

**CENTRO UNIVERSITÁRIO FEI
CARLOS ALBERTO FIAKOFSKI CADAMURO**

ANÁLISE DA APLICAÇÃO DE CONTROLADORES EM ROBÔS MÓVEIS

São Bernardo do Campo

2019

Carlos Alberto Fiakofski Cadamuro

ANÁLISE DA APLICAÇÃO DE CONTROLADORES EM ROBÔS MÓVEIS

Dissertação apresentada ao Centro
Universitário FEI para obtenção do título de
Mestre em Engenharia Elétrica. Orientado
pelo Prof. Dr. Flavio Tonidandel.

São Bernardo do Campo

2019

Cadamuro, Carlos Alberto Fiakofski.
ANÁLISE DA APLICAÇÃO DE CONTROLADORES EM
ROBÔS MÓVEIS / Carlos Alberto Fiakofski Cadamuro. São
Bernardo do Campo, 2019.
106 p. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Flavio Tonidandel.

1. PID. 2. PID discreto. 3. RST. 4. Robôs Móveis
Omnidirecionais. 5. BLDC. I. Tonidandel, Flavio, orient. II. Título.

Dedico este trabalho aos meus amores
Danniella e Millena que mesmo longe me
apoiaram nessa jornada longa e distante.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Agradeço ao Centro Universitário FEI pelo apoio e inclusão em suas instalações e laboratórios como um igual.

Agradeço a Equipe RoboFEI pelo apoio nas decisões e por compreenderem que apesar das diferenças somos um time!

“Deus e o Diabo é que me guiam, mais
ninguém!

Todos tiveram pai, todos tiveram mãe;

Mas eu, que nunca principio nem acabo,

Nasci do amor que há entre Deus e o
Diabo.” (RÉGIO, 2002)

RESUMO

Neste trabalho serão apresentadas algumas técnicas de controle aplicadas em um motor brushless, BLDC, utilizado em um robô terrestre omnidirecional. Será apresentado um modelo para simulação deste tipo de motor que será derivado em um modelo de estudo de caso com dados reais. A cinemática direta e a cinemática inversa desse tipo de robô foram desenvolvidas, bem como os meios de se obter um modelo dinâmico do robô omnidirecional. Baseado na pesquisa feita sobre o assunto, foram escolhidos para serem testados em simulação os controladores PID contínuo, o PID discreto e um controlador por alocação de polos na forma RST. As métricas de comparação aplicadas em simulação foram o tempo de resposta ou de subida, T_r , o tempo de assentamento, T_s , o máximo sobressinal, M_p e o erro em regime permanente, ess , que devem ser menores do que os obtidos em malha aberta. As simulações para obtenção de todos esses dados foram todas feitas no MATLAB®. Após obtidos esses dados foi escolhido um controlador para ser testado no robô omnidirecional real da equipe RoboFEI. Como o principal resultado é esperado uma melhora significativa em seu desempenho nas velocidades desenvolvidas.

Palavras-chave: PID, PID discreto, RTS, Robôs Móveis, Robôs Omnidirecionais, motor *brushless*, BLDC.

ABSTRACT

This work presents some control techniques applied in a brushless motor, BLDC, used in an omnidirectional robot. A model for simulation of this type of motor will be presented that will be derived in a case study model with real data. The direct kinematics and inverse kinematics of this type of robot were developed, as well as the means of obtaining a dynamic model for omnidirectional robot. Based on the research done, it was decided to simulate the classic PID controllers, the discrete PID and a pole allocation controller in the RST form. The comparison metrics applied in simulation were rise time, T_r , settling time, T_s , maximum overshoot, M_p , and steady-state error, e_{ss} , which should be lower than those obtained in open loop. The simulations to obtain all this data were done in MATLAB®. With this data in hand a controller was chosen to be tested in a real omnidirectional robot of the RoboFEI team. It was expected a significant improvement in its performance on speed development.

Keywords: PID, discrete PID, RTS, Mobile Robots, Omnidirectional Robots, brushless motor, BLDC.

LISTA DE ILUSTRAÇÕES

Figura 1 - Robô da Equipe RoboFEI sem capa.....	21
Figura 2 - Representação esquemática do sistema de jogo SSL.....	22
Figura 3 - Arquitetura interna do robô.	23
Figura 4 - Robôs perfilados antes de uma partida.....	24
Figura 5 - Dimensões do campo SSL.....	25
Figura 6 - Robôs embalados para viagem.	28
Figura 7 - Esquemático de um robô omnidirecional.	31
Figura 8 - Vista explodida das rodas omnidirecionais RoboFEI.	32
Figura 9 - Esquemático dos motores <i>brushless</i>	33
Figura 10 - Exemplo de controle MA.	35
Figura 11 - Exemplo de um sistema de controle MF.	36
Figura 12 - Implementação básica do formato RST.	40
Figura 13 - Diagrama de blocos do Controle por vetor de correntes.....	43
Figura 14 - Diagrama de blocos com <i>feedback</i> e <i>feedforward</i>	44
Figura 15 - Diagrama de blocos do controlador PID proposto.	46
Figura 16 - Esquemático atual da equipe RoboJackets.	47
Figura 17 - Novo esquemático da equipe RoboJackets.	47
Figura 18 - Esquemático do controle MPC implementado pela equipe.....	48
Figura 19 - Diagrama de blocos ER-Force antigo.	49
Figura 20 - Diagrama de blocos da nova arquitetura de controle da ER-Force.	50
Figura 21 - a) Robôs em preparação para o jogo. b) Robôs prontos para entrar em campo para uma partida.....	54
Figura 22 - Vista superior do robô com o plano de referência.....	56
Figura 23 - Cinemática euclidiana das rodas.	58
Figura 24 - Resposta do modelo em MA.	65
Figura 25 - Diagrama de Bode.	66
Figura 26 - representação do diagrama de blocos padrão.....	68
Figura 27 - Resposta do sistema em MF.	69
Figura 28 - Comparação entre modelos discreto e contínuo.....	71
Figura 29 - Resposta ao degrau PID discreto.	71
Figura 30 - Lugar das raízes com os polos do controlador.	73

Figura 31 - Resposta ao degrau RST em MF.....	74
Figura 32 - Representação do controlador RST no <i>Simulink</i>	75
Figura 33 - Comparativo dos controladores em MF.....	78
Figura 34 - Foto do campo da Equipe RoboFEI.	80
Figura 35 - Quadrado de 1m x 1m MF sem controlador.	82
Figura 36 - Quadrado de 1m x 1m com controlador.	83
Figura 37 - Perfil de velocidades no eixo x do robô.	84
Figura 38 - Capturas de tela do robô sem o controlador.	84
Figura 39 - Capturas de tela do robô com o controlador	85

LISTA DE TABELAS

Tabela 1 - Partes dos motores BLDC da Maxon.	34
Tabela 2 - Informações sobre o FPGA da equipe RoboFEI.....	55
Tabela 3 - Dados do Motor EC-45 flat 50W.....	63
Tabela 4 - Comportamento do sistema em função dos parâmetros do PID. ..	67
Tabela 5 - Comparativo das métricas de desempenho.	77
Tabela 6 - Comparativo das médias de velocidade.....	83

LISTA DE SIGLA E ABREVIATURAS

2D	Duas dimensões
3D	Três dimensões
4WD	Four Wheel Drive
A	Corrente em amperes
AGV	<i>Automated Guided Vehicles</i>
BLDC(s)	<i>Brushless Direct Current</i>
CA	Corrente Alternada
CBR	Competição Brasileira de Robótica
CC	Corrente contínua
CLP	Controlador lógico programável
D	Derivativo
DC	<i>Direct current</i>
EE	Espaço de estados
ess	Erro em regime permanente
FEI	Fundação Educacional Inaciana
FIFA	Federação Internacional de Futebol
FPGA	<i>Field Programmable Gate Array</i>
FT	<i>Função de transferência</i>
I	Integral
IME	Instituto Militar de Engenharia
IMU	<i>Inertial Measurement Unit</i>
NBR	Borracha de Nitrilo de Butadieno
MA	Malha aberta
MF	Malha fechada
MIMO	<i>Multiple Input Multiple Output</i>
mm	Milímetros
MOSFET	<i>Metal Oxide Semiconductor Field Effect Transistor</i>
Mp	Máximo sobressinal
MPC	<i>Model Predictive Control</i>
P	Proporcional
PD	Proporcional + Derivativo

PI	Proporcional + Integral
PID	Proporcional + Integral + Derivativo
PWM	<i>Pulse Width Modulation</i>
RST	<i>Reference, Signal (of Control) and Tracking</i>
R.U.R.	<i>Rossum's Universal Robots</i>
SDK	<i>Software Development KIT</i>
SI	Sistema Internacional de Unidades
SISO	<i>Single Input Single Output</i>
SLIT	Sistema linear invariante no tempo
SSL	<i>Small Size League</i>
TDP	<i>Team Description Paper</i>
Tr	Tempo de resposta/subida
Ts	Tempo de assentamento
US\$	<i>Dólar Americano</i>
USP	Universidade de São Paulo
Vcc	Tensão em corrente contínua
Z	Variável de tempo discreto

SUMÁRIO

1	INTRODUÇÃO	19
1.1	MOTIVAÇÃO	23
1.2	OBJETIVO	25
1.3	JUSTIFICATIVA.....	26
1.4	DOMÍNIO	27
1.5	ORGANIZAÇÃO DO TRABALHO.....	29
2	REVISÃO BIBLIOGRÁFICA	30
2.1	ROBÓTICA MÓVEL	30
2.2	MOTORES BRUSHLESS – BLDC	32
2.3	TEORIA BÁSICA DE CONTROLE	35
2.3.1	PID contínuo	36
2.3.2	PID discreto	38
2.3.3	Controle com dois graus de liberdade	39
3	TRABALHOS CORRELATOS	41
3.1	TRABALHOS RELACIONADOS NA LIGA SSL.....	41
3.1.1	Warthog Robotics	41
3.1.2	RoboIME	42
3.1.3	OP-Amp	42
3.1.4	KIKS	43
3.1.5	UBC Thunderbots	44
3.1.6	RoboTeam Twente	45
3.1.7	RoboJackets	46
3.1.8	RoboDragons	48
3.1.9	ER-Force	49
3.2	TRABALHOS RECENTES DE CONTROLE DE MOTORES BLDC	51
4	PROJETO E SIMULAÇÃO	54
4.1	O ROBÔ	54
4.1.1	Cinemática Do Robô	57
4.1.2	A Dinâmica do Robô	59
4.1.3	Modelos do motor BLDC	62

4.2	SINTONIA DOS CONTROLADORES	66
4.2.1	PID contínuo	68
4.2.2	PID discreto	69
4.2.3	Controlador na forma RST	72
4.2.4	O sinal de controle	76
4.3	COMPARATIVO ENTRE OS RESULTADOS	76
5	ESTUDO DE CASO	80
5.1	APLICAÇÃO NO ROBÔ	81
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	86
6.1	TRABALHOS FUTUROS	87
	REFERÊNCIAS	88
	ANEXO I – CÓDIGOS EM MATLAB®	95
	ANEXO II - DIAGRAMAS SIMULINK - TEMPO DE SIMULAÇÃO 0,5S:	102

1 INTRODUÇÃO

Robôs são máquinas que tem interessado a maior parte da população através da história. Em geral eles são máquinas ou dispositivos operados automaticamente ou por controle remoto (KURFESS, 2005). Esse interesse é confirmado nos diversos dispositivos desenvolvidos atualmente, como aspiradores de pó robôs, drones, e outros tantos que vem juntar esse fascínio com a necessidade humana dos tempos modernos.

Apesar do termo robô derivar da palavra tcheca *robotá*, que foi introduzida na peça teatral R.U.R, *Rossum's Universal Robots*, escrita na década de 20 por Karel Čapek (WIKIPÉDIA, A ENCICLOPÉDIA LIVRE, 2018), ele hoje relaciona uma diversidade de máquinas, equipamentos e dispositivos capazes de realizar tarefas de forma autônoma ou pré-programada.

As pesquisas em robótica se desenvolveram principalmente para a indústria, como uma forma de melhora da qualidade dos produtos e aumentar a quantidade produzida (SCATENA, 2008).

Atualmente o mercado global de sistemas robóticos tem projeções de crescimento de 14% ao ano para os próximos três anos e encerrou o ano de 2017 com valores próximos a US\$ 48 bilhões (BAUER, 2018).

Outra área de aplicação é a área médica, onde robôs são utilizados tanto em cirurgias quanto em reabilitação de pacientes (ROSÁRIO, 2005), com muitos mais usos possíveis com o avanço da robótica.

A flexibilização das plantas industriais traz novos desafios na área de robótica, onde um robô antes fixo em uma posição passa a ser necessário em vários pontos da manufatura (SIEGWART e NOURBAKSH, 2004). O interesse da comunidade cresce na área de robótica móvel, um novo expoente na robótica industrial, principalmente devido aos robôs de serviço, que cresceu 85% entre 2016 e 2017 e tende a continuar crescendo (BAUER, 2018).

Existem diversas formas de desenvolvimento de tecnologia em robótica móvel utilizadas pela comunidade científica. Vale ressaltar aqui uma outra comunidade que foca particularmente nessa classe de robôs, a RoboCup. A RoboCup é uma comunidade que promove a robótica móvel e a pesquisa em Inteligência Artificial (ROBOCUP FEDERATION, 2016) através de competições amigáveis de futebol onde

cada categoria aborda um ou mais temas necessários ao desenvolvimento de um jogador de futebol robótico que possa competir com humanos.

Os robôs móveis por princípio têm que se locomover. Essa locomoção pode ser por terra, por ar, ou pela água (GARCIA, JIMENEZ, *et al.*, 2007). Nas categorias de futebol da RoboCup, a locomoção sempre ocorre de forma terrestre.

Dentre todas as classes de robôs que se locomovem no meio terrestre, para as aplicações industriais, se destacam os que utilizam rodas como forma de locomoção. Os robôs que se locomovem por rodas são, de longe, os mais comuns (SIEGWART e NOURBAKSH, 2004), pois se adaptam bem a diversas aplicações, tem boa manobrabilidade e grande capacidade de carga.

As plantas industriais vêm se otimizando em relação a espaço útil no passar dos anos. Essa otimização passou a exigir dos robôs móveis uma melhoria significativa na manobrabilidade desses robôs.

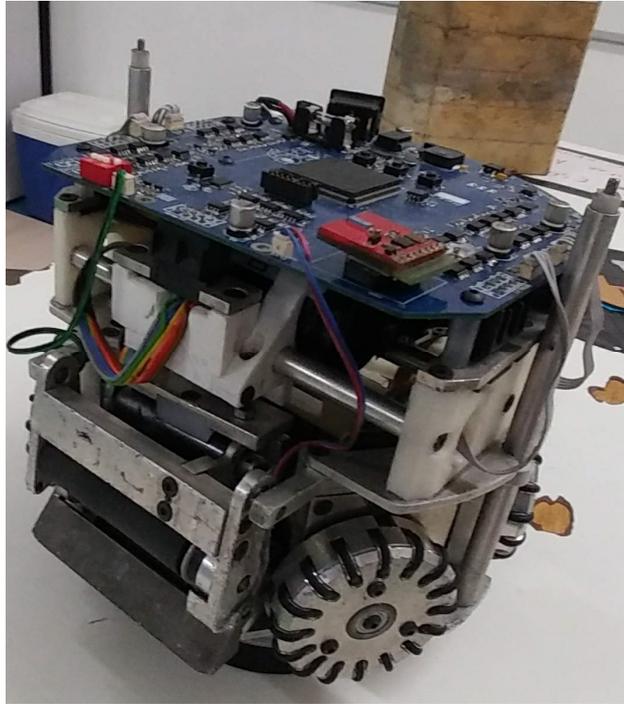
Os robôs com rodas fixas, também chamados de não-holonômicos, que não podem se mover de forma paralela aos seus eixos, necessitam de um amplo campo de giro para realizar uma curva, ou seja, eles possuem movimentos restritos (PIN e KILLOUGH, 1994). Uma alternativa a esses robôs é uma classe de robôs chamados omnidirecionais, que podem ser holonômicos quando podem se mover em qualquer direção a qualquer momento (SIEGWART e NOURBAKSH, 2004), como é o caso do robô sob estudo. Os robôs omnidirecionais podem se locomover em quantas dimensões forem necessárias para descrever a sua posição, ou seja, no plano eles deixam de ter frente e lateral e passam a se locomover em todas as direções e rotacionar.

Existe na RoboCup uma categoria que se encaixa perfeitamente nesses padrões, a categoria SSL, *Small Size League*, que possui robôs omnidirecionais com 4 rodas motorizadas que necessitam de controle em diversos níveis, desde o seu motor *brushless* até o seu posicionamento em campo.

Os robôs da categoria SSL devem ter um diâmetro máximo de 180mm e uma altura máxima de 150mm. Além disso, como é utilizado um sistema de visão padronizado, os robôs devem ter capas identificadoras para o sistema de visão poder identificá-los. Um robô da equipe RoboFEI pode ser visto na Figura 1. O sistema frontal de drible e chute que também pode ser observado na figura, deve permitir que outro robô possa retirar a bola em uma eventual disputa. Para isso só é permitido uma cobertura da bola de 20% devendo os restantes 80% estar livre para esta disputa de

bola. Outras especificações estão disponíveis nas regras e podem ser verificadas no próprio sítio da RoboCup (COMMITTEE, 2018).

Figura 1 - Robô da Equipe RoboFEI sem capa.



Fonte: Autor.

As capas protegem as partes internas do robô e o topo das capas utilizam um sistema de cor que mostra para o sistema de visão de que time é o robô, no caso amarelo ou azul, e qual o número dele (ZICKLER, LAUE, *et al.*, 2009).

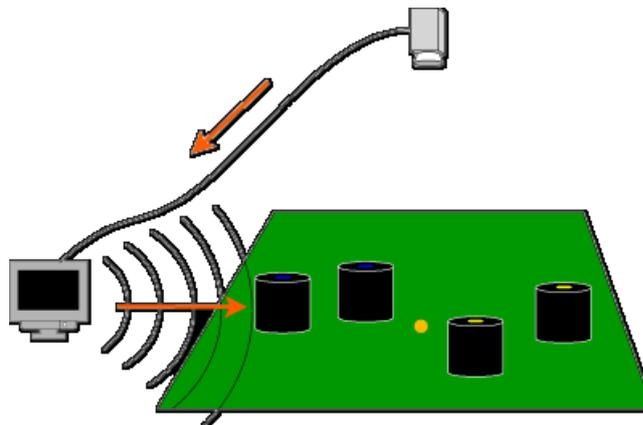
Como a robótica móvel possui uma necessidade de interdisciplinaridade de várias áreas do conhecimento como Mecatrônica, Tecnologias de Informação, Inteligência Artificial, Sistemas de Comunicação, Engenharia de Controle dentre outras (ROSÁRIO, 2005) ela se torna um objeto de estudo bastante central nessa área de pesquisa.

O estudo de técnicas de controle para esse tipo de robôs é um tema recorrente na comunidade e foi realizado em diversas abordagens diferentes. Em alguns trabalhos a técnica de controle era aplicada com foco na trajetória, buscando uma forma do robô omnidirecional seguir mais rapidamente uma trajetória pré-estabelecida (WATANABE, 1998).

Outros estudos buscam uma correção dinâmica das rodas, tanto motorizadas como com direção controlada para melhorar o seguimento de trajetória gerada remotamente e assim melhorar o desempenho dos robôs omnidirecionais holonômicos (HOLMBERG e KHATIB, 2000).

Na Figura 2 podem ser vistos os diversos sistemas que envolvem a comunicação do sistema de visão, que decodifica a imagem em informações que são passadas via rede para o computador da estratégia de cada time onde são calculadas velocidades e direções que são passadas aos robôs via um sistema de rádio proprietário. Cada robô transforma essa informação em parâmetros que são aplicados aos motores e demais dispositivos.

Figura 2 - Representação esquemática do sistema de jogo SSL.



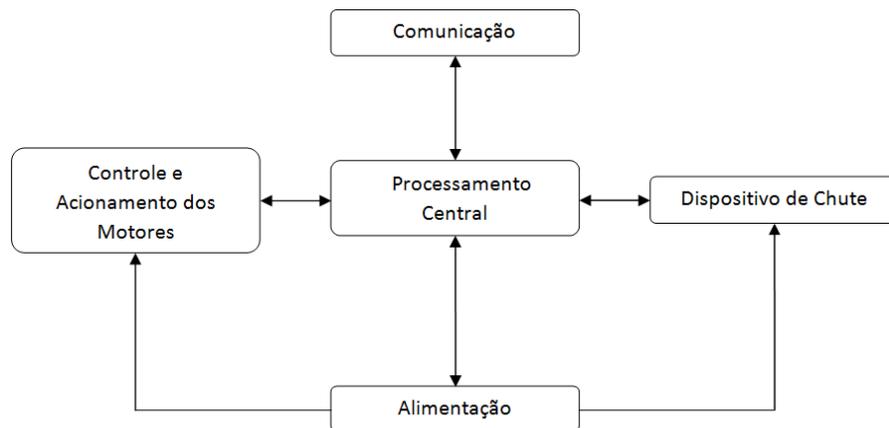
Fonte: Adaptado de (ROBOCUP FEDERATION, 2016).

O sistema de visão possui uma malha de controle no computador da estratégia que controla a posição dos robôs, e esta malha de controle corrige a posição dos robôs em campo, bem como a sua orientação, e envia essa informação na forma de velocidades individuais para cada robô. Essas informações são decodificadas no próprio robô e são aplicadas aos motores através de um controlador que será escolhido nesse trabalho.

Nesse contexto a Figura 3 mostra a arquitetura interna do robô da equipe RoboFEI, onde a comunicação é um sistema de rádio bidirecional onde são enviadas e recebidas mensagens entre o computador da estratégia e o robô. O processamento central controla as funções principais do robô, como a própria comunicação, manda os comandos de chute e acionamento dos motores. A alimentação é um sistema que

fornece energia para todo o robô e onde existe um monitoramento de carga da bateria onde esta informação é enviada ao computador da estratégia. O Sistema de chute controla o drible e o tipo de chute que será dado bem como a sua intensidade. O controle e acionamento dos motores é onde será aplicado o controlador escolhido neste trabalho.

Figura 3 - Arquitetura interna do robô.



Fonte: Autor.

Existe uma diversidade de motores elétricos utilizados para locomoção de robôs, tanto de corrente alternada como de corrente contínua. No universo de motores de corrente contínua uma classe específica deles chama a atenção quando se busca desempenho e economia: os motores BLDCs, *Brushless Direct Current Motors*.

Como os robôs disponíveis da categoria SSL no Centro Universitário da FEI utilizam este tipo de motor, o presente trabalho também utilizará esses motores *brushless* como exemplificação e testes do que será proposto.

1.1 MOTIVAÇÃO

A RoboCup é uma competição pensada para promover o desenvolvimento da robótica e da inteligência artificial. Ela cumpre com o seu propósito por meio de competições amistosas em diversas categorias, cada qual com foco em uma subárea da robótica ou da inteligência artificial ou ambas (ROBOCUP FEDERATION, 2016).

