do trabalho.

No capítulo 3, um controlador não linear, com um compensador de incertezas, foi escolhido para realizar o controle de rastreamento de trajetória de um robô móvel omnidirecional. Ao final de cada sub sessão, são apresentadas os resultados das simulações numéricas implementadas com e sem a compensação de incertezas para avaliar o comportamento do controlador.

O capítulo 4 apresenta os conceitos sobre a aprendizagem por reforço e introduz a classe de problemas do tipo MAB (*Multi-Armed Bandit*, em inglês, ou Bandido de Vários Braços, em tradução livre)[1], contexto que engloba o algoritmo escolhido para a tomada de decisão do robô móvel.

No capítulo 5 é efetuada uma descrição detalhada do Robotino®, robô autônomo didático da Festo® utilizado para etapa experimental do trabalho. A implementação para a estratégia de controle proposta no capítulo 3, aplicadas ao Robotino®, tanto simulada quanto realizadas experimentalmente, são apresentadas. Além disso, a combinação da estratégia de controle com o algoritmo de tomada de decisão aplicado Robotino® real são apresentados.

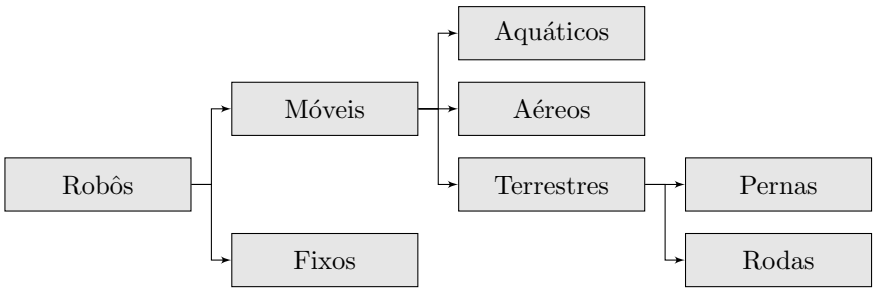
O capítulo 6 trata das considerações finais e conclusões obtidas com este trabalho, além de abordar algumas propostas para trabalhos futuros.

[1] nomenclatura posta que faz alusão ao "bandido de um braço", expressão utilizada para se referir às máquinas de caça-niqueis.

2 Robôs Móveis

Os dispositivos robóticos podem ser categorizados de acordo com a sua estrutura (móvel ou fixa) e ambiente em que atuam (Fig. 2.1). Robôs móveis são ferramentas importantes devido sua capacidade de locomoção e interação com o ambiente de trabalho, na realização de tarefas repetitivas, que oferecem risco para a atividade humana; ou a possibilidade de acesso a locais remotos, ou mal iluminados (MURPHY, 2000). Os robôs móveis têm sido empregados na realização de atividades cotidianas, como em aspiradores de pó, cortadores de grama e robótica educacional, e ainda em atividades mais complexas como tratamento de lixo tóxico, agricultura, exploração subaquática e espacial.

Figura 2.1 – Classificação dos Robôs.



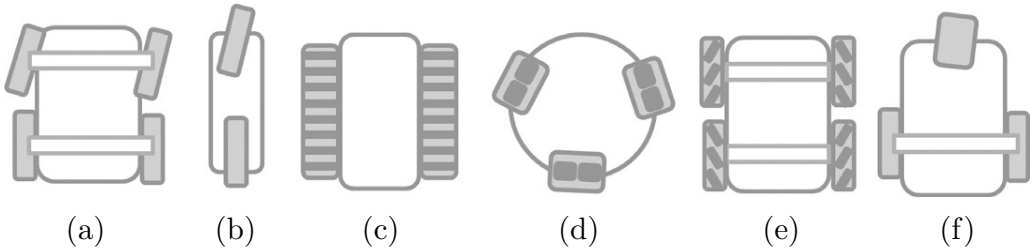
Fonte: Elaborada pelo autor.

Dentre os diferentes tipos de robôs móveis, os robôs terrestres com rodas, representam uma categoria que exige uma configuração específica quanto ao tipo de roda, o sistema de tração, direção e até a forma física do robô. Essas características devem ser escolhidas de maneira apropriada para desempenhar uma função proposta. Portanto, esses robôs podem ser classificados de acordo com sua estrutura e tipos de rodas, como ilustrado na Fig. 2.2.

Robôs com configuração quadriciclo (Fig. 2.2a) são constituídos por pelo menos um eixo direcional e um eixo com rodas motorizadas. A escolha dos eixos é determinada pelo projetista, já que os modelos podem diferir entre si. Geralmente o eixo dianteiro é o direcional e as rodas motorizadas estão situadas no eixo traseiro. Esse arranjo é conhecido como sistema de direção Ackerman[1], que, além de prover maior estabilidade, evita o deslizamento nas rodas, reduzindo erros de odometria. Por outro lado, os veículos de duas rodas (Fig. 2.2b), semelhante a motos ou bicicletas, são atraentes para algumas aplicações devido à sua seção transversal estreita e melhor manobrabilidade. No entanto,

[1] Rudolph Ackermann (1764-1834) - Técnico alemão.

Figura 2.2 – Modelos de Robôs Móveis com Rodas. a) Quadriciclo; b) Bicicleta; c) Esteiras; d) Omni-rodas; e) Rodas *Mecanum*; f) Diferencial.



Fonte: Adaptada de Kagan et al. (2019).

necessitam de um processo de controle contínuo para estabilização, o que exige elevado custo computacional e energético.

Veículos tracionados por esteiras (Fig. 2.2c) introduzem um movimento reto exato e adaptado a terrenos irregulares. Isso ocorre devido à grande área de contato entre a esteira e o solo, promovendo maior tração, o que é adequado quando se lida com esse tipo de cenário. Porém, são caracterizados com baixa eficiência energética para efetuar curvas. Já os robôs diferenciais, com configuração diferencial (Fig. 2.2f), dispõem de uma estrutura formada por duas rodas sobre um mesmo eixo, controladas de maneira independente, juntamente com uma terceira roda adicional para lhes conferir estabilidade, conhecida popularmente por roda boba. O Robô Diferencial é, em geral, o mais utilizado pelos pesquisadores a fim experimentar novas estratégias de controle por possuir uma cinemática simples.

Uma configuração que permite máxima manobrabilidade no plano, ou seja, locomoção em qualquer direção sem necessidade de se reorientar, é a dos Robôs Omnidirecionais (Fig. 2.2d e 2.2e). Essas máquinas são Capazes de traçar qualquer caminho no ambiente de trabalho para atingir os pontos necessários, além de possuírem 3 graus de liberdade para movimentação. Dessa forma, seu posicionamento é definido por três variáveis: duas para representar a sua posição no plano e uma para a rotação em relação ao seu eixo vertical, que é ortogonal ao plano de movimentação. Essa particularidade do sistema omnidirecional se dá graças ao conjunto de rodas que permite uma maior versatilidade na locomoção do robô. Para isso, são utilizadas rodas universais para o conjunto que possui três rodas (Fig. 2.2d) ou Rodas *Mecanum* na estrutura com quatro rodas (Fig. 2.2e). Ambos os tipos de rodas possuem um conjunto de rolos em sua circunferência, porém nas rodas universais os rolos são posicionados perpendicularmente ao eixo rotacional enquanto nas rodas *Mecanum* são posicionados a um ângulo de 45° do eixo da roda.

Para desempenhar uma tarefa de maneira satisfatória, um robô móvel precisa de dispositivos que lhe permitam movimentar-se, extrair informações sobre si e/ou sobre o

ambiente em sua volta. Deslocamento, posição relativa dos objetos presentes no ambiente, identificação da presença de pessoas ou objetos, são algumas das funções exercidas pelos sensores e atuadores presentes em um robô móvel.

2.1 Sensores

Os principais elementos de percepção de sistemas robóticos são os sensores, que podem ser divididos em: *i)* sensoriamento proprioceptivo, relacionado ao movimento, como o reconhecimento de posição e orientação do corpo no espaço, por exemplo, a corrente elétrica do motor, velocidade angular das rodas e tensão da bateria; e *ii)* sensoriamento extraceptivo, referente a dependência com ambiente externo, que podem incluir temperatura, condições de luminosidade, distância ao obstáculo ou intensidade do campo magnético. Existem diferentes tipos de sensores, e segundo Siegwart, Nourbakhsh e Scaramuzza (2011) eles podem ser classificados em sensores de contato físico, medição de movimento, distância e ópticos.

Entre os sensores de contato físico ou proximidade, estão inseridos os sensores anticolisão, um dos sensores mais básicos e usados na robótica. Seu principal objetivo é sinalizar quando o limite da zona de movimento for atingido para que nenhum outro seja realizado nesta direção. São recomendados para praticamente todos os dispositivos móveis, pois garantem uma segurança tanto para o sistema quanto para o operador.

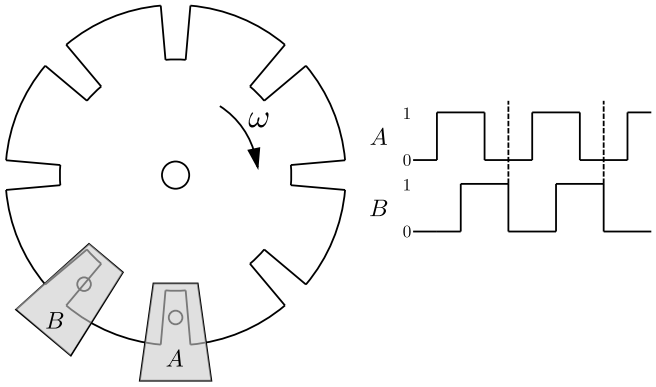
Outra classe de sensores é a de medição de movimento, como giroscópio, acelerômetro e bússola, usados para determinar a orientação e inclinação do robô. A combinação desses sensores compõe a unidade de medição inercial (IMU, do inglês *Inertial measurement unit*), dispositivo capaz de aferir a posição, velocidade e aceleração relativas de um veículo em movimento. Grande parte das IMUs disponíveis no mercado avaliam a variação de seis graus de liberdade do veículo: posição (x, y, z) e orientação (rolagem, arfagem, guinada) no sistema tridimensional de coordenadas.

Os sensores de distância, por outro lado, detectam obstáculos por meio da emissão e recepção de ondas sonoras ou infravermelhas. Ao contrário dos sensores anticolisão, os sensores de distância podem alertar um robô sobre um obstáculo antes de atingi-lo, o que promove uma navegação mais segura. Seu funcionamento consiste em detectar a presença de um objeto através da recepção de um sinal emitido pelo próprio sensor, após refletir em uma determinada superfície. A distância para o obstáculo é calculada com base no tempo entre a emissão e recepção do sinal.

Os codificadores ópticos, também conhecidos como *encoders*, são dispositivos eletromecânicos que contam ou reproduzem pulsos elétricos a partir do movimento rotacional de seu eixo, permitindo a medição de velocidade e/ou posição angular das rodas. Discos encoders em fita (ou discos perfurados), montados coaxialmente com o motor em rotação,

permitem a detecção de presença e ausência dos trechos escuros ou transparentes do disco. Essa configuração gera uma sequência de ondas quadradas que indicam o sentido e a velocidade de rotação (2.3).

Figura 2.3 – Representação operacional do *encoder*.



Fonte: Elaborada pelo autor.

2.2 Atuadores

Os indicadores ou atuadores, por outro lado, são dispositivos que se relacionam com a saída de um sistema. Esses mecanismos permitem a sinalização ou movimentação dos robôs, garantindo a segurança estrutural da máquina e a interação eficiente com o ambiente.

Dispositivos indicadores podem ser acoplados às saídas a fim de notificar as condições do robô e de seus componentes. Apesar de não operar efetivamente com movimentação ou locomoção, os indicadores são dispositivos de saída essenciais em determinados sistemas. Aviso da finalização de uma missão, falha em algum componente e o indicativo da carga da bateria são exemplos de aplicações dos indicadores, que são geralmente feitas por meios de sinais luminosos (LEDs, do inglês *Light Emitting Diode*), sonoros ou visuais (*display*).

Atuadores são equipamentos ou dispositivos que convertem energia hidráulica, pneumática ou elétrica em energia mecânica, proporcionando a movimentação do sistema. Movimentação de cargas ou locomoção são alguns dos propósitos pelos quais os atuadores são utilizados, e independentemente da forma de conversão energética, podem ser classificados em: *i)* pistão, projetados para produzir movimentação linear através de hastes; e *ii)* motores: dispositivos que convertem energia em movimento rotativo em torno de seu próprio eixo (ROMERO et al., 2014).

A maioria dos robôs móveis são projetados com baterias para o armazenamento de energia elétrica, o que por consequência torna a conversão elétrica-mecânica a mais requisitada. Os três principais tipo de motores empregados na robótica móvel podem ser

classificados de acordo com Dudek e Jenkin (2010) em: motores de passo, servomotores e motores de corrente contínua (DC motor).

Os motores de passo possuem um mecanismo de orientação do eixo sem o uso de sensores complexos para monitorar o movimento, operando com elevada exatidão, visto que a posição do eixo de saída se move a uma quantidade controlada por uma série de campos eletromagnéticos ativados e desativados eletronicamente. As aplicações mais usuais dos motores de passos são mecanismos que precisam de posicionamento exato como impressoras 3D e juntas de braços robóticos (ROMERO et al., 2014).

Motores de corrente contínua são os mais usados para realização de deslocamento de robôs baseados em rodas ou esteiras, já que permitem o giro contínuo em ambos os sentidos de rotação e o controle de velocidade. Porém, diferentemente dos motores de passo e servomotores, não possuem um mecanismo de posicionamento preciso, tornando necessário o uso de um *encoder* (DUDEK; JENKIN, 2010).

Já os servomotores são dispositivos eletromecânicos que permitem um controle fino de posição, uma vez que combinam um motor de corrente contínua padrão com um eixo sensor de orientação, permitindo especificar o ângulo de posicionamento através de um controle proporcional em malha fechada. O controle do servo motor é obtido por um sinal com formato seguindo a modulação PWM (PWM, do inglês *Pulse Width Modulation*), tornando os servomotores bastante aplicáveis em dispositivos eletrônicos, como robôs articulados, comandos de aeromodelos e bases de câmeras para monitoramento.

2.3 Modelos Matemáticos de Robôs Omnidirecionais

A movimentação de um robô móvel omnidirecional pode ser modelada por uma ou mais equações diferenciais. O estudo e entendimento dessas equações são de extrema importância para o conhecimento das características e controle dos sistemas. Desse modo, a modelagem efetuada nas seções seguintes inclui a cinemática e a dinâmica do corpo rígido referente a um robô móvel omnidirecional de três rodas. A cinemática estuda o deslocamento, velocidades e acelerações presentes no corpo, enquanto a dinâmica aborda também as forças envolvidas no sistema. O desenvolvimento matemático deste capítulo está concentrado em robôs móveis omnidirecionais e seguiu o proposto por Barreto S. et al. (2014), Raj e Czmerk (2017).

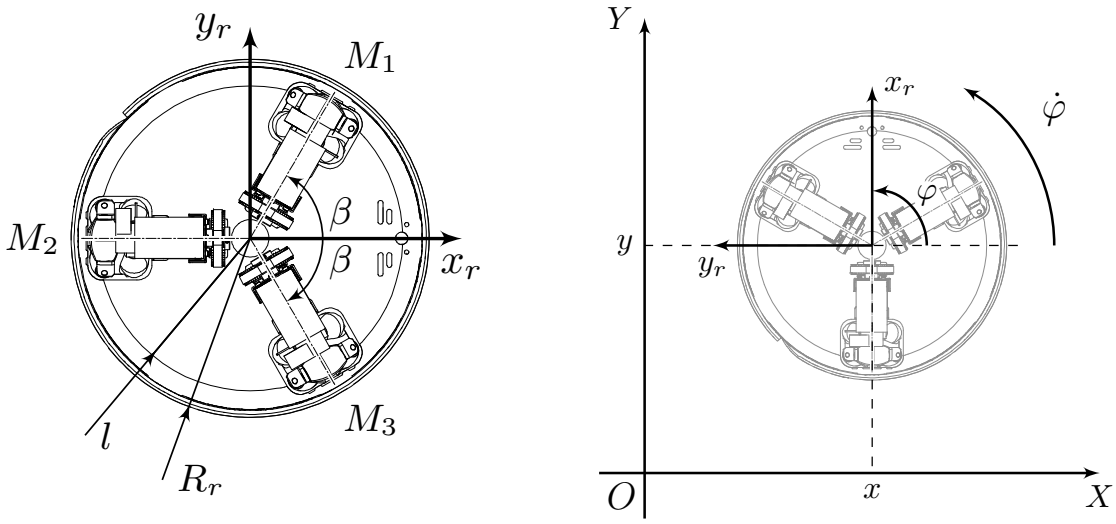
2.3.1 Modelo Cinemático

Um esquema da vista superior de um robô móvel omnidirecional com seu sistema de coordenadas geométricas é mostrado na Fig. 2.4. Considera-se que cada uma das rodas é controlada de maneira independente (rotacionando no sentido horário ou anti-horário)

sem derrapar durante o movimento. Os rolos presentes na roda omnidirecional permitem o livre deslizamento na direção ortogonal ao movimento de rotação.

A modelagem cinemática do sistema consiste em encontrar a relação entre a velocidade angular nas rodas do robô e as velocidades do robô nos eixos X e Y (Fig. 2.4b), e a velocidade de sua orientação φ.

Figura 2.4 – Vista superior de um robô móvel omnidirecional.



(a) Parâmetros geométricos. (b) Sistema de coordenadas.

Fonte: Adaptada de (WEBER; BELLENBERG, 2010).

A partir da Fig. 2.4b temos que

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\varphi}(t) \end{bmatrix} = \mathbf{R}_o^\top(t) \begin{bmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \\ \dot{\varphi}_r(t) \end{bmatrix}, \tag{2.1}$$

onde e a matriz de transformação $\mathbf{R}_o^\top(t)$ para a conversão entre o sistema de coordenadas inercial e o sistema de coordenadas móvel do robô é definida por

$$\mathbf{R}_o^\top(t) = \begin{bmatrix} \cos(\varphi(t)) & \operatorname{sen}(\varphi(t)) & 0 \\ -\operatorname{sen}(\varphi(t)) & \cos(\varphi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

A relação entre a velocidade angular das rodas do robô ($\dot{\varphi}_i$, sendo $i = 1, 2, 3$) e as velocidades do robô no referencial móvel o vetor coluna, $\dot{\mathbf{x}}_r = \begin{bmatrix} \dot{x}_r & \dot{y}_r & \dot{\varphi}_r \end{bmatrix}^\top$, pode ser

observada geometricamente pela Fig. 2.4a e definida conforme a Eq. 2.3,

$$\begin{bmatrix} \dot{\varphi}_1(t) \\ \dot{\varphi}_2(t) \\ \dot{\varphi}_3(t) \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\operatorname{sen}(\beta) & \cos(\beta) & l \\ 0 & -1 & l \\ \operatorname{sen}(\beta) & \cos(\beta) & l \end{bmatrix} \begin{bmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \\ \dot{\varphi}_r(t) \end{bmatrix}, \tag{2.3}$$

sendo r é o raio das rodas e l a distância das rodas até o centro geométrico do robô, e $\beta = 60°$, já que as rodas estão dispostas 120° uma da outra.

