

UNIVERSIDADE DE SÃO PAULO-USP
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
CURSO ENGENHARIA ELÉTRICA - ÂNFASE EM ELETRÔNICA

Murilo da Silva Furlan

**Um Estudo Sobre a Performance de
Controladores PID e SMC em Robôs
Omnidirecionais com Mais de Uma
Malha**

São Carlos
2018

Murilo da Silva Furlan

**Um Estudo Sobre a Performance de
Controladores PID e SMC em Robôs
Omnidirecionais com Mais de Uma
Malha**

Trabalho de Conclusão de Curso apresentada ao Curso Engenharia Elétrica - Ênfase em Eletrônica da Escola de Engenharia de São Carlos como parte dos requisitos para a obtenção do título de Engenheiro Eletricista.

Área de concentração: Robótica

Orientador: Prof. Dr. Ivan Nunes da Silva

São Carlos
2018

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

Furlan, Murilo
FF985u Um Estudo Sobre a Performance de Controladores PID
e SMC em Robôs Omnidirecionais com Mais de Uma Malha /
Murilo Furlan; orientador Ivan Nunes . São Carlos,
2018.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2018.

1. Controladores. 2. Robótica. 3. PID. 4. Simulink.
5. Sliding Mode. 6. Malha Dupla. I. Título.

FOLHA DE APROVAÇÃO

Nome: Murilo da Silva Furlan

Título: "Um estudo sobre a performance de controladores PID e SMC em robôs omnidirecionais com mais de uma malha"

Trabalho de Conclusão de Curso defendido e aprovado
em 03/12/2018,

com NOTA 9,0 (Nove, zero), pela Comissão Julgadora:

Prof. Associado Ivan Nunes da Silva - Orientador - SEL/EESC/USP

Mestre Rafael Guedes Lang - Doutorando - SEL/EESC/USP

Mestre Adam Henrique Moreira Pinto - Doutorando - ICMC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

*Este trabalho é dedicado a todos os Engenheiros,
de formação e coração,
que instigam, investigam, criam e fascinam*

Agradecimentos

Agradeço primeiramente ao meus pais, pelo apoio incondicional aos meus estudos e ao meu futuro profissional, além do indispensável apoio financeiro. Agradeço minha mãe Sandra pelo carinho, pela ternura e por trazer todo o conforto necessário para que pudesse estudar. Ao meu pai Fernando por, além de ser a pessoa em que mais me inspiro, é o Engenheiro Elétrico que mais me inspiro. Agradeço também ao resto dos meus familiares e principalmente ao meu afilhado Levi, por me trazer alegria e afeto.

À minha namorada Ana Laís, pelo amor incondicional, pelo afeto e também por ser um exemplo como Engenheira. Me companhando desde o início a graduação, em momentos de estudos, trabalhos e descontração e me apoiando, sobre tudo, nos maus momentos.

Ao meu orientador Ivan Nunes, por me permitir desenvolver esse trabalho. Ao Warthog em geral, por me mostrar engenharia na prática e me certificar que esse é o meu caminho. Um agradecimento especial ao PMO que foi onde encontrei minha área na engenharia. Não posso deixar de agradecer pessoas fundamentais para esse trabalho como o Adam Pinto, o Nícolas Rosa, o Rafael Magossi, o Rafael Lang e o Guilherme Caixeta, que de alguma forma me deram conhecimento e apoio. Alguns deles, não só neste trabalho, mas também na vida acadêmica.

Agradeço aos meus amigos que moraram comigo, especialmente ao João Mário e ao Nicolau, que me auxiliaram durante a graduação e, embora sejamos tão diferentes, nos tornamos um trio inseparável. Agradeço ao Bacon, por seu apoio a minha vida profissional, por ser o centro das minhas maiores amizades e, principalmente, por me ajudar a me tornar quem sou. Além disso, agradeço a todos eles pelos vitais momentos de descontração.

Pro fim, agradeço a Universidade de São Paulo, por me prover todos esse conhecimento e a todos os seus funcionários, sejam professores, técnicos ou com qualquer outro cargo, que tornaram possível esse aprendizado.

Resumo

FURLAN, M. S. F **Um Estudo Sobre a Performance de Controladores PID e SMC em Robôs Omnidirecionais com Mais de Uma Malha.** 47 p. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2018.

Os sistemas de controle tem se mostrado indispensável para o campo da robótica. Com diversos estudos nessa área, muitas arquiteturas de controle e formas de ajustá-las foram desenvolvidas. No entanto, com muitas maneiras de fazer um controlador, surgiu a necessidade de compará-las e escolher um controlador melhor para a aplicação desejada. O objetivo deste trabalho é implementar um SMC e comparar com quatro controladores PID usados no Warthog Robotics em um sistema com duas malhas, através de um método objetivo de comparação. Foram utilizadas métricas como overshoot, rise time, ITAE e ISE, implementadas no MatLab. Um dos resultados mais importantes, foi descobrir que o controlador com filtro derivativo tem o melhor desempenho entre eles, para essa aplicação. Foi possível concluir que a metodologia utilizada para comparar esses controladores funcionou bem e pode ser replicada.

Palavras-chave: Controladores, Robótica, PID, Simulink, Sliding Mode, Malha Dupla.

Abstract

FURLAN, M. S. F **A Study About the Performance of PID and SMC Controllers in Omnidirectional Robots with More Than One Loop.** 47 p. Final Paper – São Carlos School of Engineering, University of São Paulo, 2018.

Control systems have proved indispensable for the field of robotics. With many studies in this area, a lot of control architectures and ways of adjusting them were developed. However, with many ways to make a controller, arose the need to compare them and choose a better controller for the desired application. The objective of this work is to implement a SMC and compare with four PID controllers used in Warthog Robotics in a double loop system, through an objective comparison method. Metrics such as overshoot, rise time, ITAE and ISE, implemented in MatLab were used. One of the most important results was to find that the derivative filter controller has the best performance among them for this application. It was possible to conclude that the methodology used to compare these controllers worked well and could be replicated.

Keywords: Controller, Robotics, PID, Simulink, Sliding Mode, double loop.

Lista de ilustrações

| | |
|--|----|
| Fig. 1.1 Robôs SSL do Warthog Robotics | 22 |
| Fig. 2.1 Exemplo de do erro de um SMC | 26 |
| Fig. 2.2 Velocidade do SSL | 26 |
| Fig. 2.3 Diagrama de Blocos com a Malha Dupla | 27 |
| Fig. 3.1 Métricas Básicas | 30 |
| Fig. 4.1 Simuink do SMC. | 34 |
| Fig. 4.2 Posição Global do SMC em Malha Dupla | 35 |
| Fig. 4.3 Posição Global do SMC em Malha Simples | 35 |
| Fig. 4.4 Erros em ω do SMC em Malha Dupla | 35 |
| Fig. 4.5 Erros em ω do SMC em Malha Simples | 36 |
| Fig. 4.6 Posição Global: (Esquerda) Caminho Completo e (Direita) com zoom. | 36 |
| Fig. 4.7 Erros em ω | 37 |
| Fig. 4.8 Erros em ω com zoom | 37 |
| Fig. 4.9 ITAE com Mlaha Dupla | 39 |
| Fig. 4.10 Posição Global Apenas com o Controlador de Motor | 40 |
| Fig. 4.11 Média de Erros entre os Motores com zoom | 40 |

Lista de tabelas

| | | |
|------------|--|----|
| Tabela 2.1 | Constantes ZN Malha Fechada | 24 |
| Tabela 2.2 | Constantes do PID | 28 |
| Tabela 4.1 | ITAE e ISE para Diferentes Tensões em SMC em Malha Dupla | 34 |
| Tabela 4.2 | ITAE em Malha Dupla | 38 |
| Tabela 4.3 | ISE em Malha Dupla | 38 |
| Tabela 4.4 | ITAE e ISE para o Controlador de Motor | 41 |

Lista de siglas

| | |
|-------------|---|
| IA | Inteligência Artificial |
| IAE | Integral do Erro Absoluto |
| ISE | Integral do Erro Quadrático |
| ITAE | Integral do Erro Absoluto vezes o Tempo |
| ITSE | Integral do Erro Quadrático vezes o Tempo |
| NIPI | Índices de Performance Não Intrusivos |
| PID | Controlador Proporcional Integrativo Derivativo |
| SSL | Small Size League |
| SMC | Controlador Sliding Mode |
| WR | Warthog Robotics |
| VSS | Sistema de Estrutura Variável |
| ZN | Ziegler-Nichols |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 21 |
| 2 | Robôs e Controladores | 23 |
| 2.1 | Controladores | 23 |
| 2.1.1 | PID | 23 |
| 2.1.2 | Sliding Mode | 24 |
| 2.2 | Robô | 25 |
| 3 | Métricas de Performance de Controladores | 29 |
| 3.1 | Métricas Básicas | 29 |
| 3.1.1 | <i>Rise Time</i> | 29 |
| 3.1.2 | <i>Overshoot</i> | 30 |
| 3.1.3 | <i>Settling Time</i> | 30 |
| 3.2 | Índices Integrais | 30 |
| 3.2.1 | IAE e ISE | 31 |
| 3.2.2 | ITAE e ITSE | 31 |
| 3.3 | Índices de Mínima Variância | 31 |
| 4 | Resultados | 33 |
| 4.1 | Implementação do Sliding Mode | 33 |
| 4.2 | Malha Dupla | 36 |
| 4.3 | Malha Simples do Motor | 39 |
| 5 | Conclusão e Diretivas Futuras | 43 |
| | Referências | 45 |

CAPÍTULO 1

Introdução

A RoboCup é uma das competições de tecnologia mais importantes do mundo, reunindo pessoas de vários países, para competir entre si e desenvolver conteúdo científico na área de robótica como um todo. Uma de suas categorias de futebol é a Small Size League (SSL), onde um time com seis robôs autônomos joga contra outro. Assim como em quase todos os tipos de robô, é essencial que os modelos de SSL implementem algum sistema de controle para sua posição e velocidade [1]. Uma das equipes que participa da RoboCup é o Warthog Robotics (WR), um grupo de pesquisa brasileiro da USP, que vem estudando maneiras de melhorar os controladores de seus robôs SSL.

O grupo de pesquisa WR, tem diversas frentes na robótica. Além dos robôs SSL (figura 1.1) o grupo também desenvolve robôs da categoria de futebol Very Small Size, que são robôs menores que os Small Size, autônomos e diferenciais, que jogam em times de três contra três. O grupo também desenvolve robôs para competições de combate, que requerem um piloto. Ultimamente, os membros do WR vêm desenvolvendo técnicas de gestão de pessoas, cujo um dos resultados, foi a formação de uma área de desenvolvimento em controle, que desenvolveu boa parte do conhecimento usado nessa pesquisa.

Com o avanço da robótica, muitas arquiteturas de controladores foram desenvolvidas, resultando em várias formas de controlar sistemas. Devido a essa diversidade, torna-se necessário comparar esses tipos de controladores e encontrar melhores opções de arquitetura e ajustes para cada aplicação. A partir disso, algumas métricas foram desenvolvidas para obter uma comparação numérica em diferentes aspectos, como a rapidez com que o controlador converge, a precisão da convergência, a quantidade de erros antes da convergência, entre outros.

O objetivo deste trabalho é propor uma metodologia de controle diferente das já utilizadas pelo grupo. Após implementado em simulação, acrescentar aos outros quatro controladores já usados para controlar as velocidades dos motores do robô WR SSL, e compará-los através de uma metodologia objetiva. Com este estudo, o grupo de pesquisa será capaz de escolher o melhor controlador entre esses cinco, e se concentrar em melhorar o escolhido e, portanto, obter um robô mais preciso durante o jogo.

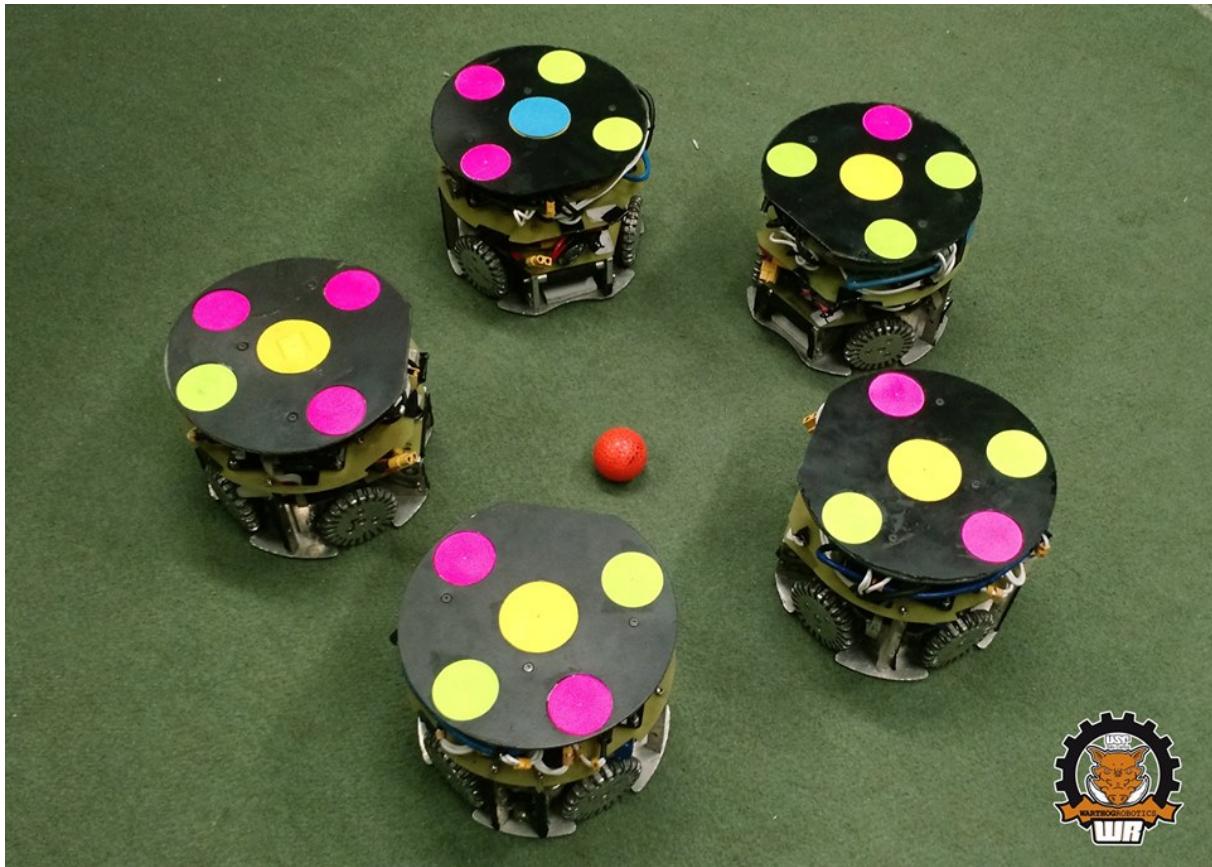


Fig. 1.1: Robôs SSL do Warthog Robotics

Durante a análise, algumas informações sobre esses cinco controladores, como a formulação matemática, são necessárias para uma melhor compreensão deste trabalho, e essa informação é apresentada na seção 2, bem como alguns conhecimentos sobre o robô do WR. É neste capítulo que o novo controlador é apresentado e descrito. Na metodologia para comparar o desempenho dessas diferentes arquiteturas, serão utilizadas algumas métricas mostradas na seção 3. No capítulo 4 será apresentado a implementação do controlador e também a aplicação das métricas de comparação para, por fim, na seção 5, concluir qual é o melhor dentre estes controladores para ser aprimorado e, consequentemente, obter um robô mais preciso e elevar a chance do grupo WR de vencer suas partidas.

CAPÍTULO 2

Robôs e Controladores

2.1 Controladores

2.1.1 PID

A necessidade de controlar sistemas de forma mais precisa começou a surgir na década de 1920, pois os controles baseados em on-off já não atendiam as necessidades e foi quando os primeiros controladores PID (Proporcional Integrativo Derivativo) começaram a ser desenvolvidos por Sperry e Minorski [2]. Posteriormente, na década de 40, Ziegler e Nichols desenvolveram um método de tunagem desses parâmetros de controle para optimizar sua velocidade de convergência e precisão. Essa metodologia ficou conhecida como método de Ziegler-Nichols (ZN) [3] e será mais abordado abaixo. Vale ressaltar também que na década de 80 e 90, com o avanço da computação, começaram a surgir métodos de auto-tuning para optimizar ainda mais os coeficientes PID, sendo um dos seus principais precursores, Astrom [4].

A ideia desse tipo de controlador é multiplicar o erro entre a entrada e a saída do sistema, sua derivada e sua integral em função do tempo, por constantes denominadas K_p (proporcional ao erro), K_d (proporcional a derivada do erro) e K_i (proporcional sua integral). Dessa forma, se obtêm a equação 2.1 no domínio do tempo e a equação 2.2 no domínio da frequência, onde $e(t)$ ou $E(s)$ são os erros e $u(t)$ ou $U(s)$ são as saídas do controlador. A variável τ na equação 2.1 é apenas uma auxiliar na integração, para o resultado dessa operação fornecer um função no tempo.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.1)$$

$$U(s) = K_p E(s) + K_i \frac{1}{s} E(s) + s K_d E(s) \quad (2.2)$$

Com o crescimento de sistemas micro-controlados, surgiu a necessidade de discretizar essas funções para controlar sistemas de forma digital. A equação 2.3 representa o PID discreto, onde os os coeficientes q_0 , q_1 e q_2 são os que devem ser tunados. Os termos $e[k]$

Tabela 2.1: Constantes ZN Malha Fechada

| Controlador | K_p | T_i | T_d |
|-------------|------------|-------------------|-----------------|
| P | $0,5 K_p$ | - | - |
| PI | $0,45 K_p$ | $\frac{P_u}{1,2}$ | - |
| PID | $0,6 K_p$ | $\frac{P_u}{2}$ | $\frac{P_u}{8}$ |

$u[k]$ também são o erro e a saída, respectivamente, porém agora no tempo discreto, onde k é o multiplicador do tempo de amostragem T [5].

Os termos q_0 , q_1 e q_2 vem das equações 2.4, onde T_d e T_i são, respectivamente, análogos ao K_d e K_i e G_{anho} é o ganho. A partir dessa equação é possível obter sua função de transferência, dessa vez através da Transformada Z, mostrada na equação 2.5.

$$u[k] = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (2.3)$$

$$\begin{aligned} q_0 &= G_{anho}(1 + \frac{T_d}{T}) \\ q_1 &= -G_{anho}(1 + 2\frac{T_d}{T} - \frac{T}{T_i}) \\ q_2 &= G_{anho} \frac{T_d}{T} \end{aligned} \quad (2.4)$$

$$G_{PID} = q_0 \frac{(z - z_{01})(z - z_{02})}{z^2 - z} \quad (2.5)$$

Onde:

$$\begin{aligned} z_{01} + z_{02} &= -\frac{q_1}{q_0} \\ z_{01} z_{02} &= \frac{q_2}{q_0} \end{aligned} \quad (2.6)$$

O método de sintonia PID ZN, é baseado em encontrar uma resposta oscilatória do sistema aumentando o K_p , e a partir dai, utilizar o período de oscilação, P_U para determinar as constantes K_p , T_d e T_i com a tabela 2.1 [6,7]. As relações entre K_i , K_d e T_i , T_d podem ser encontradas na formula 2.7.

$$\begin{aligned} K_i &= \frac{K_p}{T_i} \\ K_d &= K_p T_d \end{aligned} \quad (2.7)$$

2.1.2 Sliding Mode

A topologia de Sliding Mode Controller (SMC) parte da ideia de se usar um controle com um Sistema de Estrutura Variável (VSS), isto é, permite-se que o controlador troque de estado a qualquer momento, selecionando um função dentre uma gama, dependendo de um parâmetro [8].