Neste trabalho o interesse recai sobre a categoria RoboCup SSL, *Small Size League*, que tem seu foco na inteligência multiagentes e controle em ambiente

altamente dinâmico com um híbrido de sistemas centralizados e outros distribuídos (COMMUNITY, 2008).

Esta competição surgiu na década de 90 e segue até os dias de hoje com várias modificações em sua estrutura. A equipe RoboFEI participa desta categoria desde 2008 na CBR, Competição Brasileira de Robótica, onde conseguiu diversos resultados expressivos. A Figura 4 mostra o time de robôs da equipe RoboFEI se preparando para um jogo da liga SSL.

Figura 4 - Robôs perfilados antes de uma partida.



Fonte: Autor.

No princípio da categoria as partidas de futebol eram feitas com poucos robôs e sem restrições de força de chute ou velocidade, o que levou a categoria a desenvolver estratégias de jogo baseadas em chutes extremamente fortes de longa distância.

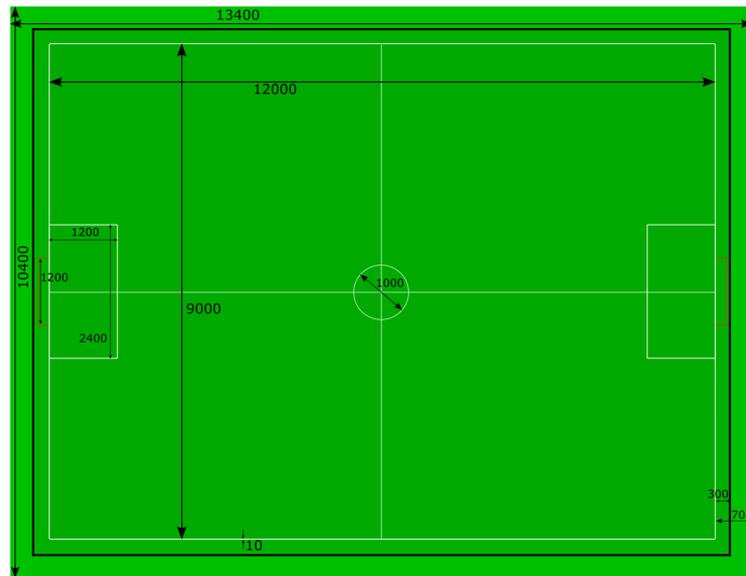
Nesse momento inicial da liga existiu algum esforço dos times para o desenvolvimento da robótica propriamente dita com trabalhos apresentados na área de controle de movimentação, controle de motores, melhoria da dinâmica dos robôs, entre outros, mas logo a comunidade abandonou este desenvolvimento devido ao pequeno ganho proporcionado em relação ao esforço despendido.

Recentemente ocorreram mudanças nas regras que limitaram a velocidade da bolinha para 6,5m/s e um aumento significativo na área do campo que passa de 9m x

6m para 12m x 9m, o que dificulta muito a estratégia do chute forte em direção ao gol. As dimensões e o formato do campo podem ser observados no esquemático mostrado na Figura 5.

Essas mudanças reacenderam o interesse da comunidade em melhorar a movimentação dos seus robôs e, consecutivamente, em desenvolver estratégias de controle de *hardware* em baixo nível, diretamente nos robôs.

Figura 5 - Dimensões do campo SSL.



Fonte: Adaptado de (ROBOCUP FEDERATION, 2016).

É aqui que este trabalho se encaixa nas aspirações da equipe RoboFEI e da liga SSL na RoboCup.

1.2 OBJETIVO

O objetivo deste trabalho é analisar a aplicação de três tipos de controladores, propostos pela literatura ou utilizados por outras equipes, para os robôs omnidirecionais da equipe RoboFEI e com isso melhorar o seu desempenho.

Esses controladores são o PID clássico (Proporcional + Integral + Derivativo), o PID discreto, o controlador por alocação de polos aplicado da forma RST, *Two Degrees of Freedom Digital Controller*, cujo nome deriva do inglês relativo a *Reference, Signal (of Control) and Tracking* (LANDAU, 1998), todos estes tratando de uma planta

via função de transferência com sintonia através dos métodos apresentados por outras equipes, mostrados na seção 3.1.

1.3 JUSTIFICATIVA

O problema de controle de motores *brushless* é bastante antigo, o que significa que já foi largamente explorado e estudado em pesquisas pelo mundo todo (KANGSANAT, LIM e HARVEY, 1988). Devido a isso, foram encontrados trabalhos semelhantes na comunidade científica nacional e internacional.

Destacam-se aqui o trabalho *Description of the Warthog Robotics SSL 2017* que relata um desenvolvimento em sua eletrônica embarcada onde o ponto principal foi a modificação do PID que era sintonizado de forma contínua e foi adaptado para um PID sintonizado diretamente na forma discreta que depois foi aplicado um filtro no termo derivativo (LANG, OLIVEIRA, *et al.*, 2017).

Também as pesquisas apresentadas em *RoboIME: on the road to RoboCup 2017* que mostra um grande esforço no sentido de sintonizar novamente o PID de suas rodas para melhorar tanto o deslizamento delas no tapete de competição como para melhorar a sua aceleração e desaceleração (CONSENZA, COUTO, *et al.*, 2017).

Para os estudos simulados será utilizado o MATLAB® R2018a e para um estudo de caso em que será implementado experimentalmente em um robô real da equipe RoboFEI na atual plataforma o compilador próprio da Xilinx, o Xilinx SDK – *Software Development KIT Release 13.1 Build SDK_O.40d*, cuja linguagem é C. Este código é compilado e é transformado de uma forma que o FPGA, *Field Programmable Gate Array*, em português "Arranjo de Portas Programáveis em Campo", do robô consiga executá-lo.

Como a arquitetura atual do *hardware* do robô é bastante restrita a mudanças, o levantamento de um modelo experimental dos motores não é possível. Com a finalidade de contornar este obstáculo será utilizado em simulação um modelo obtido através das relações entre a indutância, a resistência e o torque do motor, como por exemplo o modelo proposto por Baldursson, (2005) e largamente citado e utilizado por toda comunidade que utiliza o MATLAB® como ferramenta de simulação. Existem outros modelos como o proposto por MACDOUGALL, ELLIS, *et al.*, (2018), ou ainda os modelos propostos em estudos independentes como uma nova forma de controlar

motores BLDC por uma nova abordagem de Patel e Pandey (2013). Para a utilização do modelo os dados do motor serão retirados do catálogo do fabricante do motor utilizado pela equipe, a Maxom Motors.

Em simulação serão utilizados critérios clássicos de performance de controladores como

- Tempo de Subida, T_r ;
- Máximo Sobressinal, M_p ;
- Tempo de Assentamento, T_s e;
- Erro em regime permanente.

No caso deste estudo compara-se o comportamento do robô com o controlador escolhido e sem ele para validar a proposta.

Estas medições são possíveis e serão feitas através do sistema de visão padronizado da liga SSL disponível no laboratório da equipe RoboFEI. Este sistema de visão é um projeto *open source* mantido por toda a comunidade SSL. Foi desenvolvido para unificar a liga com um sistema confiável e acessível para todo o mundo (ZICKLER, LAUE, *et al.*, 2009).

1.4 DOMÍNIO

O domínio de estudo deste trabalho é a RoboCupSoccer (ROBOCUP FEDERATION, 2016), uma competição de futebol feita por robôs autônomos em diversas categorias. O objetivo da RoboCup é desenvolvimento da robótica e da inteligência artificial.

Esse desenvolvimento é alcançado por meio de competições amigáveis em cenários multiagentes colaborativos e com múltiplos objetivos em um ambiente dinâmico formado por robôs totalmente autônomos.

O objetivo principal da RoboCup é desenvolver a robótica colaborativa a um nível em que em 2050 o Campeão mundial da RoboCup dispute uma partida oficial da FIFA, Federação Internacional de Futebol, com o Campeão da Copa do Mundo de Futebol do mesmo ano e vença a partida (ROBOCUP FEDERATION, 2016).

Para a robótica atingir este nível de sofisticação existem hoje oito ligas na RoboCup que tentam contemplar boa parte das minúcias necessárias para tal feito. São elas (ROBOCUP FEDERATION, 2016):

- *Humanoid Leagues:*
 - *Kid Size;*
 - *Teen Size;*
 - *Adult Size;*
- *Middle Size League;*
- *Small Size League;*
- *Standard Platform League;*
- *Simulation League 2D;*
- *Simulation League 3D.*

Cada uma dessas ligas possui regras próprias cobrindo desde tamanho de campo até número de robôs em campo permitidos, algumas com mínimos e máximos. As regras de cada liga são alteradas em momentos oportunos devido a atualizações tecnológicas ou a estagnação da categoria.

Neste contexto, este trabalho vai focar nos robôs chamados F180, que possuem este nome devido ao diâmetro máximo permitido, da *Small Size League* – SSL. Esses robôs possuem locomoção omnidirecional com tração nas quatro rodas realizada por motores *brushless* com capacidade de sensoriamento por meio de sensores *Hall*. A Figura 6 mostra quatro robôs da Equipe RoboFEI embalados para viagem a competição RoboCup 2018.

Figura 6 - Robôs embalados para viagem.



Fonte: Autor.

Esses motores, chamados também de motores BLDC serão os motores controlados ao fim deste trabalho.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado de forma a fornecer meios e teorias para o leitor entender as conclusões. Para tanto essa organização se dá na seguinte forma.

Na seção 2 são apresentados alguns dos conceitos básicos para o desenvolvimento do trabalho. É feita uma breve introdução aos conceitos da engenharia de controle bem como aos conceitos de robótica. Também são explicados os fundamentos dos motores BLDC.

Na seção 3 são apresentados os trabalhos atuais desenvolvidos na área, tanto para os motores BLDC como para robôs similares participantes da liga SSL.

Na seção 4 são apresentados os modelos do motor BLDC, a cinemática do robô omnidirecional de quatro rodas, a modelagem dinâmica, os controladores propostos e a métrica de validação teórica que será utilizada.

Na seção 5 está um estudo de caso da aplicação de um dos controladores estudado em um robô real da equipe RoboFEI.

Finalmente, na última seção são apresentadas as considerações finais deste trabalho e algumas sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Esta seção apresenta a revisão bibliográfica sobre a robótica móvel, fundamento sobre motores BLDC e a teoria de controle, tudo isso com um direcionamento para a arquitetura atual do robô da equipe RoboFEI, apresentando conceitos básicos da área de forma a fundamentar as técnicas mostradas e descrever o funcionamento dessas técnicas.

2.1 ROBÓTICA MÓVEL

A robótica é uma área que engloba pesquisa, desenvolvimento e aplicação de recursos para a realização de determinadas tarefas, sejam elas repetitivas ou não. Os robôs industriais em geral são fixos e possuem braços articulados que executam tarefas nas linhas de montagem ou fabricação, principalmente no que se refere ao transporte de materiais de uma estação para outra ou na própria estação de trabalho (SIEGWART e NOURBAKHS, 2004).

A necessidade de transportar coisas motivou a criação de uma nova categoria de robôs, os robôs móveis, que são dispositivos mecânicos montado sobre uma base não fixa, que age sob o controle de um sistema computacional provido de sensores e atuadores que possibilitam essa movimentação. Estes robôs podem estar em três subgrupos quanto ao meio em que se locomovem (GARCIA, JIMENEZ, *et al.*, 2007):

- Aéreos;
- Aquáticos;
- Terrestres.

Neste trabalho serão abordados especificamente de robôs terrestres móveis. A robótica móvel é uma subárea da robótica que também utiliza conceitos multidisciplinares de diversas áreas, como engenharia, ciência da computação, eletrônica, mecânica, automação, entre outros como a sua *area matrem*.

Os robôs terrestres móveis ainda se dividem em três tipos principais, os movidos por pernas, os movidos por esteiras e os movidos por rodas (PIERI, 2002). A roda tem sido de longe o mecanismo de locomoção mais popular em robótica móvel e em veículos feitos pelo homem em geral. Pode alcançar eficiências muito boas, e

faz isso com uma implementação mecânica relativamente simples (SIEGWART e NOURBAKHS, 2004).

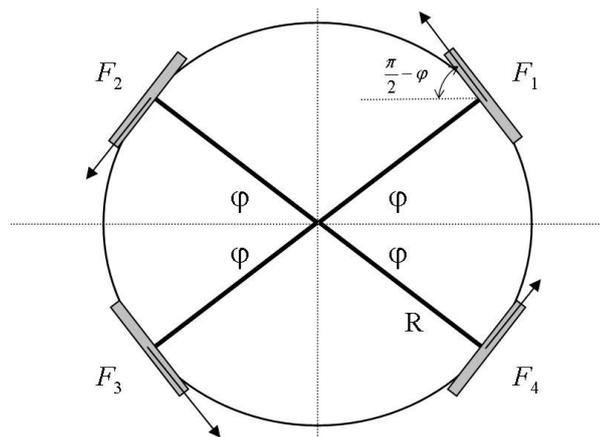
Os primeiros robôs móveis datam de 1968 e consistiam principalmente de veículos teleguiados (*Automated Guided Vehicles – AGVs*), utilizados para transportar ferramentas em ambientes industriais e seguindo uma trajetória predefinida (GARCIA, JIMENEZ, *et al.*, 2007).

A robótica móvel terrestre por rodas se divide em duas subcategorias sendo uma de robôs não-holonômicos e uma de robôs omnidirecionais. Essa diferenciação se deve as restrições de movimentação que os primeiros têm, pois não podem se mover paralelamente ao seu eixo.

Os robôs omnidirecionais, objetos deste estudo, por outro lado movem-se em qualquer direção a partir de qualquer configuração inicial, independentemente da sua orientação frontal. Isso quer dizer que este tipo de robô controla de forma independente todos os graus de liberdade no espaço bidimensional.

Os veículos omnidirecionais desse estudo possuem uma disposição de rodas como pode ser observada abaixo na Figura 7. Cada roda possui um motor BLDC e a figura mostra suas forças de acoplamento entre o solo e a roda.

Figura 7 - Esquemático de um robô omnidirecional.



Fonte: Adaptado de (ROJAS e FÖRSTER, 2006).

As forças de acoplamento entre a roda e o solo são denotadas por F_1, F_2, F_3 e F_4 . Essas rodas possuem um arranjo de desenho especiais, muito similar as rodas suecas de 90° , que possuem tração num sentido e permitem movimento passivo noutro. As rodas utilizadas pela equipe RoboFEI podem ser vistas na Figura 8 e são classificadas

como rodas universais. O robô da equipe RoboFEI possui um ângulo de 33° em relação aos eixos, ou seja, na Figura 7 o ângulo $\varphi = 33^\circ$ para o eixo das rodas universais.

Figura 8 - Vista explodida das rodas omnidirecionais RoboFEI.



Fonte: Autor.

Elas são montadas em um corpo usinado com rodas transversais cobertas com *o'rings* num total de 16 rodas, seu apoio se dá por meio de 2 rolamentos 5X13X4 ZZ flangeado tudo fixado por parafusos M3 x 6mm com cabeça escareada, acionamento por sextavado interno de 2,5mm e com uma engrenagem interna que proporciona uma relação de 3:1 do motor para a roda.

Essas rodas possuem um diâmetro nominal com os *o'rings* instalados 58mm e sem eles 55 mm e tem uma proximidade máxima com o solo de 1,302mm quando dois *o'rings* estão em contato estável com o solo. Os *o'rings* tem uma especificação de diâmetro interno de 6mm e seção de 2,6mm em NBR, Borracha de Nitrilo de Butadieno ou simplesmente borracha nitrílica.

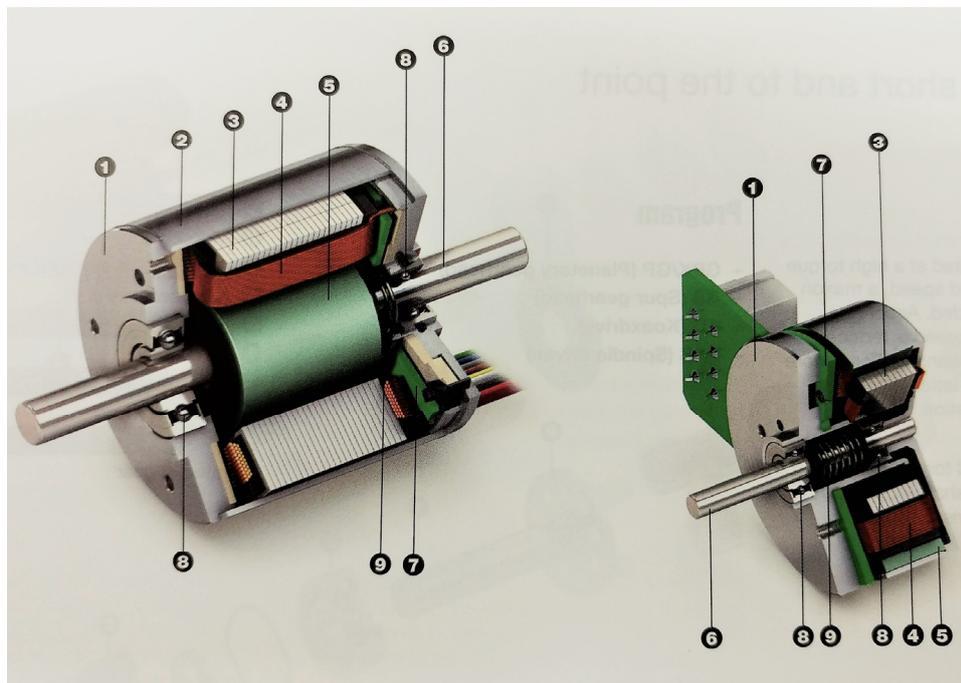
2.2 MOTORES BRUSHLESS – BLDC

Os motores BLDC nada mais são do que motores de corrente contínua sem escovas, podendo também ser constituídos de ímãs permanentes ligados a um eixo ou um cilindro rotativo, que são empurrados e/ou puxados por campos eletromagnéticos dos enrolamentos elétricos da carcaça do motor.

Seu funcionamento é idêntico ao dos mais comuns motores CC, de Corrente Contínua, com escovas, porém com os campos permanente e excitado invertidos. Algumas vezes eles são chamados de *inside-out CC motors* porque sua armadura está no estator e os ímãs no rotor (BALDURSSON, 2005).

A inversão das fases que no motor CC se dá pelo comutador de escovas, e no motor *brushless* acontece por um arranjo de transistores do tipo MOSFET de alta potência. Na Figura 9 aparece uma vista parcial de motores BLDC.

Figura 9 - Esquemático dos motores *brushless*.



Fonte: Catálogo da Maxon Motor (AG, 2017).

Nessa figura são mostrados os dois principais tipos de motores da Maxon Motors sendo o da esquerda tradicional e o da direita chamado *flat*. Na tabela 1 encontram-se as legendas para as partes destacadas na foto.

Os motores BLDC são acionados por impulsos de tensão retangulares, acompanhado de uma dada posição do rotor. O fluxo magnético gerado no estator interage com o fluxo do rotor, o qual é gerado pelos ímãs do rotor, definindo o torque e, assim, acelerando do motor.

Esse tipo de motor em geral fornece uma relação peso x potência significativamente melhor e uma maior eficiência do que os motores escovados tradicionais (HUI, BASU e SUBBIAH, 2003). Os motores BLDC podem ser usados

numa ampla variedade de aplicações. Neste trabalho eles são utilizados no robô móvel.

Tabela 1 - Partes dos motores BLDC da Maxon.

Parte	Unidade
1	Flange
2	Encapsulamento
3	Pilha de aço laminado
4	Enrolamento
5	Ímã permanente
6	Eixo
7	Circuito impresso para os sensores <i>hall</i> .
8	Rolamento de esferas
9	Mola de pré-carga do rolamento

Fonte: O autor.

Os motores do projeto são de corrente contínua, sem escovas (*brushless*) e com três fases. A principal vantagem de se usar motor sem escova é a sua vida útil maior, a redução de interferências eletromagnéticas e sua eficiência que minimiza as perdas (HUI, BASU e SUBBIAH, 2003).

Porém, esses motores são significativamente mais caros e necessitam de um *driver* de controle mais sofisticado, mas que com o barateamento da microeletrônica se torna viável (HUI, BASU e SUBBIAH, 2003).

Dentro de cada motor existem três sensores de efeito *Hall*, um para cada fase, responsáveis por analisar os movimentos dos ímãs permanentes presentes no rotor e, através do campo magnético gerado, retornar essas informações para um controlador que retorna o próximo pulso.

Os sinais de controle PWM, *Pulse Width Modulation*, enviados pelo controlador não tem corrente suficiente para a ativação dos enrolamentos dos motores nem atingem a tensão necessária para tal. Além disso, seria bastante arriscado ter uma comunicação direta entre eles, pois qualquer defeito nos motores poderia danificar o controlador por completo. Para resolver esse problema, é utilizado um circuito de *driver* entre eles que amplifica os sinais enviados pelo controlador e drenará toda a corrente necessária para os motores.

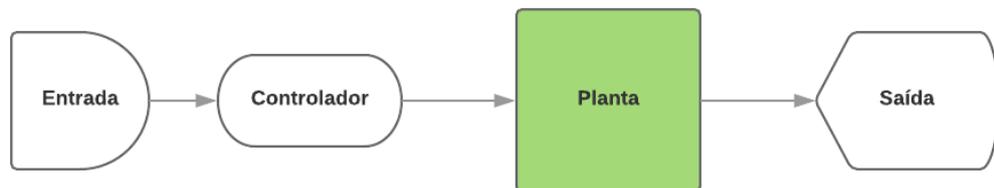
2.3 TEORIA BÁSICA DE CONTROLE

As aplicações da engenharia de controle são as mais vastas possíveis, indo desde um simples banho em que se deseja uma determinada temperatura na água do chuveiro até o lançamento de um foguete para o espaço.

A ideia principal por trás de um sistema de controle é se manipular algum tipo de atuador para obter uma resposta desejada. Para atingir esse objetivo existe uma infinidade de abordagens, sendo algumas mais comuns, outras nem tanto (CASTRUCCI, BITTAR e SALES, 2011).

As principais abordagens da engenharia de controle são as estratégias em Malha Aberta, MA e em Malha Fechada, MF. A abordagem em MA usa um controlador ligado diretamente a planta de forma a se obter a saída desejada, como pode ser observado na Figura 10. No controle em MA as ações de controle independem da saída do sistema, ou seja, elas estão desconectadas.

Figura 10 - Exemplo de controle MA.