Combinando as Eqs. 2.2 e 2.3 temos a relação entre a velocidade angular nas rodas do robô e as velocidades do robô no sistema de coordenadas inercial, resultando na Eq. 2.4.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\varphi}(t) \end{bmatrix} = \mathbf{T}_r \begin{bmatrix} \dot{\varphi}_1(t) \\ \dot{\varphi}_2(t) \\ \dot{\varphi}_3(t) \end{bmatrix} \tag{2.4}$$

Os elementos da matriz $\mathbf{T}_r$ estão descritos no Anexo A.

2.3.2 Modelo Dinâmico

Uma estratégia muito utilizada devido a sua simplicidade conceitual para derivar as equações de movimento é a abordagem Lagrangiana. Que baseia-se na definição da função de Lagrange [2] para sistemas mecânicos:

$$L(\mathbf{q}, \dot{\mathbf{q}}, t) = T(\mathbf{q}, \dot{\mathbf{q}}, t) - U(\mathbf{q}, t), \tag{2.5}$$

que consiste na diferença entre a energia cinética total $T(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ e potencial $U(\mathbf{q}(t))$ do sistema, onde $\mathbf{q}(t)$ é um conjunto de coordenadas generalizadas e $\dot{\mathbf{q}} = \dfrac{d\mathbf{q}}{dt}$ as velocidades generalizadas, sendo definidas em 2.6 e 2.7 o desenvolvimento do modelo do robô.

$$\mathbf{q} = \begin{bmatrix} \varphi_1(t) & \varphi_2(t) & \varphi_3(t) \end{bmatrix}^\top \tag{2.6}$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\varphi}_1(t) & \dot{\varphi}_2(t) & \dot{\varphi}_3(t) \end{bmatrix}^\top \tag{2.7}$$

As equações de movimento para um sistema não-conservativo (ou dissipativo), podem ser obtidas aplicando a equação de Euler-Lagrange (Eq. 2.5) dada por

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q}(t) \tag{2.8}$$

[2] Joseph Louis Lagrange (1736-1813) - Matemático italiano, naturalizado francês.

em que $\mathbf{Q}(t)$ é o vetor de forças generalizadas externas atuando no sistema, o qual na robótica é tratado como o vetor de forças conjuntas e torque $\boldsymbol{\tau}(t)$.

Considerando que o robô móvel se locomove sobre uma superfície plana, a energia potencial é invariável, portanto, a equação de movimento 2.5 levará em conta somente a parcela da energia cinética para o desenvolvimento.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t) = T(\mathbf{q}, \dot{\mathbf{q}}, t) \tag{2.9}$$

A energia cinética total para um corpo rígido (Eq. 2.10) contém a energia translacional somada com a rotacional do corpo rígido.

$$T(\mathbf{q}, \dot{\mathbf{q}}, t) = \frac{1}{2} m_r |\mathbf{v}|^2 + \frac{1}{2} \boldsymbol{\omega}^\top \boldsymbol{\Theta} \boldsymbol{\omega}, \tag{2.10}$$

$$\boldsymbol{\Theta} = \text{diag} \left\{ \frac{h_r^2 + 3R_r^2}{12}, \ \frac{h_r^2 + 3R_r^2}{12}, \ \frac{R_r^2}{2} \right\}, \tag{2.11}$$

sendo $\mathbf{v} = \begin{bmatrix} \dot{x} & \dot{y} & 0 \end{bmatrix}^\top$ e $\boldsymbol{\omega} = \begin{bmatrix} 0 & 0 & \dot{\varphi} \end{bmatrix}^\top$ o vetor de velocidade decomposto em velocidade translacional e rotacional respectivamente, $\boldsymbol{\Theta}$ a matriz de inércia, m_r a massa, h_r a altura e R_r é o raio do robô. Substituindo $\mathbf{v}$, $\boldsymbol{\omega}$ e $\boldsymbol{\Theta}$ na Eq. 2.10 temos

$$T(\mathbf{q}, \dot{\mathbf{q}}, t) = \frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} m_r \dot{y}^2 + \frac{1}{2} \left(m_r \frac{R_r^2}{2} \right) \dot{\varphi}^2. \tag{2.12}$$

A Eq. 2.12 pode ser simplificada escrevendo-a na forma matricial da seguinte maneira:

$$T(\mathbf{q}, \dot{\mathbf{q}}, t) = \frac{1}{2} \dot{\mathbf{x}}^\top \mathbf{M} \dot{\mathbf{x}}, \tag{2.13}$$

em que $\dot{\mathbf{x}}$ é o vetor das velocidades do robô no sistema inercial e $\mathbf{M} = \text{diag}\{m_r, \ m_r, \ m_r R_r^2/2\}$.

Utilizando a Eq. 2.4 para substituir o vetor de velocidades pelas variáveis generalizadas (Eqs. 2.6 e 2.7), temos que

$$T(\mathbf{q}, \dot{\mathbf{q}}, t) = \frac{1}{2} \dot{\mathbf{q}}^\top \mathcal{M} \dot{\mathbf{q}}, \tag{2.14}$$

sendo $\mathcal{M} = \mathbf{T}_r^\top \mathbf{M} \mathbf{T}_r$ os elementos da matriz $\mathcal{M}$ descritos no Anexo A.

Dessa forma, efetua-se as derivadas presentes na Eq. 2.8, sabendo que o vetor de forças externas é dado por $\boldsymbol{\tau}(t) = \begin{bmatrix}\tau_1(t) & \tau_2(t) & \tau_3(t)\end{bmatrix}^\top$, obtém-se as equações de movimento para o sistema.

$$\mathcal{M} \begin{bmatrix}\ddot{\varphi}_1(t) \\ \ddot{\varphi}_2(t) \\ \ddot{\varphi}_3(t)\end{bmatrix} = \begin{bmatrix}\tau_1(t) \\ \tau_2(t) \\ \tau_3(t)\end{bmatrix} \tag{2.15}$$

Por fim, combinando a derivada da Eq. 2.4 com a Eq. 2.15, obtemos uma expressão que relaciona o torque aplicado nos motores e as variáveis de estado $\dot{\mathbf{x}}$.

$$\ddot{\mathbf{x}} = \dot{\mathbf{T}}_r \mathbf{T}_r^{-1} \dot{\mathbf{x}} + \mathbf{T}_r \mathcal{M}^{-1} \boldsymbol{\tau} \tag{2.16}$$

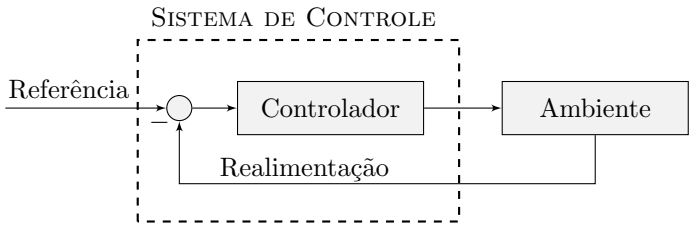
A partir da modelagem matemática do robô móvel omnidirecional, é possível realizar o projeto do controlador. É importante destacar que o modelo obtido não leva em consideração determinadas fontes de incertezas associadas ao sistema real, como o atrito presente nos atuadores, ou presença de ruído nos sensores. O desenvolvimento do controlador será abordado no capítulo seguinte.

3 Controle não linear de robôs móveis

Controle pode ser definido como um processo utilizado para a determinação do comportamento de um sistema, o qual pode ocorrer de forma manual ou automática. De forma simplificada, os sistemas de controle se apoiam em dois tipos de variáveis: a controlada e a manipulada, também chamada de sinal de controle. A variável controlada representa uma grandeza ou condição capaz de ser medida e/ou controlada, e normalmente é a saída do sistema. Em contrapartida, o sinal de controle é uma grandeza ou condição modificada pelo controlador para alterar os valores da varável controlada, representando a entrada do sistema. Assim, controlar significa determinar o valor da variável controlada e aplicar o sinal de controle ao sistema para corrigir ou limitar os desvios do valor medido a partir de um valor desejado (OGATA, 2010).

Para que o controle ocorra de maneira desejada, é preciso que o controlador saiba o quão distante está da referência. Essa avaliação pode ser feita utilizando o controle em malha fechada (Fig. 3.1), que realimenta o controlador com as informações do ambiente e captadas por meio de sensores (OGATA, 2010). Eletrodomésticos, aerogeradores e veículos autônomos possuem sistemas de controle importantes e indispensáveis para seu funcionamento.

Figura 3.1 – Controle em malha fechada.



Fonte: Elaborada pelo autor.

Existem diversas abordagens para controle de sistemas. Dentre elas, o controle não-linear tem se destacado devido suas aplicações no controle de aeronaves, automóveis, sistemas espaciais e robóticos. As estratégias de controle não linear mais empregadas são o controle por modos deslizantes, controle não linear adaptativo, e a linearização por realimentação, também conhecida como Torque Computado (do inglês, *Computed Torque Control*) (SLOTINE; LI, 1991). O controle de linearização por realimentação foi escolhida para o controle do sistema proposto neste trabalho.

3.1 Linearização por Realimentação

A Linearização por Realimentação (FBL, do inglês *Feedback Linearization*) é uma estratégia muito utilizada para o controle de sistemas não lineares. A ideia central é transformar algebricamente a dinâmica de um sistema não linear (total ou parcialmente) em linear em malha fechada, um sistema dinâmico equivalente, porém mais simples. Diferentemente das metodologias convencionais de linearização, o FBL não apresenta a necessidade de aproximação em torno de um ponto de operação, representando o sistema em sua totalidade (SLOTINE; LI, 1991). Isso pode ser melhor compreendido no desenvolvimento a seguir.

Reescrevendo a Eq. 2.16 de forma genérica, como uma Equação Diferencial Ordinária (EDO) para um sistema dinâmico não linear, de segunda ordem, invariante no tempo, MIMO (do inglês, *Multiple Input-Multiple Output*), com n graus de liberdade (GDL) e n entradas, tem-se:

$$\ddot{\mathbf{x}} = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}) + \mathbf{B}\mathbf{u}, \tag{3.1}$$

em que $\mathbf{B} \in \mathbb{R}^{n \times n}$, o vetor $\mathbf{f} \in \mathbb{R}^n$ incorpora os efeitos centrífugos e de Coriolis [1] e da força peso, $\mathbf{x} \in \mathbb{R}^n$ são os estados do sistema e $\mathbf{u} \in \mathbb{R}^n$ representa o sinal de controle.

Consideração 3.1. *Os estados do sistema podem ser medidos ou estimados.*

Para solucionar um problema de rastreamento de trajetória, deseja-se que $\mathbf{x} \to \mathbf{x}_d$ conforme $t \to \infty$, onde $\mathbf{x}_d = \left[x_d, \dot{x}_d, \cdots, \dot{x}_d^{(n-1)}\right]^\top$ é o vetor com a trajetória desejada para cada grau de liberdade do sistema.

Consideração 3.2. *As trajetórias desejadas são conhecidas.*

Deve-se, portanto propor $\mathbf{u}$ de modo que $\tilde{\mathbf{x}} \to 0$, em que $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ é o erro de rastreamento. Assim a lei de controle $\mathbf{u}$ é definida como

$$\mathbf{u} = \mathbf{B}^{-1}\left(-\mathbf{f} + \ddot{\mathbf{x}}_d - 2\mathbf{\Lambda}\dot{\tilde{\mathbf{x}}} - \mathbf{\Lambda}^2\tilde{\mathbf{x}}\right), \tag{3.2}$$

sendo $\mathbf{\Lambda}$ uma matriz diagonal com entradas positivas λ_i.

Substituindo a Eq. 3.2 na Eq. 3.1, temos que

$$\ddot{\tilde{\mathbf{x}}} + 2\mathbf{\Lambda}\dot{\tilde{\mathbf{x}}} + \mathbf{\Lambda}^2\tilde{\mathbf{x}} = \mathbf{0}. \tag{3.3}$$

[1] Gaspard-Gustave Coriolis (1792-1843) - Matemático e engenheiro mecânico francês.

A Eq. 3.3 é uma EDO vetorial com solução convergente em zero, i.e. $\lim_{t\to\infty} \mathbf{x}(t) \to \mathbf{0}$, podendo-se concluir que o sistema em malha fechada é exponencialmente estável. Portanto, quando o sistema possui parâmetros perfeitamente conhecidos, ocorre convergência exponencial do erro a zero.

Para melhor compreensão do método apresentado, considere o modelo não linear de um robô móvel omnidirecional descrito pela Eq. 2.16. A lei de controle obtida pelo método de linearização por realimentação é apresentada na seguinte forma:

$$\boldsymbol{\tau} = \mathcal{M}\mathbf{T}_r^{-1}\left(-\dot{\mathbf{T}}_r\mathbf{T}_r^{-1}\dot{\mathbf{x}} + \ddot{\mathbf{x}}_d - 2\boldsymbol{\Lambda}\dot{\tilde{\mathbf{x}}} - \boldsymbol{\Lambda}^2\tilde{\mathbf{x}}\right), \tag{3.4}$$

resultando em uma dinâmica equivalente em malha fechada para o sistema representada pela Eq. 3.3 que, mais uma vez possui convergência exponencial a zero desde que as entradas da matriz $\boldsymbol{\Lambda}$ sejam positivas.

Simulações computacionais foram realizadas a fim de avaliar o comportamento do controlador. As Eqs. 3.3 e 3.4 foram implementadas computacionalmente em C++, a uma taxa de 50 Hz para o simulador e 25 Hz para o controlador. A solução numérica da equação diferencial de 2ª ordem do sistema foi realizada utilizando o método de Runge-Kutta [2] de 4ª ordem, convertendo o sistema em duas equações de 1ª ordem. A Fig. 3.2 apresenta os resultados obtidos considerando as entradas $\lambda_i = 1$ e o seguinte vetor de estados desejados: $\mathbf{x}_d = \begin{bmatrix} a\cos(ft) & -a\sin(ft) & -ft - \pi/2 \end{bmatrix}^\top$, com a amplitude $a = 0,5$ e frequência $f = \pi/15$ para um tempo de simulação de 60s. Os parâmetros referentes ao sistema usados na simulação, que são os mesmos do Robotino® (sistema a ser utilizado nos experimentos do capítulo 5), estão mostrados na Tab. 3.1.

Tabela 3.1 – Parâmetros do Robô móvel

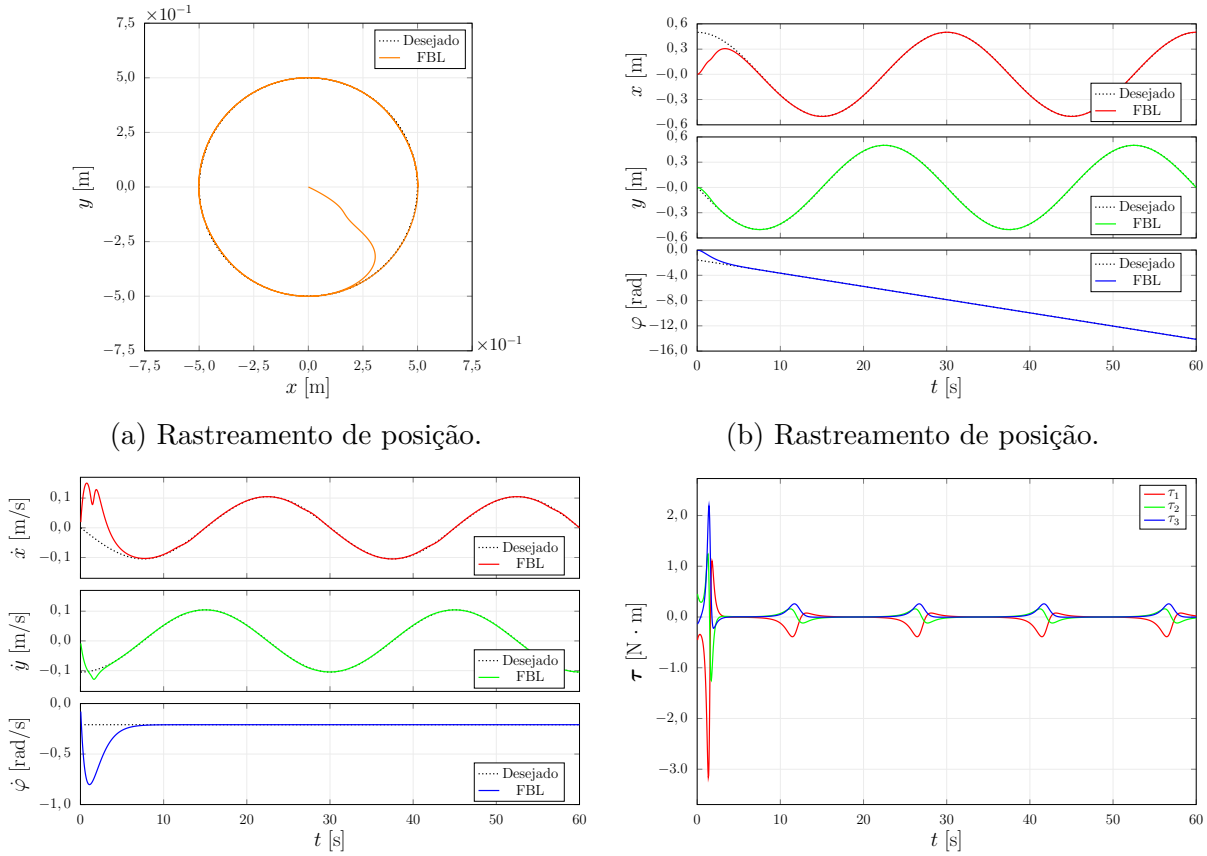
Parâmetro	Unidade	Valor	Descrição
l	[m]	0,294	Distância do CG roda até o CG do robô
β	[°]	60	Metade do ângulo entre as rodas
r	[m]	0,051	Raio das rodas
m_r	[kg]	11,2	Massa do robô
R_r	[m]	0,207	Raio do robô
g	$[m/s^2]$	9,81	Aceleração gravitacional

Fonte: Elaborado pelo autor.

Nas Figs. 3.2b e 3.2c, observa-se que os estados convergiram para as trajetórias definidas por volta dos oito segundos, e se mantiveram nelas sem nenhum erro. Porém, o controlador só foi capaz de realizar o rastreamento de forma eficiente pois o modelo matemático do sistema era inteiramente conhecido. Para confirmar o efeito na presença

[2] Carl David Tolmé Runge (1856–1927) e Wilhelm Kutta (1867–1944) – Matemáticos alemães

Figura 3.2 – Resultados para simulação utilizando FBL para um sistema sem incertezas.



(a) Rastreamento de posição.

(b) Rastreamento de posição.

(c) Rastreamento de velocidade.

(d) Esforço de controle.

