

INSTITUTO FEDERAL DO ESPÍRITO SANTO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO

**NEMERSON DESTEFANI BERNARDES**

**IMPLEMENTAÇÃO DO PID FRACIONÁRIO COM FILTRO DE KALMAN EM UM  
ROBÔ MÓVEL DIFERENCIAL**

Serra  
2017

NEMERSON DESTEFANI BERNARDES

**IMPLEMENTAÇÃO DO PID FRACIONÁRIO COM FILTRO DE KALMAN EM UM  
ROBÔ MÓVEL DIFERENCIAL**

Dissertação apresentada à Coordenadoria do Curso de Mestrado Profissional em Engenharia de Controle e Automação do Instituto Federal do Espírito Santo, como requisito parcial para o título de Mestre em Controle e Automação.

Orientadores: Prof. Dr. Marco Antonio de Souza Leite Cuadros e Prof. Dr. Gustavo Maia de Almeida.

Serra

2017

### Dados Internacionais de Catalogação na Publicação (CIP)

---

B522i  
2017      Bernardes, Nemerson Destefani  
            Implementação do PID fracionário com filtro de Kalman em um  
            robô móvel diferencial / Nemerson Destefani Bernardes. - 2017.  
            94 f.; il.; 30 cm

            Orientadores: Prof. Dr. Marco Antonio de Souza Leite Cuadros  
            e Prof. Dr. Gustavo Maia de Almeida.  
            Dissertação (mestrado) - Instituto Federal do Espírito Santo,  
            Programa de Pós-graduação em Engenharia de Controle de  
            Automação, 2017.

            1. Controladores PID. 2. Controle de processos. 3. Robôs  
            móveis. I. Cuadros, Marco Antonio de Souza Leite. II. Almeida,  
            Gustavo Maia de. III. Instituto Federal do Espírito Santo. IV. Título.

CDD 629.8

---

Bibliotecária Rogeria Gomes Belchior - CRB6/ES 417

**MINISTÉRIO DA EDUCAÇÃO**  
INSTITUTO FEDERAL DO ESPÍRITO SANTO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

**NEMERSON DESTAFANI BERNARDES**

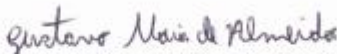
**IMPLEMENTAÇÃO DO PID FRACIONÁRIO COM FILTRO DE KALMAN EM UM  
ROBÔ MÓVEL DIFERENCIAL**


Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Controle e Automação do Instituto Federal do Espírito Santo, como requisito parcial para obtenção de título de Mestre em Engenharia de Controle e Automação.


Aprovado em 15 de dezembro de 2017

**COMISSÃO EXAMINADORA**

  
Prof. Dr. Marco Antonio de Souza Leite Cuadros  
Instituto Federal do Espírito Santo  
Orientador

  
Prof. Dr. Gustavo Maia de Almeida  
Instituto Federal do Espírito Santo  
Orientador

  
Prof. Dr. Luiz Alberto Pinto  
Instituto Federal do Espírito Santo  
Membro Interno

  
Prof. Dr. Celso José Munaro  
Universidade Federal do Espírito Santo  
Membro externo

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus por ter me dado a oportunidade de poder ingressar no mestrado e também por me permitir ter saúde e equilíbrio para transpor tão grandes desafios. Agradeço enormemente a minha amada mãe Nair, que conseguiu garantir minha educação com muita dificuldade, mas com muito amor e sabedoria, mesmo após a perda do meu querido pai Nelson, que faleceu quando eu ainda me preparava para ingressar na Escola Técnica Federal do Espírito Santo. Agradeço em especial a minha esposa Débora e minha querida filha Laura, as quais me proporcionaram equilíbrio e me confortaram nos momentos de maiores dificuldades nesse longo processo. Também agradeço imensamente aos amigos de pesquisa do GAI, em especial ao Felipe e ao Vinícius pelo apoio, paciência e parceria em todo o decorrer do curso. Por fim, mas não menos importante agradeço aos meus orientadores Marco Antonio de Souza Leite Cuadros e Gustavo Maia de Almeida, que além de terem me orientado e capacitado de forma plena, estiveram ao meu lado com grande parceria e amizade, o que foi um grande diferencial e motivador para o término dessa fase tão relevante da minha jornada.

“Desconfio dos veementes. O sujeito que esbraveja  
está a um milímetro do erro e da obtusidade”

(Nelson Rodrigues - 1912/1980)

## RESUMO

Controladores do tipo PID em sua forma tradicional são amplamente utilizados em grande parte dos controles de processos industriais. O desempenho desses controladores, independentemente da complexidade dos processos aos quais são aplicados, justifica essa utilização maciça, uma vez que, para o alcance de uma boa sintonia, é necessária apenas a definição de três parâmetros através de métodos de sintonia que já são bem conhecidos pelos profissionais de automação. Entretanto, com o avanço do cálculo fracionário, a utilização de PIDs fracionários mostrou-se mais eficiente usando-se 5 parâmetros de sintonia. Em particular, permitiu a generalização dos bem conhecidos controladores industriais PI, PD e PID para controladores que utilizam operadores derivativos e de integração fracionários. Desta forma, vários novos controladores como o  $PI^\lambda$ ,  $PD^\mu$  e  $PI^\lambda D^\mu$  começaram a ser implementados e até comparados com os tradicionais controladores de ordem inteira. Geralmente, busca-se a aplicação de PIDs de ordens fracionárias em processos, inicialmente de ordens superiores ou de ordens fracionárias, os quais foram aproximados para ordens inteiras, uma vez que, comprovadamente, os controladores de ordem fracionária, têm um desempenho mais adequado na estabilidade, no percentual de ultrapassagem e no tempo de assentamento desses sistemas. Este trabalho tem a finalidade de avaliar o uso do PID fracionário para o controle de velocidade das rodas de um robô móvel diferencial, com o objetivo de melhorar o seu desempenho em relação ao PID inteiro, atualmente implementado. A fim de possibilitar a mitigação de ruídos do sistema, foi implementado um filtro de Kalman discreto na variável de entrada do processo. O desempenho do controlador foi comparado com um controlador PID tradicional implementado na mesma malha de controle do robô móvel diferencial utilizado. Os resultados mostraram que o PID fracionário com Filtro de Kalman possibilita uma sintonia mais fina com maior rapidez de resposta e menor variabilidade, o que confirma a aplicação em processos com menores constantes de tempo.

Palavras-chave: PID Inteiro. PID Fracionário. Filtro de Kalman. Robôs móveis.

## ABSTRACT

PID-type controllers in their traditional form are widely used in most industrial process controls. The performance of these controllers, regardless of the complexity of the processes to which they are applied, justifies this massive utilization, since to achieve a good tuning, consists only of the parameters by means of tuning methods that are already well known by the Automation professionals. However, with the advancement of the fractional calculation, it has been shown that the application of fractional PIDs can be more efficient using 5 tuning parameters. In particular, it allowed the generalization of the well-known industrial controllers PI, PD and PID to controllers that use fractional integration and derivative operators. In this way, several new controllers such as  $PI^\lambda$ ,  $PD^\mu$  and  $PI^\lambda D^\mu$  started to be implemented and even compared to traditional full-size controllers. Generally, the application of PIDs of fractional orders in processes of higher orders or fractional orders, which have been approximated to whole orders, is proven, since the fractional order controllers have been shown to have a more adequate performance in the stability, And settling time of these systems. This work has the purpose of evaluating the use of fractional PID to control the speed of the wheels of a mobile differential robot, in order to improve its performance in relation to the whole PID, currently implemented. In order to enable the elimination of system noise, a Kalman filter was implemented in the process input variable. The performance of the controller was compared with a traditional PID controller implemented in the same control loop of the mobile differential robot used. The results showed that the fractional PID with Kalman Filter allows a finer tuning with faster response and less variability, which confirms the application in processes with lower time constants.

Keywords: PID Integer. Fractional PID. Kalman Filter. Mobile Robots.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Evolução do cálculo fracionário.....	21
Figura 2 - Diagrama de blocos PID tradicional.....	25
Figura 3 - Efeito da variação do ganho proporcional em malha fechada .....	25
Figura 4 - Efeito da variação do ganho integral em malha fechada .....	26
Figura 5 - Efeito da variação do ganho derivativo em malha fechada.....	26
Figura 6 - Região de trabalho do PID tradicional e fracionário.....	28
Figura 7 - Detalhe simulação preliminar de um PID fracionário .....	28
Figura 8 - Detalhamento Bloco de Bode do Simulink.....	29
Figura 9 - Bloco do PID fracionário no Simulink.....	30
Figura 10 - Variação do ganho integral .....	31
Figura 11 - Variação do $\lambda$ .....	31
Figura 12 - Variação do ganho derivativo.....	32
Figura 13 - Variação do $\mu$ .....	32
Figura 14 - Diagrama de Bode para variações do $K_i$ .....	33
Figura 15 - Diagrama de Bode para variações do $\lambda$ .....	33
Figura 16 - Diagrama de Bode para variações do $K_d$ .....	34
Figura 17 - Diagrama de Bode para variações do $\mu$ .....	34
Figura 18 - Modelo clássico de processo .....	36
Figura 19 - Esquema IMC .....	36
Figura 20 - Fluxograma AG.....	38
Figura 21 - Algoritmos preditor / corretor.....	43
Figura 22 - Esquemático de operação do Filtro de Kalman discreto.....	44
Figura 23 - Robô móvel diferencial.....	45
Figura 24 - Esquemático do robô móvel.....	46
Figura 25 - Modelo do robô móvel de tração diferencial .....	46
Figura 26 - Motor DC do robô.....	48
Figura 27 - Encoder do robô.....	49
Figura 28 - NI myRIO .....	50
Figura 29 - Controladores PIDs tradicionais das rodas .....	51
Figura 30 - Mudança no acionamento de corrente do robô para polias .....	55
Figura 31 - Circuito da placa do robô .....	56
Figura 32 - Robô após melhorias .....	57

Figura 33 - PID ideal com anti wind-up .....	59
Figura 34 - PID tradicional no LabView .....	60
Figura 35 - Cálculo do binômio do PID fracionário .....	62
Figura 36 - Cálculo do operador fracionário .....	62
Figura 37 - Cálculo Fracionário em blocos .....	63
Figura 38 - Ambiente de programação do LabView .....	63
Figura 39 - Tela de programação Matlab .....	64
Figura 40 - Tela programação LabView .....	65
Figura 41 - Validação PID Matlab e LabView .....	65
Figura 42 - Validação PID fracionário no Matlab e LabView .....	66
Figura 43 - Implementação filtro de Kalman LabView .....	67
Figura 44 - Tela de implementação da Odometria do robô .....	68
Figura 45 - Base para geração da trajetória do robô .....	69
Figura 46 - Gráfico dos erros de trajetória.....	70
Figura 47 - Interface dos controladores do robô.....	71
Figura 48 - Resposta ao degrau roda esquerda.....	72
Figura 49 - Resposta ao degrau roda direita.....	73
Figura 50 - Resposta ao degrau e modelo da roda esquerda .....	74
Figura 51 - Resposta ao degrau e modelo da roda direita .....	74
Figura 52 - Resposta ao degrau PI da roda direita.....	77
Figura 53 - Resposta ao degrau PI da roda esquerda .....	78
Figura 54 - Resposta ao degrau PIF da roda direita .....	79
Figura 55 - Resposta ao degrau PIF da roda esquerda .....	80
Figura 56 - Medição de velocidade da roda esquerda com KF .....	81
Figura 57 - Medição de velocidade da roda direita com KF .....	81
Figura 58 - Resposta ao degrau PIK e PIFK da roda direita .....	83
Figura 59 - Resposta ao degrau PIK e PIFK da roda esquerda .....	83
Figura 60 - Erro absoluto de trajetória ao longo do tempo de cada controlador .....	85
Figura 61 - Teste de trajetória .....	86
Figura 62 - Teste de trajetória final.....	87
Figura 63 - Controle de velocidade da roda direita PIF e AG.....	87
Figura 64 - Controle de velocidade da roda esquerda PIF e IMC .....	88

## LISTA DE TABELAS

Tabela 1 - Resumos das ações do PID.....	27
Tabela 2 - Características do motor do robô .....	48
Tabela 3 - Características do encoder do robô .....	49
Tabela 4 - Parâmetros de sintonia IMC.....	75
Tabela 5 - Parâmetros AG.....	76
Tabela 6 - Sintonia PIDs inteiros .....	76
Tabela 7 - Índices de desempenhos dos PIDs inteiros .....	77
Tabela 8 - Sintonia dos PIDs fracionários .....	78
Tabela 9 - Índices de desempenho dos PIDs fracionários .....	79
Tabela 10 - Índices de desempenho dos PIDs com filtro de Kaman .....	82
Tabela 11 - Índices de desempenho dos PIDs fracionários com filtro de Kalman .....	82
Tabela 12 - Sintonia controlador de trajetória.....	84
Tabela 13 - Resultados testes de trajetória .....	85

## LISTA DE ABREVIATURAS, SIGLAS

AG - Algoritmo genético

B - Critério de desempenho do IMC

CLP - Controlador lógico programável

De - Distância euclidiana

Ess - Erro em estado estacionário

FT - Função de transferência

GAIIn - Grupo de Pesquisa de Automação Industrial do Campus Serra do Instituto Federal do Espírito Santo

GPS - Global Positioning System

IMC - Internal model control

ITAE - Time-weighted absolute of error

KF - Filtro de Kalman

NI - National Instruments

OS – Overshoot

$PD^{\mu}$  - Controlador proporcional e integral fracionário

$PI^{\lambda}$  - Controlador proporcional, integral e derivativo fracionário

PID - Controlador proporcional, integral e derivativo

$PI^{\lambda}D^{\mu}$  - Controlador proporcional integral

PIF - Controlador proporcional integral fracionário

PIFK - Controlador proporcional integral com filtro de Kalman

PIK - Controlador proporcional integral fracionário

SDCD - Sistema digital de controle distribuído

SubVI - Subroutine virtual instruments

TA - Tempo de assentamento

TU - Tempo de ultrapassagem

Var – Variabilidade

VI - Virtual instruments

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	14
1.1	OBJETIVO	17
1.1.1	<b>Objetivos específicos</b>	17
1.2	METODOLOGIA	18
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	20
2.1	A EVOLUÇÃO DO CÁLCULO FRACIONÁRIO	20
2.1.1	<b>Cálculo fracionário por Grunwald-Letnikov</b>	22
2.2	CONTROLADORES PID	24
2.2.1	<b>Efeitos das ações Proporcional, integral e derivativa</b>	25
2.2.2	<b>Controlador PID Fracionário</b>	27
2.2.3	<b>Simulações dos PIDs no Simulink</b>	28
2.3	MÉTODOS DE SINTONIA	35
2.3.1	<b>O método IMC</b>	35
2.3.2	<b>Algoritmo Genético</b>	37
2.4	O FILTRO DE KALMAN	38
2.4.1	<b>O Filtro de Kalman discreto</b>	39
2.4.2	<b>As origens computacionais do filtro de Kalman</b>	40
2.4.3	<b>O algoritmo do filtro de Kalman discreto</b>	42
<b>3</b>	<b>O ROBÔ MÓVEL DIFERENCIAL</b>	45
3.1	DESCRIÇÃO DO SISTEMA DE ACIONAMENTOS	47
3.2	PLACA NI MYRIO	49
3.3	CONTROLADOR PID ORIGINAL DAS RODAS DO ROBÔ MÓVEL	50
3.4	ODOMETRIA	51
3.4.1	<b>Dead reckoning</b>	53
3.5	CONTROLADOR DE TRAJETÓRIA	53
<b>4</b>	<b>IMPLEMENTAÇÃO</b>	55
4.1	ADEQUAÇÕES REALIZADAS NO ROBÔ	55
4.2	SOFTWARES UTILIZADOS	57
4.3	PID TRADICIONAL (INTEIRO)	58
4.4	PID FRACIONÁRIO	60
4.5	VALIDAÇÃO DOS PIDS IMPLEMENTADOS	64
4.6	FILTRO DE KALMAN	66

4.7	DEAD RECKONIG (ODOMeTRIA).....	67
4.8	GERADOR DE TRAJETÓRIA .....	68
4.9	CONTROLADOR DE TRAJETÓRIA BACKSTEPPING.....	69
<b>5</b>	<b>RESULTADOS</b> .....	<b>72</b>
5.1	MODELAGEM DOS ACIONAMENTOS.....	72
5.2	RESPOSTAS DOS PIDS DAS RODAS.....	74
5.2.1	<b>Sintonia e desempenho do PID inteiro</b> .....	<b>75</b>
5.2.2	<b>Sintonia e desempenho do PID fracionário</b> .....	<b>78</b>
5.2.3	<b>Sintonia e desempenho dos PIDs com filtro de Kalman</b> .....	<b>80</b>
5.3	CONTROLE DE TRAJETÓRIA .....	84
<b>6</b>	<b>CONCLUSÕES E RECOMENDAÇÕES</b> .....	<b>89</b>
	<b>REFERÊNCIAS</b> .....	<b>91</b>

## 1 INTRODUÇÃO

Os sistemas de controle fazem parte da vida cotidiana, tanto em nossas casas, indústrias, além dos sistemas de comunicação e transporte e grande impacto nos campos da economia, biologia e medicina.

A aplicação do controlador do tipo PID (Proporcional Integral Derivativo) tendo como base a diferença entre um valor de referência e a resposta real do processo surgiu na década de 1940, e desde então, tornou-se uma ferramenta padrão no que tange as estratégias para controle de processo no meio acadêmico e industrial (JOHAN, 2002).

Desde os sistemas mecânicos aos sistemas integrados e microprocessados, os controladores PID têm sobrevivido a muitas mudanças resultantes da evolução das tecnologias. O microprocessador teve uma influência significativa na implementação de controladores PIDs, sendo atualmente, uma boa parte baseada nessa tecnologia. Assim surgiram oportunidades para a inserção de recursos adicionais, como ajuste automático, programação de ganho, adaptação contínua e utilização de cálculos matemáticos mais complexos.

Os controladores PID podem ser associados a lógicas computacionais, funções sequenciais, diagramas de blocos de função através de SDCDs e CLPs. Também são utilizados em sistemas de automação complexos como na produção de energia, transporte, mineração, siderurgia, óleo e gás (JOHAN, 2002).

Próximo a 96% dos controladores aplicados na indústria são do tipo PID e a maioria do tipo PI. Devido a esta constatação, ainda há uma área muito ativa dentro do meio acadêmico e industrial no campo de pesquisa e desenvolvimento de softwares envolvendo a sintonia e otimização de controladores PID (PINTO, 2014).

Com objetivo de melhorar o desempenho de controladores PID, Podlubny (1997 e 1999) propôs uma generalização dos controladores industriais PI, PD e PID em controladores que utilizam cálculo de derivadas e integrais fracionárias,



possibilitando o desenvolvimento de novos controladores como  $PI^\lambda$ ,  $PD^\mu$  e  $PI^\lambda D^\mu$ , sendo que  $\lambda$  e  $\mu$  são números reais positivos.

Um exemplo de aplicação de controladores com alto grau de complexidade é a robótica móvel. Em condições usuais aos seres humanos, robôs normalmente precisam realizar suas tarefas em ambientes dinâmicos, com a presença de objetos, animais e pessoas em movimento.

Nas indústrias de processos em diversos segmentos, plantas nucleares e nos vários campos da medicina, atualmente é aplicada uma série de manipuladores robóticos. Os sistemas de controle desses locais têm características que exigem posicionamento rápido e preciso que só são possíveis quando os controladores aplicados conseguem efetivamente bom desempenho, independente da complexidade (SHARMA et al., 2017).

Um dos problemas mais comuns da robótica móvel é a navegação, que aumenta consideravelmente em robôs que efetuam suas tarefas em ambientes mais controlados. Por se tratar de um problema complexo, diversas técnicas são utilizadas na tentativa de solucionar este problema. Dentre elas estão a aquisição e o processamento de dados sensoriais, tomada de decisão, planejamento de trajetória e controle de movimento (BECKER, 2015).

Os robôs são elementos essenciais na sociedade de hoje. Eles são capazes de executar muitas tarefas de forma repetitiva e precisa sem a ajuda requerida pelos humanos (ABDALLA; HAMZAH, 2013). O projeto de um robô é uma tarefa multidisciplinar que envolve a engenharia mecânica, eletrônica, informática, inteligência artificial e controle. São exemplos de sistemas de controle de um robô móvel o controle de posição de um braço ou a força exercida por uma ferramenta, o controle de velocidade das rodas de um robô diferencial, a implementação de algoritmos que permitam a autonomia para execução de tarefas com o intuito de mapeamento de ambientes e possíveis obstáculos do deslocamento (MARTINS, 2007).

Neste trabalho serão aplicados dois controladores, PID e  $PI^\lambda D^\mu$  para controle de velocidade das rodas de um robô móvel diferencial e comparados os seus desempenhos. Também será aplicado o filtro de Kalman, proposto na década de

1960, trazendo uma nova forma de filtragem linear e solução para problemas de predição (KALMAN, 1960), a fim de amenizar os efeitos dos ruídos das medições de velocidades, uma vez que se tratando do controle de velocidade automático das rodas de um robô móvel diferencial, um ponto de grande preocupação é o comprometimento do controle ocasionado pelos acoplamentos através de polias e correias e também pela utilização de inversor de frequência, sendo estes aspectos importantes para a realização da parametrização e sintonia dos controladores utilizados.

Os controladores serão sintonizados através de algoritmo genético (AG) e pelo método do modelo interno de controle (IMC). Através da utilização dos softwares Matlab e LabView serão realizadas simulações e implementações dos controladores, do algoritmo genético e do algoritmo do filtro de Kalman discreto. Através de testes práticos será realizado o refinamento da sintonia. Por fim, os desempenhos dos controladores serão avaliados no controle de trajetória do mesmo robô móvel diferencial utilizado.

O índice de desempenho dos controladores será avaliado com o cálculo do *IAE* (integral do erro absoluto) e através das medições dos Erro em estado estacionário (*Ess*), Sobressinal (*OS*), Tempo de assentamento (*TA*) e o Tempo de ultrapassagem (*TU*).

A dissertação está organizada da seguinte forma: No capítulo 2 será apresentada a fundamentação teórica dos assuntos abordados para o desenvolvimento do trabalho. No capítulo 3 será apresentada uma explanação do robô móvel diferencial utilizado para testes e implementações do trabalho. No capítulo 4 serão abordadas as implementações realizadas no LabView e Matlab para os testes de validações dos controladores e do filtro de Kalman. No capítulo 5 serão apresentados os resultados alcançados no decorrer do trabalho e que possibilitaram a implementação dos controladores e filtro de Kalman no robô móvel e por fim no capítulo 6 serão mencionados os potenciais trabalhos futuros possibilitados pelo desenvolvimento dos estudos atuais, conclusões e recomendações.

## 1.1 OBJETIVO

Implementar um controlador PID inteiro e de ordem fracionária com filtro de Kalman discreto para controle de velocidade das rodas de um robô móvel diferencial e avaliar os respectivos desempenhos no controle de trajetória.

### 1.1.1 Objetivos específicos

A seguir são apresentados os objetivos específicos do trabalho, organizados da seguinte forma:

- a) Melhorar a estrutura física do robô móvel diferencial com a implementação de um sistema de transmissão de movimento mais eficiente e que tenha menos folga, refazer os circuitos eletrônicos em uma única placa, instalar os encoders e testar o sistema de acionamento;
- b) Implementar a odometria e modelar as duas rodas;
- c) Implementar os controles de velocidade das rodas de um robô móvel diferencial, com o objetivo de aplicar dois controladores PID fracionários em conjunto com um filtro de Kalman desenvolvidos através do LabView;
- d) Comparar as respostas dos controladores através da análise de índices de desempenho, limitações e restrições dos sistemas;
- e) Implementar um gerador e um controlador de trajetória para o robô móvel diferencial;
- f) Avaliar o desempenho dos controladores no controle de trajetória do robô.

## 1.2 METODOLOGIA

A fim de avaliar a implementação do PID fracionário e do filtro de Kalman, foram realizados testes iniciais em uma planta de nível e pressão.

Para a continuidade do trabalho foi utilizado um robô móvel diferencial do Grupo de Pesquisa de Automação Industrial (GAI), do Instituto Federal do Espírito Santo, situado no Município de Serra. A malha de controle escolhida como objeto do trabalho, foi o controle automático do acionamento eletromecânico das rodas do robô. Esse sistema gera o movimento do robô e possui um controlador PID que utiliza um encoder incremental por roda para o cálculo da velocidade e também como parte do processo de localização do dispositivo. Entretanto, com a finalidade de eliminar causadores de possíveis não linearidades, será necessário corrigir alguns aspectos construtivos do robô, principalmente nos sistemas de transmissão de velocidade, uma vez que é comum a existência de um *backslash* bastante pronunciado. As placas eletrônicas estão desorganizadas e precisam de ser refeitas para eliminar possíveis causas de erros, como os maus contatos nas conexões. Os encoders instalados no robô possuem apenas um canal, o que não permite, de forma física, realizar a leitura de reversões, assim os encoders serão substituídos por encoders de dois canais.

Após realizar as mudanças físicas, será implementado um programa para no dispositivo microprocessado Ni MyRIO usando o software LabView. Este programa realizará a leitura dos encoders e o acionamento das rodas do robô, além de permitir coletar os dados para análise. A partir dos dados coletados, serão estabelecidas as restrições em relação ao trabalho, considerando as ferramentas de simulação disponíveis e também as instalações atuais do laboratório utilizado.

A partir do modelamento das malhas das rodas será realizada a sintonia dos controladores de ordem inteira e fracionária através da utilização do IMC e do AG. Mediante os resultados obtidos, serão avaliadas e comparadas as características dos controladores através das especificações em relação ao desempenho em malha fechada utilizando-se como parâmetros *Ess*, *TA*, *OS*, *IAE*, margem de ganho e margem de fase.

Posteriormente serão implementados algoritmos para geração e controle de trajetória. Por fim será avaliado o desempenho dos controladores no controle de trajetória do mesmo robô móvel utilizado.

## 2 FUNDAMENTAÇÃO TEÓRICA

O cálculo fracionário é uma ferramenta matemática que encontrou aplicação no aprimoramento do controle automático nos últimos anos. Particularmente, permitiu a generalização dos conhecidos controladores industriais, em controladores mais sofisticados que utilizam derivadas fracionárias e operadores de integração (CASTILLO et al., 2010)

### 2.1 A EVOLUÇÃO DO CÁLCULO FRACIONÁRIO

Em 1695 L'Hôpital perguntou, em carta, ao seu amigo Leibniz, pai da notação moderna do cálculo diferencial, qual seria o significado de  $D_n y \equiv d^n y / dx^n$  quando  $n$  fosse igual a  $1/2$ . Mesmo sem muito rigor e com um tom profético Leibniz afirmou: Segue que  $D^{1/2} x$  será igual a  $x\sqrt{dx}:x$ . “Este é um aparente paradoxo do qual, um dia, importantes aplicações serão obtidas” (LOVERRO, 2004).

A ideia de uma derivada de ordem genérica também não escapou da atenção de Euler que, em 1730, escreveu que a dificuldade em se obter tais derivadas poderia ser melhor entendida, com o auxílio de interpolações na derivada.