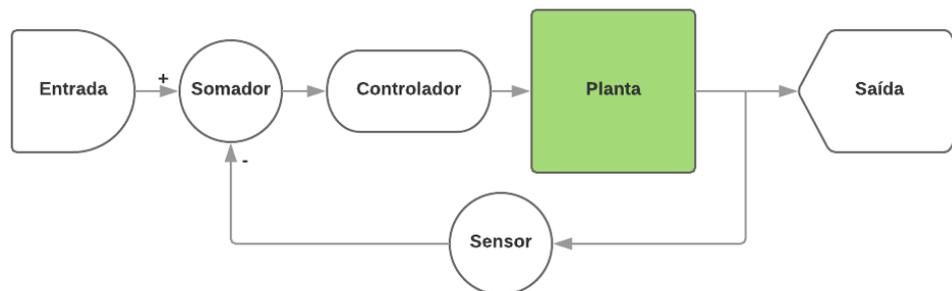
Fonte: O autor.

A abordagem em Malha Fechada é a mais comumente utilizada nos problemas tradicionais de controle. Nesse caso o controlador recebe a informação da saída através de algum tipo de sensor comparada com o valor de referência da entrada do sistema, como pode ser visto na Figura 11. As ações de controle passam a ser pautadas nas informações da entrada e da saída do sistema e quando essas informações forem equivalentes sabe-se que a saída atingiu o valor esperado (CASTRUCCI, BITTAR e SALES, 2011) (OGATA, 2000).

Os sistemas podem ser do tipo SISO, *Single Input Single Output*, onde eles têm somente uma entrada e uma única saída ou MIMO, *Multiple Input Multiple Output*, em que eles possuem uma ou mais entradas e uma ou mais saídas sendo pelo menos mais que uma entrada ou saída (MAYA e LEONARDI, 2014).

Essas abordagens podem ser usadas tanto em modelos com Função de Transferência, FT, que é uma representação matemática do sistema que equivale a razão da saída pela entrada do sistema e é usada em sistemas SISO (OGATA, 2000), como em modelos em Espaço de Estados, EE, muito mais abrangentes e poderosos do que as FT (MAYA e LEONARDI, 2014) e que é uma representação matricial dos sistemas tanto SISO quanto MIMO.

Figura 11 - Exemplo de um sistema de controle MF.



Fonte: Autor.

O objeto de controle é um motor BLDC, portanto é um SLIT, Sistema Linear Invariante no Tempo. Ele também é um sistema do tipo SISO, onde aplica-se uma tensão nas bobinas e ele retorna à velocidade no eixo do motor. Outro fato que é bom ser ressaltado aqui é que para esse motor considera-se as condições iniciais nulas e ignoram-se as variáveis internas do sistema.

Apesar do Modelo em EE ser válido em todas as características explanadas acima, será utilizado a abordagem em Função de Transferência para as devidas comparações.

2.3.1 PID contínuo

Apesar de nos últimos anos ter existido um grande esforço no estudo das técnicas de controle ótimo e controle robusto e que tem produzidos ótimos resultados, ainda hoje muitos computadores industriais e controladores lógicos programáveis, CLPs, ainda utilizam o PID internamente (CASTRUCCI, BITTAR e SALES, 2011).

Isso acontece principalmente pelo controlador PID ser usado desde a década de 50 em sua forma padronizada eletrônica, montada com amplificadores operacionais, até hoje, usado na sua forma digital (CASTRUCCI, BITTAR e SALES, 2011).

O controlador PID é uma junção dos termos proporcional, integrador e derivativo do sinal do erro, formado pela diferença entre o sinal de referência e o sinal de saída, ou conforme as equações abaixo.

O termo proporcional, equação 1, é a multiplicação de um ganho K_p multiplicado pelo erro:

$$P = K_p * e(t) \quad (1)$$

O termo integral, equação 2, é a multiplicação de um ganho K_i pela integral do erro:

$$I = K_i * \int_0^t e(\tau) d\tau \quad (2)$$

O termo derivativo, equação 3, é a multiplicação de um ganho K_d pela derivada do erro:

$$D = K_d * \frac{de(t)}{dt} \quad (3)$$

A saída do controlador é a soma algébrica dos termos, Equação (4), cujo resultado é aplicado diretamente no atuador da planta e o erro é a diferença entre a saída e a referência. Para utilização deste tipo de controlador deve-se somente dimensionar adequadamente os valores dos ganhos K_p , K_i e K_d .

$$u(t) = K_p * e(t) + K_i * \int_0^t e(\tau) d\tau + K_d * \frac{de(t)}{dt} \quad (4)$$

$$e(t) = ref(t) - saída(t)$$

As equações do controlador PID podem ser escritas de diversas formas, mas sempre mantendo a ideia apresentada acima. Nestes termos, ainda pode-se zerar os termos I e D e se obterá um controlador do tipo P, zerando-se somente o termo I obtêm-se um controlador do tipo PD e zerando o termo D obtêm-se um controlador do tipo PI. A sintonia destes controladores deve ser feita de acordo com a regras básicas de sintonia manual (ÅSTRÖM e HÄGGLUNG, 1995).

A aplicabilidade de controladores contínuos é baixa em sistemas que não possuem computador. Se a aplicação possuir um computador com a ferramenta computacional MATLAB® fica fácil de aplicar controladores contínuos em plantas reais, pois o próprio MATLAB® já discretiza internamente esses controladores com métodos próprios, mas apresenta para o usuário uma versão ainda contínua do controlador. Não é o caso da maioria dos robôs que possuem sistemas mais simples e, portanto, os controladores devem ser discretizado para serem aplicados.

2.3.2 PID discreto

Uma forma de resolver este tipo de problema é a discretização do PID e sua aplicação com um filtro para a resolução do termo diferencial. Essa solução pode ser reescrita como uma equação a diferenças que pode ser implementada em um *hardware* mais simples (LANDAU, 1998).

Partindo da equação genérica do PID clássico apresentada na Equação (4), e com a definição de erro apresentada juntamente como sendo a diferença entre a referência e a saída do sistema, pode-se iniciar a discretização para algum dos modelos disponíveis na literatura, como os que utilizam discretização direta por Séries de Taylor ou Séries de Euler para gerar a equação discreta no plano z do controlador, ou ainda utilizando as aproximações numéricas como a de Tustin para criar um PID discreto (CASTRUCCI, BITTAR e SALES, 2011).

Uma adaptação bem comum é a que utiliza uma relação entre a localização dos zeros do controlador e os seus ganhos (ÅSTRÖM e HÄGGLUNG, 1995). A Equação (5) mostra a equação do controlador e as relações entre os ganhos da equação a diferenças e os zeros do controlador. Os zeros podem ser escolhidos da forma que o projetista julgar mais conveniente.

$$G_{PID}(z) = q_0 \frac{(z - z_1)(z - z_2)}{z^2 - z} \quad (5)$$

$$z_1 + z_2 = \frac{-q_1}{q_0} \quad e \quad z_1 z_2 = \frac{q_2}{q_0},$$

onde tem-se z_1 e z_2 como os zeros do controlador e q_1 , q_2 e q_0 como os ganhos do controlador na equação a diferenças. Vale ressaltar que z é a variável complexa

discreta e que o ganho q_0 ainda pode representar uma relação entre o sinal de controle e o erro do sistema naquele instante. Uma análise de pior caso seria um degrau de valor máximo, ou seja, o motor ir para sua tensão máxima e ele estar anteriormente em repouso, como apresentado na Equação (6)

$$q_0 \cong \frac{u(0)}{e(0)} \quad (6)$$

A antitransformada Z pode ser aplicada para se obter a equação a diferenças que seria implementada em um controlador, um *hardware*, que executa os cálculos para a aplicação na planta, no caso no motor. A Equação (7) mostra a equação a diferenças do controlador

$$u(k) = u(k - 1) + q_0 e(k) + q_1 e(k - 1) + q_2 e(k - 2), \quad (7)$$

onde o $e(k)$ denota o erro do sistema e k é a variável discreta de tempo, que varia de $k=0; 1; 2; \dots; \infty$.

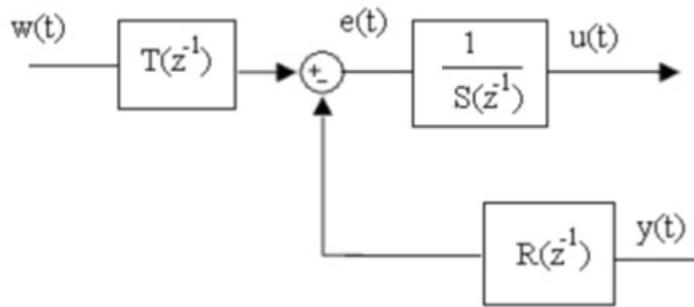
2.3.3 Controle com dois graus de liberdade

Para a implementação de controladores em tempo real, torna-se necessário a utilização de controladores discretos no tempo como o PID discreto. No projeto de um controlador através de alocação de polos, utilizados no projeto de um controlador discreto, utiliza-se uma lei de controle linear, designada forma RST, *Reference, Signal (of Control) and Tracking*. Muitas estratégias de controle podem ser aplicadas para o projeto do controlador RST por uma reformulação adequada (ÅSTRÖM e WITTENMARK, 1990) (LANDAU, 1998).

O formato RST é bastante utilizado por ser fácil de ser aplicado em tempo real e fácil de ser implementado como uma equação a diferenças. O modelo básico do controlador com dois graus de liberdade, o RST, pode ser observado na Figura 12.

A forma de implementação RST é interessante, pois facilita a aplicação com equações a diferença. Existe uma comparação do controlador PID implementado de forma convencional e o mesmo controlador PID na forma RST que mostra saídas idênticas (CORRÊA, SILVA e DEDINI, 2010).

Figura 12 - Implementação básica do formato RST.



Fonte: Adaptado de (CORRÊA, SILVA e DEDINI, 2010).

Da Figura 12 é fácil derivar a equação do controlador que pode ser vista na Equação (8). As equações relativas ao RST serão obtidas quando o método de projeto estiver definido pelo projetista. Dessa forma serão geradas as funções de transferência de cada parte do controlador.

$$T(z^{-1})w(t) = S(z^{-1})u(t) + R(z^{-1})y(t), \quad (8)$$

onde $w(t)$ é a referência, $u(t)$ é o sinal de controle aplicado na planta e $y(t)$ é a saída da planta.

A alocação de polos é outra forma de projeto de controladores com as mais diversas abordagens. Numa das mais utilizadas são escolhidos os polos em MF e projeta-se o compensador a partir deles e do modelo da planta (CASTRUCCI, BITTAR e SALES, 2011).

Pode-se ainda utilizar diversos outros métodos para projeto de controladores, como o método do Lugar das Raízes, compensadores em avanço e atraso de fase, cada método desse foi largamente estudado pela literatura e está disponível em diversos livros da área de Engenharia de Controle. A escolha do método depende bastante do projetista, das especificações de projeto escolhidas *a priori* e do *hardware* onde ele vai aplicar este controlador. Vale ressaltar também que a aplicação em tempo real é um fator bastante importante nessa escolha.

3 TRABALHOS CORRELATOS

Nesta seção será apresentado o que foi feito na área nos últimos anos. Cabe ressaltar aqui que este tipo de desenvolvimento foi um dos focos das pesquisas da comunidade nos anos de 2005 até 2007, não havendo mais artigos na comunidade sobre isso após esse período até 2017.

Com as mudanças nas regras da RoboCup anunciadas em 2017, este assunto voltou ao centro das atenções, fomentando muitas pesquisas e desenvolvimentos na área. Equipes que nunca haviam se preocupado com o controle de movimentação dos robôs passaram a buscar conhecimento na área de controle de movimentação, mais especificamente controle de motores, para melhorar o desempenho nas competições.

As deficiências que a equipe RoboFEI possui nessa área motivaram este trabalho. Para a escolha do controlador que deverá ser utilizado nos robôs da equipe, foram estudados os trabalhos apresentados por outras equipes da liga SSL bem como alguns trabalhos não relacionados a RoboCup, mas que tratam de controle de motores BLDC.

3.1 TRABALHOS RELACIONADOS NA LIGA SSL

Aqui serão apresentados os trabalhos da comunidade realizados já em 2017 e alguns desdobramentos em 2018 sobre o controle de movimentação dos robôs em outras equipes e comparar os resultados obtidos por estas equipes com o que se espera para a equipe RoboFEI. Com base nesses trabalhos e em seus resultados será escolhido os modelos de controladores que serão testados em simulação para analisar a sua aplicabilidade na equipe.

3.1.1 Warthog Robotics

A equipe Warthog, da Universidade de São Paulo em São Carlos, USP de São Carlos, fez um desenvolvimento em sua eletrônica embarcada onde o ponto principal foi a modificação do PID que era contínuo e foi adaptado para um PID discreto que depois foi aplicado um filtro no termo derivativo (LANG, OLIVEIRA, *et al.*, 2017).

Foi utilizado para o ajuste dos ganhos do controlador mostrado na Equação 9 o Matlab® RLTOOL, semelhante ao SISOTOOL que é uma ferramenta que plota automaticamente o lugar geométrico das raízes (ou *rootlocus*) do sistema sob estudo.

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$

onde,

$$q_0 = K \left(1 + \frac{T_D}{T_0}\right), \quad q_1 = -K \left(1 + 2\frac{T_D}{T_0} - \frac{T_0}{T_i}\right), \quad q_2 = K \left(\frac{T_D}{T_i}\right), \quad (9)$$

onde se tem que K é a constante de proporcionalidade, ou constante de ganho, T_0, T_i e T_D são parâmetros do controlador de proporção, integração e derivação respectivamente.

3.1.2 RoboIME

A equipe RoboIME do IME, Instituto Militar de Engenharia no Rio de Janeiro, fez um grande esforço no sentido de sintonizar novamente o PID de suas rodas para melhorar tanto o deslizamento delas no tapete de competição, como também para melhorar a sua aceleração e desaceleração (CONSENZA, COUTO, *et al.*, 2017).

A equipe utilizou o método de Ziegler-Nichols para ajustar os parâmetros do controlador e paralelamente melhoraram o modelo matemático do robô para obter um resultado melhor do que eles tinham antes.

A equipe utiliza o *software* LabVIEW®, que é bastante flexível e fácil de implementar mudanças significativas na arquitetura e nos parâmetros para controlar todo o seu robô. Segundo a equipe, pensa-se em ajustar parâmetros para três PIDs que possam atuar de forma independente conforme o tipo de manobra o robô esteja realizando a mando da estratégia.

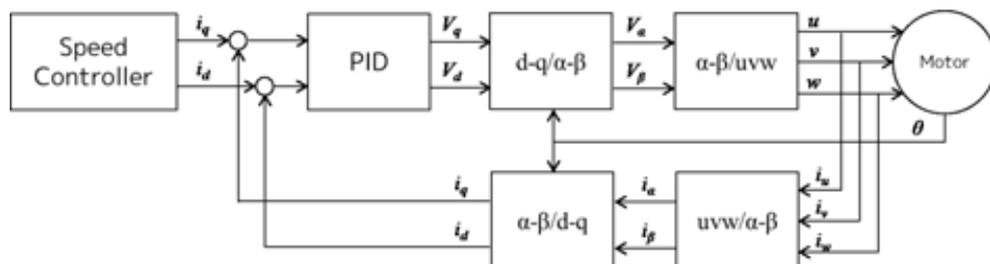
3.1.3 OP-Amp

A equipe OP-Amp da Asagami Works em Osaka – Japão, resolveu inovar o seu controle dos motores das rodas, unindo o seu controlador PID a uma técnica de controle avançada chamada *Current Vector Control*, ou Controle pelo Vetor de

correntes. Esta técnica é largamente usada em motores CA, Corrente Alternada, de alto desempenho onde a corrente do motor é tratada como componentes vetoriais de corrente de torque e corrente de campo (YOSHIMOTO, HORII, *et al.*, 2017).

A Figura 13 adaptada do seu penúltimo TDP, *Team Description Paper*, mostra a forma de aplicar este controlador.

Figura 13 - Diagrama de blocos do Controle por vetor de correntes.



Fonte: Adaptado de (YOSHIMOTO, HORII, *et al.*, 2017)

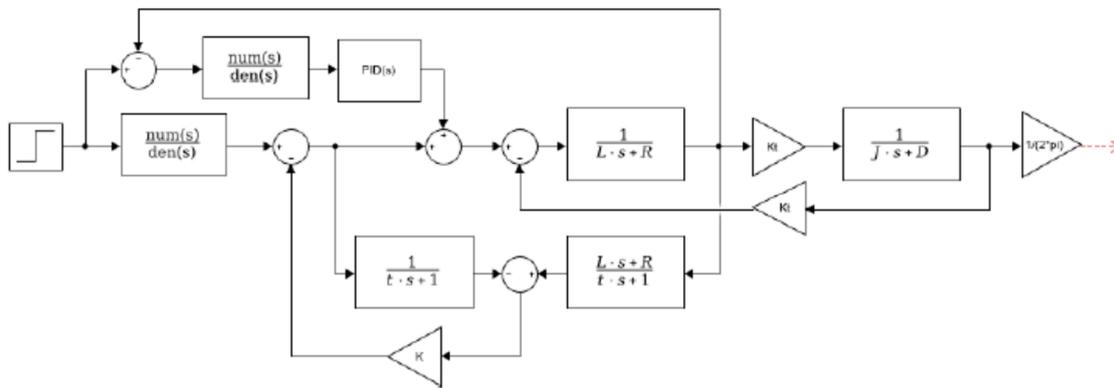
3.1.4 KIKS

A equipe KIKS, do *National Institute of Technology, Toyota College*, em Ashi – Japão, reprojeteu seu robô em 2016 e implementou um controlador simples por *feedback*, ou retroalimentação, como os outros mostrados por outras equipes, que eles julgaram adequado ao seu novo projeto (OHNO, MIMURA, *et al.*, 2017). Esse controlador aumentou muito o escorregamento das rodas no tapete.

Durante a RoboCup daquele ano foi visível para equipe a ineficiência deste controlador, pois devido a um aumento da potência dos motores no robô novo, as rodas começaram a escorregar no tapete de forma que o robô ficava desgovernado em alguns casos.

Para resolver o problema, foi adotado pela equipe uma estratégia de controle *feedforward*, ou alimentação por antecipação, que em teoria pode antecipar o escorregamento através de um observador de estados e diminuir a corrente de torque no motor. O diagrama de blocos do controlador novo da equipe KIKS é mostrado na Figura 14 onde o PID representa o controle por *feedback* para manter o seguimento de trajetória e o ganho K com o estimador de estados como *feedforward* para diminuir a corrente no motor quando houver escorregamento.

Figura 14 - Diagrama de blocos com *feedback* e *feedforward*.



Fonte: Adaptado de (OHNO, MIMURA, *et al.*, 2017).

3.1.5 UBC Thunderbots

A equipe UBC Thunderbots de *The University of British Columbia*, em Vancouver – Canadá, propôs em 2017 uma implementação avançada de um filtro de Kalman para evitar o escorregamento das rodas do seu robô (BAI, GONG, *et al.*, 2017).

A ideia da equipe era trabalhar em três frentes distintas com a finalidade de melhorar a movimentação do seu robô. A primeira frente de trabalho seria determinar matematicamente através da matriz de acoplamento, que representa a cinemática do robô, se estava ocorrendo deslizamento da roda ou não.

A segunda frente de trabalho era mandar as informações do sistema de visão diretamente para o robô para o controle de baixo nível processar essas informações e utilizá-las para evitar o escorregamento das rodas e melhorar o desempenho do robô.

A terceira frente utilizaria a ideia de fusão de sensores para prever com maior certeza a posição do robô no campo entre os *frames* da câmera do sistema de visão, e, com isso, melhorar o seu desempenho e estimar se ocorreu o deslizamento das rodas no tapete e assim evita-lo. Acabaram por tentar implementar o filtro de Kalman para realizar esta terceira frente.

No decorrer da competição de 2017, a equipe percebeu que seu desenvolvimento ainda se encontrava em fase inicial e, portanto, não conseguiria efetivar nenhuma das três frentes de trabalho propostas sem antes melhorar o controlador de movimento do robô que até aquele momento era do tipo *bang-bang* o

que gerava invariavelmente escorregamento no tapete e trancos em sua movimentação.

Esse tipo de controlador, que deriva do clássico controle ON-OFF, liga-desliga, não é o mais indicado para controle de motores pois ele só possui duas posições de acionamento, que equivale a 100% ou a 0%, o que consecutivamente eleva a corrente e dificulta bastante a movimentação em curtas distâncias (MACDOUGALL, ELLIS, *et al.*, 2018).

A solução da equipe para este problema foi implementar um PID que deve fornecer uma resposta suave e precisa para qualquer cenário de movimentação apresentado ao robô, seja para trajetos longos ou curtos.

Até o momento da submissão do documento, eles tinham conseguido chegar a um modelo do seu motor EC-45 flat de 30W da Maxon Motors, similar ao da equipe RoboFEI. A função de transferência proposta pela UBC Thunderbots relaciona a tensão com a velocidade angular do motor e pode ser vista na Equação 10.

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (10)$$

onde $J = 92,5 \text{ gcm}^2$ representa o momento de inercia, $b \cong 0$ representa o atrito viscoso que pode ser desprezado, $L = 0,56 \text{ mH}$ a indutância na armadura do motor, $R = 1,2\Omega$ representa a resistência na armadura e $K = 25,5 \text{ mNm}$ a força eletromotriz constante que relaciona a tensão na armadura e velocidade no eixo, ou seja, o torque do motor.

3.1.6 RoboTeam Twente

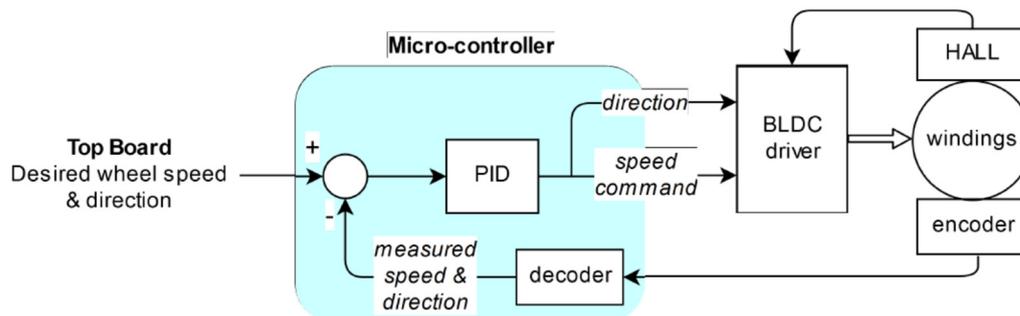
A equipe RoboTeam Twente, um time da *University of Twente*, da Holanda, fez uma mudança bastante semelhante a proposta do novo robô da equipe RoboFEI. A ideia da equipe é retirar o acionamento do motor BLDC de dentro do seu FPGA e utilizar um *driver* específico de acionamento deste tipo de motor (DOORNKAMP, EGDOM, *et al.*, 2018).

A equipe pretende também melhorar bastante a resposta de seu controlador PID com uma melhoria no *encoder* utilizado no motor para medir sua

posição/velocidade. Atualmente a resolução deles é de 6 pulsos por volta e eles pretendem utilizar um novo *encoder* de 1024 pulsos por volta. Isso deve melhorar significativamente a resposta do PID principalmente em trajetos curtos.

A proposta do seu novo controlador pode ser observada na Figura 15, muito similar ao que existe hoje no projeto do novo robô da equipe RoboFEI.

