



LUAN VIEIRA MENDES

**CONTROLE FUZZY TOLERANTE A FALHAS PARA ROBÔS
MÓVEIS**

LAVRAS – MG

2019

LUAN VIEIRA MENDES

CONTROLE FUZZY TOLERANTE A FALHAS PARA ROBÔS MÓVEIS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Daniel Augusto Pereira
Orientador

LAVRAS – MG

2019

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Mendes, Luan Vieira

Controle Fuzzy tolerante a falhas para robôs móveis / . –
Lavras : UFLA, 2019.
55 p. : il.

Dissertação (mestrado acadêmico)–Universidade Federal
de Lavras, 2019.

Orientador: Prof. DSc. Daniel Augusto Pereira.

Bibliografia.

1. Controle tolerante a falhas. 2. Detecção de falhas. 3.
Robôs móveis. I. Pereira, Daniel Augusto. II Título.

LUAN VIEIRA MENDES

CONTROLE FUZZY TOLERANTE A FALHAS PARA ROBÔS MÓVEIS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 13 de Setembro de 2019.

Prof. DSc. Daniel Augusto Pereira	UFLA
Prof. DSc. Bruno Henrique Groenner Barbosa	UFLA
Prof. DSc. Eduardo Bento Pereira	UFSJ

Prof. DSc. Daniel Augusto Pereira
Orientador

**LAVRAS – MG
2019**

RESUMO

O problema de detecção de falhas em atuadores de robôs móveis autônomos (RMAs) com aplicação em controle de trajetória foi abordado nessa dissertação. Para essa aplicação, um robô móvel com duas rodas tracionadas e uma roda livre com cinemática não linear é demandado a seguir uma trajetória predefinida em um determinado tempo. Um ambiente sem obstáculos é considerado na execução dessa tarefa. O objetivo deve ser completado mesmo após falhas surgirem nos motores do robô. A proposta apresentada aqui emprega um Controlador Lógico Fuzzy para o controle de trajetória. Para a detecção de falhas em atuadores do RMA, os estados dos motores são estimados por um Filtro de Kalman, permitindo que um parâmetro denominado perda de efetividade (PDE) seja indiretamente calculado. Esse PDE estimado é utilizado para reconfigurar o sinal de controle gerado pelo Controlador Fuzzy, garantindo um Controle Tolerante a Falhas. Na necessidade de uma ação conjunta entre dois ou mais robôs móveis com a finalidade de completar uma trajetória, um algoritmo que aborda a ocorrência de falhas severas (aquelas que impossibilitam um funcionamento adequado) foi desenvolvido. Essa solução proposta possibilita o cumprimento da trajetória pelos robôs sem falhas solucionando uma parcela do Controle Cooperativo Tolerante a Falhas. As validações das propostas aqui apresentada foram feitas mediante simulações. Como resultado principal, o Controle de Trajetória Tolerante a Falhas utilizando um Controlador Fuzzy e um reconfigurador de sinal de controle obteve um erro de aproximadamente 0,3% do tempo alvo da trajetória.

Palavras-chave: Detecção de falhas, Controle tolerante a falhas, Robôs Móveis Autônomos, Controle cooperativo tolerante a falhas, Perda de efetividade

ABSTRACT

The problem of fault detection caused by actuators with application to trajectory tracking of wheeled mobile robots (WMRs) is addressed in this thesis. For the considered application, a wheeled mobile robot -having nonlinear kinematics- is required to follow a predefined trajectory in a stipulated time. An environment free of obstacles is used for this task. The goal has to be accomplished despite the presence of faults that may occur in the actuators of the robot. The proposal presented here uses a Fuzzy Logical Controller to the trajectory tracking problem. For the fault detection problem in the actuators of the WMRs, the states and a loss of effectiveness (LOE) parameter are estimated by a Kalman Filter. The calculated LOE value is used to reconfigure the control signal generated by the Fuzzy Controller guaranteeing the Fault-Tolerant Control (FTC). In the need of a cooperative work for two or more robots with the goal to conclude the trajectory, an algorithm which consider severe faults was developed. The proposed solution grants the aim to be completed by the fault-free robots, leaving those with severes faults behind. Simulations were made in order to validate the proposed solutions. As a main result of the Fault-Tolerant Control, an error of aproximadetely 0,3% of the trajectory goal time was reached.

Keywords: Fault detection, fault-tolerante control, Wheeled mobile robots, fault-tolerant cooperative control

LISTA DE FIGURAS

Figura 2.1 – Classificação dos robôs móveis	13
Figura 2.2 – Classificação dos robôs móveis	14
Figura 2.3 – Esquemático motor CC	15
Figura 2.4 – Robô móvel de duas rodas tracionadas	16
Figura 2.5 – Princípio de realimentação unitária	18
Figura 2.6 – Diagrama de blocos de realimentação do motor CC	19
Figura 2.7 – Configuração do Controlador Lógico Fuzzy	21
Figura 2.8 – Representação dos erros de posição e orientação	22
Figura 2.9 – Funções de pertinência do CLF.	23
Figura 2.10 – Esquemático do Controlador Fuzzy desenvolvido	24
Figura 2.11 – Esquemático resumido do trabalho	25
Figura 4.1 – Diagrama de blocos com o estágio de reconfiguração	31
Figura 4.2 – Diagrama de blocos final do CCTF com três robôs	32
Figura 5.1 – Resposta ao degrau do controlador PID	34
Figura 5.2 – Controle de trajetória sem a ocorrência de falhas	35
Figura 5.3 – Rotação angular dos motores do robô	36
Figura 5.4 – Superfície do CLF	36
Figura 5.5 – Filtro de Kalman no motor esquerdo	38
Figura 5.6 – Filtro de Kalman no motor direito	38
Figura 5.7 – Efetividade dos motores.	39
Figura 5.8 – Trajetória do robô perante a ocorrência de uma falha parcial	40
Figura 5.9 – Trajetória completa do robô perante uma falha parcial.	41
Figura 5.10 – Erros de distância com e sem o reconfigurador.	42
Figura 5.11 – Efetividade estimada do motor esquerdo do robô 1 e a falha simulada	43
Figura 5.12 – Falha parcial de 50% no motor esquerdo do robô 1	44
Figura 5.13 – Falha severa no motor direito do segundo robô	45
Figura 5.14 – Efetividade estimada do motor direito do robô 2	46
Figura 5.15 – Velocidade linear do robô 2	47

LISTA DE TABELAS

Tabela 2.1 – Valores dos parâmetros para cálculos dos ganhos	20
Tabela 2.2 – Conjunto de regras Fuzzy	24
Tabela 3.1 – Taxas de falhas típicas de elementos mecânicos e eletromecânicos	27
Tabela 4.1 – Parâmetros de simulação	32

SUMÁRIO

Lista de Figuras	4
Lista de Tabelas	5
1 Introdução	7
1.1 Motivação e justificativa da pesquisa	7
1.2 Descrição do problema	9
1.3 Organização do trabalho	10
2 Controle de trajetória para robôs móveis	11
2.1 Modelagem do robô móvel	12
2.1.1 Motores de corrente contínua	12
2.1.2 Modelo cinemático	15
2.2 Controle de trajetória	17
2.2.1 Controlador PID	18
2.2.2 Controlador Fuzzy	20
3 Detecção de falhas em atuadores	26
3.1 Filtro de Kalman	27
4 Controle de trajetória tolerante a falhas	30
4.1 Reconfiguração das velocidades de referência	30
4.2 Controle cooperativo para múltiplos robôs móveis	31
5 Resultados simulados e discussões	33
5.1 Controle de trajetória	33
5.1.1 Controlador PID	33
5.1.2 Controlador Fuzzy	33
5.2 Detecção de falhas em atuadores	37
5.3 Controle de trajetória tolerante a falhas	39
5.3.1 Reconfiguração do controle	39
5.3.2 Controle cooperativo	41
6 Conclusões	48
REFERÊNCIAS	50
A Controle Tolerante a Falhas	54
B Controle Cooperativo Tolerante a Falhas	55

1 INTRODUÇÃO

1.1 Motivação e justificativa da pesquisa

Nos últimos anos, o desenvolvimento de robôs móveis autônomos não tripulados tem atraído a atenção da comunidade científica, militar e industrial. Uma das tarefas desafiadoras no desenvolvimento desses sistemas robóticos é a interação com o ambiente e com outros robôs.

Robô Móvel Autônomo (RMA) é um tipo de sistema robótico complexo com vários subsistemas capazes de obter informações de si próprio e do ambiente através de sensores, movimentar-se em um ambiente com obstáculos e realizar tarefas específicas (ZIXING; HANGEN; HONG, 2002). RMAs têm sido utilizados para diversas aplicações como: operações militares, vigilância, segurança, monitoramento ambiental e exploração em minas e planetária. Durante essas missões, um RMA trabalha cooperativamente com outros RMAs em um ambiente desconhecido onde a intervenção humana é cara, devagar, perigosa, ou impossível (VERMA et al., 2004). Diante dessa situação, é essencial que os robôs ajam independentemente para detectar e isolar as possíveis falhas internas e cooperativamente se readaptarem de forma que possam cumprir com o objetivo mesmo com as limitações causadas por essas falhas.

Por exemplo, quando robôs móveis são usados para explorações planetárias em missões espaciais, eles devem ser capazes de operarem por um grande período sem intervenção de uma central de controle na terra. Isso significa que com a presença de falhas, os sistemas devem se adaptar para trabalhar com limitações.

Falhas são comportamentos anômalos presentes em componentes de um sistema causadas ou não por uma ação externa. Em robótica móvel, a detecção de falhas é usada para prevenir comportamentos inesperados de seu sistema. Falhas em componentes de subsistemas robóticos podem ser classificadas como: total, parcial e sem falhas (DING, 2008). A ocorrência de uma falha total remete à perda de um componente cuja a substituição do mesmo é a única solução. Falhas parciais são aquelas que permitem a reconfiguração do sistema e simulam um funcionamento normal.

O diagnóstico de falhas tem sido pesquisado em diferentes subsistemas de RMAs, tais como motores (ZANARDELLI et al., 2005), engrenagens (ZHENG; LI; CHEN, 2002), rodas (Roumeliotis; Sukhatme; Bekey, 1998), suspensões (Luo et al., 2005), sensores (CARLSON; MURPHY, 2003), etc.

Cada RMA deve possuir uma unidade de detecção de falhas (UDF) com a tarefa de detectar comportamentos anômalos devido a falha de componentes, determinar o tipo e a localização da falha e identificar a gravidade da falha (CHAMSEDDINE; AMOOZGAR; ZHANG, 2015). Com a informação da ocorrência de falhas, o robô é reconfigurado para atuar com a presença da anomalia dos componentes. Essa reconfiguração é feita por Controladores Tolerante a Falhas (CTF). Esses CTFs são capazes de se ajustarem de forma a simular um funcionamento normal do sistema quando possível (KAMEL; YU; ZHANG, 2017). Inúmeras técnicas de UDF e CTF têm sido estudadas nas últimas décadas (ROTONDO, 2018).

Em Roumeliotis, Sukhatme e Bekey (1998), um filtro de Kalman adaptativo foi projetado para detecção de falhas e implementado em um RMA. Um algoritmo de predição de falhas baseado em erro foi desenvolvido em Dixon, Walker e Dawson (2001). Este algoritmo pode ser usado mesmo com a imprecisão do modelo dinâmico do robô. Em Ji e Sarkar (2007), um conjunto de controladores foi desenvolvido para lidar com hipóteses de falhas em diferentes sensores. Em Skoundrianos e Tzafestas (2004) e Goel et al. (2000), redes neurais são utilizadas para detecção de falhas em sensores e atuadores. No trabalho de Shabaniyan e Montazeri (2011), uma rede neuro-fuzzy foi utilizada a detecção de falhas em sistemas dinâmicos.

Em missões que múltiplos RMAs são necessários, perante a ineficiência de um CTF para reconfigurar o sistema, é preciso que um controlador cooperativo tolerante a falhas (CCTF) seja usado para que os outros robôs, sem a presença de falha, se adaptem ao robô defeituoso. O objetivo do CCTF é garantir que o time de robôs possa reconfigurar para cumprir com suas devidas tarefas mesmo perante a uma falha interna de um indivíduo. Caso seja detectado por um robô uma falha severa e diagnosticada impossibilidade de completar o objetivo, os outros robôs do grupo devem se organizar e seguir sem o robô impossibilitado. De outra forma, se a falha interna detectada for parcial e cabível de recombinação, os controladores dos outros robôs devem ser reconfigurados (KAMEL; YU; ZHANG, 2017).

Apesar de estudos com RMAs utilizando somente CCTF serem recentes, os autores que abordam o assunto focam principalmente em falhas de comunicação que acontecem quando ocorrem falhas de transmissão de dados entre os robôs (IZADI; GORDON; ZHANG, 2013; YANG; JIANG; ZHANG, 2014), problemas de desvio de obstáculos com robôs com comportamento anormal em seus componentes (SABER; MURRAY, 2003) e situações com falhas severas nos atuadores em que o robô é impossibilitado de continuar a trajetória (THUMATI; DIERKS; SARANGAPANI, 2012). Nos casos em que o robô não consegue seguir o caminho,

Mead, Long e Weinberg (2009) sugerem estratégias de comportamento e alinhamentos para os outros RMAs completarem o objetivo. Em Chamseddine, Zhang e Rabbath (2012), veículos aéreos autônomos (VAA) são usados e um algoritmo de formação é proposto baseado em técnicas de replanejamento de trajetórias. Em (THUMATI; DIERKS; SARANGAPANI, 2012), falhas na previsão de sistemas dinâmicos de RMAs são tratadas utilizando recombinação com redes neurais. O trabalho de Xu et al. (2014) sugere um CTF fuzzy adaptativo para um time de veículos aéreos autônomos (VAAs) com falhas totais em atuadores.

Estudos que abrangem times de RMAs com unidades de UDF e implementam CTFs individuais e CCTFs para o conjunto ainda são poucos. Em Kamel, Yu e Zhang (2017), os autores implementam um filtro de Kalman de dois estágios para UDF, propõem um CTF para lidar com falhas individuais plausíveis de ajustes e desenvolvem um CCTF para ocorrência de falhas severas. No CCTF, estratégias de formação e recombinação do time são propostas e implementadas.

1.2 Descrição do problema

Os problemas de detecção de falhas em motores de corrente contínua (CC) dos robôs móveis, controle de trajetória tolerante a falhas e controle cooperativo tolerante a falhas para múltiplos robôs móveis são abordados nesta dissertação.

Alguns dos problemas clássicos da robótica móvel é o controle de trajetória. Solucionar esse problema é o primeiro passo para a automatização de um robô móvel. Assim, é garantido que ele possa se locomover por um ambiente com uma trajetória predefinida sem se colidir com um objeto. Outro problema comum é a falha ou degradação de desempenho de componentes elétricos e mecânicos dos RMAs afetando a sua mobilidade (KLANCAR et al., 2017). Por isso, é necessário que seja implementada uma unidade que detecta comportamentos anômalos nos subsistemas robóticos e um controle que permite que o sistema se adapte para simular um funcionamento normal quando possível. Outro problema abordado nessa dissertação é a cooperação de dois ou mais RMAs na ocorrência de uma falha, podendo ela ser severa ou não.

O objetivo geral deste trabalho é projetar um controlador de trajetória de um robô móvel tolerante a falhas em motores de corrente contínua. Além dessa proposta, um algoritmo para o controle cooperativo de múltiplos robôs móveis é utilizado para problemas com a ocorrência de falhas severas nos motores.

As soluções propostas nesse trabalho permitem que múltiplos robôs móveis possam seguir uma trajetória considerando falhas lineares em pelo menos um de seus atuadores.

1.3 Organização do trabalho

Esse trabalho está organizado da seguinte forma:

No capítulo 2 é tratado o controle de trajetória de robôs móveis. O tipo de robô utilizado e os atuadores são apresentados e o modelo cinemático para o robô é detalhado. Dois tipos de controladores são apresentados e implementados para o controle de trajetória. São eles: Fuzzy e PID. Os controladores são complementares e dependentes um do outro. O PID assegura as velocidades de referência do robô para o controle de trajetória feito pelo Fuzzy. Eles são testados via *software* e os resultados simulados obtidos entre os dois controladores são apresentados e discutidos.

No capítulo 3 é apresentada o modelo de uma unidade de detecção de falhas e discute as possíveis ocorrências nos componentes de um RMA. Para a detecção de falhas e estimação de estados, um Filtro de Kalman é usado para estimar a eficiência dos atuadores do robô e utilizá-la como parâmetro para ajuste da velocidade dos motores.

O capítulo 4 mostra o método utilizado para a reconfiguração dos valores de referência de velocidade necessária para que o robô consiga cumprir a trajetória mesmo perante a uma falha parcial. O controle cooperativo para múltiplos robôs móveis também é mostrado e indicam as condições para a cooperatividade entre os robôs.