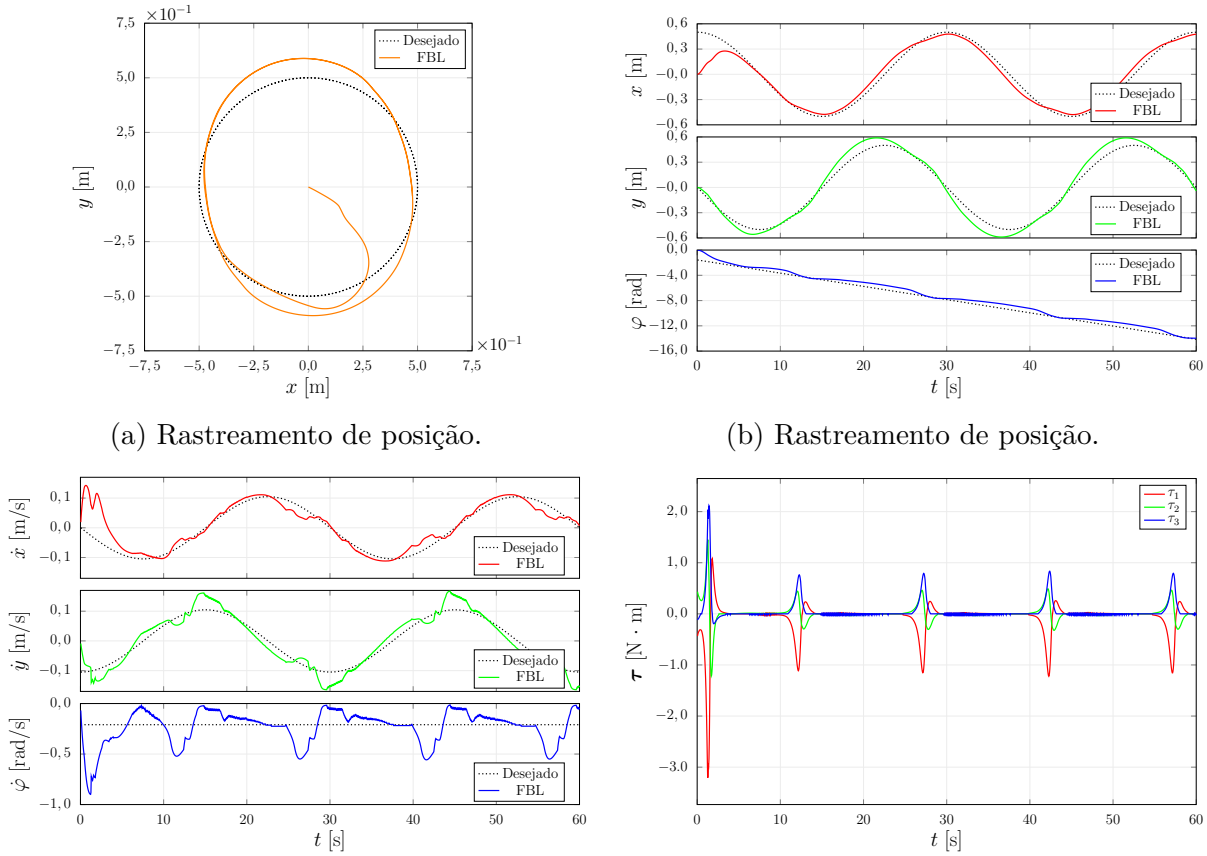
Fonte: Elaborada pelo autor.

de incertezas sobre o controlador, uma nova simulação foi realizada, mas agora com um efeito de zona morta simétrica nos atuadores definida pela Eq. 3.5

$$\tau_i = \begin{cases} \tau_i + \delta & \text{se } \tau_i \leq -\delta, \\ 0 & \text{se } -\delta < \tau_i < \delta, \\ \tau_i - \delta & \text{se } \tau_i \geq \delta \end{cases} \tag{3.5}$$

A partir dos resultados mostrados na Fig. 3.3, com adição da zona morta, verifica-se que a presença de não-linearidades ocasionou um efeito tanto no rastreamento de trajetória de $\mathbf{x}$ quanto de $\dot{\mathbf{x}}$ no controle. Essa perda de performance do controlador fica mais nítida observando o espaço de fase do erro mostrados na Fig. 3.4a. No caso de parâmetros perfeitamente conhecidos (Fig. 3.4a), obtém-se convergência a zero para o erro ($\tilde{\mathbf{x}} \to 0$ para $t \to \infty$), o que não acontece com parâmetros estimados diferentes dos parâmetros do modelo, (Fig. 3.4b). Aparentemente é um erro pequeno, porém, em situações de aplicações reais, maiores incertezas estão associadas ao sistema, ampliando o efeito.

Figura 3.3 – Resultados para simulação utilizando FBL com incertezas nos parâmetros.



(a) Rastreamento de posição.

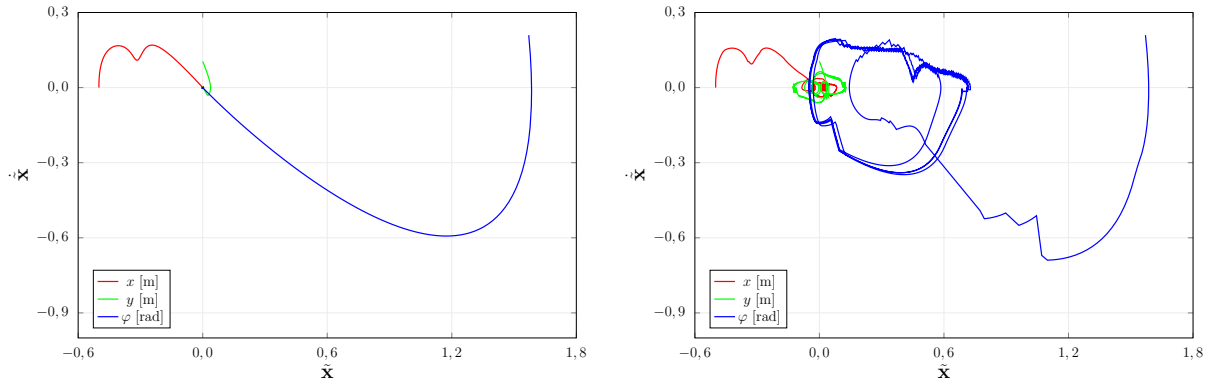
(b) Rastreamento de posição.

(c) Rastreamento de velocidade.

(d) Esforço de controle.

Fonte: Elaborada pelo autor.

Figura 3.4 – Comparação do espaço de fase do erro para o método de Linearização por Realimentação para o modelo com adição de incertezas.



(a) Sem incertezas.

(b) Com incertezas.

Fonte: Elaborada pelo autor.

Melhor desempenho de controle geralmente requer um modelo exato da planta, que na prática pode ser muito difícil de se obter. Conforme os resultados apresentados, observa-se que há uma limitação associada a abordagem de linearização por realimentação com a presença de incerteza acerca dos parâmetros modelados para o sistema (SLOTINE; LI, 1991). Quando o sistema a ser controlado é incerto, é necessária a utilização de estratégias robustas ou a incorporação de compensadores para melhorar o desempenho do controlador. Trabalhos foram realizados combinando a estratégia de linearização por realimentação com algoritmos inteligentes, a fim de melhorar o rastreamento de trajetória de sistemas não lineares incertos (JAGANNATHAN; COMMURI; LEWIS, 1998; BOUTALIS, 2004; FERNANDES et al., 2012; TANAKA; FERNANDES; BESSA, 2013; BESSA et al., 2017; SANTOS; BESSA, 2019).

Tanaka, Fernandes e Bessa (2013) utilizaram a lógica *Fuzzy* para melhorar o controle do posicionamento de um oscilador de Van der Pol. Em contrapartida, Santos e Bessa (2019) e Fernandes et al. (2012) incorporaram às leis de controle, compensadores baseados em Redes Neurais Artificiais para solucionar o problema acerca do posicionamento de um atuador eletro-hidráulico, a fim de reduzir os erros associados a modelagem que faz uso da linearização por realimentação. Esses autores concluíram que a abordagem de controle, utilizando as redes neurais, foi capaz de reconhecer automaticamente a dinâmica não modelada e compensá-la, levando ao rastreamento preciso da posição. Dessa forma, considerando a simplicidade e rigorosidade metodológica da Linearização por realimentação e a eficiência das Redes Neurais em compensar as incertezas de sistemas não-lineares, minimizando o erro de rastreamento, este trabalho propõe a utilização de um compensador baseado em ANN em conjunto com a estratégia de linearização por realimentação.

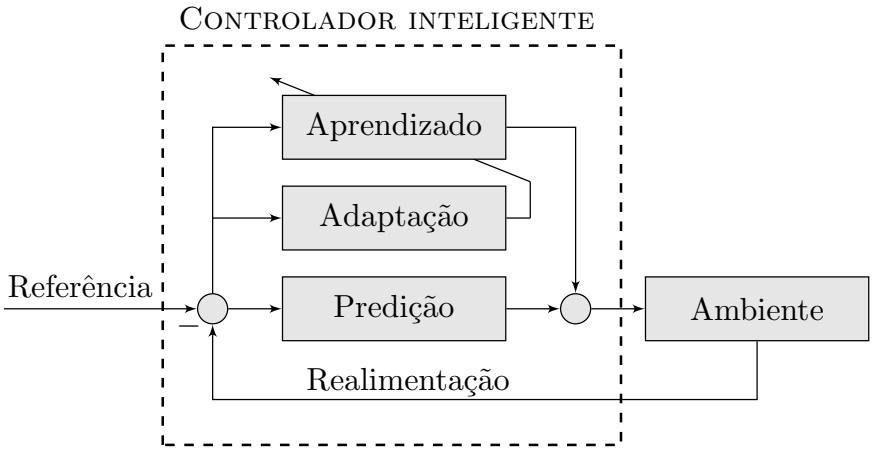
3.2 FBL com compensador baseado em ANN

As redes neurais artificiais são inspiradas em redes neurais biológicas e atuam na construção de sistemas capazes de exibir comportamentos baseados na inteligência humana, como compreensão da linguagem, aprendizagem e raciocínio (BOUTALIS, 2004). Dessa forma, a aplicação de ANN faz com que o sistema de controle emule os principais traços de inteligência: aprendizagem, adaptação e predição (BESSA et al., 2018).

A topologia de controlador inteligente proposta por Bessa et al. (2018) é mostrada na Fig. 3.5. Em seu esquema, os autores caracterizaram predição e aprendizado como responsáveis por incorporar conhecimento ao sistema, permitindo uma ação de controle apropriada. A adaptação, por outro lado, ajusta os mecanismos de aprendizagem ao decorrer do tempo, considerando mudanças tanto na planta quanto no ambiente.

Retomando o sistema mostrado na Eq. 2.16, porém, considerando a presença de

Figura 3.5 – Topologia proposta para um controlador inteligente.



Fonte: Adaptada de (BESSA et al., 2018).

incerteza acerca de $\mathbf{B}$ e $\mathbf{f}$, se apresentando na forma:

$$\begin{aligned} \mathbf{B} &= \hat{\mathbf{B}} + \Delta\mathbf{B}, \\ \mathbf{f} &= \hat{\mathbf{f}} + \Delta\mathbf{f}, \end{aligned} \tag{3.6}$$

onde $\hat{\mathbf{B}}$ e $\hat{\mathbf{f}}$ são, respectivamente, estimativas de $\mathbf{B}$ e $\mathbf{f}$, e $\Delta\mathbf{B}$ e $\Delta\mathbf{f}$ as incertezas associadas às respectivas estimativas, tem-se a seguinte expressão resultante:

$$\ddot{\mathbf{x}} = \hat{\mathbf{f}}(\dot{\mathbf{x}}, \mathbf{x}) + \hat{\mathbf{B}}\mathbf{u} + \mathbf{d}, \tag{3.7}$$

sendo $\mathbf{d} = \Delta\mathbf{f} + \Delta\mathbf{B}\mathbf{u}$ o vetor de incertezas associadas ao sistema.

Para o sistema dado pela Eq. 3.7, considere a lei de controle para o método de linearização por realimentação seguindo a mesma estrutura

$$\mathbf{u} = \hat{\mathbf{B}}^{-1}\left(-\hat{\mathbf{f}} - \hat{\mathbf{d}} + \ddot{\mathbf{x}}_d - 2\mathbf{\Lambda}\dot{\tilde{\mathbf{x}}} - \mathbf{\Lambda}^2\tilde{\mathbf{x}}\right), \tag{3.8}$$

onde $\hat{\mathbf{d}}$ é uma estimativa para $\mathbf{d}$.

Substituindo a Eq. 3.8 na Eq. 3.7 temos que a dinâmica do erro é dada por:

$$\ddot{\tilde{\mathbf{x}}} + 2\mathbf{\Lambda}\dot{\tilde{\mathbf{x}}} + \mathbf{\Lambda}^2\tilde{\mathbf{x}} = \mathbf{d} - \hat{\mathbf{d}}, \tag{3.9}$$

sendo $\ddot{\tilde{\mathbf{x}}} = \ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d$.

Definindo $\mathbf{s}$ como uma medida do erro combinado de $\tilde{\mathbf{x}}$ e $\dot{\tilde{\mathbf{x}}}$,

$$\mathbf{s}(\tilde{\mathbf{x}}) = \dot{\tilde{\mathbf{x}}} + \mathbf{\Lambda}\tilde{\mathbf{x}}, \tag{3.10}$$

temos a dinâmica do erro em função de $\mathbf{s}$:

$$\dot{\mathbf{s}} + \mathbf{\Lambda}\mathbf{s} = \mathbf{d} - \hat{\mathbf{d}}. \tag{3.11}$$

Nesse ponto, considera-se que as redes neurais podem desempenhar funções universais de aproximação, ou seja, $\mathbf{d} = \hat{\mathbf{d}}^* + \boldsymbol{\varepsilon}$, com $\hat{\mathbf{d}}^*$ sendo a estimativa ótima e $\boldsymbol{\varepsilon}$ o erro mínimo de aproximação, uma rede do tipo funções de base radial (RBF do inglês *Radial basis function*) de camada oculta única é adotada para estimar a dinâmica não mensurada:

$$\hat{\mathbf{d}} = \mathbf{W}^\top \mathbf{\Psi}(\mathbf{s}), \tag{3.12}$$

na qual $\mathbf{W} \in \mathbb{R}^{3n\times 3}$ é a matriz dos pesos,

$$\mathbf{W}^\top = \begin{bmatrix} \mathbf{w}_1^\top & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} \\ \mathbf{0}_{1\times n} & \mathbf{w}_2^\top & \mathbf{0}_{1\times n} \\ \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \mathbf{w}_3^\top \end{bmatrix}, \tag{3.13}$$

sendo $\mathbf{w}_i \in \mathbb{R}^n$ o vetor dos pesos para cada grau de liberdade e $\mathbf{\Psi} \in \mathbb{R}^{3n\times 1}$ a matriz que incorpora as funções de ativação,

$$\mathbf{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_1 \\ \boldsymbol{\psi}_2 \\ \boldsymbol{\psi}_3 \end{bmatrix}, \tag{3.14}$$

onde $\boldsymbol{\psi}_i \in \mathbb{R}^n$, representa o vetor com as funções de ativação para cada grau de liberdade, em que n é o número de neurônios.

Para provar estabilidade do controlador utilizando o compensador, a limitação e as propriedades de convergência dos sinais de erro devem garantir a estabilidade do sistema, a qual será investigada de acordo com o Teorema da Estabilidade assintótica de Lyapunov [3] (SLOTINE; LI, 1991).

Definição 3.1. *Uma função $V(x)$ é dita definida positiva se, $V(x) > 0$ e $V(0) = 0$ $\forall$ $x \neq 0$.*

Definição 3.2. *Uma função $V(x)$ é dita definida negativa se $-V(x)$ for definida positiva.*

[3] Aleksandr Mikhailovich Lyapunov (1857-1918) - Matemático e físico russo.

Teorema 3.1. *Dada uma função V definida positiva, o sistema será assintoticamente estável (no sentido Lyapunov) se $\dot{V}$ for definida negativa.*

Isto posto, podemos definir a seguinte função V como candidata de Lyapunov:

$$V = \frac{1}{2}\mathbf{s}^\top \mathbf{s} + \frac{1}{2}\sum_{i=1}^{3} \eta_i^{-1} \tilde{\mathbf{w}}_i^\top \tilde{\mathbf{w}}_i. \tag{3.15}$$

Como η é uma constante estritamente positiva, podemos ver que para $\mathbf{s} = \mathbf{0}$ e $\mathbf{w} = \mathbf{0}$, $V(s) = 0$, além de que para $\mathbf{s} \neq \mathbf{0}$ e $\mathbf{w} \neq \mathbf{0}$, $V(\mathbf{s}) > \mathbf{0}$. Portanto, a função candidata é definida positiva. Agora analisemos $\dot{V}(\mathbf{s})$.

$$\begin{aligned}
\dot{V} &= \mathbf{s}^\top \dot{\mathbf{s}} + \sum_{i=1}^{3} \eta_i^{-1} \tilde{\mathbf{w}}_i^\top \dot{\mathbf{w}}_i \\
&= -\mathbf{s}^\top [\mathbf{\Lambda s} - (\mathbf{d} - \hat{\mathbf{d}})] + \sum_{i=1}^{3} \eta_i^{-1} \tilde{\mathbf{w}}_i^\top \dot{\mathbf{w}}_i \\
&= -\mathbf{s}^\top [\mathbf{\Lambda s} - \boldsymbol{\varepsilon} - (\mathbf{d}^* - \hat{\mathbf{d}})] + \sum_{i=1}^{3} \eta_i^{-1} \tilde{\mathbf{w}}_i^\top \dot{\mathbf{w}}_i \\
&= -\mathbf{s}^\top (\mathbf{\Lambda s} - \boldsymbol{\varepsilon}) - \sum_{i=1}^{3} s_i \tilde{\mathbf{w}}_i^\top \boldsymbol{\psi}_i + \sum_{i=1}^{3} \eta_i^{-1} \tilde{\mathbf{w}}_i^\top \dot{\mathbf{w}}_i \\
&= -\mathbf{s}^\top (\mathbf{\Lambda s} - \boldsymbol{\varepsilon}) + \sum_{i=1}^{3} \eta_i^{-1} \tilde{\mathbf{w}}_i^\top [\dot{\mathbf{w}}_i - \eta_i s_i \boldsymbol{\psi}_i]
\end{aligned}$$

Fazendo a atualização dos pesos $\mathbf{w}_i$ de acordo com $\dot{\mathbf{w}}_i = \eta_i s_i \boldsymbol{\psi}_i$, temos que

$$\dot{V} = -\sum_{i=1}^{3} s_i(\lambda_i s_i - \varepsilon_i) \leq -\sum_{i=1}^{3} |s_i|(\lambda_i |s_i| - \xi_i), \tag{3.16}$$

de modo que $\xi_i \geq |\varepsilon|$ seja um limite superior relacionado ao erro de aproximação. Considerando que $\dot{V}$ é negativa semidefinida quando $|s_i| > \xi_i/\lambda_i$, os limites de $\mathbf{w}$ não podem ser garantidos com $\dot{\mathbf{w}}_i = \eta_i s_i \boldsymbol{\psi}_i$ quando $|s_i| \leq \xi_i/\lambda_i$. Para solucionar esse problema, recorremos ao algoritmo de projeção para garantir que $\mathbf{w}_i$ sempre permaneça dentro de uma região $\mathcal{W} = \left\{\mathbf{w}_i \in \mathbb{R}^n | \mathbf{w}_i^\top \mathbf{w}_i \leq \mu_i^2\right\}$:

$$\mathbf{w}_i = \begin{cases} \eta_i s_i \boldsymbol{\psi}_i & \text{se } ||\mathbf{w}_i||_2 < \mu_i \text{ ou} \\ & \text{se } ||\mathbf{w}_i||_2 = \mu_i \text{ e } \eta_i s_i \mathbf{w}^\top \boldsymbol{\psi}_i \leq \mathbf{0} \\ \left(I - \dfrac{\mathbf{w}_i \mathbf{w}_i^\top}{\mathbf{w}_i^\top \mathbf{w}_i}\right) \eta_i s_i \boldsymbol{\psi}_i & \text{caso contrário,} \end{cases} \tag{3.17}$$

com μ_i representando o limite superior desejado para $||\mathbf{w}_i||_2$.