Figura 15 - Diagrama de blocos do controlador PID proposto.



Fonte: Adaptado de (DOORNKAMP, EGDOM, *et al.*, 2018)

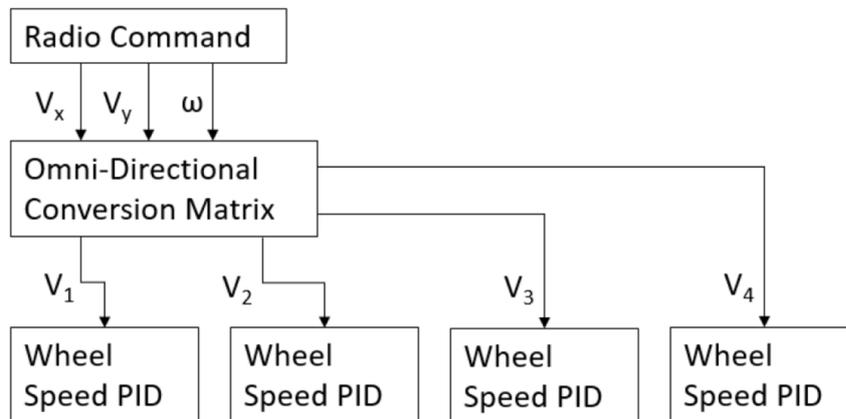
A equipe pretende usar um único controlador PID com cruzamento de zero de forma a garantir a velocidade na roda e o sentido de rotação, pois o novo encoder é incremental e não fornece sentido de rotação do motor, somente uma contagem de pulsos.

3.1.7 RoboJackets

A equipe RoboJackets, de *Georgia Institute of Technology*, na Georgia – Estados Unidos da América, possui uma arquitetura semelhante à da equipe RoboFEI, onde as velocidades enviadas pelo rádio ao robô são desacopladas pela matriz da cinemática inversa e colocadas em um controlador para cada roda, como é mostrado na Figura 16 (ALMAGRO, FELTRACCO, *et al.*, 2018).

Essa arquitetura funciona muito bem para manter a velocidade em x e em y do robô, porém quando se trata da velocidade angular, ela é controlada na equipe RoboFEI pela malha de controle da visão. A equipe RoboJackets pretende realizar este controle totalmente independente do sistema de visão.

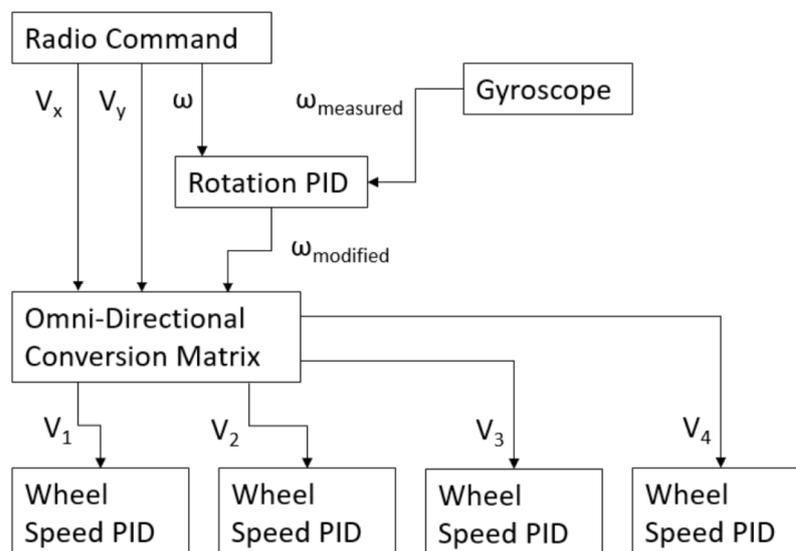
Figura 16 - Esquemático atual da equipe RoboJackets.



Fonte: Adaptado de (ALMAGRO, FELTRACCO, *et al.*, 2018).

A proposta da equipe RoboJackets é passar a utilizar um sensor do tipo giroscópio para melhorar a sua cinemática e incluir um controlador exclusivo para a velocidade angular do robô. O esquemático de como ficará o seu robô é mostrado na Figura 17.

Figura 17 - Novo esquemático da equipe RoboJackets.



Fonte: Adaptado de (ALMAGRO, FELTRACCO, *et al.*, 2018).

Eles pretendem utilizar uma IMU, *Inertial Measurement Unit*, modelo MPU-6050 que contém um giroscópio interno. A ideia do time é no futuro utilizar os outros acelerômetros disponíveis para ajustar melhor os controladores internos do robô.

3.1.8 RoboDragons

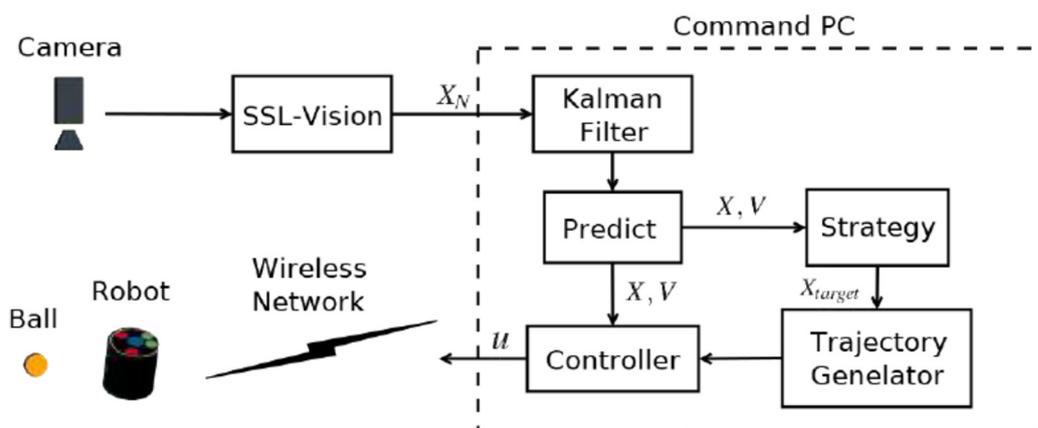
A equipe RoboDragons, da *School of Information Science and Technology*, de Aichi – Japão, é uma equipe mais antiga com um projeto bem mais desenvolvido, mas que ainda assim continua realizando melhorias no controle do seu robô.

Segundo a equipe, somente após possuir um controle de velocidade do tipo PID bem sintonizado eles puderam passar a desenvolver e implementar um controlador muito mais sofisticado para seguimento de trajetória (ITO, KUSAKABE, *et al.*, 2018).

A equipe propõe a construção de um controle MPC, *Model Predictive Control*, ou seja, um controle preditivo baseado em modelo para geração das velocidades que serão enviadas aos robôs de forma a manter uma velocidade mais constante e limitada em sua variação.

Nesse caso, a cinemática direta do robô omnidirecional foi refeita e usada como modelo para o controlador, foram inseridas restrições de mudança de velocidade e performance no robô para limitar o controlador. As suas saídas são, portanto, as velocidades V_x e V_y que serão enviadas para o robô. Um esquemático de funcionamento pode ser visto na Figura 18, onde X_N é o vetor com a informação dos n-ésimos robôs vindos do sistema de visão.

Figura 18 - Esquemático do controle MPC implementado pela equipe.



Fonte: adaptado de (ITO, KUSAKABE, *et al.*, 2018).

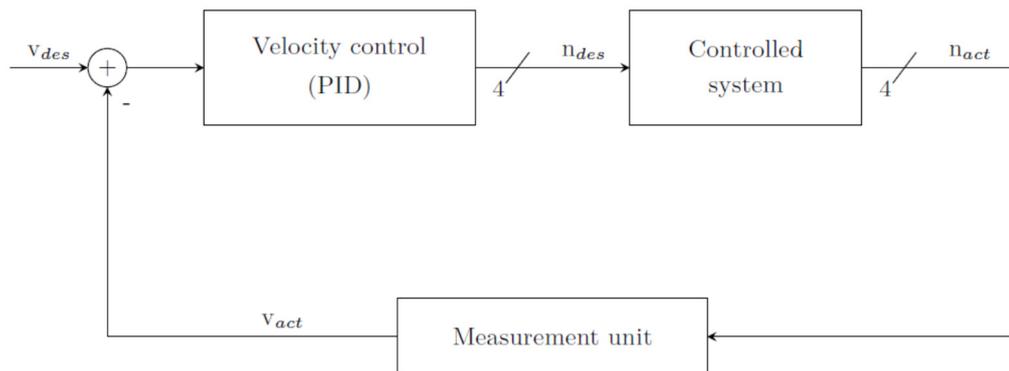
Este controlador de alto nível só é possível pois a equipe possui um controlador PID de baixo nível plenamente funcional operando nos robôs.

3.1.9 ER-Force

A equipe ER-Force da FAU, *Friedrich-Alexander University*, em Erlangen – Alemanha, decidiu reimplementar seu sistema de controle sem levar em conta os desenvolvimentos anteriores (BÜHLMAYER, BURK, *et al.*, 2017). Eles haviam utilizado por muitos anos um sistema bem simples, que pode ser observado na Figura 19, com um PID clássico.

A nova proposta de controlador da equipe é para controlar também a corrente do motor. Este controle será feito com um observador baseado em estados com dois graus de liberdade, onde um deles seria a corrente aplicada no motor e o outro um limitador para evitar picos. A nova arquitetura proposta pode ser vista na Figura 20.

Figura 19 - Diagrama de blocos ER-Force antigo.

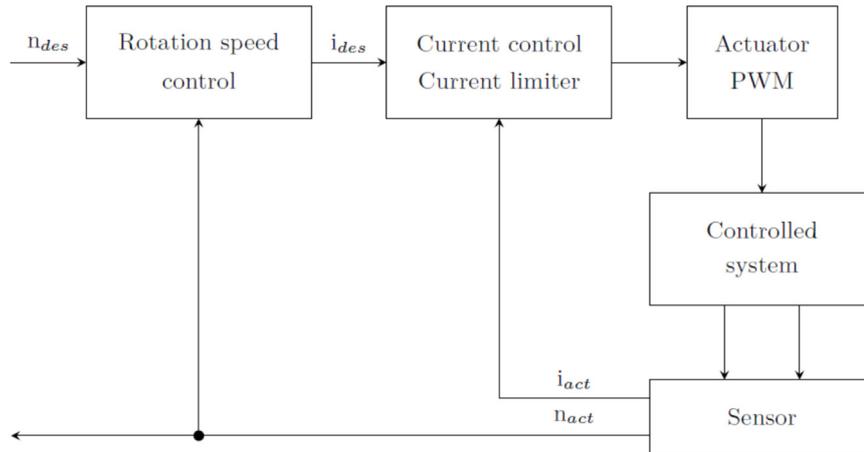


Fonte: Adaptado de (BÜHLMAYER, BURK, *et al.*, 2017).

A equipe ER-Force da FAU, *Friedrich-Alexander University*, de Erlanger – Alemanha, é uma equipe de ponta que pretendia trazer uma arquitetura totalmente reformulada para a RoboCup de 2017 ou no máximo em 2018, pois seus robôs aceleravam e desaceleravam muito mais lentamente do que os outros times, o que limitava a sua capacidade de manobra (LOBMEIER, BURK, *et al.*, 2018).

Como mudanças de arquitetura e de *hardware* acabam sendo lentas e caras, eles ficaram impossibilitados de modificar o seu robô para a RoboCup 2017, o que os motivou a modificar a sua estratégia de controle de baixo nível.

Figura 20 - Diagrama de blocos da nova arquitetura de controle da ER-Force.



Fonte: Adaptado de (BÜHLMAYER, BURK, *et al.*, 2017).

Até este momento era utilizado um simples PI com *feedback* simples para cada motor. Esse controlador, em geral, atende perfeitamente motores *brushless*, mas a sua performance não era satisfatória. A equipe pretende criar estratégias de jogo com manobras mais ousadas de seus robôs, o que leva a necessidade de controladores mais eficientes.

Partindo desse ponto, foi modelado o motor através de suas características comportamentais elétricas e obtiveram o modelo apresentado na Equação (11).

$$\frac{di(t)}{dt} = -\frac{R}{L}i(t) + \frac{1}{L}U(t), \quad (11)$$

com L e R sendo respectivamente a indutância e a resistência interna do motor EC-45 flat da Maxon Motors. A variável i é a corrente e U a tensão, que posteriormente formaram a variável de estado x e a entrada u respectivamente. Isso tudo resultou no modelo em Espaço de Estados, EE, mostrado na Equação (12), sendo y a saída do sistema.

$$\dot{x}(t) = -\frac{R}{L}x(t) + \frac{1}{L}u(t) \quad (12)$$

$$y(t) = x(t)$$

Com essa nova representação do motor é possível aplicar técnicas muito mais avançadas de controle. A equipe implementou um controlador simples em EE, com uma matriz do observador e uma matriz do controlador com ganhos calculados a partir da fórmula de Ackerman (LOBMEIER, BURK, *et al.*, 2018).

Os resultados foram tão surpreendentes para o time que eles seguiram essa vertente para criar e implementar um controlador multivariável em EE discreto e utilizaram esse novo modelo na RoboCup 2018.

3.2 TRABALHOS RECENTES DE CONTROLE DE MOTORES BLDC

Foi realizada uma busca em outras áreas não relacionadas com a robótica móvel omnidirecional, mas que visavam o controle de motores BLDC para outros fins com o objetivo de descobrir o que as outras áreas usam como controladores e qual o desempenho atingido por eles.

O que foi constatado é que a maioria dos trabalhos que utilizam motores *brushless* de ímãs permanentes utilizam um controlador PID clássico.

Primeiramente um dos trabalhos mais citados sobre controles de motores BLDC trata do controle de posição a partir da realimentação de velocidade, neste trabalho são feitas análises completas do motor *brushless* e um projeto completo de um controlador do tipo PI. Este trabalho também trata da leitura de sensores e padronização de valores para serem utilizados no controlador (KANGSANANT e LIM, 1988).

Em outro artigo métodos de controle avançados são utilizados para controlar a posição de forma robusta de motores DC sem escovas (BLDC). Nesse artigo é utilizado um controlador linear quadrático mais o observador de torque na carga para obter um sistema aproximado de um motor BLDC linearizado para um servo AC, usando o método de orientação de campo. Os ganhos são obtidos sistematicamente a partir de uma análise discreta do espaço de estados do sistema. O distúrbio de carga é detectado por um observador da entrada desconhecida e inacessível e é compensado por alimentação sem exigir informações de corrente ruidosa proveniente do motor (KO, LEE, *et al.*, 1993).

Outro trabalho interessante utiliza um arranjo ortogonal para projeto do controlador robusto e realiza simulação numérica por elementos finitos para garantir

a eficiência do controlador frente a projetos com otimização convencional (WANG, LIU, *et al.*, 1999).

Em outro trabalho são abordados problemas bastante específicos de vibração em motores BLDC que podem atrapalhar os sistemas de posicionamento e são geralmente tratados através da análise de Fourier das harmônicas entre as fases desse tipo de motor. No texto os autores apresentam uma alternativa a isso e mostram seu funcionamento com simulação e experimentos reais (PARK, PARK, *et al.*, 2000).

No artigo de HUI, BASU e SUBBIAH, (2003) são apresentadas diversas abordagens de acionamento de motores BLDC utilizando um simples PI para cálculo do sinal de controle.

Existe um trabalho que constata a necessidade primordial de um bom sistema de controle de velocidade das rodas de um robô omnidirecional de quatro rodas tracionadas para manter trajetórias. Nesse trabalho é apontado a necessidade de um controle por realimentação de saída como o PID, além disso ao realizar experimentos com o robô foi constatado também que o escorregamento das rodas em certas superfícies pode ser um desafio adicional a mobilidade do robô (MARIAPPAN, SING, *et al.*, 2014).

Um trabalho sobre monociclo trata do controle através de um PID do motor BLDC que aciona o monociclo para equilíbrio do equipamento que se assemelha muito a um pêndulo invertido que deve operar em um ângulo de $\pm 10^\circ$ acionado/estabilizado pelo motor BLDC controlado por um PID (RIATTAMA, BINUGROHO, *et al.*, 2016).

Um exemplo é o trabalho de Fu-cheng e Xue-wei de (2016) que trata de seccionadoras de alta tensão acionadas por motores BLDC. O controlador adotado para garantir a performance da seccionadora foi um PID clássico.

Outro trabalho mais recente que utiliza controla avançado para motores BLDC é o de VINIDA e CHACKO, (2016) que utiliza um controlador H infinito para melhorar a manobrabilidade em submarinos acionados por motores BLDC.

Em um trabalho um pouco mais antigo foi encontrado um modelo do motor que foi apresentado como sendo completo e suficiente que fornece uma função de transferência em malha aberta do BLDC da Maxon usando todos os parâmetros necessários e suficientes disponíveis (PATEL e PANDEY, 2013). Esse modelo é um desenvolvimento do modelo proposto por Baldrusson (2005) e vai ser usado nos desenvolvimentos deste trabalho.

Outros trabalhos poderiam ser apresentados aqui, porém a maioria deles concluem que o melhor controlador para o seu problema é o PID clássico. Alguns autores abordam outros algoritmos de controle obtendo resultados satisfatórios.

4 PROJETO E SIMULAÇÃO

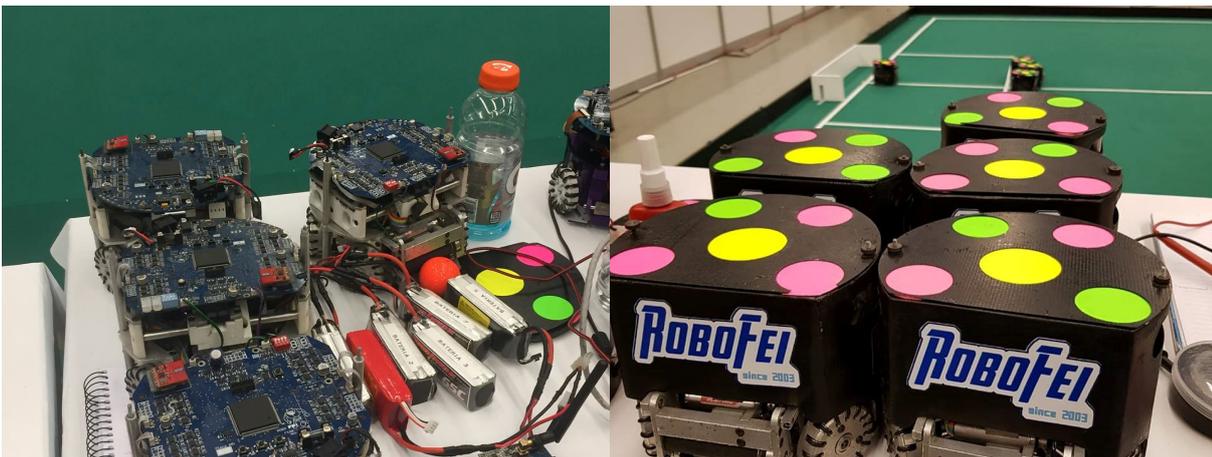
Nessa seção serão apresentados alguns desenvolvimentos necessários para a conclusão deste trabalho. Também serão levantados dados sobre o robô omnidirecional da equipe RoboFEI onde será realizado o estudo de caso de forma a atingir o objetivo proposto.

4.1 O ROBÔ

Recentemente diversas motivações levaram a equipe de robótica da categoria SSL do Centro Universitário da FEI, Equipe RoboFEI, a um processo de reestruturação profunda em seus métodos e processos. Levando a equipe a um projeto de robô novo e a reconstrução a partir do zero do *software* da estratégia da equipe.

Nesse contexto, o presente trabalho criará uma base para o desenvolvimento sólido e para perpetuação dos trabalhos envolvendo as estruturas de controle do robô.

Figura 21 - a) Robôs em preparação para o jogo. b) Robôs prontos para entrar em campo para uma partida.



a)

b)

Fonte: Autor

A equipe RoboFEI possui atualmente 8 robôs com duas arquiteturas eletrônicas distintas, cada qual com as suas vantagens e desvantagens. A Figura 21 mostra alguns robôs da equipe RoboFEI em preparação para um jogo na RoboCup 2018, a),

e prontos para entrar em campo, b). Até o presente momento nenhum deles possui nenhum tipo de controle relacionado a sua movimentação a não ser através do sistema de visão e de programação para resolução imediata de problemas, feito de forma improvisada sem possibilidade de replicação ou generalização para todos os robôs.

Para resolver este tipo de problema deve-se conhecer um pouco mais da arquitetura dos robôs da equipe. A arquitetura interna dos robôs da equipe RoboFEI utiliza um FPGA da XILINX cujas informações estão na Tabela 2, que controla através da leitura dos sensores de efeito *hall* dos motores a próxima sequência de sinais PWM que será enviada aos motores através de um circuito de acionamento.

Tabela 2 - Informações sobre o FPGA da equipe RoboFEI.

Característica	Modelos/Especificações
Modelo	Xilinx Spartan 3 XC3S400 (XILINX)
Memória PROM	Xilinx XCF04S
Memória SRAM	AT25DQ321 (ADESTO TECHNOLOGIES)
Processador	Xilinx MicroBlaze
Microcontrolador	RISC, 32 bits e clock 50Mhz

Fonte: Autor.

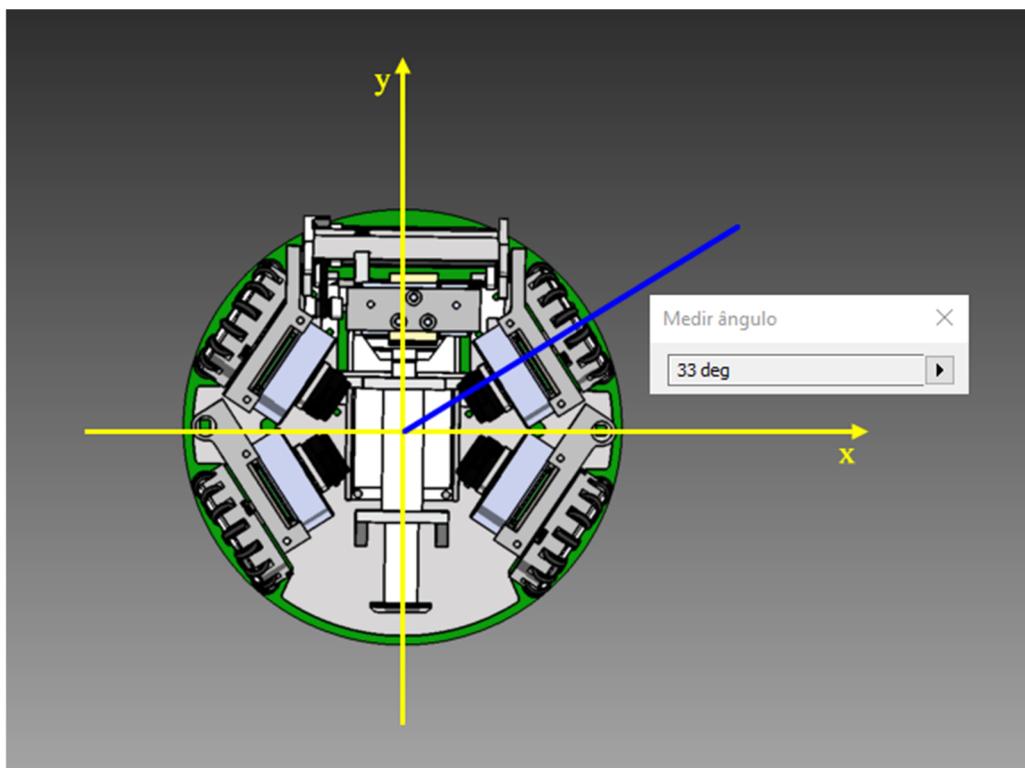
O circuito de acionamento tem como característica principal acionar, através do *driver* TC4427 (MICROCHIP TECHNOLOGY), um grupo formado por três MOSFET's modelo IRF7389 (INFINEON TECHNOLOGIES) em meia-ponte H, que farão o controle do acionamento de cada enrolamento do motor em cada uma das suas três fases.