No capítulo 5, os resultados de detecção de falhas em atuadores usando Filtro de Kalman, controle de trajetória com controlador Fuzzy, controle de trajetória perante falhas parciais e controle cooperativo de múltiplos robôs móveis são apresentados e discutidos.

O capítulo 6 inclui um resumo dos principais tópicos, contribuições apresentados nessa dissertação e a conclusão do mesmo. Além disso, sugestões e considerações para trabalhos futuros são expostos e discutidos nesse capítulo.

2 CONTROLE DE TRAJETÓRIA PARA ROBÔS MÓVEIS

Robôs Móveis Autônomos são capazes de se deslocar em um ambiente desconhecido partindo de um ponto para outro qualquer seguindo uma trajetória. Para isso, é necessário um estudo da cinemática do sistema robótico em questão que correlaciona a sua posição e velocidade. Com a relação definida, um controlador é projetado para controlar a velocidade de cada uma das rodas a fim de obter a posição e orientação desejadas (SIEGWART; NOURBAKHS, 2004).

Os robôs móveis são classificados considerando a quantidade e o tipo de rodas presente no sistema, a localização e o modelo de atuadores e o tipo da estrutura do corpo. As Figuras 2.1 e 2.2 mostram as configurações existentes para robôs móveis.

Existem dois modelos de robôs móveis com duas rodas. O primeiro tem a localização de suas rodas lineares com uma roda tracionada na parte traseira e uma livre na parte dianteira. Como exemplo, as motocicletas. O segundo modelo contém ambas as rodas diferenciáveis tracionadas e localizadas paralelamente. Já com três rodas, há 5 modelos diferentes. O primeiro, tem sua geometria circular com duas rodas centradas tracionadas paralelamente e um ponto de contato na dianteira. O segundo tem sua geometria retangular e duas de suas rodas estão localizadas na parte traseira sendo ambas tracionadas diferenciais e uma roda livre na parte dianteira. O terceiro robô também tem sua geometria retangular e duas rodas tracionadas conectadas na parte traseira e uma direcional na dianteira. O quarto modelo é de um robô circular com suas rodas tracionadas e arranjadas em 120 graus. O quinto modelo tem sua geometria triangular com suas rodas tracionadas e com direção.

Para robôs com 4 rodas, existem 7 modelos disponíveis. O primeiro é conhecido como carro com tração traseira. Suas duas rodas traseiras são tracionadas e conectadas e as dianteiras são conectadas e com direção limitada. O segundo modelo é o carro com a tração traseira, onde as suas rodas tracionadas e conectadas possuem direção na parte dianteira e na parte traseira as duas rodas não tem direção, não são tracionadas e nem conectadas. O terceiro modelo é o carro com tração em 4 rodas onde suas rodas são tracionáveis e direcionáveis na parte traseira e dianteira. O quarto modelo tem duas rodas traseiras tracionáveis diferenciáveis e duas rodas dianteiras omnidirecionais. O quinto modelo tem suas quatro rodas omnidirecionais e não conectadas entre si. O sexto modelo é o único com geometria circular apresentada aqui. O mesmo tem duas rodas tracionadas em suas laterais e dois pontos de contatos localizados uma na parte de trás e uma na parte da frente. O sétimo e último modelo apresentado tem suas quatro rodas

do tipo castor tracionadas. Há modelos de 5 rodas ou mais que não serão apresentados aqui devido a sua pouca utilização para pesquisa em robôs móveis.

Neste capítulo, o modelo matemático e o estudo cinemático do robô serão apresentados e discutidos. Além disso, para o controle de trajetória, controladores PID e Fuzzy são utilizados e explicados.

2.1 Modelagem do robô móvel

O modelo de um sistema mecatrônico descreve o seu comportamento perante uma ação externa. Na robótica móvel, o modelo de um motor permite equacionar a velocidade angular, torque e posicionamento do eixo a partir da tensão ou corrente aplicada (ORIOLO; LUCA; VENDITTELLI, 2002). Já o modelo cinemático de um robô móvel relaciona a sua posição e orientação com a velocidade de cada atuador considerando sua geometria.

Nesta seção, os modelos dos atuadores utilizados e da cinemática do robô serão matematicamente apresentados assim como os controladores responsáveis pelo controle de trajetória do robô.

2.1.1 Motores de corrente contínua


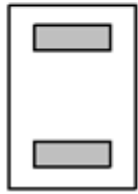
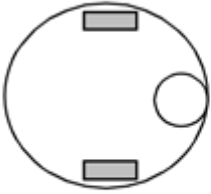
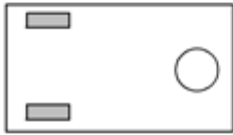

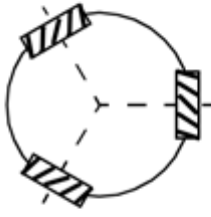
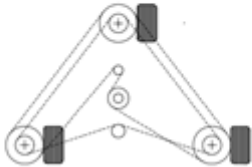
Um dos componentes mais utilizados como um atuador em controle mecânico é o motor de corrente contínua. O esquemático elétrico e mecânico de um motor CC pode ser visto na Figura 2.3. No esquema, o circuito com a fonte de alimentação com a tensão v aplicada, resistência R , indutância L , força contra eletromotriz e e a direção da corrente elétrica i são mostrados. Além desses, um campo magnético fixo, o rotor com os parâmetros de amortecimento b e o momento de inércia J , o torque T , a posição do rotor θ e sua variação $\dot{\theta}$ estão representados na imagem.

De acordo com a lei de tensões de Kirchhoff, a equação elétrica do motor é descrita como:

$$L \frac{di(t)}{dt} + Ri(t) + K\dot{\theta}(t) = v(t), \quad (2.1)$$

sendo t o instante de tempo. O torque produzido no rotor é proporcional à corrente, sendo representado por:

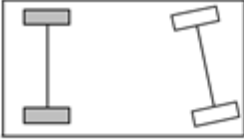
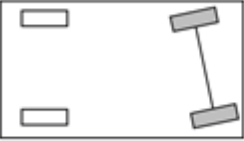
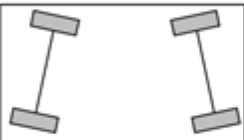

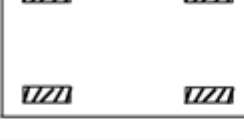

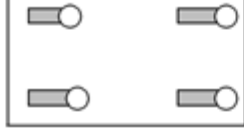
Figura 2.1 – Classificação dos robôs móveis

Nº Rodas	Localização	Descrição	Exemplos
2		Uma roda tracionada na parte traseira e outra não tracionada	Bicicleta, Motocicleta
		Duas rodas tracionadas diferenciáveis	Robô Cye
3		Duas rodas centrais diferenciáveis com um ponto de contato	Robô smartRob
		Duas rodas independentes tracionadas na parte traseira e uma omnidirecional na parte da frente	Vários robôs indoors
		Duas rodas tracionadas conectadas na parte traseira e uma direcional na dianteira	Mini caminhões Piaggio
		Três rodas tracionadas arranjadas a 120°	Robô Palm Pilot (CMU), Stanford Robô Tribolo
		Três rodas sincronizadas tracionadas com direção	Robô Denning MRV-2, I-Robot B24, Nomad 200

Fonte: (SIEGWART; NOURBAKHS, 2004) (Adaptado)

$$T(t) = K_i i(t) \quad (2.2)$$

Figura 2.2 – Classificação dos robôs móveis

Nº Rodas	Localização	Descrição	Exemplos
4		Duas rodas tracionadas conectadas na parte traseira e duas rodas conectadas com direção na parte da frente	Carro com tração traseira
		Duas rodas conectadas entre si não tracionadas na parte traseira e duas rodas conectadas entre si tracionadas com direção na parte da frente	Carro com tração dianteira
		Duas rodas conectadas entre si direcionáveis na parte traseira e dianteira	Carro com tração 4 rodas
		Duas rodas tracionadas diferenciáveis na parte traseira e duas rodas omnidirecionais na parte dianteira	Robô Charlie (DMT-EPFL)
		4 Rodas omnidirecionais não conectadas entre si	Carnegie Mellon Uranus
		Duas rodas tracionadas centradas e dois pontos de contato	EPFL Khepera, Hyperbot Chip
		Quatro rodas tracionadas do tipo castor	Nomad XR4000

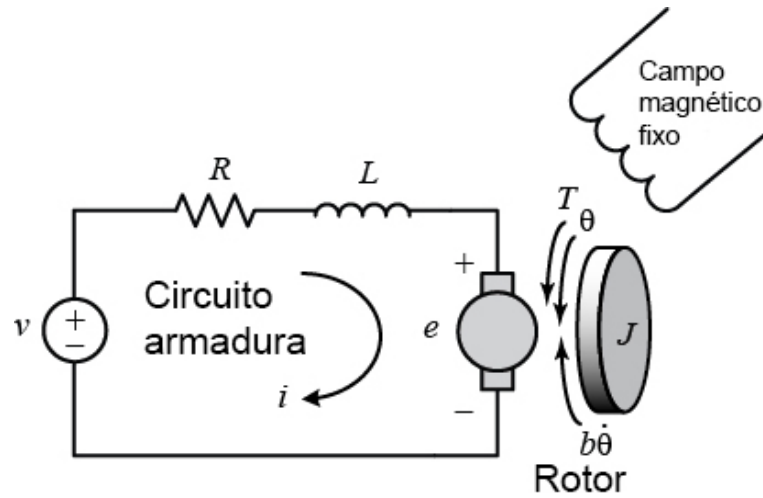
Fonte: (SIEGWART; NOURBAKSH, 2004) (Adaptado)

onde K_t é a constante de torque. A partir da segunda lei de Newton, a equação dinâmica do motor CC pode ser expressa por:

$$J\ddot{\theta}(t) + b\dot{\theta}(t) = T(t) - T_l(t), \quad (2.3)$$

e $\ddot{\theta}$ é a aceleração angular e T_l é o torque de carga.

Figura 2.3 – Esquemático motor CC



Fonte: (MICHIGAN, 2012) (Adaptado)

De acordo com as equações (2.2) e (2.3), o modelo no espaço de estados que descreve o motor cc considerando a rotação angular (w) como saída do sistema é:
em que w e v são as velocidades angulares e lineares respectivamente.

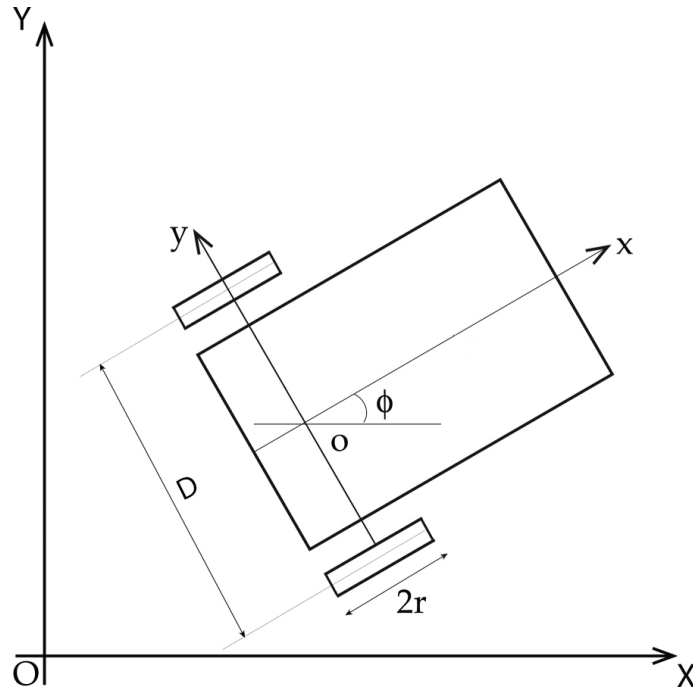
$$\begin{bmatrix} \dot{w}(t) \\ \dot{i}(t) \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K_t}{J} \\ -\frac{K_t}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} w(t) \\ i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v, \quad (2.4)$$

$$w(t) = [1 \quad 0] \begin{bmatrix} w(t) \\ i(t) \end{bmatrix}$$

2.1.2 Modelo cinemático

Cinemática é o estudo mais básico de como sistemas mecânicos se movimentam. Em robótica móvel, o objetivo do modelo cinemático é relacionar o seu movimento no eixo inercial levando em consideração sua geometria e os esforços dos atuadores (SIEGWART; NOUR-BAKHSI, 2004). A Figura 2.4 mostra o modelo do robô utilizado neste trabalho. Ele consiste de duas rodas de raio r conectadas a dois motores independentes. As rodas estão separadas entre si por uma distância D . O centro de massa do robô está localizado em o . A posição do robô no eixo de coordenadas globais $\{O, X, Y\}$ pode ser descrita pelo vetor $q = [x \ y \ \phi]^T$ onde x e y são as coordenadas do centro de massa do robô e ϕ é sua orientação em relação ao eixo global.

Figura 2.4 – Robô móvel de duas rodas tracionadas



Considerando que as rodas girem sem deslizamento e não há deslocamento lateral do robô, a equação matemática que representa a orientação do corpo referente ao eixo global na forma diferenciável é:

$$A(q)\dot{q} = \dot{x} \sin \phi - \dot{y} \cos \phi = 0 \quad (2.5)$$

onde $A \in \mathbb{R}^{m \times n}$ é a matriz de restrições definida com as coordenadas generalizadas q (Jian Wang; Chun-Yi Su; Oya, 2004). O modelo cinemático não linear (ou modelo cinemático de primeira ordem) do robô pode ser representado como (ORIOLO; LUCA; VENDITTELLI, 2002):

$$\dot{q} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} \cos \phi(t) & 0 \\ \sin \phi(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}, \quad (2.6)$$

O vetor de entradas pode ser descrito como $u(t) = [v(t) \quad w(t)]^T$. As velocidades lineares e angulares do robô são descritas da seguinte maneira:

$$v(t) = \frac{v_L(t) + v_R(t)}{2}, \quad w(t) = \frac{v_L(t) - v_R(t)}{D}, \quad (2.7)$$

sendo v_L e w_L as velocidades lineares e angulares da roda tracionada esquerda e v_R w_R as velocidades lineares e angulares da roda tracionada direita. As velocidades lineares e angulares estão relacionadas como:

$$v_L(t) = w_L(t) \cdot r, \quad v_R(t) = w_R(t) \cdot r, \quad (2.8)$$

portanto, o modelo cinemático do robô em relação a velocidade linear e angular pode ser representado em matrizes como:

$$\begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} w_R(t) \\ w_L(t) \end{bmatrix}, \quad (2.9)$$

então, as equações das restrições para o modelo escolhido (sem deslizamento e deslocamento lateral das rodas) são:

$$\begin{aligned} \dot{x} \sin \phi - \dot{y} \cos \phi &= 0, \\ \dot{x} \cos \phi + \dot{y} \sin \phi &= r w_R + D/2 \dot{\phi}, \\ \dot{x} \cos \phi + \dot{y} \sin \phi &= r w_L - D/2 \dot{\phi} \end{aligned} \quad (2.10)$$

portanto a matriz de restrições A que define o modelo é:

$$A = \begin{bmatrix} \sin \phi & -\cos \phi & 0 \\ \cos \phi & \sin \phi & D/2 \\ \cos \phi & \sin \phi & -D/2 \end{bmatrix} \quad (2.11)$$

2.2 Controle de trajetória

Trajетória é definida por um caminho que um determinado objeto em movimento segue no espaço em função do tempo (MEHTA; MEHTA, 2018). Em robótica móvel, geração de trajetória define a posição e orientação em que o robô deve estar em um determinado instante de tempo (SIEGWART; NOURBAKHS, 2004).

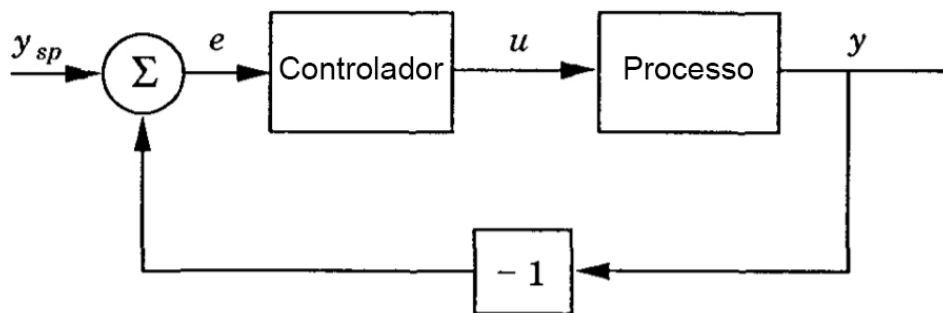
O problema de controle de trajetória para robôs móveis tem sido um assunto bastante abordado nas últimas décadas e está relacionado a lei de controle que força o robô a seguir um caminho com uma referência no tempo (BETOURNE; CAMPION, 1996; CAMPION; BASTIN; D'ANDREA-NOVEL, 1993). Há vários métodos para resolver esse problema, por exemplo, Redes Neurais Artificiais, Controle PID, Controle Fuzzy, Controle Fuzzy Adaptativo, Algoritmos genéticos (VELASCO-VILLA; ALVAREZ-AGUIRRE; RIVERA-ZAGO, 2007; WEI-

BING et al., 2009; KECSKÉS; ODRY, 2012). Cada método soluciona o problema com suas vantagens e desvantagens. Neste trabalho, são implementados dois métodos que trabalham em conjunto: controlador PID e Fuzzy. O primeiro, é usado para assegurar a rotação angular no eixo do motor como saída tendo um valor de referência na entrada e o segundo é utilizado para o controle de trajetória dos robôs.