Portanto, adotando 3.17 juntamente com $||\mathbf{w}_i(0)||_2 < \mu_i$, implica em $|s_i(\tilde{\mathbf{x}})| \leq \dfrac{\xi_i}{\lambda_i}$ e $||\mathbf{w}_i(t)||_2 < \mu_i$ a medida que $t \to \infty$.

Lembrando que $s_i(\tilde{\mathbf{x}}) = \dot{\tilde{x}}_i + \lambda_i \tilde{x}_i$, $|s_i(\tilde{\mathbf{x}})| \leq \lambda_i^{-1}\xi_i$ se torna

$$-\lambda_i^{-1}\xi_i \leq \dot{\tilde{x}}_i + \lambda_i \tilde{x}_i \leq \lambda_i^{-1}\xi_i, \tag{3.18}$$

multiplicando 3.18 por $e^{\lambda_i t}$ tem-se

$$-\lambda_i^{-1}\xi_i e^{\lambda_i t} \leq \frac{d}{dt}(\tilde{x}_i e^{\lambda_i t}) \leq \lambda_i^{-1}\xi_i e^{\lambda_i t}, \tag{3.19}$$

e integrando 3.19 entre 0 e t

$$\int_0^t -\lambda_i^{-1}\xi_i e^{\lambda_i t} \leq \tilde{x}_i e^{\lambda_i t} \leq \int_0^t \lambda_i^{-1}\xi_i e^{\lambda_i t}, \tag{3.20}$$

$$-\lambda_i^{-2}\xi_i e^{\lambda_i t} + \mathcal{C} \leq \tilde{x}_i e^{\lambda_i t} \leq \lambda_i^{-2}\xi_i e^{-\lambda_i t} + \mathcal{C}, \tag{3.21}$$

com $\mathcal{C} = \lambda_i^{-2}\xi_i + \lambda_i|\tilde{x}(0)|$ sendo um valor constante.

Dividindo 3.21 por $e^{\lambda_i t}$, observa-se que para $t \to \infty$ que $\dfrac{\mathcal{C}}{e^{\lambda_i t}} \to 0$, logo

$$-\lambda_i^{-2}\xi_i \leq \tilde{x}_i \leq \lambda_i^{-2}\xi_i. \tag{3.22}$$

Aplicando a Eq. 3.22 na Eq. 3.18, para $t \to \infty$ temos,

$$-2\lambda_i^{-1}\xi_i \leq \dot{\tilde{x}}_i \leq 2\lambda_i^{-1}\xi_i. \tag{3.23}$$

Assim, conclui-se que o controlador proposto definido pelas Eqs. 3.8, 3.12 e 3.17 garante a convergência exponencial do erro de rastreamento para a região fechada a seguir

$$\mathcal{X} = \left\{\tilde{\mathbf{x}} \in \mathbb{R}^2 \mid |\tilde{x}_i^{(j)}| \leq (j+1)!\lambda_i^{j-2}\xi_i \ , \ j = 0, 1 \text{ e } i = 1, 2, 3\right\}, \tag{3.24}$$

onde i e j representam o grau de liberdade e a ordem do erro, respectivamente.

Considerando novamente o modelo utilizado na simulação numérica anterior representado pela Eq. 2.16, é proposta uma nova lei de controle com um compensador baseado em redes neurais artificiais, definida por:

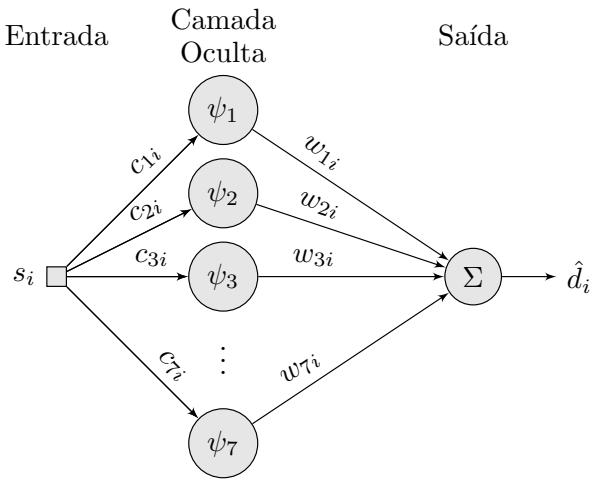
$$\boldsymbol{\tau} = \mathbf{M}\mathbf{T}_r^{-1}\left(-\dot{\mathbf{T}}_r\mathbf{T}_r^{-1}\dot{\mathbf{x}} - \hat{\mathbf{d}} + \ddot{\mathbf{x}}_d - 2\boldsymbol{\Lambda}\dot{\tilde{\mathbf{x}}} - \boldsymbol{\Lambda}^2\tilde{\mathbf{x}}\right), \tag{3.25}$$

sendo $\hat{\mathbf{d}}$ o vetor que contém os compensadores para cada estado.

Com o interesse em verificar o efeito da aplicação do compensador baseado em redes neurais, mais uma simulação foi realizada. Utilizando a mesma porcentagem de incerteza na matriz de massa $\mathbf{M}$ da simulação anterior, e parâmetros referentes ao sistema presentes na Tab. 3.1.

A Fig. 3.6 representa o modelo do neurônio utilizado para cada grau de liberdade do sistema. Em que, s_i são as entradas, $\psi_1, \psi_2, \cdots, \psi_7$ as funções de ativação, $w_{1i}, w_{2i}, \cdots, w_{7i}$ os pesos, Σ representa o combinador linear e $\hat{d}_i$ os sinais de saída.

Figura 3.6 – Rede neural artificial utilizada para cada entrada.



Fonte: Adaptada de (BESSA et al., 2018).

Considerando a rede neural adaptativa, foram utilizadas as RBFs como funções de ativação mostradas na Eq. 3.26.

$$\boldsymbol{\psi}_i(\tau_i, \sigma_{i,k}, c_{i,k}) = \exp\left(\frac{-(\tau_i - c_k)^2}{2\sigma_k^2}\right), \tag{3.26}$$

Para cada grau de liberdade i foram definidos sete neurônios k com centros $\mathbf{c}_i$ da seguinte maneira

$$\mathbf{c}_i = \left[-\frac{\xi_i}{2}, -\frac{\xi_i}{4}, -\frac{\xi_i}{8}, 0, \frac{\xi_i}{8}, \frac{\xi_i}{4}, \frac{\xi_i}{2}\right]^\top, \tag{3.27}$$

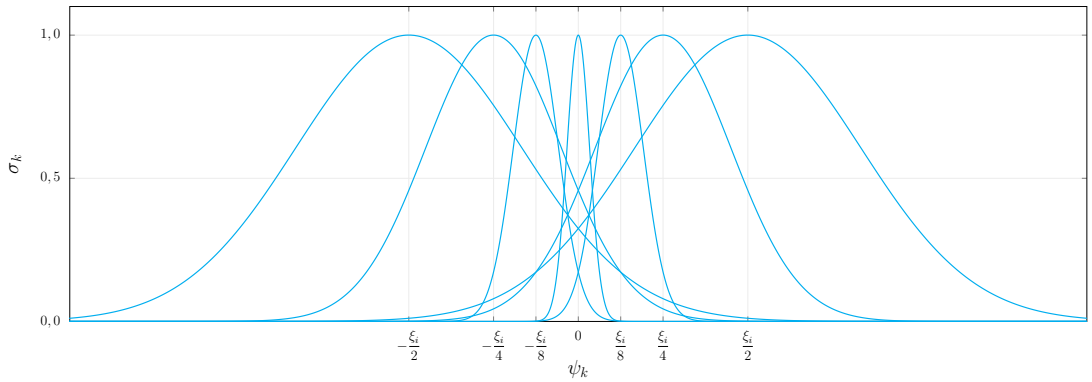
e larguras $\boldsymbol{\sigma}_i$ definidas por

$$\boldsymbol{\sigma}_i = \left[\frac{\xi_i}{3}, \frac{\xi_i}{5}, \frac{\xi_i}{15}, \frac{\xi_i}{30}, \frac{\xi_i}{15}, \frac{\xi_i}{5}, \frac{\xi_i}{3}\right]^\top, \tag{3.28}$$

com a camada limite $\boldsymbol{\xi} = [0,5; 0,5; 3,0]^\top$ e taxas de aprendizagem $\boldsymbol{\eta} = [30,0; 30,0; 35,0]^\top$. Os vetores dos pesos foi inicializado como $\mathbf{w}_i = \mathbf{0}$ e atualizado conforme a Eq. 3.17, como

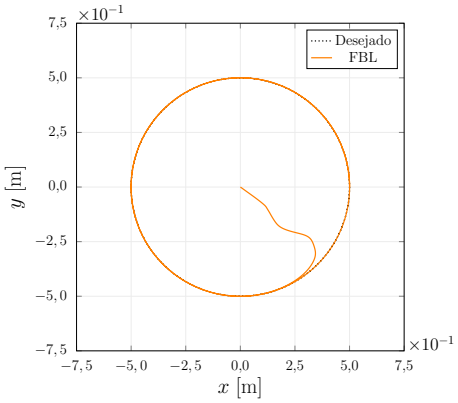
o valor do limitante da norma do vetor dos pesos de $\mu = 0.1$. As funções de ativação selecionadas estão representadas graficamente na Fig. 3.7. Os resultados obtidos para a simulação estão na Fig. 3.8.

Figura 3.7 – Funções de ativação.

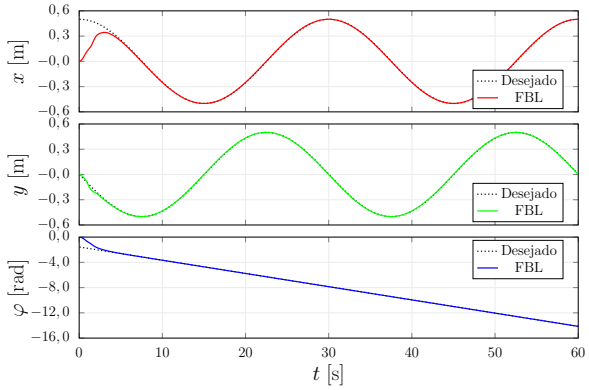


Fonte: Elaborada pelo autor.

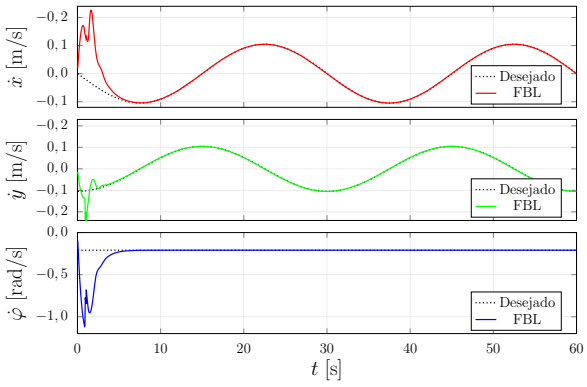
Figura 3.8 – Resultados para simulação utilizando FBL com compensação ANN.



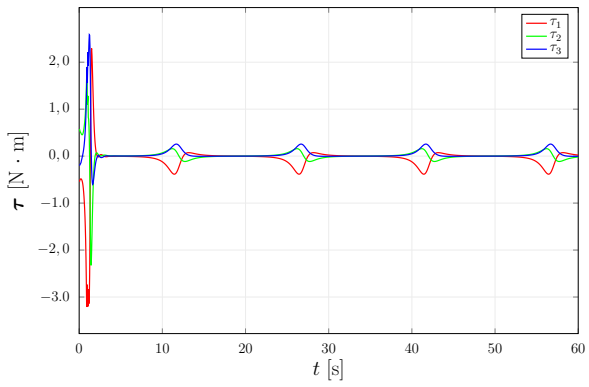
(a) Rastreamento de posição.



(b) Rastreamento de posição.



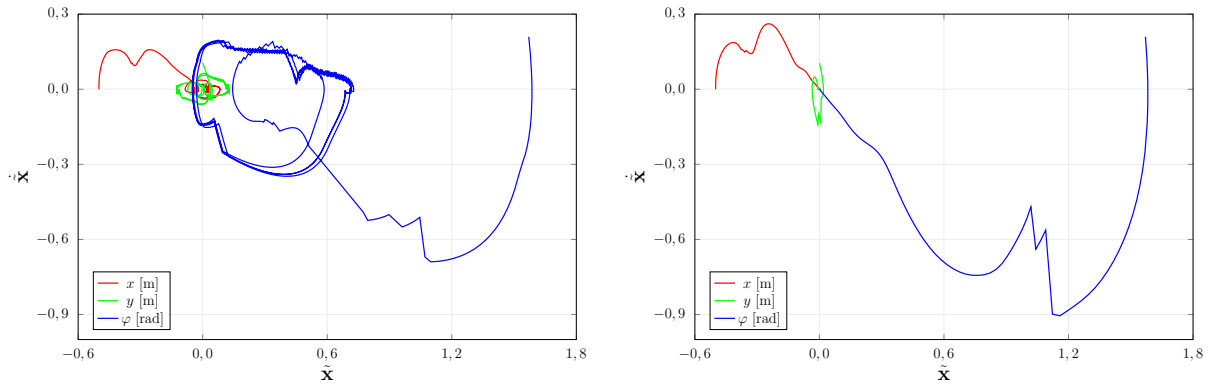
(c) Rastreamento de velocidade.



(d) Esforço de controle.

Fonte: Elaborada pelo autor.

Figura 3.9 – Comparativo entre o espaço de fase do erro para FBL com e sem o compensador baseado em redes neurais artificiais.



(a) FBL. (b) FBL + ANN.

Fonte: Elaborada pelo autor.

Com base nos gráficos da Fig. 3.8, a proposta do controle por meio do método de linearização por realimentação compensado pela rede neural artificial, mesmo na presença da zona morta nos motores, permitiu ao sistema robótico rastrear tanto a trajetória quanto a velocidade desejada. Agora, para demonstrar o desempenho aprimorado do controle inteligente, o plano de fase do erro associado à simulação da seção anterior mostrado na Fig. 3.9b. Comparando com a Fig. 3.9a, que apresenta o erro de rastreamento obtido com a linearização por realimentação convencional, fica fácil verificar que o controlador proposto fornece um erro de rastreamento praticamente nulo, melhorando consideravelmente o resultado da estratégia de controle de linearização por realimentação convencional.

4 Aprendizagem por Reforço

A evolução dos sistemas robóticos tem se tornado cada vez mais evidente ao decorrer do tempo. Após uma fase inicial do desenvolvimento de manipuladores na área da robótica industrial, engenheiros e pesquisadores têm empregado esforços na construção de robôs cada vez mais audaciosos (KIM; LASCHI; TRIMMER, 2013). Esse avanço é possível, em parte, graças às pesquisas na área de aprendizagem de máquinas. A aprendizagem por reforço (*Reinforcement Learning*), juntamente com a aprendizagem supervisionada, e aprendizagem não supervisionada formam as três principais abordagens do aprendizado de máquina. É principalmente por aprender através da interação contínua com o ambiente que a aprendizagem por reforço destaca-se das demais abordagens, sempre avaliando cada ação executada na busca de boas recompensas, como mencionado no capítulo 1.

Além do agente e do ambiente, é possível identificar quatro subelementos principais de um sistema de aprendizado por reforço: *i)* uma política; *ii)* um sinal de recompensa; *iii)* uma função de valor; e, opcionalmente, *iv)* um modelo de ambiente. A política define a forma que o agente interage com o ambiente, e determina o perfil de escolha para as ações a serem tomadas a partir das recompensas obtidas. Já um sinal de recompensa define o objetivo de um problema de aprendizado por reforço, sendo uma resposta positiva ou negativa de acordo com a interação do agente com o ambiente, correspondendo a base principal para alteração da política de aprendizado. Em contrapartida, a função de valor especifica quais são as escolhas mais vantajosas a longo prazo, mediante ao acúmulo de recompensas para escolhas ao decorrer das interações. O último elemento, presente em alguns sistemas de aprendizado é o modelo do ambiente, que de maneira geral inclui modificações ambientais para a tomada de decisão (SUTTON; BARTO, 2018).

Dessa forma, o agente interage com o ambiente, aprendendo através da tomada de decisão, obtendo recompensas positivas ou negativas a cada interação até chegar no seu objetivo. Neste trabalho, um contexto não-associativo, onde o agente atuará em uma situação por vez, foi utilizado através da abordagem do problema MAB (*Multi-Armed Bandit*, em inglês, ou Bandido de Vários Braços, em tradução livre), que representa um dos objetos de estudo da estatística, engenharia elétrica e ciência da computação (RUSSO et al., 2018).

4.1 O problema do MAB

O problema do MAB foi introduzido por Robbins (1952) por meio da construção de estratégias de convergência na seleção populacional, e nomeado em analogia às máquinas de caça-níquel ("bandido de um braço"), a partir dos estudos de Bush e Mosteller (1953).

Nestes estudos os autores desenvolveram um modelo estocástico aplicado ao aprendizado animal, por meio de ensaios em ratos e humanos. Nos experimentos, os ratos deparavam-se com o dilema da escolha entre esquerda ou direita após iniciar, na base de um labirinto em T, sua movimentação, sem conhecer qual dos lados encontraria recompensa. Já os humanos passaram por um experimento similar, onde podiam escolher entre de duas alavancas, esquerda ou direita, que retornavam um pagamento aleatório, retornando ou não recompensa pela decisão tomada.

Durante o processo de aprendizado, um dilema fundamental enfrentado pelo agente é a tomada de decisão diante de opções incertas. O principal objetivo ao solucionar problema do MAB é maximizar a recompensa média obtida. O agente deve avaliar se opta por explorar uma nova opção que parece menos vantajosa, ou prospectar, continuando na opção de maior recompensa média até o momento. O equilíbrio entre exploração e prospecção é a base dos MAB. Prospectar é o mais aconselhável a se fazer para obter a melhor recompensa na próxima ação, todavia, para conhecer a melhor alternativa a longo prazo, é necessário que o agente explore (SUTTON; BARTO, 2018).