Para melhorar o que a equipe possui, alguns posicionamentos foram necessários. Entre os mais importantes está a separação da aceleração e desaceleração no robô, que antes era feita em uma única equação utilizando séries de Taylor, mas devido à alta dinâmica do jogo, com mudança constante de cenário e objetivo, esses parâmetros de aceleração e desaceleração não eram realizados por completo em um ambiente de jogo. Quando a movimentação era feita com o jogo parado a solução antiga era bastante eficiente, mas durante a partida a informação de desaceleração era sempre calculada e logo em seguida descartada devido a mudança dos objetivos de posicionamento.

Inicialmente foi feita uma parametrização nas velocidades dos motores, fazendo-os trabalharem e serem medidos em uma grandeza equivalente, no caso de -100% até +100% para ambos os casos. Essa escolha foi feita para facilitar a comunicação entre o *software* da estratégia no computador central e o *firmware* interno de cada robô, deixando assim todas as partes em um mesmo patamar de comunicação. Isso ainda levou a equipe a um entendimento de que separar o controle de movimentação sendo parte realizado no *hardware*, mais especificamente no *firmware* do robô e parte no *software* de estratégia.

Essa abordagem de controle distribuído é usada em outras situações em que parte de uma estratégia de controle é feita fora da máquina, em uma central de processamento do sistema por exemplo, e outra bem mais específica é feita na própria máquina. As máquinas autônomas integradas a centrais de processamento contam com a presença de diversos processadores em que todas as tarefas/trabalhos são alocadas automaticamente pelo sistema (ROSÁRIO, 2005).

Figura 22 - Vista superior do robô com o plano de referência.



Fonte: O autor.

Seguindo esse pensamento foi colocado parte do controle de movimentação no *software* geral de estratégia que roda em um computador externo, e parte no *firmware* interno do robô. O *software* manda as velocidades que o robô deve estar a cada 100ms e o controlador do *firmware* se responsabiliza por manter o mais precisamente possível essas velocidades.

Dessa maneira é possível melhorar a forma como os robôs se locomovem em ambas as arquiteturas eletrônicas existentes. Como uma alternativa de melhoria, foi modificado o protocolo de comunicação entre o *software* de estratégia e os robôs, de maneira que cada robô receba uma velocidade relativa ao seu eixo x, V_x , uma velocidade relativa ao seu eixo y, V_y , e uma velocidade angular $\dot{\omega}$, sendo as velocidades medidas em m/s e a velocidade angular em rad/s.

Essas velocidades são convertidas no *software* da estratégia para os valores parametrizados de -100% até +100% e depois enviado ao robô, resolvendo assim outro problema relativo ao protocolo de comunicação que disponibiliza somente um *byte* para a transmissão dos parâmetros V_x , V_y e $\dot{\omega}$.

Dentro do *firmware*, o robô calcula a cinemática direta das rodas e manda a referência para os motores, cada qual passando por um controlador que é o objeto de estudo deste trabalho. A Figura 22 mostra os eixos e a sua relação com a angulação das rodas do robô.

4.1.1 Cinemática Do Robô

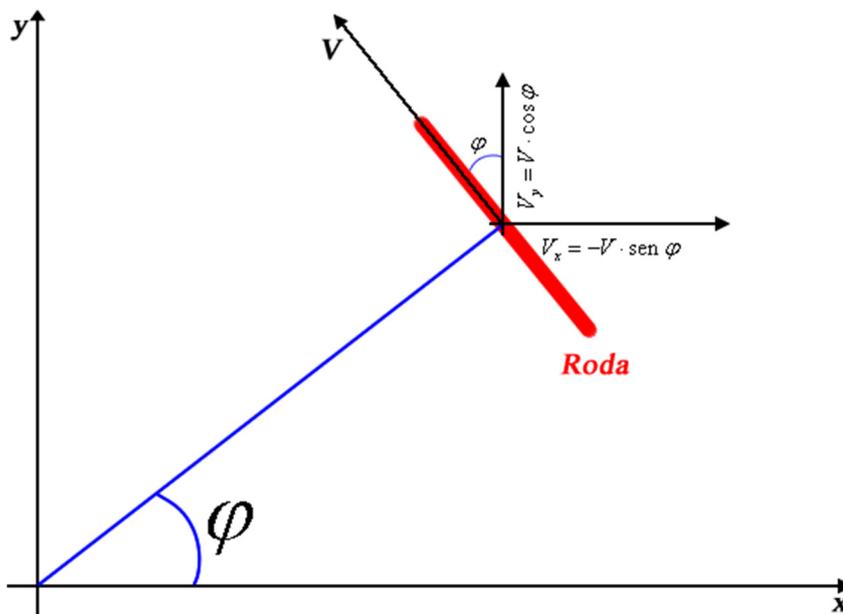
Para o cálculo cinemático do robô atual da equipe RoboFEI, foi considerada a ordem usada pela equipe RoboFEI em que a roda número 1 é a superior esquerda da Figura 22 e as outras são sequenciais no sentido anti-horário.

A cinemática pode ser feita baseada no exemplo da roda 4 mostrada na Figura 23, para facilitar a visualização com o plano cartesiano alinhado a esquerda, em que a velocidade da roda V é decomposta em V_x e V_y . De forma análoga é calculada para as outras rodas do robô supondo que todas giram no sentido horário como positivo.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -\text{sen}(\varphi) & -\text{cos}(\varphi) & 1 \\ \text{sen}(\varphi) & -\text{cos}(\varphi) & 1 \\ \text{sen}(\varphi) & \text{cos}(\varphi) & 1 \\ -\text{sen}(\varphi) & \text{cos}(\varphi) & 1 \end{bmatrix} \cdot \begin{bmatrix} V_x \\ V_y \\ \dot{\omega} \end{bmatrix} \quad (13)$$

Na Equação (13), o primeiro vetor, $\text{motores} = [v_1 \ v_2 \ v_3 \ v_4]^T$ representa a velocidade em cada uma das rodas. A matriz com os senos e cossenos é chamada de matriz de acoplamento, doravante denominada matriz C , que converte propriamente as velocidades em x e em y para as velocidades de cada roda. O último vetor representa as informações mandadas ao robô no *byte* específico do protocolo de comunicação, $V = [V_x \ V_y \ \dot{\omega}]^T$.

Figura 23 - Cinemática euclidiana das rodas.



Fonte: o autor.

Se fosse necessário obter as velocidades absolutas do robô em função das velocidades das rodas seria de simples resolução se a matriz C de acoplamento fosse quadrada. Assim seria possível calcular a inversa, C^{-1} e, quando multiplicada pela esquerda dos dois lados da equação resultaria na expressão apresentada na Equação (14).

$$V = C^{-1} \cdot \text{motores} \quad (14)$$

Como a matriz de acoplamento não é quadrada deve-se utilizar a sua pseudoinversa, denotada por C^+ . A matriz pseudoinversa é uma equivalente a matriz inversa que atenda as quatro condições de Moore-Penrose que formalizou essa definição (BOOS, 2015). A matriz C^+ corresponde a uma matriz que multiplicada pela matriz original gera uma matriz identidade quadrada com dimensão $m \times m$, pois sendo $C_{m \times n}$ e $C_{n \times m}^+$ isso implica que $C^+C = CC^+ = I_{m \times m}$.

Segundo Rojas, (2006), a matriz de desacoplamento, ou a pseudoinversa de C adaptada para o robô da equipe RoboFEI é mostrada na Equação (15).

$$C^+ = \frac{1}{4} \begin{bmatrix} \frac{-1}{\text{sen}(\varphi)} & \frac{1}{\text{sen}(\varphi)} & \frac{1}{\text{sen}(\varphi)} & \frac{-1}{\text{sen}(\varphi)} \\ \frac{-1}{\text{cos}(\varphi)} & \frac{-1}{\text{cos}(\varphi)} & \frac{1}{\text{cos}(\varphi)} & \frac{1}{\text{cos}(\varphi)} \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (15)$$

Dessa forma, podem ser confirmado os valores de velocidade realizando a multiplicação matricial mostrada na Equação (16). O *firmware* atual do robô utiliza a Equação (13) para a decomposição inicial das velocidades. Feito isso, uma função converte e escreve estes valores nos motores.

$$\begin{bmatrix} V_x \\ V_y \\ \dot{\theta} \end{bmatrix} = C^+ \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (16)$$

4.1.2 A Dinâmica do Robô

Existem diversas abordagens para a modelagem dinâmica de robôs móveis por rodas. Cada qual traz características próprias que podem ser mais interessantes conforme a necessidade.

A maioria dos trabalhos relaciona somente os modelos cinemáticos tendo em vista que as velocidades são baixas e o atrito pode ser desprezado (DE SÁ, 2016).

Os modelos cinemáticos descrevem o robô em função da velocidade e orientação das rodas, enquanto os modelos dinâmicos descrevem o robô em função das forças generalizadas aplicadas pelos atuadores, por exemplo, torques nas rodas (LAGES, 2013).

O método de Newton-Euler usa o equilíbrio de forças e momentos para derivar as equações de movimento. Isso significa que os balanços de força e momento devem ser considerados das três dimensões (WATANABE, SATO, *et al.*, 2000)

Ainda pode-se estender o que foi apresentado por WATANABE, (1998), onde o método de Newton-Euler é aplicado utilizando explicitamente não linearidades relativas as rodas que inclui também o atrito de Coulomb e viscoso ao mesmo tempo em sua formulação (REN e MA, 2013).

Para este trabalho foram feitas buscas em aplicações semelhantes dentro da própria liga SSL e percebeu-se que dois trabalhos se destacam, o clássico trabalho do Rojas e do Förster (2006) que inspirou muitos desenvolvimentos na liga com abordagens cinemáticas e dinâmicas e, mais recentemente, o trabalho da equipe ZJUNlict que figura entre os principais vencedores da liga e teve o desenvolvimento de uma análise dinâmica do robô para melhorar o seu desempenho e precisão de deslocamento (LI, XIONG, *et al.*, 2015).

De forma ilustrativa o desenvolvimento que será mostrado a seguir será baseado no fato de o robô possuir um sistema referencial próprio em que o deslocamento para frente do robô equivale ao seu deslocamento em y positivo e zero em x .

Assumindo que o centro de massa do robô esteja no seu centro geométrico e que sua massa $M = 3,1\text{ kg}$ e seu momento de inércia $I = 0,0126\text{ kg}\cdot\text{m}^2$ e o seu raio seja $R = 0,09\text{ m}$ a aceleração do centro de massa pode ser calculada como na Equação (17).

$$a = \frac{1}{M}(F_1 + F_2 + F_3 + F_4), \quad (17)$$

onde F_i são as forças de tração mostrada na figura 2 e $i = 1, \dots, 4$. A aceleração angular do robô pode ser calculada como mostrado na Equação (18).

$$\dot{\omega} = \frac{I}{R}(f_1 + f_2 + f_3 + f_4), \quad (18)$$

onde f_i denota a magnitude da força F_i e $i = 1, \dots, 4$. Esse cálculo é possível porque as rodas exercem as forças de forma tangente ao cilindro e apontam para o mesmo sentido o que permite que seja trabalhado somente com a magnitude das forças e com o seu sentido (ROJAS e FÖRSTER, 2006).

A partir da Segunda Lei de Newton mostrada na Equação (19) é possível decompor geometricamente as forças para se ter as acelerações em x e em y .

$$F = m.a \quad (19)$$

Levando em conta o problema geométrico mostrado na Figura 17 é possível derivar as equações mostradas na Equação (20) que relacionam a Equação (17) com a Equação (19).

$$\begin{aligned} M.a_x &= -f_1.\text{sen}(\varphi) - f_2.\text{sen}(\varphi) + f_3.\text{sen}(\varphi) + f_4.\text{sen}(\varphi) \\ M.a_y &= f_1.\text{cos}(\varphi) - f_2.\text{cos}(\varphi) - f_3.\text{cos}(\varphi) + f_4.\text{cos}(\varphi) \end{aligned} \quad (20)$$

Reescrevendo essas equações juntamente com a Equação (18) de forma matricial tem-se o mostrado na Equação (21).

$$\begin{bmatrix} a_x \\ a_y \\ \dot{\omega} \end{bmatrix} = \frac{1}{M} \cdot \begin{bmatrix} -\text{sen}(\varphi) & -\text{sen}(\varphi) & \text{sen}(\varphi) & \text{sen}(\varphi) \\ \text{cos}(\varphi) & -\text{cos}(\varphi) & -\text{cos}(\varphi) & \text{cos}(\varphi) \\ \frac{M.R}{I} & \frac{M.R}{I} & \frac{M.R}{I} & \frac{M.R}{I} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (21)$$

Ainda se pode realizar mais um desenvolvimento aqui em relação ao momento de inércia. Para um cilindro uniforme o momento de inércia pode ser calculado como $I = \frac{1}{2}.M.R^2$ e para um anel sem massa interna, um cilindro oco, o momento de inércia pode ser calculado como $I = M.R^2$, portanto não é absurdo assumir que $I = \alpha.M.R^2$ com $\frac{1}{2} \leq \alpha \leq 1$. A massa do robô é distribuída em seu corpo de forma simétrica, mas não uniforme, o que faz com que seu momento de inércia seja uma combinação do

momento de inércia de um cilindro sólido com massa uniforme e um anel cilíndrico com massa uniforme.

Dessa forma, com algumas manipulações matemáticas tem-se a equação final mostrada na Equação (22) onde a matriz 3 x 4 mostrada é chamada de Matriz de acoplamento de forças C_α (ROJAS e FÖRSTER, 2006).

$$\begin{bmatrix} a_x \\ a_y \\ R.\dot{\omega} \end{bmatrix} = \frac{1}{M} \cdot \begin{bmatrix} -\text{sen}(\varphi) & -\text{sen}(\varphi) & \text{sen}(\varphi) & \text{sen}(\varphi) \\ \cos(\varphi) & -\cos(\varphi) & -\cos(\varphi) & \cos(\varphi) \\ \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (22)$$

Esse desenvolvimento foi feito para fornecer uma base sólida aos desenvolvimentos futuros que serão propostos ao final deste trabalho.

4.1.3 Modelos do motor BLDC

Como foi mostrado anteriormente, existem diversas abordagens para a modelagem de motores BLDC. Aqui se espera que o modelo escolhido seja o mais abrangente possível, onde existem diversas informações sobre como ele é obtido e sendo esse publicado e aceito na comunidade científica, para tanto optou-se pelo modelo proposto por Patel e Pandey (2013).

O motor BLDC que será usado neste trabalho é o EC-45 *flat* 50W da Maxon Motors. O seu número de série é 251601 e seus dados retirados do catálogo do fabricante podem ser vistos na Tabela 3.

Essas são todas as informações fornecidas pelo fabricante sobre esse motor e elas são necessárias e suficientes para se obter um modelo que relacione a entrada em tensão com uma saída em velocidade angular (PATEL e PANDEY, 2013).

$$G(s) = \frac{\omega}{V} \quad (23)$$

Dessa forma, a função de transferência do motor relaciona a entrada em tensão, V , com a saída em velocidade angular, ω , (SINGH, KATAL e MODANI, 2014) e pode ser vista na Equação (23).

Tabela 3 - Dados do Motor EC-45 flat 50W.

Item	Descrição na Tensão nominal	Unidade	Valor
1	Tensão nominal	V	24,02
2	Velocidade sem carga	rpm	6700
3	Corrente sem carga	mA	201
4	Velocidade nominal	rpm	5260
5	Torque nominal	mNm	84,3
6	Corrente nominal	A	2,36
7	Torque de partida	MNm	822
8	Corrente de partida	A	24,5
9	Eficiência máxima	%	83
10	Resistência final fase a fase	Ω	0,978
11	Indutância final fase a fase	mH	0,573
12	Constante de torque	mNm/A	33,5
13	Constante de velocidade	rpm/V	285
14	Gradiente velocidade/torque	rpm/mNm	8,32
15	Constante mecânica de tempo	ms	11,8
16	Inércia do Rotor	gcm ²	135
17	Número de fases	-	3

Fonte: O autor.

A função de transferência $G(s)$ pode ser vista na Equação (24) onde s é a variável complexa de Laplace.

$$G(s) = \frac{1}{K_e} \frac{1}{\tau_m \cdot \tau_e \cdot s^2 + \tau_m \cdot s + 1} \quad (24)$$

Nessa equação K_e representa o torque elétrico em $\frac{V \cdot s}{rad}$, que τ_m e τ_e representam a constante mecânica e a constante elétrica respectivamente. O torque elétrico tem uma relação com o torque mecânico, K_t , que pode ser vista na Equação (25).

$$K_e = K_t \cdot 0,0605 \quad (25)$$

Essa equação relaciona a potência elétrica com a potência mecânica. As constantes elétrica e mecânica podem ser calculadas conforme mostrado na Equação (26).

$$\begin{aligned}\tau_e &= \frac{L}{3.R} \\ \tau_m &= \frac{I.3.R}{K_e.K_t}\end{aligned}\quad (26)$$

Onde L é a indutância, item 11 na Tabela 3; R é a resistência, item 10 na Tabela3; I é a inercia do rotor, item 16 na Tabela 3 e K_e é obtido como mostrado anteriormente. Diretamente da tabela tem-se K_t , a constante de torque, que é o item 12 da Tabela 3 e τ_m que é o item 15 da Tabela 3, a constante mecânica de tempo.

Todos os dados devem ser colocados em unidades do SI, Sistema Internacional de Unidade, tratado cujo Brasil é signatário e cuida das unidades de medida. Com alguma manipulação matemática é possível obter a função de transferência mostrada na Equação (27).

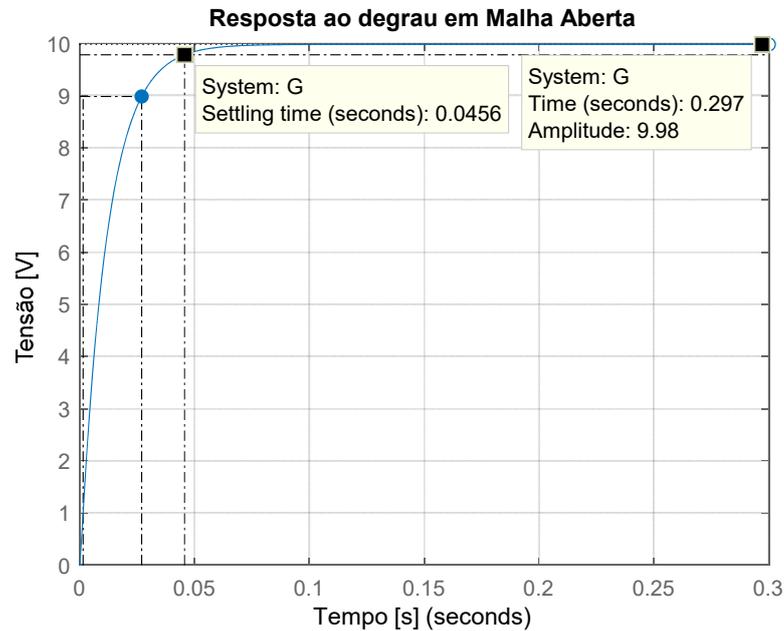
$$G(s) = \frac{9,98}{2,304.10^{-6}.s^2 + 0,0118.s + 1}\quad (27)$$

Os valores mostrados em algumas figuras são gerados diretamente pelo MATLAB® e estão em notação de engenharia de acordo com o SI, onde, por exemplo, $e-6 = 10^{-6}$ padrão também em que o MATLAB® opera.

Para construção do modelo foi desenvolvido um código em MATLAB® chamado Constantes.m onde deve-se inserir os dados necessários e suficientes para a geração do modelo em unidades do SI e dessa forma pode-se então modificar o motor utilizado como exemplo para qualquer outro motor BLDC que possua essas informações. Este código pode ser visto no Anexo I.

Para análise da função de transferência foi desenvolvido um código em MATLAB® que gera os gráficos do sistema em Malha Aberta, MA, para resposta ao degrau e outras observações também disponível com o nome de Modelo_MA.m no Anexo I.

Figura 24 - Resposta do modelo em MA.



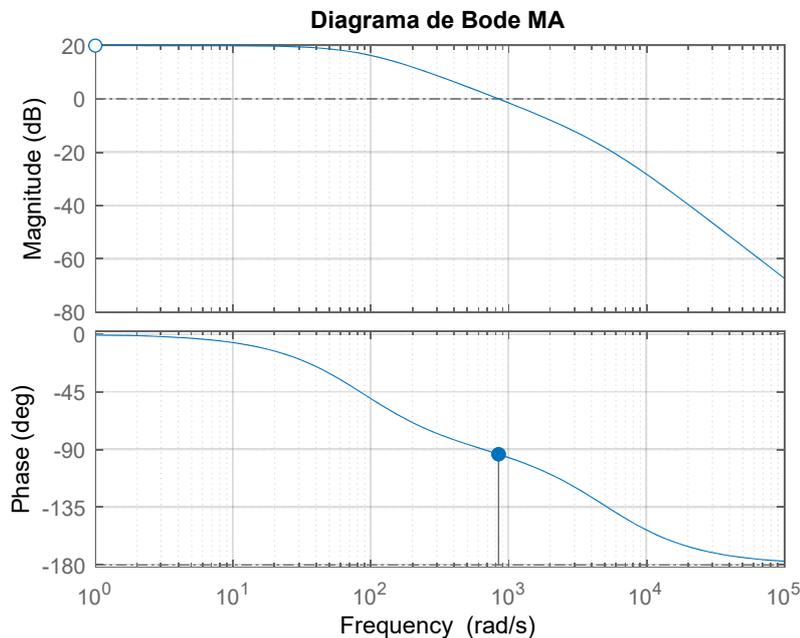
Fonte: Autor.

A Figura 24 mostra a resposta do sistema em Malha Aberta, dela retira-se algumas informações sobre o sistema como o valor final do sistema em estado estacionário $ess = 8,98$. Também é possível obter o tempo de subida e o tempo de assentamento, sendo o $Tr = 0,0255s$ e o $Ts = 0,0456s$ respectivamente. Esses valores são obtidos diretamente da figura gerada pelo MATLAB® no próprio *software*.