No Sliding Mode, esse parâmetro que determina a mudança de estado é chamado de Sliding Surface, e pode ser determinado pela equação 2.8. Para controle de motores, basicamente, ao definir a Sliding Surface, o motor passa a ligar para um lado quando

o sistema está abaixo desse parâmetro e para o outro lado quando ele está acima desse parâmetro [9, 10].

$$\sigma = \left(\frac{d}{dt} + p \right)^k e \quad (2.8)$$

A Sliding Surface, σ , depende de k , que define o número de derivadas do erro (e) que serão incluídas no sistema e de p , que é um parâmetro a ser tunado para otimizar o controle. Esse primeiro termo, que contém a derivada em função do tempo e p , é multiplicado pelo erro, que terá, basicamente, o objetivo de definir o sentido positivo ou negativo de σ , selecionando o modo de operação do motor. Dessa forma, definindo como u a saída do controlador, em função do tempo, e V_m a tensão máxima do motor de corrente contínua, a equação do controlador pode ser descrita como a 2.9.

Quanto mais alto for o parâmetro inteiro k , mais complexo e com maior gasto computacional ficará o sistema, já que este terá que calcular um número k de derivadas. Já p , define o quanto essas derivadas influenciarão na Sliding Surface, sendo que quanto maior o p for, mais o erro prevalecerá em relação às suas derivadas em função do tempo, para decidir qual será o sinal da Sliding Surface e, consequentemente, o lado que o motor deverá girar.

$$u = -\operatorname{sgn}(\sigma)V_m$$

ou seja

$$(2.9)$$

$$u = \begin{cases} -V_m, & \sigma > 0 \\ V_m, & \sigma < 0 \end{cases}$$

A resposta do controlador fica parecida com a da figura exemplo 2.1. É possível perceber que, quando ele atinge a velocidade requisitada, sua forma fica como uma onda dente de serra.

2.2 Robô

O robô SSL do WR é um robô omnidirecional com quatro motores e, consequentemente, quatro rodas. A Inteligência Artificial (IA), não embarcada, envia velocidades, em três eixos, ao robô: Y , é a velocidade para avançar (positivo) e para recuar (negativo); X , é a velocidade para ir para ambos os lados; e ω , é a velocidade para fazer o robô girar em seu próprio eixo. Estas três velocidades do corpo rígido tornam-se as quatro velocidades de cada motor através de uma multiplicação de matrizes expressa na Equação 2.10. Essa relação de velocidades pode ser observada na figura 2.2. Através de relações geométricas de mudança de bases, foi obtida a matriz 4×3 \mathbf{T} . Esta matriz, assim como o modelo da planta do SSL para simulações usado neste trabalho, foi desenvolvida por [11].

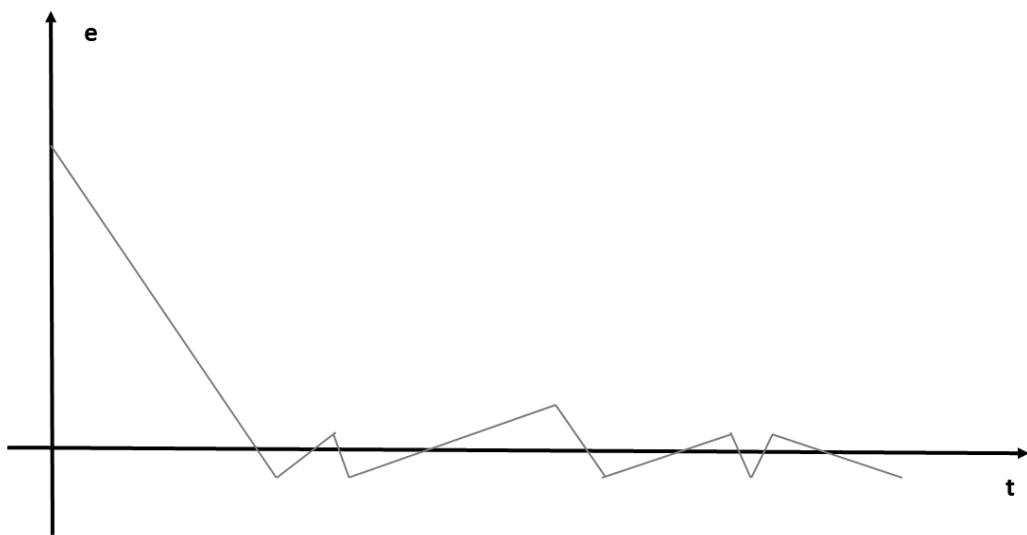


Fig. 2.1: Exemplo de do erro de um SMC

Fonte: próprio autor

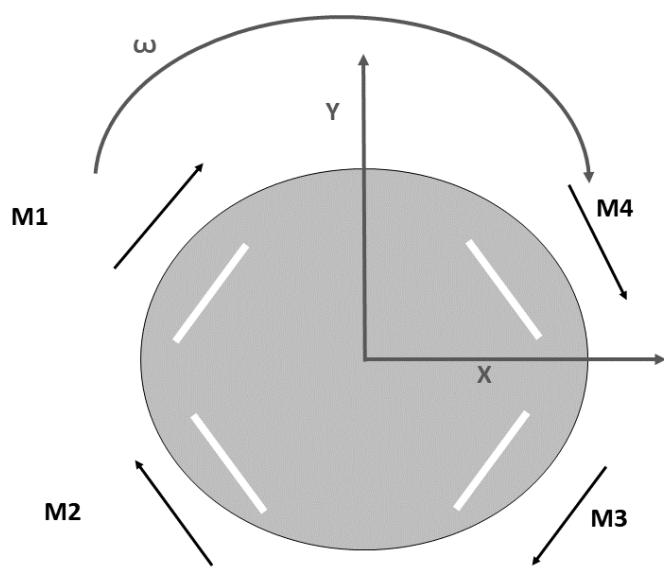


Fig. 2.2: Velocidade do SSL

Fonte: próprio autor

$$(T) \begin{pmatrix} X \\ Y \\ \omega \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} \quad (2.10)$$

Existem muitos controladores atuando no movimento do robô, mas, para este estudo, serão considerados apenas os controladores embarcados. Dois controladores são implementados no firmware: o controle da velocidade do corpo rígido (X , Y e ω) e o controle dos quatro motores (ω_1 , ω_2 , ω_3 e ω_4). O diagrama de blocos do Simulink, mostrado na Figura 2.3, representa o funcionamento do sistema. Os controladores comparados nesta pesquisa são apenas os que atuam nos motores e o controlador discreto do tipo PID, do corpo rígido, foi o mesmo durante todos os testes.

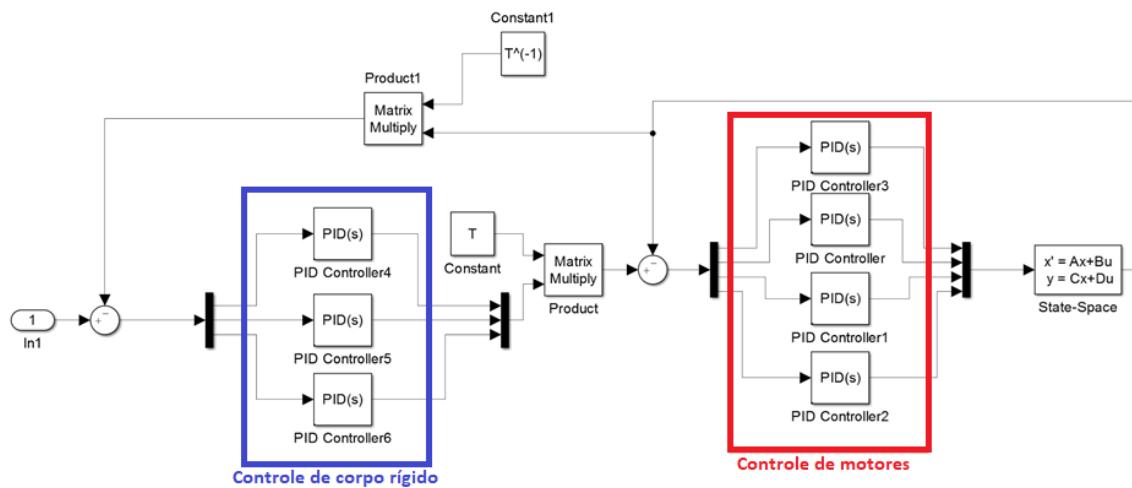


Fig. 2.3: Diagrama de Blocos com a Malha Dupla

Fonte: próprio autor

Quatro controles de motor já haviam sido implementados pelo grupo, sendo um deles um PID continuo, dois PID's discretos com diferentes tunagens e mais um PID discreto, porém com um filtro derivativo. Com mais um pólo derivativo, é possível filtrar as altas frequências, que geralmente são de erros aleatórios e desta forma, aumentar a performance do controlador. No entanto, ao adicionar este pólo, altera-se completamente a dinâmica dos controladores [12].

Para se obter o controlador com mais esse pólo, adiciona-se um filtro a ele através da equação 2.11, modificando o K_d , onde N , é a frequência de corte do filtro. A partir de 2.11 é possível obter as equações controlador, sendo 2.12 no tempo discretizado e 2.13 na frequência [13].

$$K_d = K_d \left(\frac{N}{s + N} \right) s \quad (2.11)$$

Tabela 2.2: Constantes do PID

| | K _p | K _d | K _i |
|--------------|----------------|-------------------------|----------------|
| PID Continúo | 0,184 | -2,259 10 ⁻³ | 0,701 |

| | q ₀ | q ₁ | q ₂ |
|----------------|----------------|-------------------------|-------------------------|
| PID Discreto 1 | 0,016 | -0,013 | -0,410 10 ⁻³ |
| PID Discreto 2 | 0,018 | -8,362 10 ⁻³ | 3,564 10 ⁻³ |

| | $\frac{a_1}{a_0}$ | $\frac{a_2}{a_0}$ | $\frac{b_0}{a_0}$ | $\frac{b_1}{a_0}$ | $\frac{b_2}{a_0}$ |
|-------------------------|-------------------|-------------------|-------------------|-------------------|------------------------|
| PID Discreto Com Filtro | 1,693 | -0,693 | 0,021 | -0,029 | 9,859 10 ⁻³ |

$$u[k] = \frac{a_1}{a_0}u(k-1) + \frac{a_2}{a_0}u(k-2) + \frac{b_0}{a_0}e(k) + \frac{b_1}{a_0}e(k-1) + \frac{b_2}{a_0}e(k-2) \quad (2.12)$$

$$G(z) = K_p + \frac{K_i T_s z}{z - 1} + \frac{K_d N(z - 1)}{(1 + NT_s)z - 1} \quad (2.13)$$

Nestes quatro controladores, as constantes PID foram obtidas pelo RLTOOL, do Matlab e algumas pelo método de Ziegler-Nichols [3]. Todas as constantes usadas para comparar os quatro controladores forma mantidas as mesmas durante todos os testes estão mostradas na tabela 2.2.

CAPÍTULO 3

Métricas de Performance de Controladores

Foi feita uma revisão sobre as métricas e índices que poderiam ser úteis para esta aplicação. Essas métricas e índices foram divididos, neste trabalho, em dois grupos: as métricas básicas, bem conhecidas na literatura, e os índices integrais. Por outro lado, todos os métodos de comparação aqui usados, são não intrusivos (NIPI) [14].

3.1 Métricas Básicas

As métricas básicas podem ser observadas com uma entrada na forma de *step* e, em seguida, analisadas graficamente. Podem ser chamadas de básicas porque elas não têm basicamente nenhum cálculo matemático envolvido. Todas as três métricas básicas apresentadas abaixo são exemplificadas na figura 3.1. Para um melhor entendimento será utilizado um exemplo de um controlador que precisa manter a tensão de saída em cinco volts.

3.1.1 *Rise Time*

Uma métrica para medir a rapidez com que um controlador atua é o *Rise Time*. Esta métrica é definida como o tempo entre o momento do *step* e a primeira vez que a saída alcança o valor requerido na entrada, ou seja, a primeira vez que o erro é zero. Quanto menor o tempo de subida, melhor o controlador. No exemplo mostrado na Figura 3.1 seu valor é meio segundo.

Muitas vezes, um baixo *Rise Time* está associado a uma resposta bastante oscilatória, e, portanto, apenas assegurar que essa métrica tenha um valor baixo não garante um controlador com ótimo desempenho.

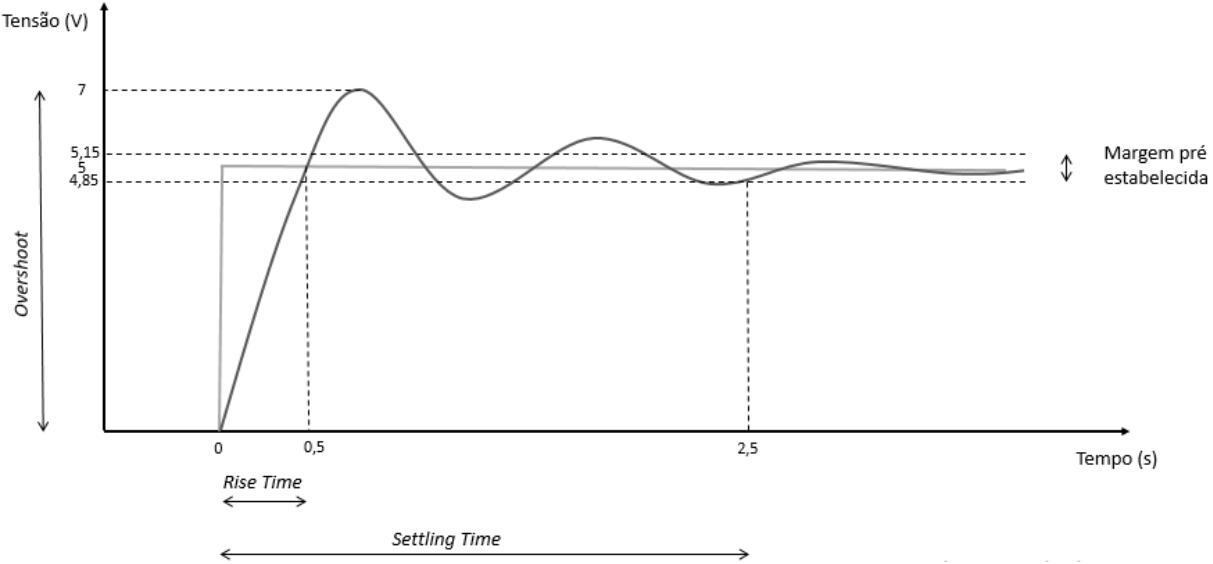


Fig. 3.1: Métricas Básicas

3.1.2 *Overshoot*

A métrica *Overshoot*, mostra quanto o primeiro erro impacta no sistema. É determinado por quanto o controlador extrapola da entrada necessária na primeira vez que o erro tenta convergir para zero e, geralmente, é medido em porcentagem. Quanto maior o *overshoot*, pior é o controlador. Na Figura 3.1 o *Overshoot* é de 140%.

Um alto *Overshoot* também está atrelado a um baixo *Rise Time*, visto que quanto mais rápido o controlador tenta convergir, mais ele extrapola a saída desejada. Um balanço entre estas duas métricas é necessário para aumentar a qualidade do controle, priorizando uma delas dependendo da aplicação.

3.1.3 *Settling Time*

O *Settling Time* analisa a rapidez com que o controle converge para um erro zero. Depois de estabelecida uma margem de erro, esta métrica é definida como o tempo entre o *step* e a última vez que o erro é maior que essa margem, em módulo. Usando uma borda de 3%, este sistema teve um *Settling Time* de dois segundos e meio.

3.2 Índices Integrais

Existem alguns índices que usam a integral do erro no tempo. Quatro deles foram escolhidos e são descritos abaixo [15], mas apenas dois deles foram usados para comparar os controladores. Para essas métricas, foi utilizado o erro em função do tempo expresso pela Equação 3.1, onde $r(t)$ é o sinal de referência e $y(t)$ é a saída da planta.

$$e(t) = r(t) - y(t) \quad (3.1)$$

3.2.1 IAE e ISE

IAE (Integral do Erro Absoluto) e ISE (Integral do Erro Quadrado) são índices obtidos das integrais apresentadas nas Equações 3.2, onde a entrada é um *step*. Como o ISE usa o quadrado do erro, o índice tende a crescer exponencialmente para um erro ligeiramente grande. Portanto, um pequeno ISE significa que o controlador não possui altos picos de erro [16].

Vale ressaltar que, na teoria, as integrais para calcular esses índices vão de zero a infinito, porém na prática, é impossível estender esse teste até infinito. Muitas vezes é estabelecido um tempo máximo, suficientemente grande para que o controlador obtenha uma certa convergência, e mantido o mesmo tempo para todos os controladores que cujos índices serão comparados.

$$\begin{aligned} IAE &= \int_0^{\infty} |e| dt \\ ISE &= \int_0^{\infty} e^2 dt \end{aligned} \tag{3.2}$$

3.2.2 ITAE e ITSE

ITAE (Integral do Erro Absoluto multiplicado pelo Tempo) e ITSE (Integral do Erro Quadrado multiplicado pelo Tempo) seguem a mesma ideia do IAE e do ISE. Mas a integral é multiplicada pelo tempo, portanto, quanto mais longe o erro é do tempo zero, mais esse erro contribui para o aumento do índice [16]. O índice é descrito pela Equação 3.3.

$$\begin{aligned} ITAE &= \int_0^{\infty} t |e| dt \\ ITSE &= \int_0^{\infty} t e^2 dt \end{aligned} \tag{3.3}$$

3.3 Índices de Mínima Variância

Na literatura existem muitos índices baseados no conceito de Variância Mínima, conceito de controle estocástico [17]. Em 1989, Harris propôs um índice que compara a variância mínima do controlador com sua variância real, e com isso é possível comparar o desempenho do controlador [15, 18, 19]. Este índice foi o primeiro a utilizar a teoria da variância mínima, e ficou conhecido como Índice de Harris. Um dos principais avanços desta métrica foi que ela não precisa de uma entrada específica e, por causa disso, o desempenho do controle pode ser analisado em tempo real, tornando-se importante para a aplicação industrial [20].

CAPÍTULO 4

Resultados

Inicialmente foi implementado um controlador com a topologia Sliding Mode na planta para aumentar o número de controladores a ser comparado e, dessa forma, poder concluir qual a melhor opção para esse sistema.

Com uma entrada para fazer o robô girar em seu próprio eixo em $\pi/2$ rad/s ($V_x = 0$, $V_y = 0$, $V_\omega = \pi/2$) por 16 segundos, foram realizados dois testes: um utilizando os dois controladores e um utilizando apenas o controlador do motor. Esta entrada foi escolhida porque produz um step no controlador de motores. Desta forma, foi possível comparar não apenas o melhor controlador para esta aplicação trabalhando sozinho, mas também o controlador que funcionou melhor com todo o sistema de loop duplo. Todos estes testes foram simulados através do MatLab e Simulink.

A metodologia proposta é, primeiro, comparar, utilizando duas métricas básicas, e analisar se os resultados palpáveis de comparação: o *Overshot* e o *Rise Time* para ver qual tem a melhor resposta no início da atuação do controlador. Já o *Settling Time* não precisa ser usado, uma vez que os índices integrais podem indicar qual controlador tem menos erros após algum tempo de atuação. Depois disso, usar os índices ITAE e ISE para fazer uma comparação numérica: ITAE para ver que o controlador tem menos erros após o tempo de estabilização e o ISE para ver que o controlador tem picos de erro maiores. Os índices de mínima variância não foram necessários, pois sua grande utilidade é em plantas reais e em tempo real, que não é o objetivo desse estudo no momento.

4.1 Implementação do Sliding Mode

Foi definido que nesse controlador SMC o parâmetro K , explicado na sessão sobre Sliding Mode, é igual a zero. Dessa forma, as derivadas do erro no tempo não influenciam na Sliding Surface e portanto o próprio erro será o σ . Assim sendo, o motor ligará para um lado quando o erro for negativo e para o outro quando, o mesmo, for positivo. Assim é

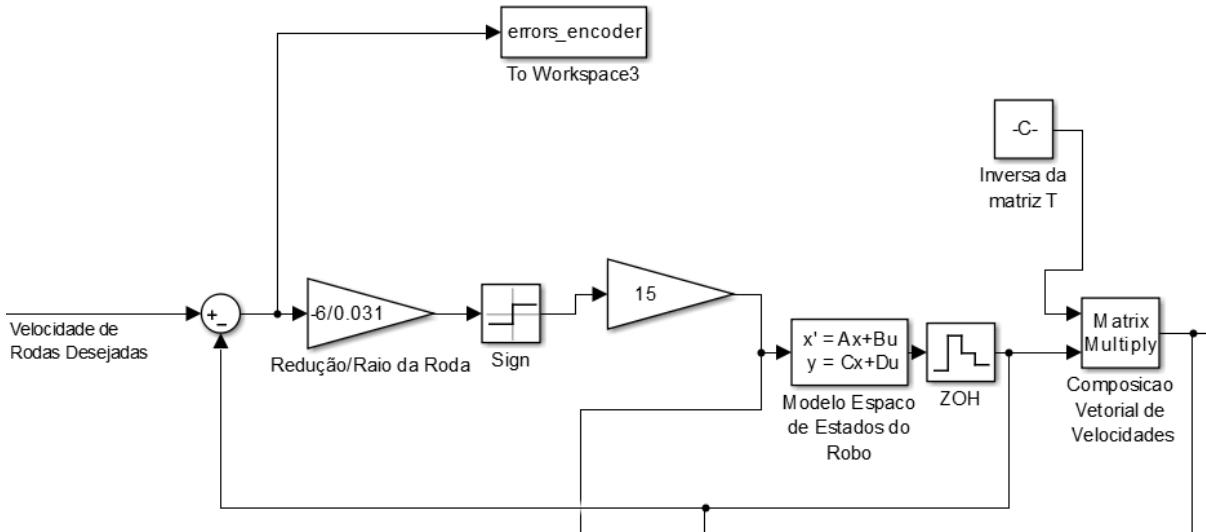


Fig. 4.1: Simuink do SMC.

Tabela 4.1: ITAE e ISE para Diferentes Tensões em SMC em Malha Dupla

| Tensão | 15 | 14,5 | 14 | 13,5 | 13 | 12,5 | 12 |
|--------|--------|--------|--------|--------|---------------|--------|--------|
| ITAE | 67,267 | 65,100 | 62,719 | 60,523 | 58,314 | 58,654 | 59,038 |
| ISE | 16,918 | 15,784 | 14,705 | 13,725 | 12,739 | 12,813 | 12,897 |

possível ver nas equações abaixo o modelo do controlador, sendo a tensão do motor 15V.

$$\begin{aligned} \sigma &= \left(\frac{d}{dt} + p \right)^0 e \\ \sigma &= e \end{aligned} \quad (4.1)$$

$$u = -\text{sgn}(e)15$$

ou seja

(4.2)

$$u = \begin{cases} -15, & e > 0 \\ 15, & e < 0 \end{cases}$$

A implementação foi feita através do Simulink, onde o bloco Sign inverte o sinal do erro e mantém o ganho unitário. Então, esse sinal é multiplicado por 15, para dar o valor de tensão do motor. Isso pode ser observado na figura 4.1.

Então, esse controlador, foi simulado em malha dupla. Após alguns testes, foi possível concluir que, se a tensão fosse diminuída alguns volts, robô simulado ficava com uma melhor dinâmica. Então foram comparados os ITAE's e os ISE's na tabela 4.1 e concluiu-se que, para operar com o SMC com K igual a 0, a melhor tensão que o motor deve operar é algo próximo de 13V.

Com essa tensão, foram obtidos o caminho global do robô em 16 segundos em ambas as condições: em malha dupla, figura 4.2 e em malha simples, figura 4.3. Além da

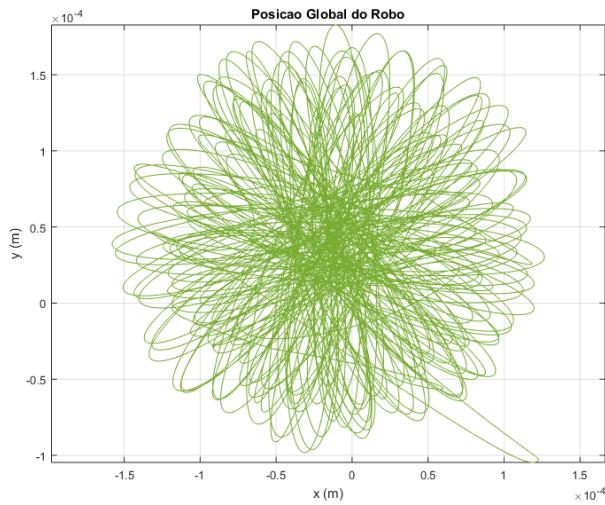


Fig. 4.2: Posição Global do SMC em Malha Dupla

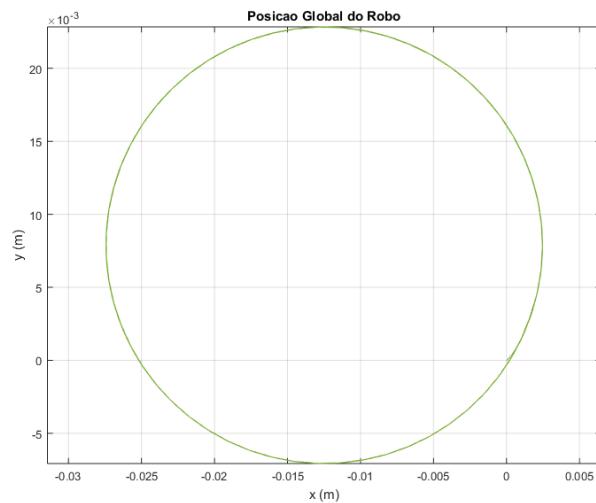


Fig. 4.3: Posição Global do SMC em Malha Simples

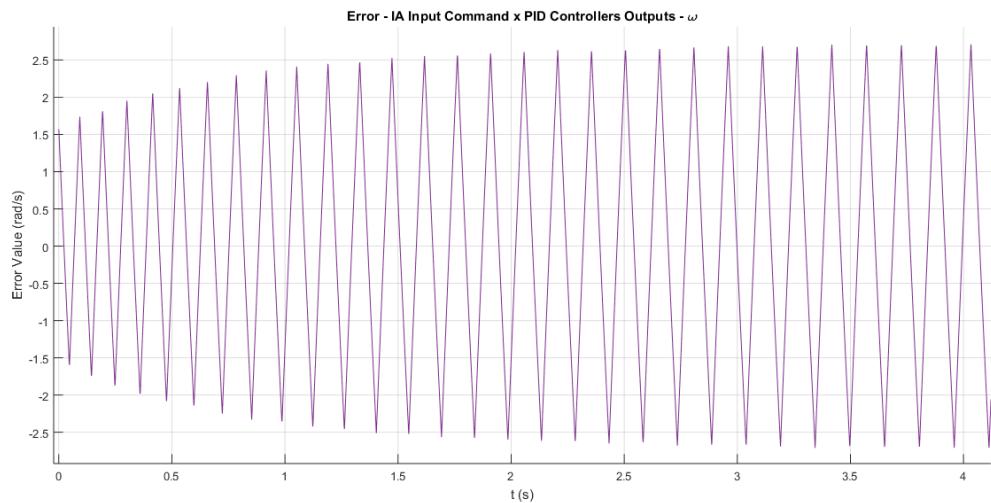


Fig. 4.4: Erros em ω do SMC em Malha Dupla

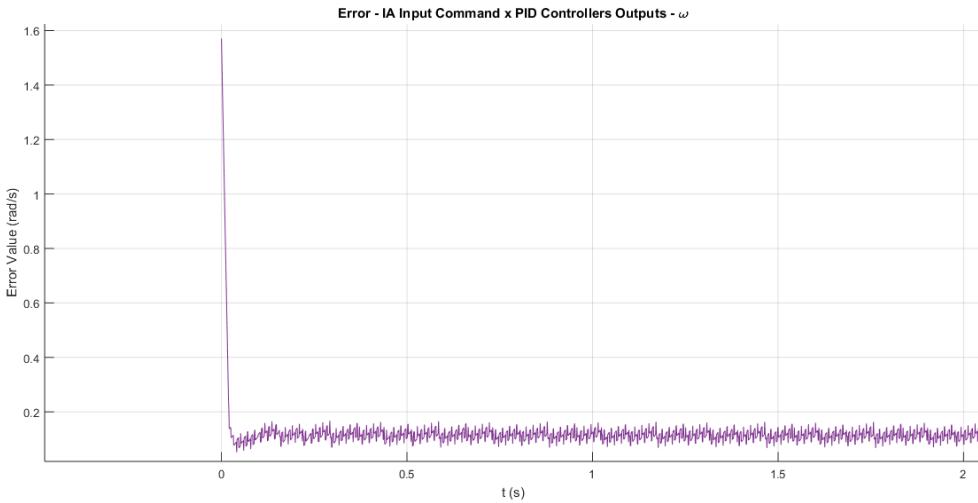


Fig. 4.5: Erros em ω do SMC em Malha Simples

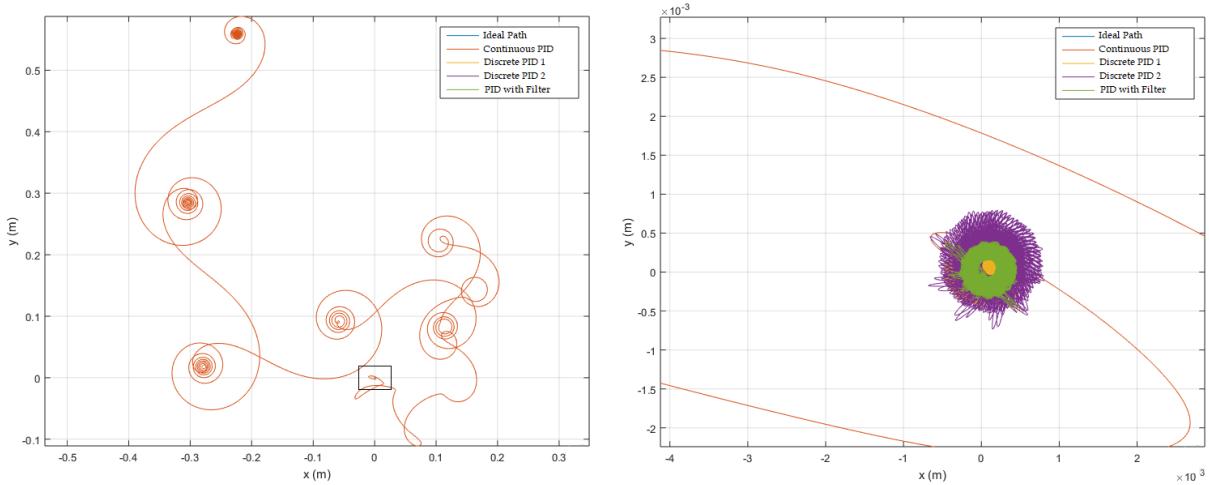


Fig. 4.6: Posição Global: (Esquerda) Caminho Completo e (Direita) com zoom.

posição global, os índices de comparação também foram coletados para as próximas seções. Também podem ser vistos nas imagens 4.4 e 4.5, os erros em ω para esse SMC, respectivamente, com ambas as malhas e com apenas a malha do controlador.

4.2 Malha Dupla

Na imagem da posição global do robô (Figura 4.6) o PID contínuo teve o pior desempenho porque, uma vez que o robô é um sistema micro-controlado, um controlador discreto é mais apropriado. Mas os outros métodos PID tinham quase o mesmo desempenho e precisavam de algum outro método para serem comparados.

Na imagem dos erros na Figura 4.7 a informação é a mesma: o PID contínuo não consegue convergir. Nem mesmo com um zoom na imagem, na figura 4.8 fica claro qual controlador funcionou melhor.

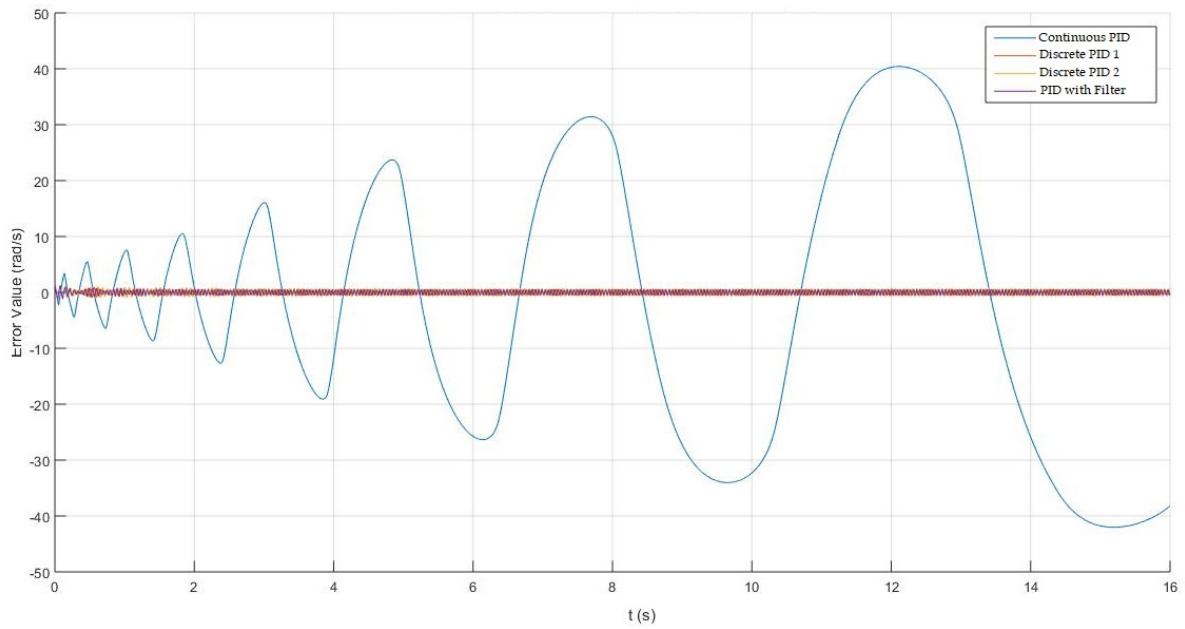
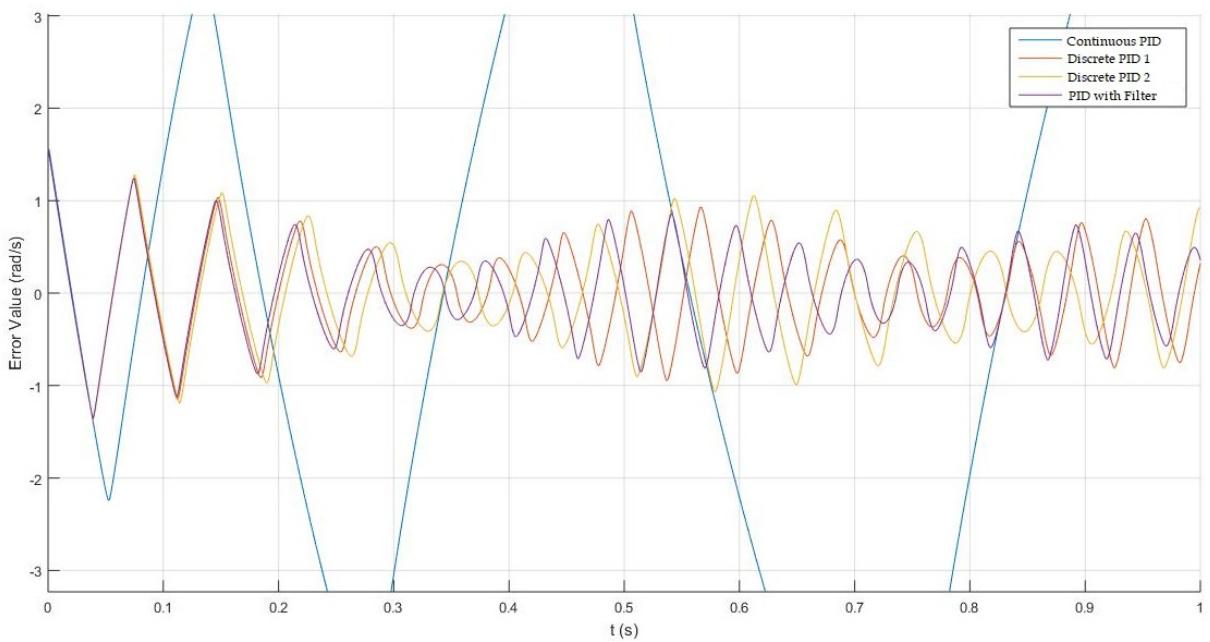
**Fig. 4.7:** Erros em ω **Fig. 4.8:** Erros em ω com zoom

Tabela 4.2: ITAE em Malha Dupla

| Controlador | ITAE | | | |
|-------------------------|----------|-------|--------|-----------------|
| | Mean | X | Y | ω |
| PID Discreto com Filtro | 14,894 | 2,880 | 1,031 | 40,772 |
| PID Discreto 1 | 16,543 | 3,219 | 1,158 | 45,251 |
| PID Discreto 2 | 19,863 | 3,886 | 1,414 | 54,289 |
| PID Contínuo | 1143,353 | 9,223 | 53,893 | 3366,943 |
| SMC | 58,314 | 0,285 | 0,283 | 174,372 |

Tabela 4.3: ISE em Malha Dupla

| Controlador | ISE | | | |
|-------------------------|----------|--------|--------|-----------------|
| | Mean | X | Y | ω |
| PID Discreto com Filtro | 0,702 | 0,010 | 0,001 | 2,094 |
| PID Discreto 1 | 0,859 | 0,012 | 0,002 | 2,563 |
| PID Discreto 2 | 1,219 | 0,018 | 0,002 | 3,638 |
| PID Contínuo | 3216,439 | 0,229 | 3,990 | 9645,098 |
| SMC | 12,738 | 0,0001 | 0,0001 | 38,216 |

As métricas básicas para comparar o controlador não funcionaram, pois o *Overshot*, o *Rise Time* são muito próximos. Assim foram utilizados o ITAE e o ISE, que funcionam bem para sistemas oscilatórios, retornando um resultado mais apto à comparação.

Na Figura 4.9 o ITAE para os controladores PID foi mostrado, e novamente o contínuo teve o pior desempenho em ordens de grandeza e os demais tiveram desempenho similar. Nas Tabelas 4.2 e 4.3, os respectivos números ITAE e ISE mostraram que o controlador com filtro derivativo obteve o melhor resultado em todos os eixos. Nessa mesma tabela, é possível observar os índices para o SMC, que foram menores que do PID contínuo, porém maiores que os discretos.

Vale ressaltar que as métricas do Slide Mode ficaram bem maiores para ω em relação aos PID's discretos porém no eixo X e Y, esta metodologia, obteve os melhores resultados. Esse resultado é importante, pois mostra o motivo da posição global do robô com o SMC ficar tão próximo do caminho requisitado, já que seus maiores erros ficaram, de fato, na velocidade que ele deveria controlar: ω . Dessa forma, analisando somente a posição global do robô, seria obtida uma conclusão equivocada, pois o robô gira bem próximo do centro, porém controlando sua velocidade de rotação pior que os PID's.

Entre os controladores discretos sem o filtro, o primeiro deles é o que obteve o melhor desempenho, embora ambos tenham tido respostas satisfatórias. Além disso, é possível observar na figura 4.8 que o primeiro controlador discreto tem uma resposta mais próxima ao controlador com o filtro que ao segundo controlador sem filtro.

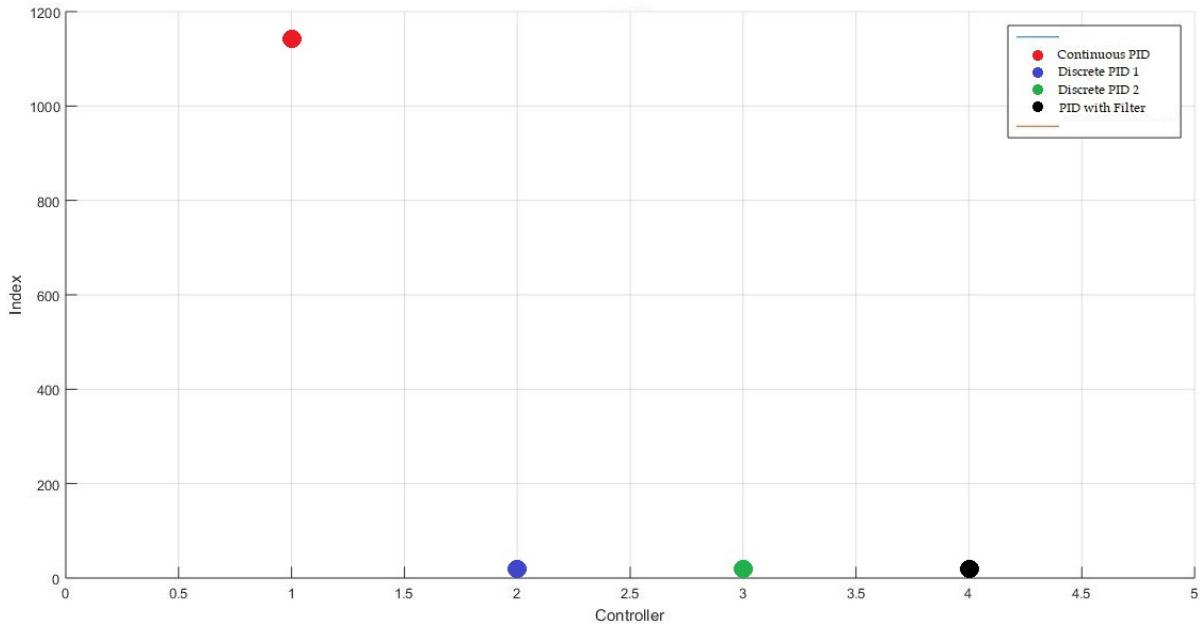


Fig. 4.9: ITAE com Mlaha Dupla

4.3 Malha Simples do Motor

Com apenas o controlador do motor, a posição global em 16 segundos é diferente e pode ser vista na figura 4.10. E mais uma vez é notável, que o PID contínuo não fez um bom caminho. Agora os erros de velocidade não são mais da velocidade do corpo rígido, mas da velocidade angular das rodas.

Olhando para os erros médios entre as 4 rodas, as métricas básicas podem fornecer algumas informações sobre os controladores. Na Figura 4.11, o PID contínuo teve o *Rise Time* mais rápido, mas também o *Overshoot* e *Settling Time* mais alto. O segundo PID discreto e o PID com o filtro tiveram quase a mesma resposta, com o *Rise Time* mais lento e os melhores *Overshoot* e *Settling Time*. Dessa forma, o primeiro PID discreto teve uma resposta intermediária, com um Tempo de *Overshoot* e *Settling Time* maior que o segundo e aquele com filtro, mas um *Rise Time* mais rápido.

Analizando o ITAE na Tabela 4.4 pode-se notar que o valor para todos os quatro controladores é quase o mesmo. Isso aconteceu porque o ITAE é mais sensível aos erros longe do tempo zero, e como os erros se estabilizam em menos de um segundo, não é possível obter um resultado comparativo. No entanto, o PID contínuo tem o melhor desempenho nesta métrica.

Olhando para o ISE na mesma tabela, é possível perceber que o primeiro PID discreto teve o melhor desempenho. Isso vem da resposta intermediária: ela sobe rápido o suficiente para não contribuir tanto para a métrica e, ao mesmo tempo, não tem um grande *Overshoot*, diminuindo seu ISE.

Quanto ao SMC, é possível ver que ele tem o maior ITAE entre os controladores e o menor ISE. Seu ITAE é o maior pois seu *Rise Time* é o maior de todos, podendo ser

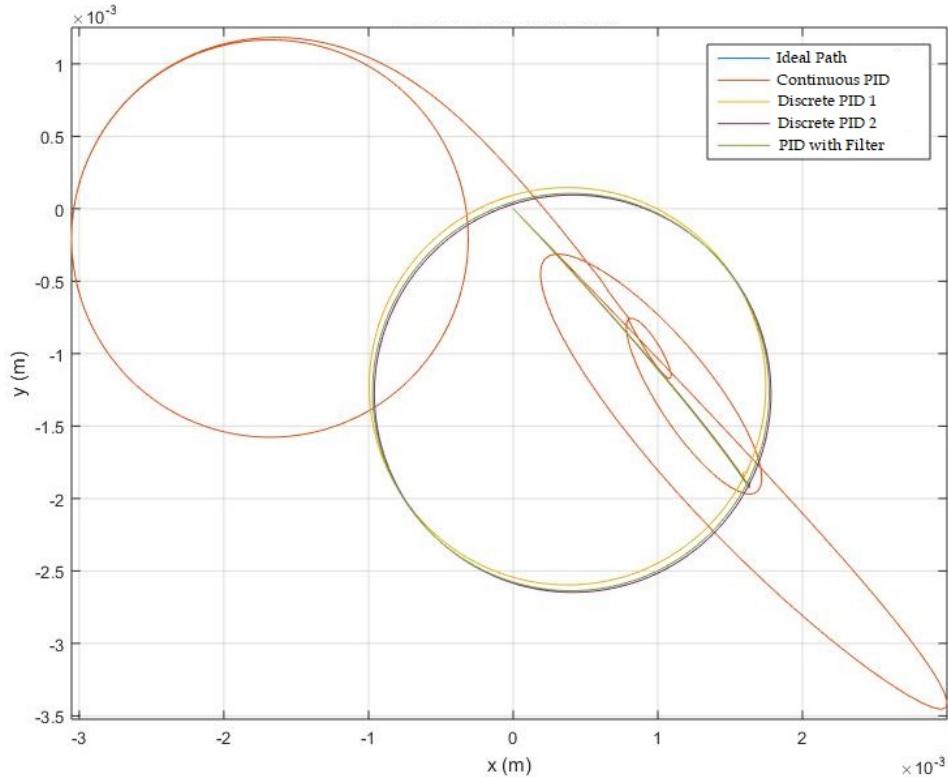


Fig. 4.10: Posição Global Apenas com o Controlador de Motor

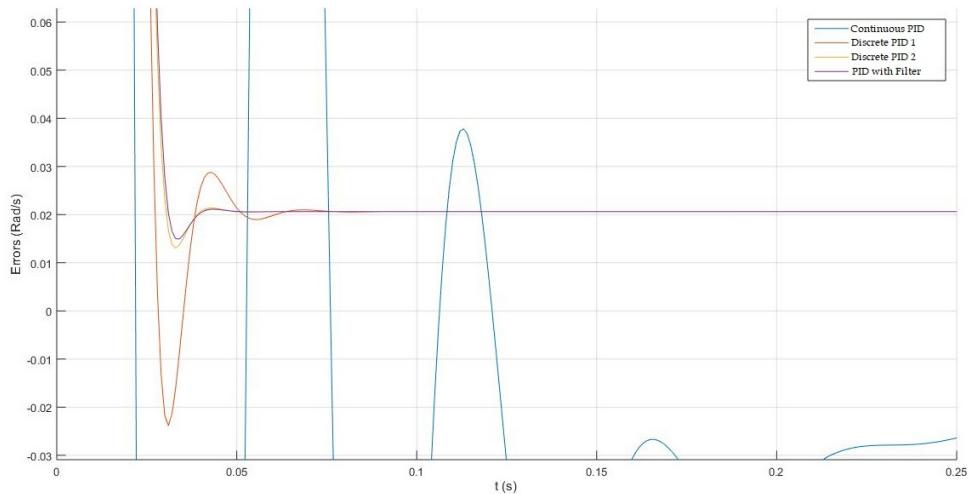


Fig. 4.11: Média de Erros entre os Motores com zoom

observado nas imagens do erro de ω , 4.5 para o SMC e na 4.11. O resultado de seu ISE é baixo, pois uma vez que o controlador atinge o erro zero, sua margem de erro fica bem baixa e não oscila com erros consideráveis como o PID. Pode se dizer que uma das grandes vantagens de se usar a topologia de SMC é obter-se um *Rise Time* muito próximo de seu *Settling Time*.

Tabela 4.4: ITAE e ISE para o Controlador de Motor

| | ITAE |
|-------------------------|---------|
| PID Contínuo | 2.63587 |
| PID Discreto 1 | 2.63886 |
| PID Discreto 2 | 2.63887 |
| PID Discreto com Filtro | 2.63888 |
| SMC | 6.1084 |

| | ISE |
|-------------------------|--------|
| PID Discreto 1 | 0.1147 |
| PID Discreto 2 | 0.1151 |
| PID Discreto com Filtro | 0.1154 |
| PID Contínuo | 0.1451 |
| SMC | 0.0786 |

CAPÍTULO 5

Conclusão e Diretivas Futuras

Um dos resultados mais relevantes deste trabalho foi descobrir que todos os três PIDs discretos, assim como o SMC, têm boas respostas e o contínuo não funciona muito bem. Além disso, o primeiro PID discreto teve uma resposta diferente daquele com filtro e o segundo discreto, com uma tunagem diferente. Essa diferença deve ser útil para o Warthog Robotics, pois eles podem analisar se é mais vantajoso usar um controlador com um *Overshoot* e *Rise Time* altos (primeiro PID discreto, controlador mais rápido mas menos preciso) ou um controlador com um *Overshoot* e *Rise Time* mais baixos (segundo PID discreto e discreto PID com um filtro derivado, controlador mais lento, mas mais preciso).

Com ambos os controladores, o PID com o filtro derivativo teve a melhor resposta e, como os PID's discretos sem o filtro funcionaram melhor com um único controlador, o controlador de corpo rígido combina mais com o PID com filtro. Então, se o controlador de corpo rígido permanecesse o mesmo, a arquitetura com o filtro é uma escolha melhor do que a arquitetura sem o filtro.

O SMC teve bons resultados, porém devido a seus altos índices na variável ω , é possível concluir que ele não consegue manter uma velocidade diferente de zero. Tendo mantido baixos as métricas de comparação de X e Y foi o que trouxe ao robô um bom caminho global.

Com uma melhor escolha de um controlador, o Warthog Robotics melhorará a precisão de seus robôs nos jogos e terá mais chances de ganhar. Além disso, se novas arquiteturas de controladores forem implementadas, o grupo pode usar a mesma metodologia apresentada neste trabalho para descobrir se esse novo controlador terá uma resposta melhor ou pior do que esses quatro controladores comparados.

Escolher um bom controlador em robótica é algo inquestionável, independentemente das variáveis a serem controladas. Portanto, metodologias para a comparação do desempenho de controles, como a apresentada nesse trabalho, são algo de extrema importância para a área. Conhecer as possíveis métricas de um controlador e saber tunar ele para maximizar esses índices para cada aplicação, é um importante passo para um se obter um

sistema robótico de excelência.

Um possível trabalho futuro é estudar os índices de mínima variância, ou qualquer outra métrica que analise o controlador em tempo real e implementa-la na telemetria dos robôs do Warthog. Dessa forma, seria possível determinar entradas do robô (velocidades requeridas) que fazem o controle demorar para convergir ou ter um comportamento inesperado, além de acompanhar o desempenho dos controladores durante as partidas. Um outro trabalho que possa ser desenvolvido é estudar controladores de malha dupla, e tentar aprimorar o funcionamento em conjunto dos controles: o controlador de corpo rígido e o controlador de motor. Também é possível aprimorar o Sliding Mode Controller, inserindo derivadas e tunando o parâmetro p .

Referências

- [1] M. Wang and J. N. K. Liu, “Autonomous Robot Navigation Using Fuzzy Logic Controller,” pp. 26–29, 2004.
- [2] E. C. A. Balestrino, “Advanced PID controllers in MIMO systems ,” 2010.
- [3] N. B. N. J. G. Ziegler, “Optimum Settings for Automatic Controllers ,” 1942.
- [4] T. H. K. J. ASTROM, “Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins ,” 1984.
- [5] M. L. Aguiar, “Controle Digital Notas de Aula de Teoria e de Laboratório ,” 2016.
- [6] H. C. G. De Campos, M. C. M. M.; Teixeira, “Controles típicos de equipamentos e processos industriais,” 2006.
- [7] P. E. Stockler, *TCA: Controle de Processos 2S / 2012 Universidade de Brasília Depto. Engenharia Elétrica*.
- [8] V. I. UTKIN, “Variable Structure Systems with Sliding Modes,” 1977.
- [9] P. d’Armi, “A QUICK INTRODUCTION TO SLIDING MODE CONTROL AND ITS APPLICATIONS.”
- [10] K. Y.S.Kishore Babu, “SLIDING MODE SPEED CONTROL OF A DC MOTOR,” 2011.
- [11] J. M. P. R. d. P. G. N. B. B. L. C. R. R. G. L. I. N. d. S. R. A. F. R. Leonardo B Farconi, Ecyo Farias, “State-Space Modeling and Offline Evolutive Parameter Estimation of a Generic Robotic Platform.”
- [12] K. Astrom, “PID controllers: theory, design and tuning,” p. 343, 1995. [Online]. Available: <http://ci.nii.ac.jp/naid/10013391165/>

- [13] A.J.lsaksson and S.F.Graebe, “Derivative filter is an integral part of PID design ,” 2002.
- [14] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, and M. F. Silva, “Comparison of PID Controller Tuning Methods : Analytical / Classical Techniques versus Optimization Algorithms,” *Carpathian Control Conference (ICCC), 2017 18th International*, pp. 533–538, 2017.
- [15] G. Ferreira and T. Knop, “Avaliação de Desempenho Baseada em Variância Mínima de Controladores para Malha de pH,” 2011.
- [16] F. Fermino, “Estudo comparativo de métodos de sintonia de controladores PID,” p. 90, 2014.
- [17] K. Åström, “Introduction to stochastic control theory,” 2012. [Online]. Available: <https://books.google.nl/books?hl=en&lr=&id=KregHuYKKVoC&oi=fnd&pg=PP1&dq=astrom,+stochastic+control&ots=H9gr9I82pm&sig=kodCHpEwgbh82zdZFL5VggA8kaQ>
- [18] R. Magdalena, “Critérios de avaliação de desempenho para controladores aplicados a mancais magnéticos,” 2014.
- [19] T. J. Harris, “Assessment of Control Loop Performance,” *Canadian Journal of Chemical Engineering*, vol. 67, no. 5, pp. 856–861, 1989.
- [20] M. L. Tyler and M. Morari, “Performance monitoring of control systems using likelihood methods,” *Automatica*, vol. 32, no. 8, pp. 1145–1162, 1996.
- [21] M. Jelali, “An overview of control performance assessment technology and industrial applications,” *Control Engineering Practice*, vol. 14, no. 5, pp. 441–466, 2006.
- [22] E. Kharisov, N. Hovakimyan, and K. Åström, “Comparison of Several Adaptive Controllers According to Their Robustness Metrics,” *AIAA Guidance, Navigation, and Control Conference*, no. August, 2010. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/6.2010-8047>
- [23] M. Kantner, B. Bodenheimer, P. Bendotti, and R. Murray, “An experimental comparison of controllers for a vectored thrust, ducted fan engine,” *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, pp. 1956–1961. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=531229>
- [24] P. Dias De Oliveira Evald, J. Mor, D. Debiaze De Paula, A. Ferreira, S. Silva Da Costa Botelho, and R. Azzolin, “Performance comparison of control strategies applied on welding robot plant,” *IEEE International Symposium on Industrial Electronics*, 2017.

- [25] M. Zul, F. Bin, M. Zawawi, I. Elamvazuthi, A. Binti, A. Aziz, and S. A. Daud, “Comparison of PID and Fuzzy Logic Controller for DC Servo Motor in the development of Lower Extremity Exoskeleton for Rehabilitation,” 2017.
- [26] Q. J. LSuping Li, “STUDY ON PID PARAMETERS TUNING METHOD BASED ON MATLAB/SIMULINK ,” 2011.
- [27] S. P. . G. S. Qureshi, M. S., “Assessment of DC Servo Motor with Sliding Mode Control Approach,” 2016.
- [28] V. I. Utkin, “Sliding Mode Control Design Principles and Applications to Electric Drives ,” 1993.