Em 1812, Laplace definiu a derivada fracionária em termos de uma integral, e em 1819 a primeira menção, em um texto científico, às derivadas de ordem fracionária foi feita por Lacroix. Em um livro de mais de 700 páginas, Lacroix dedicou menos de duas destas a um problema que visava obter a derivada de ordem fracionária de um polinômio  $y = x^m$ . Para tanto, partiu-se do seguinte fato: No caso em que  $n$  é um número natural temos a equação (1), para  $m \geq n$ :

$$\frac{d^n}{dx^n} y = \frac{d^n}{dx^n} x^m = \frac{m!}{(m-n)!} x^{m-n} \quad (1)$$

Sendo que  $n$  também pode ser um número real, é necessário estender a função fatorial para estes casos, assim, fazendo uso da função gama (CAPELAS, 2005), conforme equação (2):

$$\frac{d^n}{dx^n} y = \frac{d^n}{dx^n} x^m = \frac{\Gamma(m+1)}{\Gamma(m-n+1)} x^{m-n} \quad (2)$$

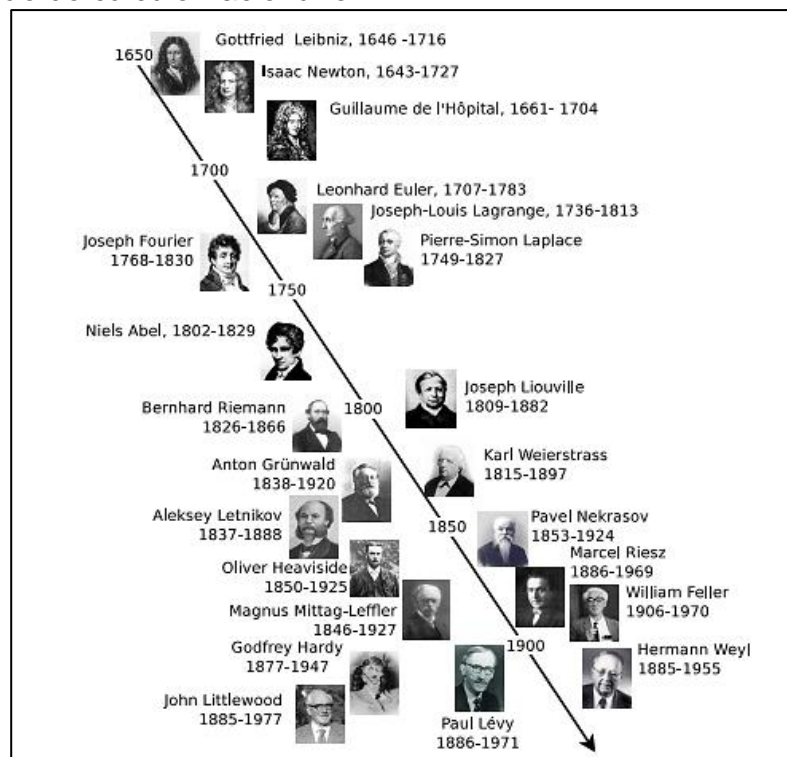
Um caso particular da equação acima pode ser obtido tomando-se  $m=1$  e  $n = 1/2$ , de onde segue o resultado mostrado na equação (3)

$$\frac{d^{1/2}}{dx^{1/2}} y = \frac{d^{1/2}}{dx^{1/2}} x = \frac{2\sqrt{x}}{\sqrt{\pi}} \quad (3)$$

É interessante notar que a dedução feita por Lacroix apesar de ser em certa instância ingênua, coincide neste caso, com a obtida pelo método mais aceito nos tempos atuais, ou seja, o método proposto por Riemann-Liouville.

Muitos cientistas relevantes aprimoraram o cálculo fracionário nestes três últimos séculos, como apresentado na Figura 1.

Figura 1 - Evolução do cálculo fracionário



Fonte: Loverro (2007)

### 2.1.1 Cálculo fracionário por Grunwald-Letnikov

A derivada de Grunwald-Letnikov, introduzida em 1867, é uma das definições usadas para o cálculo da derivada fracionária, mas também é uma extensão do cálculo da derivada inteira ou usual, como mostrado na equação (4).

$$D^\alpha f(x) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{n=0}^{\infty} (-1)^n \frac{\Gamma(\alpha+1)}{n! \Gamma(\alpha-n+1)} f(x-nh) \quad (4)$$

De fato, a generalização proposta por Grunwald-Letnikov é comprovada pelas equações (5), (6), (7) e (8). A base dessa teoria aplicada é que  $h$  deve se aproximar de zero, enquanto  $n$  se aproxima do infinito, sendo  $\alpha$  infinito negativo.

$$D^n f(x) = \lim_{h \rightarrow 0} h^{-n} \sum_{m=0}^n (-1)^m \binom{n}{m} f(x-mh) \quad (5)$$

$$D^\alpha f(x) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^n (-1)^m \frac{n!}{m! \Gamma(n-m+1)} f(x-mh) \quad (6)$$

$$D^\alpha f(x) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^{\infty} (-1)^m \frac{n!}{m! \Gamma(n-m+1)} f(x-mh) \quad (7)$$

$$D^\alpha f(x) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^{\frac{x-a}{h}} (-1)^m \frac{n!}{m! \Gamma(n-m+1)} f(x-mh) \quad (8)$$

No entanto, a generalização é feita de modo que qualquer  $a$  desejado menor que  $x$  possa ser escolhido. A razão pela qual  $x - a$  é considerada nas equações acima em vez de um único número positivo pode ser visto considerando o caso de  $\alpha = -1$ , onde o cálculo deve ter o mesmo resultado, conforme mostrado pelas equações (9) a (12), através de uma integração.

$$\int_a^x f(t) dt = D^{-1} f(x) = \lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} h \sum_{m=0}^n \frac{\Gamma(m+1)}{m! \Gamma(1)} f(x-mh) \quad \therefore \quad (9)$$

$$\lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} \sum_{m=0}^n h f(x-mh) = \lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} \int_0^{nh} f(x-t) dt \quad \therefore \quad (10)$$



$$\lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} \int_{x-nh}^x f(t) dt \therefore \quad (11)$$

$$a = x-nh \therefore n = \frac{x-a}{h}, \quad h = \frac{x-a}{n} \quad (12)$$

Nota-se que, para a resolução de cálculos que contenham derivadas fracionárias os limites também devem ser considerados como nos cálculos de integrais sendo de ordem inteira ou não. Isso também mostra que as derivadas fracionárias são não-locais, o que as torna menos útil na descrição de fenômenos da natureza de forma geral.

No caso da derivada generalizada pela fórmula binomial, equação (4), uma vez que  $n$  tende ao infinito independentemente de  $h$ ,  $x - a$  deve ser infinito, de modo que seja equivalente à derivada Grunwald-Letnikov, com um limite inferior do infinito negativo. Desta forma a equação (8) pode ser generalizada agora para obter uma fórmula equivalente à equação (4), conforme mostrado na equação (13).

$$D^\alpha f(x) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^{\frac{x-a}{h}} (-1)^m \frac{\Gamma(\alpha+1)}{m! \Gamma(\alpha-m+1)} f(x-mh) \quad (13)$$

Ou equivalentemente através da equação (14).

$$D^\alpha f(x) = \lim_{n \rightarrow 0} \left(\frac{n}{x-a}\right)^\alpha \sum_{m=0}^n (-1)^m \frac{\Gamma(\alpha+1)}{m! \Gamma(\alpha-m+1)} f\left(x-m\left(\frac{x-a}{n}\right)\right) \quad (14)$$

As generalizações abordadas anteriormente possuem um inconveniente teórico. A função gama não pode ser utilizada com valores negativos ou iguais a zero. No entanto, para valores negativos de  $\alpha$  podem ser utilizados os seguintes valores, como segue mostrado na equação (15).

$$\binom{-n}{m} = \frac{(-1)^m \Gamma(n+m)}{m! \Gamma(n)} \quad (15)$$

Desta forma as equações (13) e (14) podem ser reescritas para as seguintes formas (16) e (17).

$$D^{-\alpha} f(x) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{m=0}^{\frac{x-a}{h}} (-1)^m \frac{\Gamma(\alpha+m)}{m! \Gamma(\alpha)} f(x-mh) \quad (16)$$

$$D^{-\alpha} f(x) = \lim_{n \rightarrow \infty} \left(\frac{n}{x-a}\right)^{\alpha} \sum_{m=0}^n \frac{\Gamma(\alpha+m)}{m! \Gamma(\alpha)} f\left(x-m\left(\frac{x-a}{n}\right)\right) \quad (17)$$

Como foi explorado anteriormente, os limites das derivadas fracionárias devem ser considerados iguais aos considerados nas integrais. Operações de limites apenas tendem a zero em derivadas de ordem inteira. Devido a este fato, as derivadas fracionárias são consideradas derivadas não locais, o que pode explicar o fato delas não serem tão usuais para descreverem processos físicos naturais.

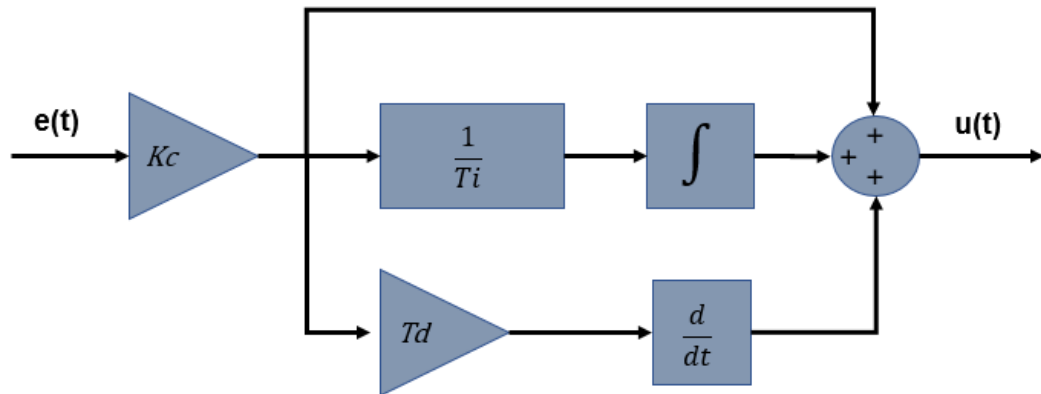
## 2.2 CONTROLADORES PID

O PID (Proporcional-Integral-Derivativo) é o algoritmo de controle mais aplicado na indústria. A popularidade deste tipo de controlador pode ser atribuída em parte, ao seu desempenho robusto em uma ampla gama de condições de funcionamento, e em parte à sua simplicidade funcional, que permite aos engenheiros operá-lo de uma forma simples e direta, favorecendo à sintonia e ao entendimento dos parâmetros a serem determinados (PINTO, 2014).

Como o nome sugere, o algoritmo PID é composto por três modos de controle, o proporcional (P), o integral (I) e o derivativo (D), que são variados para obter uma resposta mais adequada às características da planta na qual aplica-se o controlador. Abaixo segue a equação (18) que é a equação geral de um controlador PID tradicional na sua forma acadêmica (ideal) e na Figura 2 é mostrado o diagrama de blocos do mesmo controlador.

$$u(t) = K_c \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (18)$$

Figura 2 - Diagrama de blocos PID tradicional

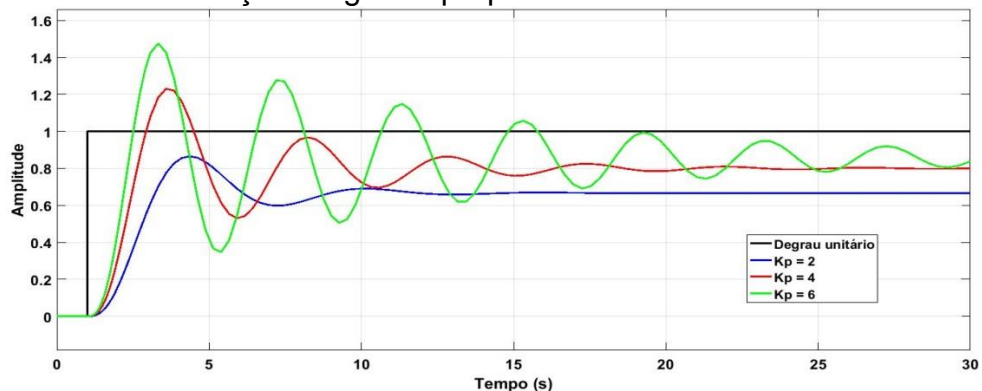


Fonte: Elaborado pelo autor

### 2.2.1 Efeitos das ações Proporcional, integral e derivativa

Nas próximas figuras expostas abaixo são apresentados os efeitos dos ganhos de um controlador PID tradicional aplicado em um processo de terceira ordem,  $P(s)=1/(s+1)^3$ , através da simulação deste processo utilizando-se o software Matlab.

Figura 3 - Efeito da variação do ganho proporcional em malha fechada

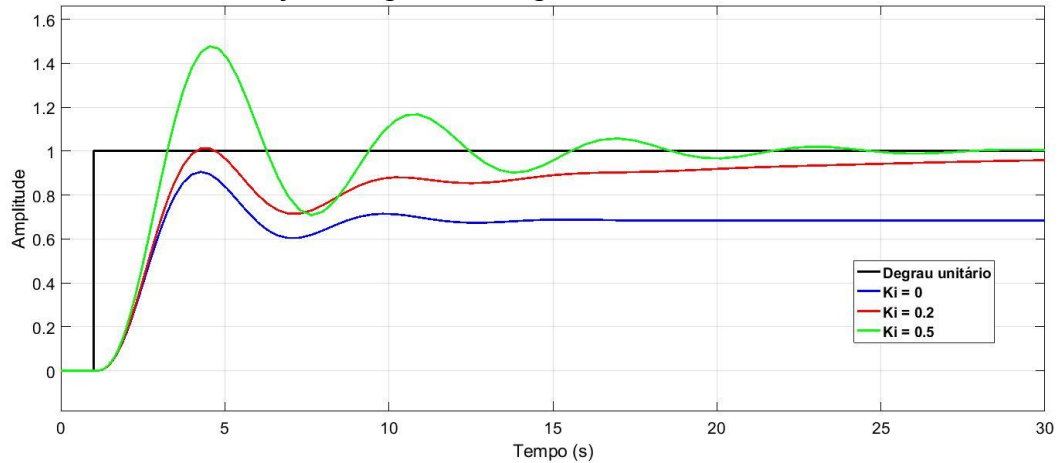


Fonte: Elaborado pelo autor

Através da Figura 3 observa-se o efeito da variação do ganho proporcional  $K_p$  entre 2 e 6. A resposta ao degrau unitário do sistema com o ganho proporcional igual a 2 possui uma característica mais amortecida quando comparada com o controlador com o ganho proporcional igual a 6. O  $TU$  tende a diminuir para ganhos

proporcionais maiores, mas o sistema torna-se mais oscilatório, o que pode levar, em caso de grandes valores de  $K_p$ , à instabilidade. Uma importante observação é que mesmo com a redução do  $Ess$  do sistema, o mesmo nunca será zerado.

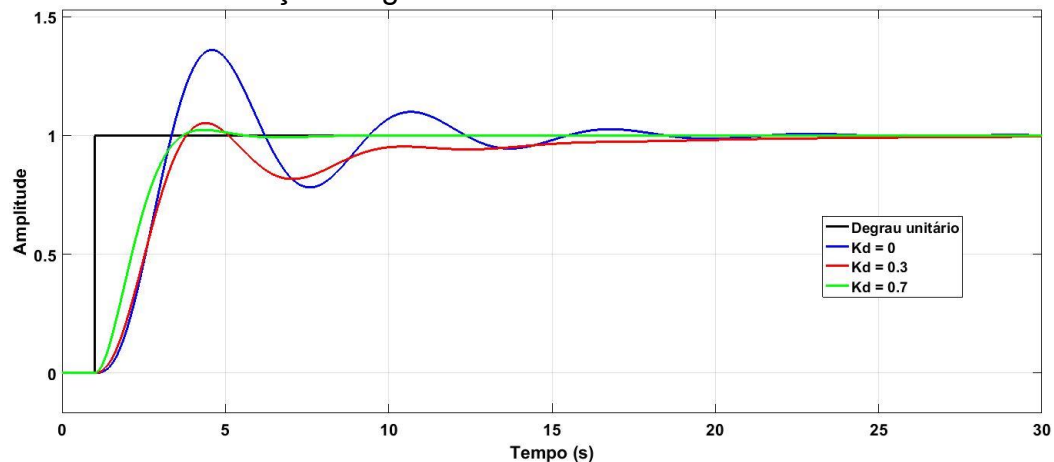
Figura 4 - Efeito da variação do ganho integral em malha fechada



Fonte: Elaborado pelo autor

Agora mantendo-se o ganho proporcional  $K_p$  constante em 2.17 e variando-se o ganho integral  $K_i$  de 0 a 0.5, pode-se notar na Figura 4 que o  $Ess$  tende a zerar com a elevação do ganho integral, mas também é gerado um maior OS para valores maiores, além de possibilitar uma consequente instabilidade em caso de elevações exageradas.

Figura 5 - Efeito da variação do ganho derivativo em malha fechada



Fonte: Elaborado pelo autor

Por fim, testa-se o efeito no controlador simulado, quando o ganho proporcional  $K_p$  é constante em 2.17, o ganho integral  $K_i$  é constante em 0.39 e o ganho derivativo é variado de 0 a 0.7. Observa-se na Figura 5 que, com o acréscimo do ganho derivativo, o  $Ess$  não sofre alteração, mas o  $OS$  é drasticamente reduzido, tornando o sistema mais amortecido sem comprometimento relevante do  $TU$ . Na Tabela 1 segue um resumo dos efeitos de aumento dos ganhos em um controlador PID na configuração ideal, em uma malha fechada de controle.

Tabela 1 - Resumos das ações do PID

Aumento do ganho	TU	OS	TA	Ess
$K_p$	Diminui	Aumenta	Pequenas mudanças	Diminui
$K_i$	Diminui	Aumenta	Aumenta	Elimina
$K_d$	Pequenas mudanças	Diminui	Diminui	Pequenas mudanças

Fonte: Elaborado pelo autor

### 2.2.2 Controlador PID Fracionário

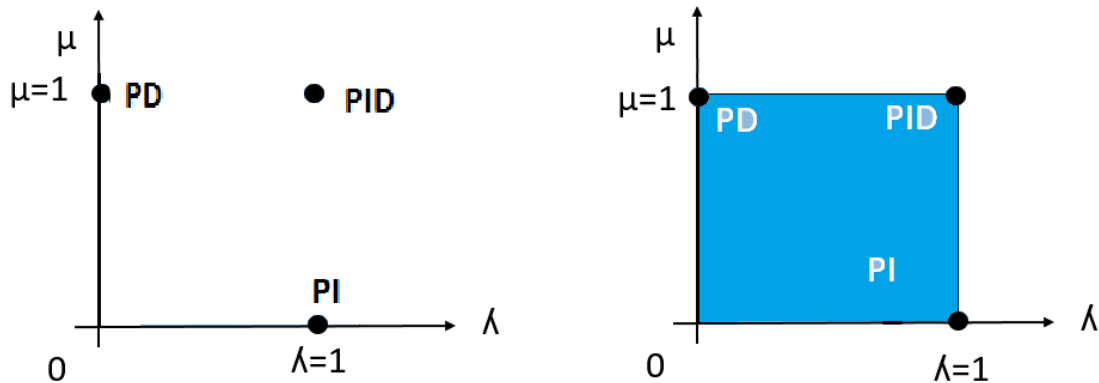
O controlador PID fracionário, também representado por  $PI^\lambda D^\mu$ , emprega o cálculo fracionário para parametrização de controladores PID. Este cálculo permitiu a generalização dos controladores industriais para uma forma mais sofisticada com a utilização de derivadas e operadores de integração fracionários. Como equação geral o  $PI^\lambda D^\mu$ , apresenta-se a função de transferência contínua, no domínio da frequência, equação (19),

$$G(s) = K_c \left( 1 + \frac{1}{s^\lambda T_i} + T_d s^\mu \right) \quad (19)$$

onde  $\lambda$  e  $\mu$  são as ordens fracionárias integral e derivativa, respectivamente. Nota-se que fazendo  $\lambda$  e  $\mu$  iguais a 1 em (19) resume-se à equação do PID tradicional. Assim, pode-se observar que os PIDs tradicionais são um caso particular dos controladores  $PI^\lambda D^\mu$ . A Figura 6 ilustra graficamente a região de variação dos

parâmetros fracionários, representada pela cor azul, enquanto no PID tradicional essa região limita-se aos três pontos PD, PI, PID. Sendo assim, os controladores  $PI^\lambda D^\mu$  agregam maior flexibilidade ao projeto, e permitem o controle de processos reais com maior exatidão, tornando-o, contudo, mais complexo.

Figura 6 - Região de trabalho do PID tradicional e fracionário

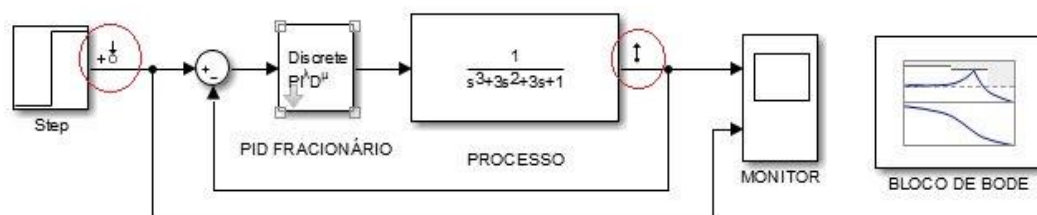


Fonte: Vinagre et al. (2005).

### 2.2.3 Simulações dos PIDs no Simulink

A fim de ilustrar a resposta ao degrau de um sistema controlado por um PID nas configurações inteira e fracionária, foram realizados testes no ambiente de simulação do Matlab, onde foi implementado um controlador PID fracionário e um processo de terceira ordem, conforme a Figura 7.

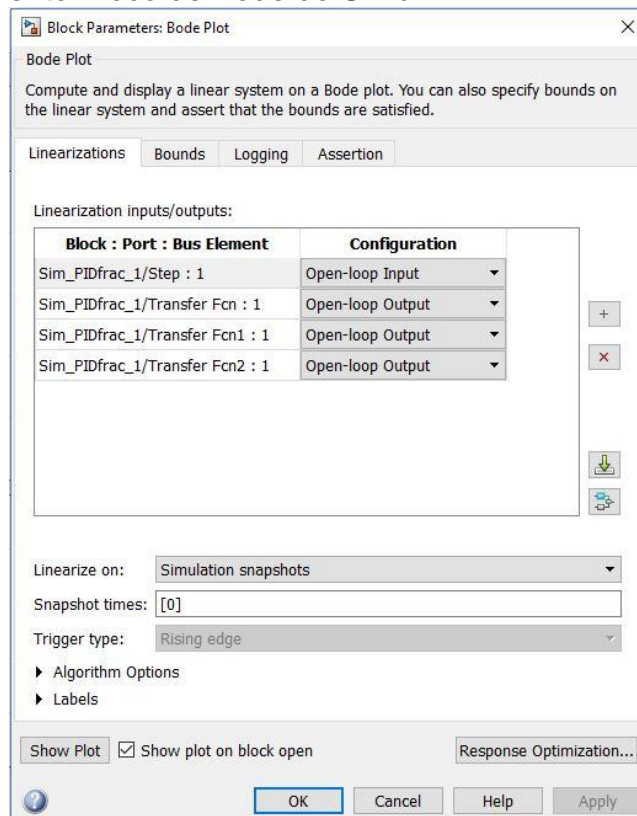
Figura 7 - Detalhe simulação preliminar de um PID fracionário



Fonte: Elaborado pelo autor

O bloco utilizado para a coleta dos gráficos dos diagramas de Bode foi o *Bode Plot*, nomeado como Bloco de Bode na Figura 7. Mesmo em malha fechada é possível realizar a análise determinando as malhas de entrada e saída, conforme segue destacado com círculos vermelhos da Figura 7, e detalhado na Figura 8, onde os parâmetros a serem definidos são apresentados.

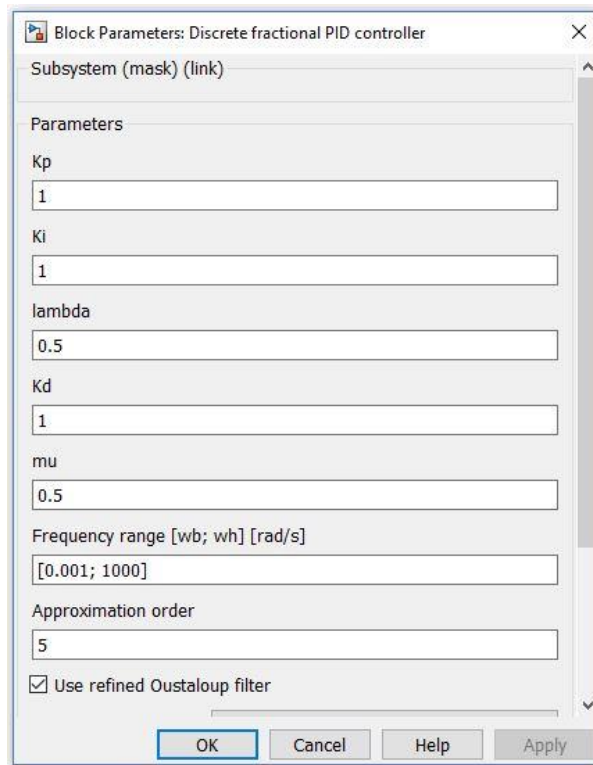
Figura 8 - Detalhamento Bloco de Bode do Simulink



Fonte: Elaborado pelo autor

Os blocos do  $PI^{\lambda}D^{\mu}$  utilizados na simulação são disponibilizados através de uma biblioteca específica do Matlab para implementações de controladores e operadores fracionários, chamada FONCOM. Na Figura 9 segue mostrado esse bloco de forma genérica e os parâmetros que podem ser inseridos, como os ganhos  $P$ ,  $I$  e  $D$  além dos coeficientes fracionários e da ordem da  $FT$ .

Figura 9 - Bloco do PID fracionário no Simulink

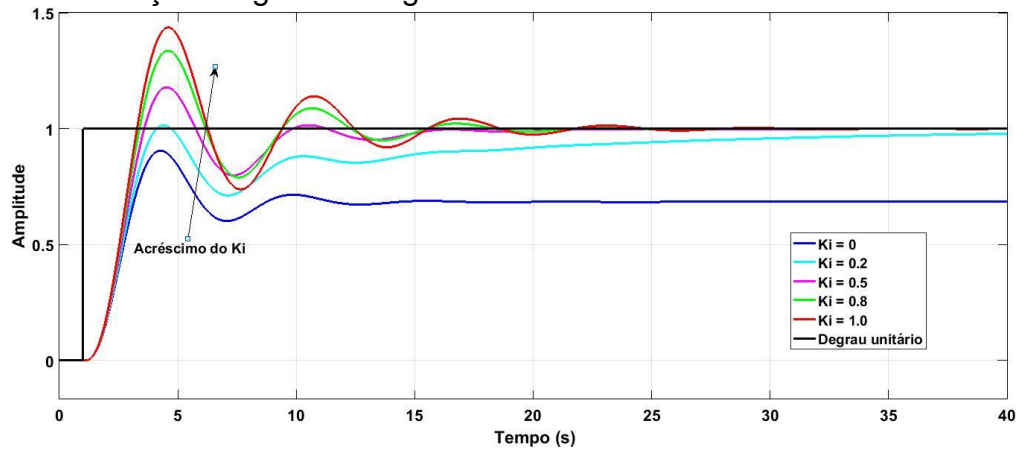


Fonte: Elaborado pelo autor

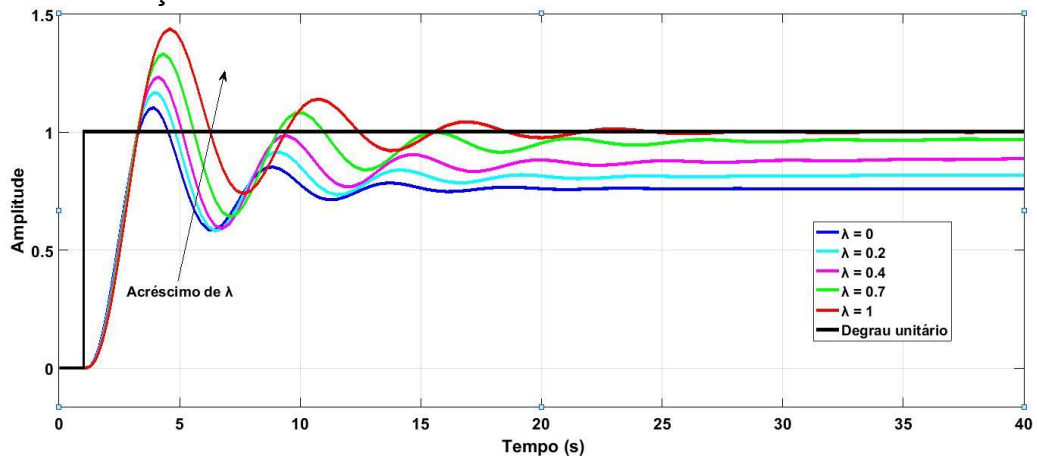
A partir de uma sintonia otimizada do controlador PID, foram estabelecidas faixas de variações dos ganhos  $K_i$  e  $K_d$  e dos coeficientes integral e derivativo  $\lambda$  e  $\mu$ , as quais permitiram uma observação das significativas alterações e comparações dos parâmetros de desempenho da malha de controle em questão. A seguir são apresentadas as respostas ao degrau a partir da seguinte configuração: controlador PID com variações do  $K_i$  de 0 a 1 e do  $K_d$  de 0 a 1.4; controlador PIDF com variações de 0 a 1, tanto para o coeficiente  $\lambda$ , operador integral, quanto para o  $\mu$ , operador derivativo.



Figura 10 - Variação do ganho integral



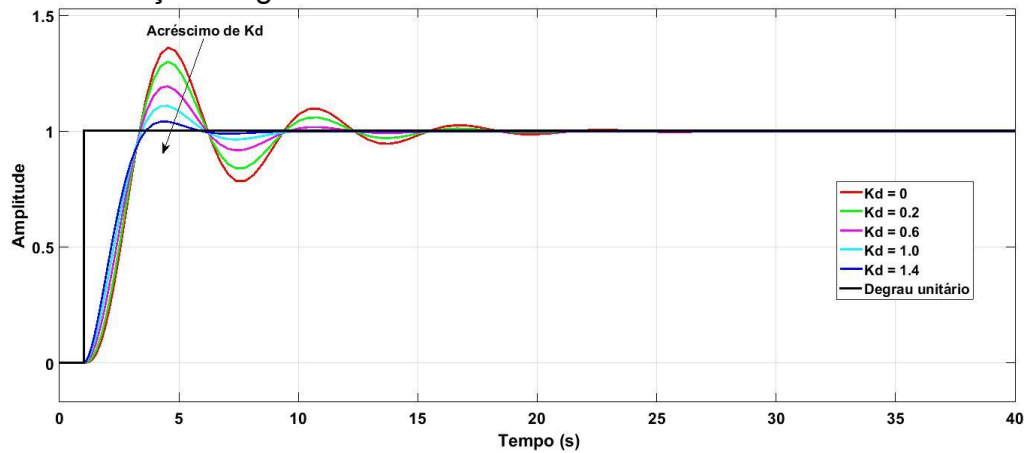
Fonte: Elaborado pelo autor

Figura 11 - Variação do  $\lambda$ 

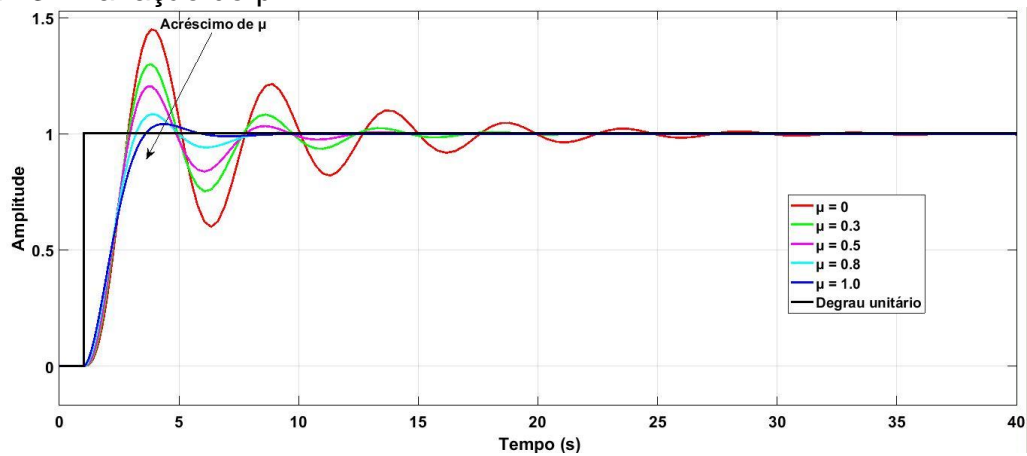
Fonte: Elaborado pelo autor

Através das Figuras 10 e 11 observam-se as características das respostas ao degrau quando há um acréscimo nos valores de  $K_i$  e  $\lambda$ . É visível que em ambos os casos há redução do  $Ess$ , sendo que para alterações do  $K_i$  a redução é mais brusca do que para o  $\lambda$ . Um fator relevante é que com o acréscimo de  $\lambda$ , o  $TU$  é menor em comparação com o acréscimo de  $K_i$ .

Figura 12 - Variação do ganho derivativo



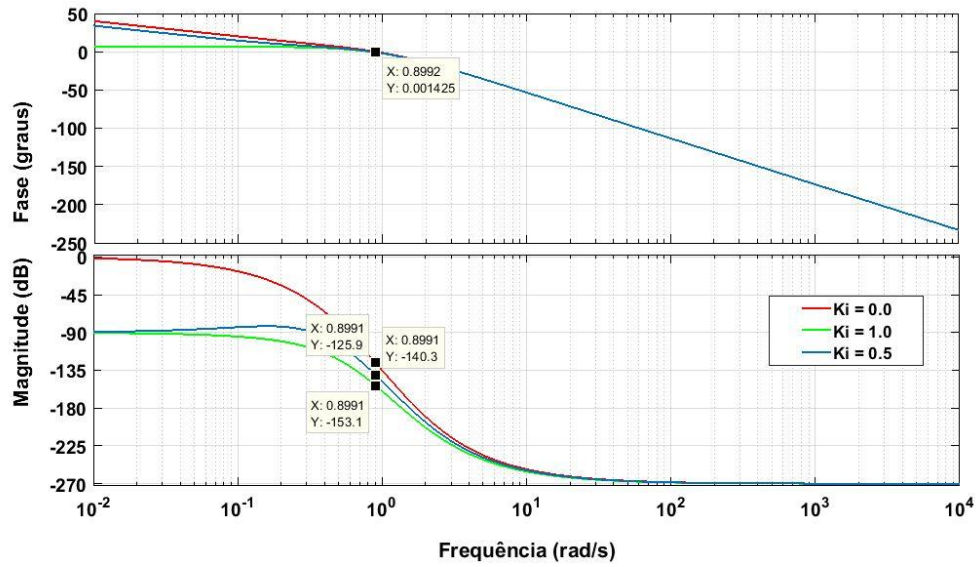
Fonte: Elaborado pelo autor

Figura 13 - Variação do  $\mu$ 

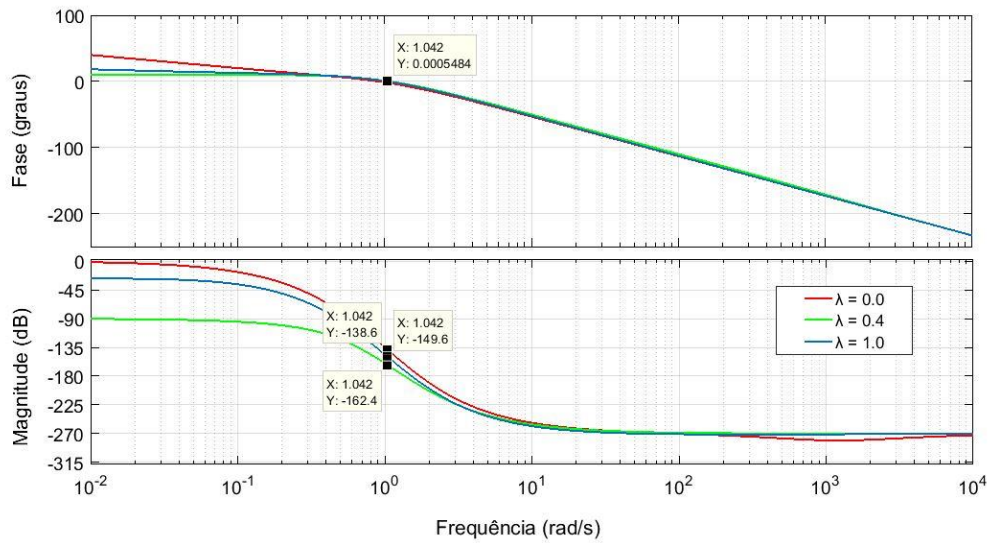
Fonte: Elaborado pelo autor

Nas Figuras 12 e 13, com a variação do  $K_d$  e  $\mu$ , observam-se consideráveis mudanças no comportamento dos sistemas, sendo destacados o amortecimento, OS e  $TA$ . A mesma característica de redução do  $TU$  também é observada no gráfico de variação do  $\mu$  com relação ao  $K_d$ .

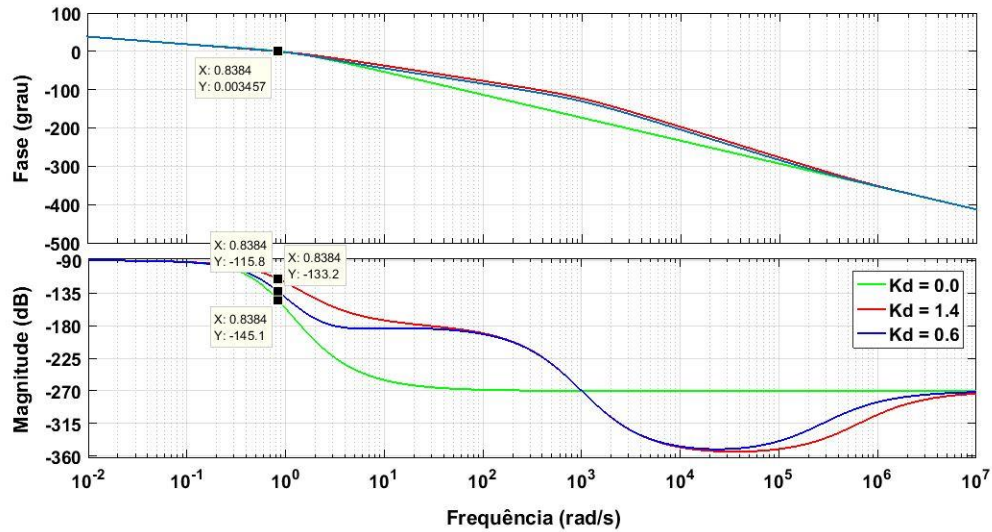
Utilizando-se o diagrama de Bode para avaliação da estabilidade relativa do sistema simulado, são alterados os coeficientes fracionários  $\lambda$  e  $\mu$  do controlador, além dos ganhos  $K_d$  e  $K_i$ , a fim de avaliar-se o comportamento do sistema em relação às margens de fase e de ganho, conforme segue apresentado nas Figuras 14 a 17.

Figura 14 - Diagrama de Bode para variações do  $K_i$ 

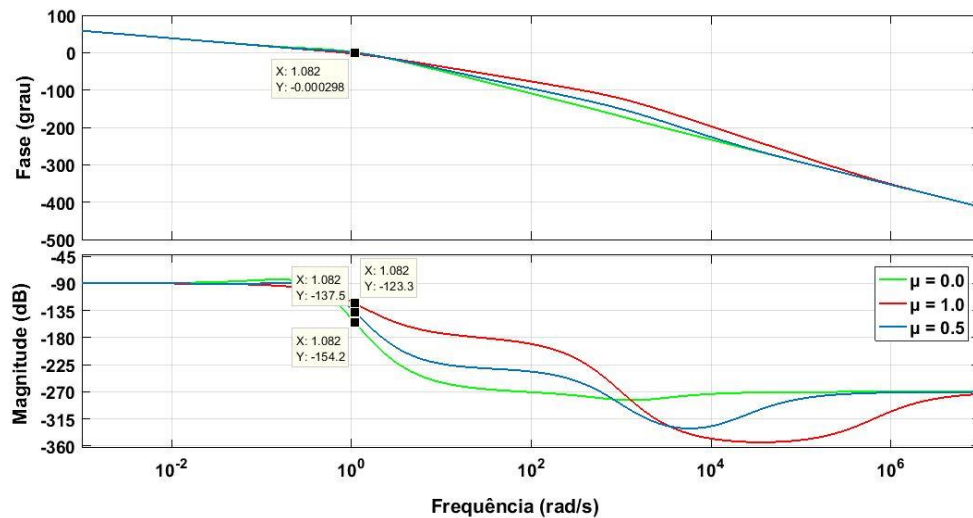
Fonte: Elaborado pelo autor

Figura 15 - Diagrama de Bode para variações do  $\lambda$ 

Fonte: Elaborado pelo autor

Figura 16 - Diagrama de Bode para variações do  $K_d$ 

Fonte: Elaborado pelo autor

Figura 17 - Diagrama de Bode para variações do  $\mu$ 

Fonte: Elaborado pelo autor

Observa-se nos gráficos acima que também é possível mudar o comportamento do sistema em relação à margem de fase. Com o acréscimo de  $K_i$ , a margem de fase variou de  $26,9^\circ$  a  $54,1^\circ$ . Quando houve variação apenas do  $\lambda$ , a margem de fase variou de  $17,6^\circ$  a  $41,4^\circ$ .

Em relação à variação do  $K_d$ , a ação gerou uma faixa de margem de fase de  $34,9^\circ$  a  $64,2^\circ$ , e com a variação apenas do  $\mu$ , houve uma faixa de margem de fase de  $25,8^\circ$  a  $56,7^\circ$ . Esses resultados de simulações mostram que é possível, através da

associação dos ganhos e coeficientes fracionários, a busca por uma sintonia mais precisa e com maiores possibilidades.

## 2.3 MÉTODOS DE SINTONIA

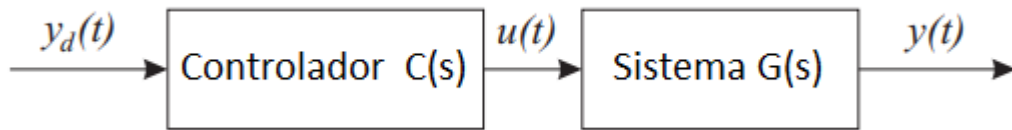
Os parâmetros dos controladores PIDs precisam ser sintonizados a fim de que o desempenho do sistema a ser controlado possa ser o mais próximo do requerido. Desta forma é fundamental que se eleja um ou mais métodos de sintonia para que a busca pelos parâmetros possa ser realizada da forma mais precisa possível.

### 2.3.1 O método IMC

*Internal Model Control* (IMC) refere-se a um procedimento sistemático para o dimensionamento de sistemas de controle com base no conceito de *Q-Parameterization* que é a base para muitas técnicas de controle moderno. O que torna o IMC particularmente atraente é que este método apresenta uma metodologia prática de fácil entendimento para projetar controladores. Como consequência, o IMC tem sido um método de dimensionamento muito popular nos processos indústrias, particularmente como um meio para sintonizar controladores de tipo *loop* único, tipo PID (RIVERA; FLORES, 1986).

A estrutura IMC tem como objetivo a partir do modelo do processo e de uma especificação de desempenho obter o controlador adequado. Portanto, o controlador possui um modelo interno do processo que pode ser utilizado apenas na fase de projeto ou que também pode ser usado na fase de operação (CAMPOS; TEIXEIRA, 2006). Desta forma, a forma clássica de representação de um processo é apresentada a Figura 18.

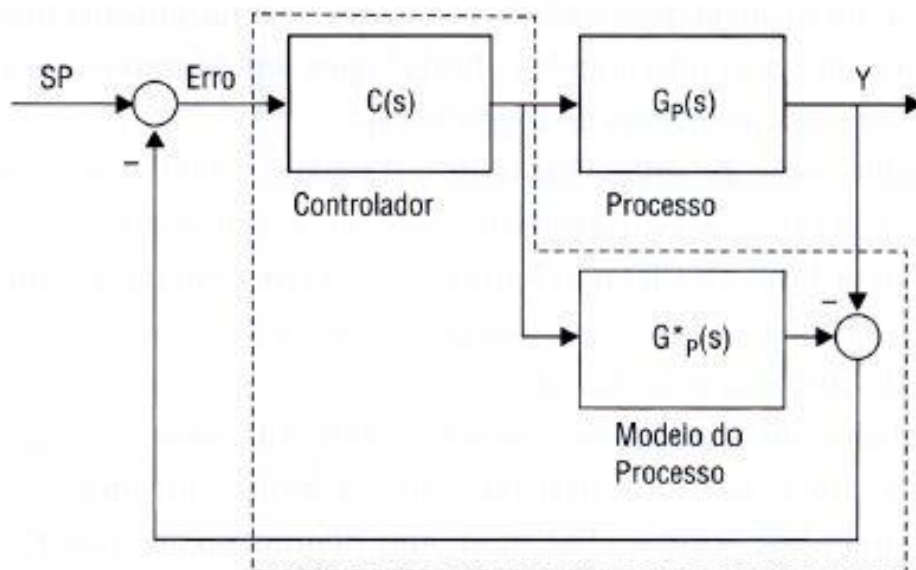
Figura 18 - Modelo clássico de processo



Fonte: Elaborado pelo autor

onde  $y_d(t)$  é o sinal de entrada do controlador,  $C(s)$  é a representação matemática do controlador,  $u(t)$  é o sinal de controle proveniente do controlador,  $G(t)$  é a representação matemática do processo a ser controlado e  $y(t)$  é o sinal de saída resultante do processo de controle. Na Figura 19 é apresentado o esquema do IMC utilizado para a sintonia.

Figura 19 - Esquema IMC



Fonte: Campos; Teixeira (2006).

Desta forma chega-se à função de transferência em malha fechada do sistema, equação (20),

$$\frac{Y(s)}{SP(s)} = \frac{G_P(s)C(s)}{(1+G_P(s)C(s))} \quad (20)$$

e o controlador correspondente conforme a equação (21).

$$C(s) = K_c \left( 1 + \frac{1}{T_i(s)} + T_D(s) \right) \quad (21)$$

O IMC é utilizado para determinar a melhor sintonia do controlador  $C(s)$  de tal forma que a resposta do sistema a um degrau na referência de entrada seja representada por uma função de transferência resultante de primeira ordem em malha fechada com uma constante de tempo igual a  $\beta$ , conforme apresentado na equação (22).

$$\frac{Y(s)}{SP(s)} = \frac{1}{(\beta s + 1)} \quad (22)$$

Este parâmetro  $\beta$  é o critério de desempenho do IMC e define o quão rápido se deseja que a saída do processo acompanhe a referência estabelecida. É muito importante que este parâmetro seja escolhido respeitando as restrições dinâmicas do processo, como tempo morto, capacidade de resposta dos atuadores e aplicação.

### 2.3.2 Algoritmo Genético

O AG é uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca, fundamentado principalmente por Holland (1992).

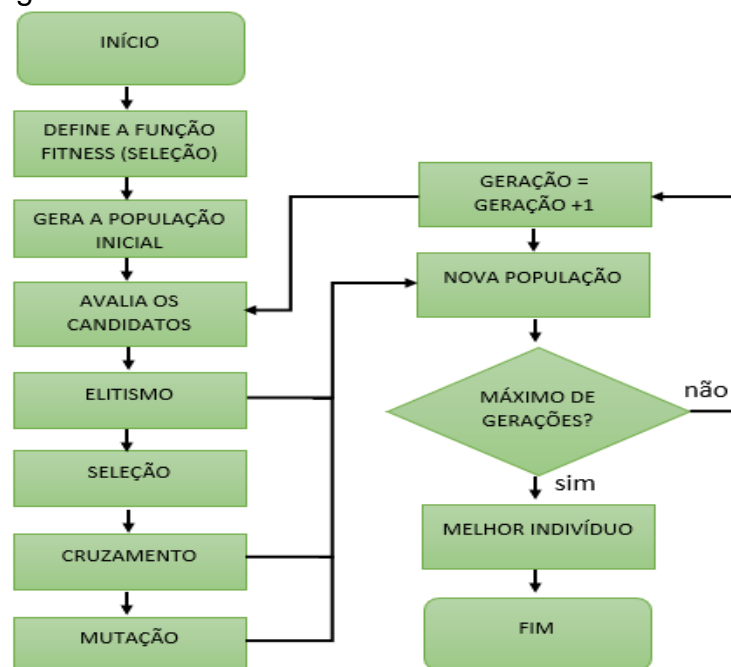
Sendo uma técnica já consolidada no campo da inteligência artificial, este método computacional baseia-se nos mecanismos de evolução das espécies onde os indivíduos mais fortes e mais adaptados ao meio ambiente sobrevivem e transmitem seu patrimônio genético às sucessivas gerações (ALMEIDA, 2011). Para isso são utilizadas técnicas como hereditariedade, mutação, seleção natural e recombinação ou *crossing over*.

O AG clássico usa um esquema de seleção individual chamada seleção por roleta. Nele cada indivíduo ocupa uma área proporcional ao seu nível de aptidão, dentro da "roleta". Assim, os indivíduos com maior aptidão estão associados com uma maior área e indivíduos com menor aptidão, com fatias menores. Em seguida, a simulação

roda  $n$  vezes, selecionando os  $n$  indivíduos que permanecerão no próximo passo. Os mais aptos têm maior chance de serem selecionados e também podem ser eleitos mais de uma vez, enquanto os menos aptos, com menor probabilidade, podem desaparecer após as primeiras gerações.

Um parâmetro fundamental do algoritmo é o estabelecimento da função de aptidão, usada para avaliar o nível de desempenho do sistema com os parâmetros selecionados, e como eles são capazes de resolver o problema estabelecido. Desta forma foi definida como função de aptidão do AG o IAE, também podendo ser chamada de função *fitness*. A seguir, na Figura 20, é representado o fluxo simplificado de execução de um AG genérico.

Figura 20 - Fluxograma AG



Fonte: Elaborado pelo autor

## 2.4 O FILTRO DE KALMAN

Em 1960, R.E. Kalman publicou seu famoso artigo descrevendo uma solução recursiva para o problema de filtragem linear de sinais discretos. Desde então,



devido, em grande parte, aos avanços na computação digital, o filtro de Kalman tem sido objeto de extensas pesquisas em aplicações, em especial para o desenvolvimento de sistemas para navegação autônoma (WELCH; BISHOP, 2001).

O filtro de Kalman é uma ferramenta matemática recursiva que vem desempenhando um papel cada vez mais importante em diversas áreas de pesquisas, principalmente por possibilitar, através de implementações computacionais, a redução de ruídos em sinais de processos, como também a redução de erros acumulados em cálculos de posição e deslocamentos que utilizam instrumentos, os quais podem acumular erros ao longo do tempo (WELCH; BISHOP, 2001).

O filtro de Kalman permite que através de implementações computacionais seja possível estimar o estado de um processo. Ele pode ser aplicado na resolução de problemas em diversas áreas como, na medicina, em processos químicos, na engenharia, petróleo e gás, indústria naval, siderurgia, mineração, entre outras.

#### **2.4.1 O Filtro de Kalman discreto**

Na década de 1960, após a publicação de um importante artigo, Rudolf Emil Kalman descreveu uma solução recursiva para um problema de filtragem linear de dados discretos. Após esse fato muitos avanços computacionais ocorreram, o que gerou um crescimento considerável das pesquisas e investigações acerca do assunto. Uma abordagem mais geral pode ser encontrada em (MAYBECK, 1979), enquanto uma discussão mais introdutória pode ser encontrada em (SORENSEN, 1970).

O filtro de Kalman trata da problemática geral de tentativa de estimação dos estados  $x \in R^n$  de um processo discretizado qualquer, baseado na equação (23) à diferença estocástica linear a seguir,

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (23)$$

onde  $A$  e  $B$  são matrizes oriundas da modelagem do processo em espaço de estados, e a variável aleatória  $w_k$  é o ruído desse mesmo processo. Tem-se também a medição do processo  $z_k$  que descreve a relação entre o estado do processo com o respectivo ruído, tal que seja  $Z \in R^m$ ,

$$z_k = Hx_k + v_k \quad (24)$$

onde  $H$  é uma matriz e a variável aleatória  $v_k$  representa o ruído da medição. Presume-se que sejam independentes uns dos outros e com probabilidade normal de distribuições.

$$p(w) \sim N(0, Q) \quad (25)$$

$$p(v) \sim N(0, R) \quad (26)$$

Na prática, as matrizes de covariância do ruído do processo  $Q$  e da medição  $R$  podem mudar com cada passo de tempo ou medição, no entanto aqui assume-se que elas são constantes.

A matriz  $A$  ( $n \times n$ ) na equação à diferença (23) relaciona o estado no instante de tempo anterior ( $k - 1$ ) com o estado no passo atual  $k$ , na ausência de outra função de acionamento ou ruído de processo. Observa-se que na prática  $A$  pode mudar com cada instante de tempo, mas no trabalho assume-se que ela será constante. A matriz  $B$  ( $n \times m$ ) relaciona a entrada de controle opcional com o estado  $u \in R$  no estado  $x$ . A matriz  $H$  ( $n \times m$ ) na equação da medição do processo (24) relaciona o estado com a medição  $z_k$ . Na prática  $H$  pode mudar a cada instante de tempo ou medição, mas aqui assume-se que será constante.

#### 2.4.2 As origens computacionais do filtro de Kalman

Define-se  $\hat{x}_k^- \in R^n$  como o estado a priori estimado no passo  $k$  dado o conhecimento do processo antes do passo  $k$ , e  $\hat{x}_k \in R^n$  sendo o estado estimado a posteriori no

passo  $k$  dada a medição  $z_k$ . Pode-se então definir os erros a priori e a posteriori sendo,

$$e_k^- \equiv x_k - \hat{x}_k^- \quad (27)$$

$$e_k \equiv x_k - \hat{x}_k \quad (28)$$

então a covariância do erro estimado a priori é,

$$P_k^- = E[e_k^- e_k^{-T}] \quad (29)$$

e a covariância do erro estimado a posteriori é,

$$P_k = E[e_k e_k^T] \quad (30)$$

onde  $E$  é uma matriz que armazenará os erros calculados no decorrer do processamento. Com o intuito de definirem-se as equações para o filtro de Kalman, é determinado o objetivo de encontrar-se uma equação que calcule um estado a posteriori  $\hat{x}_k$  como uma combinação linear a priori de um  $\hat{x}_k^-$  estimado e a diferença ponderada entre a medição atual  $z_k$  e uma predição de medição  $H\hat{x}_k^-$  como mostrado em (31).

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (31)$$

A diferença  $(z_k - H\hat{x}_k^-)$  em (31) chama-se de medição residual. Essa medição residual reflete a discrepância entre a medição prevista e a medição real do processo. Uma medição residual igual a zero, significa que as medições real e prevista estão completamente em acordo.

A matriz  $K$  ( $n \times m$ ) em (31) deve ser escolhida para ser o ganho do filtro ou para ser um fator de junção, a qual tem um papel de minimizar a covariância do erro a posteriori. Essa minimização pode ser obtida através das equações (31) e (28) e em seguida com (30). Realizando as manipulações e simplificações possíveis e igualando o resultado a zero, resolve-se a equação para o resultado de  $K$ . Um

detalhamento completo do método foi explorado por Maybeck (1979). A minimização da equação (30) é dada por.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (32)$$

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (33)$$

Da equação (33) pode-se notar que quando a covariância da medição do erro  $R$  aproxima-se de zero, o ganho  $K$  faz com que a parte residual tenha um peso mais significativo, conforme segue na equação (34).

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad (34)$$

Por outro lado, quando a covariância do erro estimado a priori  $P_k^-$  tende para zero, o ganho  $K$  faz com que a parte residual tenha um peso menos significativo, conforme segue na equação (35).

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (35)$$

Outra forma de abordar o valor estabelecido para o ganho  $K$  de forma resumida e de fácil compreensão é que, com a covariância do erro de medição  $R$  aproximando-se de zero, a medição atual  $z_k$  é tida como "confiável", enquanto que a medição estimada  $H\hat{x}_k^-$  é tida como menos confiável. Por outro lado, com a covariância do erro estimado  $P_k^-$  aproximando-se de zero, a medição atual  $z_k$  torna-se cada vez menos confiável, enquanto a medição estimada  $H\hat{x}_k^-$  torna-se cada vez mais confiável.

### 2.4.3 O algoritmo do filtro de Kalman discreto

O filtro de Kalman estima um processo usando uma forma de controle realimentada para a determinação do estado do processo em algum momento e, em seguida, obtém uma resposta sob a forma de medição (sinal inicialmente ruidoso).

Como tal, as equações para a implementação do filtro de Kalman dividem-se em dois grupos: equações de atualização de tempo e equações de atualização da medição. As equações de atualização de tempo são responsáveis por projetar para a frente (no tempo) as estimativas do estado atual e de covariância do erro para obtenção a priori para o próximo passo de tempo.

As equações de atualização de medição são responsáveis pela realimentação, isto é, por incorporar uma nova medição na estimativa a priori a fim de obter uma estimativa a posteriori melhorada.

As equações de atualização de tempo também podem ser pensadas como equações preditoras.

As equações de atualização da medição podem ser consideradas como equações de correção. Na verdade, o algoritmo de estimativa é semelhante ao de um algoritmo preditor / corretor para problemas numéricos, como mostrado na Figura 21.

Figura 21 - Algoritmos preditor / corretor



Fonte: Welch; Bishop (2006).

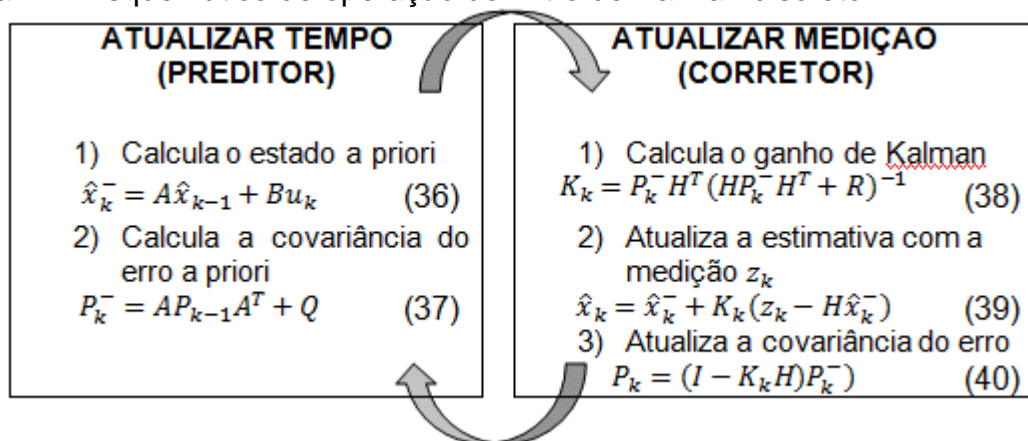
As equações específicas para a atualização do tempo são apresentadas nas equações (36) e (37)

Pode-se notar como as equações (36) e (37) projetam o estado e as estimativas da covariância a partir do passo do tempo  $k-1$  para o passo  $k$ .  $A$  e  $B$  são referentes à equação (23), enquanto que  $Q$  é referente à equação (25). Abaixo seguem as equações atualizadas das medições para o filtro de Kalman.

A primeira tarefa durante a atualização de medição é calcular o ganho de Kalman  $K_k$ . Nota-se que a equação dada anteriormente como equação (32) é a mesma que a equação (38). O próximo passo agora é medir o processo para obter  $z_k$ , e em seguida, gerar um estado estimado a posteriori, assim incorporando a medição realizada na equação (39). Novamente a equação (36) é a mesma equação (31), repetida aqui para complementar as deduções e confirmações. O passo final é atualizar a covariância do erro a posteriori através da equação (40).

Após cada atualização da medição e do tempo, o processo é repetido com a previsão estimada a posteriori usada para estimar ou prever as novas estimativas a priori. Esta natureza recursiva é uma das características mais destacadas do filtro de Kalman, o que torna as suas implementações muito mais interessantes do que outros filtros com propósitos similares. A Figura 22 mostra um esquemático completo de operação do filtro de Kalman, combinando o diagrama de alto nível da Figura 21.

Figura 22 - Esquemático de operação do Filtro de Kalman discreto

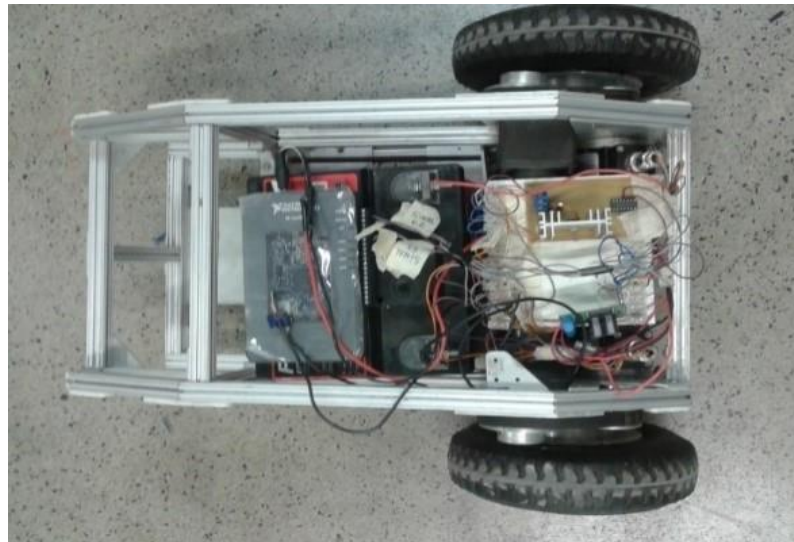


Fonte: Welch; Bishop (2006).

### 3 O ROBÔ MÓVEL DIFERENCIAL

Considerando a grande necessidade em relacionar a teoria de controle com a prática e devido à grande relevância da robótica nos cursos tecnológicos e aplicações industriais, foi proposta a utilização da robótica móvel para mostrar a aplicação de controladores PID estudados. O robô móvel diferencial utilizado segue apresentado na Figura 23, e foi projetado e construído no GAln por alunos de graduação em engenharia de controle e automação do IFES Serra.

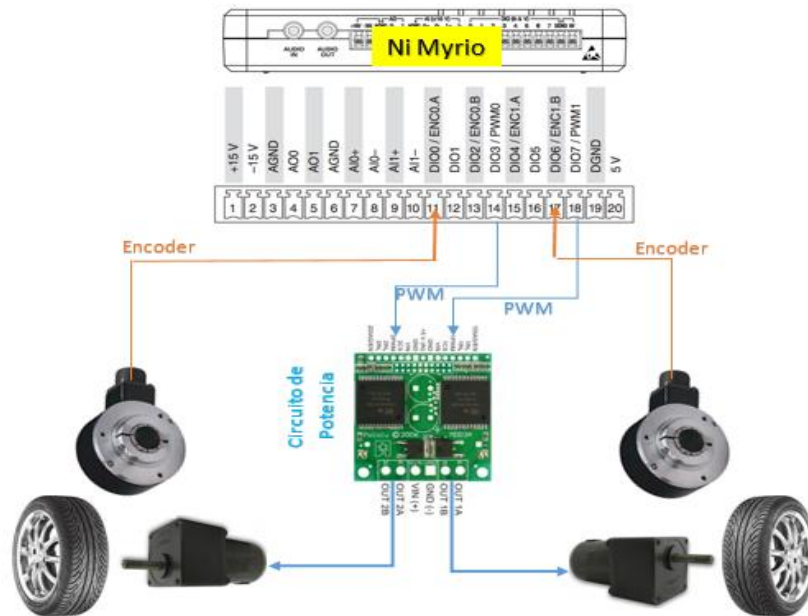
Figura 23 - Robô móvel diferencial



Fonte: Elaborado pelo autor

Na Figura 24 segue o esquemático do robô móvel com os encoders, motores de acionamentos de corrente contínua, circuito de potência e uma placa NI myRIO, que será descrita à frente.

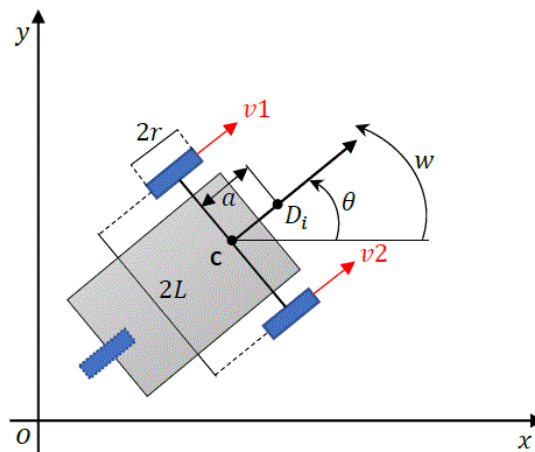
Figura 24 - Esquemático do robô móvel



Fonte: Bernardes et al. (2017)

O robô possui duas rodas motoras paralelas de tração independentes uma da outra e uma roda livre. Como segue mostrado na Figura 25, as duas rodas motoras são idênticas, com raio  $r$ , sendo a distância entre as rodas igual a  $2L$ . A posição do robô, definida no sistema de eixos cartesianos  $xy$ , é definida por  $x$  e  $y$ . O centro de massa do robô móvel está localizado no ponto  $C$ , que fica no centro do eixo das rodas, com distância  $L$  uma da outra. Esse ponto será utilizado como referência, ou ponto de interesse para o controle do robô.

Figura 25 - Modelo do robô móvel de tração diferencial



Fonte: Tommasi et al. (2015).



Caso as velocidades das duas rodas sejam idênticas ( $v_1=v_2$ ) o deslocamento do robô será em linha reta. Sendo a velocidade da roda direita ( $v_2$ ) menor do que a da roda esquerda ( $v_1$ ) o robô se deslocará em forma circular para a direita. Sendo a velocidade da roda direita ( $v_2$ ) maior, o robô se deslocará também em forma circular, mas para a esquerda. Contudo, o robô descrito não pode executar movimentos transversais, sendo necessário executar uma sequência de manobras para executar movimentos mais complexos. Este tipo de robô com restrições de movimento é denominado de não holonômico. A equação (41) representa o modelo cinemático do robô móvel. Este modelo será utilizado no desenvolvimento do controlador,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \quad (41)$$

onde  $v$  e  $w$  são as velocidades, linear e angular do robô, respectivamente. Ressalta-se que em tal modelo considera-se como a posição do robô o ponto de interesse C. O modelo descrito também pode ser representado usando as equações (42) a (45).

$$v_1 = w_1 r \quad (42)$$

$$v_2 = w_2 r \quad (43)$$

$$v = \frac{v_1 + v_2}{2} \quad (44)$$

$$w = \frac{v_2 - v_1}{L} \quad (45)$$

Onde  $w_1$  e  $w_2$  são as velocidades angulares das rodas direita e esquerda.

### 3.1 DESCRIÇÃO DO SISTEMA DE ACIONAMENTOS

O sistema de acionamento do robô móvel utilizado é formado por duas rodas simétricas de 111 milímetros de raio constituídas por um aro em alumínio e na parte externa um pneu de borracha dimensionado para a aplicação. As duas rodas descritas são montadas no chassi do robô, as quais através de um sistema de

engrenagens, são acopladas a dois motores de corrente contínua, Figura 26, com características segundo a Tabela 2.

Figura 26 - Motor DC do robô



Fonte: Elaborado pelo autor

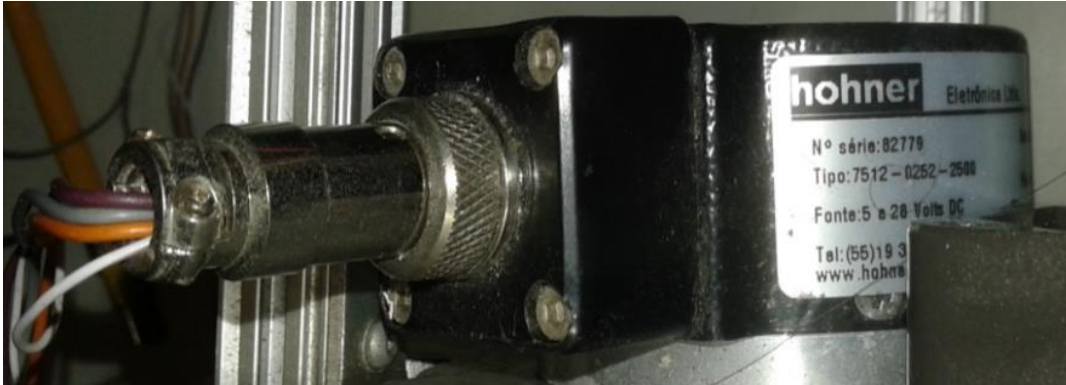
Tabela 2 - Características do motor do robô

Tipo	Motor DC
Fabricante	LINIX Motor – HENGDIAN GROUP
Modelo	55ZY12-25-02/60JB30GO832
Alimentação	12 Vcc
Potência	25W
Rotação	100 rpm
Acoplamento	Frangue / Pinhão

Fonte: Elaborado pelo autor

Além dos motores, utilizados para imprimirem movimento ao conjunto, há dois encoders incrementais, Figura 27, acoplados diretamente às transmissões para a geração de pulsos no decorrer do movimento do robô com as características descritas na Tabela 3.

Figura 27 - Encoder do robô



Fonte: Elaborado pelo autor

Tabela 3 - Características do encoder do robô

Tipo	Incremental
Fabricante	HOHNER Eletrônica
Modelo	7512-0252-2500
Alimentação	5 a 28 Vcc
Pulsos por volta	1800
Acoplamento	Eixo vazado

Fonte: Elaborado pelo autor

### 3.2 PLACA NI MYRIO

Os sinais provenientes dos encoders e as saídas dos sinais PWM para acionamento dos motores de cada roda são recebidos por uma placa microcontroladora NI myRIO embarcada da *National Instruments*, que é um dispositivo embarcado muito utilizado para projetos de estudantes. Possui entradas e saídas nas laterais do dispositivo, com conectores MXP e MSP. São disponibilizadas também entradas analógicas, saídas analógicas, linhas de entradas e saídas digitais, acelerômetro integrado, uma FPGA Xilinx e um processador dual-core ARM Cortex-A9. Alguns modelos também contam com suporte a Wi-Fi, sendo o caso do dispositivo utilizado no robô móvel diferencial. Pode-se programá-lo com o LabVIEW ou em C. Com seus dispositivos integrados, experiência consistente de software e biblioteca de materiais didáticos e

tutoriais, o myRIO oferece uma excelente ferramenta acessível para estudantes e educadores no desenvolvimento de experimentos práticos de laboratório. A seguir na Figura 28 é apresentado o NI myRIO utilizado.

Figura 28 - NI myRIO



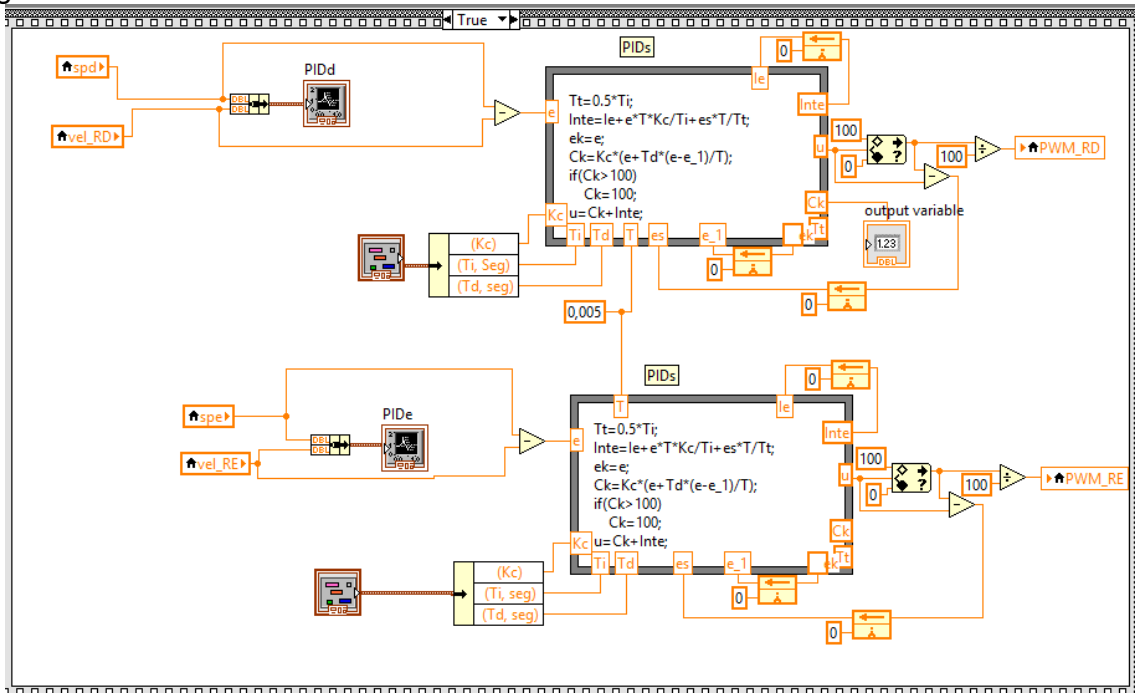
Fonte: National Instruments (2017)

### 3.3 CONTROLADOR PID ORIGINAL DAS RODAS DO ROBÔ MÓVEL

Considerando o objetivo principal do trabalho, já definido em tópicos anteriores, foi adotado como ponto de partida um controlador PID implementado para controle de velocidade das duas rodas do robô móvel diferencial utilizado, que também foi utilizado por Tommasi (2015). Os dois controladores possuem a configuração na forma ideal, equação (18), e foram programados em LabView pelos autores. A sintonia desses dois controladores tradicionais também será objeto de desenvolvimento dos trabalhos futuros.

Na Figura 29 seguem os dois controladores implementados no ambiente de programação do LabView.

Figura 29 - Controladores PID's tradicionais das rodas



Fonte: Elaborado pelo autor

### 3.4 ODOMETRIA

A odometria é um método muito utilizado para estimar a posição e a orientação de robôs móveis através da integração dos deslocamentos incrementais de suas rodas, medidos a partir de um referencial fixo. Os encoders mais utilizados como transmissores de deslocamento são foto codificadores acoplados diretamente aos eixos dos motores ou engrenagens. Como a ideia fundamental da odometria é a integração contínua e incremental da informação que descreve o movimento ao longo do tempo, é inevitável o acúmulo de erros locais a cada iteração (FERNANDES, 2003).

Segundo a teoria abordada por Freitas et al. (2016), para especificar a posição do robô, na Figura 25 foi escolhido o ponto  $C$ , que a partir de agora será chamado de ponto  $P$ . O ponto  $P$ , que é o ponto médio entre as rodas, é utilizado como ponto de referência da posição. A posição  $P$  pode ser descrita pelas coordenadas  $x$  e  $y$  e pelo ângulo  $\theta$  ( $P = [x \ y \ \theta]$ ). A distância de cada pulso é calculada através da equação (46),

$$DistPulsos = 2\pi R/NP \quad (46)$$

onde  $R$  é o raio da roda e  $NP$  é a quantidade de pulsos por uma volta. Posteriormente é calculada a distância percorrida por cada roda através das equações (47) e (48),

$$\Delta S_d = DistPulsos * P_d \quad (47)$$

$$\Delta S_e = DistPulsos * P_e \quad (48)$$

onde  $\Delta S_d$  e  $\Delta S_e$  são as distâncias percorridas no intervalo  $\Delta t$  das rodas direita e esquerda, respectivamente e  $P_d$  e  $P_e$  são a quantidade de pulsos medidos através dos encoders das rodas direita e esquerda respectivamente. A distância percorrida pelo ponto médio do robô é calculada através das equações (49) e (50),

$$\Delta S = \frac{\Delta S_d + \Delta S_e}{2} \quad (49)$$

$$\Delta \theta = \frac{\Delta S_d - \Delta S_e}{L} \quad (50)$$

onde  $L$  é a distância entre o ponto médio  $C$  até cada roda. Em seguida são calculados os deslocamentos nos eixos  $x$  e  $y$  no intervalo de tempo  $\Delta t$  através das equações (51) e (52).

$$\Delta x = \Delta S \cos[(\theta + \Delta \theta / 2)] \quad (51)$$

$$\Delta y = \Delta S \sin[(\theta + \Delta \theta / 2)] \quad (52)$$

Assim, a partir das variações de deslocamento e de  $\theta$ , pode-se calcular a posição atualizada ( $P'$ ) através da equação (53).

$$P' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta S * \cos(\theta + \frac{\Delta \theta}{2}) \\ \Delta S * \sin(\theta + \frac{\Delta \theta}{2}) \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} x + \Delta x \\ y + \Delta y \\ \theta + \Delta \theta \end{bmatrix} \quad (53)$$

A odometria é então utilizada para estimar a pose do robô e por consequência as velocidades das rodas. Nela os sensores e encoders óticos instalados no robô transmitem os dados de rotação das rodas para o sistema de controle no NI myRIO. Assim, através da equação (54) é determinada a velocidade de cada roda do robô,

$$Vr = \frac{|Pi| * Pr}{Pv * t} \quad (54)$$

onde  $Pi$  é o número de pulsos instantâneos das rodas,  $Pv$  é a quantidade de pulsos por volta pré-determinada,  $Pr$  é o perímetro de cada roda e  $Vr$  é a velocidade de cada roda.

### 3.4.1 Dead reckoning

A odometria abordada no tópico anterior permite boa precisão em curtas distâncias e geralmente pode ser implementada com baixo custo. Entretanto devido à integração incremental do movimento do robô, ela leva ao acúmulo de erros proporcionais à distância navegada.

Os erros acumulados na odometria de um robô móvel podem surgir pelo simples fato da superfície do deslocamento possuir alguma imperfeição ou contaminante que possa prejudicar a aderência das rodas ao chão, ocasionando derrapagens indesejadas. Outros fatores como diâmetro desiguais das rodas, desalinhamentos, resolução do encoder utilizado, taxa de amostragem finita podem contribuir para o acúmulo de erros.

O termo *dead reckoning* surgiu na área da engenharia de aviação como um meio de determinar as coordenadas de uma aeronave. Para tal o piloto utilizava-se de uma bússola, da velocidade da aeronave e o tempo de voo. Esta técnica é bastante útil em pequenos deslocamentos, porém, ela fica comprometida devido a vários fatores como derrapagens, modelagem errada, intempéries (ROMERO et al., 2014).

## 3.5 CONTROLADOR DE TRAJETÓRIA

Considerando as diferentes estratégias para a solução de problemas acerca do controle de trajetória em robôs móveis encontradas na literatura, é possível tratar o

assunto de forma mais simples comparado aos problemas inerentes à estabilização de malhas de controle, uma vez que a trajetória gerada respeite as restrições não-holonômicas do robô. Assim, de posse do erro em relação ao tempo, entre a posição do robô e uma referência móvel a ser seguida, é possível a implementação de um algoritmo que execute a correção automaticamente (VIEIRA, 2005).

Como exemplos de controladores de trajetória utilizados na robótica móvel destacam-se o controlador cinemático de seguimento de trajetória de ganhos fixos (MARTINS et al., 2008), (RESENDE et al., 2012), o controlador *backstepping* (FIERRO; LEWIS, 1998), (YE, 2008) e o controlador *fuzzy* do tipo Mandani próprio.

A escolha do controlador de trajetória aplicado neste trabalho baseou-se na pesquisa realizada por Tommasi (2015) que definiu como melhor solução para o controle de trajetória do robô com as características já descritas anteriormente, o *backstepping*.



## 4 IMPLEMENTAÇÃO

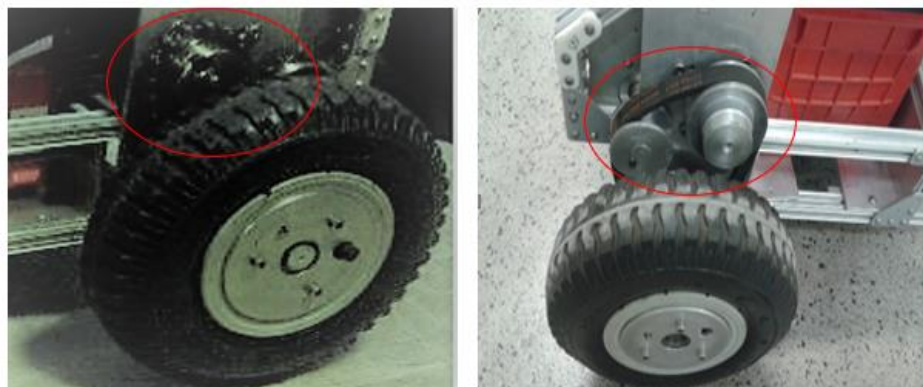
Para que os testes práticos pudessem ser realizados, tornou-se necessária a implementação dos controladores e também a realização de adequações estruturais do robô móvel diferencial utilizado como planta de testes. Nos tópicos a seguir serão apresentadas essas etapas.

### 4.1 ADEQUAÇÕES REALIZADAS NO ROBÔ

Para uma adequada implementação dos controladores PIDs nos sistemas de acionamentos do robô móvel diferencial utilizado, foram necessárias algumas adequações na estrutura, placa eletrônica e cabeamento, como também algumas melhorias nas implementações dos programas via LabView.

Inicialmente foi avaliado o sistema de transmissão de movimento, sendo este concebido originalmente por acoplamentos via correntes, o que foi melhorado a partir da montagem de duas polias interligadas via correias em cada roda. O sistema original gerava grande folga no conjunto, o que acarretava em erros de medição além de ruídos. Na Figura 30 é apresentado um detalhe do acionamento do robô antes com correntes e depois via polias e correias após adequações.

Figura 30 - Mudança no acionamento de corrente do robô para polias

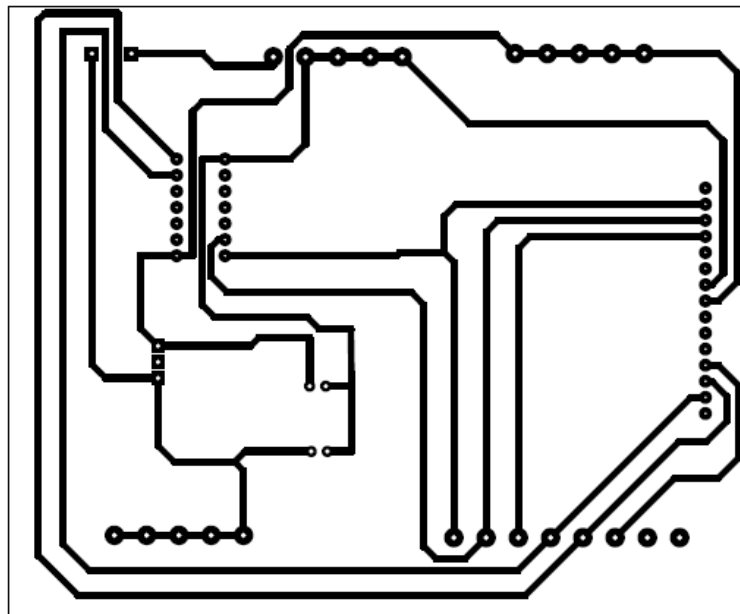


Fonte: Elaborado pelo autor

Após as adequações no sistema de acionamento, foi identificado que a placa dos circuitos eletrônicos de interface entre os sensores embarcados, motores de corrente contínua precisavam ser melhoradas. Assim, foi substituído o elemento controlador original, microcontrolador programável PIC 18F4550, por uma placa NI myRIO, o que trouxe maior confiabilidade e melhor interface entre os dispositivos embarcados, diminuindo as possibilidades de maus contatos, o que poderia gerar erros de medição e também falhas intempestivas.

Uma das etapas de melhorias foi realizada através do levantamento detalhado das interligações existentes, o que possibilitou a confecção de uma nova placa eletrônica, conforme segue apresentado na Figura 31.

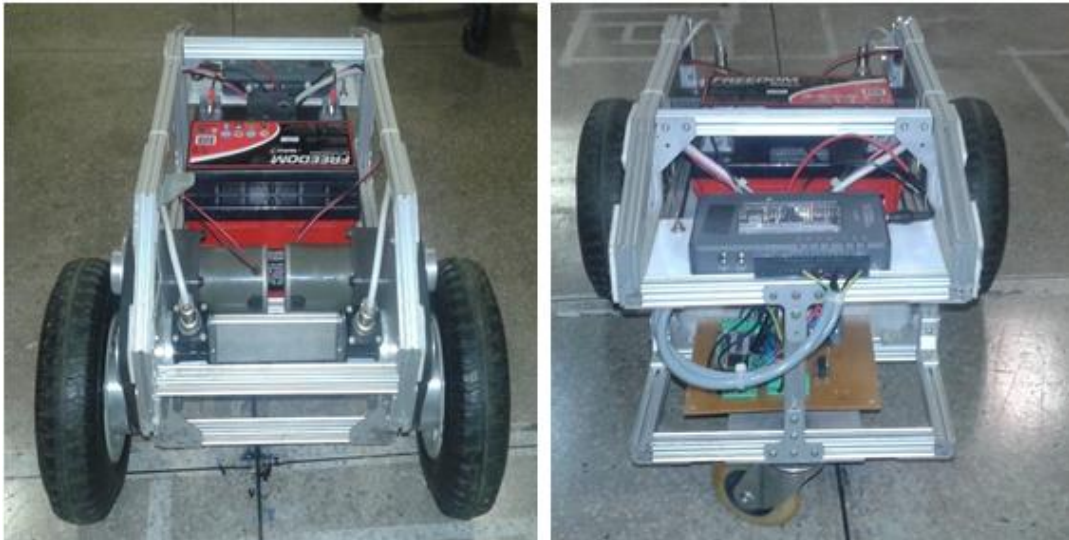
Figura 31 - Circuito da placa do robô



Fonte: Elaborado pelo autor

Em seguida foi realizada a reorganização dos circuitos com a inclusão da nova placa eletrônica, novas rotas de cabos e também foi substituída a bateria de 12V da fonte de energia do robô, como apresentado na Figura 32.

Figura 32 - Robô após melhorias



Fonte: Elaborado pelo autor

## 4.2 SOFTWARES UTILIZADOS

Para a programação dos ambientes de simulação e também para a implementação dos PIDs e Filtro de Kalman utilizados na pesquisa foram utilizados dois softwares de programação, o MATLAB e o LabView, ambos muito conhecidos e aplicados em desenvolvimento de projetos e diversos estudos acadêmicos, devido à facilidade de interação entre programador e o ambiente de programação.

O MATLAB (Matrix Laboratory) é um software interativo de alta performance voltado para o cálculo numérico, muito utilizado em laboratórios de pesquisas e também em ambientes industriais. Devido a sua forma de integrar a análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em um ambiente de fácil aplicação, é possível a utilização tanto para modelamentos de processos como também para a programação de algoritmos complexos, tendo a grande vantagem de disponibilizar para os usuários uma série de funções que facilitam e simplificam a programação necessária, ao contrário da programação tradicional.

O LabView (Laboratory Virtual Instrument Engineering Workbench) é uma linguagem de programação gráfica originária da National Instruments. A primeira versão surgiu em 1986 para o Macintosh e atualmente é bem aplicado em Windows. O LabView é

vastamente utilizado no meio acadêmico como também no meio industrial e profissional como solução para medições de variáveis de processos, programações de controladores e integração com o processo, execução de cálculos de alta complexidade entre outros. Devido à possibilidade de interface com o MATLAB, a sua utilização traz grandes possibilidades em pesquisas relacionadas com a área de controle e automação.

A programação é feita de acordo com o modelo de fluxo de dados, o que oferece a esta linguagem vantagens para a aquisição de dados e para a sua manipulação, e se baseia na utilização de instrumentos virtuais ou simplesmente VIs. É possível criar uma interface visual através de uma tela de supervisão com o diagrama de programação gerado, o que traz maior versatilidade ao aplicativo.

#### 4.3 PID TRADICIONAL (INTEIRO)

A forma do PID que foi implementado seguiu a configuração ideal, conforme equação (18), que pode ser reescrita conforme segue apresentado na equação (55).

$$u(k) = K_c \left[ e(k) + \frac{T}{T_i} \sum_{i=1}^k e(i) + T_d \left( \frac{e(k) - e(k-1)}{T} \right) \right] \quad (55)$$

Quando um controlador com ação integral é usado e os atuadores chegam a seus limites, a parte integral continuará sendo acumulada sem provocar nenhuma mudança no atuador. Quando o erro mudar de sentido (negativo para positivo ou vice e versa) a parte integral provocará um atraso na ação do controle. Devido à esta característica conhecida como *Wind-up*, o sistema tende a ficar lento e oscilatório.

Para evitar esta limitação, como na maioria dos controladores foi utilizada a técnica *back calculation and tracking*. Basicamente quando a saída do atuador satura, o termo integral é recalculado de forma que seu valor permaneça no valor limite do atuador. É vantajoso fazer esta correção não instantaneamente, mas dinamicamente

com uma constante de tempo  $T_f$ . A Figura 33 apresenta o diagrama em blocos de um controlador PID com anti *wind-up* do tipo *back calculation*.

O sistema apresenta um laço de realimentação adicional. A diferença entre o valor da entrada e da saída do atuador constituem um erro  $e_s(t)$  que é realimentado à entrada do integrador com um ganho  $1/T_f$ . Nota-se que quando não há saturação do sinal de controle, o erro  $e_s(t)$  é igual a zero e, portanto, este laço não tem nenhum efeito quando o controlador está operando linearmente, ou seja, quando sua saída não está saturada. Quando ocorre a saturação,  $e_s(t)$  será diferente de zero e o sinal aplicado na entrada do integrador será,

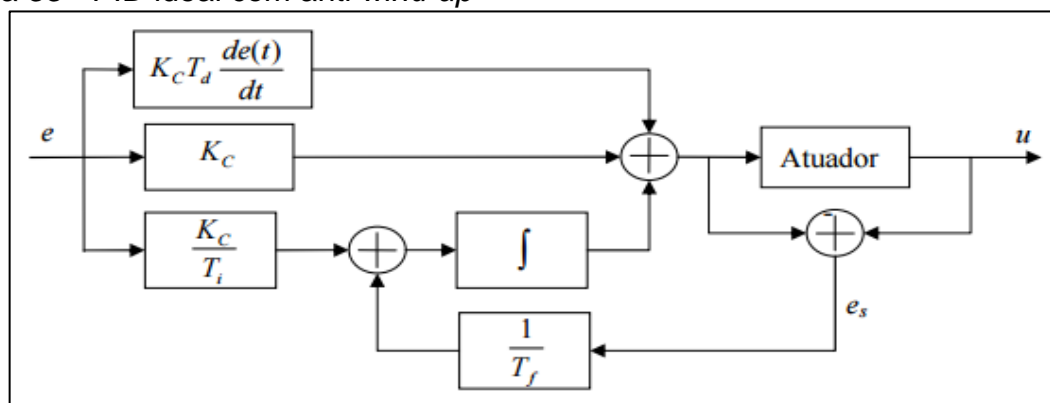
$$\frac{1}{T_f} e_s(t) + \frac{K_c}{T_i} e(t) \quad (56)$$

sendo que, em regime permanente, tem-se:

$$e_s(t) = -\frac{K_c T_f}{T_i} e(t) \quad (57)$$

ou seja, a entrada do integrador será igual a zero prevenindo assim que o mesmo se carregue demasiadamente. O tempo para que a entrada do integrador chegue a zero é determinado pelo ganho  $1/T_f$ , onde  $T_f$  pode ser interpretado como a constante de tempo que determina o quão rápido a entrada do integrador será levada a zero. Assim, a escolha de valores bem pequenos para  $T_f$  pode parecer vantajosa à primeira vista. Entretanto, deve-se ter cuidado na escolha de  $T_f$  especialmente em sistemas com ação derivativa.

Figura 33 - PID ideal com anti *wind-up*



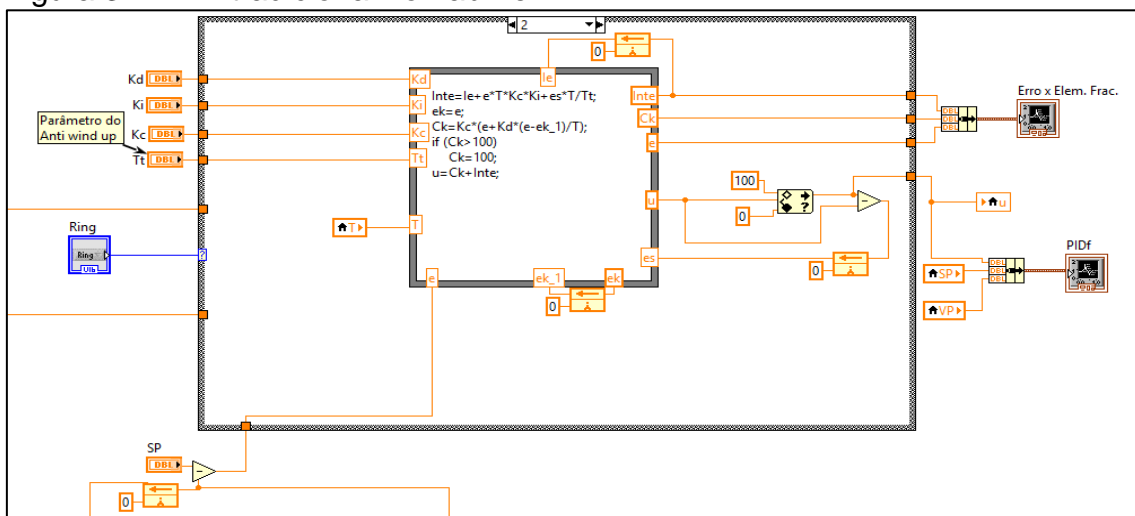
Fonte: Elaborado pelo autor

Considerando a configuração do PID com anti *wind-up*, a equação (55) foi reescrita, conforme exposto através da equação (58).

$$u(k) = K_c \left[ e(k) + T_d \left( \frac{e(k) - e(k-1)}{T} \right) \right] + T \sum_{i=1}^k \left( \frac{K_c}{T_i} e(i) + \frac{1}{T_f} e_s(i) \right) \quad (58)$$

Desta forma foram avaliados os blocos disponíveis no LabView e realizada a melhor forma de ser programado, conforme segue na Figura 34.

Figura 34 - PID tradicional no LabView



Fonte: Elaborado pelo autor

Para que houvesse uma similaridade entre os controladores implementados, optou-se por utilizar um bloco de script do LabView para a programação do PID.

#### 4.4 PID FRACIONÁRIO

De acordo com o que já foi abordado na teoria do cálculo fracionário no capítulo 2, estabeleceu-se como estratégia de implementação do controlador PID fracionário, as definições matemáticas de Grunwald-Letnikov e Riemann-Liouville, que conforme exposto por Podlubny (1999), são equivalentes para uma vasta gama de aplicações de engenharia. Desta forma, a implementação do controlador PID fracionário foi

realizada empregando-se o software LabView, da *National Instruments*, usando-se como referência o trabalho realizado por Auralius (2010).

Para a implementação foi utilizado o método para definição de derivadas fracionárias desenvolvido por Grundwald-Letnikov, e que já foi exposto no tópico 2.1.1. Portanto, reescrevendo as equações para obtenção do cálculo numérico da ordem fracionária, tem-se,

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} \left\{ \frac{1}{h^\alpha} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^j \binom{\alpha}{j} f(t-jh) \right\} = \lim_{h \rightarrow 0} \left\{ \frac{1}{h^\alpha} \Delta_h^\alpha f(t) \right\} \quad (59)$$

$${}_{(t-L)} D_t^\alpha f(t) \approx T^{-\alpha} \sum_{j=0}^{\lfloor N(t) \rfloor} b_j f(t-jT) \quad (60)$$

onde  $L$  é chamado de tamanho da memória,  $N(t) = \min \left\{ \left\lfloor \frac{t}{T} \right\rfloor, \left\lfloor \frac{L}{T} \right\rfloor \right\}$ , e o termo  $b_j$ , chamado de coeficiente binomial pode ser calculado recursivamente por:

$$b_0 = 1; b_j = \left( 1 - \frac{1+\alpha}{j} \right) b_{j-1}, \text{ sendo } j \geq 1. \quad (61)$$

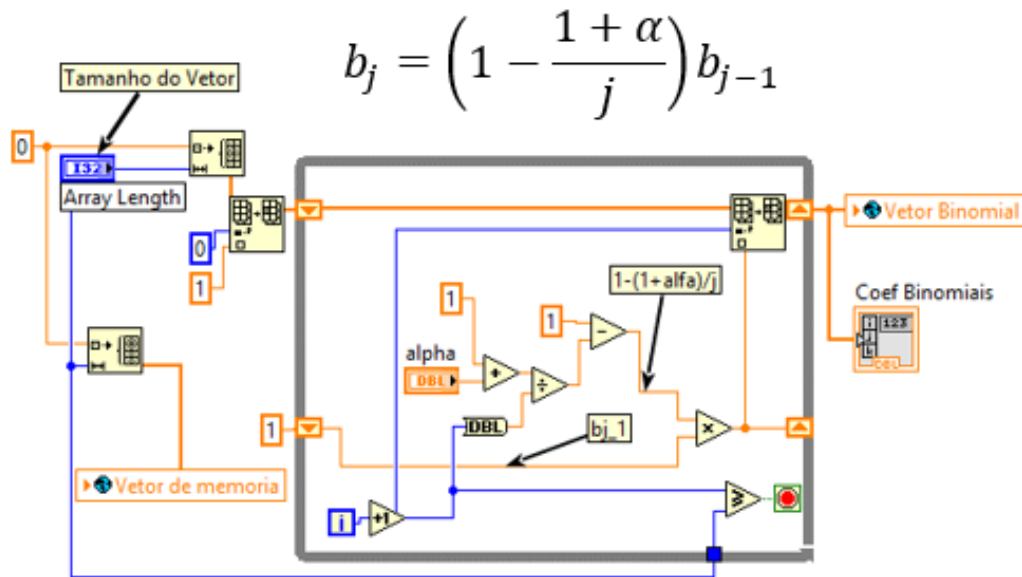
Dado o método de solução para uma equação de ordem fracionária mostrada acima, também se faz necessária a verificação do emprego de Transformada de Laplace, que assim como para equações diferenciais de ordem inteira, também são aplicáveis às de ordens fracionárias, conforme exposto por Podlubny (1994). Assumindo-se que as condições iniciais da função  $f(t)$  sejam iguais a zero, chega-se às transformadas de Laplace associadas ao cálculo fracionário.

$$L \left\{ {}_a D_t^\alpha f(t) \right\} = s^\alpha F(s) \quad (62)$$

$$L \left\{ {}_a D_t^{-\alpha} f(t) \right\} = s^{-\alpha} F(s) \quad (63)$$

Desta forma, como ponto de partida para a implementação no LabView, considerando  $a$  e  $t$  os limites da operação, faz-se necessário o cálculo dos coeficientes binomiais representado na equação (61), conforme ilustrado na Figura 35.

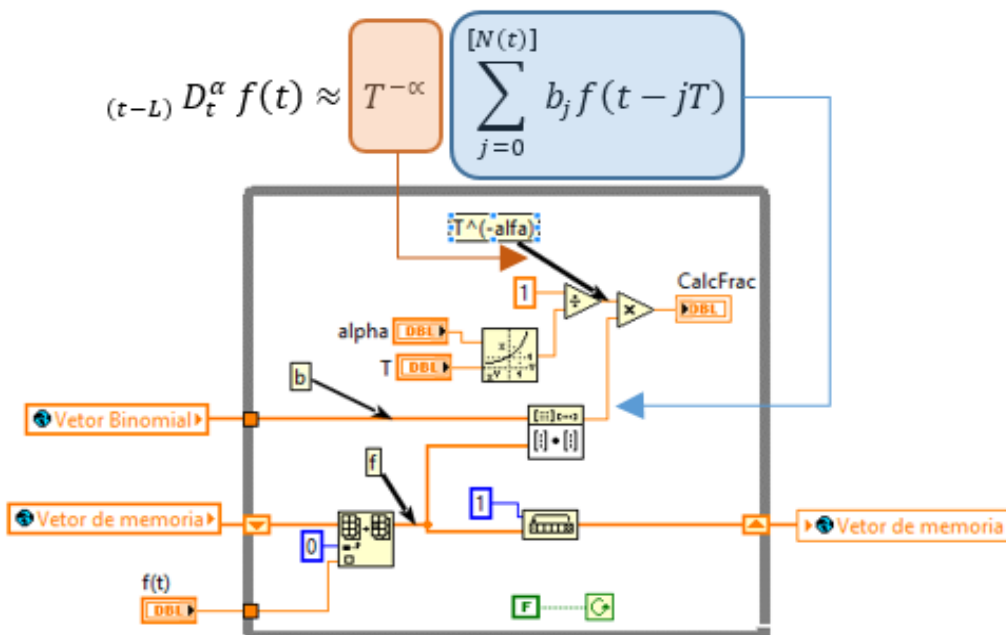
Figura 35 - Cálculo do binômio do PID fracionário



Fonte: Elaborado pelo autor

Após o cálculo dos coeficientes binomiais procede-se com o cálculo do operador fracionário:  $(t-L)D_t^\alpha f(t)$ , representado na equação (60) e mostrado na Figura 36.

Figura 36 - Cálculo do operador fracionário

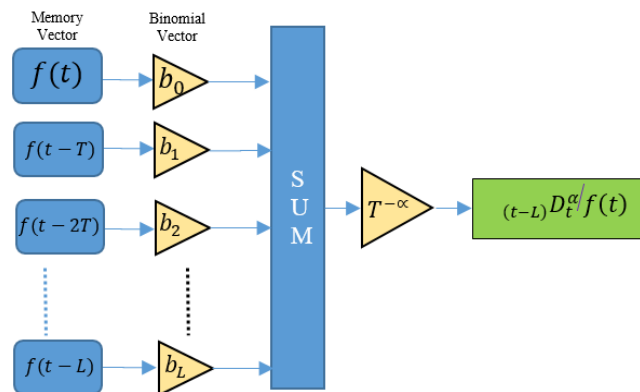


Fonte: Elaborado pelo autor



A Figura 37 mostra as operações necessárias para se obter o cálculo fracionário de uma função  $f(t)$ . Assim é necessário ter uma memória de tamanho  $L$  para armazenar os valores passados de  $f(t)$ , que serão multiplicados pelos coeficientes binomiais, posteriormente somados e multiplicados por  $T^{-\alpha}$ .

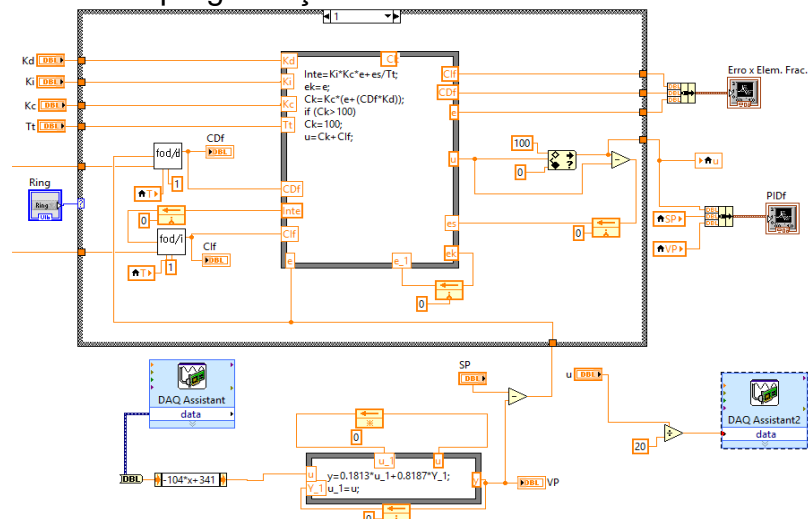
Figura 37 - Cálculo Fracionário em blocos



Fonte: Elaborado pelo autor

Na Figura 38 é ilustrado o programa implementado no LabView, para o PID fracionário onde foi criado um vetor binomial e de memória para cada ação de controle (integral e derivada), com a única diferença do valor de  $\alpha$ .

Figura 38 - Ambiente de programação do LabView



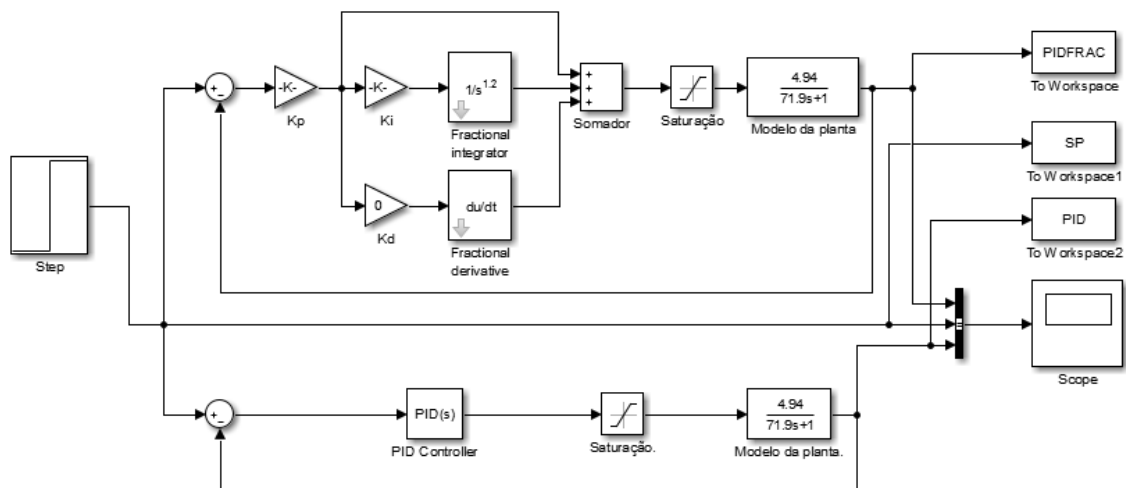
Fonte: Elaborado pelo autor

Importante notar que foram estabelecidos limites para o sinal de controle  $u$  (0 e 100), para que seja contornada a chamada saturação do sinal de controle. Desta forma, caso a saída do controlador exceda o máximo do sinal de controle, o termo integrador passa a ser reduzido com uma taxa de variação ajustada pelo parâmetro  $T_f$ .

#### 4.5 VALIDAÇÃO DOS PIDS IMPLEMENTADOS

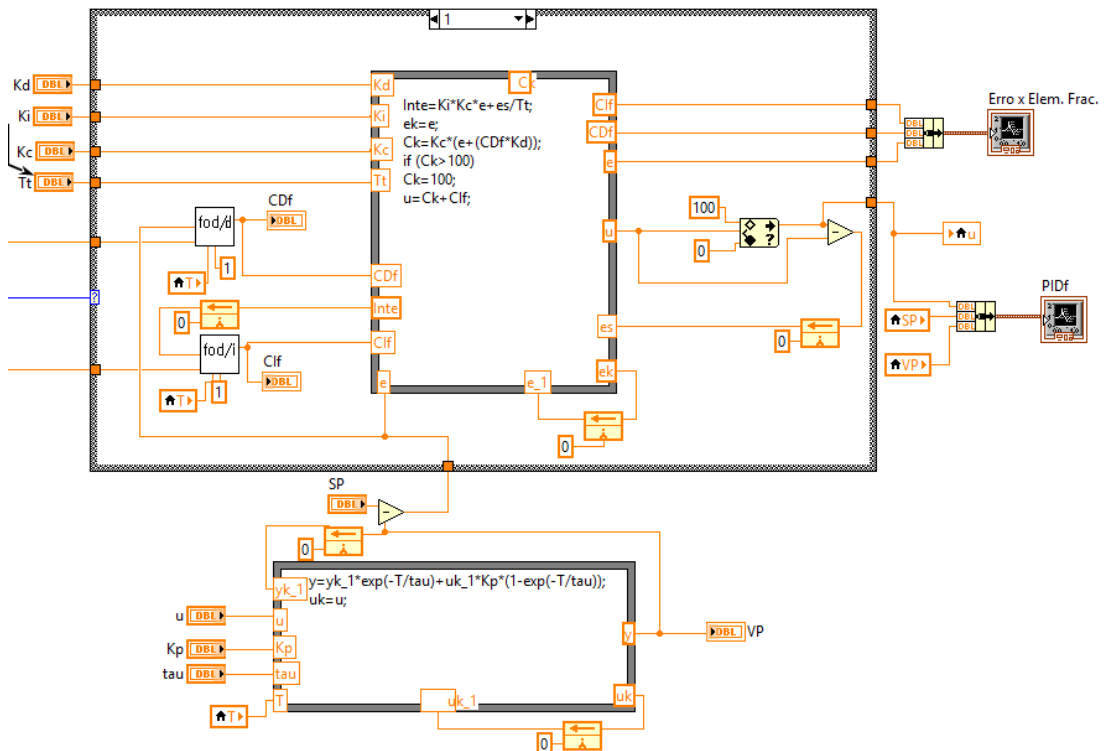
A fim de comprovação dos métodos utilizados e da programação realizada no LabView para a implementação dos PIDs inteiro e fracionário, foram comparadas as respostas ao degrau de um processo hipotético definido através da função de transferência  $G(s) = 4,94 / (71,9s+1)$  simulados no Simulink do Matlab com as programações realizadas no LabView. Nas Figuras 39 e 40 seguem os ambientes de programação utilizados para as validações.

Figura 39 - Tela de programação Matlab



Fonte: Elaborado pelo autor

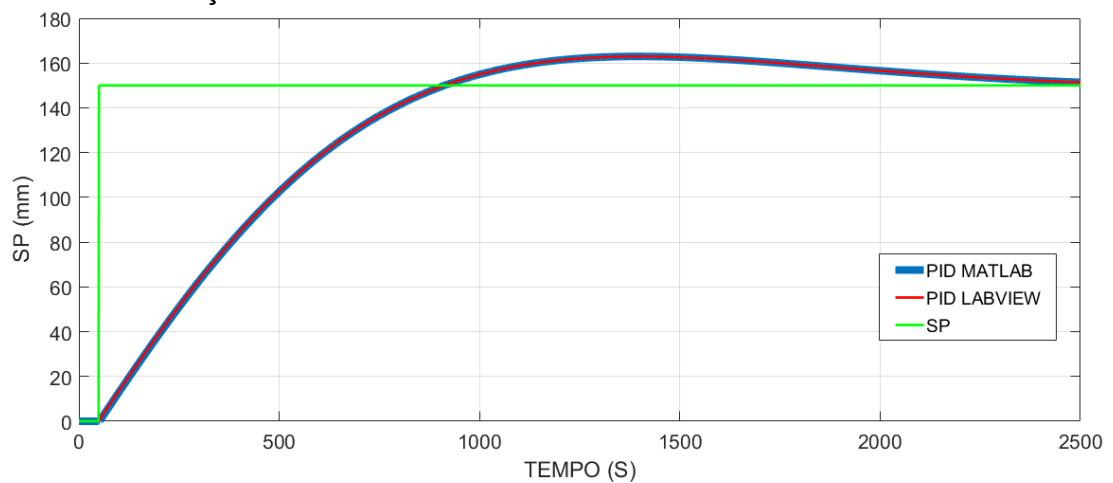
Figura 40 - Tela programação LabView



Fonte: Elaborado pelo autor

Através de um degrau de 0 para 150 mm, foram obtidas as seguintes respostas ao degrau.

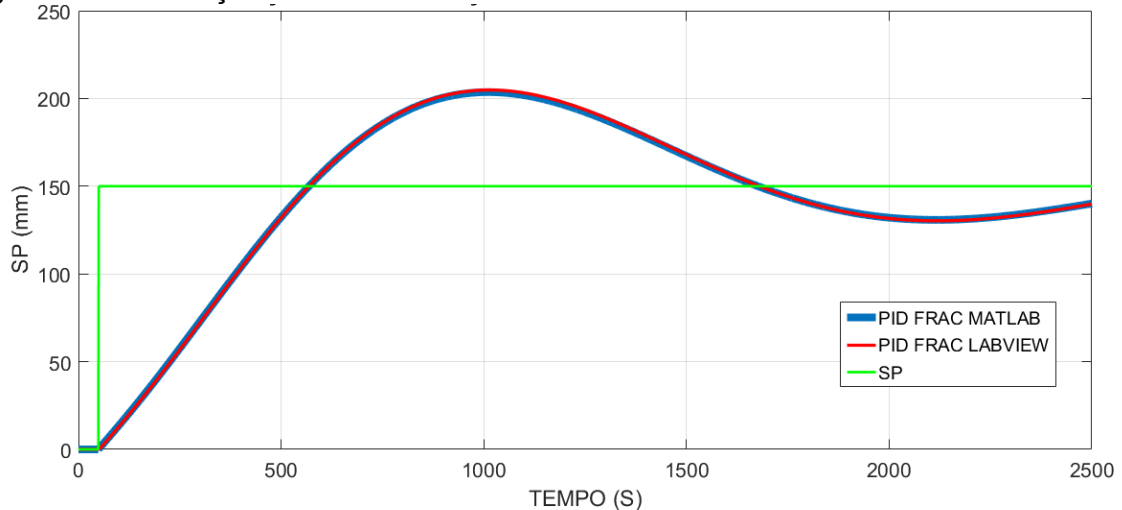
Figura 41 - Validação PID Matlab e LabView



Fonte: Elaborado pelo autor

Observando a Figura 41 é possível confirmar que a implementação do PID no LabView está condizente com os blocos PIDs do Simulink.

Figura 42 - Validação PID fracionário no Matlab e LabView



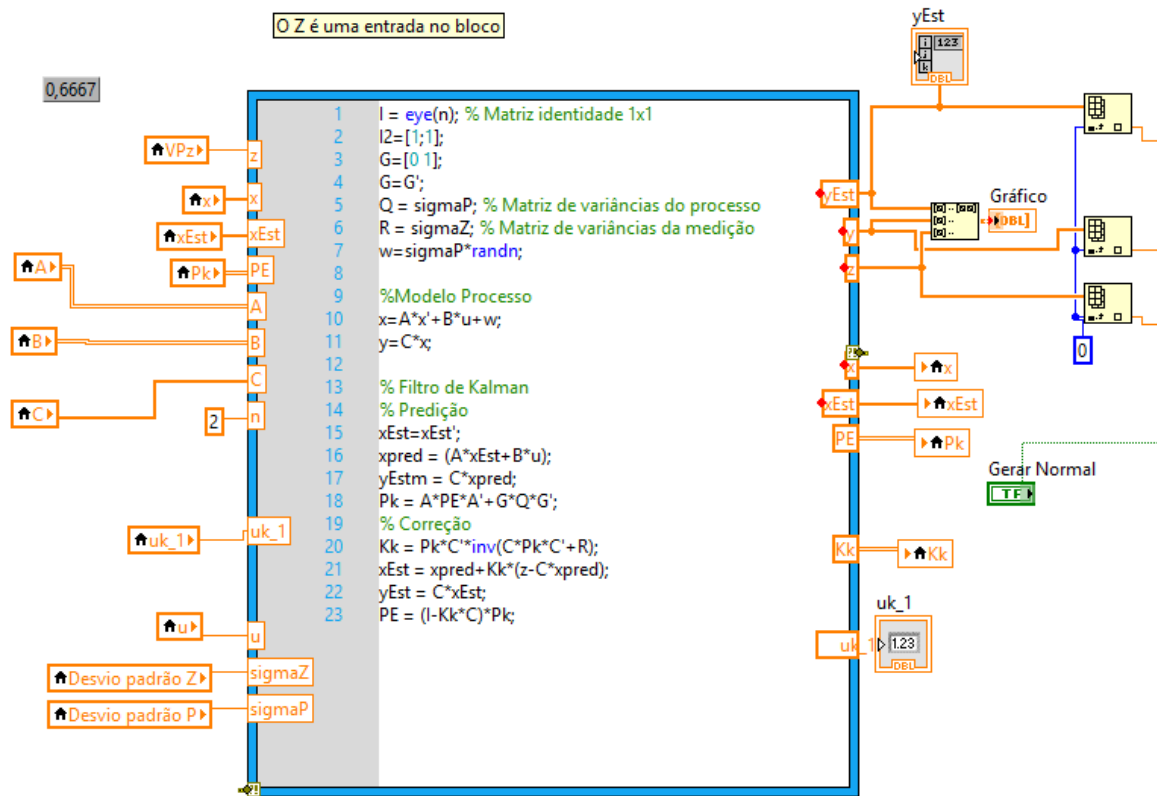
Fonte: Elaborado pelo autor

Da mesma forma, observando as características das respostas ao degrau na Figura 42 para um PID fracionário do tipo  $PI^{1.2}$ , tanto para o Matlab quanto para o LabView, é possível confirmar que a implementação do PID fracionário no LabView também está condizente com os blocos PIDs fracionários do Simulink.

#### 4.6 FILTRO DE KALMAN

Em sequência à implementação dos controladores PIDs inteiro e fracionário no LabView, partiu-se para o filtro de Kalman. Através da utilização de uma tela de programação em texto, disponível no LabView, foi possível a implementação das equações que regem o filtro de Kalman, conforme já bem explorado no tópico 2.3 deste trabalho. Assim, após uma bateria de simulações e testes, chegou-se a um modelo adequado à aplicação e necessário para o desenvolvimento dos trabalhos, como segue exposto na Figura 43.

Figura 43 - Implementação filtro de Kalman LabView

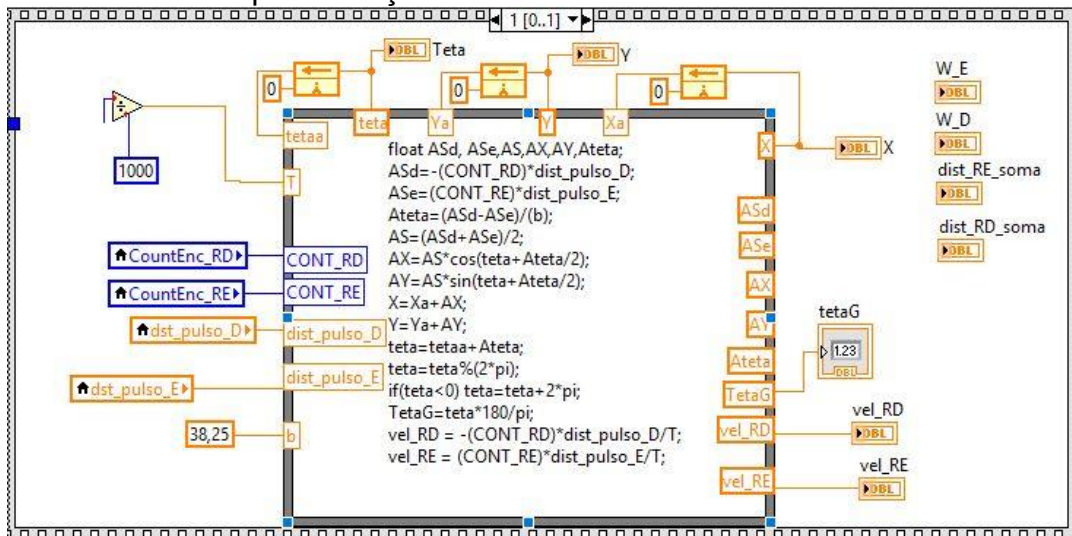


Fonte: Elaborado pelo autor

#### 4.7 DEAD RECKONING (ODOMETRIA)

A partir da base teórica abordada no tópico 3.4, implementou-se o cálculo da odometria do robô móvel como segue mostrado na Figura 44.

Figura 44 - Tela de implementação da Odometria do robô



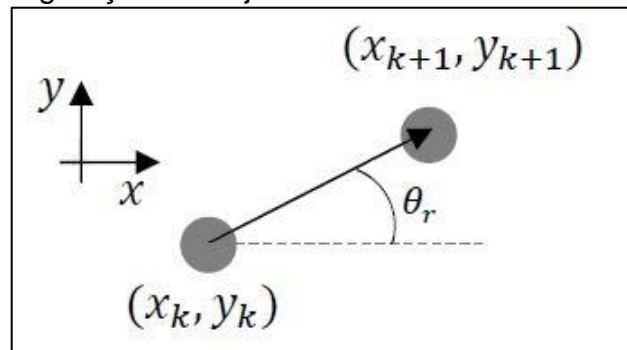
Fonte: Elaborado pelo autor

Com as medições dos pulsos oriundos dos encoders embarcados e através das especificações do robô diferencial (distância entre rodas, diâmetros das rodas e quantidade de pulsos por volta) são calculados as velocidades de cada roda e o ângulo de orientação atual. À frente estes dados calculados serão utilizados para o cálculo do erro que será aplicado nos controladores.

#### 4.8 GERADOR DE TRAJETÓRIA

Com a odometria do robô já implementada, partiu-se para a geração de trajetória, sendo esta responsável pela definição da referência de posição do robô. Através do LabView, este gerador foi implementado tendo como passo inicial a interpolação do caminho a ser seguido para posteriormente criar-se um número maior possível de referências de posição. O método utilizado de interpolação foi o *Hermite Interpolation* (CIARLET; RAVIART, 1972). Tendo-se como base a Figura 45, origina-se a equação (64), aplicada para o cálculo de  $\theta_r$  para cada índice da matriz de referências, obtendo-se consequentemente a matriz  $[x_r, y_r, \theta_r]$ .

Figura 45 - Base para geração da trajetória do robô



Fonte: Ayres et al. (2017)

$$\theta_{r_k} = \tan^{-1} \left( \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \right) \quad (64)$$

Após a etapa de interpolação da trajetória e de posse das orientações de referência para cada índice da mesma, calcula-se através da equação (65) os referenciais das velocidades linear e angular,

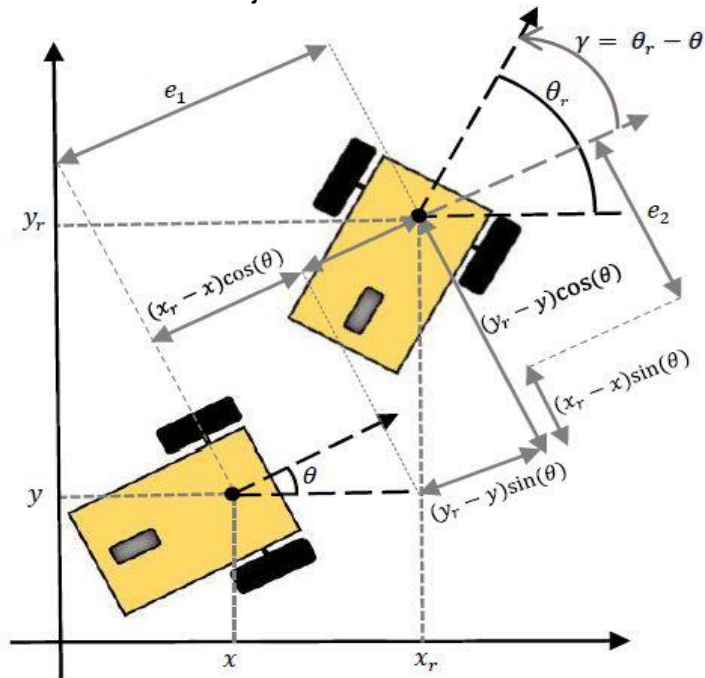
$$\begin{bmatrix} v_{r_k} \\ \omega_{r_k} \end{bmatrix} = \begin{bmatrix} (\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}) / \Delta t \\ (\theta_{k+1} - \theta_k) / \Delta t \end{bmatrix} \quad (65)$$

onde  $\Delta t$  é o tempo de amostragem utilizado para a implementação no LabView.

#### 4.9 CONTROLADOR DE TRAJETÓRIA BACKSTEPPING

A técnica de controle *backstepping* utiliza os dados determinados pelo gerador de trajetória  $[x_r, y_r, \theta_r]^k$  e os dados de localização definidos pela odometria  $[x, y, \theta]^k$  para obter os erros característicos de posicionamento do robô. Na Figura 46 são ilustrados os erros a serem considerados na implementação do controlador de trajetória no LabView e em seguida na equação (66) são apresentados na forma matricial para melhor visualização, onde utiliza-se o ponto de centro de massa do robô como referência.

Figura 46 - Gráfico dos erros de trajetória



Fonte: Ayres et al. (2017)

$$\begin{bmatrix} e_1 \\ e_2 \\ \gamma \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (66)$$

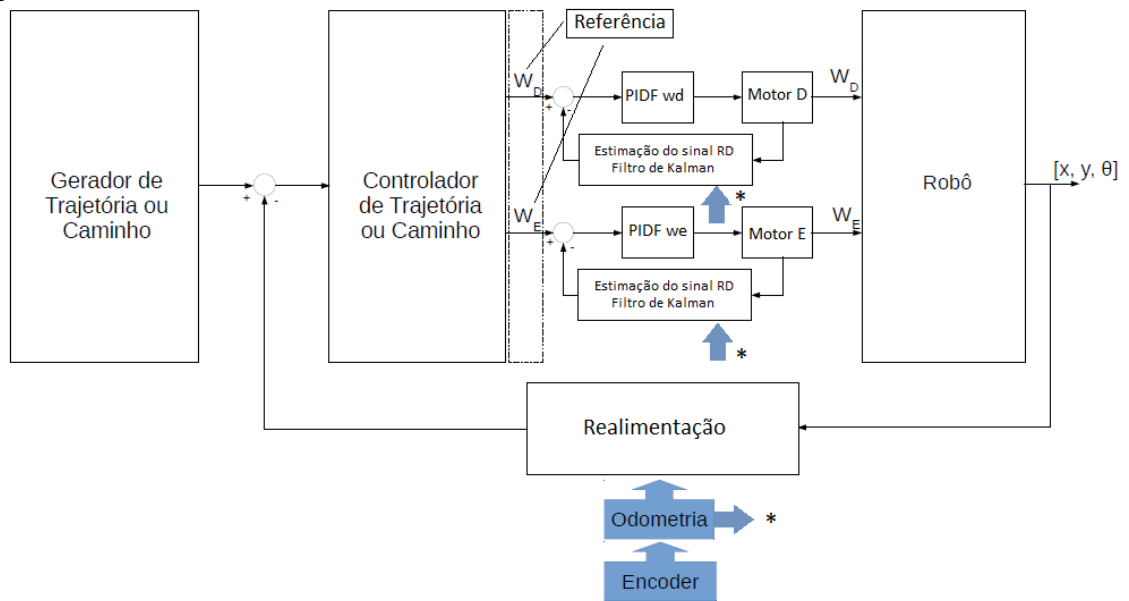
Finalmente para a implementação final do controlador de trajetória *backstepping* foi utilizada a lei de controle apresentada por Kanayama et al. (1990), conforme equação (67),

$$\begin{bmatrix} V_c \\ W_c \end{bmatrix} = \begin{bmatrix} V_r \cos(\gamma) + k_1 e_1 \\ W_r + k_2 V_r e_2 + k_3 V_r \sin(\gamma) \end{bmatrix} \quad (67)$$

onde  $V_c$  e  $W_c$  são as velocidades linear e angular de saída do controlador que serão aplicadas como referência para os controladores PID's de cada roda respectivamente e  $V_r$  e  $W_r$  são velocidades de referência determinadas pelo gerador de trajetória. Na Figura 47 são melhor ilustradas as interfaces e realimentações entre os controladores, dispositivos embarcados e acionamentos do robô móvel.



Figura 47 - Interface dos controladores do robô



Fonte: Elaborado pelo autor

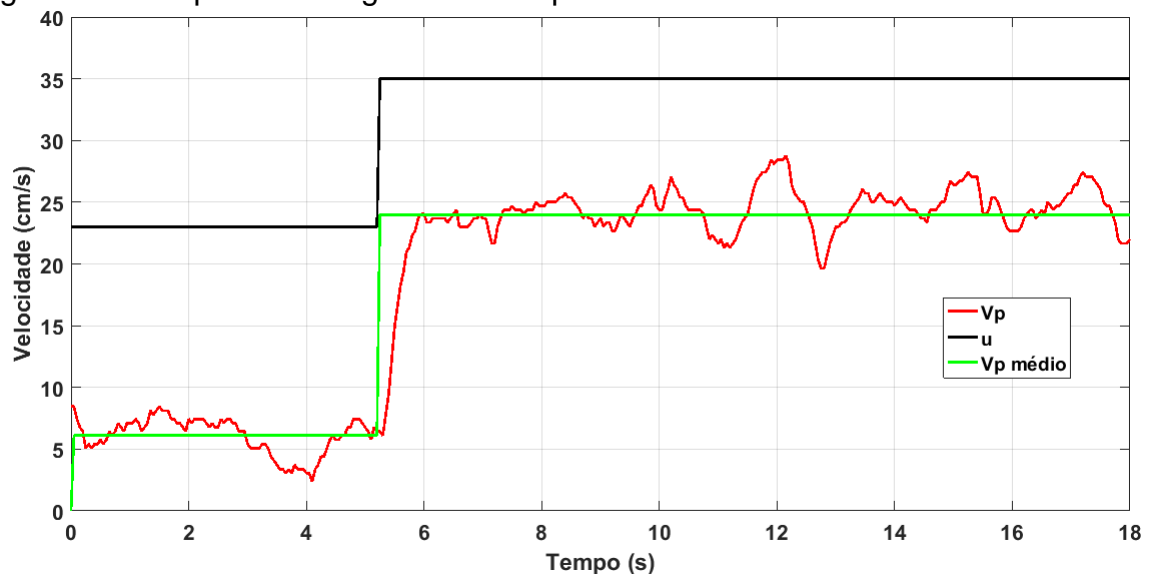
## 5 RESULTADOS

Nesta etapa do trabalho, após as implementações adequações e melhorias realizadas no robô e coleta de dados dos testes em laboratório, são apresentados os resultados.

### 5.1 MODELAGEM DOS ACIONAMENTOS

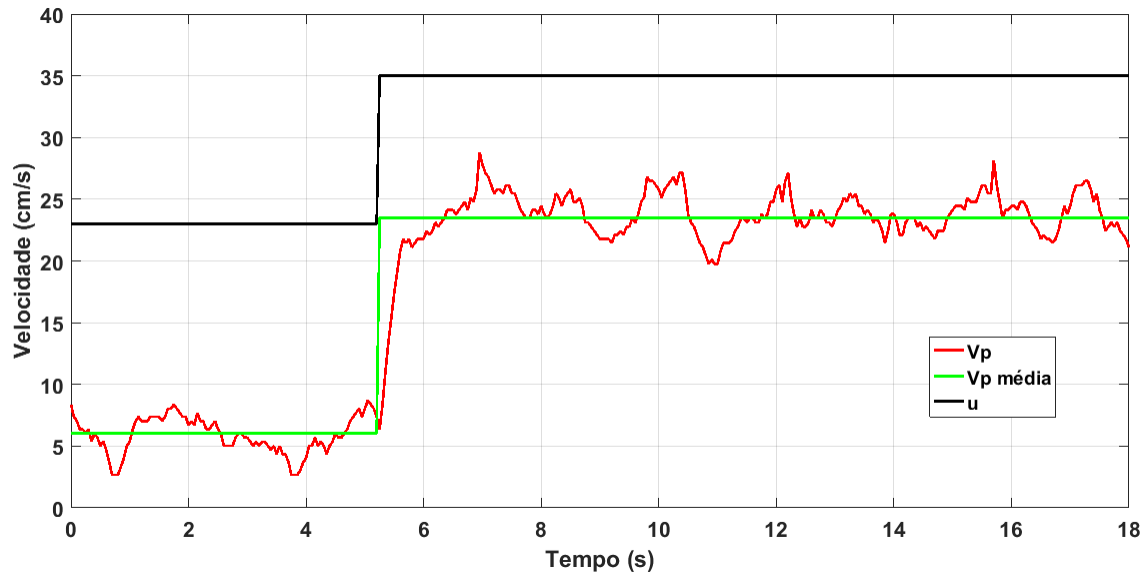
Como ponto de partida para a implementação e sintonia dos PIDs de controle das velocidades das rodas do robô móvel, foi realizada a modelagem do sistema através da resposta ao degrau. Considerando as características físicas do sistema de acionamento do robô móvel já bem exploradas no capítulo 3, foi definido como método de modelamento a resposta ao degrau. Desta forma com o robô em solo, foi atribuído um valor inicial da ação de controle  $u$  de 23% e logo em seguida incrementado para 35% em ambas as rodas. Abaixo seguem os gráficos coletados do LabView.

Figura 48 - Resposta ao degrau roda esquerda



Fonte: Elaborado pelo autor

Figura 49 - Resposta ao degrau roda direita



Fonte: Elaborado pelo autor

Devido à característica randômica observada no sinal de velocidade tanto na roda direita como na roda esquerda, foi estabelecida a média dos valores coletados,  $V_p$  média, como base para a modelagem. Assim foi possível determinar as equações características dos dois acionamentos, conforme sequência de cálculos a seguir:

a) Roda esquerda,

$$k = \frac{\Delta y}{\Delta u} \therefore k = \frac{\bar{v}_f - \bar{v}_i}{u_f - u_i} \therefore k = \frac{23,98 - 6,11}{35 - 23} \therefore k = 1,489 \quad (68)$$

$$\tau = t, \text{ para } 0,632 * V_p \therefore \tau = 0,37 \text{ s} \quad (69)$$

então,

$$G_{Re}(s) = \frac{k}{\tau s + 1} \therefore G_{Re}(s) = \frac{1,489}{0,37s + 1} \quad (70)$$

b) Roda direita:

$$k = \frac{\Delta y}{\Delta u} \therefore k = \frac{\bar{v}_f - \bar{v}_i}{u_f - u_i} \therefore k = \frac{23,51 - 6,06}{35 - 23} \therefore k = 1,454 \quad (71)$$

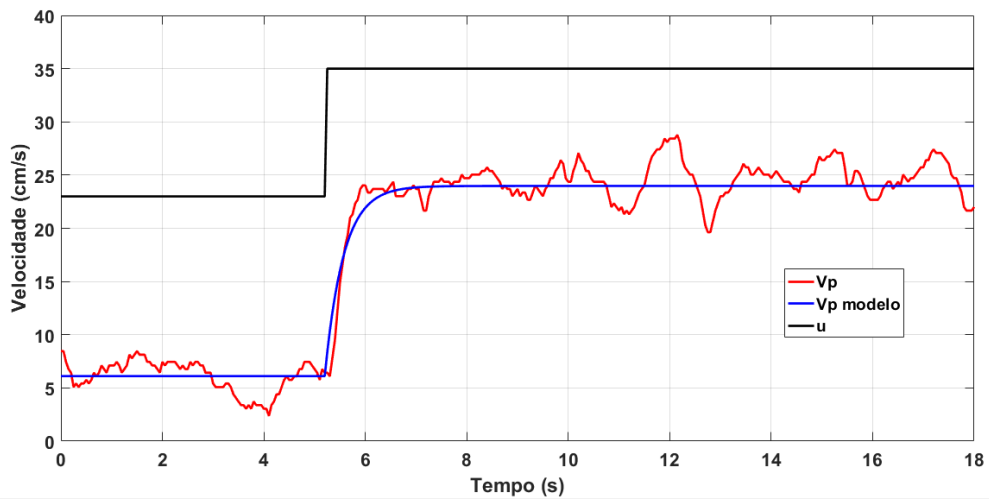
$$\tau = t, \text{ para } 0,632 * V_p \therefore \tau = 0,31 \text{ s} \quad (72)$$

então,

$$G_{Rd}(s) = \frac{k}{Ts+1} \quad \therefore G_{Rd}(s) = \frac{1,454}{0,31s+1} \quad (73)$$

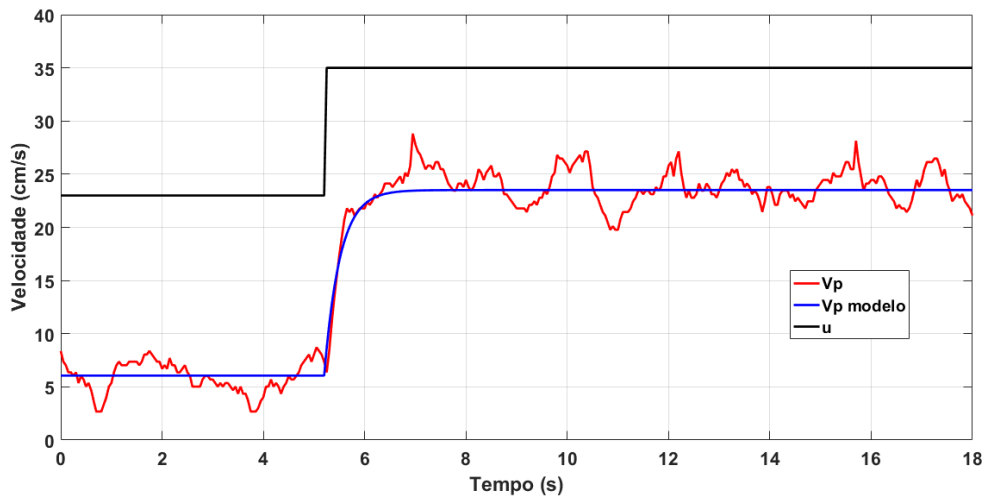
As Figuras 50 e 51 mostram os resultados dos modelos encontrados frente à resposta ao degrau de cada roda.

Figura 50 - Resposta ao degrau e modelo da roda esquerda



Fonte: Elaborado pelo autor

Figura 51 - Resposta ao degrau e modelo da roda direita



Fonte: Elaborado pelo autor

## 5.2 RESPOSTAS DOS PIDS DAS RODAS

Após a etapa de modelamento dos acionamentos das rodas, partiu-se para a sintonia das malhas de controle compostas por dois controladores PIDs implementados através do LabView, controladores estes já apresentados no tópico 3.2. Os métodos de sintonia utilizados foram o IMC e o AG. A seguir serão apresentados o detalhamento e resultados das sintonias.

### 5.2.1 Sintonia e desempenho do PID inteiro

Os métodos de sintonia escolhidos para a sintonia dos PIDs inteiros foram o IMC e o AG. No método IMC, conforme abordado no tópico 2.3.1, é definido o critério de desempenho  $\beta$ , para que os parâmetros do controlador possam ser dimensionados respeitando as características dos acionamentos. Desta forma, optou-se por uma sintonia mais robusta, com um valor de  $\beta$  elevado (em torno de três vezes o valor da constante de tempo em malha aberta do processo modelado). Assim, através das equações (20) e (22) foram determinados os parâmetros dos controladores PIDs, conforme Tabela 4.

Tabela 4 - Parâmetros de sintonia IMC

Sintonia pelo método IMC					
Roda direita			Roda esquerda		
Processo					
$K_c$	$\tau$	$\beta$	$K_c$	$T$	$\beta$
1,454	0,310	0,930	1,489	0,370	1,110
Controlador PI					
$K_p$	$K_i$	$K_d$	$K_p$	$K_i$	$K_d$
0,229	3,220	0,000	0,224	2,700	0,000

Fonte: Elaborado pelo autor

Para o método de sintonia através de AG foi necessário estabelecer os parâmetros do algoritmo, como tamanho da população, número de gerações, taxa de cruzamento e taxa de mutação além de considerar a modelagem do sistema para os

testes de desempenho com o objetivo de alcançar a escolha dos melhores parâmetros. Na Tabela 5 seguem os parâmetros definidos.

Tabela 5 - Parâmetros AG

Parâmetros do Algoritmo Genético	
Tamanho da população	200
Número de gerações	20
Taxa de cruzamento	75%
Taxa de mutação	1%

Fonte: Elaborado pelo autor

Com os parâmetros do AG estabelecidos realizaram-se as buscas pelas melhores sintonias para os PIDs. Após o processo de buscas foram estabelecidos os seguintes valores mostrados na Tabela 6. Utilizou-se o *IAE* como o método de escolha dos melhores indivíduos, nesse caso os parâmetros de sintonias dos controladores.

Tabela 6 - Sintonia PIDs inteiros

Método	Acionamento	Fitness (IAE)	Sintonia	
			$K_p$	$K_i$
IMC	Direito	-	0,229	3,22
	Esquerdo	-	0,224	2,7
AG	Direito	28,49	0,220	2,70
	Esquerdo	32,94	0,370	2,46

Fonte: Elaborado pelo autor

Em seguida, de posse dos parâmetros de sintonia dos PIDs inteiros partiu-se para os testes no robô. Como padronização dos testes foi definido um percurso retilíneo de aproximadamente 5 a 8 metros no laboratório de testes localizado no GAIIn e dois valores de referência a serem imputados às rodas, sendo estes 20 cm/s por 5 segundos e 40 cm/s por 10 segundos aproximadamente. Embora todos os dados e

gráficos estivessem disponíveis no LabView, foi necessária a exportação dos mesmos para o Excel e Matlab para posteriormente serem utilizados nos cálculos dos índices de desempenho estabelecidos para a avaliação e comparação dos controladores. Sendo estes o *IAE*, *OS* e *TA*.

Para os cálculos do *OS* e do *TA*, foi estabelecido que o período transitório finaliza-se após a estabilização do sinal de saída da velocidade em  $\pm 20\%$  do valor de referência aplicado. Na Tabela 7 são apresentados os resultados dos PIDs inteiros.

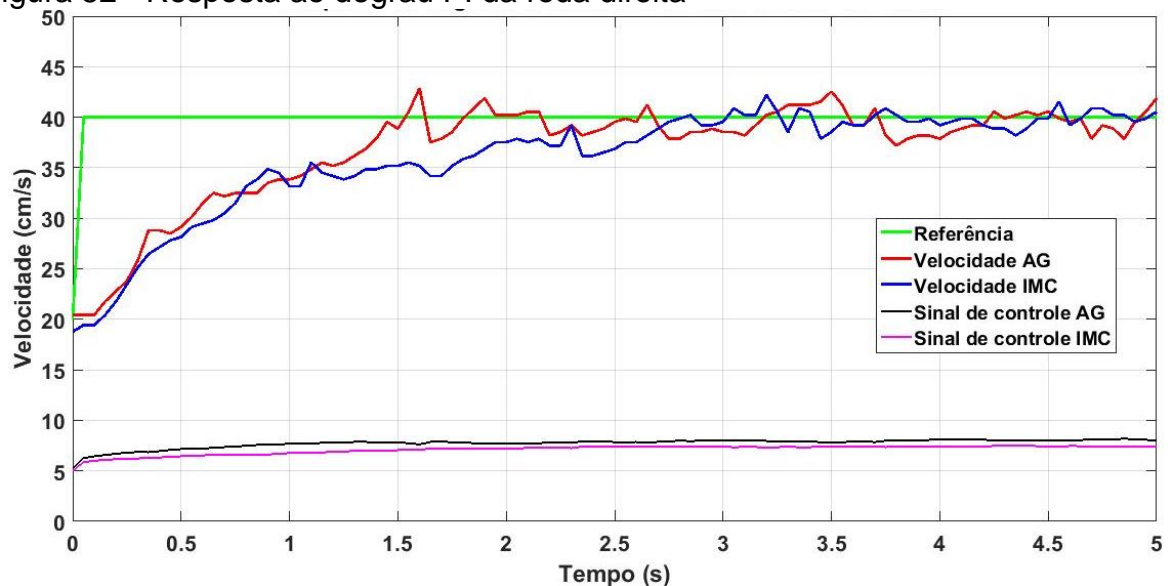
Tabela 7 - Índices de desempenhos dos PIDs inteiros

Método	Acionamento	$K_p$	$K_i$	Desempenho		
				IAE	OS (%)	TA (s)
IMC	Direito	0,229	3,220	20,21	11,09	2,55
	Esquerdo	0,224	2,700	15,32	11,55	1,90
AG	Direito	0,220	4,400	16,99	14,44	1,40
	Esquerdo	0,370	2,460	17,13	8,17	3,30

Fonte: Elaborado pelo autor

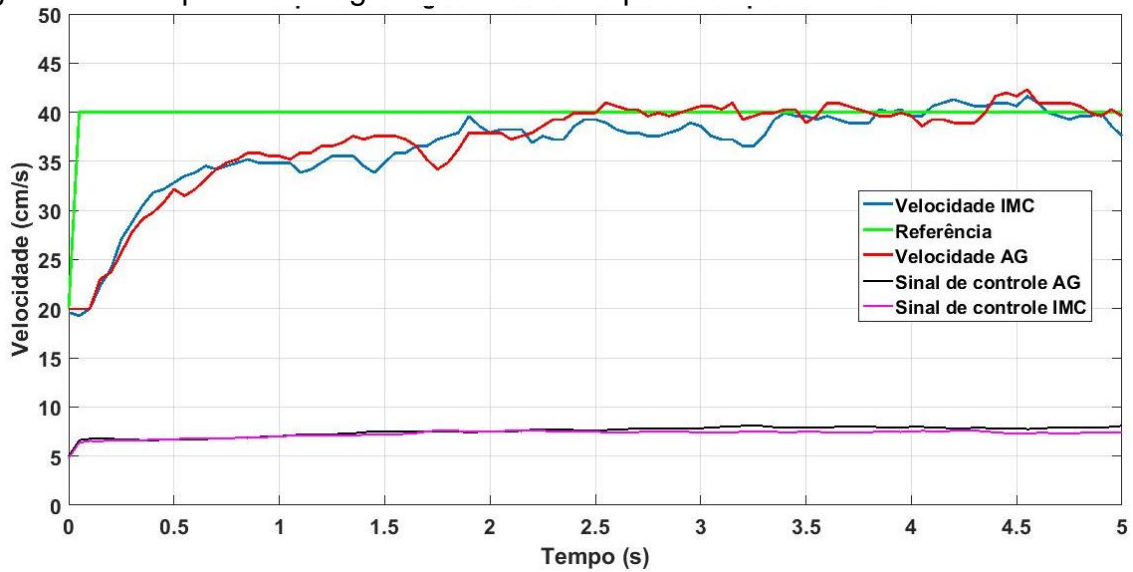
Nas Figuras 52 e 53 são apresentados os gráficos dos testes realizados.

Figura 52 - Resposta ao degrau PI da roda direita



Fonte: Elaborado pelo autor

Figura 53 - Resposta ao degrau PI da roda esquerda



Fonte: Elaborado pelo autor

### 5.2.2 Sintonia e desempenho do PID fracionário

Da mesma forma como realizado para os PIDs inteiros, foram feitas as buscas dos parâmetros de sintonia dos PIDs fracionários através do AG. Na Tabela 8 são apresentados os resultados das buscas.

Tabela 8 - Sintonia dos PIDs fracionários

Método	Acionamento	Fitness	Sintonia		
			$K_p$	$K_i$	$\lambda$
AG	Direito	19,31	0,49	4,67	0,85
	Esquerdo	22,14	0,50	4,93	0,73

Fonte: Elaborado pelo autor

A sintonia estabelecida para o PID inteiro através do método IMC, já apresentada na Tabela 7, também foi utilizada para os testes dos PIDs fracionários, com uma variação do  $\lambda$  de 0,5 a 0,9. Em seguida os parâmetros foram inseridos nos controladores PIDs implementados no LabView e foram realizados os testes de desempenho para cada roda, como apresentado na Tabela 9.



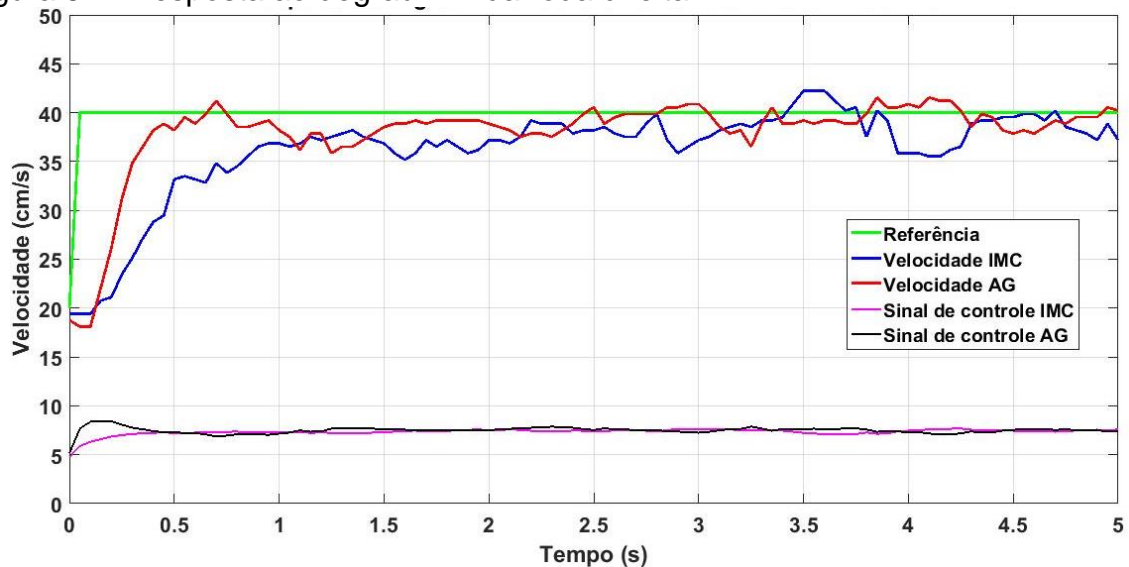
Tabela 9 - Índices de desempenho dos PID's fracionários

Método	Sintonia / Desempenho											
	Roda direita						Roda esquerda					
	$K_p$	$K_i$	$\lambda$	IAE	OS(%)	TA(s)	$K_p$	$K_i$	$\lambda$	IAE	OS(%)	TA(s)
AG	0,49	4,67	0,85	10,50	7,74	3,25	0,50	4,93	0,73	12,72	3,09	4,55
	0,49	4,67	0,75	12,40	7,74	3,65	0,50	4,93	0,65	14,67	4,78	4,30
	0,49	4,67	0,90	10,40	17,11	3,20	0,50	4,93	0,80	11,36	9,86	4,45
	0,49	4,67	1,00	11,90	14,44	3,95	0,50	4,93	0,90	10,35	9,86	2,65
	0,50	8,50	0,6	12,50	24,49	3,25	0,50	8,50	0,630	11,94	26,78	3,20
IMC	0,204	6,44	0,50	36,70	0,00	>5	0,216	5,40	0,50	40,97	0,00	>5
	0,204	6,44	0,60	29,40	0,00	4,65	0,216	5,40	0,60	34,37	0,00	>5
	0,204	6,44	0,80	18,80	11,09	4,25	0,216	5,40	0,80	20,60	8,17	4,35
	0,204	6,44	0,90	16,70	19,47	4,50	0,216	5,40	0,90	18,04	13,24	4,75

Fonte: Elaborado pelo autor

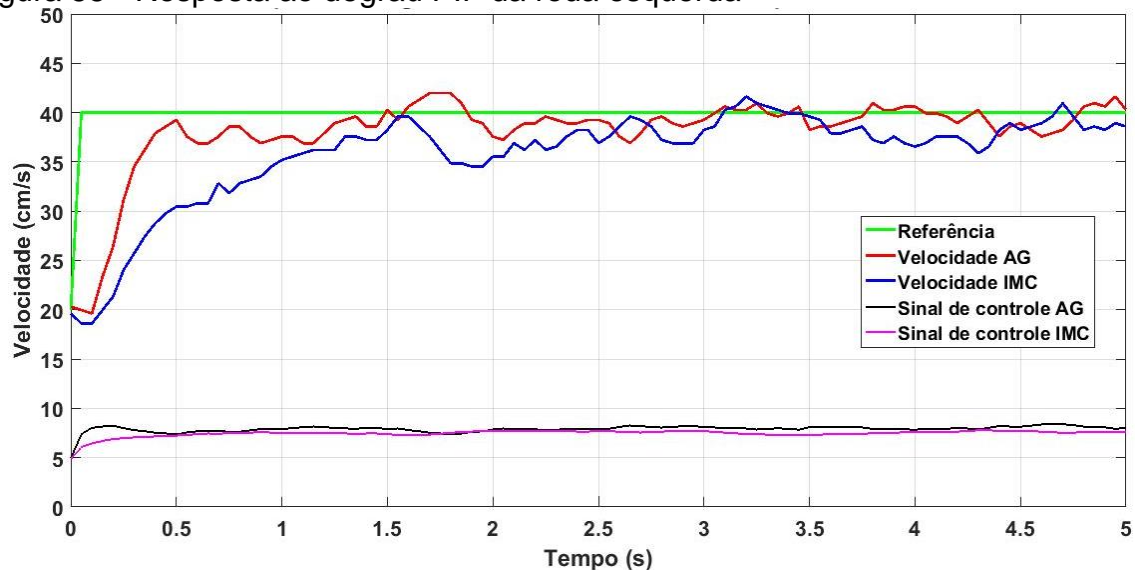
Nas Figuras 54 e 55 são apresentados os gráficos dos testes realizados correspondentes aos melhores desempenhos, considerando para a seleção o menor IAE, em seguida o menor OS (>0), e por último o menor TA.

Figura 54 - Resposta ao degrau\_PIF da roda direita



Fonte: Elaborado pelo autor

Figura 55 - Resposta ao degrau PIF da roda esquerda

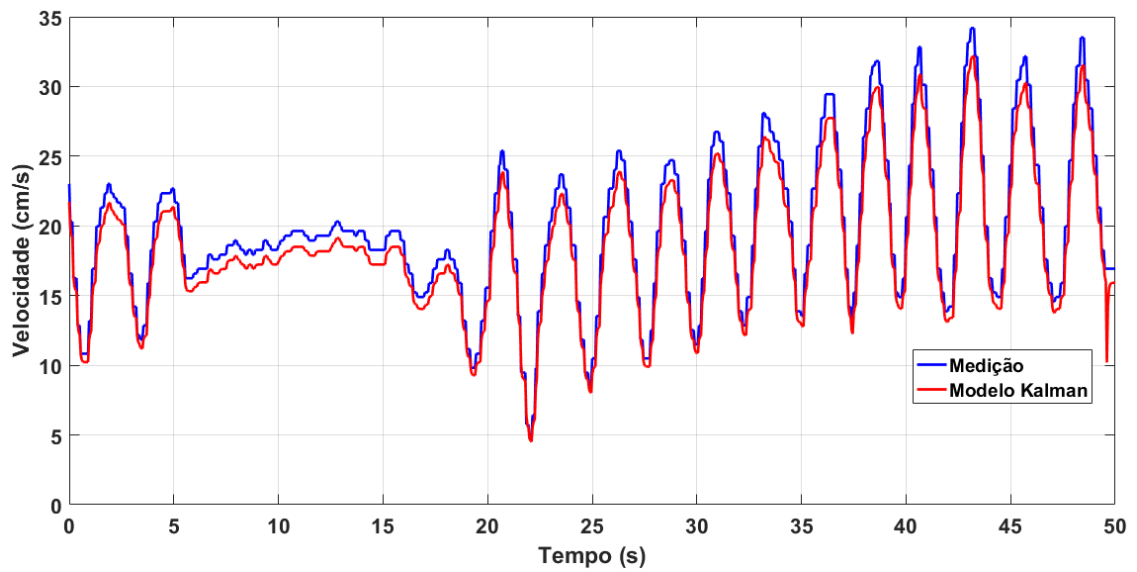


Fonte: Elaborado pelo autor

### 5.2.3 Sintonia e desempenho dos PIDs com filtro de Kalman

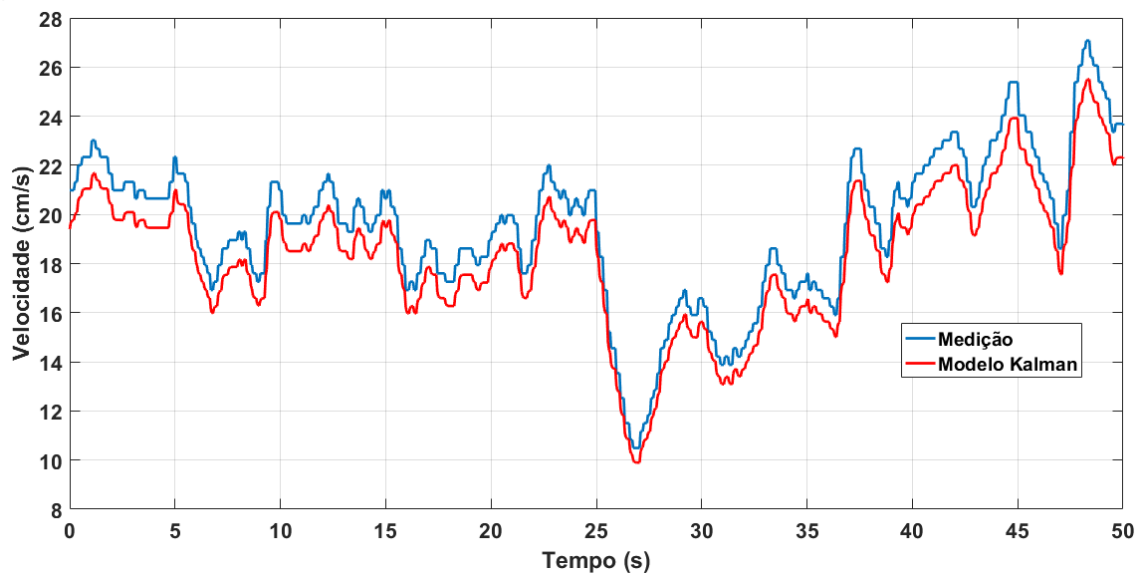
Finalizando os testes de desempenho para o controle de velocidade das rodas do robô móvel foram avaliados os mesmos controladores só que agora com a inclusão do filtro de Kalman implementado no LabView. O filtro foi incluído no circuito dos controladores com o objetivo de minimizar possíveis ruídos nas medições de velocidade das rodas causadas pelas características construtivas dos acionamentos, que ao longo do trabalho mostraram-se possíveis de geração de tais sintomas. Nota-se nas Figuras 56 e 57 que o Filtro de Kalman implementado não contribuiu significativamente para minimização da característica randômica das velocidades das rodas, o que comprova a não existência de componentes ruidosas consideráveis nas medições.

Figura 56 - Medição de velocidade da roda esquerda com KF



Fonte: Elaborado pelo autor

Figura 57 - Medição de velocidade da roda direita com KF



Fonte: Elaborado pelo autor

Em seguida também foram coletados os dados de desempenho dos PIDs inteiros e fracionários conforme apresentado nas Tabelas 10 e 11.

Tabela 10 - Índices de desempenho dos PID's com filtro de Kaman

Método	Desvio padrão Kalman		Roda	$K_p$	$K_i$	Desempenho		
	Processo	Medição				IAE	OS (%)	TA (s)
			Esquerdo	0,224	2,7	20,34	23,40	4,35
AG	0.001	0.0001	Direito	0,220	4,4	16,41	29,52	>5
			Esquerdo	0,370	2,46	15,05	20,01	4,95

Fonte: Elaborado pelo autor

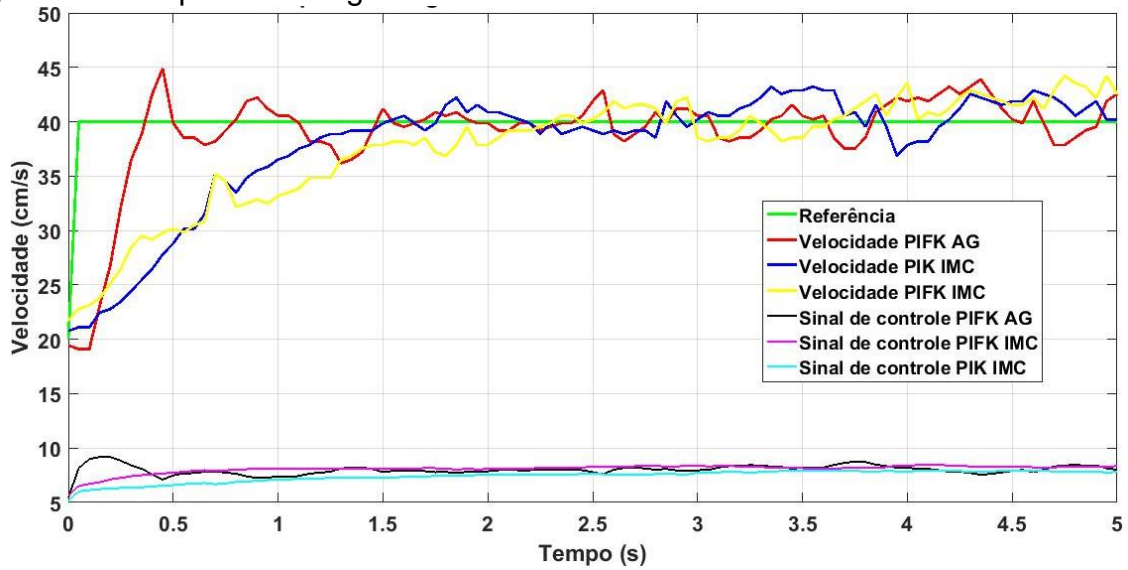
Tabela 11 - Índices de desempenho dos PID's fracionários com filtro de Kalman

Método	Sintonia / Desempenho											
	Roda direita						Roda esquerda					
	$K_p$	$K_i$	$\lambda$	IAE	OS(%)	TA(s)	$K_p$	$K_i$	$\lambda$	IAE	OS(%)	TA(s)
AG	0,49	4,67	0,85	12,54	30,78	4,45	0,50	4,93	0,73	10,56	21,05	4,90
	0,49	4,67	0,75	10,33	24,49	4,35	0,50	4,93	0,65	10,11	14,94	2,45
	0,49	4,67	0,90	11,57	21,14	4,95	0,50	4,93	0,80	11,35	25,09	4,75
	0,49	4,67	1,00	14,35	27,84	4,70	0,50	4,93	0,90	13,72	28,48	>5
	0,50	8,50	0,63	12,97	39,57	3,90	0,50	8,50	0,63	12,17	42,01	4,60
IMC	0,204	6,44	0,50	34,40	0,00	>5	0,216	5,40	0,50	38,07	0,00	>5
	0,204	6,44	0,60	30,30	0,00	4,80	0,216	5,40	0,60	34,46	0,00	5,00
	0,204	6,44	0,80	18,60	16,10	4,20	0,216	5,40	0,80	18,92	3,10	4,80
	0,204	6,44	0,90	16,20	16,10	4,00	0,216	5,40	0,90	16,57	14,90	2,65

Fonte: Elaborado pelo autor

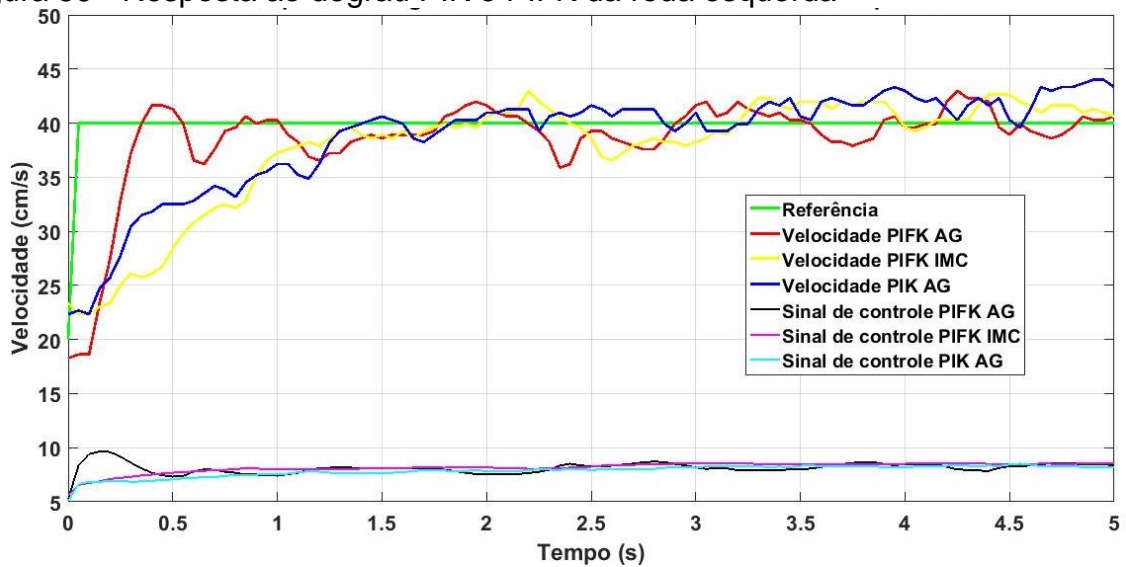
Nas Figuras 58 e 59 são apresentados os gráficos dos testes realizados correspondentes aos melhores desempenhos.

Figura 58 - Resposta ao degrau PIK e PIFK da roda direita



Fonte: Elaborado pelo autor

Figura 59 - Resposta ao degrau PIK e PIFK da roda esquerda



Fonte: Elaborado pelo autor

Os dados coletados e os gráficos mostram que o PI<sup>λ</sup> com filtro de Kalman sintonizado através de AG possui uma resposta mais rápida em relação à mudança da referência, sem grandes alterações no sinal de controle.

### 5.3 CONTROLE DE TRAJETÓRIA

Após a realização dos testes de desempenho dos controladores nas rodas do robô móvel, partiu-se para os testes de forma integrada dos controladores PIDs das rodas em conjunto com o controlador de trajetória. Este controlador como já abordado anteriormente, também foi implementado no LabView e possui um gerador de trajetória que gera as referências para os controladores PIDs de cada roda, assim possibilitando avaliar através do cálculo do *IAE* quais sintonias testadas são as mais adequadas para a realização de uma trajetória pré-definida.

O controlador o *backstepping* utilizado como controlador principal do robô foi sintonizado pelo método de tentativa e erro através de três ganhos,  $k_1$ ,  $k_2$  e  $k_3$ , conforme equação (67) e apresentado na conforme Tabela 12. A fim de definir um padrão para a análise dos controladores PIDs das rodas utilizou-se a mesma sintonia para todo o teste.

Tabela 12 - Sintonia controlador de trajetória

Sintonia do controlador de trajetória	
$k_1$	2,4
$k_2$	3,8
$k_3$	3,1

Fonte: Elaborado pelo autor

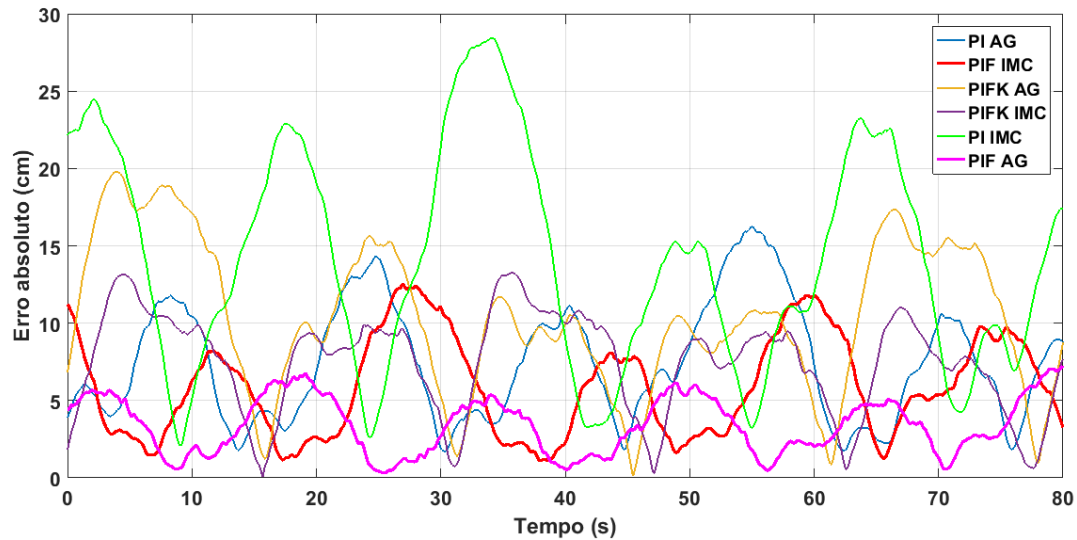
Em seguida foram realizados os testes de trajetória considerando melhores resultados alcançados nos testes iniciais dos controladores PIDs das rodas. Como elemento de comparação do desempenho utilizou-se o *IAE*, obtido através da distância euclidiana ( $De$ ), sendo esta última calculada segundo a equação (74),

$$De = \sqrt{(x_r.d - x_r.m)^2 + (y_r.d - y_r.m)^2} \quad (74)$$

onde  $x_r.d$  é a posição desejada na coordenada  $x$ ,  $x_r.m$  é a posição medida na coordenada  $x$ ,  $y_r.d$  é a posição desejada na coordenada  $y$  e  $y_r.m$  é a posição medida

na coordenada  $y$ . A Figura 60 apresenta os erros de cada controlador testado ao longo do tempo.

Figura 60 - Erro absoluto de trajetória ao longo do tempo de cada controlador



Fonte: Elaborado pelo autor

Observa-se que os controladores  $PI^{\lambda}$  sintonizados por AG e IMC apresentaram os menores erros absolutos calculados a partir da  $De$ . Na Tabela 13 são apresentados os dados de cada controlador das rodas nos testes de trajetória a partir do cálculo do  $IAE$ .

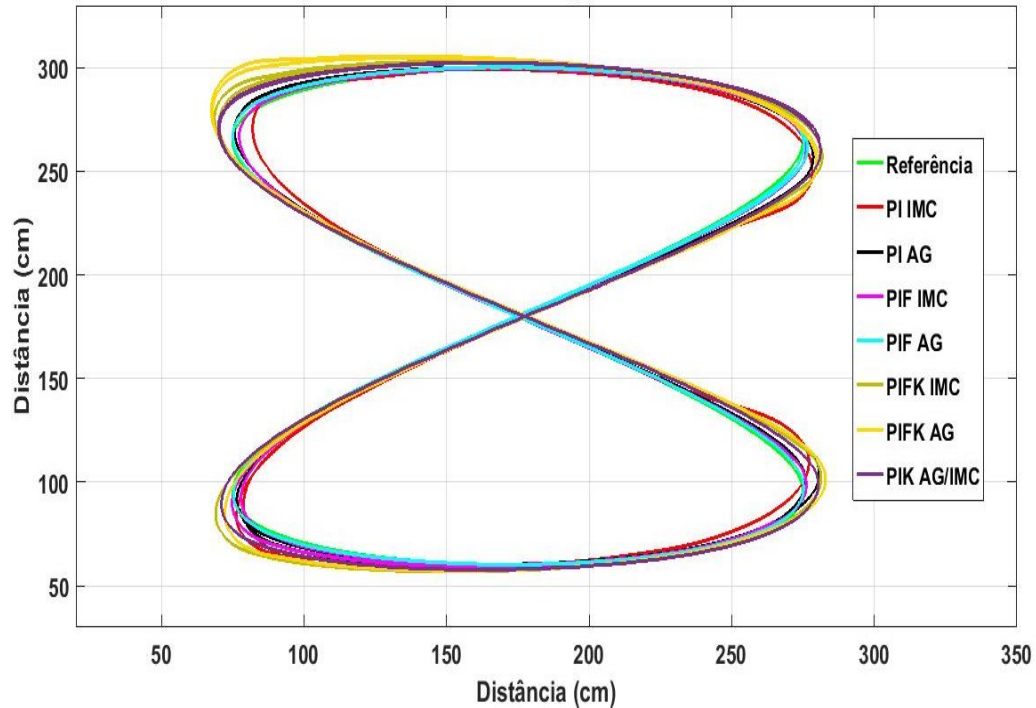
Tabela 13 - Resultados testes de trajetória

Controlador	Método Sintonia	Teste de trajetória <i>backstepping</i>						
		Roda direita			Roda esquerda			
		$K_p$	$K_i$	$\lambda$	$K_p$	$K_i$	$\lambda$	IAE
PI	IMC	0,229	3,220	-	0,224	2,700	-	1123,10
	AG	0,220	4,400	-	0,370	2,460	-	594,12
PIK	AG/IMC	0,229	3,220	-	0,370	2,460	-	1101,04
PIF	IMC	0,229	6,440	0,800	0,224	5,400	0,800	471,84
	AG	0,490	4,670	0,850	0,500	4,930	0,900	344,67
PIFK	IMC	0,229	6,440	0,900	0,224	5,400	0,900	596,20
	AG	0,490	4,670	0,750	0,500	4,930	0,650	832,40

Fonte: Elaborado pelo autor

Na Figura 61 são apresentados os gráficos dos testes de trajetória realizados.

Figura 61 - Teste de trajetória

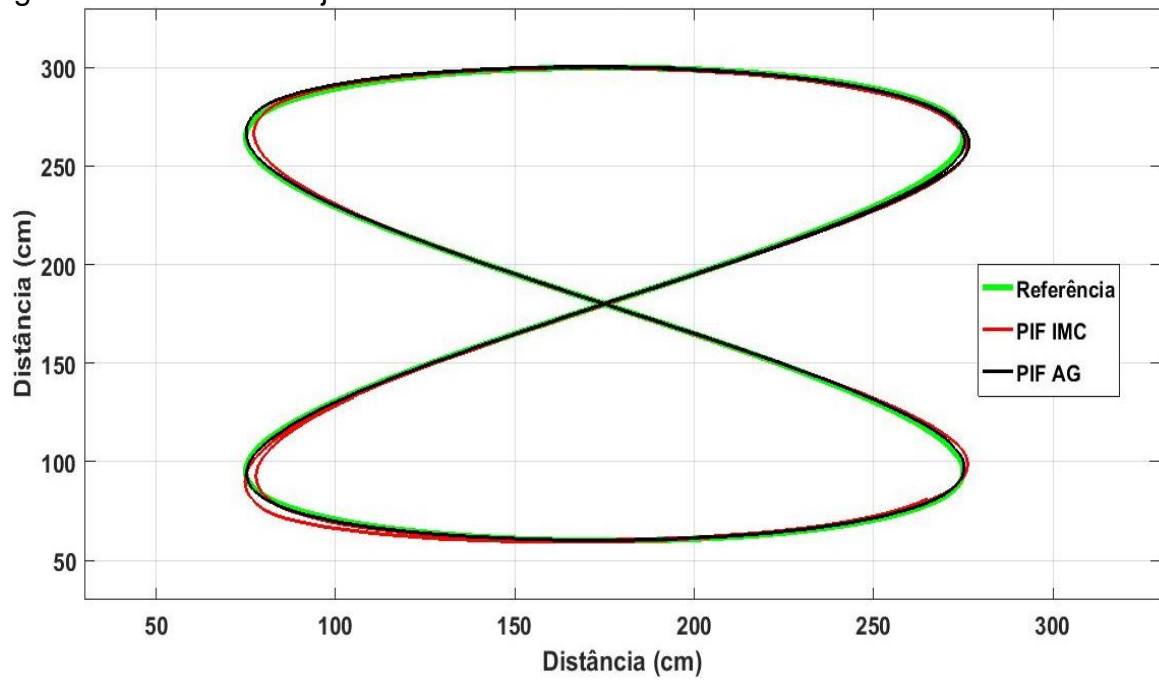


Fonte: Elaborado pelo autor

Nas Figuras 62, 63 e 64 são apresentados os gráficos dos testes de trajetória dos  $PI^{\wedge}$  sintonizados pelo IMC e pelo AG, os quais apresentaram melhor desempenho. Embora a trajetória apresentada pelo controlador  $PI^{\wedge}$  sintonizado através do AG seja a mais próxima da trajetória de referência, os gráficos dos controladores de ambas as rodas mostram uma ação de controle mais agressiva o que acarreta uma mudança com maior amplitude das velocidades das rodas. Assim, após a comparação do comportamento nos testes práticos, de forma geral, o controlador  $PI^{\wedge}$  sintonizado através do IMC, mostrou-se mais adequado para a aplicação no robô móvel diferencial.

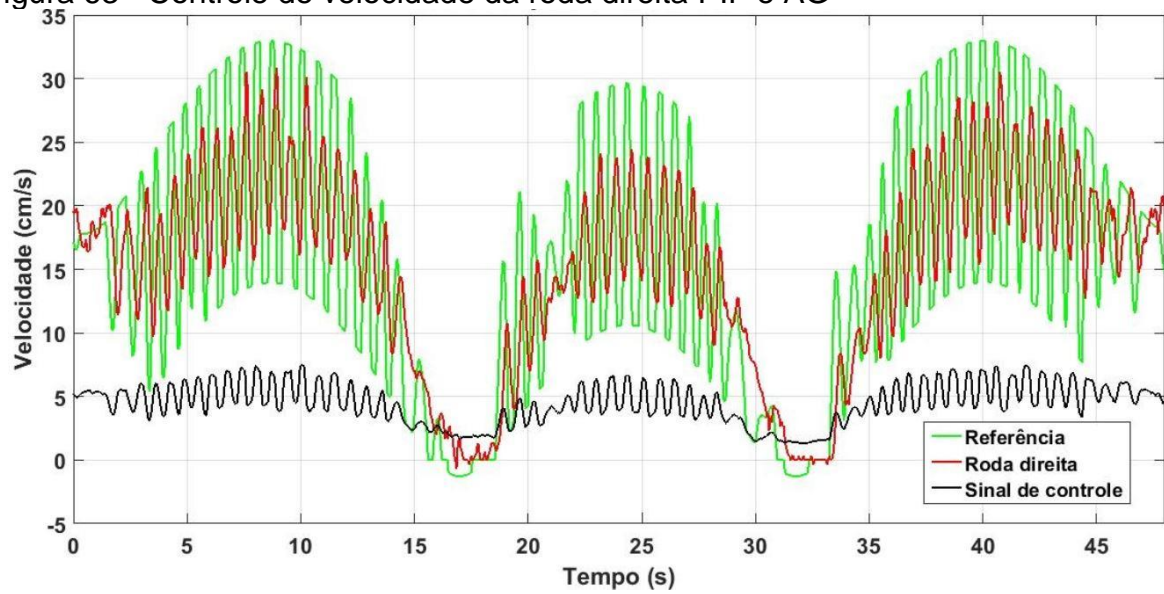


Figura 62 - Teste de trajetória final



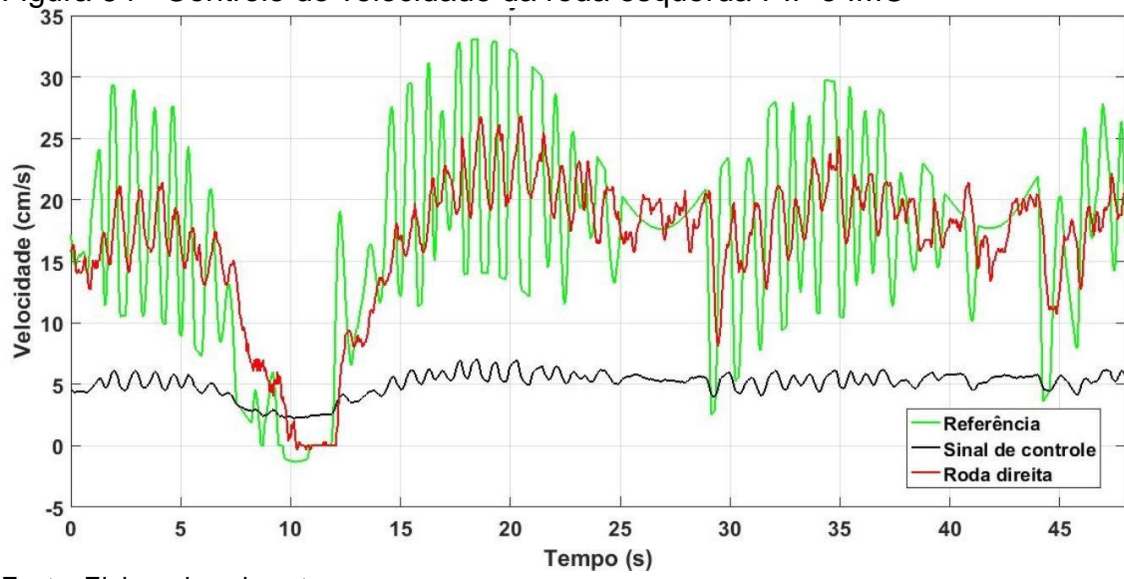
Fonte: Elaborado pelo autor

Figura 63 - Controle de velocidade da roda direita PIF e AG



Fonte: Elaborado pelo autor

Figura 64 - Controle de velocidade da roda esquerda PIF e IMC



Fonte: Elaborado pelo autor

## 6 CONCLUSÕES E RECOMENDAÇÕES

No trabalho desenvolvido foram realizadas as implementações de controladores PID e  $PI^{\lambda}D^{\mu}$  em conjunto com um filtro de Kalman discreto. No decorrer da pesquisa foram realizados testes dos controladores em uma planta de nível (CASTRO et al., 2016). Os resultados mostraram que o  $PI^{\lambda}D^{\mu}$  apresentou desempenho ligeiramente melhor, comprovados pelos indicadores de desempenho *IAE*, *OS* e *TA*. Quanto ao sinal de controle, o  $PI^{\lambda}D^{\mu}$  fracionário permitiu um menor esforço para a o alcance e manutenção do sinal de referência desejado.

Em sequência foi implementado um PID para controle de velocidade nos acionamentos das rodas de um robô móvel diferencial (BERNARDES et al., 2017). Nesta etapa foram implementadas melhorias nos acionamentos, nos circuitos eletrônicos e instrumentos embarcados do robô móvel diferencial, em conjunto com a implementação e testes do PID para controle de velocidade das rodas, as quais geraram resultados significativos para o seguimento do trabalho principalmente no aperfeiçoamento da odometria.

Também foram testados os controladores inteiro e fracionário em conjunto com o filtro de Kalman em uma planta de pressão (BERNARDES et al., 2017). Nesta etapa os resultados alcançados através do  $PI^{\lambda}D^{\mu}$  foram ligeiramente melhores em relação ao *ITAE*, *OS* e *TA*, quando comparados ao PID. Já a introdução do filtro de Kalman nos controladores aprimoraram todos os resultados, reforçando a performance do controle fracionário.

Avançando ao objetivo final do trabalho foram implementados além dos PID,  $PI^{\lambda}D^{\mu}$  e o filtro de Kalman, um gerador de trajetória e um controlador *backstepping* para o controle de trajetória do robô móvel. Os testes de desempenho dos controladores com e sem o filtro de Kalman através da análise do *IAE*, *OS* e *TA* possibilitaram a escolha das melhores sintonias para os testes seguintes no controlador de trajetória.

Considerando o objetivo de reduzir a folga do sistema de acionamento do robô, além das melhorias estruturais realizadas, era esperado uma contribuição significativa com a implementação do filtro de Kalman nos sinais dos encoders das rodas, o que

os testes práticos comprovaram não ser possível, uma vez que os gráficos e resultados numéricos referentes ao controle de velocidade das rodas mostraram que há uma variabilidade característica imputada pela forma construtiva do robô, que não foi possível ser filtrada pelo filtro de Kalman, uma vez que o objetivo de filtros dessa natureza é a minimização de ruídos.

Após os testes de trajetória ficou evidente tanto nos gráficos colhidos como também na comparação do *IAE*, que os controladores de velocidade implementados nas rodas do tipo  $PI^{\wedge}$ , tanto sintonizados pelo *IMC* quanto pelo *AG*, permitiram uma trajetória mais próxima à referência, gerando um melhor *IAE*. No entanto, o controlador sintonizado através do *AG* tornou a resposta mais agressiva, o que para um sistema com controle de trajetória, inicialmente não seria tão vantajoso. Sendo assim, mesmo com um *IAE* um pouco mais elevado que o melhor resultado, o  $PI^{\wedge}$  sintonizado através do *IMC* foi o que alcançou o melhor desempenho e proporcionou um controle mais adequado ao robô móvel.

Como objetos de trabalhos futuros, ficou a oportunidade de instalação de rolamentos nas roldanas principais das rodas do robô móvel, o que traria maior precisão na transmissão ao conjunto, tornando-o mais adequado ao desenvolvimento de outros projetos complementares. O desenvolvimento de um método de sintonia dos ganhos do controlador de trajetória utilizado, uma vez que o processo utilizado nesse trabalho foi o de tentativa e erro, o qual pode tornar a busca dos ganhos muito subjetiva e dependente da habilidade de quem realiza os testes.

Por fim, com um maior conhecimento a respeito das características do robô móvel utilizado no trabalho, será possível o desenvolvimento de um modelamento conjunto mais preciso, o que permitirá novos testes com o filtro de Kalman e também melhores resultados das sintonias dos controladores.

## REFERÊNCIAS

ABDALLA, T. Y., HAMZAH, M. I. Trajectory tracking control for mobile robot using wavelet network. **International Journal of Computer Applications** (0975 - 8887), v. 74, n. 3, 2013.

ALMEIDA, G. M. **Controle preditivo sintonizado via algoritmo genético aplicado em processos siderúrgicos**. 2011. 171 f. Tese (Doutorado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Vitória, 2011.

AURALIUS, M. **Sharing\_Experiences**. 2010. [ONLINE]. Disponível em: <<http://sites.google.com/site/auraliusproject/fractional-order-derivative-and-integral-using-LabView>>. Acesso em: jan. 2016.

AYRES, L. M. et al. **Desenvolvimento e implantação de uma arquitetura de navegação para um robô móvel utilizando comandos de voz, algoritmo A\* e o controlador backstepping**. SBAI 2017, Porto Alegre, RS, 2017.

BECKER, T. **Navegação de um robô móvel baseado em um modelo de consciência artificial**. 2015. 91 f. Dissertação (Mestrado em Computação Aplicada) - Universidade Tecnológica Federal do Paraná, Curitiba, 2015.

BERNARDES, N. D. **Aplicação de PID fracionário e inteiro com filtro de Kalman sintonizados por algoritmo genético**. São José do Rio Preto, SP: DINCON, 2017.

\_\_\_\_\_ et al. **Implementação didática de um controlador do tipo PID para controle de velocidade num robô móvel**. Cobenge, 2017.

CAMPOS, M. C. M. M.; TEIXEIRA, H. C. G. **Controles típicos de equipamentos e processos industriais**. São Paulo: Edgard Blucher, 2006.

CAPELAS, E. O. **Funções especiais com aplicações**. São Paulo: Editora Livraria da Física, 2005.

CASTRO, F. A. et al. **Comparison of fractional and integer PID controllers tuned by genetic algorithm.** 12th IEEE/IAS International Conference on Industry Applications. 2016.

CIARLET, P.; RAVIART, P. General lagrange and hermite interpolation in  $R^n$  with applications to finite element methods. **Archive for Rational Mechanics and Analysis**, v. 46, n. 3, 1972.

FERNANDES, L. C. **Análise e Implementação de algoritmos para localização de robôs móveis.** 2003. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2003.

FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot using neural networks. **IEEE Transactions on Neural Networks** 9(4), p. 589-600, 1998.

FREITAS, B. P. S. et al. **Utilização de exemplo criados no software LabView implementados no starter kit 2.0 como ferramenta no ensino-aprendizagem da robótica.** Natal, RN, Cobenge, 2016.

HOLLAND, J. H. **Adaptation in natural and artificial systems:** an introductory analysis with applications to biology, control, and artificial intelligence. Vol. 1, [s. l.]: Massachusetts Institute of Technology, 1992.

JOHAN, A. K., **Lecture Notes for ME 155A.** Department of Mechanical and Environmental Engineering University of California Santa Barbara, 2002.

PINTO, Jan E. M. G. **Aplicação prática do método de sintonia de controladores PID utilizando o método do relé com histerese.** 2014. (Mestrado em Engenharia Elétrica e Computação) - Universidade Federal do Rio Grande do Norte, Natal, 2014.

KALMAN, R. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, v. 82, n. 1, p. 35-45, 196).

KANAYAMA, Y. et al. A stable tracking control method for an autonomous mobile robot. IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. **Proceedings...** 1990.

LOVERRO, A. **Fractional calculus**: history, definitions and applications for the engineer. USA: Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, 2004.

MARTINS, A. **O que é a robótica**. 2. ed. São Paulo: Brasiliense, 2007.

MARTINS, F. N. et al. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. **Control Engineering Practice** 16(11), p. 1354-1363, 2008.

MAYBECK, P. S. **Stochastic Models, Estimation, and Control**. Volume 1. [S.l.]: Academic Press, 1979.

PODLUBNY, I.; DORCAK, L.; KOSTIAL, I. On Fractional Derivatives, Fractional Order Dynamics Systems and  $PI^{\lambda}D^{\mu}$  - Controllers. In: IEEE CONFERENCE ON DECISION AND CONTROL. 36,. 1997, San Diego, CA. [**Anais...**], San Diego, CA, 1997.

PODLUBNY, I. Fractional-Order Systems and  $PI^{\lambda}D^{\mu}$  - Controllers. **IEEE Transactions on Automatic Control** 44(1), p. 208-214, 1999.

\_\_\_\_\_. **Fractional Differential Equations**: An introduction to fractional derivatives, fractional differential equations to methods of their solution and some of their applications. Academic Press, San Diego, 1994.

RESENDE, C. Z. et al. **Embedding obstacle avoidance to trajectory tracking for unicycle mobile robots**. IEEE/RSJ International Conference on Intelligent Robots and Systems. [S.l.: s.n.], 2012.

RIVERA, D. E.; FLORES M. E. **Control systems, robotics and automation - Vol. II - Internal Model Control**. Arizona, USA: Department of Chemical and Materials Engineering, Ira A. Fulton School of Engineering, Arizona State University, Tempe, 1986.

ROMERO, R. A. F. et al. **Robótica Móvel**. Rio de Janeiro: LTC, 2014.

SHARMA, R.; GAUR, P.; MITTAL, A. P. Optimum design of fractional-order hybrid fuzzy logic controller for a robotic manipulator. **Arabian Journal for Science and Engineering**, India, 2017.

SORENSEN, H. W. Least-Squares estimation: from Gauss to Kalman. **IEEE Spectrum**, v. 7, p. 63-68, 1970.

TOMMASI, E. E. V. et al. Estudo comparativo de controladores de seguimento de trajetória para robôs de tração diferencial: fuzzy, ganhos fixos e backstepping. SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI). 2015, Natal, Rio grande do Norte.

VIEIRA, F. C. **Controle dinâmico de robôs móveis com acionamento diferencial**. 2005. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio Grande do Norte, Natal, RN, 2005.

VINAGRE, B. M.; GODOY, A. J. C.; MARCELO, J. I. S. Teoría de control y cálculo fraccionário. **Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales**, v. 99, n. 2, p. 241-260, 2005.

WELCH, G.; BISHOP, G. **An Introduction to the Kalman Filter**. TR 95-041, University of North Carolina at Chapel Hill Department of Computer Science Chapel Hill, NC 27599-3175, 2001.

YE, Ju. **Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique**. Neuro computing. 2008.