Os polos de Malha Aberta do modelo são reais, portanto, o sistema não deve apresentar oscilação na saída e nem tampouco sobressinal. E como eles são negativos, $p_1 = -5,0353 \cdot 10^3$ e $p_2 = -0,0862 \cdot 10^3$, e reais, o sistema não apresenta oscilação na saída e é estável. Como um dos polos é muito menor do que outro, ele para de influenciar a resposta do sistema muito mais rapidamente do que o outro, o que faz com que o comportamento do sistema se pareça muito com um sistema de primeira ordem mesmo ele sendo de segunda ordem.

A Figura 25 mostra o diagrama de bode do sistema, nele é possível observar que o sistema é claramente de segunda ordem. Pelo mesmo diagrama, em malha fechada, será estável por causa das margens de fase e ganho serem negativas.

Figura 25 - Diagrama de Bode.



Fonte: Autor.

Com essa análise preliminar fica demonstrado o comportamento do modelo, como era de se esperar, muito parecido com um motor de corrente contínua comum. Esse modelo foi gerado com os dados suficientes e necessários para se obter o comportamento de um motor BLDC da Maxon (PATEL e PANDEY, 2013).

4.2 SINTONIA DOS CONTROLADORES

Para a sintonia dos controladores, foram utilizados os métodos básicos de sintonia recomendados (ÅSTRÖM e HÄGGLUNG, 1995). Nesses métodos o comportamento dos parâmetros é importante para a sintonia dos controladores, onde ela nem sempre é completamente matemática e pode ser alcançada de forma empírica com resultados bastante satisfatórios.

Inicialmente vale comentar que o PID, (Proporcional + Integral + Derivativo), é o algoritmo de controle mais comum e mais útil de todos na engenharia de controle de sistemas (ÅSTRÖM e HÄGGLUNG, 1995). E que este controlador pode ser escrito de diversas formas, desde que mantenha o seu conceito básico apresentado na seção 2.3.1.

A equação diferencial apresentada na Equação (4) pode ser escrita no domínio da frequência através da transformada de Laplace. E ainda pode ser rearranjada conforme a arquitetura do modelo. A Equação (28) mostra isso.

$$K_p + \frac{K_i}{s} + K_d \cdot s = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right), \quad (28)$$

onde K_p é o ganho proporcional, $T_i = \frac{K_p}{K_i}$ é o tempo integral e $T_d = \frac{K_d}{K_p}$ é o tempo derivativo. A mudança em cada um desses parâmetros em um caso típico altera parte da resposta do sistema, seja ela no regime transiente ou no regime permanente. Os principais fatores observados são:

- Tempo de subida, ou tempo de resposta – T_r ;
- Máximo sobressinal percentual – M_p ;
- Tempo de assentamento – T_s ;
- Erro em regime permanente, ou estacionário – Ess .

E seu comportamento segundo Åström e Hägglung, (1995), está descrito na Tabela 4.

Tabela 4 - Comportamento do sistema em função dos parâmetros do PID.

PARÂMETRO	T_r	M_p	T_s	Ess
Kp	diminui	aumenta	altera pouco	diminui
Ki	diminui	aumenta	aumenta	elimina
Kd	altera pouco	diminui	diminui	altera pouco

Fonte: Autor.

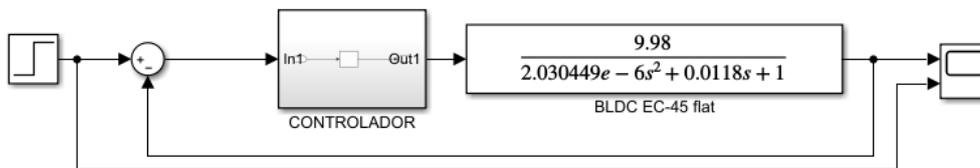
Ainda se deve levar em consideração alguns saberes que são recomendados como um possível caminho de sintonia por professores da Universidade de Michigan, Carnegie-Mellon (MESSNER e TILBURY, 1997), que são mostrados a seguir:

1. Obtenha a resposta do sistema em Malha Aberta e determine o que deve ser melhorado;
2. Adicione um ganho proporcional para melhorar o tempo de subida;
3. Adicione um ganho integral para diminuir o erro em regime permanente;
4. Adicione um ganho derivativo para diminuir o sobressinal;
5. Ajuste esses ganhos para obter a resposta desejada.

Por fim, nem sempre é necessário a utilização do controlador completo para o desempenho esperado, normalmente em sistemas mais simples um PI é o suficiente. Mantenha o controlador o mais simples possível (MESSNER e TILBURY, 1997).

Para a sintonia dos controladores foi desenvolvido um diagrama de blocos básico no *Simulink*, ferramenta de simulação poderosa do MATLAB®, mostrado na Figura 26.

Figura 26 - representação do diagrama de blocos padrão.



Fonte: Autor.

Nesse diagrama serão testados os controladores propostos e será medida a sua resposta nos critérios escolhidos para determinar qual seria a melhor escolha e por quê.

4.2.1 PID contínuo

Para sintonia e simulação desse controlador foi feito um programa em MATLAB® que emula o diagrama de blocos apresentado na Figura 26. Esse código está disponível no Anexo I. Esse programa é o MF_PIDcontínuo.m.

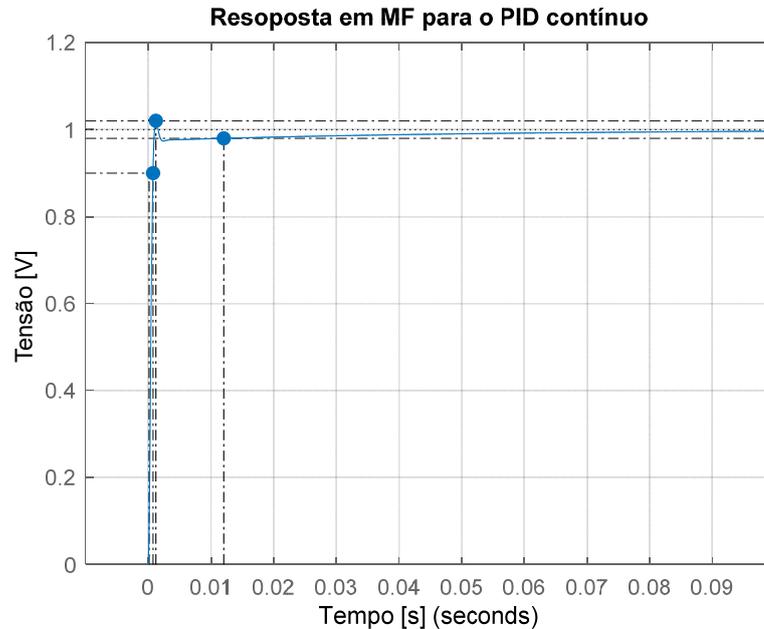
Utilizando os critérios apresentados anteriormente para sintonia empírica de um controlador PID, chegamos a resposta mostrada na Figura 27.

Com esta simulação se obteve um tempo de subida $Tr = 0,00062s$ e o tempo de assentamento foi de $Ts = 0,0125s$ e com sobressinal $Mp = 2,06\%$. Para realizar outros testes foi mudado o degrau para um degrau negativo e resultou em erro nulo em regime permanente.

$$C(s) = \frac{3.s + 60}{s} \quad (29)$$

A equação final pode ser vista na Equação (29) e mostra um PI que é o suficiente para o desempenho desejado.

Figura 27 - Resposta do sistema em MF.



Fonte: Autor.

4.2.2 PID discreto

Para fazer um projeto discreto de controladores primeiramente deve-se encontrar um período de amostragem coerente com o sistema sob estudo. Para tanto deve-se observar o critério de Nyquist, que diz que a frequência de amostragem deve ser pelo menos o dobro da máxima frequência do sinal analógico original (OGATA, 2000). Ou ainda frequência natural amortecida em malha fechada deve ser pelo menos duas vezes menor que a frequência de amostragem (OGATA, 1995)

Isso só é possível com sinais periódicos, se não se sabe qual a maior frequência a que o sistema responde fica difícil encontrar essa frequência de amostragem. Alguns autores utilizam a largura de banda para encontrar esse período de amostragem, outros se baseiam na constante de tempo do sistema (ÅSTRÖM e WITTENMARK, 1990). Neste último caso, encontra-se o valor da resposta do sistema ao degrau aplicado em Malha Aberta onde mede-se o tempo de início da resposta do

sistema até a saída atingir 63% do valor final, como no método de o Ziegler-Nichols em malha aberta (FRANKLIN, POWELL e WORKMAN, 1998). O período de amostragem deve poder captar essa dinâmica do sistema, o que é razoável dizer que ele deve ser 10 vezes menor que este tempo pelo menos.

Nesse caso, a constante de tempo do sistema estimado é de 0,012s o que nos leva a um período de amostragem de no mínimo 0,001s.

Para a sintonia do controlador discreto foi primeiramente discretizada através de um segurador de ordem zero na entrada do sistema, para poder se utilizar da ferramenta RLTOOL do MATLAB®. Ao fazer isso se obteve a função de transferência discreta apresentada na Equação (30) com período de amostragem $\Delta t = 0,001s$.

$$G(z) = \frac{0,6658.z + 0,153}{z^2 - 0,9239.z + 0,005974} \quad (30)$$

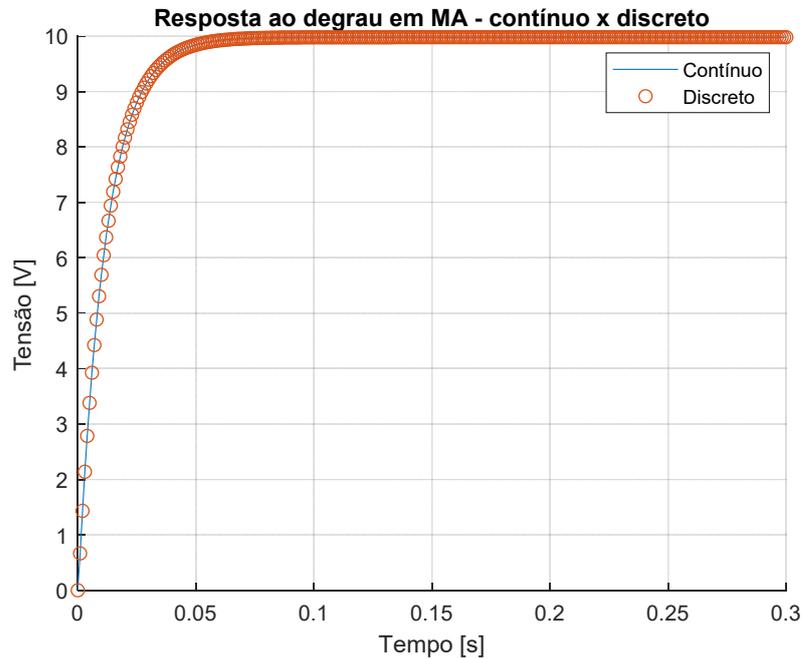
onde z é a variável de tempo discreto.

A Figura 28 mostra um comparativo entre o sistema contínuo e o sistema discreto, ambos em Malha Aberta respondendo ao degrau unitário, o que valida o modelo mostrado na Equação (30). O código em MATLAB que gerou este modelo e a Figura 28 está disponível no Anexo I.

O melhor PID discreto que o RLTOOL, ferramenta poderosa existente no MATLAB® para sintonia de compensadores, sintonizou automaticamente não satisfaz todas as especificações pensadas *a priori*. Para tanto foi selecionado o PID clássico para ser sintonizado automaticamente através das fórmulas tradicionais um compensador do tipo PID pela resposta ao degrau. Foram testados diversos métodos de sintonia disponível na ferramenta e nenhum conseguiu todos os objetivos, portanto os objetivos foram relaxados para se obter um controlador com bom desempenho que melhorasse o sistema.

Depois do relaxamento inicial, o controlador sintonizado automaticamente pelo MATLAB® ficou satisfatório. A resposta do sistema pode ser vista na Figura 29, o controlador ficou muito bem sintonizado e teve um bom desempenho. O tempo de subida foi $Tr = 0,0101s$ o tempo de assentamento foi de $Ts = 0,0366s$ e o sobressinal máximo foi de $Mp = 6,81\%$.

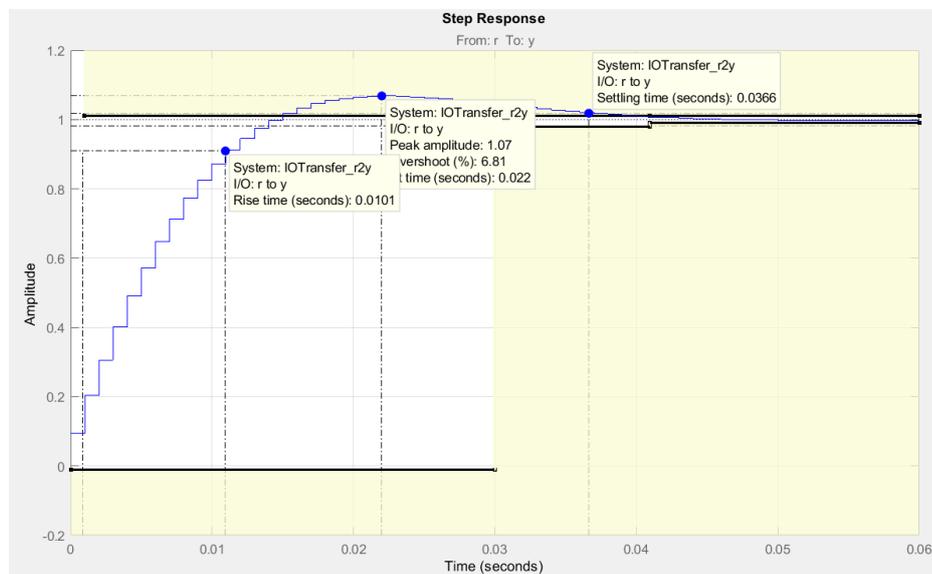
Figura 28 - Comparação entre modelos discreto e contínuo.



Fonte: Autor.

Na Figura 29 a área sombreada em amarelo seria um local indesejado para a resposta do sistema pois ela deriva diretamente das especificações iniciais. A resposta não atende a todas as especificações, mas ainda assim a resposta obtida pelo controlador é bastante satisfatória.

Figura 29 - Resposta ao degrau PID discreto.



Fonte: Autor.

O controlador ficou como mostrado na Equação (31) e segue o padrão apresentado nas Equações (5, 32).

$$C(z) = \frac{0,1567 \cdot (z - 0,8415) \cdot (z - 0,0375)}{z^2 - z} \quad (31)$$

Para encontrar os coeficientes da equação a diferenças que representa o controlador basta alguma manipulação matemática lembrando que a Equação (32).

$$C_{PID}(z) = q_0 \frac{(z - z_1) \cdot (z - z_2)}{z^2 - z} \quad (32)$$

$$z_1 + z_2 = \frac{-q_1}{q_0}, \quad z_1 \cdot z_2 = \frac{q_2}{q_0} \quad \text{e} \quad q_0 \cong \frac{u(0)}{e(0)}$$

É fácil deduzir que $q_0 = 0,1567$, $q_1 = -0,1377393$ e que $q_2 = 0,00494486$ podendo assim montar uma equação a diferenças como a mostrada na Equação (9) e seria uma equação a diferenças que dependeria somente do erro do sistema em instantes anteriores e do sinal de controle aplicado por último. Essa equação pode ser vista na Equação (33).

$$u(k) = u(k-1) + 0,1567 \cdot e(k) - 0,1377393 \cdot e(k-1) + 0,00494486 \cdot e(k-2), \quad (33)$$

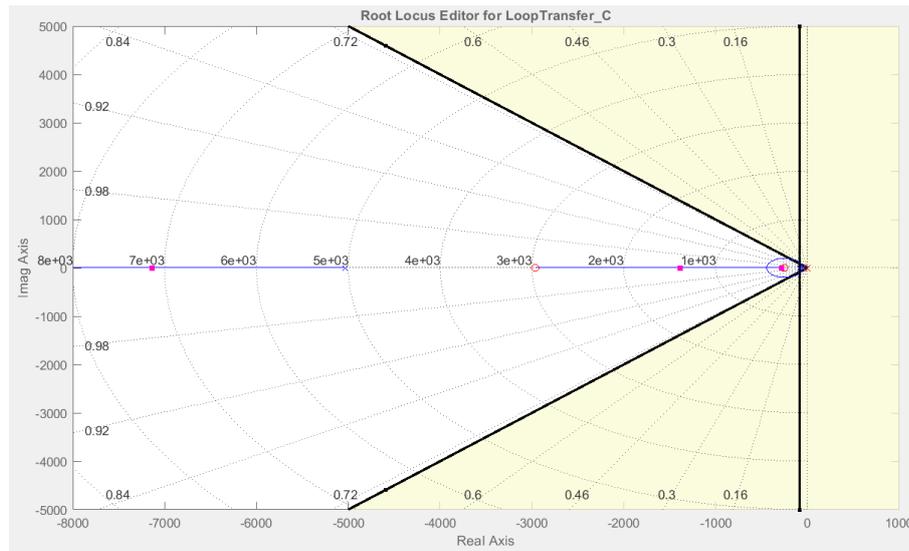
onde $u(k)$ é o sinal de controle no instante k .

4.2.3 Controlador na forma RST

Como o RST é uma forma de aplicação de controladores, o controlador aqui foi projetado por alocação de polos e ajuste de ganho. A Figura 30 mostra o lugar das raízes do sistema com os polos e zeros colocados para o controlador.

As restrições que aparecem na imagem como uma região sombreada são relativas ao sobressinal na reta mais a direita na figura e quanto ao tempo de assentamento como as retas simétricas ao eixo real que mostram onde os polos do sistema devem estar. Para atingir essas restrições o ganho do controlador é ajustado. Para o sistema ser realizável, foi adicionado um outro polo na origem do sistema.

Figura 30 - Lugar das raízes com os polos do controlador.



Fonte: Autor.

Foi feita uma sintonia pensando no melhor desempenho possível para esse modelo de motor BLDC. Assim, o controlador encontrado só foi otimizado para melhorar a performance e manter um sobressinal mínimo, como pode ser visto na resposta ao degrau na Figura 31.

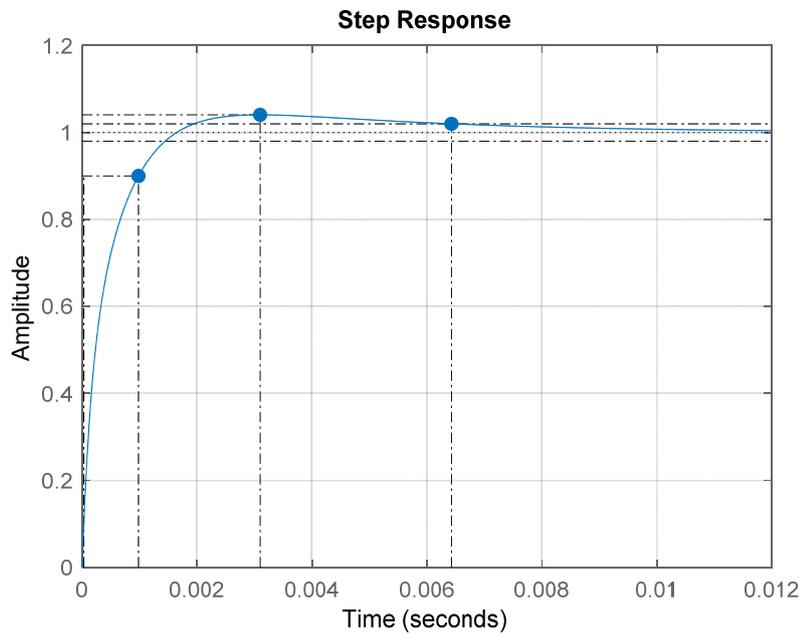
O desempenho do sistema é bastante satisfatório e com esse controlador o sistema obteve um tempo de subida $Tr = 0,00104s$, um tempo de assentamento de $Ts = 0,00643s$ e um máximo sobressinal percentual de $Mp = 4,06\%$, o que é um ótimo desempenho. A equação do controlador pode ser observada em (33).

$$C_{RST}(s) = \frac{8,4907 \cdot 10^{-4} \cdot (s + 2963) \cdot (s + 252,3)}{s^2} \quad (33)$$

Para o controlador mostrado vale uma aproximação do PID contínuo escrito da forma mostrada na Equação (28) e esta mesma representação foi transformada para a forma RST passando primeiramente para um controlador PID discreto e depois reescrito na forma RST (LANDAU, 1998).

Vale ressaltar aqui que o PID discreto é um caso particular para o controlador com dois graus de liberdade na forma RST e não devem ser assumidos como similares. A construção do RST vem das Equações Diofantinas e sua relação com a ordem dos polinômios do modelo (LANDAU, 1998).

Figura 31 - Resposta ao degrau RST em MF.



Fonte: Autor.

O controlador PID discreto em questão é representado pela forma mostrada na Equação (35) com um período de amostragem $\Delta t = 0,001s$ escolhido anteriormente. Vale ressaltar também que z^{-1} é o operador de atraso que representa o atraso de um instante de tempo, e.g. $y(k).z^{-1} = y(k-1)$.

$$C_{RST}(s) = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) = K_p \left(1 + \frac{\frac{\Delta t}{T_i}}{1 - z^{-1}} + \frac{T_d}{\Delta t} \cdot (1 - z^{-1}) \right) \quad (35)$$

Pode ser observado em diagrama de blocos padrão que o controlador relaciona o erro com o sinal de controle. Então é possível dizer que $C(z^{-1}) = \frac{u(t)}{e(t)}$ e ainda que o erro é $e(t) = w(t) - y(t)$ onde w é a representação do sinal de referência e y é a representação do sinal de saída do sistema.

Lembrando o que foi mostrado na Equação (8) e considerando o desenvolvimento matemático simples que levou a Equação (36):

$$T(z^{-1}).w(t) = S(z^{-1}).u(t) + R(z^{-1}).y(t)$$

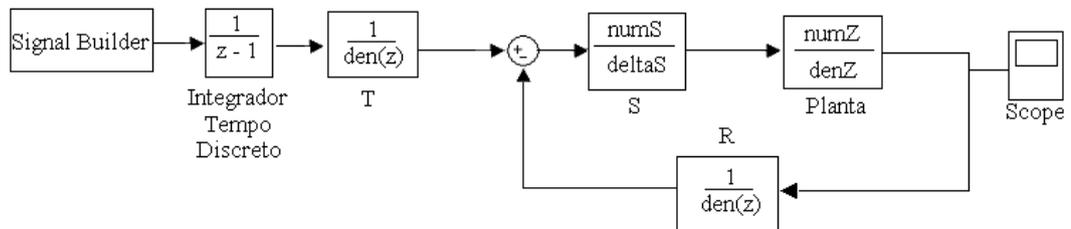
$$(1 - z^{-1}).u(t) = K_p \left(\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) - \left(1 + 2 \cdot \frac{T_d}{\Delta t} \right) \cdot z^{-1} + \frac{T_d}{\Delta t} \cdot z^{-2} \right) \cdot (w(t) - y(t)) \quad (36)$$

É fácil concluir que R e T são iguais e S é como segue na Equação (37).

$$T(z^{-1}) = R(z^{-1}) = K_p \left(\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) - \left(1 + 2 \cdot \frac{T_d}{\Delta t} \right) \cdot z^{-1} + \frac{T_d}{\Delta t} \cdot z^{-2} \right) \quad (37)$$

$$S(z^{-1}) = (1 - z^{-1})$$

Figura 32 - Representação do controlador RST no *Simulink*.