Diversos algoritmos foram desenvolvidos com o intuito de solucionar o problema do MAB, entre os mais populares encontram-se os que, de maneira heurística, rastreiam a melhor política possível para explorar o ambiente: ϵ-greedy, *Boltzmann exploration* e *Thompson Sampling*. Também existem e os que se baseiam em uma taxa de arrependimento para as escolhas feitas, UCB (do inglês *Upper Confidence Bound*) (KULESHOV; PRECUP, 2014).

Estudos comportamentais mostram que diante do dilema exploração-prospecção, animais e humanos utilizam duas estratégias para resolve-lo: a primeira é a exploração dirigida, cuja amostragem de opções informativas é incentivada por um "bônus de informação", isto é, prospecção. Já a segunda estratégia é a exploração aleatória (THOMPSON, 1933; WILSON et al., 2014). Apesar do estado do agente ser uma variável importante para a tomada de decisão em sistemas autônomos, a proposta do presente trabalho não o leva em consideração. Este trabalho utilizou o ϵ-greedy, um algoritmo com inspiração biológica, para o processo de tomada de decisão do sistema proposto.

4.2 ϵ-greedy

O ϵ-greedy busca balancear a exploração e prospecção de maneira simples: o agente sempre escolhe prospectar, porém, com uma dada probabilidade ϵ, ele escolhe uma nova ação aleatoriamente. Assim, quanto maior o valor do parâmetro ϵ, mais explorador será o agente. O parâmetro ϵ pode ser alterado no algoritmo para determinar o quanto o agente explora/prospecta.

Considerando que um agente interage com o ambiente no qual está submetido k

vezes, da forma que $\mathbf{T} = \{t_1, t_2, \cdots, t_k\}$ e que em cada tentativa ele se depara com n possibilidades de escolhas distintas, que determina a quantidade de ações disponíveis $\mathbf{A} = \{A_1, A_2, \cdots, A_n\}$ as quais retornam suas respectivas recompensas, $\mathbf{R} = \{R_1, R_2, \cdots, R_n\}$, binárias ou valores reais normalizados entre 0 e 1. Uma maneira simples de encontrar quais ações levam a uma maior recompensa é fazendo o cálculo de média de recompensas, sendo a soma das recompensas já obtidas sobre o número de vezes que essa ação foi selecionada

$$Q_n = \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}. \tag{4.1}$$

O cálculo dessa estimativa pode ser manipulado da seguinte maneira, para gerar menos custo computacional:

$$\begin{aligned} Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\ &= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\ &= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\ &= \frac{1}{n}\left(R_n + (n-1)Q_n\right) \\ &= \frac{1}{n}\left(R_n + nQ_n - Q_n\right) \\ &= Q_n + \frac{1}{n}\left[R_n - Q_n\right], \end{aligned} \tag{4.2}$$

em que R_i e A_i são, respectivamente, a recompensa e ação selecionada na i-ésima rodada.

A cada tentativa, existe pelo menos uma ação que possui valor estimado maior do que as outras. Essa ação é denominada ação gananciosa, e será a escolhida quando o agente decide prospectar.

$$A \doteq \text{argmax}_a Q(a) \tag{4.3}$$

Entretanto, é necessário que o agente explore o ambiente para conhecer a melhor alternativa a longo prazo. Por essa razão, a política ϵ-greedy possui uma probabilidade ϵ de não selecionar a ação máxima mas sim uma outra, de maneira aleatória, por mais que tenha uma recompensa menor. Dessa forma, a medida que o número de etapas aumenta, cada ação será amostrada um número infinito de vezes, garantindo que todo o $Q(A)$ convirja para $q^*(a)$ que representa o valor ideal para cada ação.

O Algoritmo 4.1 tem como base de funcionamento escolher a ação com recompensa máxima com a probabilidade $1 - \epsilon$, ou escolher uma ação aleatória com probabilidade ϵ,

Algoritmo 4.1: ϵ-greedy

Dados: ϵ,t, k, A.
para $t \leftarrow 1$ ***até*** k **faça**

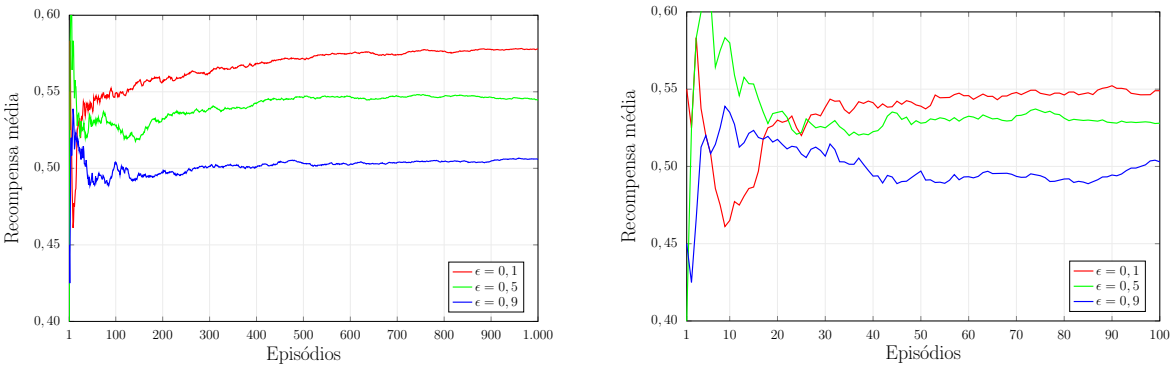
$$A \leftarrow \begin{cases} \text{argmax}_a Q(a) & \text{com probabilidade } 1-\epsilon \\ \text{ação aleatória} & \text{com probabilidade } \epsilon \end{cases}$$

$R \leftarrow bandido(A)$
$N(A) \leftarrow N(A) + 1$
$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$
fim

atualizando a média da amostra para essa ação para atualizar a probabilidade de escolher a ação .

Neste trabalho, foi feita uma análise computacional semelhante a de Sutton e Barto (2018) a fim de verificar a obtenção de recompensa média para valores de ϵ distintos, em um ambiente com duas ações possíveis. A simulação foi realizada para 20 indivíduos, cada um interagindo com o ambiente 1000 vezes para cada valor de ϵ. O ambiente disponibilizou duas opções de escolha: A ou B, com 40% e 60% de chance de retornar recompensa cada. Essas probabilidades de retorno de recompensa foram escolhidas para avaliar a capacidade do agente inteligente em selecionar a opção mais vantajosa frente à duas probabilidades de recompensa semelhantes, porém diferentes, dificultando assim realização da escolha. Dentre os valores de ϵ avaliados, $\epsilon = 0,1$ (pouco explorador), $\epsilon = 0,5$ (intermediário) e $\epsilon = 0,9$ (muito explorador) foram escolhidos e são constantes ao longo dos episódios.

Figura 4.1 – Média de recompensa obtida para cada valor de ϵ com probabilidades distintas para cada braço.



(a) 1000 episódios.

(b) 100 episódios.

Fonte: Elaborada pelo autor.

Conforme observado na Fig. 4.1a, quanto maior o número de episódios de escolha, maior a tendência de $\epsilon = 0,1$ obter a maior recompensa média ao longo dos episódios. Da mesma forma, com um menor número de episódios, como mostra a Fig. 4.1b, os valores de

recompensa média se aproximam. Além disso, é possível observar que para um número de tentativas maior que 100 a tendência da ordem de obtenção de recompensa média se manteve para os valores escolhidos.

Com a estratégia de tomada de decisão adotada no trabalho, implementada computacionalmente, mais um objetivo foi cumprido. A etapa subsequente será integrar os resultados computacionais obtidos até agora, ou seja, a modelagem matemática do sistema (capítulo 2), controlador inteligente (capítulo 3) e estratégia de tomada de decisão (capítulo 4) e validá-los experimentalmente. Essa etapa será apresentada no capítulo seguinte.

5 Robô Móvel Omnidirecional Inteligente

5.1 Robotino®

O sistema utilizado para a validação experimental foi o Robotino® (Fig. 5.1), um robô móvel omnidirecional fabricado pela Festo®, para fins educacionais, de treinamento e pesquisa. O *hardware* do Robotino® é constituído de: *i)* um chassi metálico; *ii)* uma unidade de controle; e *iii)* uma interface de entrada e saída (E/S). O chassi acomoda duas baterias recarregáveis de 12 V e 4 Ah cada, três unidades motoras, a unidade de controle, os sensores de medição de distância infravermelhos e o sensor anticolisão. Sua estrutura também oferece espaço adicional e opções de montagem para outros acessórios, sensores e/ou atuadores. A unidade de controle dispõe de um processador de 32 bits, 300 MHz e sistema operacional Linux, bem como um cartão de memória *flash* de 4 GB com API (*Aplication Programming Interface*) instalada, um ponto de acesso para comunicação sem fio (*wireless*) com taxa de transmissão de até 54 Mbps e alcance de até 100 m. Possui ainda duas portas USB 2.0, uma *Ethernet* e uma VGA (*Video Graphics Array*). Por fim, a interface de E/S que permite conectar sensores ou atuadores adicionais, conta com oito entradas analógicas, oito entradas digitais, oito saídas digitais e dois relés para atuadores adicionais.

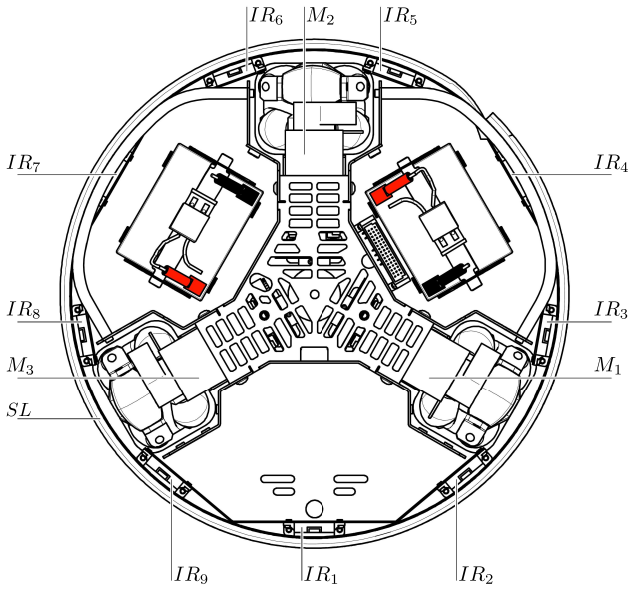
Figura 5.1 – Robotino®.

Fonte: (BLIESENER et al., 2013).

5.1.1 Sensores e Atuadores

Alguns dos sensores já vem integrados ao chassi do Robotino®, dentre eles encontram-se nove sensores de medição de distância infravermelhos, posicionados ao redor do chassi distantes 40° um do outro, com capacidade para medir de 4 a 30 centrímetros de distância. Outro sensor fixado ao redor do chassi é o sensor anti-colisão, que força a parada de qualquer operação em andamento quando detecta o contato com algo que ofereça resistência ao movimento (Fig. 5.2). Além disso, um sensor adicional e bastante relevante para mensurar as variáveis de posição e orientação durante movimentação, é a unidade de medição inercial, uma vez que a aferição da odometria somente com os *encoders* acarreta em um erro cumulativo para longas distâncias percorridas.

Figura 5.2 – Disposição dos Sensores do Robotino®.



IR_1 a IR_9: Sensores Infravermelhos de 1 a 9;

M_1 a M_3: Motores de 1 a 3;

SL: Sensor Anti Colisão.

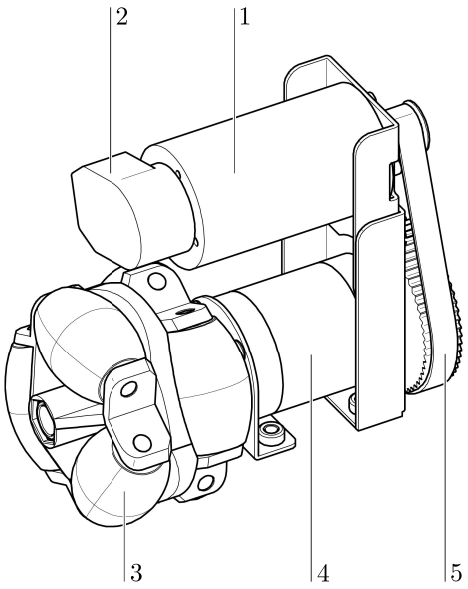
Fonte: Adaptada de (WEBER; BELLENBERG, 2010).

A base robótica móvel omnidirecional Robotino® possui três rodas omnidirecionais, dispostas 120° uma da outra, e cada roda está acoplada à uma unidade motora individualmente controlável, compostas pelos componentes indicados na Fig. 5.3.

5.1.2 Simulador

Antes da etapa experimental, é indicada a realização de testes preliminares, simulando a rotina de trabalho do robô. Dessa forma, para a realização dos testes computacionais deste trabalho, utilizou-se a versão gratuita do Robotino® SIM, um *software* disponibilizado pela Festo® que permite controlar o Robotino® em um ambiente virtual (Fig. 5.4). Apesar de algumas limitações, como não poder modificar o posicionamento de objetos dentro do ambiente simulado, a versão gratuita supre as necessidades, e simula satisfatoriamente

Figura 5.3 – Unidade Motora do Robotino®.

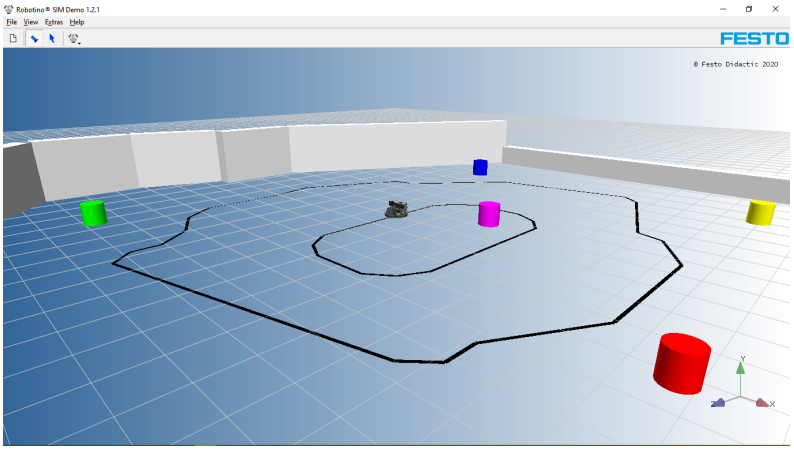


1. Motor 24V CC
2. Encoder Incremental
3. Roda omnidirecional Mecanum
4. Redutor com relação de transmissão de 16:1
5. Correia dentada

Fonte: Adaptada de (WEBER; BELLENBERG, 2010).

o despenho dos sensores, atuadores e bateria. O programa é compatível com diversos tipos de linguagens de programação, como Robotino® *View* (software para programação em blocos), C++, Java, .Net, LabVIEW, MATLAB/Simulink e ROS (*Robot Operating System*). Dentre as opções, a selecionada para a escrever o algoritmo de controle foi a linguagem C++, por ser a mesma usada para realizar as simulações numéricas, além de oferecer maior flexibilidade para edição e boa velocidade de conexão com a API 1.0. Em razão disso, o Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) utilizado foi o Microsoft® Visual Studio 2019e o endereço de IP que dá acesso ao Robotino® SIM foi o definido por padrão como: 127.0.0.1:8080.

Figura 5.4 – Ambiente do Robotino® SIM.



Fonte: Elaborada pelo autor.

5.2 Derivador por modos deslizantes

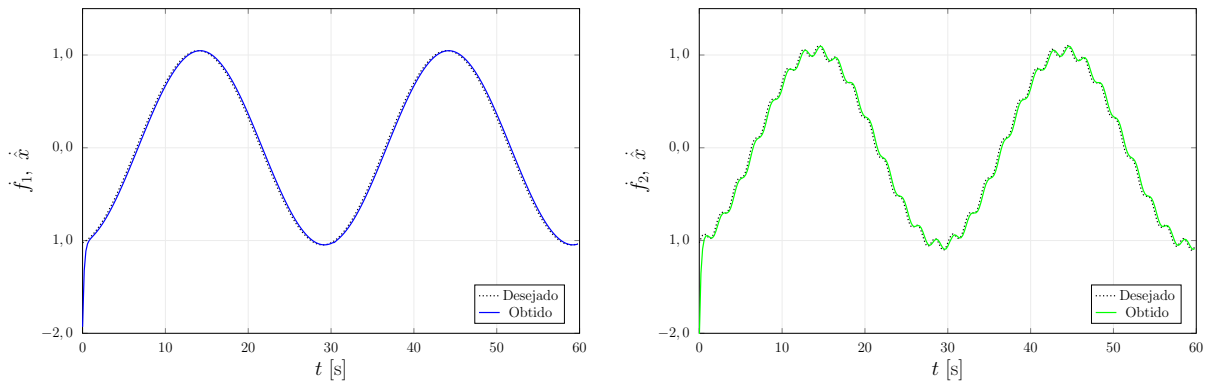
Com a unidade de medição inercial é possível aferir somente as variáveis de estado referentes a posição (x, y e φ). Entretanto, existe uma forma de obter os estados de velocidade e satisfazer a consideração de que todos os estados são conhecidos (Consideração 3.1). Com esse propósito, Bessa et al. (2018) utilizaram um derivador por modos deslizantes (SMD, do inglês *Second-order Sliding Mode Differentiator*) para obter tanto a primeira quanto a segunda derivada da posição de um sistema. Contudo, o controlador proposto neste trabalho é de segunda ordem, sendo necessário tornar o vetor de velocidades do sistema observável. O estado estimado $\hat{x}$ pode ser obtido a partir de um sinal de entrada conhecido x de acordo com

$$\dot{\hat{x}} = -\vartheta \text{sat}\left(\frac{\hat{x} - x}{\gamma}\right), \tag{5.1}$$

onde ϑ e γ são constantes estritamente positivas.

Para comprovar o funcionamento do método, uma simulação foi feita com o propósito de estimar as derivadas das funções $f_1 = -5\sin(\pi t/15)$ e $f_2 = -5\sin(\pi t/15) + 0,02\cos(\pi t)$. O método de Runge-Kutta de 4ª ordem foi utilizado para resolver a Eq. 5.1 com taxa de amostragem de 5 Hz (taxa de amostragem do *software*), para um tempo de 30 s de simulação. Os resultados da simulação adotando $\vartheta = 5$ e $\gamma = 0,5$ são mostrados na Fig. 5.5.

Figura 5.5 – Resultados para simulação do SMD.



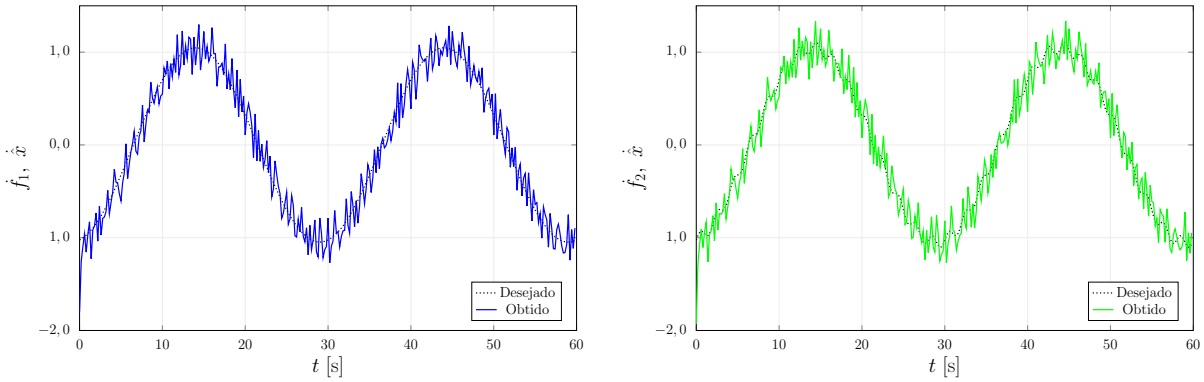
(a) $\dot{f}_1 = -\dfrac{\pi}{3}\cos\left(\dfrac{\pi}{15}t\right)$. (b) $\dot{f}_2 = -\dfrac{\pi}{3}\cos\left(\dfrac{\pi}{15}t\right) - 0,02\pi\sin(\pi t)$.

Fonte: Elaborada pelo autor.

Os resultados obtidos com essa simulação mostram que o derivador segue fielmente o estado desejado. Porém, no contexto de instrumentação, os sinais captados possuem flutuação aleatória de tensão proveniente da movimentação dos elétrons livres em circuito elétrico (ruído térmico). Para simular este fenômeno, um ruído de média nula (ruído

branco) foi adicionado as funções f_1 e f_2, com desvio-padrão de 0.05. O efeito causado pelo ruido é mostrado na Fig. 5.6.

Figura 5.6 – Resultados para simulação do SMD com adição de ruído.



(a) $\dot{f}_1 = -\frac{\pi}{3}\cos\left(\frac{\pi}{15}t\right) + \zeta.$ (b) $\dot{f}_2 = -\frac{\pi}{3}\cos\left(\frac{\pi}{15}t\right) - 0,02\pi\sin(\pi t) + \zeta.$

Fonte: Elaborada pelo autor.

A partir dos resultados, podemos observar que um pequeno ruído na função possui grande efeito em sua derivada, o que pode interferir na ação de controle. A suspeita para que não fosse possível um melhor resultado para a derivada é a baixa taxa de amostragem do simulador. Para resolver esse problema, utilizou-se o filtro, o qual está proposto na seção seguinte.

5.3 Filtro

Um filtro passa-baixas de primeira ordem foi utilizado afim de atenuar o efeito causado pelo ruído nas derivadas das funções. O filtro é definido pela seguinte expressão:

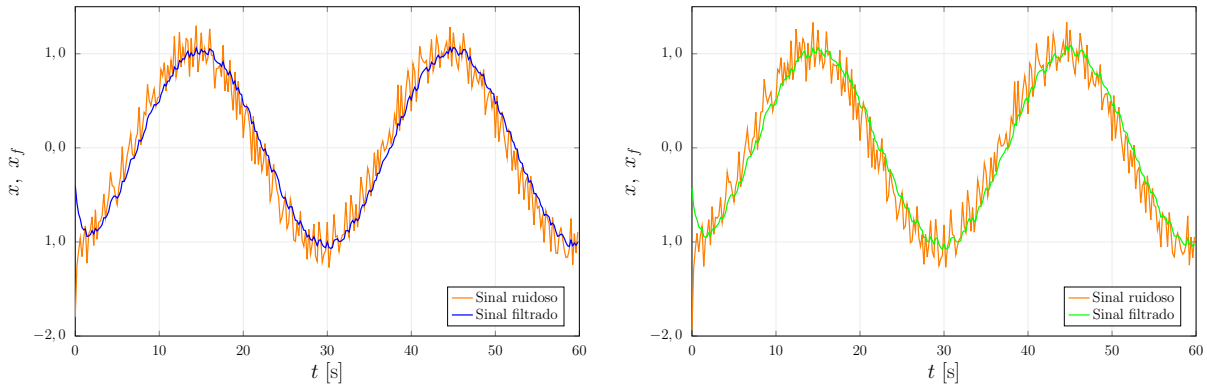
$$\dot{x}_f = \alpha_0 x - \alpha_1 x_f \tag{5.2}$$

onde x_f é o sinal filtrado de x e α_0 e α_1 são constantes positivas.

As funções da Fig. 5.6 foram usadas em uma simulação em C++ para analisar o sinal da derivada com ruído após a aplicação do filtro. O Método de Runge-Kutta de 4ª ordem foi utilizado para resolver a Eq. 5.2, com taxa de amostragem de 5Hz. As constantes foram $\alpha_0 = \alpha_1 = 1$. O resultado é mostrado na Fig. 5.7.

Nota-se que o filtro aplicado reduz consideravelmente o ruído em ambos os casos. Porém, uma certa oscilação juntamente com um pequeno atraso no sinal ainda são observados.

Figura 5.7 – Resultados para simulação do filtro passa-baixa.



(a) $\dot{f}_1 = -\frac{\pi}{6}\cos\left(\frac{\pi}{30}t\right) + \zeta.$ (b) $\dot{f}_2 = -\frac{\pi}{6}\cos\left(\frac{\pi}{30}t\right) + 0,02\pi\cos(\pi t) + \zeta.$

Fonte: Elaborada pelo autor.

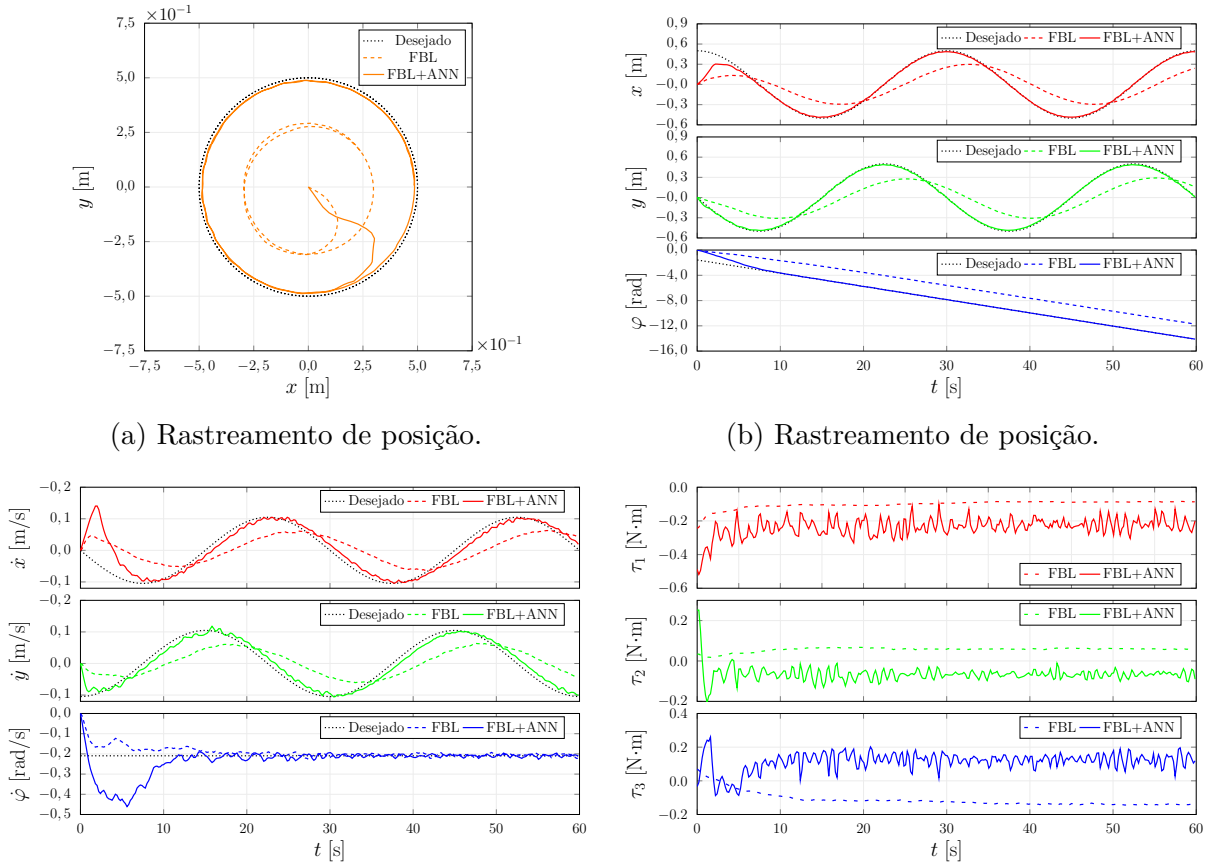
5.4 Controle do Robotino®: Implementação no Simulador

Uma nova simulação foi realizada neste trabalho para verificar o desempenho do controlador e a contribuição da rede neural para lei de controle no Robotino® SIM. O modelo da Eq. 2.16, e as leis de controle obtidas no capítulo 3 (Eq. 3.2 e 3.8) foram utilizados.

O Robotino® possui, como variáveis de entrada, a velocidade angular de cada motor e não o torque $\boldsymbol{\tau}$ conforme o que foi feito nas simulações numéricas (capítulo 3). Já que não tem como medir esse parâmetro diretamente e a dinâmica do atuador não pode ser negligenciada, é preciso realizar uma conversão dessas variáveis. Para isso, o filtro passa-baixa de primeira ordem introduzido anteriormente (Eq. 5.2) foi utilizado para simular a dinâmica do atuador, com $\alpha_0 = 29,0$ e $\alpha_1 = 1,0$ (BESSA et al., 2017). Como se trata de uma análise comparativa entre a simulação numérica e o simulador Robotino® SIM, os estados desejados, frequência de amostragem e parâmetros referentes ao sistema são os mesmos da simulação numérica realizada no 3. Os resultados obtidos no simulador para utilização do FBL e ANN são mostrados na Figura 5.8.

Analisando os gráficos obtidos com a simulação no Robotino® SIM, é possível observar que as não linearidades e incertezas são maiores que as consideradas na simulação numérica, visto que o controlador somente com FBL adquiriu um erro residual mais elevado. Além disso, a adição do compensador, mais uma vez, mostrou uma elevada contribuição no rastreamento de trajetória, e isso é melhor observado comparando os planos de fase para ambos os casos na Fig. 5.9.

Figura 5.8 – Resultados no Robotino® SIM utilizando FBL e FBL com compensação ANN.



(a) Rastreamento de posição.

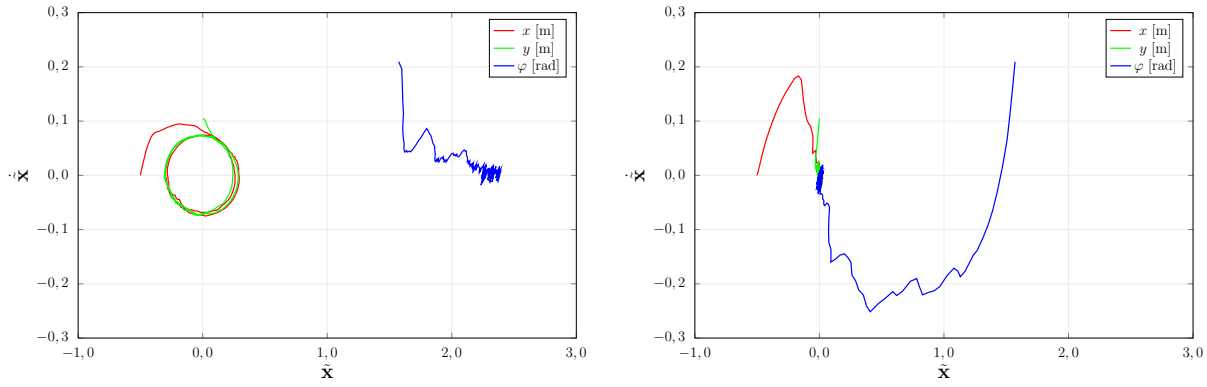
(b) Rastreamento de posição.

(c) Rastreamento de velocidade.

(d) Esforço de controle.

Fonte: Elaborada pelo autor.

Figura 5.9 – Plano de fase para o Robotino® SIM utilizando FBL com compensação ANN.



(a) Plano de fase FBL.

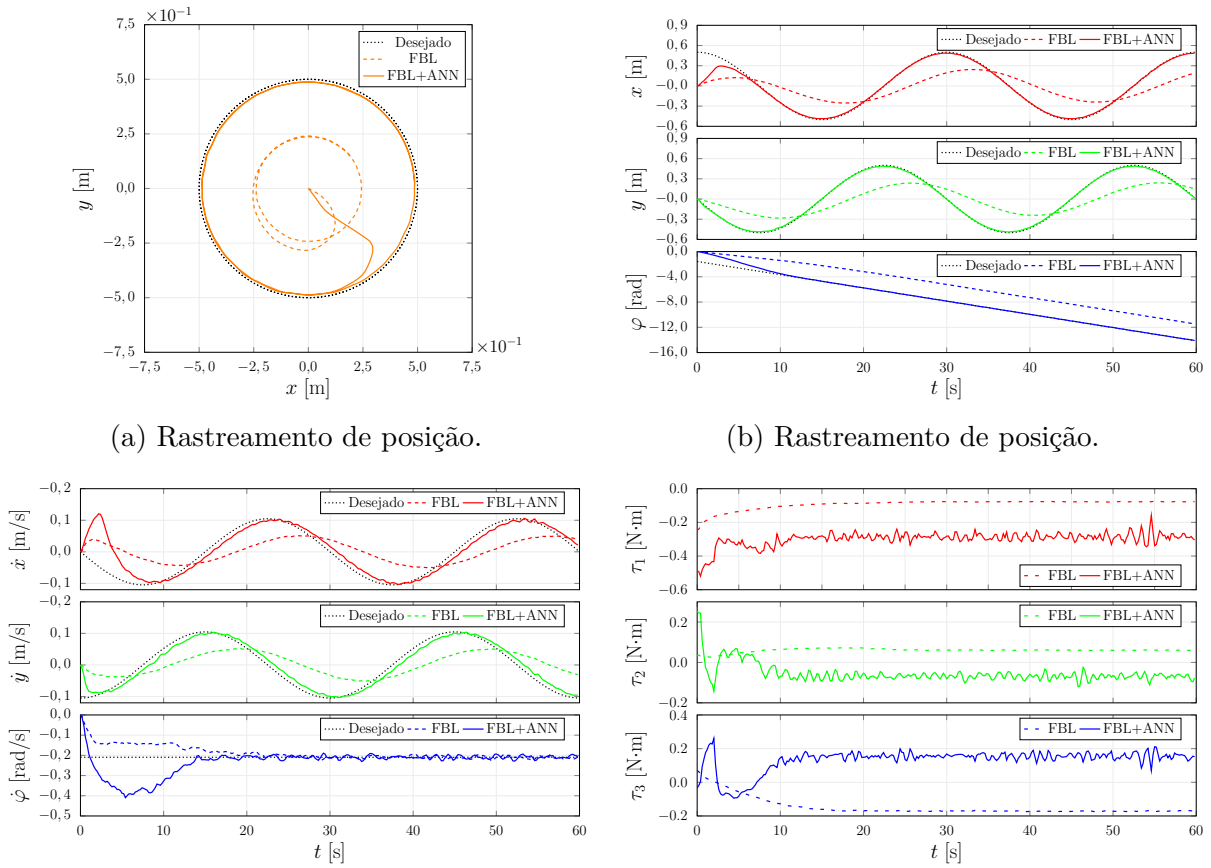
(b) Plano de fase FBL+ANN.

Fonte: Elaborada pelo autor.

5.5 Controle do Robotino®: Implementação Experimental

Para a etapa de validação experimental do controlador, todas as condições de trajetória utilizadas na etapa de simulação foram implementadas no Robotino® real. Para isso, um roteador foi acoplado ao Robotino® para acessá-lo remotamente. O nome da rede e o endereço IP coincidem com informações na placa de identificação anexada ao Robotino®, Nome da rede: Robotino.008.060 e Endereço IP: 172.26.201.2. Além disso, uma das constantes do filtro de primeira ordem utilizadas no experimento foi alterada para $\alpha_0 = 35,0$ e os pesos da rede neural são novamente aprendidos. Os demais parâmetros em ambos os controladores são os mesmos utilizados no Robotino® SIM. Os resultados experimentais são mostrados na Fig. 5.10.

Figura 5.10 – Resultados no Robotino® utilizando FBL e FBL com compensação ANN.



(a) Rastreamento de posição.

(b) Rastreamento de posição.

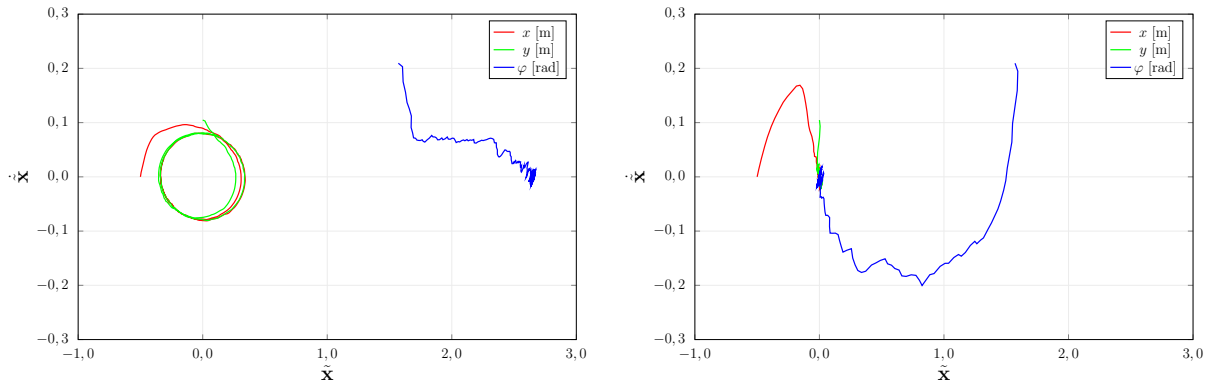
(c) Rastreamento de velocidade.

(d) Esforço de controle.

Fonte: Elaborada pelo autor.

As trajetórias desejadas obtida nos gráficos da Fig. 5.10 são teóricas, uma vez que são havia presença de um sistema externo de verificação. No entanto, filmagens da vista superior mostraram que, visualmente, se assemelham aos resultados obtidos no experimento.

Figura 5.11 – Plano de fase para Robotino® utilizando FBL e FBL com compensação ANN.



(a) Plano de fase FBL.

(b) Plano de fase FBL+ANN.

Fonte: Elaborada pelo autor.

Pode-se observar a partir dos resultados apresentados na Fig. 5.10, que o controlador proposto no capítulo 3 se mostrou mais uma vez satisfatório, uma vez que foi capaz de reduzir consideravelmente o erro associado ao rastreamento de trajetória no experimento com o Robotino®. Outro ponto interessante a se destacar é a notável similaridade dos resultados obtidos no Robotino® SIM (Figs. 5.8 e 5.9) com os do experimental (Figs. 5.10 e 5.11). Com um controlador robusto projetado e validado experimentalmente, podemos avançar nas análises comportamentais do agente inteligente. A aplicação dessa lei de controle em um agente inteligente será mostrado a seguir.

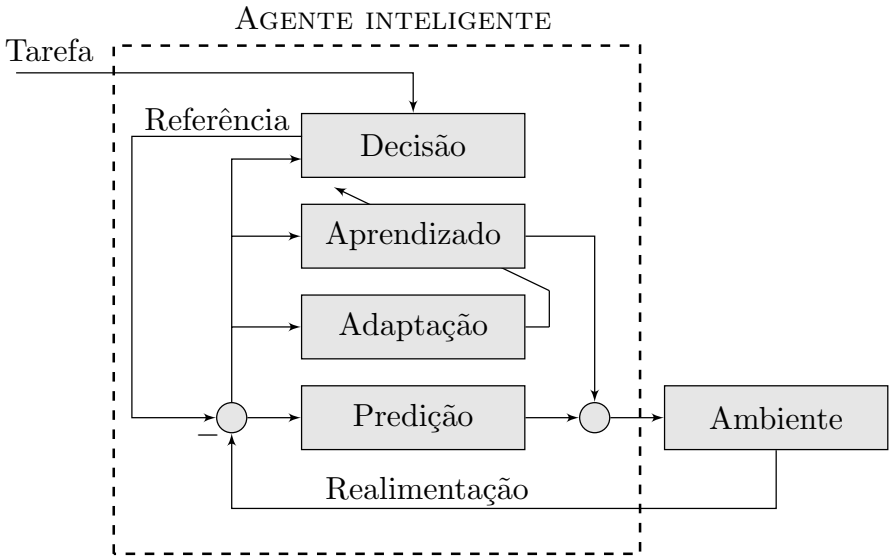
5.6 ϵ-greedy aplicado ao Robotino®

Em complementação ao controlador inteligente, um agente autônomo inteligente deve apresentar uma etapa adicional de planejamento, responsável pela tomada de decisão da referência a ser seguida pelo sistema (CASTELFRANCHI, 1995). Assim, o próprio agente deve ser capaz de definir a referência com base em uma tarefa predefinida, como mostra o diagrama da Fig. 5.12.

O agente inteligente proposto na imagem 5.12 é uma maneira simplificada de representar organismo biológicos (STEVENS, 2008). Um dos objetivos das pesquisas em comportamento animal é avaliar diferentes aspectos comportamentais e habilidades cognitivas, como ansiedade, medo, fuga, ou de que forma ocorre o processo de tomada de decisão. Experimentalmente, uma das metodologias aplicadas ao estudo comportamental da tomada de decisão em ratos (DEACON; RAWLINS, 2006) e peixes (XT, 2003; CERUTTI; LEVIN, 2008) é o labirinto em T .

Os labirintos em T convencionais apresentam dois braços perpendiculares entre

Figura 5.12 – Topologia proposta para um agente inteligente.



Fonte: Autoria de Wallace M. Bessa.

si formando um T. Segundo Deacon e Rawlins (2006), os braços laterais do labirinto, destinados ao objetivo do experimento, podem conter estímulos discriminativos (pistas) em que os animais devem responder para obter uma determinada recompensa, localizada geralmente em outra região do labirinto. No caso de peixes, o labirinto T foi usado para estudar a discriminação de cores (COLWILL et al., 2005), problemas na navegação (RODRIGUEZ et al., 1994), e efeitos de manipulações genéticas na seleção de habitat (CERUTTI; LEVIN, 2008).

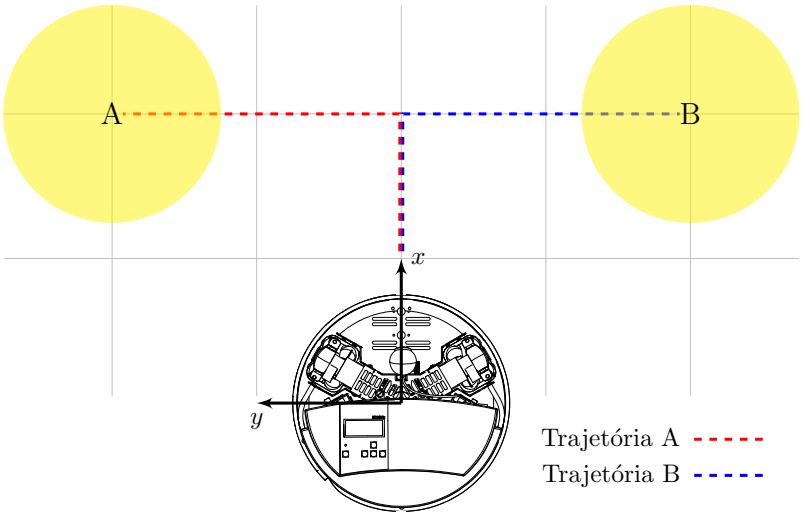
Considerando a aplicabilidade dos labirintos em T para avaliação de comportamento e tomada de decisão em animais, essa metodologia foi escolhida neste trabalho para avaliar o processo de tomada de decisão de um robô omnidirecional. Unindo a lei de controle inteligente proposta no capítulo 3 com o algoritmo de tomada de decisão do capítulo 4, o robô deve ser capaz de distinguir, dentro de um labirinto em T, qual das opções é mais vantajosa.

Em seus experimentos, Colwill et al. (2005) realizaram diferentes testes para discriminação visual em peixes. Os autores utilizaram estímulos colorimétricos, distribuídos em três experimentos, onde todos eles continham uma fase de treinamento em comum, dividida em 16 sessões com 4 tentativas cada, em condições experimentais distintas (cores e direções). Assim, os pesquisadores realizaram, nessa primeira fase, 64 tentativas para dois grupos de 6 indivíduos utilizados no trabalho.

Baseando-se nos valores encontrados na literatura, este trabalho utilizou 20 indivíduos, sendo um Robotino® considerado um indivíduo, submetidos individualmente a uma

sequência de 100 episódios de escolha com duas possíveis ações: escolher o braço A ou B. As ações A e B têm 40% e 60% de chance de retornar uma recompensa positiva respectivamente. Essas probabilidades de retorno foram determinadas com base nos resultados da simulação do algoritmo de tomada de decisão utilizado anteriormente no capítulo 4. Para cada ação A ou B foi delimitada uma trajetória em formato de L, espelhada uma da outra, definida de forma que configurasse um labirinto em T mostrado no esquema da Fig. 5.13. Ao final de cada trajetória foi posicionado um abajur para emitir um foco luz no chão de acordo com as probabilidades de recompensa de cada ação. A obtenção de recompensa foi dada através da detecção da presença de luz pelo robô no ponto de visitação final da trajetória.

Figura 5.13 – Labirinto em T montado no simulador.



Fonte: Elaborada pelo autor.

Como se trata de um formato mais simples, a trajetória total pode ser fracionada em pontos de visitação, formando conjuntamente um caminho em L. Dessa forma, os pontos de visitação das trajetórias desejadas A e B foram definidos conforme a tabela 5.1.

Na Fig. 5.13, a circunferência sombreada em amarelo no final de cada trajetória representa um foco de luz responsável pela emissão da recompensa ao longo dos episódios. Para facilitar o método de emissão das recompensas, cada abajur foi acoplado a um relé SRD-05VDC-SL-C e controlado através de um Arduíno UNO. Além disso, um sensor ultrassônico HC-SR04 foi adicionado ao sistema de recompensas a fim de detectar a passagem do Robotino® e sortear as recompensas de cada ação no momento correto. O modelo esquemático do circuito montado é ilustrado na Fig. 5.14.

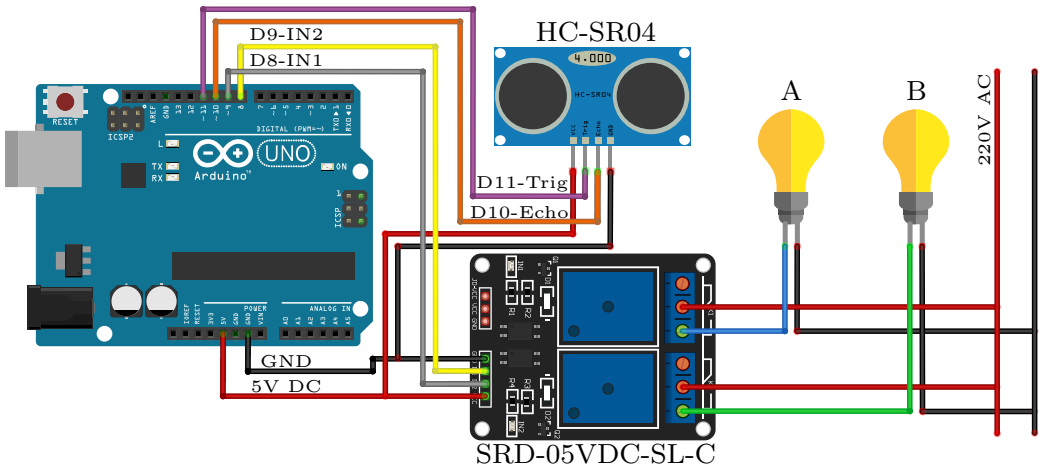
A obtenção da recompensa só foi possível em virtude da adição de um sensor LDR (do inglês, *Light Dependent Resistor* ou Resistor Dependente de Luz) ao Robotino®. A tensão de saída da Interface E/S do Robotino® é de 24V e foi reduzida para 10V utilizando

Tabela 5.1 – Pontos de visitação para as trajetórias em L.

t [s]	Trajetória A			Trajetória B		
	x_d [m]	y_d [m]	φ_d [rad]	x_d [m]	y_d [m]	φ_d [rad]
$t \leq 10$	$0,0$	$0,0$	$0,0$	$0,0$	$0,0$	$0,0$
$10 < t \leq 20$	$0,5$	$0,0$	$0,0$	$0,5$	$0,0$	$0,0$
$20 < t \leq 30$	$0,5$	$0,0$	$\pi/2$	$0,5$	$0,0$	$-\pi/2$
$30 < t \leq 40$	$0,5$	$0,5$	$\pi/2$	$0,5$	$-0,5$	$-\pi/2$
$40 < t \leq 50$	$0,5$	$0,5$	$-\pi/2$	$0,5$	$-0,5$	$\pi/2$
$50 < t \leq 60$	$0,5$	$0,0$	$-\pi/2$	$0,5$	$0,0$	$\pi/2$
$60 < t \leq 70$	$0,5$	$0,0$	$-\pi$	$0,5$	$0,0$	π
$70 < t \leq 80$	$0,0$	$0,0$	$-\pi$	$0,0$	$0,0$	π
$80 < t \leq 90$	$0,0$	$0,0$	$0,0$	$0,0$	$0,0$	$0,0$

Fonte: Elaborada pelo autor.

Figura 5.14 – Circuito das luminárias.

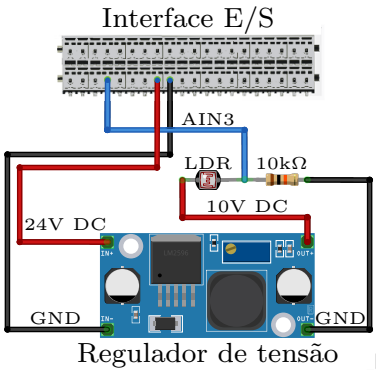


Fonte: Elaborada pelo autor.

um regulador de tensão LM2596 *Step Down* como está representado na Fig. 5.15. Mais detalhes sobre as portas da Interface E/S podem ser encontrados em Weber e Bellenberg (2010).

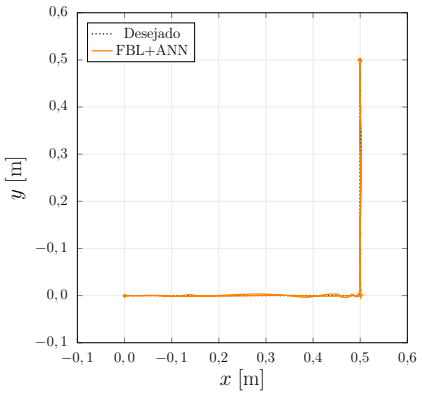
Um teste da trajetória em L foi realizado no Robotino® real a fim de avaliar o comportamento da estratégia controle proposta anteriormente na nova trajetória. Em cada ponto, o controlador teve dez segundos para estabilizar os estados, totalizando 90 segundos para um episódio. Os parâmetros do controlador foram os mesmos utilizados anteriormente. O comportamento do controlador em uma tentativa é mostrada na Fig. 5.16.

Figura 5.15 – Conexão do LDR na interface E/S do Robotino®.

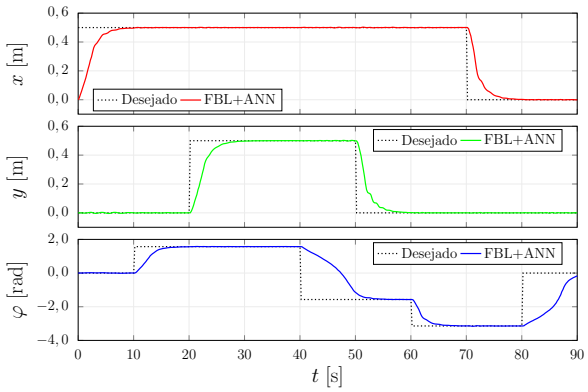


Fonte: Elaborada pelo autor.

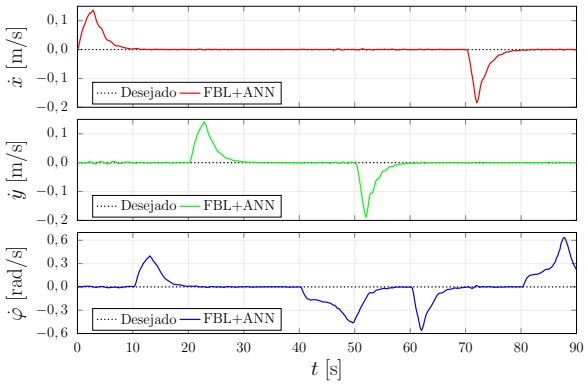
Figura 5.16 – Resultados para o controle do rastreamento da trajetória de 1 indivíduo.



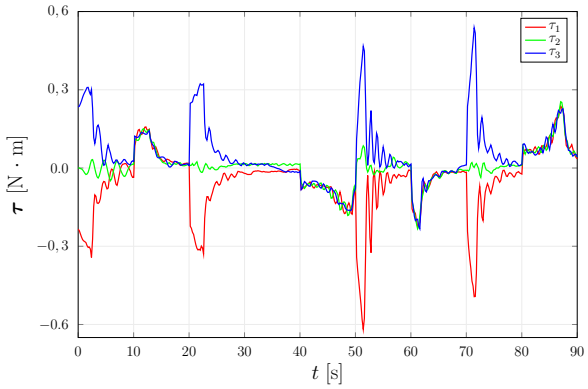
(a) Rastreamento de posição.



(b) Rastreamento de posição.



(c) Rastreamento de velocidade.



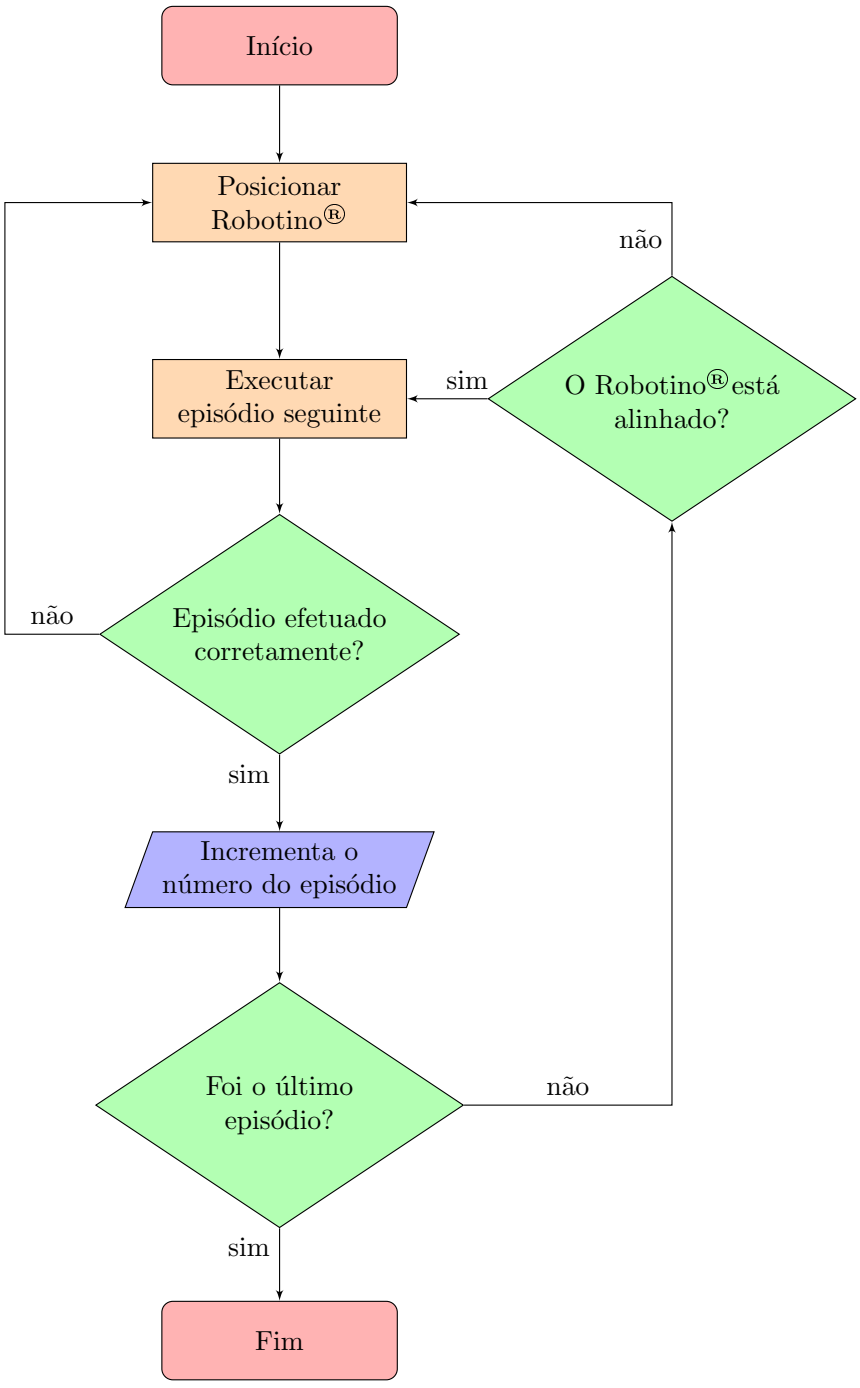
(d) Esforço de controle.

Fonte: Elaborada pelo autor.

A partir da figura, verifica-se que o controlador cumpriu com sucesso sua função de rastreamento de trajetória e velocidade. Feito isso, foi possível realizar o experimento

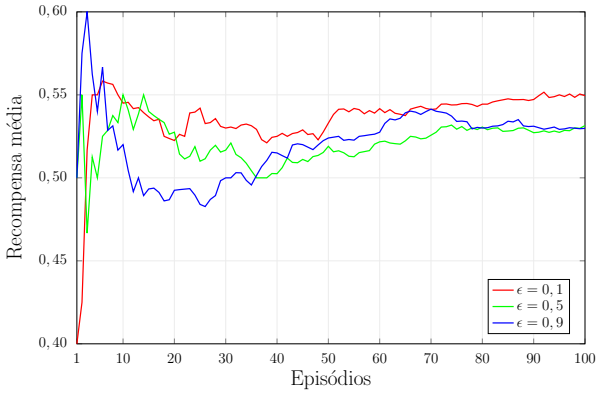
de tomada de decisão com o Robotino®. Os mesmos valores do parâmetro ϵ utilizados no capítulo 4 de $\epsilon = 0.1$, $\epsilon = 0.5$ e $\epsilon = 0.9$ foram utilizados nessa etapa. Como 20 indivíduos foram submetidos à 100 episódios de escolha cada, incluindo 3 valores de ϵ em cada situação, um total de 6000 episódios e 150 horas de análise foram realizados neste experimento. O protocolo experimental pode ser melhor compreendido por meio do fluxograma da figura 5.17 e a média de recompensas obtidas no experimento é mostrada na Fig. 5.18.

Figura 5.17 – Fluxograma do protocolo experimental.



Fonte: Elaborada pelo autor.

Figura 5.18 – Resultados para a tomada de decisão do Robotino®.



Fonte: Elaborada pelo autor.

Foi possível observar que a tomada de decisão feita pelo Robotino® real (Fig. 5.18) produziu um resultado semelhante ao encontrado na simulação, onde quanto mais próximo de zero o valor de ϵ, maior a recompensa média obtida, sendo o $\epsilon = 0.1$ o de maior recompensa média ao longo do experimento. No entanto, a Fig. 5.18 mostra que ao se aproximar dos 100 episódios, os valores de recompensa média para $\epsilon = 0.5$ e $\epsilon = 0.9$ são semelhantes, embora a tendência é que essa diferença aumente com um maior número de episódios, conforme visto na simulação.

6 Considerações Finais

Dada a crescente contribuição que a robótica móvel possui na área de sistemas autônomos, o presente trabalho apresentou o desenvolvimento de um agente autônomo inteligente. Combinou ainda uma estratégia de controle inteligente com o algoritmo de tomada de decisão ϵ-greedy, afim de realizar a tomada de decisão diante de duas opções de escolhas incertas em um labirinto em T. A elaboração do trabalho se baseou nas cinco etapas fundamentais apresentadas no capítulo 1: *i)* modelagem matemática do sistema a ser utilizado; *ii)* o projeto de um controlador inteligente; *iii)* a implementação da estratégia de tomada de decisão; *iv)* simulação computacional integrando o controlador inteligente com o algoritmo de tomada de decisão; e *v)* validação experimental do sistema autônomo com a realização de experimentos no robô móvel Robotino®.

O controlador não linear do tipo FBL foi utilizado para efetuar o rastreamento de trajetória e velocidade do Robotino®. Para compensar as limitações dessa estratégia de controle, um compensador baseado em redes neurais artificiais foi introduzido ao sistema. Foi possível observar a efetividade do controlador projetado nas simulações numéricas, no ambiente de simulação Robotino® SIM e no Robotino®real reduzindo o erro relativo ao rastreamento de trajetória.

Juntamente ao controlador inteligente, o algoritmo de tomada de decisão ϵ-greedy foi utilizado mostrando a capacidade do agente inteligente de lidar com o dilema de explorar ou prospectar um ambiente desconhecido, afim de maximizar sua recompensa média. Foram realizadas simulações numéricas para a tomada de decisão através do ϵ-greedy.

Por fim, a estratégia de controle proposta e o algoritmo de tomada de decisão foram implementados juntamente no simulador do Robotino®, sendo verificada uma maior eficiência no processo de escolha do robô, que foi capaz de perceber no ambiente qual das possíveis escolhas retornava a melhor recompensa.

Considerando-se a importância dos robôs móveis autônomos na área da robótica moderna, bem como sua capacidade de tomar decisões, sugere-se a realização de simulações e experimentos com outras estratégias de controle não lineares, como a estratégia de controle por modos deslizantes (SMC, do inglês *Sliding Modes Control* para observar o efeito do compensador aplicado. Além disso, outros algoritmos de tomada de decisão podem ser implementados para verificar a obtenção de recompensa média comparada com o ϵ-greedy. Por fim, um ambiente com diferentes proporções de recompensa ou, ainda, com mais opções de escolha pode ser construído para verificar a aquisição de recompensa do agente elaborado neste trabalho.

Referências

ANDERSEN, B. S. et al. The proximate architecture for decision-making in fish. *Fish and Fisheries*, v. 17, n. 3, p. 680–695, 2015. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/faf.12139>. Citado 2 vezes nas páginas 1 e 3.

Barreto S., J. C. L. et al. Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. *IEEE/ASME Transactions on Mechatronics*, v. 19, n. 2, p. 467–476, 2014. Citado na página 9.

BESSA, W. M. et al. A Biologically Inspired Framework for the Intelligent Control of Mechatronic Systems and Its Application to a Micro Diving Agent. *Mathematical Problems in Engineering*, v. 2018, 2018. ISSN 15635147. Citado 4 vezes nas páginas 19, 20, 24 e 35.

BESSA, W. M. et al. Design and adaptive depth control of a micro diving agent. *IEEE Robotics and Automation Letters*, v. 2, n. 4, p. 1871–1877, 2017. Citado 3 vezes nas páginas 2, 19 e 37.

BLIESENER, M. et al. Robotino workbook, festo didactic gmbh & co. *KG, Denkendorf, Germany*, 2013. Citado na página 32.

BOUTALIS, Y. S. Neural Network Approaches for Feedback Linearization. *Ceai*, v. 6, n. 1, p. 15–26, 2004. Citado na página 19.

BRINKMANN, G. et al. Reinforcement learning of depth stabilization with a micro diving agent. *Proceedings - IEEE International Conference on Robotics and Automation*, IEEE, V, p. 6197–6203, 2018. ISSN 10504729. Citado na página 3.

BUSH, R. R.; MOSTELLER, F. A Stochastic Model with Applications to Learning Author (s): Robert R . Bush and Frederick Mosteller Source : The Annals of Mathematical Statistics , Vol . 24 , No . 4 (Dec ., 1953), pp . 559-585 Published by : Institute of Mathematical Statistics Sta. *Statistics*, v. 24, n. 4, p. 559–585, 1953. Citado na página 27.

CASTELFRANCHI, C. Guarantees for autonomy in cognitive agent architecture. In: WOOLDRIDGE, M. J.; JENNINGS, N. R. (Ed.). *Intelligent Agents*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. p. 56–70. ISBN 978-3-540-49129-3. Citado na página 40.

CERUTTI, D.; LEVIN, E. Behavioral Neuroscience of Zebrafish. p. 293–310, 2008. Citado 2 vezes nas páginas 40 e 41.

COLWILL, R. M. et al. Visual discrimination learning in zebrafish (Danio rerio). *Behavioural Processes*, v. 70, n. 1, p. 19–31, 2005. ISSN 03766357. Citado na página 41.

DEACON, R. M.; RAWLINS, J. N. P. T-maze alternation in the rodent. *Nature Protocols*, v. 1, n. 1, p. 7–12, 2006. ISSN 17542189. Citado 2 vezes nas páginas 40 e 41.

DUDEK, G.; JENKIN, M. *Computational principles of mobile robotics*. 2ed.. ed. Cambridge University Press, 2010. ISBN 978-0-521-87157-0,978-0-521-69212-0,0521871573,0521692121. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=fcf0f4e452ec55632b18c18863bcb6fa>. Citado 2 vezes nas páginas 2 e 9.

ENQUIST, M.; GIRLANDA, S. *Neural Networks& Animal Behavior*. [S.l.: s.n.], 2005. 300 p. ISBN 964-7445-88-1. Citado na página 1.

ER, M. J.; DENG, C. Obstacle avoidance of a mobile robot using hybrid learning approach. *IEEE Transactions on Industrial Electronics*, v. 52, n. 3, p. 898–905, 2005. Citado na página 3.

FERNANDES, J. M. et al. Feedback linearization with a neural network based compensation scheme. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 7435 LNCS, p. 594–601, 2012. ISSN 03029743. Citado 2 vezes nas páginas 3 e 19.

FUTUYMA, D. J. *Evolution*. Sinauer Associates, 2005. ISBN 9780878931873,0-87893-187-2. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=6A22AE2EFB066D3EB06126154B41DB47>. Citado na página 1.

Gao, X. et al. A hybrid tracking control strategy for nonholonomic wheeled mobile robot incorporating deep reinforcement learning approach. *IEEE Access*, v. 9, p. 15592–15602, 2021. Citado na página 3.

HAYKIN, S. *Redes neurais: princípios e prática*. 2. ed. Porto Alegre: Bookman, 2001. ISBN 978-85-7307-718-6. Citado na página 2.

JAGANNATHAN, S.; COMMURI, S.; LEWIS, F. L. Feedback linearization using CMAC neural networks. *Automatica*, v. 34, n. 5, p. 547–557, 1998. ISSN 00051098. Citado na página 19.

JARDINE, P. T. et al. Adaptive predictive control of a differential drive robot tuned with reinforcement learning. *International Journal of Adaptive Control and Signal Processing*, v. 33, n. 2, p. 410–423, 2019. ISSN 10991115. Citado na página 3.

KAFSI, M. et al. Uncovering latent behaviors in ant colonies. p. 450–458, 2016. Disponível em: <https://epubs.siam.org/doi/abs/10.1137/1.9781611974348.51>. Citado na página 3.

KAGAN, E. et al. *Multi-Robot Systems and Swarming*. John Wiley and Sons, Ltd, 2019. 199-241 p. ISBN 9781119213154. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119213154.ch9>. Citado na página 6.

KIM, D. W.; KIM, J. I.; PARK, Y. L. A Simple Tripod Mobile Robot Using Soft Membrane Vibration Actuators. *IEEE Robotics and Automation Letters*, IEEE, v. 4, n. 3, p. 2289–2295, 2019. ISSN 23773766. Citado na página 3.

KIM, J. I. et al. Learning to Walk a Tripod Mobile Robot Using Nonlinear Soft Vibration Actuators with Entropy Adaptive Reinforcement Learning. *IEEE Robotics and Automation Letters*, Institute of Electrical and Electronics Engineers Inc., v. 5, n. 2, p. 2317–2324, apr 2020. ISSN 23773766. Citado na página 3.

KIM, S.; LASCHI, C.; TRIMMER, B. Soft robotics: A bioinspired evolution in robotics. *Trends in Biotechnology*, Elsevier Ltd, v. 31, n. 5, p. 287–294, 2013. ISSN 01677799. Disponível em: <http://dx.doi.org/10.1016/j.tibtech.2013.03.002>. Citado na página 27.

KULESHOV, V.; PRECUP, D. Algorithms for multi-armed bandit problems. v. 1, p. 1–32, 2014. Disponível em: <http://arxiv.org/abs/1402.6028>. Citado na página 28.

MURPHY, R. *An Introduction to AI Robotics*. MIT Press, 2000. (Intelligent robotics and autonomous agents). ISBN 0262133830,9780262133838. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=0f594d22e32a6db0c795f6b8b9f0f074>. Citado na página 5.

OGATA, K. *Engenharia de Controle Moderno*. [S.l.: s.n.], 2010. 912 p. ISBN 9788576058106. Citado na página 14.

RAJ, L.; CZMERK, A. Modelling and simulation of the drivetrain of an omnidirectional mobile robot. *Automatika*, v. 58, p. 232–243, 11 2017. Citado 2 vezes nas páginas 9 e 53.

RESHAMWALA, A.; P, D. Robot Path Planning using An Ant Colony Optimization Approach:A Survey. *International Journal of Advanced Research in Artificial Intelligence*, v. 2, n. 3, p. 65–71, 2013. ISSN 21654050. Citado na página 3.

RIESKAMP, J.; OTTO, P. E. SSL: A theory of how people learn to select strategies. *Journal of Experimental Psychology: General*, v. 135, n. 2, p. 207–236, 2006. ISSN 00963445. Citado na página 1.

ROBBINS, H. Some Aspects of The Sequential Design of Experiments. *Bulletin of the American Mathematical Society*, v. 58, n. 5, p. 527–535, 1952. Citado na página 27.

RODRIGUEZ, F. et al. Performance of goldfish trained in allocentric and egocentric maze procedures suggests the presence of a cognitive mapping system in fishes. *Animal Learning & Behavior*, v. 22, n. 4, p. 409–420, 1994. ISSN 00904996. Citado na página 41.

ROMERO, R. A. F. et al. *Robótica Móvel*. 1ed.. ed. [S.l.]: LTC - GRUPO GEN, 2014. ISBN 8521623038. Citado 2 vezes nas páginas 8 e 9.

RUSSO, D. J. et al. A Tutorial on Thompson Sampling Boston. *Foundations and Trends R in Machine Learning*, v. 11, n. 1, p. 1–96, 2018. Disponível em: <https://web.stanford.edu/{~}bvr/pubs/TS{_}Tutoria>. Citado na página 27.

RYER, C. H.; OLLA, B. L. Influences of food distribution on fish foraging behaviour. *Animal Behaviour*, v. 49, n. 2, p. 411–418, 1995. Citado na página 1.

SANTOS, J. D. B.; BESSA, W. M. Intelligent control for accurate position tracking of electrohydraulic actuators. *Electronics Letters*, v. 55, n. 2, p. 78–80, 2019. ISSN 00135194. Citado 2 vezes nas páginas 3 e 19.

SEELEY, T. D.; BUHRMAN, S. C. Group decision making in swarms of honey bees. *Behavioral Ecology and Sociobiology*, v. 45, n. 1, p. 19–31, 1999. ISSN 03405443. Citado na página 3.

SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. *Introduction to Autonomous Mobile Robots*. 2nd. ed. The MIT Press, 2011. (Intelligent Robotics and Autonomous Agents series). ISBN 0262015358,9780262015356. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=827c6c13d75863c8a048672b43471db6>. Citado na página 7.

SKINNER, B. F. *The Behavior of Organisms*. [s.n.], 1938. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=0271da2faa6a920f55c96f9095789998>. Citado na página 1.

SLOTINE, J.-J. E.; LI, W. *Applied nonlinear control*. 1. ed. [S.l.]: Prentice-Hall, 1991. ISBN 0-13-040890-5. Citado 4 vezes nas páginas 14, 15, 19 e 21.

STEVENS, J. R. The Evolutionary Biology of Decision Making. *Faculty Publications, Department of Psychology.*, v. 523, 2008. Disponível em: <https://digitalcommons.unl.edu/psychfacpub/523>. Citado 2 vezes nas páginas 2 e 40.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning An Introduction second edition*. 2. ed. London: MIT Press, 2018. Citado 4 vezes nas páginas 2, 27, 28 e 30.

TANAKA, M. C.; FERNANDES, J. M.; BESSA, W. M. Feedback linearization with fuzzy compensation for uncertain nonlinear systems. *International Journal of Computers, Communications and Control*, v. 8, n. 5, p. 736–743, 2013. ISSN 18419844. Citado na página 19.

THOMPSON, W. R. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, v. 25, n. 3/4, p. 285, 1933. ISSN 00063444. Citado na página 28.

THORNDIKE, E. L. Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, v. 2, n. 4, p. i–109, 1898. ISSN 0096-9753. Citado na página 2.

TINBERGEN, N. Foundations of Animal Behavior- On aims and methods of Ethology. *Z. Tierpsychologie*, v. 20, n. March, p. 410–433, 1963. Citado na página 1.

WEBER, R.; BELLENBERG, M. Robotino manual, festo didactic gmbh & co. *KG, Denkendorf, Germany*, 2010. Citado 4 vezes nas páginas 10, 33, 34 e 43.

WILSON, R. C. et al. Humans use directed and random exploration to solve the explore-exploit dilemma. *Journal of Experimental Psychology: General*, v. 143, n. 6, p. 2074–2081, 2014. ISSN 00963445. Citado na página 28.

XT, H. E. T-Maze for Zebrafish. p. 473300, 2003. Citado na página 40.

Anexos

ANEXO A – Descrição das matrizes

As equações dos elementos das matrizes $\mathbf{T}_r$ e $\mathbf{M}$ introduzidas no capítulo 2 foram retiradas do material suplementar presente no do trabalho realizado por Raj e Czmerk (2017), e são descritas de acordo com este anexo.

Matriz M

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{A.1}$$

$$m_{11} = \frac{r^2 \sec^4\left(\frac{\beta}{2}\right) m_r \left(2l^2 \csc^2\left(\frac{\beta}{2}\right) + R_r^2\right)}{32l^2} \tag{A.2}$$

$$m_{12} = \frac{r^2 \sec^4\left(\frac{\beta}{2}\right) m_r \left(\cos\left(\beta\right) R_r^2 - 2l^2\right)}{16l^2} \tag{A.3}$$

$$m_{13} = \frac{r^2 \sec^4\left(\frac{\beta}{2}\right) m_r \left(\frac{4\cos\left(\beta\right) l^2}{\cos\left(\beta\right) - 1} + R_r^2\right)}{32l^2} \tag{A.4}$$

$$m_{21} = m_{12} \tag{A.5}$$

$$m_{22} = \frac{r^2 \sec^4\left(\frac{\beta}{2}\right) m_r \left(2l^2 + \cos\left(\beta\right) R_r^2\right)}{8l^2} \tag{A.6}$$

$$m_{23} = m_{12} \tag{A.7}$$

$$m_{31} = m_{13} \tag{A.8}$$

$$m_{32} = m_{12} \tag{A.9}$$

$$m_{33} = m_{11} \tag{A.10}$$

$$\tag{A.11}$$

Matriz $\mathbf{T}_r$

$$\mathbf{T}_r = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \tag{A.12}$$

$$t_{11} = \frac{r(\sin(\beta - \varphi(t)) + \sin(\varphi(t)))}{\cos(2\beta) - 1} \tag{A.13}$$

$$t_{12} = \frac{r\sin(\varphi(t))}{\cos(\beta) + 1} \tag{A.14}$$

$$t_{13} = \frac{r(\sin(\varphi(t)) - \sin(\beta + \varphi(t)))}{\cos(2\beta) - 1} \tag{A.15}$$

$$t_{21} = \frac{r(\cos(\beta - \varphi(t)) - \cos(\varphi(t)))}{\cos(2\beta) - 1} \tag{A.16}$$

$$t_{22} = \frac{r\cos(\varphi(t))}{\cos(\beta) + 1} \tag{A.17}$$

$$t_{23} = \frac{r(\cos(\beta + \varphi(t)) - \cos(\varphi(t)))}{\cos(2\beta) - 1} \tag{A.18}$$

$$t_{31} = \frac{r}{2l(cos(\beta) + 1)} \tag{A.19}$$

$$t_{32} = \frac{r\cos(\beta)}{l(\cos(\beta) + 1)} \tag{A.20}$$

$$t_{33} = t_{31} \tag{A.21}$$