2.2.1 Controlador PID

Controlador PID é um dos algoritmos mais comuns de controle de sistemas (ÅSTRÖM; HäGGLUND, 2006). Ele utiliza a ideia de realimentação unitária que consiste em adaptar o processo a partir do erro entre a saída do sistema e a saída desejada. Esse princípio está ilustrado na Figura 2.5 sendo que $e = y_{sp} - y$ é chamado de erro de controle, y_{sp} e y , as saídas

Figura 2.5 – Princípio de realimentação unitária



desejada e atual no instante t respectivamente. Já a variável u é chamada de ação corretiva do sistema.

Controlador PID é assim denominado devido as ações P (Proporcional), I (Integrativa) e D (Derivativa) serem usados para ajustes no sistema. O termo Proporcional é o valor que multiplica o erro de controle no instante presente. O termo Integrativo é o valor acumulativo da soma dos erros passados até o instante presente. O termo Derivativo é o valor da derivação do erro no instante presente que é considerado como a predição do erro futuro (UNBEHAUEN, 2009).

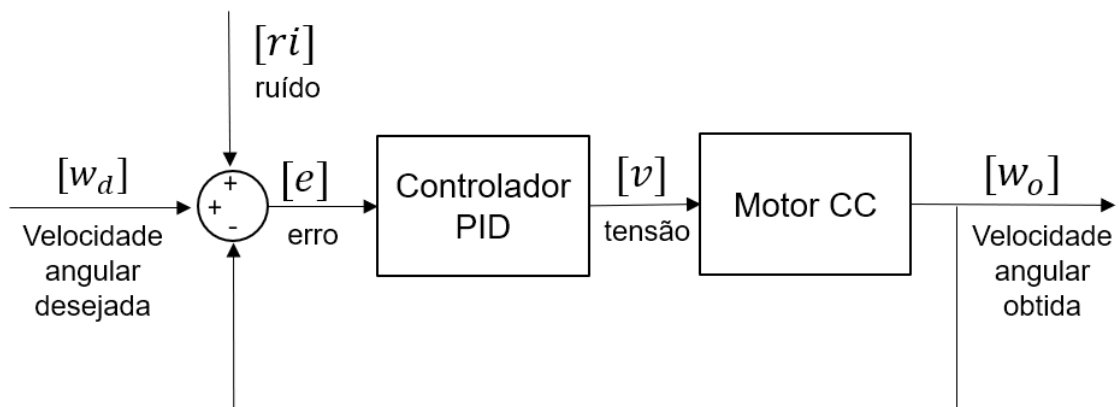
O modelo matemático que descreve esse controlador pode ser visto na Equação 2.12 ,

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.12)$$

em que K_p é o ganho proporcional, K_i é o ganho integrativo, K_d é o ganho derivativo, $e(t)$ é o erro de controle no instante t e τ é a variável de integração.

No problema de controle de trajetória tratado nesse artigo, o controlador PID é responsável por ajustar as tensões de entrada nos motores de corrente contínua para garantir velocidades angulares desejadas considerando ruídos indesejados. Com isso, é obtida a velocidade angular necessária na saída do sistema para que a posição e orientação do robô sejam atingidas com precisão. Esse é o princípio utilizado em servomotores. A relação entre essas variáveis foi deduzida na Seção 2.1.1 e a Figura 2.6 ilustra o diagrama de blocos do problema tratado. A velocidade angular de referência desejada é somada a um ruído branco gaussiano, e desse valor é subtraído a velocidade angular obtida pela realimentação, resultando no erro. Este é passado para o controlador gerando a tensão necessária para o modelo do motor tendo como saída a rotação.

Figura 2.6 – Diagrama de blocos de realimentação do motor CC



Existem inúmeros métodos para determinar as constantes de ganho do controlador K_p , K_i e K_d . Para esse problema, o primeiro método de resposta ao degrau de *Ziegler-Nichols* foi escolhido. Esse método heurístico consiste em atribuir inicialmente os ganhos K_i e K_d como zero e variar o ganho até ter como resultado uma resposta estável e com oscilações consistentes analisando a resposta do sistema operando em malha fechada (ZIEGLER; NICHOLS, 1942). Os parâmetros da constante de tempo T , tempo morto L obtidos na análise da resposta, são usados para calcular as os ganhos do controlador seguindo as relações da Tabela 2.1.

Tabela 2.1 – Valores dos parâmetros para cálculos dos ganhos

Controlador	K_p	T_i	T_d
P	T/L	∞	0
PI	0,9 T/L	L/0,3	0
PID	1,2T/L	2L	0,5L

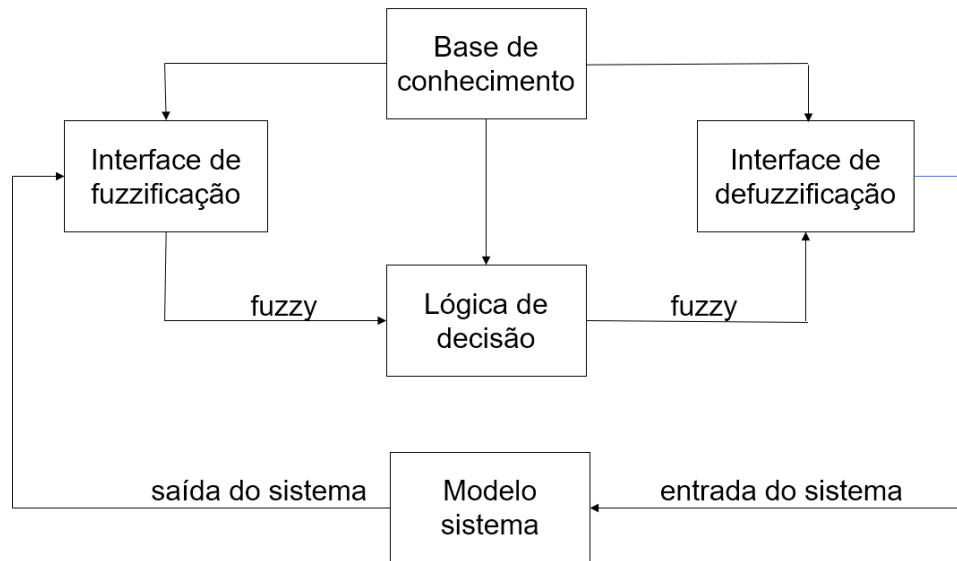
2.2.2 Controlador Fuzzy

O objetivo de utilizar um controlador é diminuir o erro entre a saída desejada do sistema e a atual em um determinado instante de tempo mantendo o sistema estável considerando sua dinâmica. No trabalho aqui apresentado, a saída desejada é o vetor $q(i) = [x(i) \quad y(i) \quad \phi(i)]^T$, que é a posição e orientação que o robô deve estar na trajetória predefinida no passo i .

Controladores Lógicos Fuzzy (CLF) são baseados em conjuntos e lógica Fuzzy apresentados por Zadeh (ZADEH, 1965; ZADEH, 1988). A lógica Fuzzy se assemelha com o pensamento do ser humano e a linguagem natural e por isso tem sido utilizada nas últimas décadas. O conceito Fuzzy pode ser entendido como uma situação em que não é possível responder simplesmente "sim" ou "não" mas "talvez" e "quase". Os valores lógicos são números entre 0 e 1 diferentemente da lógica booleana (LEE, 1990). Um controlador Fuzzy consiste de quatro componentes principais: interface de fuzzificação, base de conhecimento, lógica de decisão e defuzzificação. A interface de fuzzificação é responsável por medir os valores de entradas, reproduzir e mapear a entrada para o universo de discurso e converter a entrada em variáveis linguísticas. A base de conhecimento consiste do conhecimento *a priori* e dos objetivos da aplicação. A lógica de decisão representa as ações a serem tomadas a partir da combinação das variáveis linguísticas de entrada e saída. A etapa de defuzzificação é responsável por reproduzir e mapear as variáveis linguísticas de saída para um intervalo definido (LEE, 1990). A Figura 2.7 ilustra a configuração básica de um CLF.

A utilização de algoritmos Fuzzy na solução de problemas de controle foi iniciada com o trabalho pioneiro de Mamdani (MAMDANI, 1974), seguido por Sugeno (SUGENO, 1985), motivados pelo trabalho de Zadeh (ZADEH, 1965). No controlador apresentado por Mamdani, foi adotado um método para o processo de decisão baseado em regras do tipo "SE A ENTÃO B", nas quais A e B são as variáveis linguísticas definidas. Diante de dificuldades para a utilização do processo de decisão de Mamdani, Takagi e Sugeno (TAKAGI; SUGENO, 1983) propuseram um método de tomada de decisão que somente o antecedente das regras é formado por variáveis

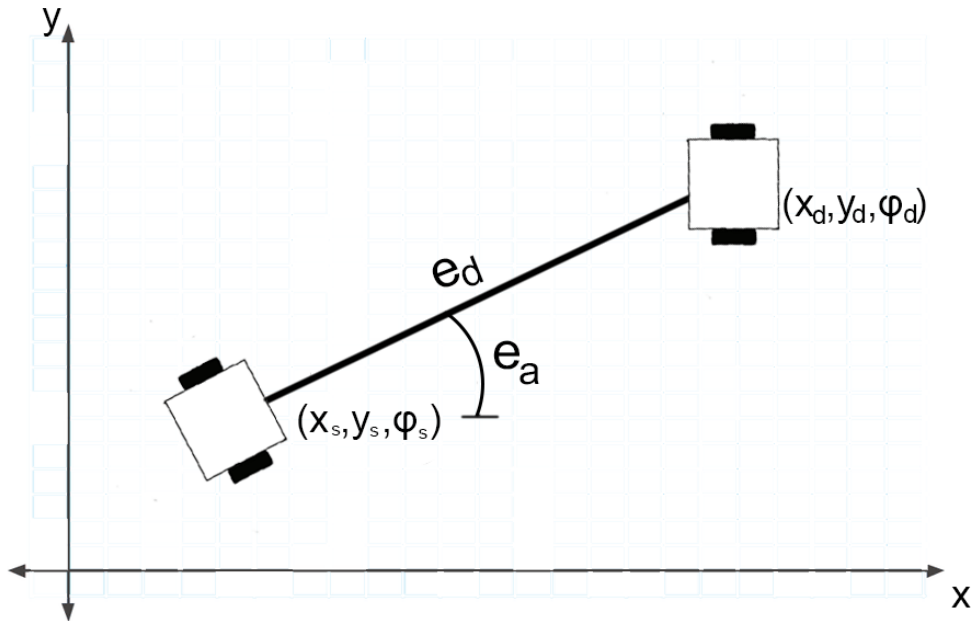
Figura 2.7 – Configuração do Controlador Lógico Fuzzy



Fuzzy. O consequente de cada regra é expresso por uma função linear de valores observados nas variáveis de entrada do sistema.

O controlador Fuzzy utilizado neste trabalho tem como objetivo determinar a velocidade referência de cada motor a partir dos erros de distâncias e angulares calculados. Esses erros representam a diferença da posição e orientação do robô desejada e atual ilustrados na Figura 2.8. Os erros de posição e_d e orientação e_a podem ser matematicamente expressos nas Equações 2.13 e 2.14,

Figura 2.8 – Representação dos erros de posição e orientação



$$e_d(t) = \sqrt{(x_d(t) - x_s(t))^2 + (y_d(t) - y_s(t))^2}, \quad (2.13)$$

$$e_a(t) = \tan^{-1} \frac{y_d(t) - y_s(t)}{x_d(t) - x_s(t)} - \phi_s(t) \quad (2.14)$$

onde x_d e x_s são as posições desejadas e atuais no eixo cartesiano X pertencentes ao vetor q_d , y_d e y_s são as posições desejadas e atuais no eixo cartesiano Y pertencentes ao vetor q_s e ϕ_s é a orientação angular atual do robô. Diante dos erros calculados, é necessário que o controlador aja para gerar duas velocidades para os motores CC a fim de minimizar o erro e garantir que o robô siga a trajetória desejada.

As funções de pertinência, que são funções características do sistema, das variáveis de entrada e saída do controlador Fuzzy usado podem ser vistas na Figura 2.9. As variáveis linguísticas utilizadas são: 'Zero' (Z), 'Baixo' (B), 'Alto' (A), 'Muito Negativo' (MN), 'Negativo' (N), 'Positivo' (P), 'Muito Positivo' (MP), 'Muito Direita' (MD), 'Direita' (D), 'Reto' (R), 'Esquerda' (E) e 'Muito Esquerda' (ME). Os valores do erro de distância e_d variam de 0 a 1 metro; erro angular e_a de -180 a 180 graus; velocidade linear v de 0,0 a 0,3 metros por segundo e velocidade angular w de -0,3 a 0,3 radianos por segundo. A quantidade de variáveis linguísticas utilizadas foi definida utilizando a base de conhecimento próprio do sistema e os valores mínimos e máximos de cada variável linguística foram estabelecidos de acordo com as limitações

do motor utilizado. As regras Fuzzy do CLF estão resumidas na Tabela 2.2. Por exemplo, a primeira regra SE o erro linear (e_d) for praticamente zero e o erro angular (e_a) for muito negativo ENTÃO a velocidade linear (v) é praticamente zero e a velocidade angular é negativa que significa girar muito à esquerda. O CLF usado tem duas entradas e duas saídas, seu esquemático foi gerado no Matlab[©] e pode ser visto na figura 2.10. As entradas são os erros e_d e e_a e as saídas são as velocidades de referência linear e angular.

Figura 2.9 – Funções de pertinência do CLF.

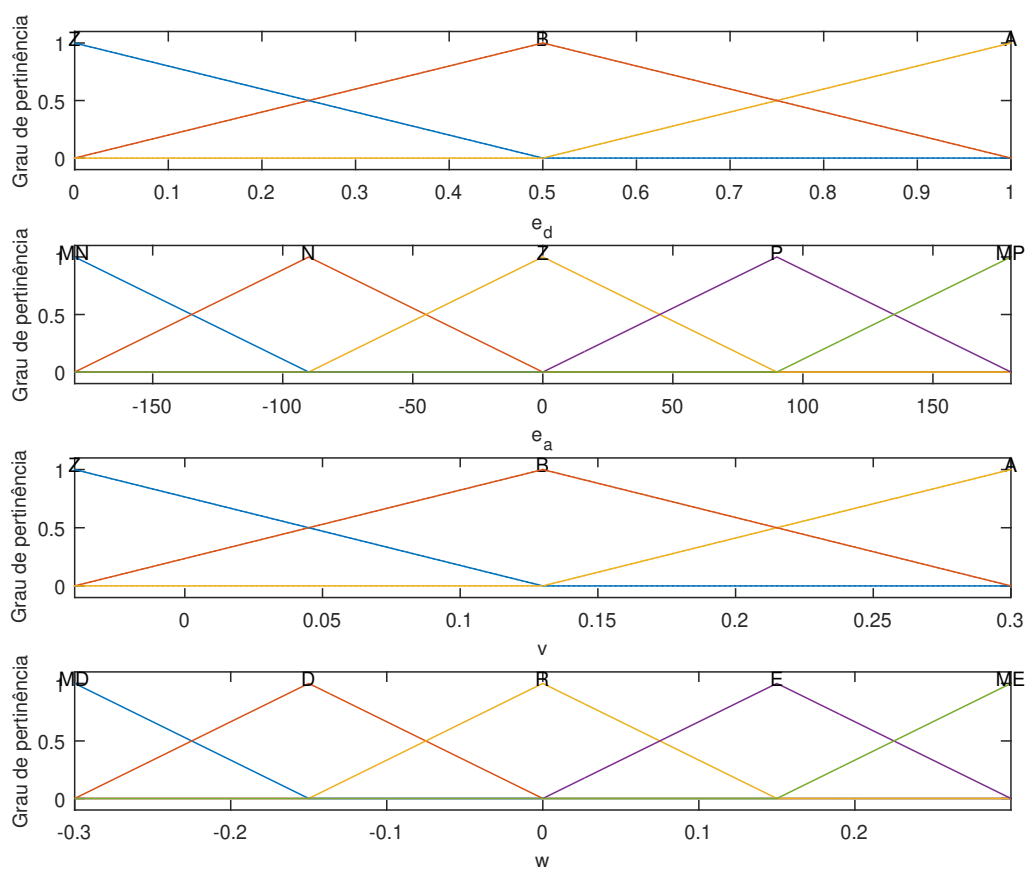
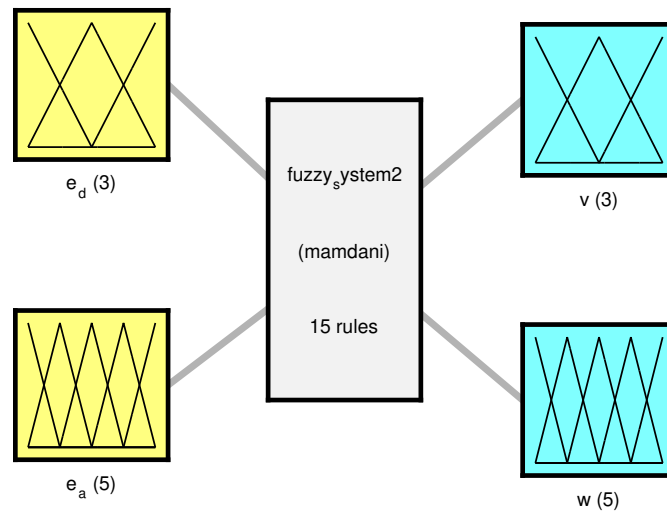


Tabela 2.2 – Conjunto de regras Fuzzy

-
- 1) SE $e_d = Z$ E $e_a = MN$ ENTÃO $v = Z$ E $w = ME$
 - 2) SE $e_d = Z$ E $e_a = N$ ENTÃO $v = Z$ E $w = E$
 - 3) SE $e_d = Z$ E $e_a = Z$ ENTÃO $v = Z$ E $w = R$
 - 4) SE $e_d = Z$ E $e_a = P$ ENTÃO $v = Z$ E $w = D$
 - 5) SE $e_d = Z$ E $e_a = P$ ENTÃO $v = Z$ E $w = MD$
 - 6) SE $e_d = B$ E $e_a = MN$ ENTÃO $v = B$ E $w = ME$
 - 7) SE $e_d = B$ E $e_a = N$ ENTÃO $v = B$ E $w = E$
 - 8) SE $e_d = B$ E $e_a = Z$ ENTÃO $v = B$ E $w = R$
 - 9) SE $e_d = B$ E $e_a = P$ ENTÃO $v = B$ E $w = D$
 - 10) SE $e_d = B$ E $e_a = MP$ ENTÃO $v = B$ E $w = MD$
 - 11) SE $e_d = A$ E $e_a = MN$ ENTÃO $v = A$ E $w = ME$
 - 12) SE $e_d = A$ E $e_a = N$ ENTÃO $v = A$ E $w = E$
 - 13) SE $e_d = A$ E $e_a = Z$ ENTÃO $v = A$ E $w = R$
 - 14) SE $e_d = A$ E $e_a = P$ ENTÃO $v = A$ E $w = D$
 - 15) SE $e_d = A$ E $e_a = MP$ ENTÃO $v = A$ E $w = MD$
-

Figura 2.10 – Esquemático do Controlador Fuzzy desenvolvido

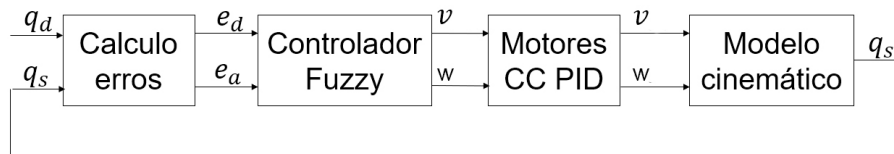


System fuzzy_system2: 2 inputs, 2 outputs, 15 rules

O diagrama de blocos do controle da trajetória pode ser resumido na Figura 2.11. Os erros são calculados no primeiro bloco e passados como entradas para o controlador Fuzzy. O mesmo gera como saídas as velocidades de referência para os motores CC, cujos controlado-

res PID implementados asseguram os valores de velocidades de rotação e os repassam para o modelo cinemático do robô.

Figura 2.11 – Esquemático resumido do trabalho



3 DETECÇÃO DE FALHAS EM ATUADORES

Um robô móvel autônomo necessita de uma variedade de subsistemas para funcionar corretamente, por exemplo, se movimentar no espaço, sentir o ambiente ao redor e comunicar com outros sistemas móveis ou estacionários. Alguns dos principais subsistemas presentes nos RMAs são (SIEGWART; NOURBAKHS, 2004):

- *Subsistema de potência*: Esta é a parte que providencia energia para todos os componentes do robô. Ela geralmente consiste de uma fonte de alimentação (geralmente baterias) e rede de distribuição de energia (tais como cabos, chaves, circuitos de proteção). Falhas nesses subsistemas podem acarretar perda total de funcionamento do sistema.

- *Subsistema de direção*: Esta é a parte que proporciona a força requerida para que o robô se mova de acordo com os comandos direcionados pelo microcontrolador. Ela normalmente contém motores, caixas de redução, amplificadores, freios, dispositivos de controle e *encoders*. Falhas nesse subsistema prejudicam a mobilidade do robô.

- *Subsistema de suspensão*: Esta é a parte responsável por amenizar vibrações do sistema mantendo-o estável. Suspensões e rodas são incluídas nessa parte. Falhas nesse subsistema podem ocasionar na instabilidade do robô e afetar sua mobilidade.

- *Subsistema de comunicação*: Robôs móveis geralmente trocam informações com o uma central de controle ou outros dispositivos móveis via wireless. Esse subsistema é o responsável por tal ato. Um problema causado nele pode gerar falta de comunicação e de troca de dados especialmente em atividades de cooperação de robôs.

- *Subsistema sensorial*: Sensores são responsáveis por quantizar medidas precisamente tais como posição, velocidade, ângulos, distâncias relativas de objetos, entre outras. Eles são responsáveis por importante parte de todo o sistema. Por isso, é necessário que os sensores embarcados no sistema sejam garantidos de perfeito funcionamento.

Na Tabela 3.1, estão vários componentes mecânicos e eletromecânicos de sistemas robóticos. Ela mostra a taxa de falhas λ por hora de funcionamento. Apesar da quantidade de subsistemas presentes em um robô móvel, um dos mais suscetíveis à ocorrência de falhas é o atuador. Por isso, nesse trabalho serão detectadas falhas de apenas uma parte do subsistema de direção: motores de corrente contínua. Serão consideradas falhas severas e passíveis de reconfiguração no componente. Nesta seção, serão apresentadas soluções para filtragem de sinais, estimação de estados e cálculo de eficiência usando um Filtro de Kalman.

Tabela 3.1 – Taxas de falhas típicas de elementos mecânicos e eletromecânicos

Elementos Mecânicos	$\lambda [h^{-1}]$	Elementos eletromecânicos	$\lambda [h^{-1}]$
Rolamentos	$1,64 \times 10^{-6}$	Atuadores em geral	26×10^{-6}
Mancais	$2,38 \times 10^{-6}$	Escovas em geral	9×10^{-6}
Correias	$19,72 \times 10^{-6}$	Cabos em geral	1×10^{-6}
Acopladores	$5,54 \times 10^{-6}$	Motores elétricos em geral	9×10^{-6}
Engrenagens	$4,69 \times 10^{-6}$	Geradores em geral	73×10^{-6}
Bombas	$43,65 \times 10^{-6}$	Reguladores em geral	13×10^{-6}
Seladores	$5,47 \times 10^{-6}$		
Válvulas hidráulicas	$8,83 \times 10^{-6}$		

Fonte: (MACDIARMIND; BART, 1998) (Traduzido)

3.1 Filtro de Kalman

O Filtro de Kalman foi primeiramente apresentado por Rudolf E. Kalman em 1960 em seu artigo descrevendo uma solução recursiva para o problema de filtragem linear ótima computacionalmente eficiente (KALMAN, 1960). Nessa solução recursiva, cada estimativa de estado atualizado é derivada da estimativa do estado anterior e a medições atual de forma que a covariância do erro estimado seja minimizada. O filtro pode ser usado tanto para estimações no passado, presente e até de estados futuros, sendo preciso até quando o sistema modelado é desconhecido (WELCH; BISHOP, 2001).

O modelo para o Filtro de Kalman assume que o estado real no tempo k é obtido através do estado no tempo $k - 1$ de acordo com

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.1)$$

onde

- x é o vetor de estados,
- A é a matriz de transição de estados,
- B é a matriz de transição de entradas externas,
- u é o vetor de entradas externas,
- w é o ruído do processo, assumido como sendo amostrado de uma distribuição normal multivariada de média zero e covariância Q .

No tempo k , uma observação (ou medição) y do estado real x é feita de acordo com

$$y_k = Cx_k + v_k \quad (3.2)$$

sendo C a matriz de observação, que mapeia o espaço de estados real no espaço de estados observados, e v é o ruído de medição, assumindo como sendo um ruído branco gaussiano de média zero e covariância R .

O estado do filtro é representado por duas variáveis:

- $\hat{x}_{k|k}$, a estimativa *a posteriori*, após a medição, do estado no tempo k , dadas as medições até o tempo k , inclusive;
- $P_{k|k}$, a matriz de covariância do erro *a posteriori* (uma medida da acurácia estimada da estimativa do estado).

O Filtro de Kalman tem dois estágios principais: predição e correção. Na predição, a estimativa do estado no passo anterior é usado para obter uma estimativa do estado no tempo atual. Esta predição é chamada de estimativa *a priori*, pois não inclui a informação vinda da observação do tempo atual. Na fase de correção, a predição *a priori* é combinada com a medição atual para refinar a estimativa do estado. A estimativa refinada é chamada de estimativa *a posteriori*.

As equações que descrevem os dois estágios são (WELCH; BISHOP, 2001):

Predição

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k, \\ P_k^- &= AP_{k-1}A^T + Q. \end{aligned}$$

Correção

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-), \\ K_k &= P_k^- C^T (CP_k^- C^T + R)^{-1}, \\ P_k &= (I - K_k C)P_k^-. \end{aligned}$$

sendo K_k (ganho de Kalman), a primeira variável a ser calculada. Após isso, o estado estimado *a posteriori* é corrigido com a medição y e o estado estimado *a priori*. O último passo é obter a estimação *a posteriori* da covariância do erro estimado.

No trabalho aqui descrito, o Filtro de Kalman é usado para estimação dos estados de rotação angular e corrente dos motores do robô móvel. Com o estado estimado, a perda de efetividade γ do motor é calculada considerando os estados reais e estimados. Esse valor representa a taxa de degradação do motor e descreve o funcionamento exato do sistema. Esse procedimento será explicado no capítulo a seguir.

4 CONTROLE DE TRAJETÓRIA TOLERANTE A FALHAS

4.1 Reconfiguração das velocidades de referência

Na reconfiguração das velocidades de referência, a estimativa da perda de efetividade γ é levada em consideração para que o controlador possa ser reconfigurado de modo a simular um funcionamento normal do RMA, quando possível, levando em consideração os limites físicos dos componentes. As velocidades mensuradas são definidas por:

$$\begin{aligned}\gamma_R &= 1 - \frac{w_{Rm}(t)}{w_R(t)}, \\ \gamma_L &= 1 - \frac{w_{Lm}(t)}{w_L(t)}\end{aligned}\tag{4.1}$$

em que γ_R e γ_L são a perda de efetividade estimada dos motores direito e esquerdo respectivamente, w_{Rm} , w_{Lm} , w_R e w_L são as velocidades angulares mensuradas do motor esquerdo e direito e as obtidas pelo modelo do sistema, nessa ordem.

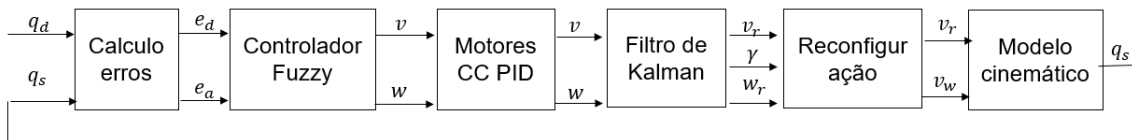
A efetividade do motor varia de 0 a 1, sendo que 0 representa falha total do motor e 1 representa o funcionamento na ausência de falhas. Quaisquer valores entre 0 e 1 são considerados falhas parciais do sistema. Dentre esses, para valores entre 0 e 0,3 considera-se que as falhas são severas e que o componente tem perda quase total, ou seja, situação em que a reconfiguração do controlador implicaria em saturação de atuador. Para valores entre 0,3 e 0,7 considera-se que a falha é moderada e pode ser compensada pelo mecanismo de reconfiguração do CTF; para valores entre 0,7 e 1 considera-se que a falha é suave e pode ser compensada pelo controlador Fuzzy, sem necessidade de reconfiguração do controle. Para a escolha desses intervalos e classificação da reconfiguração, foram levados em contas os limites físicos dos componentes e a rotação máxima estabelecida. Para que seja possível a reconfiguração na ocorrência de uma falha parcial, a velocidade escolhida como referência foi de 30% da velocidade máxima. A reconfiguração das velocidades é feita utilizando as equações

$$\begin{aligned}w_{Rr}(t) &= \frac{1}{1 - \gamma_R} w_{Rm}(t), \\ w_{Lr}(t) &= \frac{1}{1 - \gamma_L} w_{Lm}(t)\end{aligned}\tag{4.2}$$

sendo w_{Rr} e w_{Lr} as velocidades angulares de referência direita e esquerda perante a falha respectivamente.

O diagrama de blocos do sistema de controle tolerante a falhas apresentado até aqui pode ser visto na Figura 4.1. O bloco de reconfiguração foi adicionado ao esquemático apresentado anteriormente e utiliza o parâmetro γ calculado no filtro de Kalman para reajustar as velocidades dos motores esquerdo e direito que são passadas para o modelo cinemático.

Figura 4.1 – Diagrama de blocos com o estágio de reconfiguração



4.2 Controle cooperativo para múltiplos robôs móveis

O Controle Cooperativo Tolerante a Falhas é usado para prevenir comportamento anômalo dos robôs nas tarefas em que dois ou mais agentes necessitam cumprir um objetivo em conjunto. Quando há a ocorrência de uma falha parcial ou severa em um dos subsistemas de um dos robôs, os mesmos devem ser reconfigurados de forma a agir de acordo com a falha para compensar a perda de efetividade e poder cumprir com o objetivo. O algoritmo do controle cooperativo tolerante a falhas é centralizado, ou seja, um módulo de controle é usado para todos os robôs móveis. Considera-se que cada robô se comunica com os outros tendo informações de sua posição, velocidade e outros parâmetros necessários sem perda de comunicação. Os valores limiares para γ podem ser ajustados para cada tipo de robô e tarefa a ser cumprida. Dependendo da gravidade da falha, as seguintes decisões são tomadas:

- Se $\gamma = 0$, então não há a ocorrência de falhas e a tarefa pode ser cumprida.
- Se $0 < \gamma < 0,7$, é feita a reconfiguração nos controladores dos robôs de modo a cumprir com o objetivo com uma desempenho degradado.
- Se $0,7 < \gamma < 1$, a falha é considerada severa e não há a possibilidade de reconfiguração e então o robô é retirado da missão.

O diagrama de blocos do Controle Cooperativo Tolerante a Falhas de Atuadores pode ser visto na Figura 4.2 onde o primeiro bloco é do gerador de trajetórias responsável por passar

aos robôs as posições e orientações q_d a serem seguidas. A geração de trajetória é definida pelo vetor q que contém as posições $x(t)$ e $y(t)$ no instante de tempo t e o ângulo $\phi(t)$ estabelecido por $\arctan(y(t)/x(t))$. Em trabalhos que incluem obstáculos ou alterações de formações dos robôs durante o trabalho cooperativo, a trajetória deve ser recalculada a cada evento inesperado.

Nesse trabalho, três robôs são utilizados como exemplo para a simulação. Eles recebem a informação q_d do gerador de trajetória e velocidades de referência pelo bloco de decisão cooperativa dependendo do γ de cada um que é realimentado para o bloco de decisão.

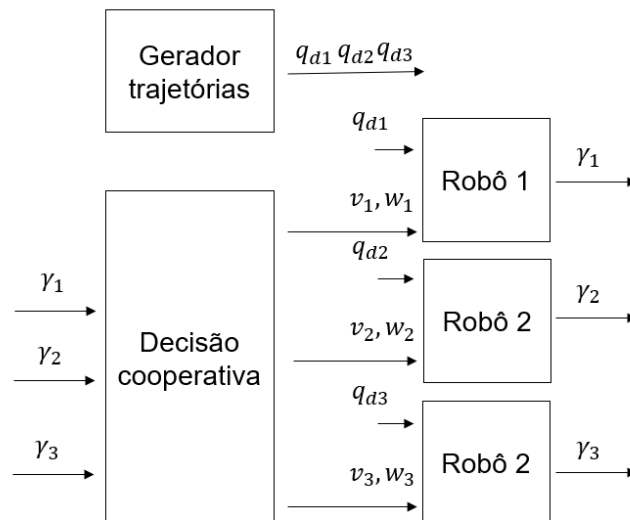
Os parâmetros para a simulação são:

Tabela 4.1 – Parâmetros de simulação

Start time	0
Final Time	80
Type	Step-variable
Solver	ode23 (Bogacki-Shampine)
Relative tolerance	1e-3
Absolute tolerance	auto

Os outros parâmetros não citados aqui foram usados como padrões do *software*.

Figura 4.2 – Diagrama de blocos final do CCTF com três robôs



5 RESULTADOS SIMULADOS E DISCUSSÕES

5.1 Controle de trajetória

Para validar os controladores propostos nesse trabalho, o modelo foi simulado no ambiente de simulação SimuLink do *software* Matlab. Trajetórias predefinidas foram passadas como referência para o modelo e o comportamento do robô móvel foi analisado ao decorrer do tempo. Nessa seção, a resposta temporal do controlador PID é observada. Após isso, as trajetórias realizadas pelo robô específicas são discutidas ao testar o controlador Fuzzy projetado.

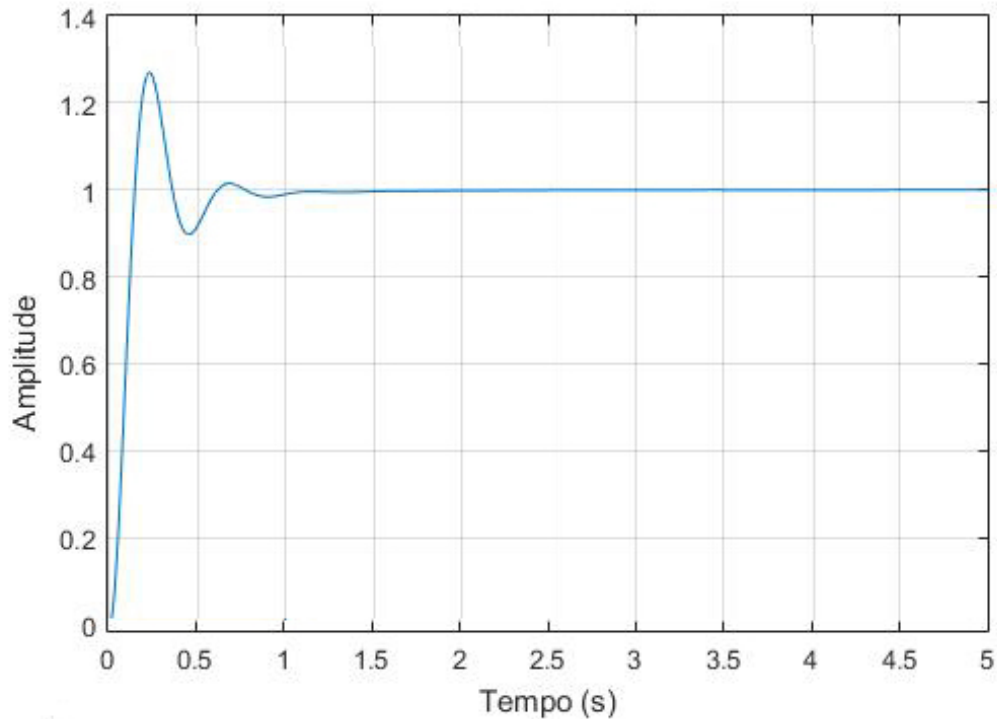
5.1.1 Controlador PID

Para efetuar os testes do controlador PID projetado, foram analisadas as respostas ao degrau do sistema. As variáveis consideradas como resultado foram o tempo até o regime permanente e oscilação. Os parâmetros do motor para simulação e desenvolvimento do controlador PID foram retirados de (MICHIGAN, 2012) e são: $J = 0.01kg.m^2$, $b = 0.1N.m.s$, $K = 0.01$, $R = 10\Omega$ e $L = 0.5Henry$. Os valores dos ganhos do controlador encontrados pelo método de Ziegler-Nichols PID *c/* overshoot foram: $K_p = 106,9$, $K_i = 147,53$ e $K_d = 1,13$. O gráfico obtido na simulação pode ser visto na Figura 5.1. A imagem mostra que o tempo de resposta ao degrau até o regime permanente, considerando $\pm 2\%$ é de aproximadamente 0,6 segundos. Pode-se concluir com esse resultado que o sinal da velocidade angular desejada é alcançada em até 0,6 segundos após o sinal de tensão de entrada ser aplicado. É possível ver no gráfico também uma oscilação na amplitude de no máximo 25% do sinal. Conclui-se com esse comportamento que até o regime permanente, a velocidade angular máxima alcançada será de no máximo 125% da desejada.

5.1.2 Controlador Fuzzy

A velocidade máxima alcançada pelo robô é de 0.3m/s (por limitação física de seus motores e escolha da velocidade passível de reconfiguração) e a posição inicial do robô é [0,0]. O primeiro vetor q_d da trajetória representa uma circunferência de raio 1m que é completada em 62,8319 segundos. O vetor foi definido por:

Figura 5.1 – Resposta ao degrau do controlador PID



$$x_d = \cos 0,1t$$

$$y_d = \sin 0,1t \quad (5.1)$$

$$\phi_d = \arctan y_d/x_d$$

onde t é o tempo de simulação e $0,1$ é a frequência angular do movimento periódico. O tempo gasto para a trajetória ser completa é de 2π segundos. A frequência angular foi escolhida de modo que o tempo de simulação para uma trajetória completa fosse 20π (62,8319) segundos. O Controlador Fuzzy tem como entrada os erros angulares e lineares entre a posição e orientação do robô atual e a desejada e como saída as velocidades angulares de referência para os motores do RMA. A Figura 5.2 apresenta o resultado obtido, sendo que a linha sólida azul é a trajetória que o robô realizou e a linha pontilhada laranja é a trajetória desejada. O tempo de simulação configurado tem duração de 8000 ciclos ou 80 segundos, com esse tempo, o robô completa uma trajetória e a recomeça logo em seguida. O tempo gasto para o robô completar a trajetória circular foi de 64,7371 segundos que significa um atraso de 1,9052 segundos. Esse atraso ocorreu devido ao erro do sinal do Controlador Fuzzy ser diferente de zero e as limitações física dos motores. Na Figura 5.3, as rotações angulares dos motores do robô durante a trajetória são mostradas. É possível ver que a partir do ciclo 6280 aproximadamente, há uma oscilação das

rotações angulares dos motores esquerdo e direito. Esse sinal inesperado é resultado da singularidade matemática na equação do e_a em arco tangente (atand) após a trajetória de referência ser completa. Essa singularidade não foi corrigida na simulação para verificar o comportamento do robô após mudanças abruptas nos sinais. Ainda na Figura 5.3, é possível ver essa mudança repentina após a trajetória ser completa na posição [1,0]. A Figura 5.4 expõe a superfície gerada do CLF. Ela mostra uma curva tridimensional de relações entre as variáveis e_a , e_d e v . É possível obter dela qualquer velocidade linear perante os erros de distância e angulares.

Figura 5.2 – Controle de trajetória sem a ocorrência de falhas

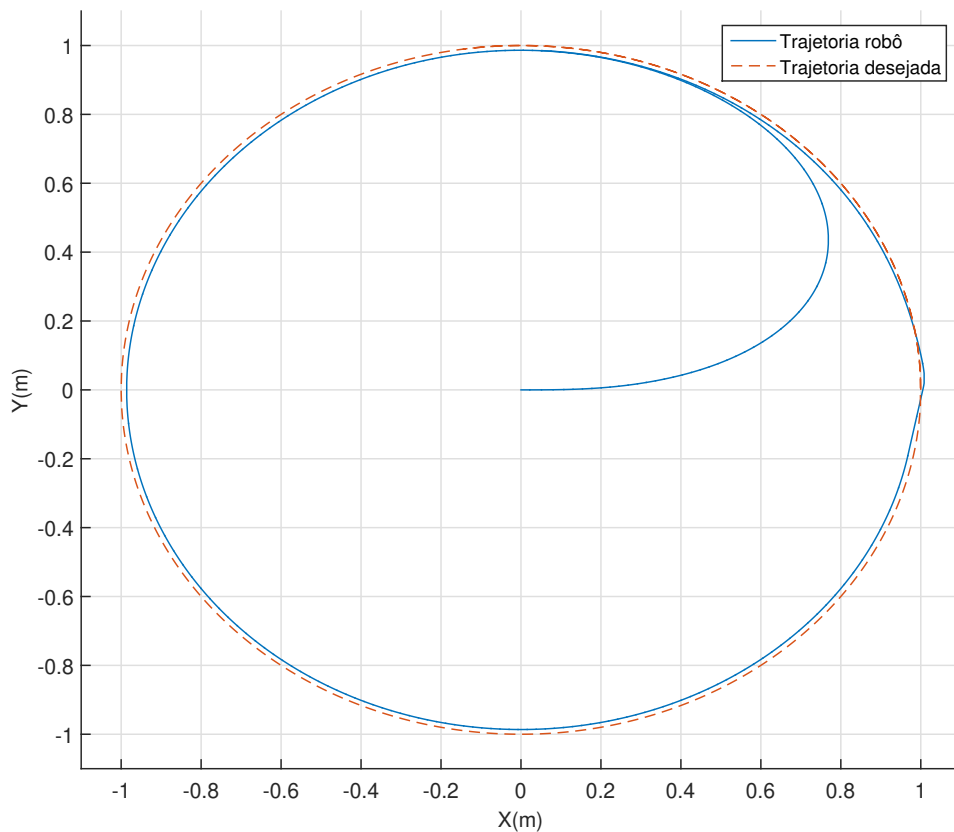


Figura 5.3 – Rotação angular dos motores do robô

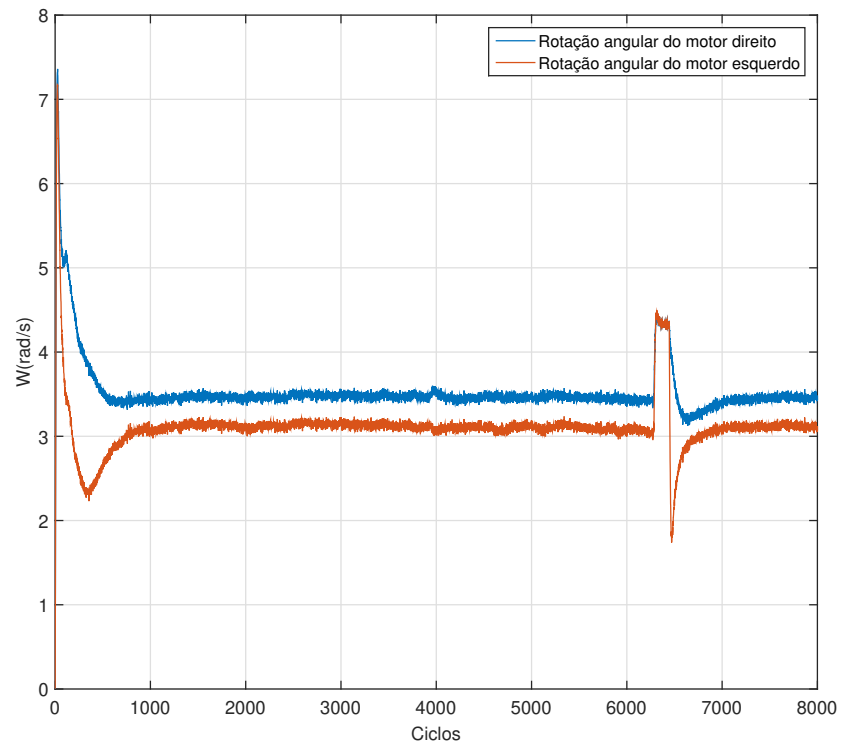
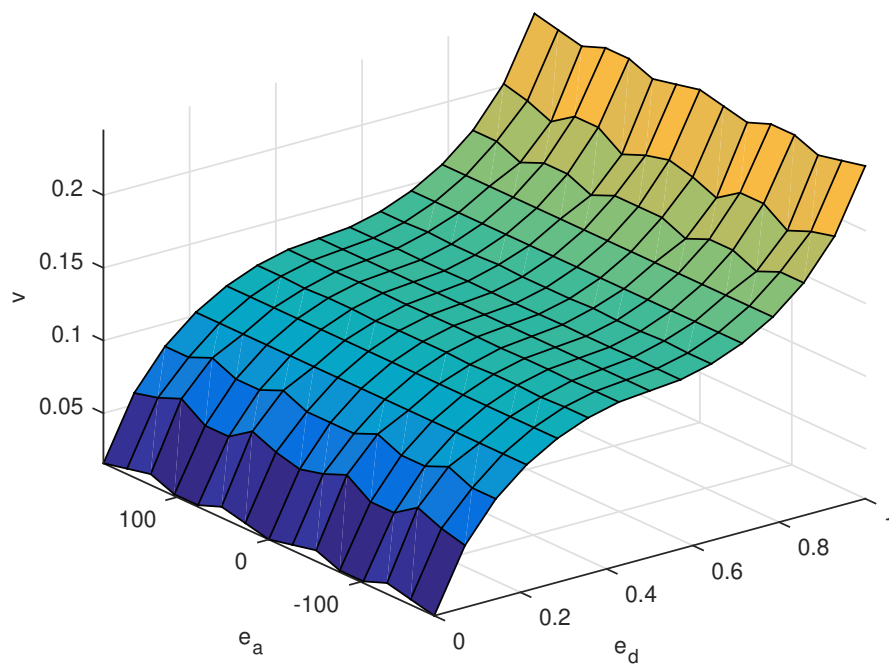


Figura 5.4 – Superfície do CLF



5.2 Detecção de falhas em atuadores

O problema tratado com o Filtro de Kalman consistiu em estimar os estados e foi utilizado para calcular a perda de efetividade dos motores do robô. Os parâmetros iniciais usados no algoritmo do Filtro são: $Q = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, $R = 10$, $x_0 = 3$ e $p_0 = 0$. Os valores de Q e R foram encontrados pelo método de tentativa e erro uma vez que foi considerado que as variâncias dos ruídos de processo e medição eram desconhecidas e aleatórias. Os valores de x_0 e p_0 foram escolhidos aleatoriamente. Como foi utilizado o modelo discreto do Filtro de Kalman, houve a necessidade de discretizar o modelo do motor. O mesmo foi discretizado pelo método ZOH (zero-order-hold) com o tempo de amostragem de $0,01s$ (FRANKLIN; POWELL; EMAMI-NAEINI, 2014). Sendo assim:

$$\begin{aligned} x &= \begin{bmatrix} 0,9048 & 0,0094 \\ -0,0002 & 0,9802 \end{bmatrix} x + \begin{bmatrix} 0,0001 & 0,0198 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x + 0 \end{aligned} \quad (5.2)$$

As Figuras 5.5 e 5.6 mostram os sinais da velocidade angular de cada motor sem utilizar o Filtro de Kalman e utilizando o Filtro de Kalman. O ruído somado aos sinais de medição tem potência de $0,001$ e é conhecido como ruído branco. O valor da potência do ruído foi escolhido para melhor demonstrar a filtragem com o método de Kalman e estimação da perda de efetividade dos motores. O ruído adicionado, é importante para tornar a simulação mais realista. Em ambas as figuras, as linhas contínuas em azul representam as rotações angulares sem a utilização do Filtro de Kalman e as linhas tracejadas em laranja são as rotações angulares após o Filtro de Kalman. A suavização pode ser notada nas linhas tracejadas que contém um sinal mais limpo do que o original. Para a simulação de detecção de falhas utilizando o Filtro de Kalman, a duração do teste foi de 80 segundos. Uma falha do tipo degrau com amplitude de 50% foi induzida a partir do tempo de 50 segundos.

Figura 5.5 – Filtro de Kalman no motor esquerdo

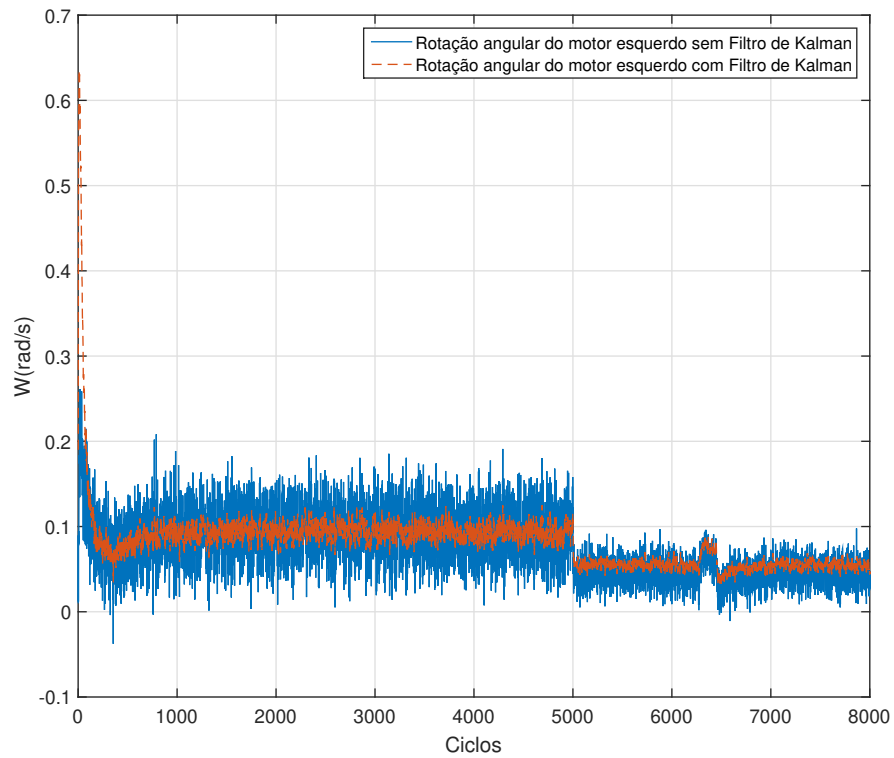
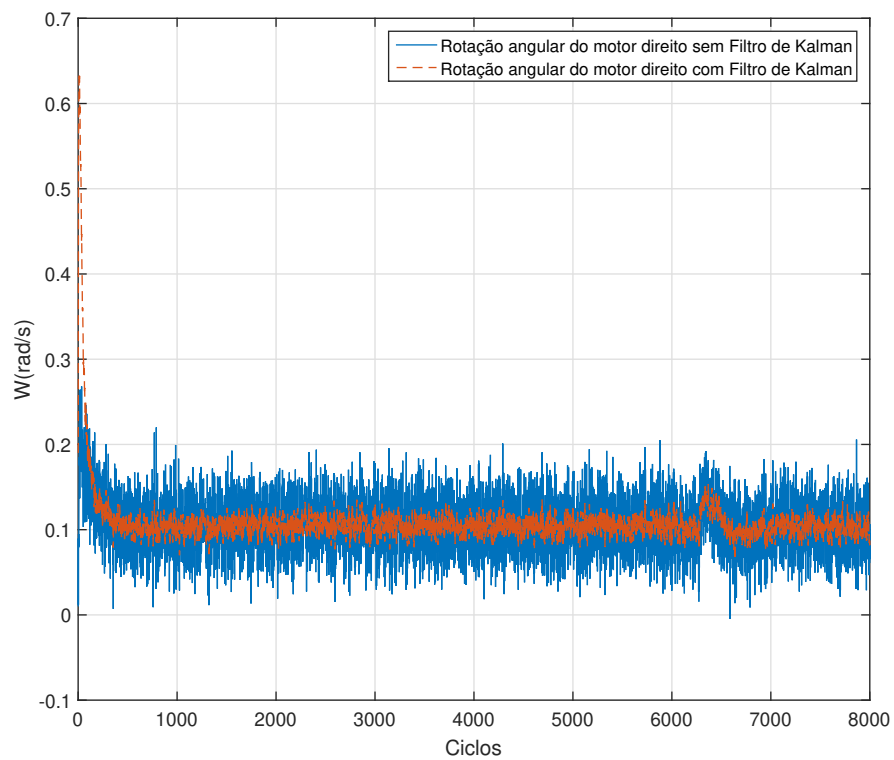
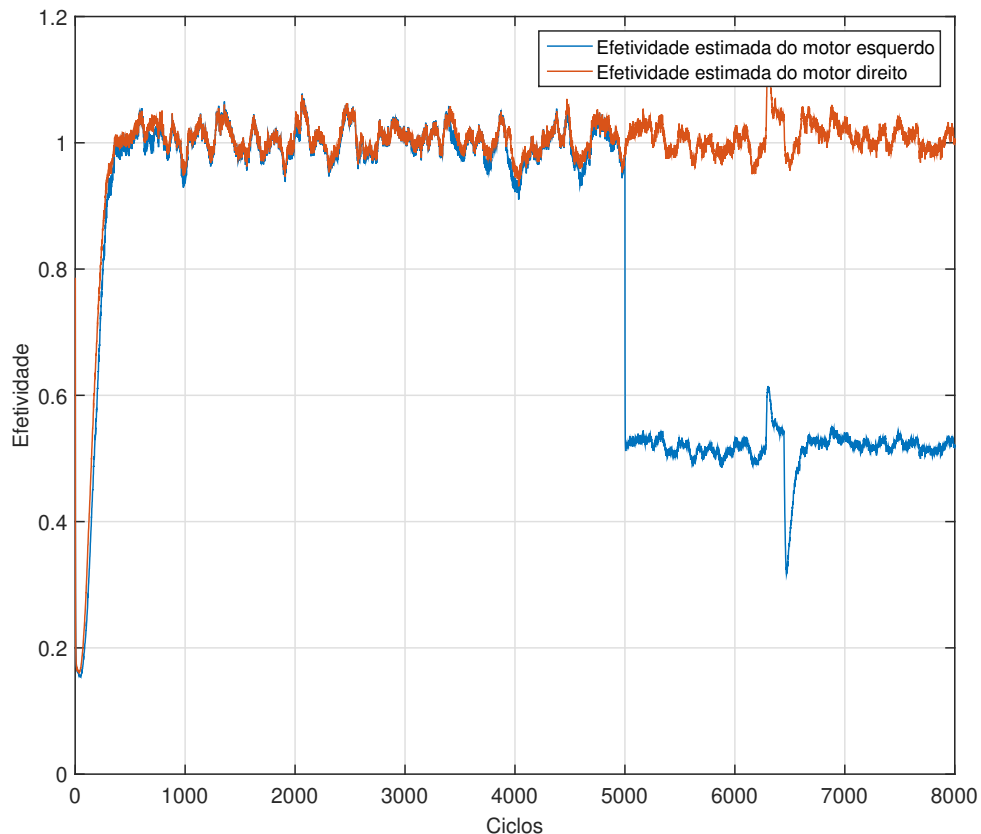


Figura 5.6 – Filtro de Kalman no motor direito



O gráfico da Figura 5.7 mostra a efetividade dos motores esquerdo e direito do robô calculada durante o percurso simulado. A linha contínua azul representa a efetividade do motor esquerdo durante o percurso e a linha segmentada laranja ilustra a efetividade do motor direito. Percebe-se que no tempo de aproximadamente 50 segundos, a efetividade estimada converge para valores próximos a 0,5 que confirma a falha simulada do motor esquerdo de 50%.

Figura 5.7 – Efetividade dos motores.



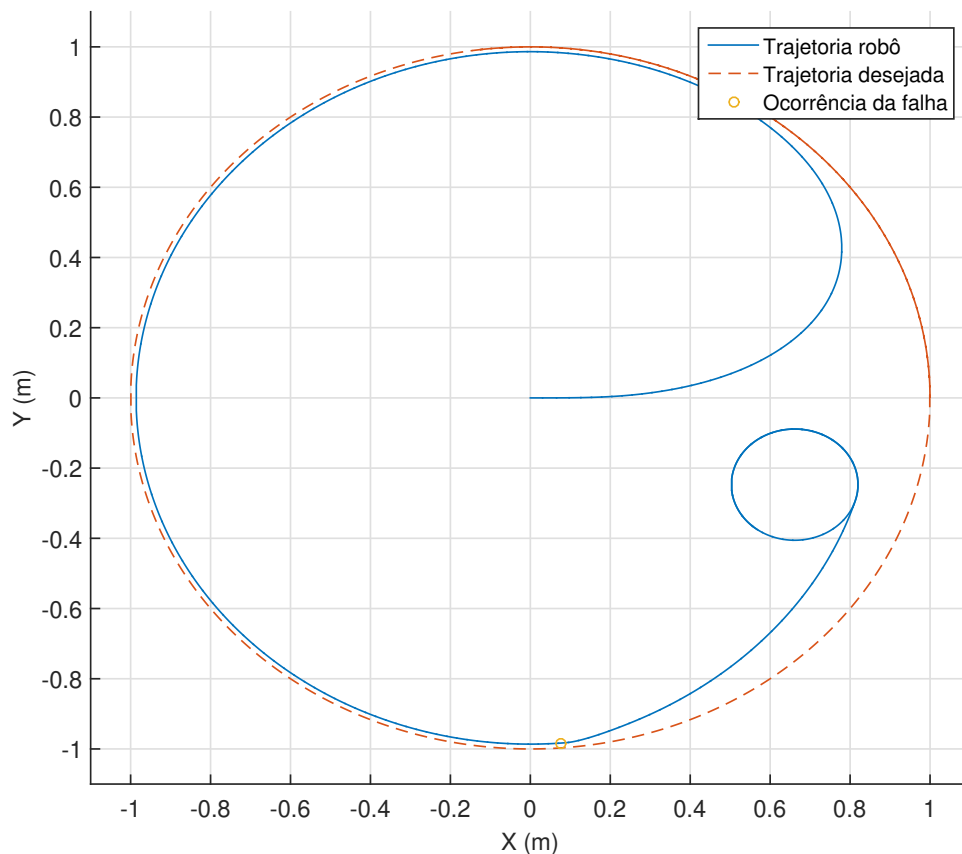
5.3 Controle de trajetória tolerante a falhas

5.3.1 Reconfiguração do controle

Para o Controle de Trajetória Tolerante a Falhas, o sistema foi submetido a uma simulação teste em que no ciclo 5000, ou 50 segundos, uma perda de 50% do motor esquerdo foi induzida a fim de verificar a estimativa do estado com o Filtro de Kalman e a perda de efetividade calculada. A Figura 5.8 demonstra a trajetória percorrida pelo robô após a simulação. O caminho feito pelo robô está representado pela linha contínua azul e a trajetória desejada

ilustrada pela linha segmentada laranja. É possível perceber que após a ocorrência da falha, demarcada pelo ponto \circ , o robô não consegue concluir a trajetória desejada. Esse comportamento ocorreu devido ao controlador Fuzzy não conseguir ajustar a velocidade de referência necessária devido a falha. Por isso, houve a necessidade de implementar um reconfigurador para ajustar os parâmetros de velocidades de referência. O método proposto efetua a compensação da falha. Esse ajuste só é possível devido a velocidade máxima inicial do robô ser limitada e portanto havendo a possibilidade de aumento de velocidade sem saturação do limite físico do atuador.

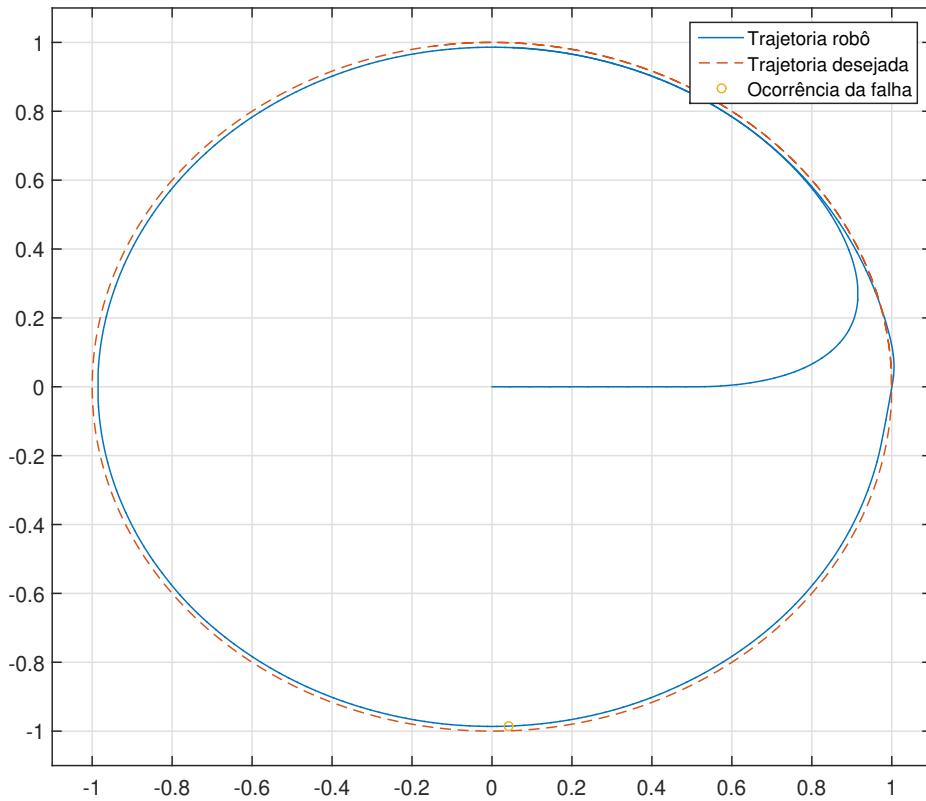
Figura 5.8 – Trajetória do robô perante a ocorrência de uma falha parcial



A Figura 5.9 mostra que o robô, mesmo perante a uma falha parcial de 50%, completou a trajetória e concluiu o objetivo após o estágio de reconfiguração. A Figura 5.10 apresenta a comparação entre os erros lineares obtidos nessa simulação com e sem a etapa de reconfiguração das velocidades de referência. É nítido perceber que após o instante da falha, no ciclo 5000 (50 segundos), o erro linear com o robô sem o reconfigurador aumenta gradualmente. Em contrapartida, o robô com reconfigurador de velocidade, representado no gráfico pela linha

contínua azul, consegue manter o erro após o instante da falha ocorrer. É possível concluir nessa Figura 5.10 que o reconfigurador e o controlador Fuzzy trabalhando juntos conseguiram resolver o problema de Controle de Trajetória Tolerante a Falhas.

Figura 5.9 – Trajetória completa do robô perante uma falha parcial.



5.3.2 Controle cooperativo

Neste trabalho, três robôs são escalados para cumprir uma tarefa em que devem percorrer as trajetórias circulares de raios 0,5, 0,7 e 0,9 metros em um tempo de 62,8319 segundos. As equações que definem a trajetória dos três robôs são:

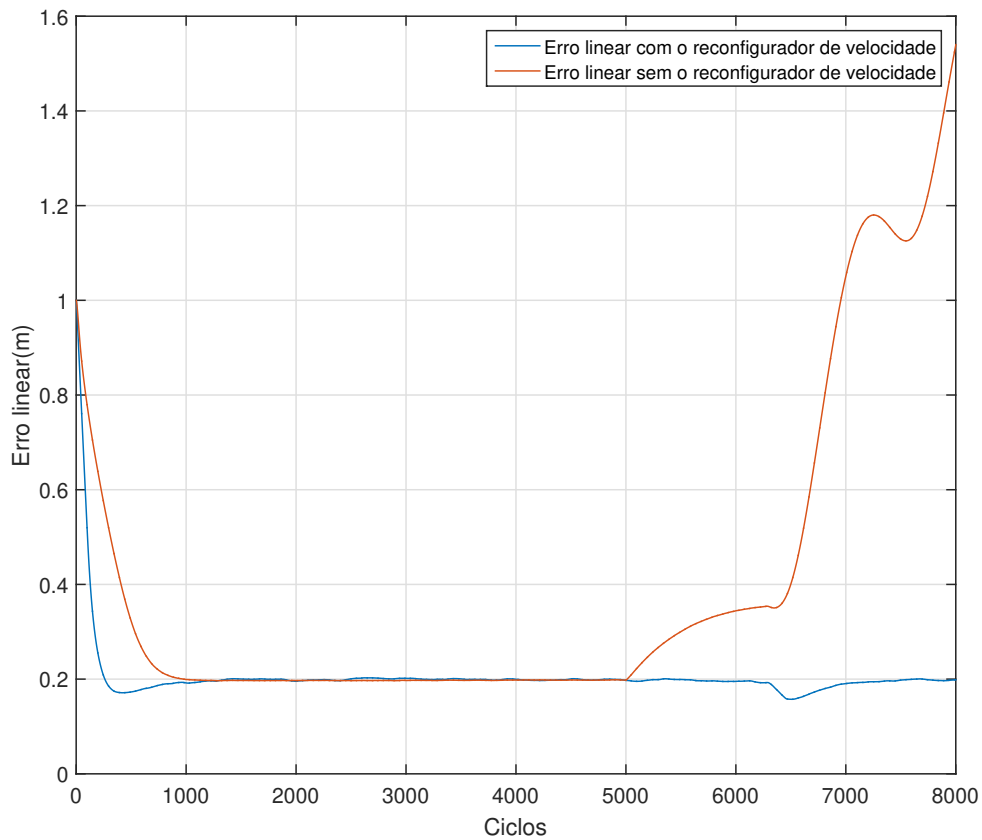
$$T_1 = 0,5 \cos 0,1t + 0,5 \sin 0,1t, \quad (5.3)$$

$$\phi_1 = \arctan 0,5 \sin 0,1t / 0,5 \cos 0,1t$$

$$T_2 = 0,7 \cos 0,1t + 0,7 \sin 0,1t, \quad (5.4)$$

$$\phi_2 = \arctan 0,7 \sin 0,1t / 0,7 \cos 0,1t$$

Figura 5.10 – Erros de distância com e sem o reconfigurador.



$$T_3 = 0,9 \cos 0.1t + 0,9 \sin 0.1t.$$

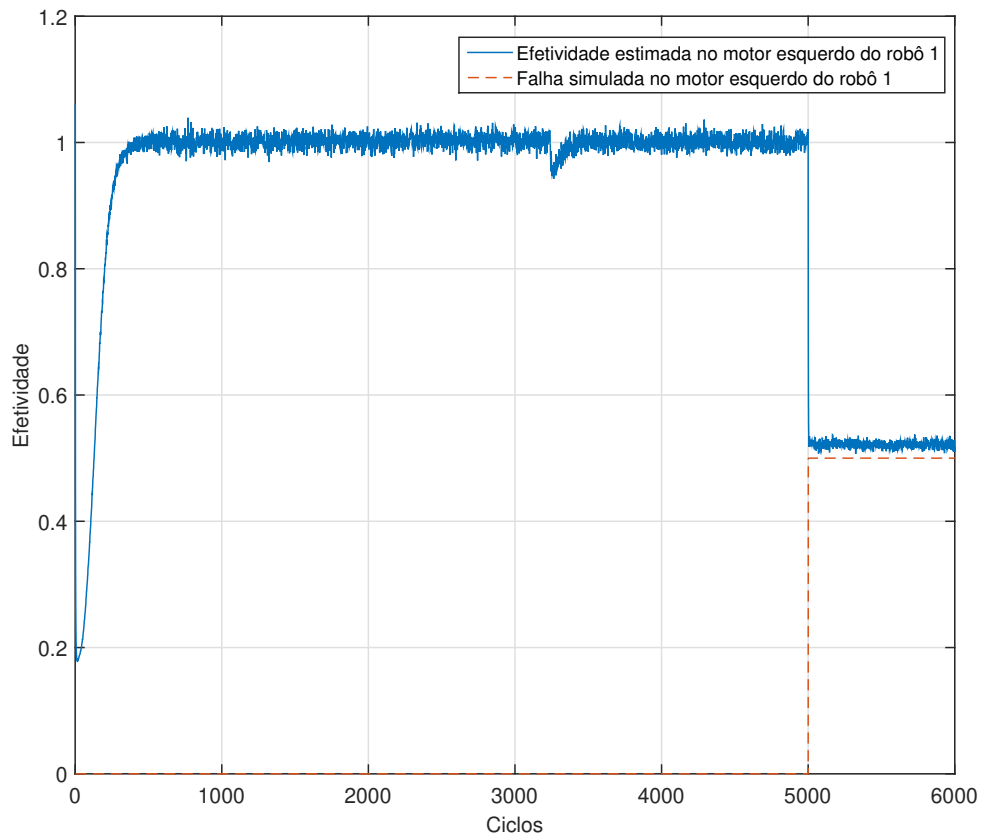
(5.5)

$$\phi_2 = \arctan 0,9 \sin 0.1t / 0,9 \cos 0.1t$$

onde T_1 , T_2 e T_3 são as trajetórias dos três robôs respectivamente. Os ângulos ϕ_1 , ϕ_2 e ϕ_3 são as orientações dos robôs 1, 2 e 3.

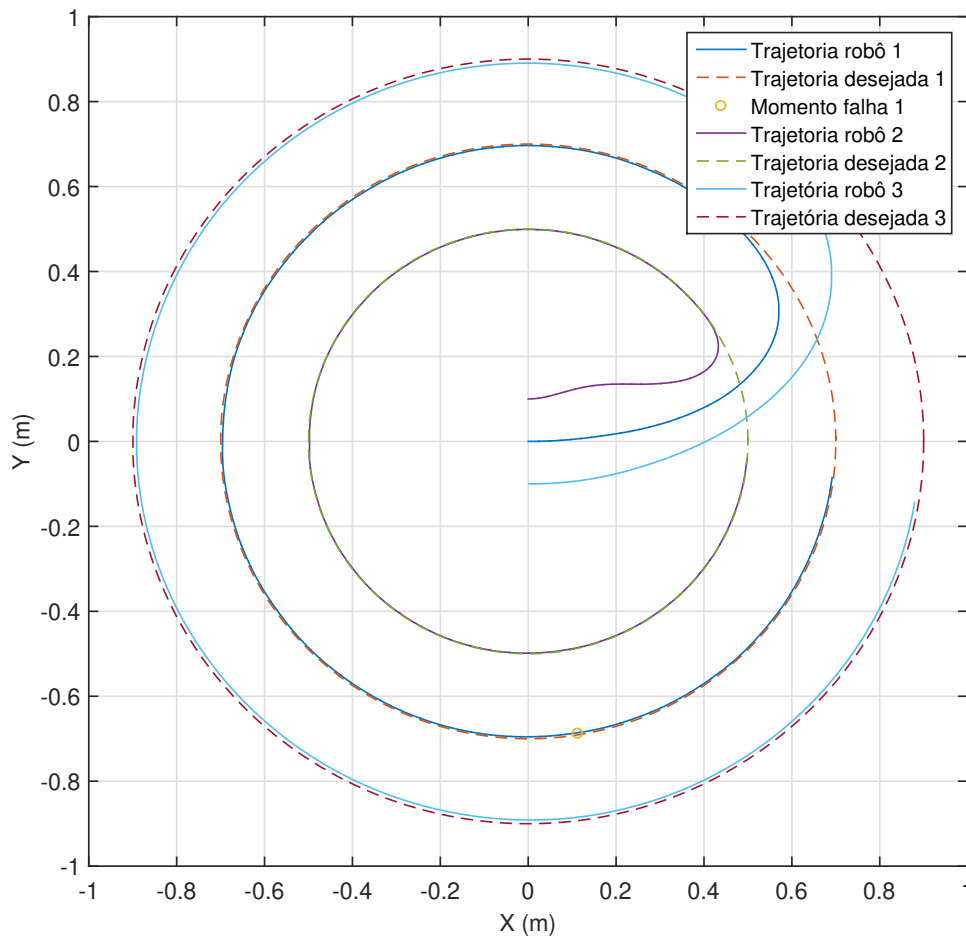
Na primeira situação, a simulação tem duração de 68 segundos e há uma ocorrência de falha parcial no motor esquerdo do robô 1 com 50% de perda de efetividade em 50 segundos. Os outros robôs não apresentam falhas em seus atuadores nesse teste. A Figura 5.11 demonstra a efetividade estimada durante o trajeto. É possível ver na figura que a efetividade do robô, representada pela linha contínua azul, converge para aproximadamente 0,5. Esse valor representa a falha simulada de 50% representada no gráfico pela linha segmentada laranja.

Figura 5.11 – Efetividade estimada do motor esquerdo do robô 1 e a falha simulada



Na Figura 5.12, os três robôs completam a tarefa determinada no tempo de 64,7371 segundos. O atraso apresentado na simulação para um único robô móvel também aparece no teste do Controle Cooperativo. Esse valor é de 1,9052 segundos. O gráfico mostra a trajetória dos 3 robôs pelas linhas contínuas e suas trajetórias de referência pelas linhas segmentadas. A trajetória do robô 1 representada pela linha contínua azul escuro, a trajetória do robô 2 designada pela linha contínua roxa e a trajetória do robô 3 caracterizada pela linha contínua azul claro. As trajetórias de referência dos robôs 1, 2 e 3 estão sendo representadas pelas linhas segmentadas vermelha, verde e vinho respectivamente. O ponto amarelo \circ representa o momento da falha no motor esquerdo do robô 1.

Figura 5.12 – Falha parcial de 50% no motor esquerdo do robô 1



Na segunda situação, a simulação teste tem duração também de 68 segundos e há ocorrência de uma falha severa no motor direito de 30% do robô 2 no instante de tempo de 50 segundos. Os robôs 1 e 3 não presenciam a ocorrência de falhas em seus motores nesse teste. A Figura 5.13 mostra a trajetória dos três robôs da simulação. As linhas contínuas azul escuro, laranja e azul claro representam as trajetórias dos robôs 1, 2 e 3, nessa ordem. Já as linhas tracejadas vermelha, vinho e roxa caracterizam as trajetórias de referências do robô 1, 2 e 3 respectivamente. O ponto \circ em verde é o momento da falha no motor direito do robô 2. Dessa imagem, é possível verificar que o robô 2, que teve a ocorrência de uma falha severa, não completou a trajetória devido a impossibilidade da reconfiguração das velocidades de referência. No entanto, os outros robôs sem falhas completaram normalmente o caminho devido no tempo determinado com uma margem de erro de aproximadamente 2 segundos. A Figura 5.14 apresenta a efetividade estimada do motor direito do robô 2 na situação 2. Esse gráfico mostra a ocor-

rência da falha severa aproximadamente em 50 segundos. É possível ver claramente na Figura 5.15 que o robô tem uma ordem de parada no instante de tempo do ciclo 5000. É o momento que ocorre a falha severa e as velocidades convergem para zero significando que o robô não é capaz de reconfiguração nem de completar a trajetória definida. Então, o robô é deixado para trás e os outros seguem com o percurso e cumprem o objetivo. Essa ação demonstra o algoritmo desenvolvido do Controle Tolerante a Falhas para Múltiplos Robôs Móveis em funcionamento. Nesse gráfico, a linha contínua em azul demonstra a velocidade linear do motor esquerdo e a linha contínua vermelha representa a velocidade linear do motor direito do robô 2.

Figura 5.13 – Falha severa no motor direito do segundo robô

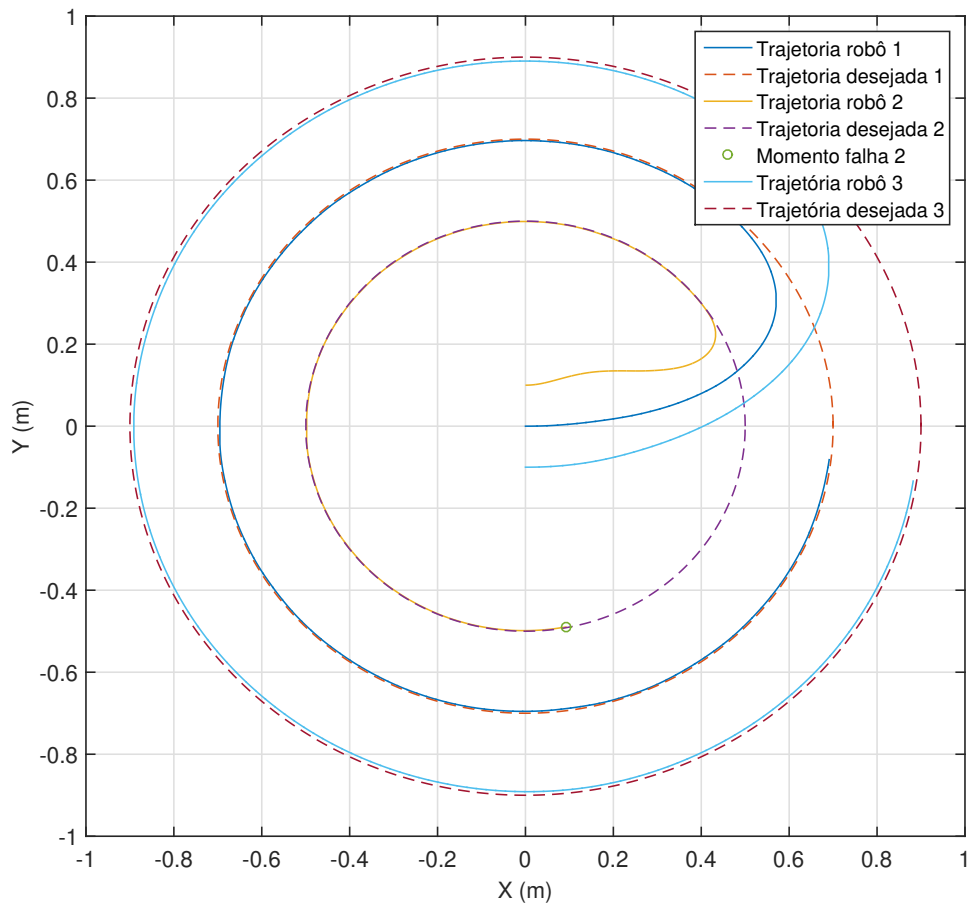


Figura 5.14 – Efetividade estimada do motor direito do robô 2

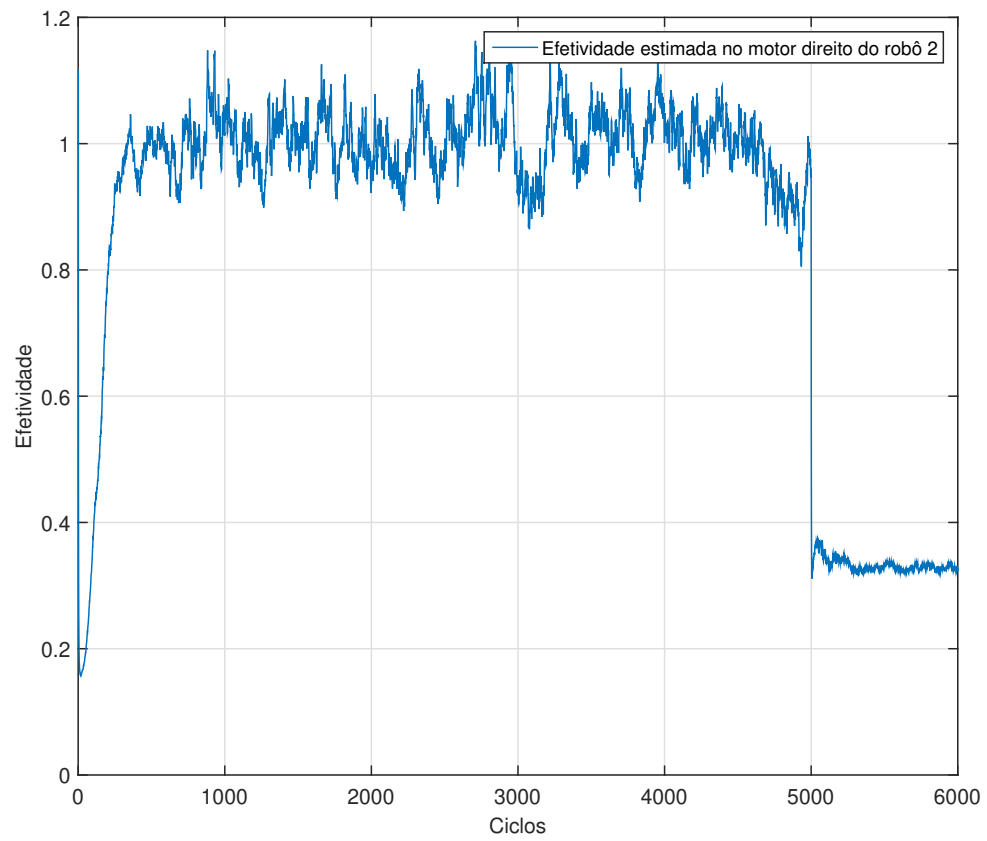
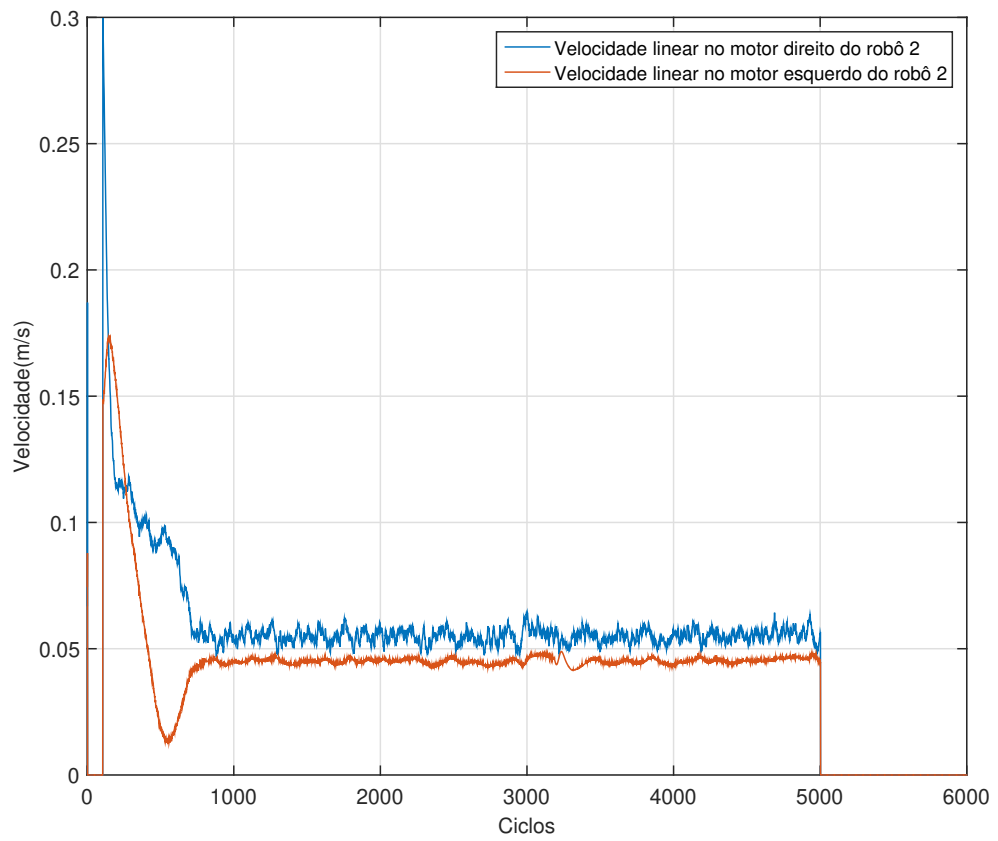


Figura 5.15 – Velocidade linear do robô 2



6 CONCLUSÕES

Esse trabalho é proposto um método para o Controle de Trajetória Tolerante a Falhas parciais em atuadores para um robô móvel com duas rodas tracionadas traseira e uma roda livre dianteira. Além dessa proposta, um algoritmo foi desenvolvido para o Controle Cooperativo Tolerante a Falhas severas para dois ou mais robôs móveis. Foram utilizadas para testes, simulações em que os robôs devem completar uma determinada trajetória em um instante de tempo definido em um ambiente livre de obstáculos.

Primeiramente, foram feitas as modelagens do robô móvel e do modelo cinemático. Após o modelo do robô utilizado no trabalho ser escolhido, a utilização de controlador PID foi fundamental para garantir velocidades de referência na saída dos motores através de uma entrada em tensão aplicada. O método utilizado para encontrar os ganhos K_p , K_i e K_d do controlador foi o de Ziegler-Nichols de um controlador PID. Para validar o controlador, um teste de resposta a um sinal degrau de entrada foi feito em simulação. Como resultado, o controlador obteve estabilização do sinal em aproximadamente 0,6 segundos e um máximo de oscilação na amplitude de 25%.

Em um segundo momento, um Controlador Fuzzy foi projetado para o controle de um trajetória predeterminada. Duas variáveis de entrada e duas de saída são consideradas. Sendo elas, o erros angulares e lineares como entradas e velocidades lineares e angulares como saídas. Para o conjunto de regras, o método usado foi o de Mamdani. Uma Trajetória circular de raio 1 metro foi definida para que o robô completasse o caminho em 62,8319 segundos e como resultado, o robô conseguiu completar a trajetória com um atraso de 1,9052, ou seja, um erro de aproximadamente 0,3% do tempo total.

Em um momento após, um Filtro de Kalman foi utilizado para estimar as velocidades angulares dos motores do robô móvel e a perda de efetividade do motor. O Filtro foi capaz de suavizar o sinal perante ruídos aleatórios inseridos na simulação ao estimar os estados.

No quarto momento, após a falha ser inserida no trabalho, percebeu-se a necessidade de implementar um reconfigurador de velocidades para trabalhar em conjunto com o Controlador Fuzzy visto que o mesmo sozinho não conseguiu fazer com que o robô completasse a trajetória na ocorrência de uma falha parcial em um de seus motores. Esse reconfigurador assegura as velocidades de referência desejadas nas saídas dos motores. Isso só é possível com uma limitação da velocidade máxima dos motores, ou seja, eles não trabalham com potência máxima em todo o percurso. Essa limitação é necessária pois a restrição física dos motores considerada.

Finalmente, um algoritmo de ações para a ocorrência de falhas severas nos motores dos robôs foi desenvolvido. Essa etapa é chamada de Controle Cooperativo Tolerante a Falhas. O método implementado atua no desligamento dos robôs quando há uma ou mais ocorrências de falhas severas em um dos motores dos robôs. Essas falhas severas não permitiriam que o(s) robô(s) completasse(m) a trajetória predefinida no tempo estabelecido. Nesse trabalho, não foi considerado a alteração do tempo determinado para completar a trajetória quando um dos robôs tivessem falhas severas. Essa mudança possibilitaria a alteração das velocidades dos robôs sem falhas e com isso todos conseguiriam completar o percurso, porém, não no tempo estabelecido. É uma alternativa que depende do objetivo inicial do trabalho.

Para trabalhos futuros, testes experimentais devem ser realizados e a robustez às incertezas de modelo, distúrbios de atuação e ruídos de medição deverão ser verificadas. Além disso, obstáculos serão tratados como desafio para o controle cooperativo dos robôs. Outros métodos de controle de trajetória serão implementados para comparação e garantia do melhor resultado. A inserção de detecção e diagnóstico de falhas para todos os componentes presentes no robô é uma possibilidade a ser estudada. Essa área de detecção e diagnóstico de falhas para robôs móveis tem sido muito estudada nas últimas décadas. Esse é um problema ainda não resolvido e há uma infinidade de métodos e estratégias para serem implementadas e testadas. Nesse trabalho, foram consideradas apenas algumas.

REFERÊNCIAS

- BETOURNE, A.; CAMPION, G. Dynamic modelling and control design of a class of omnidirectional mobile robots. In: **Proceedings of IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 1996. v. 3, p. 2810–2815 vol.3.
- CAMPION, G.; BASTIN, G.; D'ANDREA-NOVEL, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. In: **[1993] Proceedings IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 1993. p. 462–469 vol.1.
- CARLSON, J.; MURPHY, R. R. Reliability analysis of mobile robots. In: **2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)**. [S.l.: s.n.], 2003. v. 1, p. 274–281 vol.1.
- CHAMSEDDINE, A.; AMOOZGAR, M. H.; ZHANG, Y. M. Experimental validation of fault detection and diagnosis for unmanned aerial vehicles. In: _____. **Handbook of Unmanned Aerial Vehicles**. Dordrecht: Springer Netherlands, 2015. p. 1123–1155.
- CHAMSEDDINE, A.; ZHANG, Y.; RABBATH, C. A. Trajectory planning and re-planning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles. In: **2012 American Control Conference (ACC)**. [S.l.: s.n.], 2012. p. 3291–3296.
- DING, S. **Model-based fault diagnosis techniques: Design schemes, algorithms, and tools**. [S.l.]: Springer-Verlag London, 2008.
- DIXON, W. E.; WALKER, I. D.; DAWSON, D. M. Fault detection for wheeled mobile robots with parametric uncertainty. In: **2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)**. [S.l.: s.n.], 2001. v. 2, p. 1245–1250 vol.2.
- FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback Control of Dynamic Systems**. 7th. ed. Upper Saddle River, NJ, USA: Pearson, 2014.
- GOEL, P. et al. Fault detection and identification in a mobile robot using multiple model estimation and neural network. In: **Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)**. [S.l.: s.n.], 2000. v. 3, p. 2302–2309 vol.3.
- IZADI, H. A.; GORDON, B. W.; ZHANG, Y. Hierarchical decentralized receding horizon control of multiple vehicles with communication failures. **IEEE Transactions on Aerospace and Electronic Systems**, v. 49, n. 2, p. 744–759, 2013.
- Ji, M.; SARKAR, N. Supervisory fault adaptive control of a mobile robot and its application in sensor-fault accommodation. **IEEE Transactions on Robotics**, v. 23, n. 1, p. 174–178, 2007.
- Jian Wang; Chun-Yi Su; Oya, M. Robust motion tracking control of partially nonholonomic mechanical systems. In: **IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004**. [S.l.: s.n.], 2004. v. 5, p. 4608–4613 Vol.5.
- KALMAN, R. A new approach to linear filtering and prediction problems. **Journal Of Basic Engineering**, v. 82, p. 33–45, 1960.

- KAMEL, M. A.; YU, X.; ZHANG, Y. Fault-tolerant cooperative control design of multiple wheeled mobile robots. **IEEE Transactions on Control Systems Technology**, PP, n. 99, p. 1–9, 2017.
- KECSKÉS, I.; ODRY, P. Fuzzy route control of dynamic model of four-wheeled mobile robot. In: **2012 4th IEEE International Symposium on Logistics and Industrial Informatics**. [S.l.: s.n.], 2012. p. 215–220.
- KLANCAR, G. et al. **Wheeled Mobile Robotics: from fundamentals towards autonomous systems**. [S.l.]: Butterworth-Heinemann, 2017.
- LEE, C. C. Fuzzy logic in control systems: fuzzy logic controller. i. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 20, n. 2, p. 404–418, 1990.
- Luo, M. et al. Model-based fault diagnosis/prognosis for wheeled mobile robots: a review. In: **31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005**. [S.l.: s.n.], 2005. p. 6 pp.–.
- MACDIARMIND, P. R.; BART, J. J. **Reliability Toolkit: Commercial Practices Edition - A Practical Guide For Commercial Products And Military Systems Under Acquisition Reform**. [S.l.]: Reliability Analysis Center, 1998.
- MAMDANI, E. H. Application of fuzzy algorithms for control of simple dynamic plant. **Proceedings of the IEEE**, v. 121, p. 1585–1588, 1974.
- MEAD, R.; LONG, R.; WEINBERG, J. B. Fault-tolerant formations of mobile robots. In: **2009 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2009. p. 4805–4810.
- MEHTA, V. K.; MEHTA, R. **The Principles of Physics For XI**. [S.l.]: S. Chand School, 2018.
- MICHIGAN, U. **DC Motor Speed: System Modeling**. 2012. <<http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SystemModeling>>. [Online; acessado em 04/08/2019].
- ORIOLO, G.; LUCA, A. D.; VENDITTELLI, M. Wmr control via dynamic feedback linearization: design, implementation, and experimental validation. **IEEE Transactions on Control Systems Technology**, v. 10, n. 6, p. 835–852, 2002.
- ROTONDO, D. Background on fault tolerant control. In: _____. **Advances in Gain-Scheduling and Fault Tolerant Control Techniques**. Cham: Springer International Publishing, 2018. p. 129–145.
- Roumeliotis, S. I.; Sukhatme, G. S.; Bekey, G. A. Fault detection and identification in a mobile robot using multiple-model estimation. In: **Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)**. [S.l.: s.n.], 1998. v. 3, p. 2223–2228 vol.3.
- SABER, R. O.; MURRAY, R. M. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. In: **42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)**. [S.l.: s.n.], 2003. v. 2, p. 2022–2028 Vol.2.

SHABANIAN, M.; MONTAZERI, M. A neuro-fuzzy online fault detection and diagnosis algorithm for nonlinear and dynamic systems. **International Journal of Control, Automation and Systems**, v. 9, n. 4, p. 665, 2011.

SIEGWART, R.; NOURBAKHSI, I. R. Introduction to autonomous mobile robots. In: _____. **Introduction to Autonomous Mobile Robot**. [S.l.]: MIT Press, 2004. p. 30–41.

SKOUNDRIANOS, E. N.; TZAFESTAS, S. G. Finding fault - fault diagnosis on the wheels of a mobile robot using local model neural networks. **IEEE Robotics Automation Magazine**, v. 11, n. 3, p. 83–90, 2004.

ÅSTRÖM, K. J.; HÄGGLUND, T. **Advanced pid control**. [S.l.]: ISA-The Instrumentation, Systems, and Automation Society, 2006.

SUGENO, M. An introductory survey of fuzzy control. **Information Sciences**, v. 36, n. 1, p. 59 – 83, 1985.

TAKAGI, T.; SUGENO, M. Derivation of fuzzy control rules from human operator's control action. **IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis**, p. 55–60, 1983.

THUMATI, B. T.; DIERKS, T.; SARANGAPANI, J. A model-based fault tolerant control design for nonholonomic mobile robots in formation. **The Journal of Defense Modeling and Simulation**, v. 9, n. 1, p. 17–31, 2012.

UNBEHAUEN, H. D. **Control Systems, Robotics and Automation - Volume 2**. 2. ed. [S.l.]: EOLSS Publications, 2009.

VELASCO-VILLA, M.; ALVAREZ-AGUIRRE, A.; RIVERA-ZAGO, G. Discrete-time control of an omnidirectional mobile robot subject to transport delay. In: **2007 American Control Conference**. [S.l.: s.n.], 2007. p. 2171–2176.

VERMA, V. et al. Real-time fault diagnosis [robot fault diagnosis]. **IEEE Robotics & Automation Magazine**, v. 11, n. 2, p. 55–66, 2004.

WEIBING, Z. et al. Research on motion control algorithm for mirosot soccer robot. **2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)**, p. 437–440, 2009.

WELCH, G.; BISHOP, G. An introduction to the kalman filter. **Proc. SIGGRAPH Course**, v. 8, p. 1–16, 2001.

XU, Q. et al. Fault tolerant formations control of uavs subject to permanent and intermittent faults. **Journal of Intelligent & Robotic Systems**, v. 73, n. 1, p. 589–602, Jan 2014. ISSN 1573-0409.

YANG, H.; JIANG, B.; ZHANG, Y. Fault-tolerant shortest connection topology design for formation control. **International Journal of Control, Automation and Systems**, v. 12, n. 1, p. 29–36, 2014.

ZADEH, L. Fuzzy sets. **Information and Control**, v. 8, n. 3, p. 338 – 353, 1965. ISSN 0019-9958.

ZADEH, L. A. Fuzzy logic. **Computer**, v. 21, n. 4, p. 83–93, 1988.

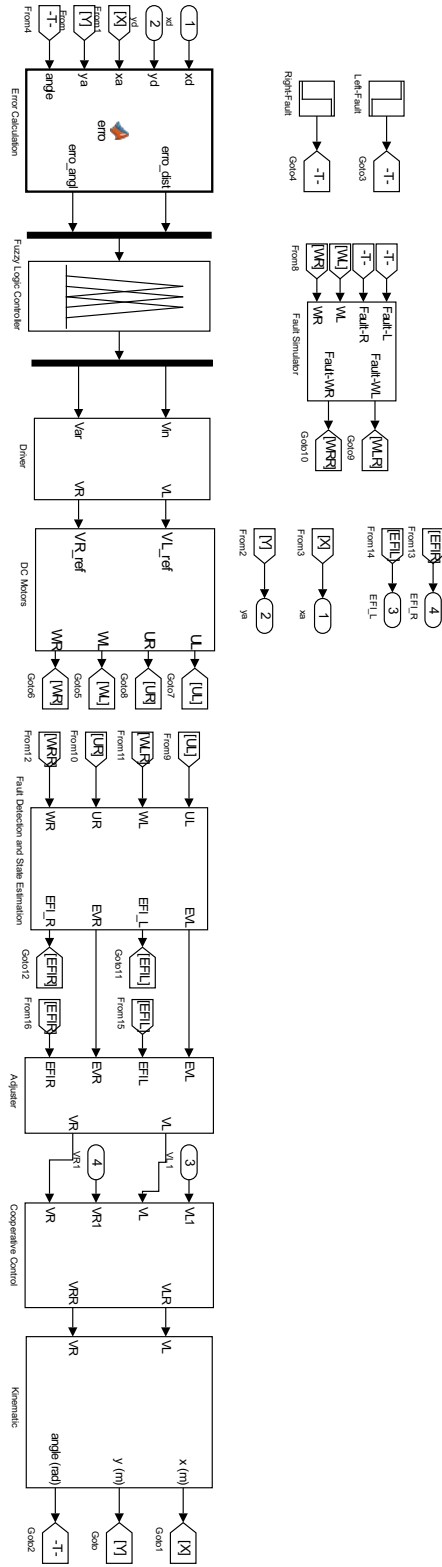
ZANARDELLI, W. G. et al. Wavelet-based methods for the prognosis of mechanical and electrical failures in electric motors. **Mechanical Systems and Signal Processing**, v. 19, n. 2, p. 411 – 426, 2005. ISSN 0888-3270.

ZHENG, H.; LI, Z.; CHEN, X. Gear fault diagnosis based on continuous wavelet transform. **Mechanical Systems and Signal Processing**, v. 16, n. 2, p. 447 – 457, 2002.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum setting for automatic controllers. **Trans. ASME**, v. 64, p. 759–768, 1942.

ZIXING, C.; HANGEN, H.; HONG, C. Some issues for mobile robot navigation under unknown environments. **Control and Decision**, v. 17, n. 4, p. 385–390, 2002.

A CONTROLE TOLERANTE A FALHAS



B CONTROLE COOPERATIVO TOLERANTE A FALHAS