Fonte: adaptado (CORRÊA, SILVA e DEDINI, 2010).

Isso posto segue a formulação do controlador na forma RST e com os ganhos equivalentes do controlador sendo $K_p = 2,64 \cdot 10^{-7}$, $T_i = 4,301075 \cdot 10^{-3}$ e $T_d = 3,11 \cdot 10^{-4}$ ainda mantendo o período de amostragem se obtém as equações mostradas na Equação (38).

$$T(z^{-1}) = R(z^{-1}) = 4,0748 \cdot 10^{-7} - 4,2821 \cdot 10^{-7} \cdot z^{-1} + 8,2104 \cdot 10^{-8} \cdot z^{-2} \quad (38)$$

$$s(z^{-1}) = (1 - z^{-1})$$

A implementação em *Simulink* pode ser feita como mostrado na Figura (32).

Esta representação é uma ótima forma de implementar em sistemas reais que não possuem um computador para operá-los e já foi usada em diversas aplicações industriais (LANDAU, 1998).

4.2.4 O sinal de controle

A simulação mostrou que para entradas de até 12Vcc o valor máximo do sinal de controle é de 17,9Vcc. Essa seria uma análise do pior caso, onde o motor estava em repouso e foi para a tensão máxima de saída com os controladores. Mas fato é que o robô usa baterias limitadas a 12,4Vcc, portanto nunca poderá assumir esses valores. Além disso existe um circuito de segurança que desarma a placa principal se a corrente passar de 3,6A e um elemento fusível de 4A, o que impede uma sobrecarga.

No *firmware* do robô existe uma lógica de programação que limita o sinal aplicado aos motores para que nunca ultrapasse um máximo de segurança, se o sinal que seria enviado aos motores for maior do que o limite pré-estabelecido, ele é substituído por esse limite.

Por fim, os motores são especificados para uma tensão de até 24Vcc, trabalhando, portanto, com metade da tensão nominal máxima.

Além desses fatores, o valor final do sinal de controle fica igual para degraus iguais aplicados aos motores, então optou-se por não realizar uma análise mais profunda desse sinal.

4.3 COMPARATIVO ENTRE OS RESULTADOS

Para a sintonia dos controladores foi utilizado o MATLAB® com suas ferramentas, como o *Simulink*, o *RLTOOLS* e o *LTIVIEW* para a simulação e ajuste dos parâmetros dos controladores que foram testados. Todos os modelos em *Simulink* e todos os códigos gerados estão disponíveis nos anexos.

Os controladores propostos foram apresentados na seção 2.3 com a teoria básica relativa a cada um e são:

- o PID contínuo;
- o PID discreto;
- o RST gerado por alocação de polos.

Para cada controlador foi gerado uma resposta ao degrau unitário como forma de encontrar as métricas principais de desempenho e poder compará-las. Isso foi feito em ambiente de simulação do MATLAB® utilizando o modelo mostrado na seção 4.1.3.

Através das respostas simuladas foram colhidos os dados apresentados na tabela 5, conforme descrito nas seções dos controladores. As métricas utilizadas para a comparação dos resultados são recomendadas como válidas para análise da resposta transitória de sistemas de controle (OGATA, 2000).

A primeira métrica é o tempo de subida, T_r , que representa o tempo entre o momento em que a saída do sistema atinge 10% da entrada e o instante que saída chegou a 90% do valor final.

O máximo sobressinal, ou valor de pico, M_p , que representa o percentual máximo atingido pela saída em relação ao valor final em regime permanente.

O tempo de assentamento, T_s , representa o tempo que o sistema leva para entrar em uma faixa aceitável de erro, geralmente entre 2% e 5%, e conta desde a aplicação do sinal de entrada até a resposta do sistema não sair mais dessa faixa. No caso do MATLAB® a faixa padrão da função degrau é em torno de 2% do valor final.

O erro em regime permanente, ou erro em estado estacionário, ess , mostra a diferença entre valores de entrada e de saída do sistema após ele atingir o tempo de assentamento, ou seja, mostra a diferença entre o valor que a saída deveria ter em relação a saída simulada após o sistema estabilizar.

A Tabela 5 mostra um comparativo entre essas métricas obtidas em simulação e o sistema original em Malha Aberta. Aqui as funções foram reescritas e sofreram arredondamentos nos valores dos polos, zeros e ganho dos controladores, então elas podem variar um pouco do que foi mostrado anteriormente em que os controladores possuíam toda a precisão numérica do MATLAB®. Os resultados continuam coerentes com o esperado e foram obtidos pelo código que também gerou a Figura 33 e está disponível no Anexo I.

Tabela 5 - Comparativo das métricas de desempenho.

Sistema	T_r	T_s	M_p	ess
PID contínuo	0,00062s	0,012s	2,06%	0
PID discreto	0,00912s	0,03470s	9,33%	0
Forma RST	0,000844s	0,00646s	4,05%	0

Fonte: Autor.

Pelos dados da tabela é visível uma diferença significativa entre os controladores aplicados ao modelo da planta, no caso o BLDC. A Figura 33 mostra

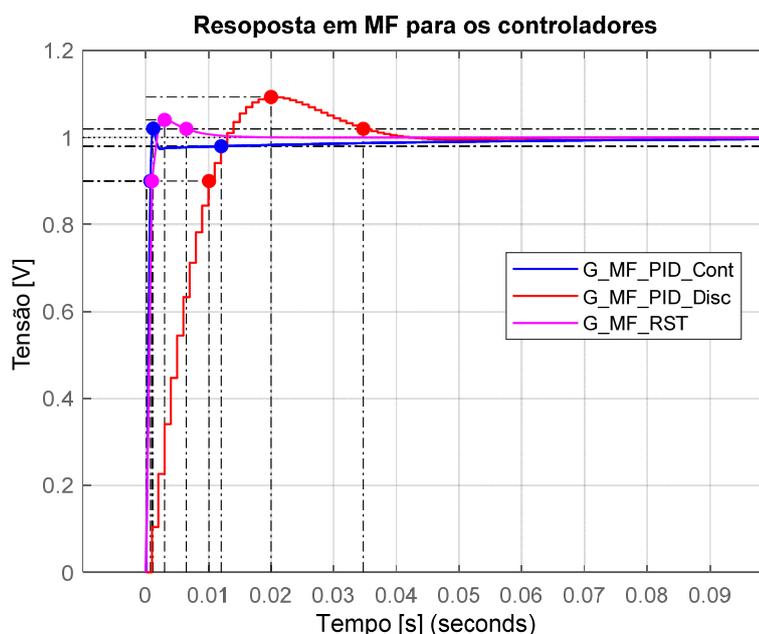
um comparativo da resposta ao degrau unitário aplicado na planta em Malha Fechada com cada um dos controladores.

Pode ser visto que existem diversas abordagens para um mesmo problema e que a escolha da melhor solução não é unívoca e depende bastante dos requisitos de desempenho bem como da disponibilidade de *hardware*.

Nota-se uma mudança no comportamento dos sistemas devido a um truncamento de valores e ganhos dos controladores. Ao pensar em um *hardware* embarcado com limitação de *bytes* para ponto flutuante os valores se alteram ainda mais.

A escolha de controladores é um compromisso entre desempenho, disponibilidade de *hardware* e simplicidade, pois quanto mais simples o sistema, menor a carga computacional. Além disso, ao levar menos tempo para executar o algoritmo, menor a probabilidade de ocorrer alguma interrupção no controlador e e assim interromper a execução do controle. Quanto mais simples o sistema, menor a necessidade de um *hardware* poderoso, porém se o desempenho é fundamental muitas vezes um *hardware* mais poderoso é necessário.

Figura 33 - Comparativo dos controladores em MF.



Fonte: Autor.

Aqui se vê um controlador mais sofisticado, o RST, com um desempenho mais efetivo na opinião do projetista, mas longe de ter a maior simplicidade. Por outro lado,

se tem um controlador com um desempenho mediano e um uso simples, no caso o PID contínuo.

A recomendação do autor, tendo em vista o conhecimento da aplicação, as conversas com outros pesquisadores e o resultado dessa pesquisa, fica no PID contínuo aplicado de forma discreta. Para essa discretização foi utilizado um segurador de ordem zero com um período de amostragem compatível ao do hardware utilizado que é de $\Delta t = 0,0016s$.

5 ESTUDO DE CASO

Nesta Seção será mostrada uma aplicação no robô real da equipe RoboFEI, dessa forma buscando melhorar a sua movimentação. Uma foto do campo da equipe RoboFEI pode ser observada na Figura 34.

Figura 34 - Foto do campo da Equipe RoboFEI.



Fonte: Autor.

Durante o desenvolvimento desse trabalho, a equipe continuou, como era de se esperar, o desenvolvimento do robô novo, do *software* da estratégia e do *firmware* também. Esse desenvolvimento resolveu alguns problemas que a equipe possuía no início desse trabalho.

O objetivo principal deste trabalho é melhorar o controle interno do robô, melhorando a cinemática e controlando os motores para dessa forma melhorar a precisão e o desempenho do robô em campo. Antes desse desenvolvimento o controle interno funcionava em malha aberta.

NA fase atual do projeto existe um fechamento da malha da visão no computador da estratégia, ainda que sem controlador sintonizado, que melhorou significativamente a posição dos robôs em campo. A malha da visão/movimentação executa seu laço de repetição em aproximadamente $15ms$ e consegue corrigir a trajetória do robô se ele estiver se distanciando do caminho planejado.

Como a melhora esperada é na performance do robô, optou-se por aplicar testes que medissem o desempenho dele em campo.

5.1 APLICAÇÃO NO ROBÔ

Para a aplicação em tempo real com o robô da equipe RoboFEI em campo existem algumas limitações. Nessa fase de desenvolvimento do projeto não é possível receber dados confiáveis dos robôs pelo sistema de rádio proprietário, portanto os dados foram todos obtidos através do sistema de visão padronizado do laboratório.

Esse sistema de visão é um projeto *open source* mantido por toda a comunidade SSL. Foi desenvolvido para unificar a liga com um sistema confiável e acessível para todo o mundo (ZICKLER, LAUE, *et al.*, 2009).

Para este teste em tempo real foi escolhido o controlador PID contínuo sintonizado na seção 4.2.1, e sua equação pode ser observada na Equação (29). Para implementar um controlador em uma planta microcontrolada ele deve ser discreto, ou seja, trabalhar com intervalos de tempo pré-determinados, também chamados de período de amostragem (CASTRUCCI, BITTAR e SALES, 2011).

Dessa forma o processador pode trabalhar como PID aproximado pelas equações de diferenças finitas (CASTRUCCI, BITTAR e SALES, 2011), ou simplesmente equações a diferenças.

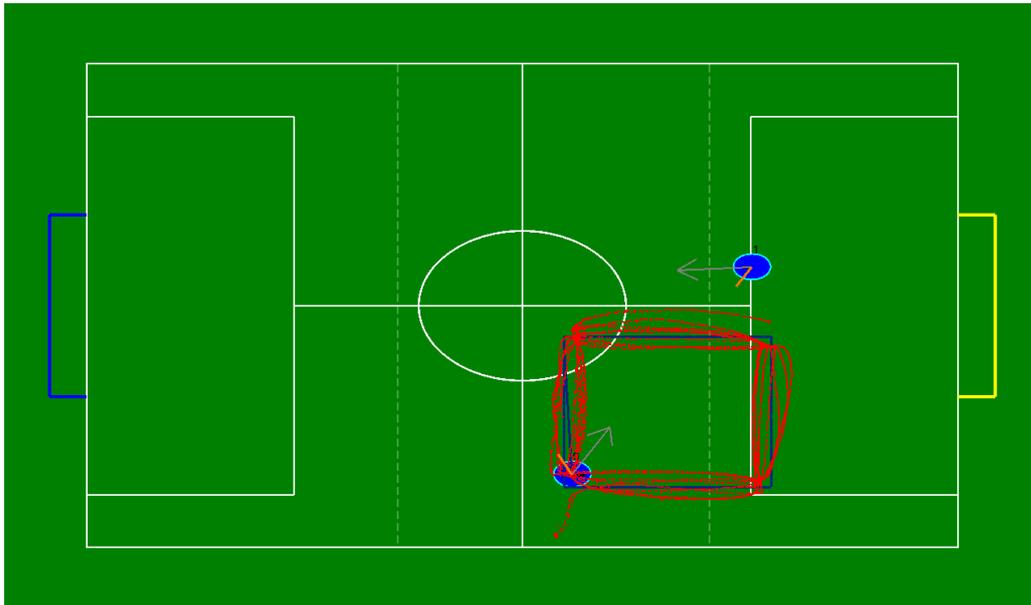
Para o controlador PID padrão pode-se deduzir uma equação a diferenças como a mostrada na Equação (36) onde n representa um instante de tempo discreto (STARR, 2006). Para essa discretização foram utilizados um segurador de ordem zero na entrada do controlador e um período de amostragem coerente com o *hardware* do robô, no caso $\Delta t = 0,0016s$.

$$u[n] = K_p \cdot e[n] + K_d \cdot (e[n] - e[n - 1]) + K_i \cdot (f[n] + f[n - 1]) \quad (36)$$

A representação $e[n] = w[n] - y[n]$ mostra que o erro é a diferença entre a entrada e a saída do sistema em cada instante n ; K_p , K_i e K_d representam os ganhos do controlador e $f[n]$ é um somatório finito do erro.

Na implementação atual foi visto empiricamente que o erro deveria ser guardado por 4 instantes de tempo para a função $f[n]$ do integrador funcionar efetivamente, ou seja, o integrador é atualizado a cada instante de tempo porém somente os últimos quatro instantes são levados em conta.

Figura 35 - Quadrado de 1m x 1m MF sem controlador.



Fonte: Autor.

As Figuras 35 e 36 mostram um robô real em campo realizando uma trajetória quadrada de 1m por 1m em que se espera ver alguma diferença no comportamento do robô.

Essa trajetória é baseada em uma sugestão de deslocamento de um robô a 1m/s durante 1s e depois uma mudança de direção em 90° no sentido anti-horário e seguir por mais 1s a m/s e assim por diante (SIEGWART e NOURBAKHS, 2004).

A Figura 35 mostra o robô fazendo um quadrado de 1m por 1m com os motores em Malha Fechada com ganho unitário enquanto a Figura 36 mostra ele realizando este movimento com o controlador PI sintonizado e implementado no *firmware* do robô.

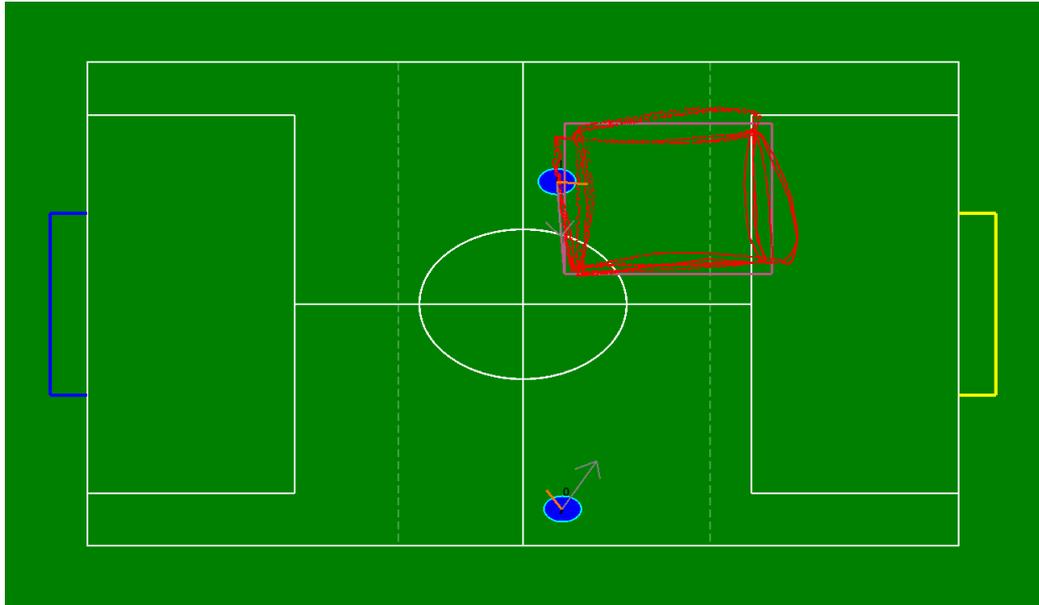
É possível observar que as trajetórias são muito semelhantes, devido ao fechamento de malha do sistema de visão que melhora o desempenho do sistema como um todo ao devolver informações para ele e corrigir a trajetória em tempo real.

Deve se levar em conta também que a geração de velocidades é uma tarefa do sistema de estratégia e não é possível ser colocado em um valor predeterminado. Devido a isso, o comportamento dos robôs com as pequenas distâncias é muito semelhante tanto com o controlador como sem ele.

Para obter uma medição de desempenho mais significativa optou-se por desenvolver a maior trajetória possível em campo, permitindo assim que o robô atinja as maiores velocidades possíveis. Para isso foi gerada uma trajetória na diagonal do

campo com o objetivo de os robôs cruzarem o campo do laboratório em diagonal indo e vindo, a maior distância existente ali. Isso foi feito com o controlador e sem o controlador.

Figura 36 - Quadrado de 1m x 1m com controlador.



Fonte: Autor.

Feito isso pode ser observado um ganho significativo em desempenho do robô com o controlador implementado, pois ele consegue atingir a velocidade máxima dos motores com carga, o que era impossível antes em malha aberta.

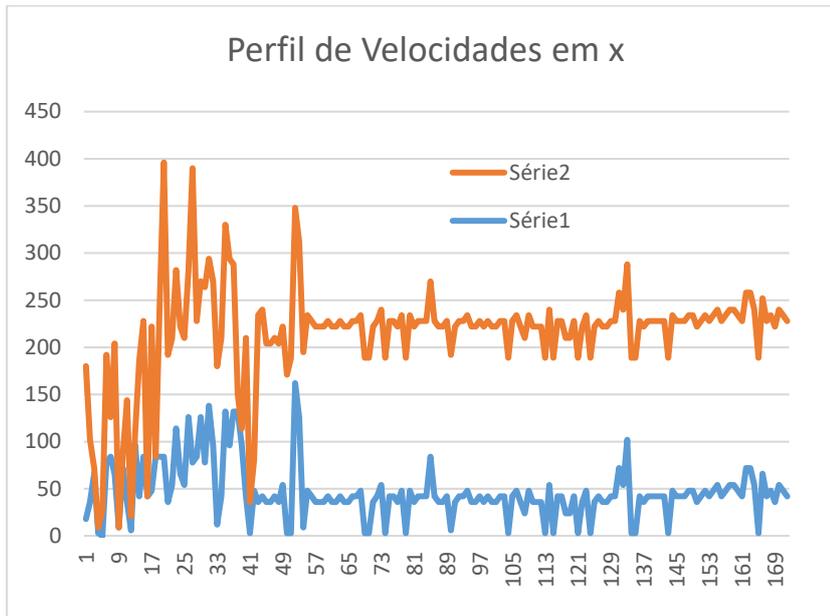
Tabela 6 - Comparativo das médias de velocidade.

Situação	Vx médio	Vy médio
Sem controlador	39,63	29,65
Com controlador	99,73	118,24

Fonte: Autor.

Ao se comparar o desempenho do robô nessas circunstâncias, foi observado que com o controlador PI implementado obteve-se um ganho significativo em velocidade do robô. Dessa forma o robô foi muito mais consistente no desenvolvimento das velocidades atingindo valores médios maiores do que sem controlador implementado.

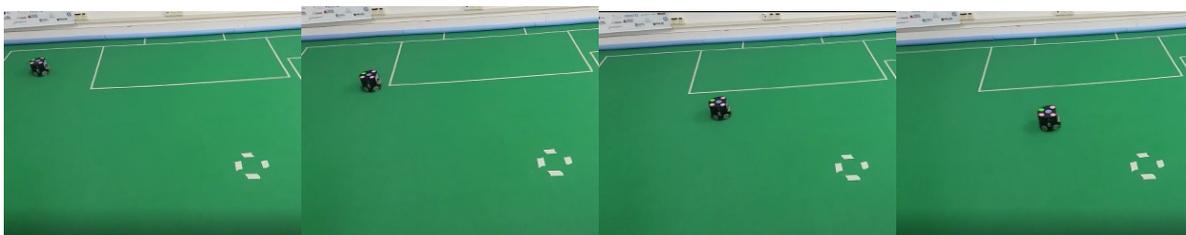
Figura 37 - Perfil de velocidades no eixo x do robô.



Fonte: Autor.

A melhora no desempenho do robô é demonstrada na Figura 37 que mostra os perfis de velocidade do robô em magnitude para o eixo x. De forma ilustrativa a figura foi gerada com 170 pontos. A média das velocidades, em valores absolutos, são observadas na Tabela 6.

Figura 38 - Capturas de tela do robô sem o controlador.



(a) 0 segundo

(b) 1 segundo

(c) 2 segundos

(d) 3 segundos

Fonte: Autor.

É fácil ver que a performance dos robôs aumentou de forma significativa, como um resultado direto da aplicação do controlador. Ainda na Figura 37 é possível observar uma rampa de aceleração do robô com o controlador implementado na "Série 2" e sem ele na "Série 1". É fácil ver um ganho significativo nesse caso.

Figura 39 - Capturas de tela do robô com o controlador



Fonte: Autor.

A combinação de fotos mostrada na Figura 38 reflete capturas de tela do vídeo feitos com um robô sem controle enquanto a Figura 39 mostra capturas no mesmo instante de tempo em um robô com controle implementado e funcionando. Essa mesma rampa de aceleração gerou os dados experimentais mostrados na Figura 37.

Nas imagens fica óbvio a melhora significativa de desempenho do robô com o controlador implementado, pois a partir da segunda captura já se percebe que o robô com o controlador começa a ganhar uma vantagem em distância em relação ao robô sem o controlador. Isso reflete claramente o resultado do trabalho.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Os estudos bibliográficos forneceram a base para o entendimento do entorno do problema apresentado, passando pelo fundamento da robótica móvel, entrando nas especificidades dos motores BLDC e seguindo até alguns tipos de controladores usados comumente em problemas similares.

A pesquisa realizada com os trabalhos mais recentes, principalmente nos que estão inseridos no contexto da comunidade RoboCup SSL, mostraram que vários grupos de pesquisa estão estudando o tema neste momento. Isso fornece um indício bastante sugestivo da importância e relevância do tema estudado.

Com os estudos da comunidade foi possível também perceber uma série de soluções distintas, mostrando que o caminho proposto aqui não é único. Dessa forma, deve-se entender que as conclusões tiradas a partir deste trabalho não são absolutas.

Para a execução do estudo e sua análise foram apresentadas diversas métricas clássicas da área de engenharia de controle de sistemas, que podem ser aplicadas em simulação para comparar os controladores propostos quanto ao desempenho, são elas:

- Tempo de Subida, T_r ;
- Máximo Sobressinal, M_p ;
- Tempo de Assentamento, T_s e;
- Erro em regime permanente.

Com todo o arcabouço juntado até aqui é possível elencar prós e contras de cada controlador estudado, mostrar os dados analíticos de simulação pelas técnicas clássicas e finalmente assim conseguir fazer uma recomendação de uso para equipe RoboFEI.

Como a equipe optou por manter as implementações o mais simples possível a escolha final ficou com o PID contínuo sem o termo derivativo aplicado discretizado. Apesar de ele não ter tido o melhor desempenho em todas as métricas, os resultados foram suficientes para as necessidades da equipe.

Conclui-se também que a escolha de uma técnica de controle é uma tarefa complicada e que deve ser feita com muita cautela, sempre observando o compromisso entre simplicidade e desempenho desejado, pois apesar de terem sido implementados em simulação somente controladores simples, existe uma diferença significativa para implementação em tempo real. Existe um controlador sintonizado, o

RST, que apresentou um desempenho superior, mas que não foi considerado pela sua complexidade de implementação.

Foram feitas simulações com os controladores e foram apresentados meios de implementação de cada um estudado para fornecer a equipe uma possibilidade de mudança futura, talvez em um novo robô, para um controlador com desempenho superior.

6.1 TRABALHOS FUTUROS

Este trabalho também serve de base para diversos desenvolvimentos futuros, pois fornece entendimento básico e nortes para diversas pesquisas.

Primeiramente foi desenvolvida uma modelagem dinâmica inicial a partir das forças de acoplamento entre as rodas e o piso, sem levar em conta o escorregamento e o atrito que pode ser complementado com outra pesquisa.

Outra proposição fica em realizar um modelo computacional, talvez em MATLAB®, do comportamento de um robô omnidirecional com tração em quatro rodas, 4WD – do inglês, Four Wheel Drive. Existem alguns estudos de robôs de serviço omnidirecionais com três rodas, mas quase nada de sobre esses de quatro rodas e esse trabalho fornece as bases para um desenvolvimento neste sentido.

Além disso, se pode pensar em utilizar controladores mais avançados para esse e outros objetivos no robô, como os controladores de alto ganho que é o caso do *Sliding Mode Control*, ou controle de modos deslizantes. Isso usado tanto para os motores como também pode ser testado na malha de alto nível com o sistema de visão. Um exemplo desse tipo de técnica aplicada a posicionamento de embarcações é um bom referencial para este tipo de trabalho (AGOSTINHO, 2009).

Outra sugestão é um desenvolvimento de um sistema de controle para a malha de controle da visão deve ser observado como um tema fértil neste tipo de estudo relacionados a sistemas de controle.

REFERÊNCIAS

AG, M. M. **Program 2017/18 - High Precision Drives and Systems**. Sachseln, p. 480. 2017.

AGOSTINHO, A. C. **CONTROLE POR MODOS DESLIZANTES APLICADO A SISTEMA DE POSICIONAMENTO DINÂMICO**. Escola Politécnica da USP. São Paulo. 2009.

ALMAGRO, J. et al. RoboJackets 2018 Team Description Paper. **2018 RoboCup**, 2018. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2018-paper37.pdf>>. Acesso em: 01 jun. 2018.

ÅSTRÖM, K. J.; HÄGGLUNG, T. **PID Controllers, Theory of Design and Tuning**. Nova Iorque: Instrument Society of America, 1995.

ÅSTRÖM, K. J.; WITTENMARK, B. **Computer Controlled Systems – Theory and Design**. New Jersey: Prentice Hall, 1990.

BAI, C. Y. et al. 2017 Team Description Paper: UBC Thunderbots. **2017 RoboCup**, 2017. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2017-final4.pdf>>. Acesso em: 01 nov. 2017.

BALDURSSON, S. **BLDC Motor Modelling and Control – A Matlab®/Simulink® Implementation**. Gotemburgo, Suécia. 2005.

BAUER, A. IFR.org. **International Federation of Robotics**, 2018. Disponível em: <<https://ifr.org/free-downloads/>>. Acesso em: 01 dez. 2018.

BOOS, E. **Métodos Iterativos para a Pseudo-Inversa de Moore-Penrose e Aplicações na Resolução de Sistemas Lineares**. Universidade Federal de Santa Catarina – UFSC. Florianópolis, p. 21 - 32. 2015.

BÜHLMAYER, J. et al. ER-Force Extended Team Description Paper RoboCup 2017. **2017 RoboCup**, 2017. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2017-final2.pdf>>. Acesso em: 01 nov. 2017.

CASTRUCCI, P. D. L.; BITTAR, A.; SALES, R. M. **CONTROLE AUTOMÁTICO**. Rio de Janeiro: LTC, 2011.

COMMITTEE, T. SSL - RoboCup. **RoboCup**, 2018. Disponível em: <<https://robocup-ssl.github.io/ssl-rules/sslrules.html>>. Acesso em: 01 dez. 2018.

COMMUNITY, R. RoboCup MainPage, 01 jan. 2008. Disponível em: <http://wiki.robocup.org/Small_Size_League>. Acesso em: 04 abr. 2018.

CONSENZA, C. S. et al. RoboIME: on the road to RoboCup 2017. **2017 RoboCup**, 2017. Disponível em: <http://wiki.robocup.org/Small_Size_League/RoboCup_2017/Teams>. Acesso em: 01 jun. 2017.

CORRÊA, F. C.; SILVA, L. C.; DEDINI, F. G. **CONTROLE DE UMA JUNTA ROBÓTICA UTILIZANDO CONTROLADOR PID NA FORMA RST**. CONEM 2010 - VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA. Campina Grande, PB: [s.n.]. 2010.

DE SÁ, L. N. **Modelagem e Controle de um Veículo Omnidirecional**. Pontifícia Universidade Católica do Rio de Janeiro - PUCRIO. Rio de Janeiro. 2016.

DOORNKAMP, C. et al. RoboTeam Twente 2018 Team Description Paper. **2018 RoboCup**, 2018. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2018-paper37.pdf>>. Acesso em: 01 jun. 2018.

FRANKLIN, G. F.; POWELL, J. D.; WORKMAN, M. **Digital Control of Dynamic System**. 3ª. ed. Massachussets: Addison-Wesley, 1998.

FU-CHENG, L.; XUE-WEI, Z. **Research on Control Algorithm of High Voltage Circuit Breaker Three Phase Permanent Magnet Brushless DC Motor**. 2016 IEEE International Conference on Power and Renewable Energy. [S.l.]: [s.n.]. 2016.

GARCIA, E. et al. The Evolution of Robotics Research. **IEEE Robotics & Automation Magazine**, v. 1, n. MARCH 2007, p. 90 - 103, 2007.

HOLMBERG, R.; KHATIB, O. Development and Control of a Holonomic Mobile Robot for Mobile Manipulation Tasks. **International Journal of Robotics Research**, Stanford, California - USA, v. 19, n. 11, p. 1066 - 1074, 2000.

HUI, T. S.; BASU, K. P.; SUBBIAH, V. **Permanent Magnet Brushless Motor Control Techniques**. National Power and Energy Conference (PECon) 2003. Bangi - Malásia: [s.n.]. 2003.

HUI, T. S.; BASU, K. P.; SUBBIAH, V. **Permanent Magnet Brushless Motor Control Techniques**. National Power and Energy Conference (PeCon) 2003 Proceedings. Bangi, Malasya: IEEE. 2003. p. 133 - 138.

ITO, M. et al. RoboDragons 2018 Extended Team Description. **2018 RoboCup**, 2018. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2018-RoboDragons.pdf>>. Acesso em: 01 jun. 2018.

KANGSANANT, T.; LIM, P. Y. A. H. A. ANALYSIS AND DESIGN OF BRUSHLESS POSITION CONTROL SYSTEM WITH DERIVED VELOCITY FEEDBACK. **IECON - International Conference on Industrial Electronics**, Singapura, 24-28 outubro 1988. 720 -724.

KANGSANAT, T.; LIM, P. Y.; HARVEY, A. **ANALYSIS AND DESIGN OF BRUSHLESS POSITION CONTROL SYSTEM WITH DERIVED VELOCITY FEEDBACK**. IECON '88 : 1988 International Conference on Industrial Electronics, Control and Instrumentation. Singapura - Malásia: [s.n.]. 1988.

KO, S. A. et al. A robust digital position control of brushless DC motor with dead beat load torque observer. **IEEE Transactions on Industrial Electronics**, OUTUBRO 1993. 512 - 520.

KURFESS, T. R. **ROBOTICS AND AUTOMATION HANDBOOK**. 1ª. ed. Flórida - USA: CRC Press LLC, v. Único, 2005.

LAGES, W. F. **Modelagem Cinemática e Dinâmica**. Universidade Federal do Rio Grande do Sul - UFRGS. Porto Alegre. 2013.

LANDAU, I. D. The R-S-T digital controller design and applications. **Control Engineering Practice**, 1 Fevereiro 1998. 155 - 165.

LANG, R. G. et al. Description of the Warthog Robotics SSL 2017. **2017 RoboCup**, 2017. Disponível em: <http://wiki.robocup.org/Small_Size_League/RoboCup_2017/Teams>. Acesso em: 01 jun. 2017.

LI, C. et al. **ZJUNlict**: RoboCup 2014 Small Size League Champion. RoboCup 2014. [S.I.]: Springer International Publishing Switzerland 2015. 2015. p. 47 - 59.

LOBMEIER, C. et al. ER-Force 2018 Extended Team Description Paper. **2018 RoboCup**, 2018. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2018-paper36.pdf>>. Acesso em: 01 jun. 2018.

MACDOUGALL, M. et al. 2018 Team Description Paper: UBC Thunderbots. **2018 RoboCup**, 2018. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2018-UBCThunderbots.pdf>>. Acesso em: 01 jun. 2018.

MARIAPPAN, M. et al. **Simultaneous Rotation and Translation Movement for Four Omnidirectional Wheels Holonomic Mobile Robot**. 2014 IEEE International Symposium on Robotics and Manufacturing Automation. [S.I.]: IEEE. 2014. p. 69 - 73.

MAYA, P.; LEONARDI, F. **Controle Essencial**. São Paulo: Pearson Education do Brasil Ltda., 2014.

MESSNER, W.; TILBURY, D. Control Tutorials for Matlab. **Carnegie-Mellon Tutorials**, 1997. Disponível em: <<http://www2.ensc.sfu.ca/people/faculty/saif/ctm/index.html>>. Acesso em: 01 dez. 2018.

OGATA, K. **DISCRETE-TIME CONTROL SYSTEMS**. Segunda. ed. Nova Jersey, EUA: Prentice Hall, v. Único, 1995.

OGATA, K. **Engenharia de Controle Moderno**. 3^a. ed. Rio de Janeiro: LTC, 2000.

OHNO, K. et al. KIKS 2017 Team Description. **2017 RobuCup**, 2017. Disponível em: <<http://wiki.robocup.org/File:Robocupssl2017-final26.pdf>>. Acesso em: 01 nov. 2017.

PARK, S. J. et al. A new approach for minimum-torque-ripple maximum-efficiency control of BLDC motor. **IEEE Transactions on Industrial Electronics**, FEVEREIRO 2000. 109 - 114.

PATEL, V. K. S.; PANDEY, A. K. Modeling and Performance Analysis of PID Controlled BLDC Motor and Different Schemes of PWM COntrollenBLDC Motor. **Internactional Journal of Scientific and Research Publications**, v. 3, n. 4, Abril 2013. ISSN ISSN 2250-3153.

PIERI, E. R. D. **CURSO DE ROBOTICA MÓVEL**. Florianópolis - SC. 2002.

PIN, F. G.; KILLOUGH, S. M. A New Family of Omnidirectional and Holonomic Wheeled Platforms for Mobile Robots. **IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION**, v. 10, n. 1^a, p. 480 - 490, ago. 1994.

RÉGIO, J. **Poemas de Deus e do Diabo**. 12^a. ed. Lisboa: Quasi Edições, v. Único, 2002.

REN, C.; MA, S. **Dynamic Modeling and Analysis of an Omnidirectional Mobile Robot**. International Conference on Intelligent Robots and Systems (IROS). Tóquio: [s.n.]. 2013. p. 4860 - 4865.

RIATTAMA, D. et al. **PENS-Wheel (One-Wheeled Self Balancing Vehicle) Balancing Control using PID Controller**. 2016 International Electronics Symposium (IES). [S.l.]: [s.n.]. 2016.

ROBOCUP FEDERATION. RoboCup, 2016. Disponível em: <<http://www.robocup.org>>. Acesso em: 15 mar. 2018.

ROBOCUP FEDERATION. RoboCup, 2016. Disponível em: <<http://www.robocup.org/objective>>. Acesso em: 04 abr. 2018.

ROJAS, R.; FÖRSTER, A. G. Holonomic Control of a robot with an omnidirectional. **KI - Künstliche Intelligenz**, v. 20, n. 2, p. 12 - 17, 2006.

ROSÁRIO, J. M. **PRINCÍPIOS DE MECATRÔNICA**. São Paulo - SP: Pearson Education do Brasil, 2005.

SCATENA, J. M. **Ambiente de Desenvolvimento de Aplicações para Robôs Móveis**. Escola de Engenharia de São Carlos da Universidade de São Paulo. São Carlos - SP, p. 145. 2008.

SIEGWART, R.; NOURBAKHSI, I. R. **Introduction to Autonomous Mobile Robots**. 1ª. ed. Cambridge, Massachusetts - USA: The MIT Press, v. Único, 2004.

SINGH, S. K.; KATAL, N.; MODANI, S. G. Optimization of PID Controller for Brushless DC Motor by using Bio-inspired Algorithms. **Research Journal of Applied Sciences, Engineering and Technology**, 20 FEVEREIRO 2014. 1302 - 1308.

STARR, G. P. **Introduction to Applied Digital Control**. 2ª. ed. Albuquerque: Department of Mechanical Engineering of University of New México, 2006.

VINIDA, K.; CHACKO, M. **An optimized H infinity strategy for robust control of sensorless BLDC propulsion motor in submarines for improved maneuverability.** 2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES). Trivandrum, India: IEEE. 2016.

WANG, H. T. et al. Application of Taguchi method to robust design of BLDC motor performance. **IEEE Transactions on Magnetics** , SETEMBRO 1999. 3700 - 3702.

WATANABE, K. **Control of an Omnidirectional Mobile Robot.** Second International Conference on Knowledge-Based Intelligent Electronic Systems. Adelaide - AUS: L.C. Jain and R.K Jain. 1998. p. 51 - 61.

WATANABE, K. et al. Analysis and Control for an Omnidirectional Mobile Manipulator. **Journal of Intelligent and Robotic Systems**, Dordrecht, 2000. 3 - 20.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. Wikipédia, a enciclopédia livre. **Wikipédia**, 2018. Disponível em: <<https://pt.wikipedia.org/wiki/Rob%C3%B4>>. Acesso em: 01 jun. 2018.

YOSHIMOTO, T. et al. OP-AmP 2017 Team Discription Paper. **2017 RoboCup**, 2017. Disponível em: <<http://wiki.robotcup.org/File:Robocupssl2017-final9.pdf>>. Acesso em: 01 nov. 2017.

ZICKLER, S. et al. **SSL-Vision: The Shared Vision System for the RoboCup Small Size League.** RoboCup. [S.l.]. 2009.

ANEXO I – CÓDIGOS EM MATLAB®

Constantes.m – código que carrega os parâmetros do motor.

```
%Parametrização
%Valores obtidos do catálogo do fabricante Maxon
Motors
%Motor EC-45 flat 50W partNumber: 251601
R = 0.978; %Ohms - Resistência na armadura entre
fases
L = 0.573e-3; %Henrys - Indutância entre fases
Kt = 33.5e-3; %Nm/A - Constante de torque
tm = 0.0118; %s - Constante de tempo
J = 13.5e-6; %kgm^2 - Inercia do rotor (fornecida em
gcm^2)
p = 3; %número de fases
% cálculo das demais constantes necessárias ao
modelo
te = L/(p*R); %s - Constante elétrica de tempo
Ke = (p*R*J)/(tm*Kt); %Vs/rad - Torque elétrico
```

Modelo_MA.m – código principal de simulação em Malha Aberta.

```
%Código principal
%limpa tudo
close all
clear all
clc

%incluir as constantes
Constantes

%Função de Transferência em MA
G = tf([1/Ke],[tm*te tm 1])

%Plotando os resultados
%Resposta ao degrau MA
figure
step(G, 0.5);
title('Resposta ao degrau em Malha Aberta');
xlabel('Tempo [s]');
ylabel('Tensão [V]');
grid on

%Lugar das Raízes
figure
rlocus(G);
title('Lugar das Raízes MA');
grid on

%Diagrama de Bode
figure
bode(G);
title('Diagrama de Bode MA');
grid on
```

MF_PIDcontinuo.m – código principal de simulação do controlador PID contínuo.

```

%Código principal
%limpa tudo
close all
clear all
clc

%incluir as constantes
Constantes

%construção do numerador e denominador da FTMA
num = [1/Ke];
den = [tm*te tm 1];

%Parâmetros do controlador
Kp = 7.5;
Ki = 50;
Kd = 0;
%Numerador e denominador do PID
numC = [Kd Kp Ki];
denC = [1 0];

%Convolução para encontrar o ramo direto
numMA = conv(num, numC);
denMA = conv(den, denC);

%Fechando a malha (nova função => FEEDBACK)
[numMF, denMF] = cloop(numMA, denMA);

%Mostrando os resultados
t = -0.01:0.001:0.3;
opt = stepDataOptions; %setup do degrau
opt.StepAmplitude = 10; %modificar o degrau
G_MF = tf(numMF, denMF);
step(G_MF, t, opt);
title('Resoposta em MF para o PID contínuo');
xlabel('Tempo [s]');
ylabel('Tensão [V]');

```

Modelo_MA_Discreto.m – código que compara o modelo discreto com o modelo contínuo.

```
%Código principal
%limpa tudo
close all
clear all
clc

%incluir as constantes
Constantes

%Função de Transferência em MA contínua
G = tf([1/Ke],[tm*te tm 1])
dt = 0.001; % período de amostragem

%Discretização do sistema
Gz=c2d(G, dt)

%Plotando os resultados
%Resposta ao degrau MA
figure
tf = 0.3; %tempo final da simulação
tc = 0:0.001:tf;
tk = 0:dt:tf;
yc = step(G, tc);
hold on
yk = step(Gz, tk);
plot(tc, yc, tk, yk, 'o')
title('Resposta ao degrau em MA - contínuo x
discreto');
xlabel('Tempo [s]');
ylabel('Tensão [V]');
grid on
```

CompRespostas.m – código que compara os três controladores e mostra em uma mesma figura.

```
%Código principal
%limpa tudo
close all
clear all
clc

%incluir as constantes
Constantes

%Função de Transferência em MA contínua
G = tf([1/Ke],[tm*te tm 1]);

%Discretização do sistema
dt = 0.001; % período de amostragem
Gz=c2d(G, dt);

%% PID contínuo
%Parâmetros do controlador
Kp = 10;
Ki = 50;
Kd = 0;
%Numerador e denominador do PID
numC_PID_Cont = [Kd Kp Ki];
denC_PID_Cont = [1 0];
PID_Cont = tf(numC_PID_Cont, denC_PID_Cont);

%Fechando a malha (nova função => FEEDBACK antiga
cloop
[numMF_PID_Cont, denMF_PID_Cont] =
feedback(PID_Cont*G,1);
%% PID discreto
%Parâmetros do controlador
z = [0.8415 0.0375];
p = [0 1];
k = [0.1567];
%FT do controlador em zero, polo e ganho
PID_Disc = zpk(z, p, k, dt);

%Fechando a malha
G_MF_PID_Disc = feedback(PID_Disc*Gz,1);
```

```
%% Controle por alocação na forma RST
z_RST = [-2963 -252.3];
p_RST = [0];
k_RST = [8.4907e-4];
%Controlador
RST = zpkm(z_RST, p_RST, k_RST);

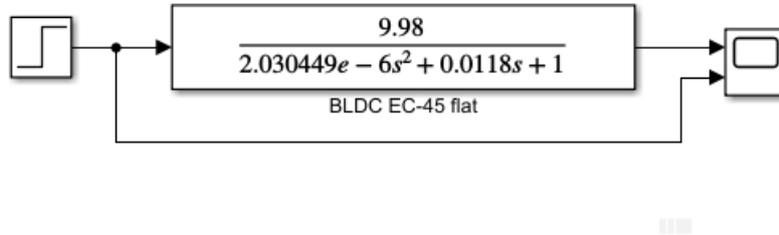
%Fechando a malha
G_MF_RST = feedback(RST*G,1);

%% Mostrando os resultados
%Parametrização do degrau
t = -0.01:0.001:0.1;
tc = -0.01:0.001:0.1;
td = -0.01:dt:0.1;
opt = stepDataOptions; %setup do degrau
opt.StepAmplitude = 1; %modificar o degrau

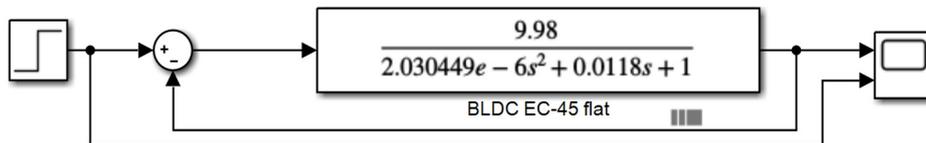
% criação da figura
G_MF_PID_Cont = tf(numMF_PID_Cont, denMF_PID_Cont);
step(G_MF_PID_Cont, t, opt);
hold on
step(G_MF_PID_Disc, td, opt);
step(G_MF_RST, tc, opt);
title('Resposta em MF para os controladores');
xlabel('Tempo [s]');
ylabel('Tensão [V]');
grid on
```

ANEXO II - DIAGRAMAS SIMULINK - TEMPO DE SIMULAÇÃO 0,5S:

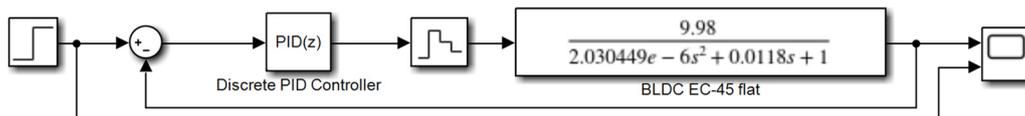
1. Malha Aberta:



2. Malha Fechada:



3. Malha fechada com controlador PID discreto:



4. Malha fechada com controlador PID contínuo:

