

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SAMUEL LOCATELLI

**AVALIAÇÃO DE CONTROLE DE
VELOCIDADE POR PLL NO MOTOR
BRUSHLESS DC**

Porto Alegre

2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SAMUEL LOCATELLI

AVALIAÇÃO DE CONTROLE DE VELOCIDADE POR PLL NO MOTOR BRUSHLESS DC

Relatório Final do Projeto de Diplomação II
apresentado ao Departamento de Engenharia
Elétrica da Escola de Engenharia da Univer-
sidade Federal do Rio Grande do Sul, como
requisito parcial para Graduação em Enge-
nharia Elétrica

Orientador: Prof. Dr. Ivan Müller

Porto Alegre

2022

SAMUEL LOCATELLI

AValiação de Controle de Velocidade por PLL no Motor Brushless DC

Relatório Final do Projeto de Diplomação II apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Prof. Dr. Ivan Müller
Orientador - UFRGS

Prof. Dr. Roberto Petry Homrich
Chefe do Departamento de Engenharia
Elétrica (DELET) - UFRGS

BANCA EXAMINADORA

Prof. Dr. Ivan Müller
UFRGS

Prof. Dr. Jeferson Vieira Flores
UFRGS

Prof. Dr. Tiago Roberto Balen
UFRGS

Dedico este trabalho ao meu avô Arlindo Scherer (in memoriam) pelos incentivos nos trabalhos práticos, e ao meu avô Luis Locatelli (in memoriam) pelos incentivos nos trabalhos artísticos.

Age como se a máxima da tua ação fosse para ser transformada, através da tua vontade, em uma lei universal da natureza.

Immanuel Kant

RESUMO

Motores elétricos fazem parte de forma considerável das tecnologias utilizadas atualmente, estando presentes nos mais diversos tipos de aplicação, e, conseqüentemente, representando também uma parte significativa do consumo de energia elétrica. Por razões de eficiência energética, e de controle de torque e velocidade de forma mais precisa, tem-se buscado pelo desenvolvimento de novas topologias de motores. Atualmente, uma das topologias mais utilizadas e mais promissora é a do motor BLDC (*Brushless DC*). Como exemplo, os motores BLDC são os empregados nos carros elétricos atuais, carregando consigo a possibilidade de redução da emissão dos gases do efeito estufa devido à substituição da utilização de combustível fóssil pela energia elétrica. Os BLDC também são os utilizados na propulsão de drones, aplicação que vem crescendo a cada dia devido ao seu grande potencial na automatização dos mais diversos tipos de serviços. No intuito de aprimorar e desenvolver novas formas de controle de velocidade para motores BLDC, buscando-se melhorar a sua eficiência, precisão e otimização das novas técnicas de implementação, apresenta-se como solução viável, o controle por PLL (*Phase-Locked Loop*). Neste trabalho, implementa-se o controle por PLL em um motor BLDC e compara-se com o método de controle PID, por ser o método mais largamente utilizado no controle de motores. Estudos utilizando métodos de controle por PLL em motores têm mostrado que este apresenta maior precisão no controle de velocidade em detrimento de um tempo maior para se alcançar a referência. Por este motivo, a investigação de um método de controle híbrido também é feita, objetivando-se juntar as melhores características de ambos. Esta junção é chamada de controle dual-PLL. Para a implementação dos controladores PLL, se utilizou a sua forma em software, sendo escolhido o código do modelo de PLL digital, já descrito em outras referências, porém com algumas modificações para que este opere com um motor dentro do laço de realimentação.

Palavras-chave: Controle de velocidade, Motor BLDC, Controle por PLL, Controle não-linear, SPLL.

ABSTRACT

Electric motors are a considerable part of the technologies used today, they are present in the most diverse types of applications, and, consequently, also represent a considerable part of the electrical energy's consumption. Motivated by reasons of energy efficiency, and more accurate torque and speed control, the development of new motor topologies have been made. Currently, one of the most used and most promising topologies is that the BLDC (Brushless Motor DC) motor, which has been widely applied in electric cars, carrying with it the hope of the emission of greenhouse gases reducing, due to the replacement of fossil fuel by electric energy. BLDCs have still been used in the drones's propulsion, an application that has been growing every day due to its great potential for automating too many kinds of services. In order to improve and develop new forms of speed control for these motors, seeking to improve their efficiency, precision and optimization of new implementation techniques, PLL (Phase-Locked Loop) control is presented as an interesting solution. In this work, PLL control is implemented in a BLDC motor, and it is compared with the PID control method, because this is currently the most widely used method in motor control. Studies using PLL control methods in motors have shown that it has better precision in the speed control at the expense of a long time to reach the reference. For this reason, the investigation of a hybrid control method is also carried out, aiming to join the best characteristics of both. This junction is called dual-PLL control. For the PLL controllers implementation, it was used in its software form, being chosen the code of the linear PLL model, one already well know code in other references, but including some modifications, like operating it with a motor inside the feedback loop.

Keywords: Speed control, BLDC motor, PLL cotrol, Nonlinear control, SPLL.

LISTA DE FIGURAS

Figura 1 – Esquemático para um acionamento com ponte completa	16
Figura 2 – Esquemático do motor com acionamento em seis estados de condução .	17
Figura 3 – (a) Posição do rotor (b) Distribuição de fluxo	19
Figura 4 – Relação entre $B(\theta)$, $e_A(\theta)$ e $f_A(\theta)$	21
Figura 5 – Esquemático para um modelo simplificado do motor BLDC	22
Figura 6 – Função de transferência do motor BLDC	24
Figura 7 – Função de transferência do motor BLDC com torque de carga externo aplicado	25
Figura 8 – Sinal <i>PRBS</i> com 200 amostras	27
Figura 9 – Blocos de um PLL	30
Figura 10 – Diagrama de blocos de um SPLL	31
Figura 11 – Pseudo-Código de um SPLL como LPLL	34
Figura 12 – Sinais contínuos e amostrados de um SPLL	35
Figura 13 – Diagrama de blocos para um DPLL como um SPLL	36
Figura 14 – (a) PFD (b) Diagrama de Estados de um PFD	37
Figura 15 – Amostragem dos sinais de um SPLL como DPLL	38
Figura 16 – Diagrama de blocos de um PLL com motor substituindo o bloco VCO	41
Figura 17 – Saída do filtro para o LPLL descrito em software	47
Figura 18 – Saída do filtro para o DPLL descrito em software	50
Figura 19 – Blocos utilizados na lógica de chaveamento	53
Figura 20 – Blocos utilizados na simulação de malha aberta	54
Figura 21 – Resposta ao degrau em malha aberta	56
Figura 22 – Ensaio <i>PRBS</i> simulado	57
Figura 23 – Resposta para a entrada <i>PRBS</i> simulada	58
Figura 24 – Projeto LGR para o PID	59
Figura 25 – Sinal de controle projetado	60
Figura 26 – Arquitetura do processo em malha fechada do ponto de vista da variável fase	62
Figura 27 – Projeto LGR para o PLL	63
Figura 28 – Sinal de controle para a variável fase	64
Figura 29 – Curva de Popov	65
Figura 30 – Tensão e corrente de fase	66
Figura 31 – Sistema dual-PLL	68
Figura 32 – Blocos de realimentação PLL	68
Figura 33 – Sistema experimental	69
Figura 34 – Resposta ao degrau em malha aberta	72

Figura 35 – Ensaio PRBS experimental	73
Figura 36 – Resposta para a entrada PRBS experimental	74
Figura 37 – Projeto do LGR para o PI experimental	75
Figura 38 – Sinal de controle PI do modelo experimental	75
Figura 39 – Projeto do LGR para o PLL experimental	77
Figura 40 – Sinal de controle PLL do modelo experimental	78
Figura 41 – Análise de Popov para o sistema experimental	78
Figura 42 – Resposta das simulações ao degrau	80
Figura 43 – Sinal de controle do PID simulado	81
Figura 44 – Sinal de controle do PLL simulado	82
Figura 45 – Rejeição ao distúrbio dos métodos simulados	83
Figura 46 – Sinal de controle PID na rejeição ao distúrbio simulada	83
Figura 47 – Sinal de controle PLL na rejeição ao distúrbio simulada	84
Figura 48 – Ensaio ao degrau do dual-PLL simulado	85
Figura 49 – Sinal de controle do dual-PLL simulado	85
Figura 50 – Resposta do PID experimental ao degrau	86
Figura 51 – Sinal de controle do PID experimental	87
Figura 52 – Sinal de controle do PLL experimental	88

LISTA DE TABELAS

Tabela 1 – Parâmetros do motor BLDC simulado	52
--	----

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analg-Digital Converter</i>
ADPLL	<i>All-Digital Phase-Locked Loop</i>
BLDC	<i>Brushless DC</i>
CC	Corrente Contínua
CA	Corrente Alternada
DAC	<i>Digital-Analog Converter</i>
DC	<i>Direct Current</i>
DCO	<i>Digitally Controlled Oscillator</i>
DPLL	<i>Digital Phase-Locked Loop</i>
DSP	<i>Digital Signal Processor</i>
ESC	<i>Electronic Speed Controller</i>
FPGA	<i>Field-Programmable Gate Array</i>
LPLL	<i>Linear Phase-Locked Loop</i>
LTI	<i>Linear Time Invariant</i>
PD	<i>Phase Detector</i>
PLL	<i>Phase-Locked Loop</i>
PFD	<i>Phase-Frequency Detector</i>
PRBS	<i>Pseudo-Random Binary Sequence</i>
PWM	<i>Pulse Width Modulation</i>
SPLL	<i>Software Phase-Locked Loop</i>
VCO	<i>Voltage Controlled Oscillator</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contextualização e Motivação	13
1.2	Descrição do Problema	14
1.3	Trabalhos Relacionados	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Modelo da Planta	16
2.1.1	Modelo do Acionamento	16
2.1.2	Modelo do Motor	18
2.1.3	Modelo do Gerador Acoplado	25
2.2	Identificação do Modelo	26
2.2.1	Resposta ao Degrau	26
2.2.2	PRBS	26
2.3	Modelo do Controlador	28
2.3.1	Controlador PID	28
2.3.2	Controlador SPLL	29
2.3.2.1	SPLL como LPLL	30
2.3.2.2	SPLL como DPLL	35
2.4	Modelo do Sensor	40
3	MATERIAIS E MÉTODOS	43
3.1	Simulações	43
3.1.1	Simulações do SPLL	43
3.1.1.1	Simulações do SPLL como LPLL	43
3.1.1.2	Simulações do SPLL como DPLL	47
3.1.2	Simulações do Processo	51
3.1.2.1	Simulações do Processo em Malha Aberta	51
3.1.2.2	Simulações do Processo em Malha Fechada com PID	57
3.1.2.3	Simulações do Processo em Malha Fechada com PLL	61
3.1.2.4	Simulações do Processo em Malha Fechada com Carga	66
3.1.2.5	Simulações do Processo em Malha Fechada com dual-PLL	67
3.2	Experimentos	68
3.2.1	Experimentos Para Identificação	70
3.2.2	Experimento do Processo em Malha Fechada com PID	73
3.2.3	Experimento do Processo em Malha Fechada com <i>PLL</i>	76

4	RESULTADOS E DISCUSSÕES	80
4.1	Resultado das Simulações	80
4.1.1	Resultado das Simulações Sem Carga	80
4.1.2	Resultado das Simulações Com Carga	82
4.1.3	Resultado das Simulações com dual-PLL	84
4.2	Resultado dos Experimentos	86
4.2.1	Resultado do Ensaio ao Degrau Experimental	86
4.3	Discussões	88
5	CONCLUSÕES	91
	REFERÊNCIAS BIBLIOGRÁFICAS	93
	APÊNDICES	95
	APÊNDICE A – SCRIPT SPLL COMO LPLL	96
	APÊNDICE B – SCRIPT SPLL COMO DPLL	98
	APÊNDICE C – CRITÉRIO DE POPOV APLICADO A UM PLL	102
	APÊNDICE D – CÓDIGO CONTROLADOR PID	106
	APÊNDICE E – CÓDIGO CONTROLADOR PLL	108

1 INTRODUÇÃO

1.1 Contextualização e Motivação

Atualmente, motores elétricos são partes integrantes de praticamente todas as plantas industriais, sendo construídas não menos do que 5 bilhões de unidades por ano ao redor do mundo. As principais aplicações são veículos elétricos, robôs, servo-sistemas, automação fabril, guindastes, talhadeiras, fresadoras, eletrodomésticos, computadores, scanners, impressoras, equipamentos hospitalares, cadeira de rodas, compressores de ar, furadeiras, trens, escadas rolantes, elevadores, entre muitas outras (SILVA, 2018).

Motores de indução monofásicos e motores de corrente contínua apresentam baixa eficiência e altos custos de manutenção quando comparados aos trifásicos (SATHYAN ANAND; MILIVOJEVIC; EMADI, 2009), (MODELING. . . , 2011). A baixa eficiência dos motores de indução é atribuída às perdas resistivas do rotor e ao desfasamento entre a fase das correntes do estator e da força contraeletromotriz. Já a baixa eficiência dos motores de corrente contínua é advinda de sua durabilidade que exige uma manutenção mais frequente em função dos desgastes das escovas, o que gera mais gastos. Uma alternativa a estes motores é o motor *BLDC*. Estes apresentam muitas vantagens quando comparadas a outras máquinas de corrente contínua, pois requerem menores manutenções pela eliminação da comutação mecânica realizada pelas escovas. Quando comparadas às máquinas de indução, os BLDC apresentam menos inércia no rotor, permitindo uma dinâmica mais rápida no seguimento de referência. Pode-se também mencionar que eles apresentam menores perdas devido aos ímãs permanentes, resultando em menos perdas no rotor. Outra qualidade destacável é o torque médio entregue proporcional ao tamanho do motor, que é mais alto do que nos demais motores, tornando os BLDCs ideais em aplicações nas quais o espaço e o peso são fatores críticos (TIBOR BALOGH; FEDAK; ĀUROVSKY, 2011). Em contrapartida, as desvantagens destas máquinas são o seu alto custo e a maior complexidade do controle e de sua eletrônica de potência (MODELING. . . , 2011), (TIBOR BALOGH; FEDAK; ĀUROVSKY, 2011).

Portanto, tendo ciência do grande potencial do motor BLDC e do seu relativamente recente uso, desperta-se o interesse na investigação de técnicas de controle pouco difundidas, já testadas em outras topologias de motores, como a técnica de controle por PLL, mas que ainda não foram testadas, ou pelo menos documentadas em larga escala, neste motor, que, como citado acima, possui no seu controle um de seus pontos fracos.

1.2 Descrição do Problema

Como já descrito, os motores BLDC são utilizados em uma série de aplicações. Nos últimos anos, os BLDC têm sido aplicados no controle de instrumentos de alta precisão, como no armazenamento de energia por *flywheel* e na propulsão de drones. A demanda por métodos de controle mais precisos fez com que pesquisas tenham sido feitas na busca de métodos de controle que expandam o leque de aplicações do BLDC (PLL. . . , 2014).

Atualmente os métodos de controle mais empregados nos motores BLDC são o controle por corrente de histerese e por PWM juntamente com a teoria de controle contínuo (SATHYAN ANAND; MILIVOJEVIC; EMADI, 2009). O controle por corrente de histerese é essencial para atingir o controle instantâneo do torque e de velocidade, produzindo uma resposta mais rápida quando comparada ao controle por somente PWM. Na maioria das aplicações, um compensador PI é suficiente para fornecer um bom controle de velocidade e torque. Em alguns casos, o controle por variáveis de estado pode ser necessário para se alcançar uma maior precisão com o BLDC. A teoria de controle discreto permite que alguns controladores possam ser implementados digitalmente com microcontroladores, microprocessadores ou *DSPs*. A digitalização destes controladores também adiciona complexidade ao processo (SATHYAN ANAND; MILIVOJEVIC; EMADI, 2009).

Uma alternativa menos complexa ao controle PID, segundo (BEST, 2003), (CAMPOS, 1991), e (PLL. . . , 2014) pode ser o método de controle por PLL, que é considerado, por estes trabalhos, o de mais fácil implementação e de mais alta precisão. O controle por PLL é um método capaz de realizar com alta precisão o controle de velocidade através da detecção do erro de fase entre o sinal de entrada e o sinal realimentado (PLL. . . , 2014). Cabe ainda ressaltar que o PLL é uma solução otimizada à nível de hardware quando se considera o controle de fase e frequência. Entretanto, quando se fala em descrição via software, como o SPLL (software PLL) investigado no presente trabalho, talvez tenha-se que a implementação de um controle PID venha a se tornar mais simples.

Atualmente, algumas pesquisas abordam o uso do PLL para o controle de velocidade em conjunto com outra forma de controle, mas o chaveamento entre os modos de operação tornam o sistema mais complexo, podendo deixá-lo instável (PLL. . . , 2014). Uma estratégia de controle utilizando um dual-PLL é mostrada em (PLL. . . , 2014), para o melhoramento de performance. Entretanto o controlador se torna mais complexo e o modelo para o objeto controlado necessita ser mais preciso.

1.3 Trabalhos Relacionados

Conforme (CAMPOS, 1991), uma estrutura dual-PLL foi utilizada com sucesso no controle de um motor de corrente contínua, no qual demonstrou-se por simulação e de forma experimental uma melhor precisão do controle de velocidade quando utilizado o controle por PLL e um tempo de acomodação menor quando utilizado um controle proporcional. Ainda, foi mostrado que uma mescla dos dois métodos resultou em um controle mais rápido e preciso.

Em (PLL..., 2014), um PLL foi descrito digitalmente em um FPGA, através de 10 estados, para ser utilizado na detecção de fase e frequência de um motor BLDC, realizando o seu controle de velocidade. Neste trabalho foi mostrado que esta alternativa é altamente precisa e capaz de entregar alta performance dinâmica. Ainda, os experimentos mostraram que a velocidade do BLDC pôde ser controlada com um erro menor que 0,1%, trabalhando-se com velocidades entre 2500 RPM e 4000 RPM. Também pôde-se verificar que a saída de velocidade foi capaz de seguir uma referência triangular de forma efetiva.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Modelo da Planta

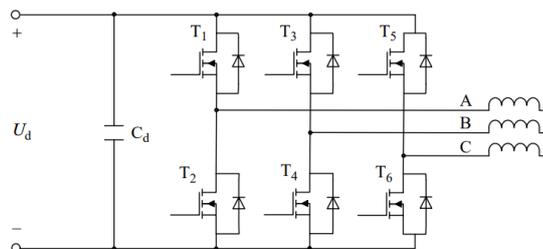
Quando se deseja controlar a velocidade de rotação e a direção do rotor de um motor BLDC, se faz necessário conhecer como o conjunto de estruturas que compõem o motor e os seus acessórios são comumente aplicadas e a ciência de como estas partes operam em conjunto, para então modificá-las e alterá-las conforme o desejado. Nesta seção são mostrados os equacionamentos destas partes.

2.1.1 Modelo do Acionamento

Na maioria das aplicações com motores BLDC, usam-se conexões em estrela nos enrolamentos, geralmente sendo controlados por chaveamentos de circuitos de controle de ponte trifásica. Os circuitos de acionamento mais empregados nestes casos são os conversores de potência CC-CA, também conhecidos por inversores. Os inversores podem comutar as chaves em diversos modos e configurações, sendo os modos meia-ponte e ponte completa os mais utilizados. Há ainda os modos de operação que utilizam pontes H como circuito de acionamento, normalmente operando no modo quatro-chaves.

Na Figura 1, é mostrado o diagrama esquemático do circuito de acionamento para o modo de operação ponte-completa, e o circuito de enrolamento do motor BLDC, sendo esta a configuração adotada neste trabalho. No diagrama, as chaves de potência T_1 , T_2 , T_3 , T_4 , T_5 e T_6 são utilizadas para permitir a passagem da corrente nos enrolamentos de acordo com a lógica produzida pelos sinais dos sensores de posição. Os principais modos de condução utilizados são os modos de condução para duas e três fases.

Figura 1 – Esquemático para um acionamento com ponte completa

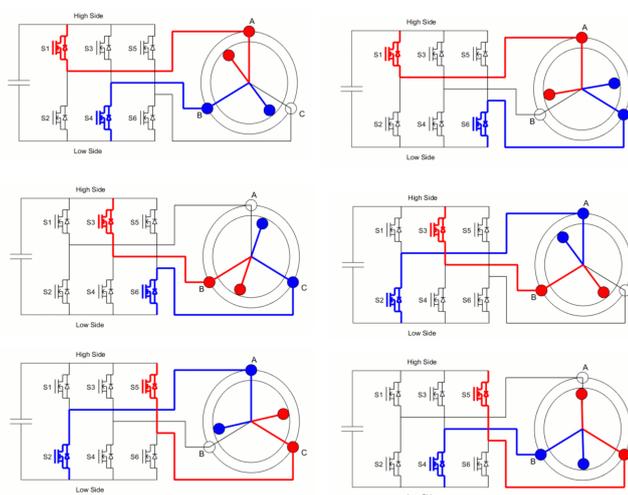


Fonte: retirado de (CHANG-LIANG, 2012)

O princípio de operação do modo de duas-fases, aplicada no inversor utilizado neste trabalho, está na condução de dois dos enrolamentos ao mesmo tempo e a suspensão de um terceiro. A ordem e o instante de condução são determinados pela informação da posição do motor gerada pelo sensor. Ainda, a ponte do motor é comutada uma vez a cada 60° de rotação do ângulo elétrico, quando o sensor muda de posição. Portanto, há seis estados nos quais dois enrolamentos diferentes estão conduzindo. O tempo de condução contínua em cada enrolamento é equivalente a 120° graus elétricos.

A Figura 2 mostra os seis estados, neles as ramificações do circuito em vermelho estão conectadas ao potencial positivo e as ramificações em azul conectadas a referência.

Figura 2 – Esquemático do motor com acionamento em seis estados de condução



Fonte: modificado de <https://au.mathworks.com/campaigns/offers/next/introduction-to-brushless-dc-motor-control/how-three-phase-inverters-work.html>

Quando alguma chave da parte superior está conduzindo, uma corrente flui da fonte para os enrolamentos conectados a ela, induzindo um torque. Similarmente, um torque é produzido pelo enrolamento conectado ao referencial. A soma desses torques geram um torque resultante que rotaciona 60° em ângulo elétrico a cada período de comutação. Por essa razão, os *ripples* de torque são menores neste modo de operação em comparação com modos de meia-ponte já que a direção do torque muda seis vezes em um único ciclo.

No modo de condução de três fases, os três enrolamentos estão sempre energizados, mudando somente a direção do fluxo das correntes. O circuito de acionamento para este modo de condução é o mesmo utilizado no modo de duas fases, mostrado na Figura 2, sendo as únicas diferenças entre estes dois métodos, a ordem de condução e o tempo, que para este modo equivale a 180° . O modo de condução de três fases pode também melhorar a utilização dos enrolamentos, assim como reduzir as ondulações de torque quando

comparado com o modo de duas fases. Entretanto, deve-se notar que o modo de condução de três fases pode possibilitar que uma chave superior e uma inferior estejam conduzindo ao mesmo tempo, exigindo uma maior precisão do acionamento. Outro fator a se mencionar é que neste modo condução o modelo matemático do BLDC se torna mais complexo, exigindo a utilização de transformadas de Clark e Park para a linearização do modelo, assim como também necessita de sensores adicionais (KRAUSE PAUL C.; WASYNCZUK; SUDHOFF, 2002), acabando por ser menos utilizados em aplicações mais compactas, como no caso de BLDCs atuando na propulsão de drones. Nestes casos, o modo de duas-fases permite que o modelo do inversor possa ser considerado um simples ganho que relaciona o sinal PWM de entrada e a tensão média de saída.

2.1.2 Modelo do Motor

Dá mesma forma que feito em (CHANG-LIANG, 2012), apresenta-se um modelo matemático nesta subseção para descrever o comportamento de um motor BLDC com a seguinte estrutura: estator conectado em estrela, com enrolamento concentrado, e rotor interno com pólos não salientes. Os sensores considerados estão separados simetricamente por 120° . Algumas aproximações também são utilizadas a fim de simplificar o modelo, são estas:

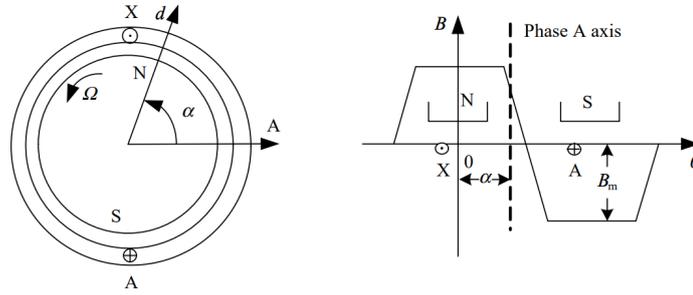
- Ignorar a saturação do núcleo, assim como as perdas por correntes parasitas e histerese;
- Ignorar a reatância de armadura, e considerar a distribuição do campo magnético através do entreferro de ar como sendo uma onda trapezoidal com um pico reto de período equivalente a 120° elétricos;
- Considerar que os condutores são distribuídos com simetria perfeita na armadura;
- As chaves de potência e os diodos de roda livre do inversor são consideradas ideais.

No modelo adotado, considera-se a expressão (2.1) para descrever o fluxo magnético dos ímãs permanentes que passam no entreferro de ar. Com Ψ_{pm} sendo a magnitude do fluxo e $N\phi_{pm}$ a densidade de fluxo concatenado pelo número de espiras. O ângulo α representa a posição do rotor.

$$\Psi_{pm}(\alpha) = N\phi_{pm}(\alpha) \quad (2.1)$$

A Figura 3 (a), à esquerda, mostra uma situação na qual o rotor gira no sentido anti-horário com o eixo direto d se movendo com ângulo α em relação ao eixo do enrolamento AX , e à direita, na Figura 3 (b), mostra-se o campo magnético em função do ângulo do rotor para a situação em que o pólo norte do ímã está alinhado com X , e o pólo sul com A .

Figura 3 – (a) Posição do rotor (b) Distribuição de fluxo



Fonte: retirado de (CHANG-LIANG, 2012)

Sabendo-se que o fluxo magnético da fase A em função da posição do rotor α é dado pela equação (2.2), com ϕ_{pm} sendo igual ao fluxo magnético da fase A, $B(\theta)$ a densidade de fluxo magnético no entreferro de ar e S a área dos enrolamentos.

$$\phi_{pm}(\alpha) = \int_{-\frac{\pi}{2}-\alpha}^{-\frac{\pi}{2}+\alpha} B(\theta)Sd\theta \quad (2.2)$$

Ainda, sabendo-se que as quedas de tensão no enrolamento da fase A são compostas pelas perdas resistivas e os efeitos dos campos eletromagnéticos, chega-se na expressão (2.3), na qual $u_A(t)$ é a tensão de fase no enrolamento A, $Ri_A(t)$ é a perda ôhmica na fase A, $L_Ai_A(t)$ é a auto-indutância da fase A e $M_{AX}i_X(t)$ é a indutância mútua da fase X na fase A.

$$u_A(t) = Ri_A(t) + \frac{\partial}{\partial t}(L_Ai_A(t) + M_{AB}i_B(t) + M_{AC}i_C(t) + \Psi_{pm}(t)) \quad (2.3)$$

A partir daí, pode-se então, realizando a substituição da equação (2.2) dentro da equação (2.3), e utilizando-se a lei de Faraday para saber que a variação do fluxo magnético gera uma força contra eletromotriz, concluir que, no caso da fase A, esta será dependente da força contra-eletromotriz $e_A(t)$, chegando-se na expressão (2.4), que pode ser simplificada como em (2.5).

$$u_A(t) = Ri_A(t) + \frac{\partial}{\partial t}(L_Ai_A(t) + M_{AB}i_B(t) + M_{AC}i_C(t)) + \frac{\partial}{\partial t}N \int_{-\frac{\pi}{2}-\alpha}^{-\frac{\pi}{2}+\alpha} B(\theta)Sd\theta \quad (2.4)$$

$$u_A(t) = Ri_A(t) + \frac{\partial}{\partial t}(L_Ai_A(t) + M_{AB}i_B(t) + M_{AC}i_C(t)) + e_A(t) \quad (2.5)$$

Sabendo-se que a permeabilidade magnética de um rotor com pólos salientes difere dependendo se esta se encontra na direção de eixo direto ou em quadratura, pode-se

concluir que a auto-indutância e a indutância mútua dos enrolamentos mudam com a posição do rotor (KRAUSE PAUL C.; WASYNCZUK; SUDHOFF, 2002). Sabendo-se que para um rotor com pólos não salientes o fluxo é isotrópico (em todas as direções), pode-se concluir que a permeabilidade do circuito magnético não é afetada pela posição do rotor, fazendo com que a auto-indutância não varie com o tempo. Como na maioria das aplicações são usados BLDCs com rotor de ímãs permanentes montados na superfície, as indutâncias dos enrolamentos não são levadas em consideração. Utilizando-se destas informações, aproximações podem ser feitas, como, por exemplo, considerar as indutâncias iguais e invariantes no tempo para todas as fases, podendo-se reescrever a equação (2.5) na forma mostrada em (2.6).

$$u_A(t) = Ri_A(t) + L \frac{\partial i_A(t)}{\partial t} + M \frac{\partial i_B(t)}{\partial t} + M \frac{\partial i_C(t)}{\partial t} + e_A(t) \quad (2.6)$$

Para que se possa relacionar a tensão $u_A(t)$ com a velocidade do motor $\omega(t)$, pode-se analisar a expressão da força contra-eletromotriz $e_A(t)$ conforme a equação (2.7), que, ao se aplicar a derivada temporal, transforma-se na expressão dada por (2.8).

$$e_A(\theta) = \frac{\partial}{\partial t} N \int_{-\frac{\pi}{2}-\alpha}^{-\frac{\pi}{2}+\alpha} B(\theta) S d\theta \quad (2.7)$$

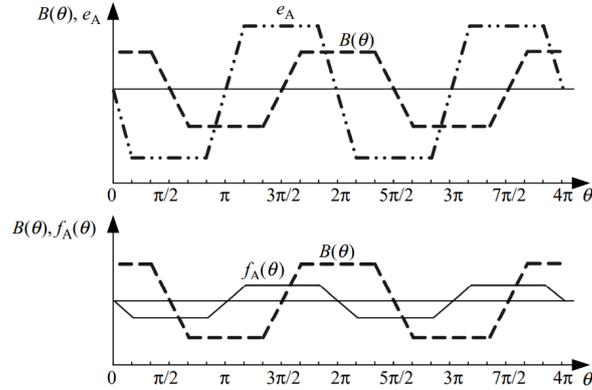
$$e_A(\theta) = NS\omega(t) [B(\theta + \frac{\pi}{2}) - B(\theta - \frac{\pi}{2})] \quad (2.8)$$

De acordo com a Figura 3 (a), a densidade de fluxo magnético no entreferro de ar é uma função periódica com um período de 2π , na qual $B(\alpha + \pi) = -B(\alpha)$, portanto a equação (2.8) pode ser reescrita como (2.9), na qual B_m é a amplitude da densidade de fluxo magnético e $f_A(\theta)$ é uma função que varia com a posição do rotor.

$$e_A(\theta) = 2NS\omega(t) B_m f_A(\theta) \quad (2.9)$$

Através da expressão (2.9), é possível constatar que $f_A(\theta)$ tem uma forma trapezoidal, assim como a densidade de fluxo magnético. Também é possível notar que ela tem uma amplitude que varia entre -1 a 1. Na Figura 4 é mostrada essa relação e o atraso de 90° da força contra-eletromotriz em relação a densidade de fluxo magnético. Ainda, é importante constatar que a força contra-eletromotriz é igual nas três fases, porém com um desfasamento de 120° . Por fim, é importante saber que a forma trapezoidal da força contra-eletromotriz é a responsável pelo motor BLDC ser considerado um motor de corrente contínua.

Das análises realizadas, pode-se concluir que os enrolamentos têm um desfasamento de 120° por fase, evidenciando que para um enrolamento do tipo estrela, a soma das

Figura 4 – Relação entre $B(\theta)$, $e_A(\theta)$ e $f_A(\theta)$


Fonte: retirado de (CHANG-LIANG, 2012)

correntes dos enrolamentos do estator será sempre igual a zero. Utilizando-se desta constatação, pode-se simplificar a equação (2.6) e reescrevê-la como em (2.10).

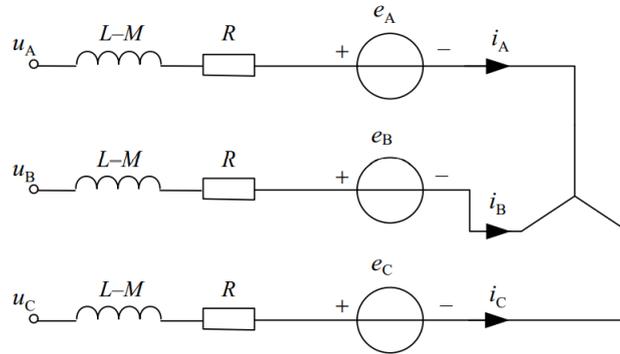
$$u_A(t) = Ri_A(t) + (L - M)\frac{\partial i_A(t)}{\partial t} + e_A(t) \quad (2.10)$$

Na maioria das aplicações dos motores BLDC, os enrolamentos do estator estão conectados em estrela sem a presença de um condutor neutro, fazendo com que a tensão de fase seja de difícil acesso para medição. Então, torna-se conveniente escrever um modelo matemático do motor em função das tensões de linha ao invés das tensões de fase, como foi feito até esta parte. Sabendo-se que as tensões de linha são aproximadamente iguais às tensões nos barramentos CC quando os transistores de potência estão ligados, um modelo matemático com as tensões de linha pode ser descrito como na equação (2.11). O modelo do circuito equivalente é apresentado na Figura 5.

$$\begin{bmatrix} u_{AB}(t) \\ u_{BC}(t) \\ u_{CA}(t) \end{bmatrix} = \begin{bmatrix} R & -R & 0 \\ 0 & R & -R \\ -R & 0 & R \end{bmatrix} \begin{bmatrix} i_A(t) \\ i_B(t) \\ i_C(t) \end{bmatrix} + \begin{bmatrix} (L - M) & (M - L) & 0 \\ 0 & (L - M) & (M - L) \\ (M - L) & 0 & (L - M) \end{bmatrix} \begin{bmatrix} \frac{\partial i_A(t)}{\partial t} \\ \frac{\partial i_B(t)}{\partial t} \\ \frac{\partial i_C(t)}{\partial t} \end{bmatrix} + \begin{bmatrix} (e_A(t) - e_B(t)) \\ (e_B(t) - e_C(t)) \\ (e_C(t) - e_A(t)) \end{bmatrix} \quad (2.11)$$

A partir de agora uma análise do torque em motor BLDC é feita para unir as características mecânicas às elétricas. A análise do torque e potência de um motor BLDC

Figura 5 – Esquemático para um modelo simplificado do motor BLDC



Fonte: retirado de (CHANG-LIANG, 2012)

pode ser feita pela perspectiva da transferência de energia. Quando o motor está operando, a potência da fonte é absorvida, e, embora alguma parte seja perdida no cobre e no entreferro, a maior parte é transferida através do entreferro de ar para o rotor. A potência transferida para o rotor, a qual é chamada de potência eletromagnética, deve ser igual a soma do produto das correntes pelas forças contra-eletromotrizes das três fases. Uma equação que expressa essa relação é dada em (2.12).

$$P_e(t) = e_A(t)i_A(t) + e_B(t)i_B(t) + e_C(t)i_C(t) \quad (2.12)$$

Ignorando-se as perdas mecânicas e parasitas, a potência eletromagnética é totalmente convertida em potência cinética. Esta variação de energia é dada pelo produto do torque pela velocidade do rotor. Igualando-se as duas formas de potências, chega-se na expressão (2.13).

$$P_e(t) = T_e(t)\omega(t) \quad (2.13)$$

Igualando-se a equação (2.12) e (2.13), cancelando-se a velocidade do rotor em ambos os lados da igualdade e substituindo-se a força contra-eletromotriz pela expressão (2.9), chega-se na expressão (2.14) para o torque elétrico, onde p é o número de par de pólos do rotor.

$$T_e(t) = pNSB_m[f_A(\theta)i_A(t) + f_B(\theta)i_B(t) + f_C(\theta)i_C(t)] \quad (2.14)$$

Quando o motor BLDC opera com o modo de comutação de 120° e se considera o regime permanente, em qualquer momento, pode-se concluir que as duas correntes dos

enrolamentos ativos terão a mesma amplitude e direção oposta, enquanto que a corrente do enrolamento desativado será nula. É possível, então, notar que o pico achatado da função $f_X(\theta)$ das duas fases que estão conduzindo, são opostos entre si nos diferentes enrolamentos, com valores de 1 e -1, possibilitando simplificar a equação (2.14) reescrevendo-a como na equação (2.15), ou como em (2.16), simplificando ainda mais, onde $K_T i(t)$ é o produto da corrente estacionária amplificado por uma constante, mostrando uma relação de proporcionalidade entre a corrente e o torque elétrico.

$$T_e(t) = 2pNSB_m i(t) \quad (2.15)$$

$$T_e(t) = K_T i(t) \quad (2.16)$$

Com o objetivo de construir um modelo matemático completo do sistema eletromecânico, deve-se incluir também a dinâmica de movimento do rotor, que pode ser equacionada como em (2.17), onde $T_L(t)$ é o torque da carga, J é o momento de inércia do rotor e B_ν é o coeficiente de atrito viscoso.

$$T_e(t) - T_L(t) = J \frac{\partial \omega(t)}{\partial t} + B_\nu \omega(t) \quad (2.17)$$

Tendo o conhecimento das equações que regem o sistema, e sendo este linear e invariante no tempo, pode-se descrever o seu modelo na forma de função de transferência.

A função de transferência é um conceito dos mais importantes na teoria de controle, pois ela é base para os projetos de controladores, principalmente aqueles que utilizam os métodos de lugar das raízes e resposta em frequência.

Para realizar a análise da função de transferência do BLDC, primeiramente, pode-se reescrever as equações diferenciais obtidas de forma a simplificá-las, como é mostrado na expressão (2.18), na qual: $u_d(t)$ é a tensão do barramento CC; $i(t)$ é a corrente que flui pelos enrolamentos; r_a é a resistência equivalente dos dois enrolamentos ou o dobro da resistências de um enrolamento com valor $2R$; L_a é a indutância equivalente dos enrolamentos ou o dobro da diferença entre auto-indutância e a indutância mútua $2(L - M)$; k_e é o coeficiente linear da força contra-eletromotriz $4NSB_m$; e $\omega(t)$ é a velocidade angular do rotor.

$$u_d(t) = r_a i(t) + L_a \frac{\partial i(t)}{\partial t} + k_e \omega(t) \quad (2.18)$$

Como a corrente é proporcional ao torque elétrico, ela pode ser relacionada com a dinâmica mecânica fazendo com que as equações elétricas possam ser combinadas. Portanto, substituindo-se a equação (2.15) dentro da equação (2.17), chega-se na expressão 2.19.

$$K_T i(t) - T_L(t) = J \frac{\partial \omega}{\partial t} + B_\nu \omega(t) \quad (2.19)$$

Em um segundo passo, pode-se considerar que não há carga no eixo do motor, $T_L(t) = 0$, desta forma, obtendo-se o modelo do BLDC sem distúrbios. Portanto a corrente dos enrolamentos pode ser dada por (2.20).

$$i(t) = \frac{J}{K_T} \frac{\partial \omega(t)}{\partial t} + \frac{B_\nu}{K_T} \omega(t) \quad (2.20)$$

Substituindo-se a corrente dada na equação (2.20) dentro da equação (2.18), tem-se (2.21).

$$u_d(t) = r_a \left(\frac{J}{K_T} \frac{\partial \omega(t)}{\partial t} + \frac{B_\nu}{K_T} \omega(t) \right) + L_a \frac{\partial \left(\frac{J}{K_T} \frac{\partial \omega(t)}{\partial t} + \frac{B_\nu}{K_T} \omega(t) \right)}{\partial t} + k_e \omega(t) \quad (2.21)$$

Reagrupando os termos para que se tenha uma equação diferencial ordinária com relação a ω , chega-se na equação (2.22).

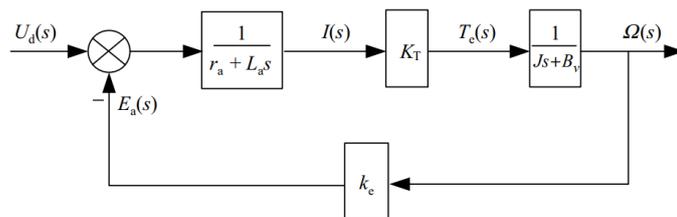
$$u_d(t) = \frac{J L_a}{K_T} \frac{\partial^2 \omega(t)}{\partial t^2} + \frac{r_a J + L_a B_\nu}{K_T} \frac{\partial \omega(t)}{\partial t} + \frac{r_a B_\nu + k_e K_T}{K_T} \omega(t) \quad (2.22)$$

Por fim, aplicando-se a transformada de Laplace, com o intuito de resolver o sistema no domínio da frequência, chega-se na expressão (2.23), na qual $G(s) = \frac{\Omega(s)}{U_d(s)}$ é a função de transferência do BLDC, que relaciona a tensão de entrada com a velocidade angular de saída.

$$G(s) = \frac{K_T}{L_a J s^2 + (r_a J + L_a B_\nu) s + (r_a B_\nu + k_e K_T)} \quad (2.23)$$

O sistema em diagrama de blocos equivalente a função de transferência do motor BLDC é mostrado na Figura 6.

Figura 6 – Função de transferência do motor BLDC



Fonte: retirado de (CHANG-LIANG, 2012)

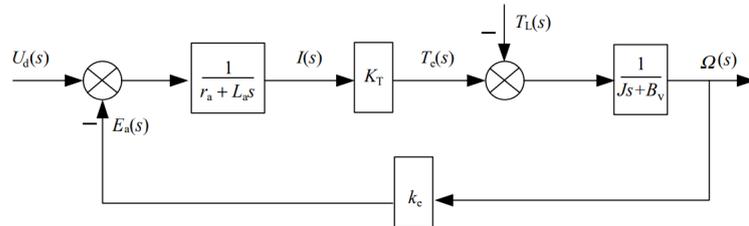
2.1.3 Modelo do Gerador Acoplado

Neste trabalho, o eixo de um gerador é acoplado ao eixo do motor BLDC a fim de simular perturbações de carga ao torque do motor. Como o gerador escolhido é originalmente um motor de corrente contínua de ímãs permanentes, quando este não estiver conectado à carga, ele não apresenta resistência ao torque do motor além do momento de inércia que o acoplamento dos eixos adiciona ao novo rotor, pois, devido a ausência dos enrolamentos de campo, não há caminho para a força contra-eletromotriz gerar uma corrente. Quando a carga é adicionada à saída do gerador, espera-se que esta, em regime permanente, apresente torque aproximadamente constante e contrário ao torque elétrico do motor. A dinâmica de um torque de carga aplicado ao eixo do motor no domínio da frequência é dada pela equação (2.24).

$$\Omega(s) = \left[-k_e \frac{1}{r_a L_a s} K_T T_L(s) - T_L(s) \right] \quad (2.24)$$

A carga aplicada ao sistema pode ser vista como uma entrada do sistema, como mostrado na Figura 7. Como se trata de um sistema linear, o princípio da superposição poderá ser aplicado a ele. Então, a saída do sistema será igual a soma das saídas geradas por cada entrada $U_d(s)$ e $T_L(s)$.

Figura 7 – Função de transferência do motor BLDC com torque de carga externo aplicado



Fonte: retirado de (CHANG-LIANG, 2012)

Quando faz-se $U_d(s) = 0$, no sistema da Figura 7, pode-se analisar a dinâmica para a entrada de carga. Desta forma a função de transferência que relaciona a entrada de torque de carga com a velocidade angular do motor, $G_L(s) = \frac{\Omega(s)}{T_L(s)}$, pode ser dada pela equação (2.25).

$$G_L(s) = -\frac{r_a + L_a s}{L_a J s^2 + (r_a J + L_a B_v) s + (r_a B_v + k_e K_T)} \quad (2.25)$$

2.2 Identificação do Modelo

Para atingir os objetivos de controle é importante se obter um modelo que seja fiel ao comportamento real do sistema. Nesta seção são discutidos dois métodos de identificação experimental do processo, utilizados neste trabalho, primeiro com resposta ao degrau, para a extração do tempo de acomodação, e, após, com a utilização de uma sequência binária pseudo-aleatória para se obter um modelo mais completo.

2.2.1 Resposta ao Degrau

Um degrau é equivalente a um sinal de entrada constante a partir de um determinado instante, o qual pode ser de diferentes naturezas, como um sinal de tensão, de velocidade ou de fase. Na resposta ao degrau, geralmente se coloca uma entrada na planta, e observa-se a saída, que tende a se acomodar em um valor constante após um regime transitório.

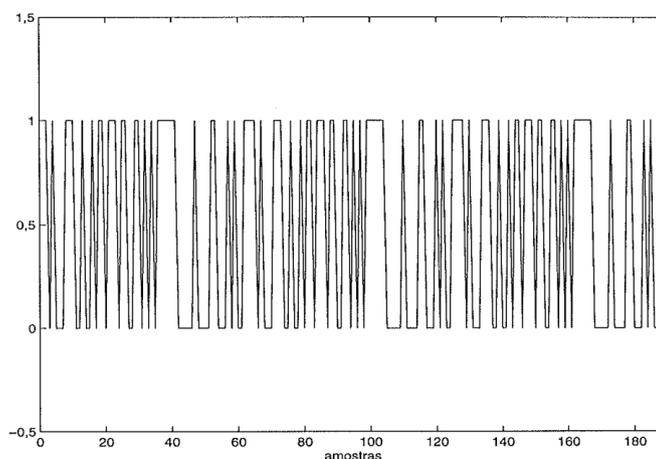
Segundo (OGATA, 2011), realizando-se ensaios ao degrau, pode-se identificar alguns parâmetros do sistema através da análise da resposta temporal. Se o sistema for de primeira ordem, pode-se identificar a constante temporal e o erro em regime permanente. Para ordens maiores, pode-se fazer aproximações que descrevam o sistema que mais se aproxima daquele comportamento, devendo-se definir os números de pólos e zeros previamente. Os experimentos ao degrau são mais utilizados em sistemas lineares de primeira ordem devido à incapacidade de identificar não-linearidades causadas por variações de frequência.

2.2.2 PRBS

Uma das formas de identificação de modelos lineares mais utilizadas é a aplicação de entradas binárias pseudo-aleatórias, ou PRBS. Em motores industriais, esta metodologia é largamente empregada. Por exemplo, em (SCHREIBER; ISERMANN, 2006), este procedimento de identificação foi utilizado para obter o modelo de um motor à combustão.

Segundo (AGUIRRE, 2007), os sinais PRBS são pulsos retangulares com duas amplitudes, por exemplo, $-V$ e $+V$ e largura variada aleatoriamente dentro de períodos bem definidos em instantes de tempo $t = 0$, $t = t_b$, $t = 2t_b$, ..., com esta largura devendo respeitar as faixas de atuação do sistema de tal forma que a entrada possa ser grande o suficiente para não ser considerada um ruído, mas não tão grande de modo que haja como uma entrada tipo degrau. Um exemplo de sinal PRBS é mostrado na Figura 8.

A utilização do método de identificação PRBS traz maior robustez ao modelo do sistema por verificar o seu comportamento para diferentes frequências em um único experimento, tornando o método mais indicado para sistemas de ordens mais elevadas. Para se realizar um ensaio com entrada PRBS, deve-se garantir que algumas condições

Figura 8 – Sinal *PRBS* com 200 amostras

Fonte: retirado de (AGUIRRE, 2007)

sejam atendidas, como por exemplo, a quantidade de amostras do sinal deve seguir a equação (2.26),

$$N = 2^n - 1 \quad (2.26)$$

onde N é o número de amostras, e n é a quantidade de bits dos *shift-registers* que são utilizados para a geração do sinal. A equação (2.26), segundo (AGUIRRE, 2007), apresenta a quantidade mínima de bits dos *shift-registers* para a geração de números aleatórios. O limite apresentado vem do fato de o sinal PRBS ser determinístico, tendo os seus valores repetidos a cada N amostras, necessitando portanto de um maior número de bits para uma maior quantidade de amostras, justificando o seu caráter pseudo aleatório.

Em (AGUIRRE, 2007), estabelecem-se alguns critérios para a escolha de V , n e t_b . A escolha de V se dá dentro da faixa de máxima de excursão do sinal de entrada do sistema, levando-se em conta também que este valor deve estar dentro da região linear. Já o número de bits n não deve ser menor que o tempo de acomodação do sistema, sob o risco de deixar o sinal “menos aleatório”. O intervalo entre bits t_b deve ser compatível com a menor constante de tempo de interesse. Se t_b for muito grande, o sistema o interpretará como uma entrada ao degrau, considerada uma identificação pobre em termos de análise do espectro. Se t_b for muito pequeno o sistema não terá tempo o suficiente para chegar na próxima transição. Resultados empíricos mostram que um t_b dentro do intervalo mostrado em (2.27) gera bons resultados, onde τ_{min} é a constante de tempo do sistema.

$$\tau_{min}/10 < t_b < \tau_{min}/3 \quad (2.27)$$

2.3 Modelo do Controlador

O controle de velocidade de motores BLDCs pode envolver uma grande quantidade de técnicas, podendo estas serem compostas por métodos PID, *Fuzzy*, *machine learning*, PLL e etc. Neste trabalho foram explorados os métodos PID e PLL.

2.3.1 Controlador PID

A estratégia de controle por PID tem sido a forma de controle mais implementada nos últimos 70 anos devido a sua simplicidade, robustez, praticidade e fácil modificação de seus parâmetros (CHANG-LIANG, 2012).

O funcionamento do PID consiste em aplicar um sinal de diferença entre a referência e o sinal de saída da planta, um sinal de erro $e(t)$, a um controlador que aplicar-lhe-á uma ação proporcional, outra integral e/ou derivativa, modificando-o e entregando um sinal de controle $u(t)$ à planta, de tal forma a mantê-la com a saída tão próxima da referência quanto desejado. A equação (2.28) mostra a relação entre o sinal de erro e o sinal de controle, $C(s) = \frac{U(s)}{E(s)}$, no domínio de Laplace para uma configuração PID paralela, onde K_p é o ganho proporcional, T_I é a constante de integração, T_D é a constante de diferenciação e P é o polo de altas frequências.

$$C(s) = K_p \left(1 + \frac{1}{T_I s} + \frac{T_D P s}{s + P} \right) \quad (2.28)$$

Nem sempre o controlador PID terá todas as ações incluídas, é usual encontrar controladores com somente ação proporcional e integral, no controle de motores BLDCs, evitando a amplificação de distúrbios e ruídos em alta frequências ocasionados pela ação derivativa.

Existem diferentes formas de se implementar um controlador PID, sendo as mais comuns a implementação analógica e a digital. Quando se objetiva melhorar a executabilidade do controlador, atualmente, opta-se pela implementação digital. Para executar este tipo de implementação, deve-se realizar o projeto do controlador para então se obter os seus parâmetros e após obter o seu equivalente discreto, possibilitando assim descrevê-lo na forma de equações de diferenças. A discretização do controlador pode ser feita através de equivalência de polos e zeros, por equivalência ZOH ou por integração numérica. Neste trabalho, utiliza-se dois dos métodos de integração numérica mais conhecidos, em situações distintas, sendo estas as técnicas de integração numérica por Backward-Euler e por Forward-Euler. As equações de recorrência para um controlador como o dado em (2.28), adotando-se os métodos de integração numérica de Backward-Euler e Forward-Euler ficam, respectivamente, como (2.29) e (2.30), onde $i(k)$ é a discretização da ação integral e $d(k)$ é a discretização da ação derivativa.

$$u(k) = i(k-1) + \frac{1}{1+PT}d(k-1) + K_p[e(k) + \frac{T}{T_i}e(k) + \frac{T_D P}{1+PT}(e(k) - e(k-1))] \quad (2.29)$$

$$u(k) = i(k-1) + (1-PT)d(k-1) + K_p[e(k) + \frac{T}{T_i}e(k-1) + T_D P(e(k) - e(k-1))] \quad (2.30)$$

Os mesmos métodos para se obter os parâmetros do controlador PID contínuo podem ser utilizados para se obter os parâmetros do PID digital. Na prática, a obtenção desse parâmetros pode ser realizada utilizando-se de diversos métodos, como através de projetos de alocação das raízes, métodos de projeto através de modelos de frequência, métodos que utilizam *machine learning*, métodos empíricos como tentativa e erro, métodos tabelados como Ziegler-Nichlos, entre outros.

Para um bom desempenho do sistema de controle é importante realizar a escolha adequada do período de amostragem. Conforme o teorema de amostragem de Shannon, a frequência de amostragem deve ser maior ou igual a duas vezes a frequência do sinal amostrado, para que o sinal amostrado se aproxime do sinal contínuo. Garantindo essas condições, o sistema de controle discreto deverá se aproximar do sistema contínuo.

Em especial, em sistemas de controle para velocidade de motores, o controlador deve ser projetado para conseguir seguir a referência de velocidade rapidamente. Então, o período de amostragem deve ser o menor o possível. Na prática, para escolher o tempo de amostragem, deve-se levar em conta a frequência de operação do sistema computacional, a frequência de chaveamento das chaves de potência, o atraso de leitura dos sensores, e as restrições de conversão dos ADC e DAC.

2.3.2 Controlador SPLL

Um sistema PLL, do ponto de vista do sinal de fase, pode ser considerado como um sistema realimentado onde o sinal de saída do bloco VCO tende a replicar o sinal de referência, tanto em fase como em frequência. Por este motivo, pode-se considerar este bloco como uma planta de um sistema de controle.

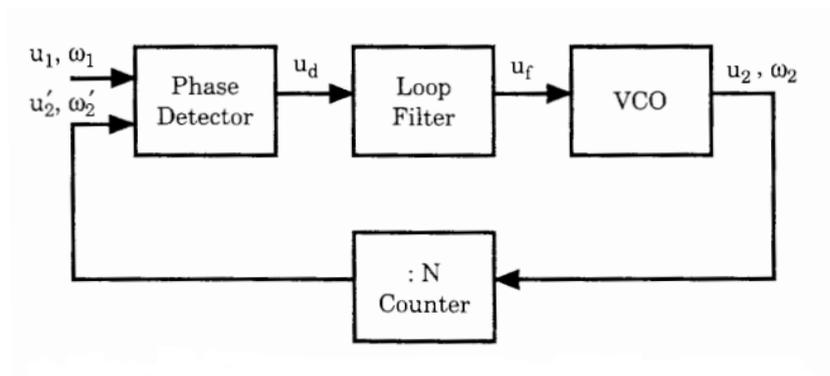
O bloco detector de fase recebe na sua entrada os sinais de referência e de saída, e gera na sua saída um sinal cuja média é proporcional à diferença dos sinais em suas entradas, portanto, este bloco pode ser interpretado como um subtrator em série com um amplificador quando a variável analisada é a fase.

Por fim, o terceiro bloco que compõe a configuração básica de um sistema PLL, é o filtro de laço. Este bloco, assim como qualquer bloco em sistemas de controle, possui uma função de transferência a qual pode ter uma gama de variações em sua forma devido aos diferentes tipos e projetos. Desta forma, pode-se traçar um paralelo entre o papel de

um filtro em sistema PLL, com o de um controlador em um sistema de controle, sendo possível aplicar-lhe os mesmos métodos de projeto de controle PID.

Um sistema PLL com os três blocos descritos acima é mostrado na Figura 9.

Figura 9 – Blocos de um PLL



Fonte: retirado de (BEST, 2003)

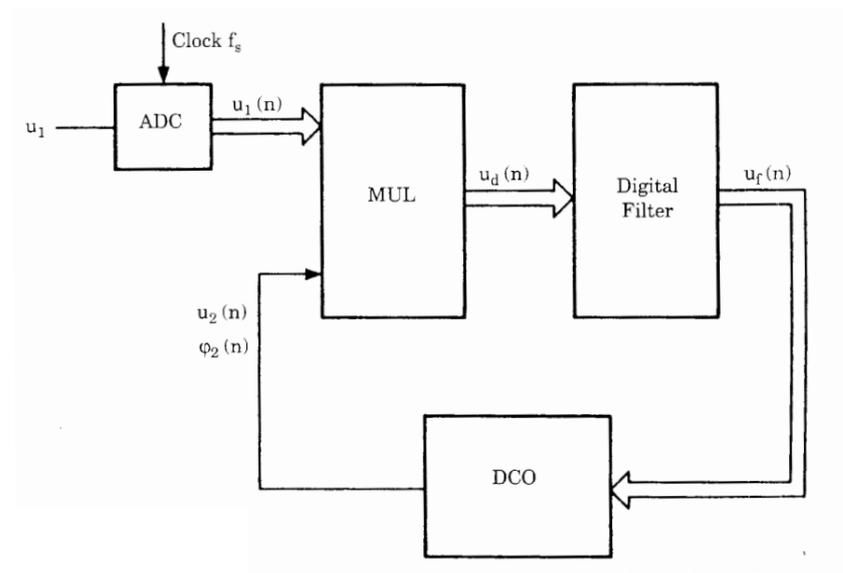
Nesta subseção são mostradas configurações de dois tipos de sistema PLL, ambas em forma de software, a fim de se compreender o seu funcionamento para que estes sistemas possam ser modificados de modo a permitir que atuem em conjunto com motores.

Ainda, é importante se ressaltar que existem diversas formas de se construir um PLL, passando por projetos de circuito inteiramente analógicos, muitas vezes chamados de LPLL (*Linear Phase-Locked Loop*); por circuitos híbridos, com componentes digitais e analógicos, chamados de DPLL (*Digital Phase-Locked Loop*); e por circuitos com somente componentes digitais, chamados de ADPLL (*All-Digital Phase-Locked Loop*). As variações de PLL podem ser implementadas em software (BEST, 2003), sendo estes chamados de SPLL.

2.3.2.1 SPLL como LPLL

O funcionamento de um SPLL, atuando como um LPLL, pode ser descrito por um diagrama de blocos como o mostrado na Figura 10. Neste diagrama pode-se ver o sinal u_1 , o qual corresponde a uma entrada de um sinal analógico periódico que serve de referência para o sistema. Seguindo a sequência do diagrama, vê-se que este sinal passa por um conversor ADC com frequência de amostragem f_s e período de amostragem T , portanto, ao se converter este sinal, gera-se a sua versão discretizada $u_1(n)$ que terá amplitude igual ao sinal de entrada nos n tempos amostrados, $t = 0, T, 2T, 3T, \dots, nT$, onde $u_1(n) = u_1(nT)$. Todos os demais sinais têm a mesma frequência de amostragem e funcionam ao mesmo tempo, ou seja, todos são amostrados com a frequência de *clock* do ADC.

Figura 10 – Diagrama de blocos de um SPLL



Fonte: retirado de (BEST, 2003)

Na Figura 10, os sinais que saem de um bloco para outro e que estão conectados por setas simples são sinais binários ou contínuos, enquanto que os sinais que estão representados por setas esparsas são enviados na forma de palavra. São utilizados outros blocos, como um multiplicador, um filtro digital e um DCO. Nesta configuração não há um divisor de frequências no laço do DCO portanto o multiplicador funciona como um detector de fase, tendo o seu sinal de saída denotado por $u_d(n)$. Vê-se, ainda, o sinal de saída do filtro digital $u_f(n)$ que reduz variações do sinal, proporcionais ao erro. O bloco DCO é uma versão digital do VCO que gera uma onda quadrada com frequência proporcional à tensão de entrada, a qual têm seus valores dados em amostras. Estes valores amostrados são denotados por $u_2(n)$. Entretanto, o DCO não calcula diretamente o valor da onda contínua $u_2(n)$, sendo este obtido do cálculo da fase deste sinal, $\phi_2(t)$. Conforme (BEST, 2003), a frequência angular do sinal de saída de um VCO é obtida da expressão (2.31), onde $\omega_2(t)$ é a frequência do sinal, ω_o é uma constante ajustável, e $K_o u_f(t)$ é o sinal de saída do filtro amplificada por um ganho.

$$\omega_2(t) = \omega_o + K_o u_f(t) \quad (2.31)$$

Desta forma, tem-se que o sinal $u_2(t)$ poderá ser descrito como na equação (2.32), onde $rect$ é uma função do tipo onda quadrada com frequência angular dada pela equação (2.31).

$$u_2(t) = rect(\omega_2(t)t) \quad (2.32)$$

Com a frequência de $u_2(t)$ definida, torna-se possível calcular a sua fase conforme a expressão (2.33).

$$\varphi_2(t) = \omega_o + K_o \int_0^t u_f(\tau) d\tau \quad (2.33)$$

Com o valor da fase obtido, deve-se utilizá-lo para calcular a saída instantânea do DCO, atribuindo-se um valor de +1 e -1 para o sinal de onda quadrada, a depender do seu desfasamento, que poderá estar num intervalo múltiplo de 2π , dividido em duas partes iguais devido a simetria do sinal. Então $u_2(t)$ deverá assumir um valor positivo para fases entre 0 a π e negativo para π a 2π .

Para um sinal discreto, considera-se um tempo de amostragem com duração entre $nT < t < (n+1)T$ e também que todos os valores se mantêm constantes durante este período. Deve-se considerar então que a variação da fase do sinal de saída do bloco DCO a cada amostra será dada pela expressão (2.34).

$$\Delta\varphi_2 = \omega_o T + K_o u_f(n) T \quad (2.34)$$

Se o valor da fase $\varphi_2(n)$, em uma amostra que ocorre com $t = nT$ é conhecida, pode-se extrapolar o seu valor para a amostra seguinte $\varphi_2(n+1)$, que ocorrerá no tempo $t = (n+1)T$ utilizando a equação de recorrência (2.35).

$$\varphi_2(n+1) = \varphi_2(n) + \omega_o T + K_o u_f(n) T \quad (2.35)$$

A expressão (2.35) poderá ser utilizada desde que se assuma que em um instante inicial, a fase é igual a zero, $\varphi_2(0) = 0$, permitindo-se desta forma extrapolar $\varphi_2(1)$ no tempo $t = 0$ e $\varphi_2(1)$ no tempo $t = 1T$, e assim por diante. Com o valor de $\varphi_2(n+1)$ conhecido, poderá se extrapolar o sinal de saída do DCO como em (2.36), onde k é um inteiro.

$$u_2(n+1) = \begin{cases} +1, & \text{se } 2k\pi \leq \varphi_2(n+1) < (2k+1)\pi \\ -1, & \text{se } (2k-1)\pi \leq \varphi_2(n+1) < 2k\pi \end{cases} \quad (2.36)$$

Os sinais de um SPLL como LPLL podem ser descritos por algoritmos que implementam as funções de cada bloco. Para o bloco multiplicador, ou detector de fase, em um instante de tempo $t = nT$, o sinal de saída $u_d(n)$ pode ser descrito como a multiplicação dos sinais de entrada por um ganho, como na expressão (2.37).

$$u_d(n) = K_d u_1(n) u_2(n) \quad (2.37)$$

Um algoritmo para descrever o comportamento do sistema LPLL pode ser implementado dentro de uma interrupção gerada periodicamente por um sinal de *clock*, sendo cada interrupção ocasionada nos instantes de tempo $t = T, 2T, 3T, \dots, nT$, para que em seguida o código do SPLL possa ser executado.

A primeira parte deste código deverá consistir em adquirir uma amostra do sinal de entrada $u_1(t)$, gerando $u_1(n)$, em seguida, a saída do multiplicador deve receber o valor do produto do sinal de entrada pelo sinal de saída do DCO, como na equação (2.37). No próximo passo, a saída do multiplicador deve ser enviada a um filtro.

O comportamento do filtro deve ser descrito por uma função de transferência que relacione a entrada com a saída, portanto, esta deverá possuir uma relação entre o sinal de saída do multiplicador $u_d(n)$ com o sinal de saída do filtro $u_f(n)$.

Conforme (BEST, 2003), o filtro que entrega o melhor desempenho em um PLL é o filtro com ação de controle do tipo PI, por este motivo a sua análise é investigada neste trabalho. A função de transferência de um filtro PI no domínio Z , $F(z) = \frac{U_f(z)}{U_d(z)}$, é obtida utilizando-se uma transformação bilinear, que no caso do sistema LPLL fica conforme a expressão (2.38),

$$F(z) = \frac{b_o + b_1 z^{-1}}{1 + a_1 z^{-1}} \quad (2.38)$$

onde os coeficientes do filtro são dados pelas seguintes expressões:

$$a_1 = -1 \quad (2.39)$$

$$b_o = \frac{T}{2\tau_1} \left(1 + \frac{1}{\tan\left(\frac{T}{2\tau_2}\right)} \right) \quad (2.40)$$

$$b_1 = \frac{T}{2\tau_1} \left(1 - \frac{1}{\tan\left(\frac{T}{2\tau_2}\right)} \right) \quad (2.41)$$

com $\frac{\tau_1}{\tau_2}$ sendo o ganho, e $\frac{1}{\tau_2}$ o zero de um filtro PI contínuo equivalente, $F(s) = \frac{1+\tau_2 s}{\tau_1 s}$.

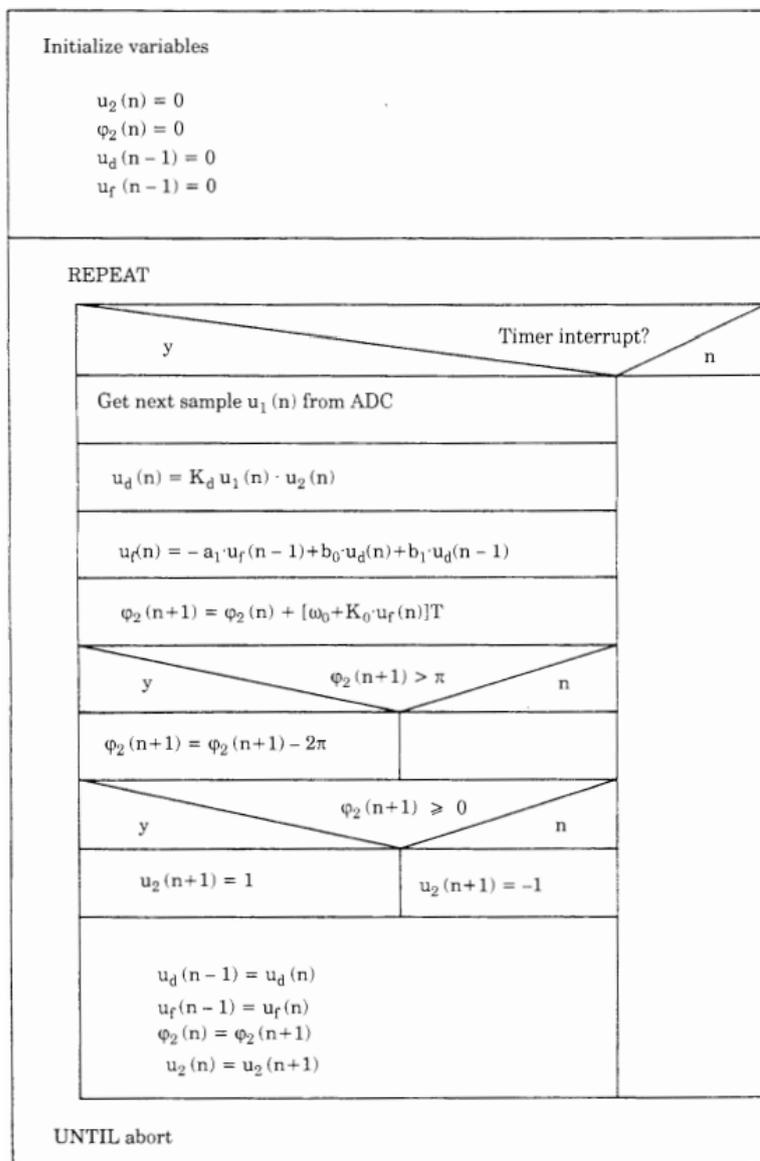
Transformando novamente a equação (2.38) para o domínio do tempo, chega-se na equação de diferenças (2.42), para a saída do filtro PI, podendo-se utilizá-la em implementações via software.

$$u_f(n) = -a_1 u_f(n) + b_o u_d(n) + b_1 u_d(n-1) \quad (2.42)$$

Um pseudocódigo que descreve o sistema do LPLL é mostrado na Figura 11.

Ainda, cabe observar que, quando o código da Figura 11 é executado, o valor de $\varphi_2(n+1)$ é incrementado a cada iteração, tornando o seu valor muito grande conforme o

Figura 11 – Pseudo-Código de um SPLL como LPLL

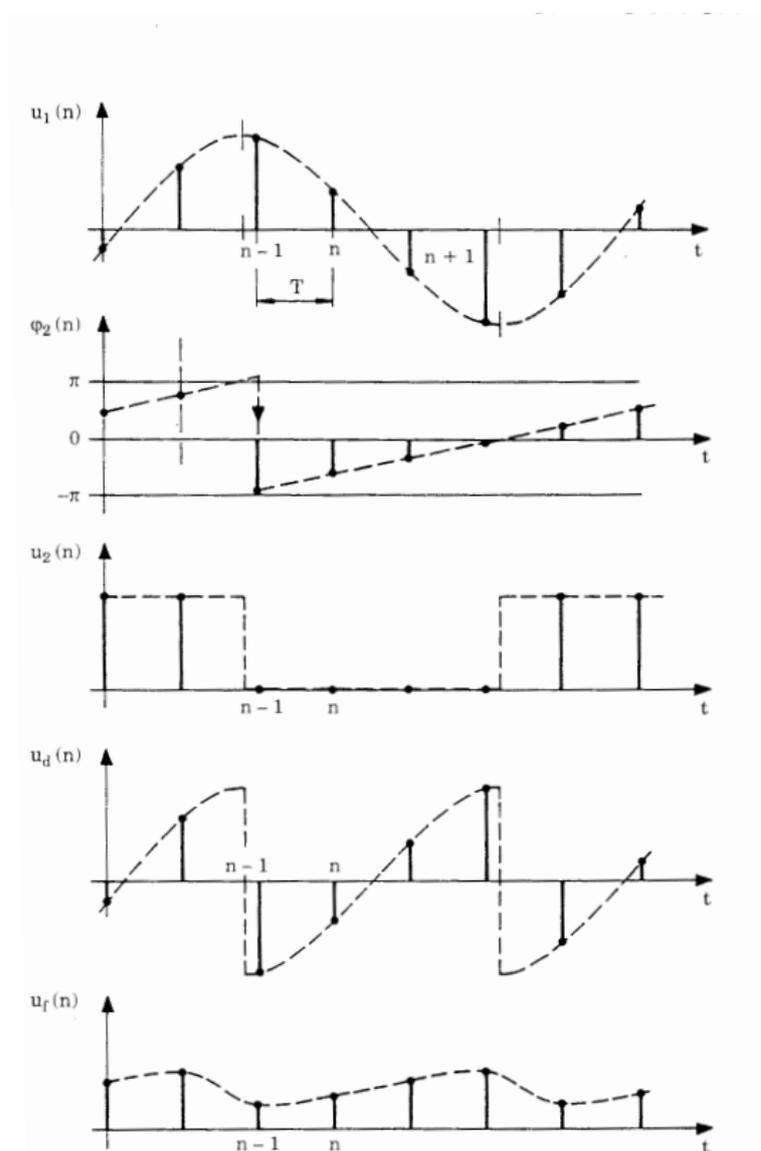


Fonte: retirado de (BEST, 2003)

tempo passa, fazendo com que este valor não seja aloável no escopo de um número *float* do processador utilizado. Para evitar o *overflow*, o valor de φ_2 é limitado no intervalo de $-\pi < \varphi_2 < \pi$. Então, quando o valor de φ_2 excede o valor de π , subtrai-se deste um valor de 2π . Desta forma, pode-se atribuir o valor de -1 para $u_2(n+1)$ quando φ_2 for menor que 0, e +1 quando φ_2 for maior que 0.

Na Figura 12 são mostrados os sinais no domínio do tempo vistos nos blocos da Figura 10, onde as linhas tracejadas representam os valores contínuos e os pontos são os valores amostrados.

Figura 12 – Sinais contínuos e amostrados de um SPLL



Fonte: retirado de (BEST, 2003)

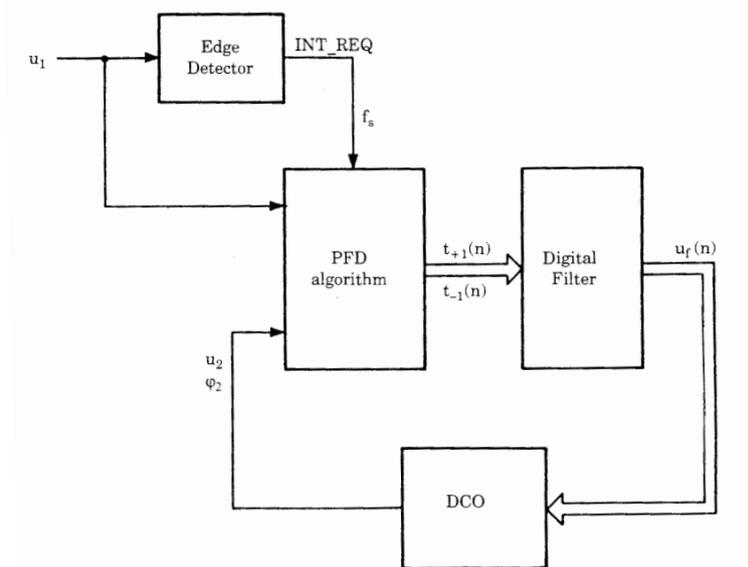
2.3.2.2 SPLL como DPLL

Conforme (BEST, 2003), torna-se mais adequada a utilização de um SPLL com comportamento de DPLL, quando o sinal u_1 é do tipo onda quadrada. Levando-se este fato em consideração, um DPLL se torna mais indicado quando um sinal de *encoder* é

utilizado na malha de realimentação. Ainda em (BEST, 2003), informa-se que no controle de motores, a utilização de um detector de fase e frequência é mais indicada do que a de um comparador de fase somente, sendo esta característica devida ao fato deste ser capaz de garantir um erro de velocidade nulo, e também por garantir o seguimento de referência independentemente das condições iniciais do sistema. Em (PLL... , 2014), um PLL com PFD é descrito em FPGA, com a mesma finalidade. Em (CAMPOS, 1991), o controle de velocidade de um motor DC é feito com um PFD, para que desta forma se pudesse modelar o comportamento do sistema para as duas variáveis, fase e frequência, e dividi-lo em duas regiões de operação. Devido a estas constatações, neste trabalho optou-se por implementar o controle de motor por PLL com a utilização de um DPLL contendo um PFD. O funcionamento básico de um DPLL na forma de software é brevemente descrito nesta seção, focando nos blocos que o diferencia de um LPLL, já descrito.

Da mesma forma que um LPLL, o comportamento de um DPLL pode ser descrito digitalmente na forma de software. Um diagrama de blocos que descreve este sistema deve ser composto por três partes: um comparador de fase e frequência, um filtro digital e um DCO. Um exemplo de DPLL em diagrama de blocos é mostrado na Figura 13.

Figura 13 – Diagrama de blocos para um DPLL como um SPLL



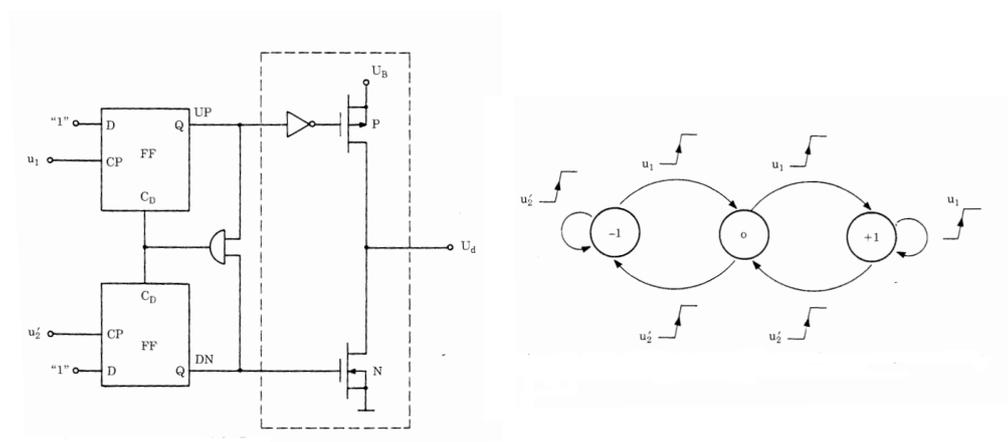
Fonte: retirado de (BEST, 2003)

No diagrama da Figura 13, além dos blocos, são mostrados os sinais e as suas ligações, como o sinal de entrada u_1 . Este sinal é responsável por conectar o bloco externo *edge detector* ao bloco PFD, gerando o tempo de interrupção, ou período de amostragem do software a cada borda de subida ou descida da onda quadrada, entregando-o ao bloco PFD, simbolizado pelo sinal f_s . Há também os sinais ϕ_2 e u_2 , os quais representam a onda

quadrada gerada pelo DCO, a qual deve conter as informações das variáveis de fase e frequência. Por fim, vê-se ainda o sinal de saída do filtro digital $u_f(n)$, e os sinais de pulso gerados pelo bloco PFD, $t_{+1}(n)$, e $t_{-1}(n)$.

A primeira parte de um bloco PFD, como se vê na Figura 14(a), é constituída por um circuito digital. Este circuito, por sua vez, é composto por dois flip-flops D com suas entradas sempre em nível alto, e RESETs ligados a uma porta AND, conectando-os às saídas. Os sinais u_1 e u_2 são conectados às entradas de *clock*, controlando desta forma a mudança das saídas para o próximo estado, a cada borda de subida. Portanto, sendo o PFD um circuito digital, o seu comportamento pode ser descrito por um diagrama de estados, como, por exemplo, o mostrado na Figura 14(b).

Figura 14 – (a) PFD (b) Diagrama de Estados de um PFD



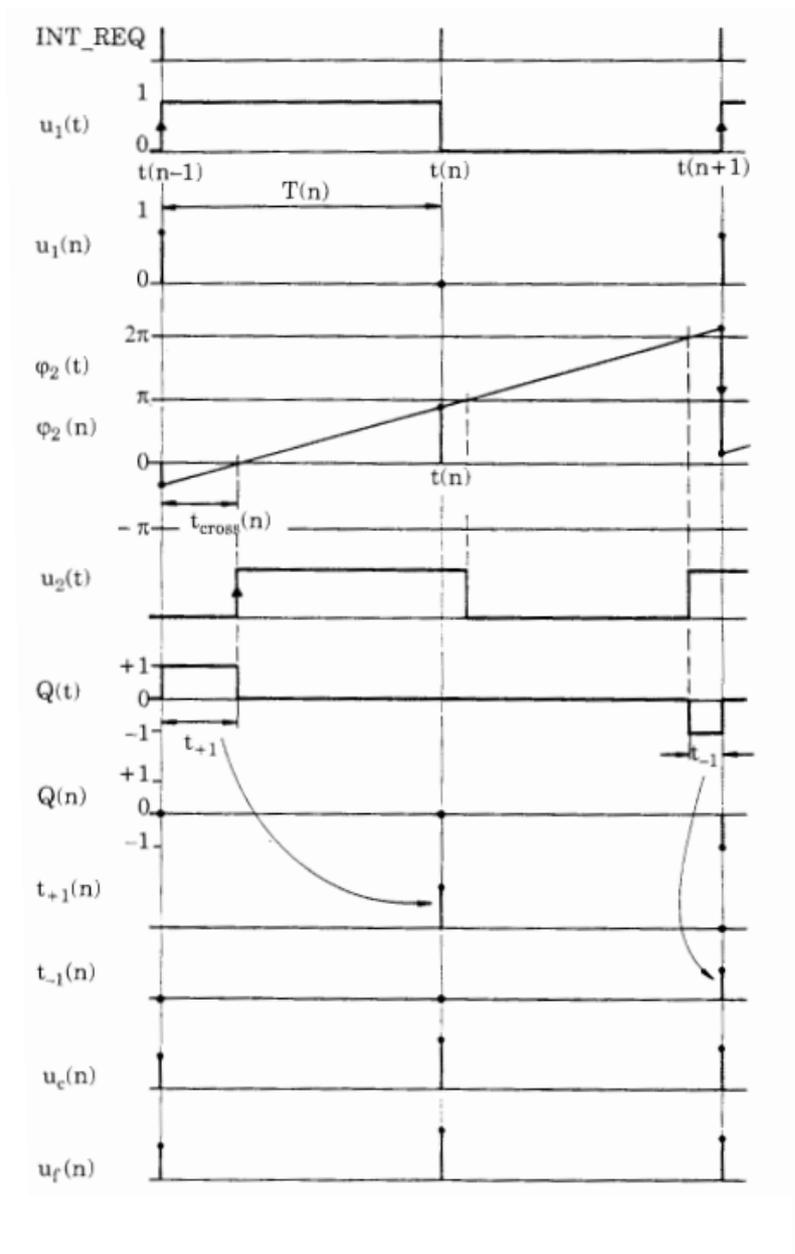
Fonte: modificado de (BEST, 2003)

Pela Figura 14(a), constata-se que os sinais de saídas dos flip-flops, UP e DN, são sinais binários, e que estes passam a ser os sinais de entrada de um bloco conectado a seguir, neste caso, um circuito chamado de *charged pump*. Analisando-se as possibilidades de entrada para o *charged pump*, nota-se a existência de quatro estados. Entretanto, pela Figura 14(b), percebe-se a omissão de um último estado, onde UP e DN estão ambos em nível alto. Esta condição se dá pela atuação da porta AND, que coloca o circuito instantaneamente no estado anterior quando UP e DN estão em nível alto. Consequentemente, gerando apenas três estados para a saída U_d : -1, 0 e +1.

Quando o PFD encontra-se no estado -1, têm-se que U_d gera uma tensão igual a tensão de referência. Quando o PFD encontra-se no estado 0, U_d se comporta como um circuito aberto, e quando o estado for +1, a saída U_d gera uma tensão igual a tensão de alimentação.

O código que descreve o comportamento do DPLL é significativamente mais complexo de se implementar, se comparado ao código que descreve o LPLL. Por este motivo, geralmente, parte-se de uma análise gráfica do comportamento dos sinais para se obter parâmetros e valores utilizados no seu algoritmo. Um exemplo deste tipo de gráfico está mostrado no Figura 15.

Figura 15 – Amostragem dos sinais de um SPLL como DPLL



Fonte: modificado de (BEST, 2003)

Uma descrição do comportamento dos sinais sinalizados na Figura 15 é dada a seguir:

- INT_REQ - Sinal que indica o momento em que uma interrupção deve ser chamada. Esta interrupção deve ocorrer toda vez que o sinal $u_1(t)$ troca de fase.
- $u_1(n)$ - Sinal que amostra $u_1(t)$ a cada interrupção, alternando o seu valor entre 0 e 1.
- $\phi_2(t)$ - Sinal de fase da onda quadrada gerada pelo DCO. Considerando-se que a frequência é constante, este sinal adquire a forma de uma reta, indicando a troca de estado de $u_2(t)$ a cada cruzamento pelos valores de $n\pi$. Para evitar o *overflow* nos registradores, retira-se 2π de seu valor toda vez que a fase cruza esta fronteira, deixando o sinal dentro do intervalo de 0 a 2π . Como, ao se retirar este valor do sinal, perde-se o tempo entre o cruzamento da fase de 2π e a interrupção, estima-se-o por semelhança de triângulos, com relação ao tempo de interrupção, como mostrado na Figura 15, onde este tempo é chamado de t_{cross} .
- $Q(t)$ - Sinal que indica o estado de Ud . Este sinal tem o seu valor modificado conforme o diagrama de estados da Figura 14(b), recebendo como entrada as bordas de subida dos sinais $u_1(t)$ e $u_2(t)$.
- $t_+(n)$ - Sinal discreto que quantifica em sua amplitude o tempo que $Q(t)$ permaneceu no estado +1 durante um período de uma interrupção. Este valor é utilizado na próxima interrupção como entrada do filtro digital.
- $t_-(n)$ - Sinal discreto que quantifica em sua amplitude o tempo que $Q(t)$ permaneceu no estado -1 durante um período de uma interrupção. Este valor é utilizado na próxima interrupção como entrada do filtro digital.
- $u_c(n)$ - Este sinal carrega a informação da tensão remanescente no capacitor de um filtro do tipo RC.
- $u_f(n)$ - Este sinal carrega a informação da tensão entregue na saída do filtro.

Dentro do algoritmo do SPLL analisado neste trabalho há três sub-algoritmos. O primeiro destes é o algoritmo para o cálculo do tempo de cruzamento. Este algoritmo tem como objetivo identificar se $\phi_2(t)$ ultrapassou o valor de 2π . Caso o tenha acontecido, deve-se subtrair um valor igual a 2π dos sinais $\phi_2(n)$ e $\phi_2(n - 1)$. Em seguida deve-se verificar se $\phi_2(n)$ é maior que zero e se $\phi_2(n - 1)$ é menor que zero. Caso esta condição seja verdadeira, significa que houve um cruzamento de fase e que uma borda de subida do sinal $u_2(t)$ aconteceu, e, portanto, há um t_{cross} a ser calculado.

O segundo algoritmo do SPLL é responsável por descrever o bloco PFD. Este algoritmo deve receber o estado anterior, $Q(n - 1)$, e analisar as bordas de subidas de $u_1(t)$ e $u_2(t)$ para então decidir o próximo estado, $Q(n)$, e calcular o tempo de permanência neste estado.

Caso o PFD no estado anterior esteja em $Q(n-1) = +1$, a única condição que passa a interessar é a ocorrência de uma borda de subida em $u_2(t)$, pois, caso isto não ocorra, o estado deve permanecer em $+1$ durante o período de interrupção, deixando desta forma Ud em nível alto. Caso contrário, o estado atual deve ir para 0 , retirando Ud do nível alto após o tempo de cruzamento, passando o restante do tempo de interrupção em aberto.

Caso o PFD no estado anterior esteja em $Q(n-1) = 0$, a ocorrência das bordas de subida tanto de $u_1(t)$ como de $u_2(t)$ interferem em Ud , resultando em quatro possibilidades. Quando a interrupção é gerada em uma borda de descida de $u_1(t)$, a ocorrência de uma borda de subida de $u_2(t)$ deve definir se o estado irá para -1 , após o tempo de cruzamento, ou se continuará no estado 0 pelo restante da interrupção. Quando a interrupção é gerada em uma borda de subida de $u_1(t)$, o próximo estado deve ir para $+1$, ficando em nível alto até o fim da interrupção, caso não ocorra uma borda de subida em $u_2(t)$, ou ficando neste nível durante um tempo t_{cross} , quando $u_2(t)$ troca de sinal.

Caso o PFD no estado anterior esteja em $Q(n-1) = -1$, a ocorrência das bordas de subida tanto de $u_1(t)$ como de $u_2(t)$ interferem em Ud . Entretanto desta vez resultando em três possibilidades, pois, quando uma borda de descida de $u_1(t)$ gera a interrupção, o comportamento do sinal $u_2(t)$ se torna *don't care*, deixando Ud em nível baixo durante este período. Quando a interrupção é gerada em uma borda de subida de $u_1(t)$, o próximo estado irá para 0 , com Ud ficando em aberto até o fim da interrupção, caso não ocorra uma borda de subida em $u_2(t)$, ou ficando neste nível durante um tempo t_{cross} , quando $u_2(t)$ troca de sinal.

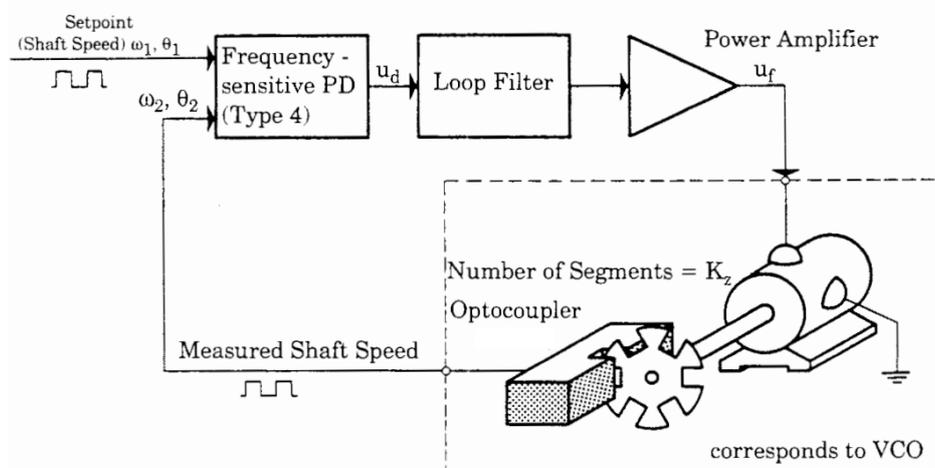
O terceiro e último algoritmo interno é o algoritmo que descreve o filtro. Como os filtros de um PLL são circuitos lineares, pode-se obter a sua função de transferência e aplicar-lhe a transformada Z para que desta forma se obtenha a sua equação de diferenças.

2.4 Modelo do Sensor

Neste trabalho, optou-se por utilizar um sensor do tipo *encoder* devido à medição do sinal de velocidade e de fase de forma direta, possibilitando assim um controle mais preciso.

O controle de velocidade de motores por PLL caracteriza-se pela substituição do bloco VCO do sistema por um motor. Para tal, deve-se fazer algumas adaptações no sistema como amplificar a tensão de saída do filtro PI, para que esta chegue condicionada à entrada do motor, e utilizar um sensor do tipo *encoder* para gerar uma onda quadrada com frequência proporcional à velocidade do motor. Um esquema de sistema PLL com a substituição do VCO por um motor é mostrado na Figura 16.

Figura 16 – Diagrama de blocos de um PLL com motor substituindo o bloco VCO



Fonte: retirado de (BEST, 2003)

O sinal de saída de um *encoder* é gerado por um feixe de luz que atravessa um disco com M ranhuras, criando um sinal com dois valores. Um valor quando o feixe de luz está atravessando a ranhura e outro valor quando este é bloqueado pelo disco. Normalmente o feixe de luz é emitido por um diodo LED e recebido por um fototransistor. Ainda com o objetivo de se obter uma onda quadrada livre de ruídos na saída, geralmente este sinal passa por um optoacoplador.

O *encoder*, além de ser o sensor de velocidade necessário para a realimentação com controlador PID, é também um sensor de fase. Isto se justifica, pois, quando aplicado a um detector de fase, a onda quadrada gerada pelo *encoder* é comparada ao sinal de referência, e, se a frequência de ambos os sinais é aproximadamente igual, situação chamada de região de trancamento de fase, a saída do detector de fase passa a ser proporcional à diferença de fases entre os sinais de referência e saída do *encoder*. Por este motivo, há duas regiões de operação para o sistema controlado por PLL, a região de trancamento e a região fora do trancamento de fase, sendo uma linear e a outra não. Portanto, deve-se analisar as características do sistema para ambas as situações, utilizando-se critérios de estabilidade e otimização de parâmetros distintos.

Estando o PLL operando na região de trancamento de fase, a variável de interesse do sinal do *encoder* se torna a própria fase, então, uma função de transferência, $H_e(s)$, que modele uma saída de fase do encoder, θ_e , pela velocidade do motor, ω , deve ser feita. A função de transferência deve resultar em uma relação como a dada na equação (2.43).

$$H_e(s) = \frac{\theta_e}{\omega} \quad (2.43)$$

Ainda, sabendo-se que a frequência da onda quadrada será M vezes maior que a velocidade angular do motor, $\omega_e = M\omega$, e que a variação da posição angular do encoder é igual a sua frequência, $\omega_e = M \frac{\partial \theta_e}{\partial t}$, a transformada de Laplace pode ser aplicada, resultando na equação (2.44), que descreve o modelo da função de transferência, a qual pode ser simplificada na forma apresentada em (2.45).

$$H_e(s) = \frac{\theta_e}{\frac{s\theta_e}{M}} \quad (2.44)$$

$$H_e(s) = \frac{M}{s} \quad (2.45)$$

Fora da região de trancamento, a variável de interesse se torna a frequência do sinal de saída do *encoder*. Uma função de transferência que relaciona a velocidade angular do motor com a velocidade angular da onda quadrada é simplesmente dada por um ganho igual ao número de ranhuras, M .

Por fim, deve-se mencionar que em muitas aplicações se utiliza um divisor de frequência na sequência de um *encoder*, podendo a frequência deste sinal ser dividida por um número N inteiro, o que resulta na geração de um sinal de *encoder* com uma frequência N vezes maior que a frequência de referência.

3 MATERIAIS E MÉTODOS

3.1 Simulações

Nesta seção são apresentadas duas formas diferentes de simulação, a primeira envolvendo *scripts* com rotinas do Matlab[®], e a segunda envolvendo a descrição de sistemas em bloco de mais alto nível.

Em 3.1.1, são mostradas as simulações de dois tipos SPLL, para entender a descrição de cada um de seus blocos na forma de software e como os critérios e faixas de operação são identificados nestes sistemas.

Em 3.1.2, estão mostradas simulações envolvendo a união de motores e sistemas PLL, sistemas de controle por PID e componentes eletro-mecânicos, assim como diversos outros blocos necessários para descrever os sistemas de controle de interesse, funcionando como uma etapa de pré-projeto para experimentos e também para a extração de conclusões sobre sistemas operando com cargas e dual-PLLs.

3.1.1 Simulações do SPLL

Neste trabalho, primeiramente, simulou-se um laço SPLL, em um *script* do software Matlab[®], sem o motor, para validar o código do sistema LPLL e se entender o funcionamento e descrição de cada bloco.

Como o controle de motores exige um comparador de fase do tipo IV, PFD, também foi necessário criar um *script* para descrever um sistema DPLL, assim, permitindo a comparação entre ambos os métodos de SPLL. Desta forma, também, pôde-se aproveitar parte do conhecimento obtido na implementação do LPLL, cuja implementação é mais simples, na implementação do DPLL.

3.1.1.1 Simulações do SPLL como LPLL

O LPLL pode ser descrito como um sistema eletrônico que, conforme mostrado na seção 2.3.2.1, tem o seu comportamento definido em SPLL através de três principais blocos: o detector de fase, PD, o filtro e o VCO. A transformação das equações que descrevem o funcionamento destes blocos em forma de código estão mostradas na mesma seção.

Com a finalidade de se validar o código SPLL, analisou-se a resposta gerada por este e comparou-se-a a resposta gerada por um LPLL teórico.

No projeto do circuito LPLL analógico e do SPLL, escolheu-se uma configuração para o sistema de forma a deixá-lo próximo àquela utilizada no controle de motores,

portanto, optou-se pela análise de um sistema contendo um filtro do tipo PI. Escolhendo-se esta configuração, garantiu-se um erro nulo aos ensaios do tipo degrau de fase, e também para o ensaio do tipo rampa de frequência. Ainda, adotou-se um sistema contendo um bloco detector de fase PD de tipo I, por este possuir uma versão digital de mais fácil reprodução, se comparado aos outros tipos.

A fim de se realizar a comparação entre o modelo de LPLL analógico e a sua versão em software, definiu-se um sistema, conforme um exemplo apresentado em (BEST, 2003), com $3Hz$ de largura de banda, com um coeficiente de amortecimento ξ de $\frac{1}{\sqrt{2}}$, com uma frequência de referência de $100kHz$, com ganho no laço direto K_0K_d de $2\pi 1000$ ($\gg \omega_n$) e um Δf_0 de $3Hz$.

Antes de se obter um SPLL equivalente ao descrito acima, adquiriu-se a sua função de transferência, para que desta forma se pudesse verificar a estabilidade e dinâmica do sistema. A demonstração dos cálculos utilizados para se modelar a função de transferência aproximada para o sistema LPLL escolhido, para as variáveis φ_1 e φ_2 , conforme a Figura 10, foram obtidos diretamente de (BEST, 2003), onde se chega na expressão apresentada em (3.1).

$$H(s)_{LPLL} = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.1)$$

Na expressão (3.1), vê-se que o sistema LPLL é aproximado por um sistema de segunda ordem em função de parâmetros bem conhecidos, como a frequência natural ω_n e o coeficiente de amortecimento ξ , os quais podem ser relacionados ao tempo de acomodação, margem de fase, largura de banda, e sobre-passo. Fez-se estas relações através de análises gráficas como a resposta em frequência (diagrama de Bode), e o ensaio ao degrau. Através destas, obteve-se, para o modelo do LPLL, um tempo de acomodação de $0,26s$, um sobre-passo de $20,00\%$, uma margem de fase de $127,00^\circ$, e uma margem de ganho infinita.

Para projetar e relacionar os parâmetros físicos do sistema com a dinâmica desejada, utilizou-se as relações entre estes e os parâmetros da função de transferência, levando-se em consideração os componentes do sistema eletrônico que formam o PLL. Segundo (BEST, 2003), tem-se que estes parâmetros estão relacionados conforme as expressões (3.2) e (3.3):

$$\omega_n = \sqrt{\frac{K_0 K_d}{N \tau_1}} \quad (3.2)$$

$$\xi = \frac{\omega_n \tau_2}{2} \quad (3.3)$$

onde K_0 é igual ao ganho do bloco VCO e K_d , o ganho do bloco PD. Pelas equações (3.2) e (3.3), também vê-se a dependência destes parâmetros com os parâmetros do controlador,

como o ganho do filtro PI, $\frac{\tau_1}{\tau_2}$, e o seu zero, $\frac{1}{\tau_2}$. Portanto, com ω_n e ξ definidos, obteve-se τ_2 , através de (3.3), que retornou um valor de 4,00ms. Em seguida, definiu-se um PFD com um ganho K_d igual a 0.79, equivalente ao ganho gerado por um PFD alimentando com 5V. Com K_d definido, atentou-se para a condição de aproximação utilizada, $K_d K_0 \gg \omega_n$, optando-se, portanto, por manter a relação $K_d K_0 = 2\pi 1000$. Desta forma, obteve-se um K_0 de 7890,00 $\frac{rad}{V.s}$. Neste ponto, pôde-se calcular τ_1 conforme (3.2), chegando-se a um valor de 17,68s.

Além do objetivo de garantir a estabilidade e a dinâmica desejada para o LPLL, com este atuando na região de trancamento, procurou-se assegurar o mesmo para a sua atuação fora desta região. Fora da região de trancamento, tratou-se de questões como: quais condições colocam o PLL na região de trancamento; quanto tempo este processo leva e quais condições podem tirá-lo desta região. Como o sistema LPLL fora da região de trancamento é altamente não-linear, utilizou-se de especificações de projeto já resolvidas para sistemas PLL, onde se estabelecem valores para faixas de operação em função da frequência de referência.

Em (BEST, 2003), deduz-se faixas de operação que delimitam os valores de frequência para o sinal de entrada de forma simétrica ao redor da frequência do sinal de referência, ω_0 . De lá, tem-se que cada uma destas faixas possui uma amplitude de operação que varia ao redor de ω_0 por um valor de $\frac{\Delta\omega}{2}$, para cima e para baixo. Portanto, pode-se dizer que os valores limites de operação, nestas regiões, são obtidos de relações como a dada em (3.4).

$$\omega_0 \pm \frac{\Delta\omega}{2} \quad (3.4)$$

A primeira faixa de interesse para um LPLL fora da região de trancamento é chamada de *hold range*, $\Delta\omega_H$. Esta faixa de operação delimita os máximos valores que a frequência de referência pode ter sem se instabilizar o sistema. Em (BEST, 2003), são desenvolvidas expressões para identificar estes limites de operação. Para a topologia LPLL com filtro PI, chega-se em um *hold range* infinito.

A segunda faixa de operação identificada, $\Delta\omega_{PO}$, é chamada de *pull-out range*, sendo esta faixa responsável por identificar o máximo degrau ou distúrbio permitido de se aplicar na frequência de referência. Em (BEST, 2003) é mostrado que, para um PLL que utiliza um PD do tipo I e um filtro PI, o valor de $\Delta\omega_{PO}$ é dado pela expressão (3.5), da qual se obteve um valor de 57,92Hz.

$$\Delta\omega_{PO} = 1,8\omega_n(\xi + 1) \quad (3.5)$$

Ainda, há o tempo que esta configuração de PLL leva para rejeitar o degrau, o qual tem o seu valor obtido da expressão (3.6), retornando para o LPLL proposto um valor de 4,45ms.

$$TP = \frac{\pi^2 \Delta\omega_0^2}{16\xi\omega_n^3} \quad (3.6)$$

A terceira faixa, chamada de *pull-in range* $\Delta\omega_P$, é identificada como a região de operação na qual sempre se garante o retorno do PLL ao estado de trancamento. Novamente, para um LPLL com filtro do tipo PI, se considerada esta faixa como sendo infinita. Por fim, há a região *lock range*, ou $\Delta\omega_L$. Região cujos valores de frequência são próximos o suficiente dos valores da referência para que se possa considerá-los iguais, e que há somente variação de fase. Para a configuração de LPLL adotada, obtém-se o valor de $\Delta\omega_L$ da expressão (3.7).

$$\Delta\omega_L = 2\xi\omega_n \quad (3.7)$$

Aplicando-se a expressão (3.7) no LPLL projetado, obteve-se um *lock range* de 26,66Hz.

A obtenção do tempo que o sistema leva, já na região de trancamento, para entrar em regime permanente, foi realizada conforme indicado em (BEST, 2003), portanto obteve-se o seu valor aproximado através da expressão (3.8), da qual extraiu-se um valor de 0,46s para o LPLL projetado.

$$TL = \frac{2\pi}{\omega_n \sqrt{1 - \xi^2}} \quad (3.8)$$

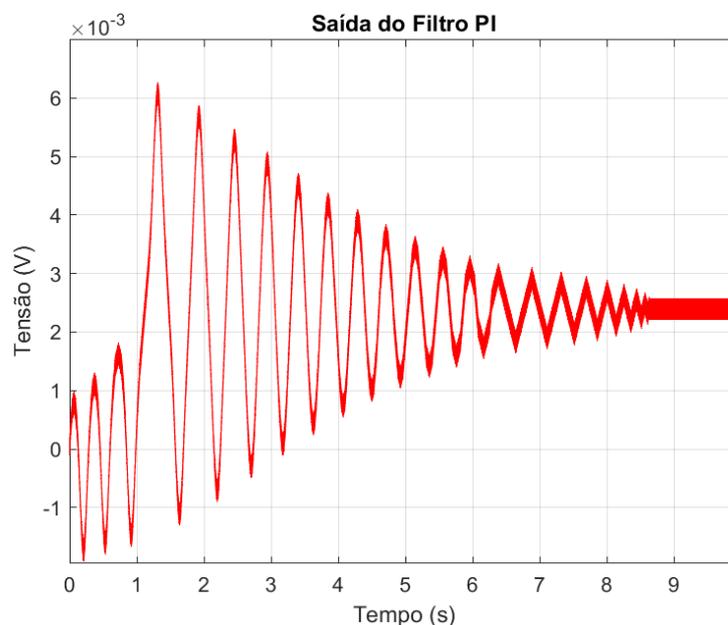
Na digitalização do filtro PI, utilizou-se as expressões (2.39), (2.40) e (2.41) para se chegar nos valores de a_1 , b_0 e b_1 , as quais retornaram os valores de -1 , $-2,25 \cdot 10^{-4}$ e $2,25 \cdot 10^{-4}$, respectivamente.

Após realizar a descrição dos blocos, definiu-se o tempo de amostragem, o qual foi arbitrado como tendo um tempo igual a um vigésimo do valor do período do sinal de referência, ou seja, escolheu-se um $T = 0,5 \cdot 10^{-6}s$, garantindo-se desta forma o cumprimento do critério de Nyquist.

Com todos os parâmetros necessários para a descrição do código SPLL determinados, e margens de operação definidas, implementou-se o SPLL em um *scrip* do Matlab®. O *script* com a inicialização das variáveis e a simulação do sistema como um todo estão mostrados no Apêndice A.

Ao se realizar a simulação do código SPLL, verificou-se o resultado da digitalização do LPLL através do sinal de saída do filtro PI simulado $u_f(t)$, o qual ficou conforme mostra a Figura 17.

Figura 17 – Saída do filtro para o LPLL descrito em software



Fonte: do Autor

Conforme (BEST, 2003), a extração do parâmetro TL , de sistemas PLL, pode ser feita através do sinal $u_f(t)$, sendo o tempo de acomodação para a fase retirado de um período da oscilação de ressonância do sinal. Já o valor aproximado de TP foi obtido do tempo em que o sinal atinge o regime permanente, definido como o tempo em que o sinal atinge uma variação menor que 2% em torno de um valor constante. Pelo gráfico apresentado na Figura 17, obteve-se um TL de 0,46s e um TP de 7,50s. Pelo valor de TL estar muito próximo do gerado pelo LPLL, ou seja, perto de 0,46s teórico proposto, considerou-se que sua digitalização apresentou o comportamento desejado, mesmo obtendo-se um TP significativamente diferente dos 0,45s teóricamente gerados por um LPLL.

3.1.1.2 Simulações do SPLL como DPLL

O DPLL é um sistema eletrônico que, conforme mostrado na seção 2.3.2.2, pode ter o seu comportamento descrito em SPLL através de seus três principais blocos: o detector de fase PFD, o filtro e o VCO. Os sinais dos quais se deduzem as equações que descrevem o funcionamento destes blocos, em forma de código, são apresentados e definidos na mesma seção.

Buscando se validar um código SPL para um DPLL, analisou-se a resposta gerada por este e comparou-se-a a resposta gerada por um DPLL teórico.

Na escolha de um circuito DPLL analógico para comparação, devido a maior facilidade de análise, e pela maior quantidade de exemplos na literatura, optou-se por uma configuração que utiliza um filtro passa-baixa RC, ao invés de um filtro PI.

Levando-se em consideração as escolhas pré-estabelecidas, definiu-se o sistema DPLL como tendo $0,01Hz$ de largura de banda, um coeficiente de amortecimento ξ de $\frac{1}{\sqrt{2}}$, uma frequência de referência de $100kHz$, um ganho no laço direto K_0K_d de $2\pi 1000$ ($\gg \omega_n$) e um Δf_0 de $3Hz$.

Antes de adquirir um SPL equivalente, utilizou-se os valores de ω_n e ξ , definidos, para se obter a função de transferência do sistema, para que desta forma se pudesse verificar a sua estabilidade e dinâmica. Os cálculos necessários para se chegar na função de transferência aproximada para o sistema DPLL escolhido, para as variáveis φ_1 e φ_2 , conforme a Figura 13, foram obtidos diretamente de (BEST, 2003), onde se chega na expressão apresentada em (3.9).

$$H(s)_{DPLL} = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.9)$$

Na expressão (3.9), vê-se que o sistema DPLL é aproximado por um sistema de segunda ordem em função de parâmetros bem conhecidos como a frequência natural ω_n e o coeficiente de amortecimento ξ , podendo-se portanto relacioná-los ao tempo de acomodação, margem de fase, largura de banda, e sobre-passo. Fez-se estas relações através de análises gráficas de resposta em frequência (diagrama de Bode), e ensaio ao degrau. Assim, conseguiu-se para o modelo do DPLL um tempo de acomodação de $7,79ms$, um sobre-passo de $20,80\%$, uma margem de fase de $127,00^\circ$, e uma margem de ganho infinita.

Visando projetar e relacionar os parâmetros físicos do sistema com a dinâmica desejada, utilizou-se as relações entre estes e os parâmetros físicos do sistema DPLL, como os componentes eletrônicos dos circuitos que o compõe. Segundo (BEST, 2003), ω_n e ξ estão relacionados com os parâmetros físicos conforme as expressões (3.10) e (3.11),

$$\omega_n = \sqrt{\frac{K_0 K_d}{N(\tau_1 + \tau_2)}} \quad (3.10)$$

$$\xi = \frac{\omega_n}{2(\tau_2 + \frac{N}{K_0 K_d})} \quad (3.11)$$

onde K_0 é o ganho do bloco VCO e K_d o ganho do bloco PFD. Pelas equações (3.10) e (3.11), também vê-se a dependência destes parâmetros com os parâmetros do filtro, τ_1 e τ_2 . Portanto, com ω_n e ξ definidos, pôde-se obter τ_2 através de (3.11), retornando um valor de

2,10ms. Após, definiu-se um PFD com um ganho K_d igual a 0,16, equivalente ao ganho gerado por um PFD alimentando com 2V. Com K_d definido, atentou-se para a condição de aproximação utilizada, $K_d K_0 \gg \omega_n$, para que, mantendo-se a relação, $K_d K_0 = 2\pi 1000$, se pudesse obter um K_0 de $39478,00 \frac{rad}{V.s}$. A partir deste ponto, pôde-se calcular τ_1 através de (3.10), obtendo-se para ele um valor de 9,99s.

Assim como feito para o LPLL, além de garantir-se a estabilidade e a dinâmica desejada para o DPLL, com este atuando na região de trancamento, procurou-se assegurar o mesmo com este atuando fora. Fora da região de trancamento, procurou-se tratar das mesmas causas observadas no projeto do LPLL.

Em (BEST, 2003), são desenvolvidas expressões para se identificar os limites de operação da faixa de *hold range* para um DPLL contendo em sua topologia um filtro RC e um detector de fase do tipo IV, onde chega-se em um *hold range* infinito.

A faixa de *pull-out range*, responsável por identificar o máximo degrau ou distúrbio permitido de se aplicar na frequência de referência, para um PLL que utiliza um detector de fase do tipo IV e um filtro RC, é dada pela expressão (3.12), de onde se obteve um valor de 8722Hz para o DPLL proposto.

$$\Delta\omega_{PO} = 11,5\omega_n(\xi + 0,5) \quad (3.12)$$

Ainda, há o tempo em que esta configuração de PLL leva para rejeitar este degrau, o qual tem o seu valor obtido da expressão (3.13), retornando para o DPLL proposto um valor de 0,01s.

$$TP = \frac{\pi^2 \Delta\omega_0^2}{16\xi\omega_n^3} \quad (3.13)$$

A terceira faixa de interesse, chamada de *pull-in range* $\Delta\omega_P$, identificada como a região de operação na qual sempre se garante o retorno do PLL ao estado de trancamento, para o DPLL proposto, foi considerada como tendo valor infinito.

Por fim, há a região *lock range*, ou $\Delta\omega_L$. Região cujos valores de frequência são próximos o suficiente dos valores da referência para que se possa considerá-los iguais, e que há somente variação de fase. Para a configuração de DPLL adotada, obteve-se o valor de $\Delta\omega_L$ da expressão (3.14).

$$\Delta\omega_L = 4\pi\xi\omega_n \quad (3.14)$$

Aplicando-se a expressão (3.14) no DPLL projetado, obteve-se um *lock range* de 5583Hz.

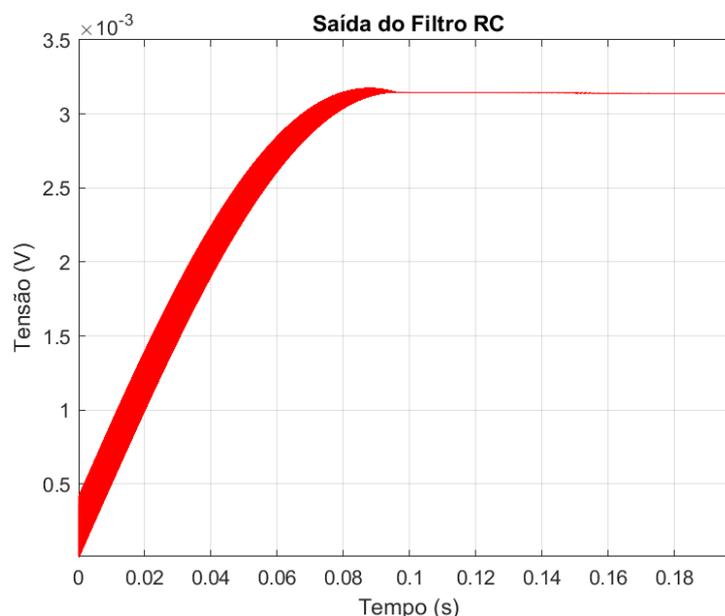
A obtenção do tempo que o sistema leva para entrar em regime permanente, ou entrar na região de trancamento, foi realizada da mesma maneira que apresentado no capítulo de medidas em (BEST, 2003). Portanto obteve-se o seu valor aproximado através da expressão (3.15), a qual retornou um valor de 141,42s para o DPLL projetado.

$$TL = \frac{2\pi}{\omega_n \sqrt{1 - \xi^2}} \quad (3.15)$$

Após se identificar as faixas de operação, definiu-se o tempo de interrupção como sendo igual a metade do valor do período do sinal de referência $u_1(t)$, ou seja, definiu-se um $T = 1,0 \cdot 10^{-5}s$, garantindo-se assim o cumprimento da análise gráfica realizada na seção 2.3.2.2.

Com todos os parâmetros do código SPLL escolhidos, e margens de operação definidas, descreveu-se o SPLL em um *script* do Matlab®. O *script* com a inicialização das variáveis e a simulação do sistema está mostrado no Apêndice B. Ao executar-se o *script*, pôde-se verificar o resultado da digitalização do DPLL através do sinal de saída do filtro $u_f(t)$ simulado, a qual ficou como mostrado na Figura 18.

Figura 18 – Saída do filtro para o DPLL descrito em software



Fonte: do Autor

Conforme (BEST, 2003), a extração do parâmetro TL de sistemas PLL, pode ser feita através do sinal de saída do filtro $u_f(t)$, sendo o tempo de acomodação para a fase retirado de um período da oscilação de ressonância. Já o tempo de acomodação para a frequência, TP , foi aproximado como o tempo em que o sinal atinge o regime permanente, geralmente definido como o tempo em que o sinal atinge uma variação menor que 2% de

um valor constante. Pelo gráfico apresentado em 18, obteve-se um TP de $0,07s$, e não se pôde verificar o valor de TL devido ao fato de este ter sido projetado maior que o valor de TP , portanto, não havendo tempo suficiente para que o sinal $u_f(t)$ pudesse oscilar. Por estes valores estarem de acordo com um DPLL teórico, considerou-se que sua digitalização apresentou um resultado coerente, indicando o funcionamento do código.

3.1.2 Simulações do Processo

Atualmente existem métodos confiáveis de simulação, nas mais distintas áreas, com respostas cada vez mais fiéis à realidade, atingidas através de softwares, como por exemplo o software Simulink®. O Simulink® possui diversas bibliotecas para simulações de diferentes tipos de sistemas com foco em sistemas de controle, por este motivo, optou-se pela sua utilização neste trabalho.

Neste trabalho, utilizou-se a biblioteca Simscape (MATHWORKS, 2022b). Esta biblioteca é especializada na simulação de sistemas físicos do tipo elétrico e mecânico, permitindo a simulação de motores, como o BLDC, e a alteração de algumas propriedades mecânicas, como a adição de torque de carga no rotor. Ainda, há blocos que simulam componentes eletro-mecânicos como *encoders*. O Simulink® também possui a biblioteca Mixed-Signal Blockset (MATHWORKS, 2022a), a qual possui diversos componentes eletrônicos. Dentre estes, há os blocos internos de sistemas PLL, os quais também foram utilizados neste trabalho.

Em 3.1.2.1, mostra-se a forma em que realizou-se as simulações, tanto em malha aberta como em malha fechada, utilizando-se as bibliotecas mencionadas acima. Ainda, se mostra os dados de entrada gerados e as saídas resultantes. Por fim, expõe-se os modelos obtidos, e os critérios utilizados para se escolher aquele com melhor resposta.

Em 3.1.2.2 e 3.1.2.3, são mostradas as ferramentas de simulação e cálculos adotados para se fechar a malha de controle e projetar o controlador PI, utilizado no método de controle PID, e o filtro PI, utilizado no método de controle PLL. Também são descritos os cuidados tomados para que os sinais de controle se comportassem de forma a respeitar os limites dos atuadores.

Ainda, em 3.1.2.4 e 3.1.2.5 se mostra os critérios adotados na escolha do distúrbio a ser implementado, e o modo no qual se fechou a malha para o sistema dual-PLL, respectivamente.

3.1.2.1 Simulações do Processo em Malha Aberta

Para simular o processo de controle de velocidade de um motor BLDC, utilizou-se o bloco BLDC da biblioteca *Simscape*. Dentre as possíveis configurações de motor, optou-se por um motor com enrolamentos em estrela e com força contraeletromotriz perfeitamente

trapezoidal. Os parâmetros utilizados no motor simulado estão apresentados na Tabela 1, os quais seguem as mesmas notações adotadas na seção 2.1.

Tabela 1 – Parâmetros do motor BLDC simulado

Parâm. Valores	
B_m	0,18Wb
p	4
L_s	0,12mH
M_s	0,02mH
L_m	0,00mH
J	0,00kgm ²
R	1,00mΩ

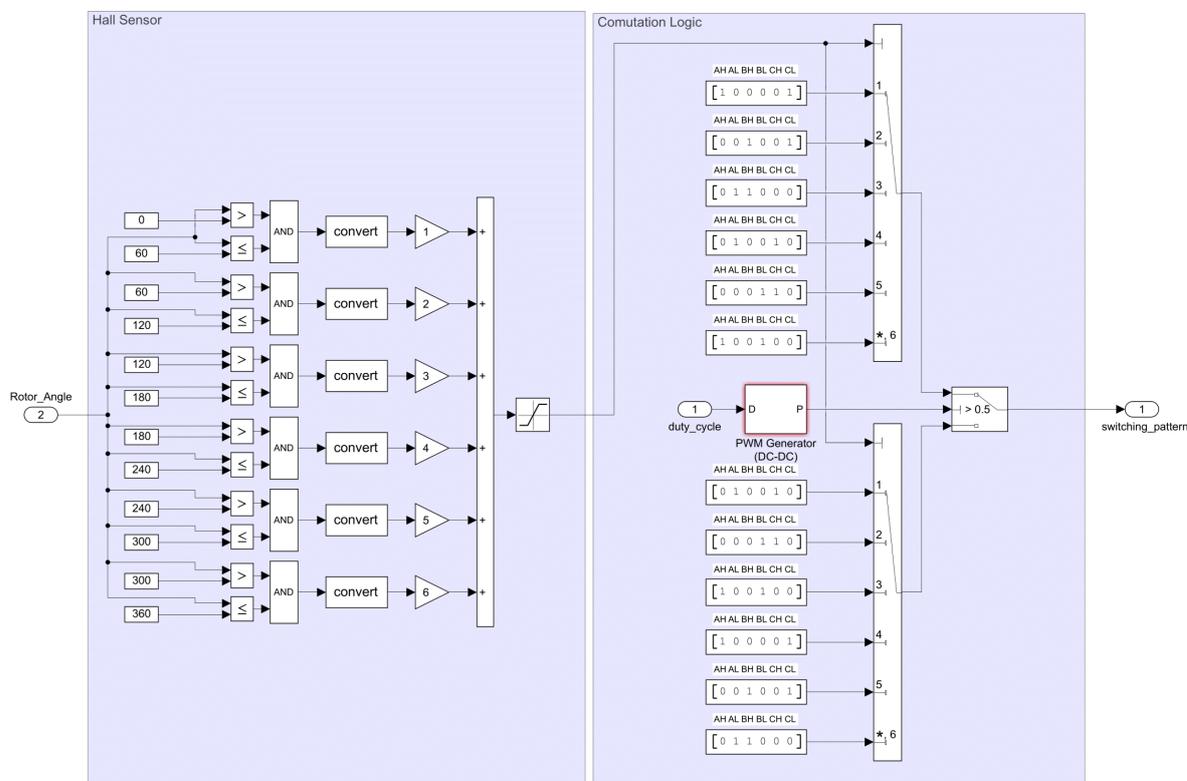
O bloco escolhido para simular o sensor de velocidade de posição foi o *Motion Sensor*. Este bloco tem como característica a capacidade de receber o sinal vindo do rotor e de decodificá-lo em duas saídas: a velocidade angular e o ângulo de fase. Com a obtenção destes sinais, utilizou-se os valores de velocidade como variável controlada no laço de realimentação e os valores de fase para a realização da lógica de mudança de estados das chaves do acionamento, conforme explicado na seção 2.1.1.

A lógica de comutação, responsável pela geração dos sinais que são enviados para a ponte trifásica, foi feita com a utilização de blocos lógicos de chaveamento em conjunto com o bloco *PWM Generator*. Na sua implementação, empregou-se blocos lógicos para dividir a região de operação do rotor em seis regiões distintas de 60°, sendo cada região responsável pelo acionamento de um estado diferente. Também utilizou-se uma chave lógica para comutar os transistores de um mesmo braço da ponte trifásica, conforme o nível alto ou baixo do sinal PWM. Os blocos e chaves escolhidos para o controle do inversor-trifásicos no Simulink® estão mostrados na Figura 19.

Para enviar o sinal que contém a lógica de chaveamento para a ponte trifásica, utilizou-se o bloco *Six-Pulse Gate Multiplexer*, o qual tem como papel controlar o bloco *Converter Three-Phase*. Por consequência, utilizou-se o bloco *Converter Three-Phase* para simular a ponte trifásica. Ainda, o bloco *Converter Three-Phase* teve de ser conectado a um sinal de alimentação e ao sinal de controle, para que se pudesse gerar os sinais de tensão e correntes enviados ao motor.

A união dos blocos descritos até aqui constitui o sistema comumente chamado de planta. Há dois principais blocos que constituem a planta: o motor, que recebe o sinal de tensão ou corrente em sua entrada e devolve em sua saída o sinal de velocidade; e o bloco de acionamento, que recebe em sua entrada um o ciclo de trabalho de um sinal PWM e gera em sua saída um sinal de tensão e corrente. A função de transferência para cada um destes blocos é discutida na seção 2.1.

Figura 19 – Blocos utilizados na lógica de chaveamento

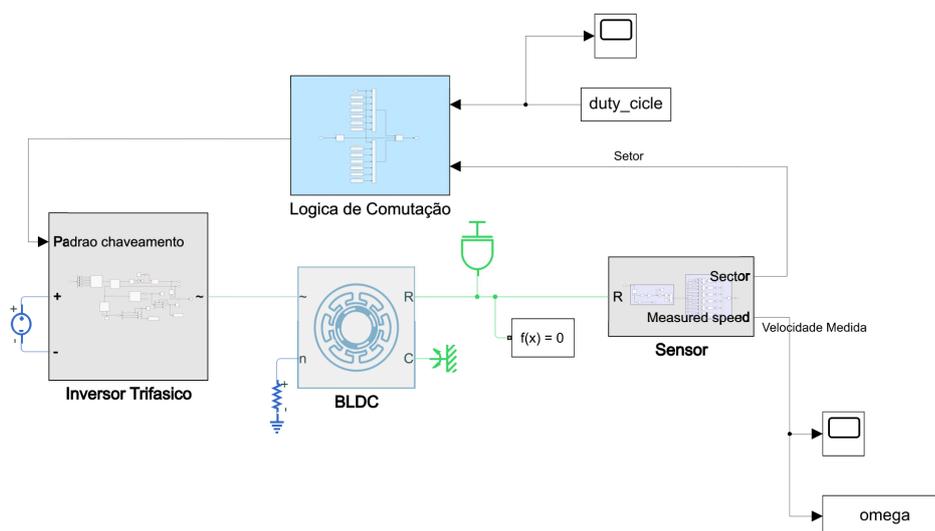


Fonte: do Autor

Do ponto de vista do método de controle PID, a planta é composta pelos blocos de acionamento e do motor, ambos em série, portanto, com a união destes tendo a sua entrada sendo considerada como o ciclo de trabalho e a sua saída a velocidade do rotor. Do ponto de vista do método de controle por PLL, a arquitetura dos blocos se mantém a mesma, quando a variável controlada é a frequência angular, entretanto, dentro da região de trancamento, esta variável passa a ser a fase. Para o controle de fase, a planta passa a ter um terceiro bloco em série, neste caso, considera-se um bloco integrador na sequência do bloco BLDC. A planta do sistema em malha aberta, simulada, tendo como variável controlada a velocidade, ficou estruturada conforme se vê na Figura 20.

A identificação do modelo da planta, mostrada na Figura 20, deu-se pela geração e coleta dos sinais de entrada e saída, ambos efetuados por um *script* do Matlab®, e pela definição do tempo de amostragem. No *script*, estes dados foram encapsulados em um objeto do tipo *iddata*, o qual foi utilizado como entrada de ferramentas de estimativa de modelo. Através destas ferramentas, pôde-se realizar comparações entre as respostas

Figura 20 – Blocos utilizados na simulação de malha aberta



Fonte: do Autor

geradas pelo modelo e pela simulação, permitindo-se verificar a qualidade do encaixe entre estas.

As estimativas de modelo foram realizadas no *script* pela função *tfest*. Esta função pertence à ferramenta System Identification Toolbox (MATHWORKS, 2022c), que é caracterizada por encontrar parâmetros que minimizam o erro entre a saída do modelo e a saída medida. Este erro, por sua vez, é calculado por uma função conhecida como função de perdas ou custo. Após ser configurada, a função de perdas retorna valores que servem de métrica para a qualidade do modelo. Dentre estas métricas há o percentual de encaixe, dado em porcentagem, conforme (3.16),

$$Fit_{Percent} = 100 \left(1 - \frac{\|y_{medido} - y_{modelo}\|}{\|y_{medido} - y_{medido}^*\|} \right) \quad (3.16)$$

de onde tem-se que y_{medido} é um vetor de tamanho N que guarda os valores de saída medidos, e y_{medido}^* é um vetor de tamanho N contendo a média destes valores. Já y_{modelo} é um vetor de mesmo tamanho contendo os valores da saída do modelo. Para esta métrica, valores próximos de 100% são considerados ideais.

Para realizar ensaios ao degrau, criou-se uma rotina em *script* para a geração de sinais constante de ciclo de trabalho, sendo estes ativados a partir de um tempo zero e enviados com um tempo de amostragem de $25\mu s$ durante um intervalo de $1,5s$ para um bloco gerador de PWM. O bloco gerador de PWM foi configurado para gerar um sinal com período de $1ms$, garantindo-se, desta forma, uma atualização a cada transição com folga.

A identificação da região de operação do motor, deu-se pela aplicação de diferentes degraus de entrada. Após a realização de alguns ensaios, observou-se empiricamente que o motor pôde operar com valores de ciclo de trabalho entre 0,5 e 1, com este não partindo ao se utilizar valores de ciclo de trabalhos menores que 0,5. Por esta razão, escolheu-se um valor de 0,75 para o ciclo de trabalho, pois, ao se escolher um valor central, dentro da faixa de operação, obteve-se a máxima amplitude de atuação.

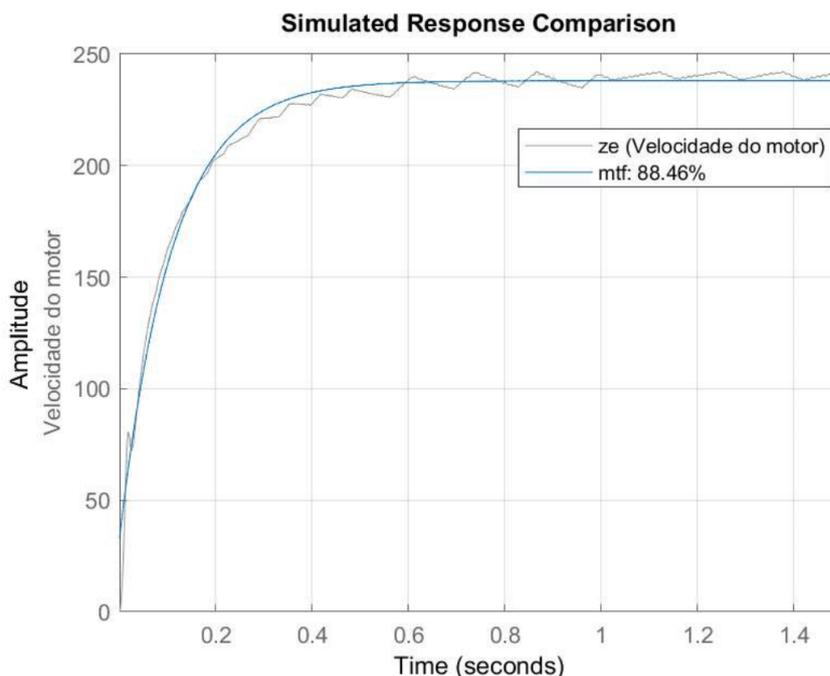
Para coletar o sinal de saída gerado nos ensaios, utilizou-se uma taxa de amostragem de $20kHz$, uma frequência muito superior a qualquer frequência natural estimada para o sistema, garantindo-se assim o atendimento do critério de Nyquist.

A aquisição do modelo através do ensaio ao degrau foi realizada com intuito de se obter o tempo de acomodação do sistema. Pois, como mencionado na seção 2.2.2, o ensaio PRBS é o ensaio mais desejado quando a planta envolve motores, fazendo-se necessária, portanto, a obtenção deste valor.

Com o intuito de adquirir o tempo de acomodação, utilizou-se a função *tfest* para se obter a estimativa do modelo por uma função de transferência de primeira ordem, com um pólo e nenhum zero. A escolha deste tipo de modelo se justificou pelo fato de o BLDC simulado ter a configuração de rotor ideal, sem momento de inércia, $J = 0$, o que, conforme a expressão (2.23), retornou uma função de transferência de primeira ordem. Ao se utilizar a função *tfest*, obteve-se um modelo com 88,46% de encaixe. As respostas ao degrau do sistema simulado e modelado podem ser vistas na Figura 21, a partir da qual se pôde medir um tempo de acomodação de $0,43s$ para uma variação máxima nos valores de saída de 2%.

A geração do sinal PRBS, deu-se pela utilização da função *idinput*. Para a geração deste sinal, além de se indicar o tipo, deve-se informar o número de amostras, e, no caso do sinal PRBS, também deve-se selecionar os valores de amplitude e a banda de operação. Os valores de amplitude, conforme discutido na seção 2.2.2, devem ser escolhidos a partir do conhecimento empírico do projetista sobre o sistema. Realizando-se simulações de ensaio PRBS, chegou-se em melhores resultados quando se utilizou amplitudes de ciclo de trabalho com valores de 0,7 e 0,8. A banda de operação foi definida com seus limites conforme a expressão dada em (2.27), na qual utilizou-se o tempo de acomodação de $0,43s$.

Figura 21 – Resposta ao degrau em malha aberta



Fonte: do Autor

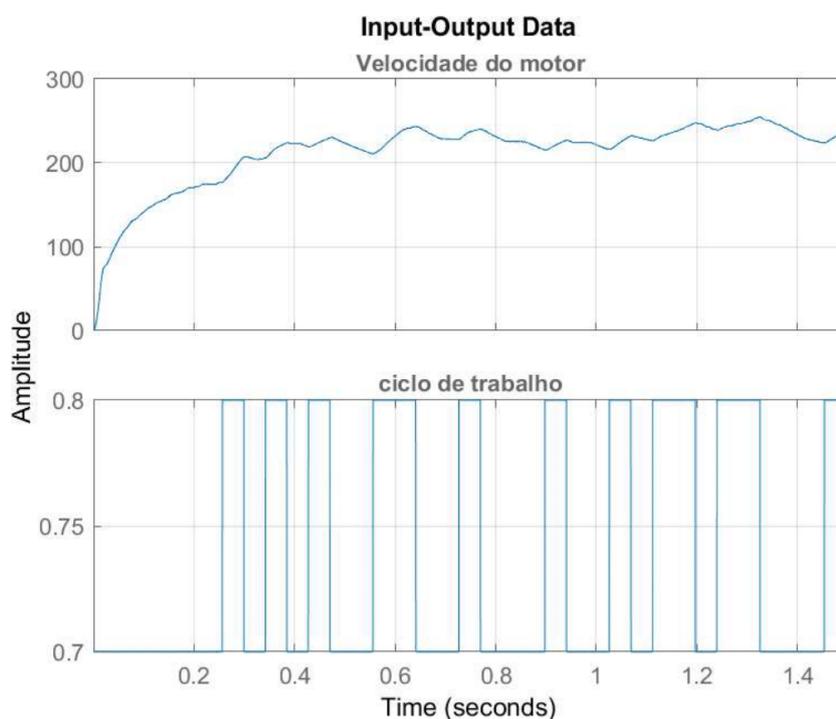
Com o sinal PRBS criado, realizou-se o seu ensaio. Após a sua realização, coletou-se novamente os sinais de saída e entrada. O resultado do ensaio contendo a saída junto de sua entrada estão mostrados na Figura 22.

Tendo-se adquirido os dados de entrada e saída, construiu-se novamente o objeto *iddata*, para utilizá-lo na identificação do sistema, através da função *tfest*.

Com a finalidade de se buscar a melhor estimativa, verificou-se diferentes ordens para o modelo, obtendo-se aproximações para plantas com 2 polos e nenhum zero; 2 polos e 1 zero; 3 polos e 2 zeros; e 4 polos e 3 zeros. Apesar de teoricamente o modelo do motor ser de primeira ordem, quando o momento de inércia do rotor não é incluso, ainda assim optou-se por verificar os resultados para sistemas com ordens diferentes. A necessidade de se realizar estas verificações surgiu do fato de que os modelos de ordens maiores são melhores na inclusão dos efeitos provocados pelas não linearidades apresentadas, tanto pelo motor como pelo acionamento.

Os modelos com as ordens citadas acima retornaram respectivamente os seguintes percentuais de encaixe: 84,67%, 86,33%, 87,19% e 71,66%, a partir dos quais pôde-se ver que o modelo sem zeros não teve o melhor desempenho ao seguir velocidades variadas.

Figura 22 – Ensaio PRBS simulado



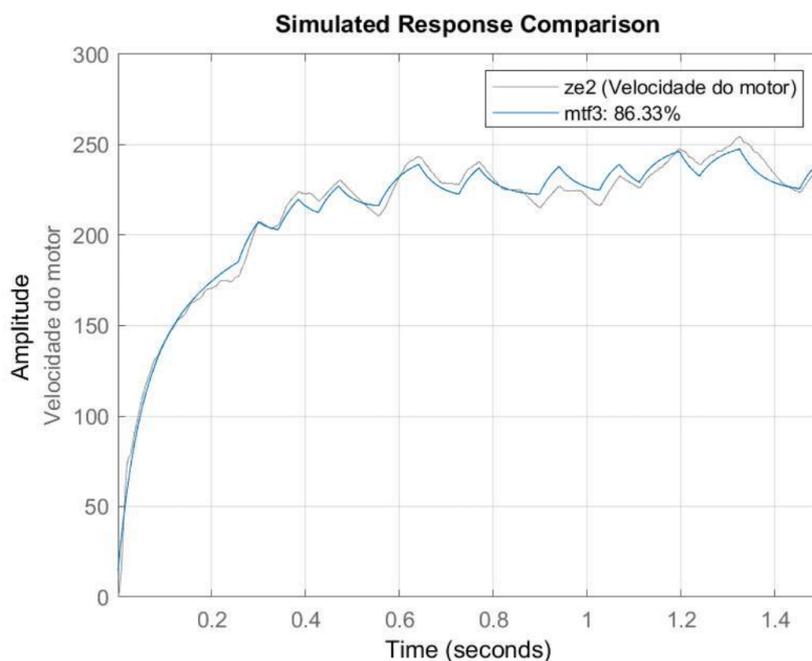
Fonte: do Autor

Isto se deve ao limitado grau de liberdade obtido ao se aproximar um sistema por duas exponenciais. Por outro lado, as aproximações com mais pólos e zeros tiveram desempenhos ainda piores, incluindo efeitos não existentes no sistema. As melhores estimativas foram obtidas com os modelos de 2 polos e 1 zero e 3 polos e 2 zeros. Como os percentuais de encaixe obtidos nestas situações foram muito próximos, optou-se pelo modelo com menos pólos, evitando-se assim uma maior complexidade, o que poderia exigir um projeto mais complicado de controlador e, ao mesmo tempo, um sistema menos robusto em termos de estabilidade. Portanto, adotou-se o modelo com 2 polos e 1 zero. A comparação da resposta simulada, em conjunto com a resposta do modelo, quando aplicado o sinal *PRBS*, para o modelo adotado, ficou conforme mostrado na Figura 23.

3.1.2.2 Simulações do Processo em Malha Fechada com PID

A função de transferência estimada para a planta, definida como tendo um zero e dois pólos, cujo encaixe do modelo retornado foi de 86,33%, foi a escolhida para modelar o sistema e projetar o controlador. Os valores encontrados para este zero e pólos foram de $5,591s^{-1}$, $23,39s^{-1}$ e $4,32s^{-1}$, respectivamente. A versão final da função de transferência, com o ganho incluso, é mostrada na expressão (3.17).

Figura 23 – Resposta para a entrada PRBS simulada



Fonte: do Autor

$$\frac{54600s + 305500}{s^2 + 27,71s + 101} \quad (3.17)$$

O fechamento da malha, para o controle de velocidade, deu-se pela adição de um bloco controlador PI e de um bloco somador para realizar a realimentação negativa. Com os blocos nesta disposição, garantiu-se um sistema de controle com arquitetura padrão.

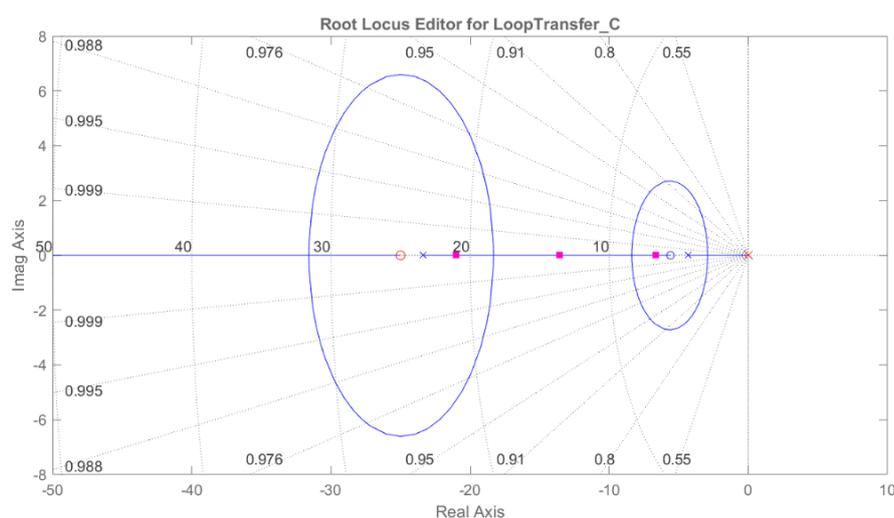
O projeto do controlador PI foi realizado com a ferramenta *rltool* do Matlab®. Nesta ferramenta realizou-se o projeto pelo método do lugar geométrico das raízes. Antes de se iniciar o projeto do controlador, alguns objetivos foram estabelecidos a fim de se entregar um controle estável e melhorado em comparação com malha aberta. Primeiramente, definiu-se que o projeto do controlador teria de entregar um sistema em malha fechada com tempo de acomodação menor que o observado em malha aberta, com sobre-passo menor que 50%, com margem de fase maior que 45° e larga margem de ganho. O segundo objetivo foi o de se garantir a não-saturação do sinal de controle, evitando assim a utilização de um controlador com *antiwindup*, deixando desta forma a ação do controlador ainda dentro da teoria de sistemas lineares durante toda a atuação, mantendo-se as previsões sobre o comportamento do sistema ainda válidas, para que, após, se pudesse realizar comparações com os outros métodos de controle equanimente.

Para um sistema de controle em malha fechada, com uma planta de segunda ordem, LTI e que não possui sobre-passo maior que 100%, tem-se pela teoria de controle que o máximo sinal de erro entregue ao controlador acontece no instante em que o sinal de entrada é aplicado. Pelo teorema do valor inicial, conclui-se que durante os instantes iniciais o maior sinal de controle é gerado, e que, para um controlador PI, este valor deve ser aproximadamente dado pelo produto entre ganho proporcional e o sinal de referência. Como se desejava que na pior condição não houvesse saturação no atuador, ou, equivalentemente, que o sinal de controle fosse menor que 1, definiu-se que a condição expressa em (3.18) deveria ser cumprida.

$$K_p r < 1 \quad (3.18)$$

Portanto, aplicando-se uma referência de $2263,18 \text{ RPM}$ ($237,00 \frac{\text{rad}}{\text{s}}$) (valor de referência r observado ao se aplicar um ciclo de trabalho de 0,75 no sistema), escolheu-se um K_p menor que $4,30 \cdot 10^{-4} \text{ RPM}^{-1}$. Com o ganho proporcional estabelecido através de seu limite, procurou-se escolher um zero real para o controlador capaz de gerar um sinal de controle com o máximo valor o mais próximo possível da saturação, para que desta forma se pudesse obter o menor tempo de acomodação. Considerou-se que esta condição estava satisfeita, após a escolha de um zero com um valor de 25 s^{-1} e um ganho K_p de $2,47 \cdot 10^{-4} \text{ RPM}^{-1}$. O LGR resultante ficou como mostrado na Figura 24.

Figura 24 – Projeto LGR para o PID



Fonte: do Autor

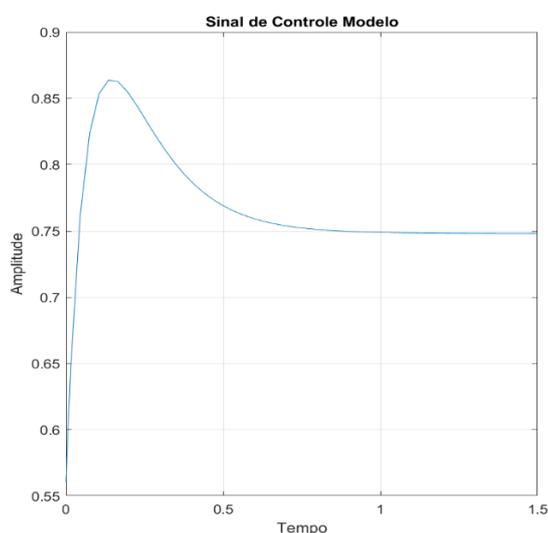
Com os parâmetros do controlador definidos e o modelo da planta conhecido, calculou-se $L(s) = G(s)C(s)$ e verificou-se as margens de estabilidade do sistema. A

validação da estabilidade do controlador foi feita por análises gráficas através do diagrama de Bode, do qual se obteve uma margem de fase de 83° e uma margem de ganho infinita.

A estimativa do tempo de acomodação e do sobre-passo do modelo foram obtidas de um ensaio ao degrau, o qual retornou valores de $0,39s$ e 0% , para o tempo de acomodação e o sobre-passo, os quais ficaram dentro dos objetivos de projeto.

Por fim verificou-se a saturação do sinal de controle. Isto foi feito traçando-se a curva do sinal de controle e observando, para uma referência de $2263,18RPM$, se o mesmo se mantinha abaixo do valor 1. O resultado gráfico para o sinal de controle está mostrado na Figura 25.

Figura 25 – Sinal de controle projetado



Fonte: do Autor

A implementação do controlador projetado, deu-se pela utilização do bloco *Discrete PID Controller*. Neste bloco descreveu-se a versão digitalizada do controlador projetado. Esta digitalização foi realizada pela aplicação do método de Foward-Euller, o qual retornou um controlador conforme mostrado em (3.19),

$$C(z) = P + I \frac{T}{(z-1)} \quad (3.19)$$

onde P é igual ao ganho proporcional do controlador PI projetado ($P = K_p$), e I igual ao produto do ganho proporcional pelo zero ($I = K_p \text{zero}$). Portanto o controlador ficou equacionado conforme mostrado em (3.20).

$$C(z) = 2,47 \cdot 10^{-4} + 0,0062 \frac{50 \cdot 10^{-6}}{(z-1)} \quad (3.20)$$

3.1.2.3 Simulações do Processo em Malha Fechada com PLL

Conforme mencionado na seção 2.4, o sensor utilizado para converter o sinal de velocidade do motor em fase é o *encoder*. A transformação entre estes sinais é realizada pelo *encoder* através da relação dada em (2.44), entretanto o sinal de fase obtido é visto somente pelo bloco PFD, não sendo ele medido de fato, por consequência, não se torna possível a sua utilização diretamente na identificação do sistema. Como a aquisição deste sinal não é de fácil implementação, optou-se por utilizar o sinal de frequência já obtido e transformá-lo em um sinal de fase ao fazê-lo passar por um bloco de *encoder* isolado. Neste caso, considerou-se um *encoder* com uma única ranhura, gerando na sua saída uma onda quadrada com fase e frequência idênticas às do rotor, portanto, escolheu-se um bloco integrador para representá-lo.

Utilizando-se o modelo do encoder, a mesma função de transferência adquirida para modelar a planta no controle por PID pôde ser aplicada no controle por PLL, não se fazendo necessária, portanto, uma nova identificação. Tendo definido o bloco do *encoder*, considerou-se o modelo da planta como sendo o modelo dado em (3.17) em série com um integrador de ganho unitário, desta forma, a função de transferência para a região de trancamento ficou conforme (3.21).

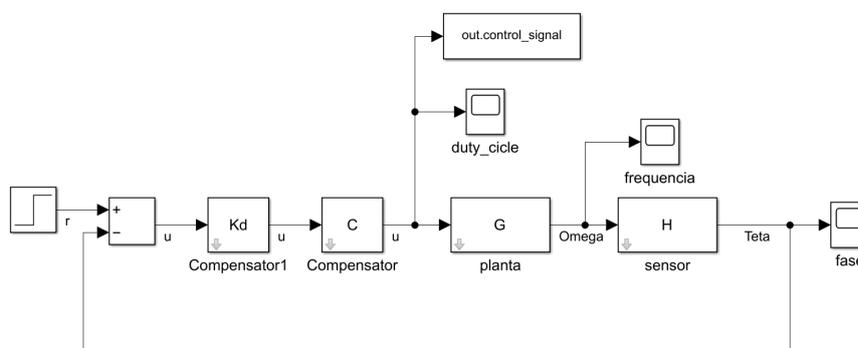
$$\frac{54.10^4 s + 3,05.10^5}{s(s^2 + 27,71s + 101)} \quad (3.21)$$

Como, quando o sistema PLL está atuando na região de trancamento, o bloco PFD se comporta como um bloco somador em série com um amplificador, o qual realiza a diferença dos sinais de fase do sinal de referência e do *encoder* e o amplifica, e, ainda, como este sinal é entregue a um filtro, responsável por alimentar um bloco VCO ou semelhante, tem-se que o filtro desempenha o mesmo papel de um controlador em um sistema realimentado, sendo portanto o seu projeto responsável por alterar a dinâmica do sistema conforme o desejado. Uma Figura ilustrando a arquitetura do sistema, quando a variável fase é analisada, está apresentada em 26.

Conforme discutido em (BEST, 2003), os filtros de sistemas PLL podem ser de diversos tipos e apresentarem funções de transferência de diferentes ordens. No controle de motores é largamente empregada a utilização de controladores PI devido ao seu bom resultado. Com a finalidade de se implementar o mesmo tipo de controle, e também para fins de comparação, optou-se pela utilização de um filtro PI no projeto.

Portanto, assim como feito para o controlador PI, o projeto do filtro PI pôde ter os seus parâmetros escolhidos utilizando-se o método do lugar geométrico das raízes, com o auxílio da ferramenta *rltool*.

Figura 26 – Arquitetura do processo em malha fechada do ponto de vista da variável fase



Fonte: do Autor

No projeto do filtro, definiu-se como objetivos de controle um tempo de acomodação para o controle de fase menor que o tempo de acomodação da frequência apresentado em malha aberta; uma resposta com sobre-passo menor que 50%; e um sistema com margem de fase maior que 45° e margem de ganho significativa.

Também almejou-se um sinal de controle que não provocasse saturação no atuador. Nesta etapa, alguns cuidados diferentes dos tidos para um sistema de controle convencional tiveram de ser tomados. Primeiramente dividiu-se o comportamento do bloco PFD em dois blocos, um bloco somador e outro amplificador. Esta separação teve como objetivo deixar a arquitetura do sistema com a forma padrão, na qual o caminho direto é constituído pelos blocos após o somador indo até o sinal de saída. Pois, desta forma, pôde-se incluir o efeito amplificador do bloco PFD no modelo, fazendo com que este pudesse ser compensado no projeto do filtro PI.

Para que este ganho fosse incluso no modelo, teve de se obter o seu valor. Conforme mostrado em (BEST, 2003), o ganho de um bloco PFD é dado pela expressão (3.22),

$$K_d = \frac{U_b}{4\pi} \quad (3.22)$$

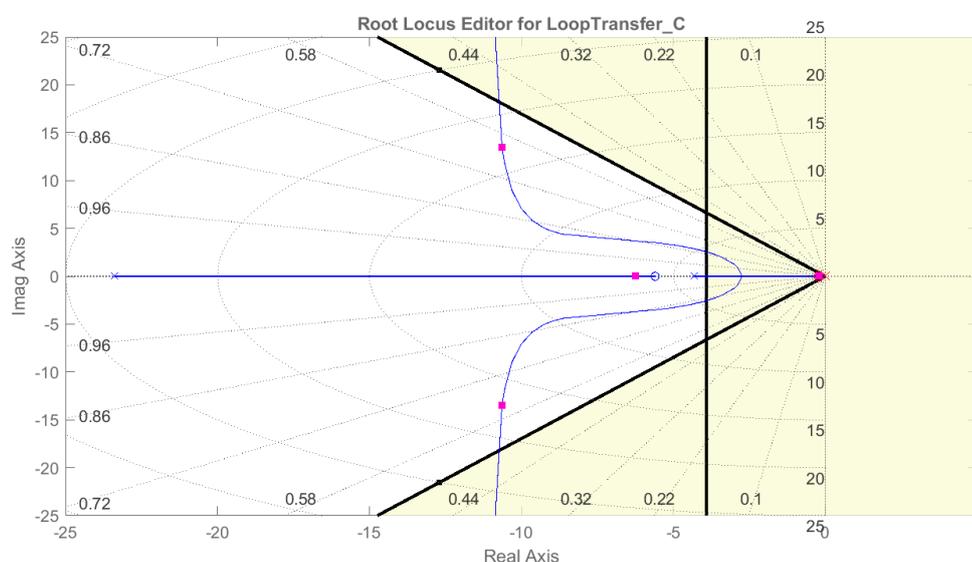
onde U_b é a alimentação do circuito *Charged Pump*, o qual representa o valor da amplitude do sinal de saída, sendo este igual a diferença entre os valores de alimentação presentes nos estados +1 e -1.

Devido a presença das não-linearidades do atuador e do PFD, a escolha da alimentação foi feita de tal forma a contorná-las. A tentativa de se evitar os efeitos destas não-linearidades deu-se por uma operação em pequenos sinais, situação em que se optou por deixar o sinal de controle próximo da região de atuação, somando-se a ele um valor de ciclo de trabalho constante no meio da faixa de operação, que neste caso foi de 0, 75.

Também procurou-se diminuir o ganho do sinal de erro para que o sistema atuasse em uma faixa menor. Por este motivo, escolheu-se uma alimentação de 0,15V, retornando um ganho relativamente pequeno, de 0,012.

Com o ganho do PFD definido, incluiu-se-o no modelo do processo e utilizou-se a ferramenta *rltool* para encontrar os parâmetros do filtro PI. No projeto LGR, escolheu-se os parâmetros com os seguintes valores: um zero em $0,20s^{-1}$ e um ganho K_p de 4,90. Ao fim do projeto, o LGR ficou conforme mostrado na Figura 27.

Figura 27 – Projeto LGR para o PLL



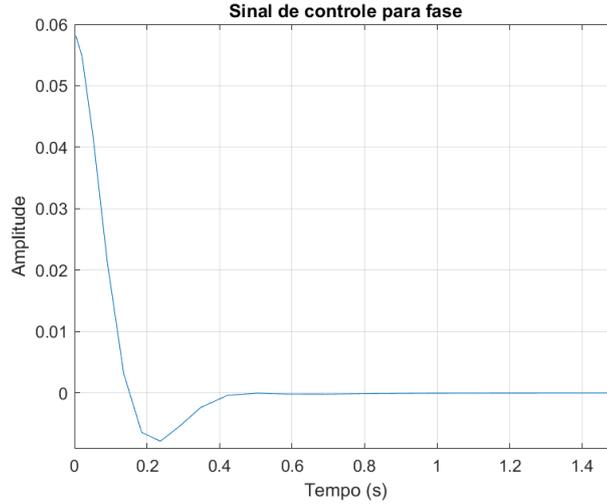
Fonte: do Autor

Com o controlador aplicado ao modelo, obteve-se um tempo de acomodação de 0,41s, um sobre-passo de 15,20%, uma margem de ganho infinita, e uma margem de fase de 55,30°. O sinal de controle, por se tratar de uma operação de pequenos sinais, manteve-se longe da saturação, com valor máximo de 0,06. Uma Figura contendo o sinal de controle é apresentada em 28.

Para a validação do projeto do filtro, ainda teve-se de testar a estabilidade do sistema com este operando fora da região de trancamento. Nesta situação a variável de interesse passou a ser a frequência, e o modelo do PFD passou a ser não-linear e desconhecido.

Sistemas com não-linearidades identificadas somente na região entre os blocos do somador e da planta são conhecidos como sistemas do tipo Lur'. Uma explicação de como aplicar uma análise de Lyanupov a um sistema com estas características, as mesmas do sistema estudado neste trabalho, estão dadas no apêndice C, onde chega-se em um critério gráfico para estabilidade conhecido como Critério de Popov.

Figura 28 – Sinal de controle para a variável fase



Fonte: do Autor

A aplicação do Critério de Popov, mostrado em (C.21), ao sistema PLL, deu-se pela obtenção da função de transferência do caminho direto do sistema, $GH(j\omega) = C(j\omega)G(j\omega)H_e(j\omega)$. Para se aplicar o critério mostrado em C, reescreveu-se esta expressão em função de sua parte imaginária e real, simplificando-a conforme (3.23), onde K é definido em (C.12).

$$\operatorname{Re}\{GH(j\omega)\} - \beta\omega\operatorname{Im}\{GH(j\omega)\} + \frac{1}{K} \geq 0 \quad (3.23)$$

Com o critério de Popov definido, pôde-se verificar que, para uma constante K pertencente ao domínio $[0, \infty]$, a pior situação, do ponto de vista da estabilidade, ocorre com K igual a ∞ . Nesta condição tem-se que o sistema é estável, desde que se garanta que a razão $\frac{\operatorname{Re}\{GH(j\omega)\}}{\omega\operatorname{Im}\{GH(j\omega)\}}$ esteja à direita de uma reta com inclinação $\frac{1}{\beta}$.

Para o sistema projetado, chegou-se em uma $GH(s)$ dada pela expressão (3.24)

$$GH(s) = K_p \frac{M}{2\pi N} \frac{[(as + b)(s + z)]}{s(s^2 + cs + d)} \quad (3.24)$$

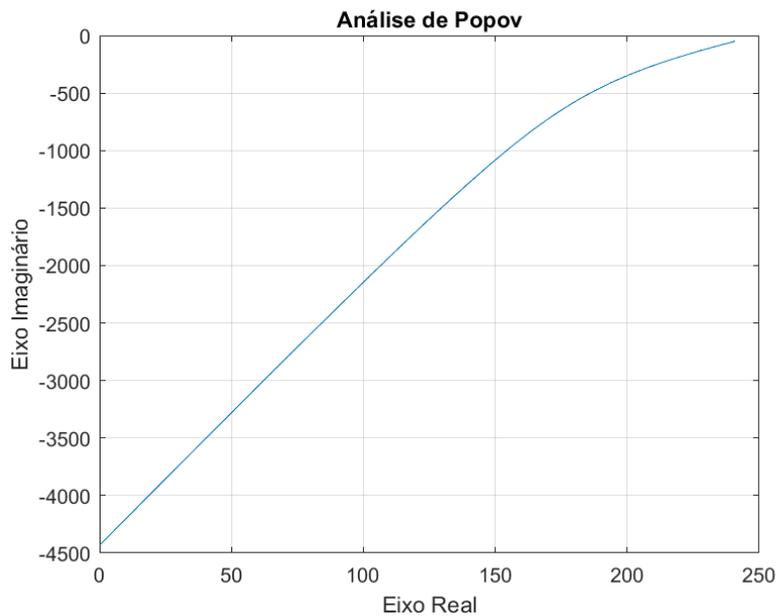
onde z é o zero e K_p o ganho do filtro PI; a, b, c e d são os coeficientes da função de transferência da planta, N é o número do divisor de frequência; e M é o número de ranhuras do encoder. Em função destes parâmetros, pôde-se separar $GH(j\omega)$ em suas parcelas reais e imaginária (multiplicada por ω), conforme (3.25) e (3.26).

$$X = K_p \frac{\omega^2(ac - az - b) + (-bcz + azd + bd)}{\omega^4 + \omega^2(c^2 - 2d) + d^2} \quad (3.25)$$

$$Y = K_p \frac{\omega^4(-a) + \omega^2(da + bz - caz - cb) + (-dbz)}{\omega^4 + \omega^2(c^2 - 2d) + d^2} \quad (3.26)$$

Conhecendo-se estas parcelas, pôde-se reescrever a inequação (3.23) na forma $X - \beta Y + \frac{1}{K}$, com $X = \text{Re} \{GH(j\omega)\}$ e $Y = \omega \text{Im} \{GH(j\omega)\}$. Dada esta nova interpretação, parametrizou-se X e Y em função da frequência ω , chegando-se na curva mostrada na Figura 29.

Figura 29 – Curva de Popov



Fonte: do Autor

Através da curva mostrada em 29, vê-se que, para quaisquer valores de ω , variando de zero a infinito, esta não se encontra fora do quarto quadrante. Isto implica que, independente do β apresentado pelo sistema, em qualquer frequência, o critério de Popov é atendido. Desta forma, pôde-se garantir um sistema estável fora da região de trancamento com a utilização do filtro projetado.

A implementação do filtro projetado em simulação, deu-se pela utilização do bloco *Discrete PID Controller*. Neste bloco utiliza-se a versão digitalizada do filtro. Esta digitalização foi realizada pela aplicação do método de Foward-Euller, o qual retornou um filtro conforme a expressão (3.27)

$$F(z) = P + \frac{IT}{z - 1} \quad (3.27)$$

onde P é igual ao ganho proporcional do filtro PI projetado ($P = K_p$), e I igual ao produto do ganho proporcional pelo zero ($I = K_p z$). Desta forma o filtro ficou equacionado conforme (3.28).

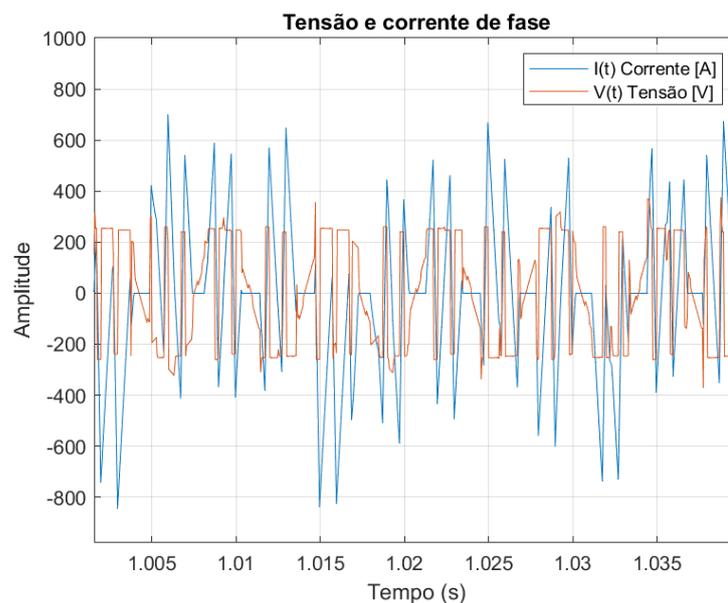
$$F(z) = 2,47 \cdot 10^{-4} + \frac{(0,0062)(50 \cdot 10^{-6})}{z - 1} \quad (3.28)$$

A implementação do *encoder* na simulação, deu-se pela utilização do bloco *Incremental Shaft Encoder* pertencente a biblioteca *Simscape*, o qual teve como função receber um sinal vindo do rotor, e transformá-lo em uma onda quadrada com M vezes a frequência de rotação do motor, e com a amplitude desejada. Na simulação, optou-se por deixar ambos valores unitários.

3.1.2.4 Simulações do Processo em Malha Fechada com Carga

Para analisar a capacidade dos sistemas estudados em eliminar o distúrbio de saída, optou-se por simular uma adição de carga ao rotor na forma de torque, conforme discutido na seção 2.1.3. A escolha da ordem de grandeza da carga aplicada deu-se pela estimativa da potência média do motor, com este operando em uma velocidade média de $237 \frac{rad}{s}$, em regime permanente. Nesta condição, obteve-se os sinais de corrente e tensão de uma das fases do inversor-trifásico. As formas destes sinais estão mostradas na Figura 30.

Figura 30 – Tensão e corrente de fase



Fonte: do Autor

Em posse dos valores de tensão e corrente, utilizou-se a expressão (3.29) para se calcular a potência média, a qual retornou um valor de $11,38kW$.

$$P_{media} = \frac{3}{T - T_0} \int_{T-T_0}^T I(t)V(t)dt \quad (3.29)$$

Desprezando-se todas as perdas, e considerando que toda a potência entregue ao motor foi transformada em energia cinética para o rotor, utilizou-se a relação dada em 2.13, de onde obteve-se um torque médio de $104,39Nm$ para o motor. Trabalhando-se com esta ordem de grandeza, após alguns testes, empiricamente, optou-se por realizar os ensaios de rejeição ao distúrbio com uma carga de $500Nm$ aplicada ao rotor.

A implementação da carga, através de simulação, deu-se pela utilização do bloco *Ideal Torque Source*.

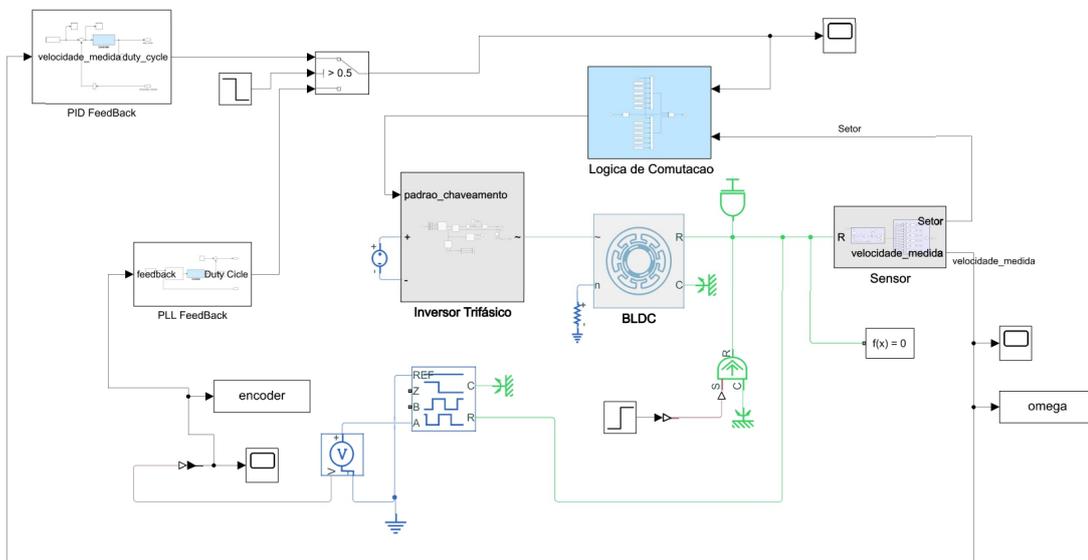
3.1.2.5 Simulações do Processo em Malha Fechada com dual-PLL

Uma investigação sobre a técnica dual-PLL também foi realizada. Nesta técnica são misturadas duas formas de controle, visando unir as melhores características de cada uma. O dual-PLL investigado neste trabalho é composto pelos métodos de controle PID e PLL, mantendo os controladores projetados apresentados nas seções anteriores, objetivando-se um controle rápido como o controle PID e que tenha a precisão no segmento de referência obtida por um controle PLL.

A implementação da simulação deste sistema deu-se pela criação de dois caminhos de realimentação e pelo chaveamento dos sinais de controle. O caminho de realimentação foi composto, no caso do controle PID, pelo bloco somador e o bloco controlador, e, no caso do controle PLL, pelo blocos PFD, filtro PI e *encoder*. Os sinais de referência em ambos os casos foram elaborados para manter o rotor girando com uma velocidade de $237 \frac{rad}{s}$. Para o controle PID, gerou-se um sinal do tipo degrau com um valor constante de 237 a partir do tempo zero. Para o controle PLL, gerou-se uma onda quadrada de amplitude unitária com a frequência de $237 \frac{rad}{s}$. Os blocos de mais alto-nível que compõem o sistema simulado no Simulink® estão mostrados na Figura 31. Os blocos que compõem a realimentação do controle por PLL estão mostrados na Figura 32.

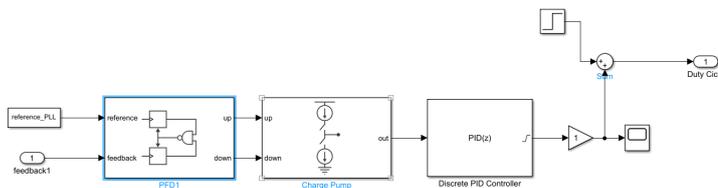
A realização do chaveamento para um dual-PLL pode ser feita de diferentes formas. Uma das formas mais encontradas é o chaveamento por região de operação, onde troca-se o método de controle quando o sinal de saída se encontra dentro de uma determinada região de velocidade, deixando desta forma o chaveamento mais generalista, sem depender de características intrínsecas do sistema. Neste trabalho, por praticidade, optou-se pelo chaveamento temporal, onde procurou-se deixar o sistema operando com um controle PID durante um tempo igual ao seu tempo de acomodação, trocando-se após o sinal de controle pelo gerado no controlador PLL.

Figura 31 – Sistema dual-PLL



Fonte: do Autor

Figura 32 – Blocos de realimentação PLL



Fonte: do Autor

3.2 Experimentos

Neste trabalho, além de se verificar a comparação dos métodos de controle PID e PLL à nível de simulações, buscou-se o mesmo em experimentos.

Para a realização dos experimentos, utilizou-se um motor BLDC do tipo A2212/6T 2200KV com o seu eixo acoplado a um motor CC do tipo Buehler Optically encoded servo

motor. Este acoplamento possibilitou a utilização do encoder Q9898 que contém 1024 ranhuras. À sua escolha também atribui-se a tentativa de se gerar uma carga no eixo do BLDC, com o motor CC atuando como um gerador.

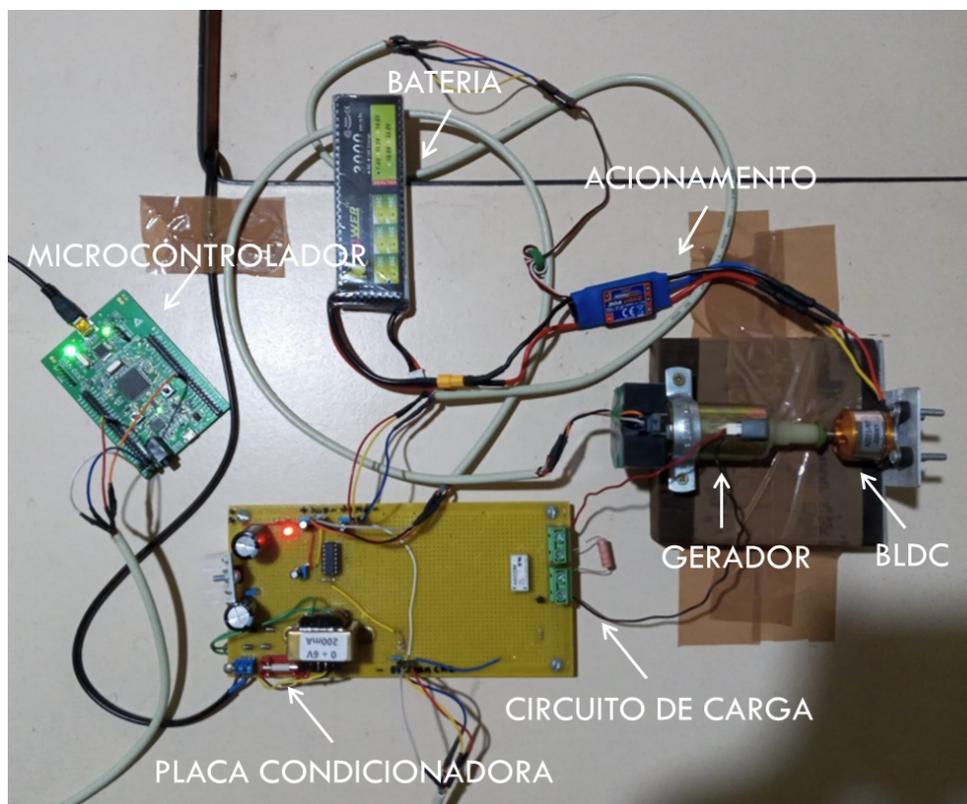
Visando implementar os blocos somador e PFD, assim como o controlador e o filtro PI, e, ainda, para se obter o sinal do *encoder* e controlar o acionamento, utilizou-se a placa de desenvolvimento STM32F4X-DISCOVERY contendo o microcontrolador STM32F407VET6.

Com o intento de condicionar os sinais e conectar uma carga ao gerador, confeccionou-se um circuito contendo um *buffer* do tipo SN74LS07N para o sinal de *encoder*, e um relé V23105 ativado por um transistor BC237B.

Para acionar o motor, utilizou-se um ESC UBEC 30A energizado por uma bateria Li-Po de duas células, o qual foi programado conforme o manual (HOBBYKING, 2022a) para operar com as configurações padrões de fábrica, alterando-se o range de acionamento para uma escala completa.

Uma imagem da bancada contendo todo o sistema experimental é mostrada na Figura 33.

Figura 33 – Sistema experimental



Fonte: do Autor

3.2.1 Experimentos Para Identificação

Para se obter os valores de velocidade, utilizou-se a arquitetura de *timer* inclusa no microcontrolador escolhido, configurada para receber em sua entrada de *clock* o sinal provindo do *encoder*. Para que a aquisição deste sinal pudesse ocorrer de forma a respeitar os critérios de amostragem, optou-se por gerar uma interrupção de contador com uma fração do número de ranhuras. Como pré-escala deste contador, procurou-se empregar um número baixo, porém não tão baixo a ponto de se perder a precisão na contagem do tempo decorrido entre duas interrupções. Por esta razão, optou-se por gerar uma interrupção a cada 32 bordas de subida do sinal de *encoder*. Desta forma, pôde-se estipular a velocidade angular como sendo a razão entre um 32 avos da frequência de revolução do rotor (32/1024) pelo tempo decorrido entre duas interrupções, utilizando uma precisão de $1\mu s$ para o seu cálculo.

Para controlar e programar o ESC, responsável pelo acionamento do motor, gerou-se um sinal PWM através do microcontrolador de forma a emular o comportamento do controle HK-7X. Conforme indicado no manual do HK-7X (HOBBYKING, 2022b), o sinal de controle tem de operar como um PWM com período de $20ms$ e um ciclo de trabalho variando entre 0,05 e 0,1, a depender do comando que se deseja enviar. Para o envio do sinal de controle, que controlará o PWM responsável pelo acionamento do motor, tem-se uma zona de atuação, entre os valores de ciclo de trabalho, de 0,05 a 0,1, caracterizando-se assim um atuador com saturação. Para que esta faixa operasse como um ciclo de trabalho de forma direta, transformou-se-a em uma faixa de valores com amplitude entre 0 e 1.

Na escolha do tempo de amostragem, procurou-se levar em consideração tanto o atendimento do critério de Nyquist como o tempo de atualização do sinal. Para que o critério de Nyquist seja atendido, faz-se necessário a utilização de um tempo mínimo de amostragem duas vezes menor que o período da frequência natural do sistema. Como previamente sabe-se que a frequência natural do sistema é da ordem de dezenas, a escolha de um período de amostragem com centenas de microssegundos deveria garantir o seu cumprimento. Para que o tempo de atualização do sinal seja menor que o seu tempo de amostragem, optou-se pela execução de sua medida a cada 32 ranhuras, o que para um motor com velocidade angular na casa dos milhares de radianos por segundo, deveria retornar em regime permanente um intervalo de atualização na casa das dezenas de microssegundos. Portanto, levando-se estas duas condições em consideração, escolheu-se um tempo de amostragem de $500\mu s$.

Para o método de controle PID, considerou-se a planta como sendo o conjunto de blocos que têm como sua entrada o sinal de ciclo de trabalho e a sua saída a velocidade do rotor. Como, para o método de controle por PLL dentro da região de trancamento, a variável controlada é a fase, considerou-se a planta como sendo aquela obtida para a

frequência com um terceiro bloco em série, neste caso, considerou-se um bloco integrador na sequência do bloco BLDC.

Assim como feito para a planta simulada, a identificação do modelo, deu-se pela geração e coleta dos sinais de entrada e saída, ambas efetuadas por um *script* do Matlab®. No *script*, estes dados foram encapsulados em um objeto do tipo *iddata*, utilizado como entrada de ferramentas de estimativa de modelo. Através destas ferramentas, pôde-se realizar comparações entre as respostas geradas pelo modelo e pelo experimento, permitindo-se verificar a qualidade do encaixe. As estimativas de modelo foram efetuadas no *script* através da função *tfest*.

Para realizar ensaios ao degrau, implementou-se um *firmware* responsável por gerar um sinal PWM com ciclo de trabalho constante, sendo este ativado a partir do tempo zero e enviado para o motor com um tempo de amostragem de $500\mu s$ durante um intervalo de 1,5s.

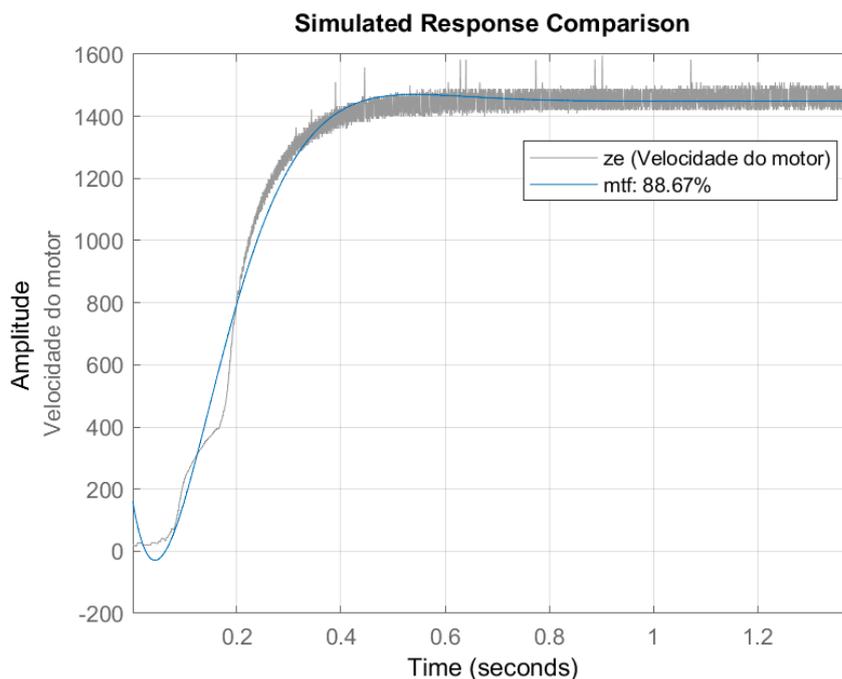
A identificação da região de operação do motor, deu-se pela busca e alteração das faixas de operação do motor de forma empírica. A alteração da faixa de operação foi feita de forma a deixar a escala do ciclo de trabalho, a qual variava de 0,05 a 0,1, com valores entre 0,1 e 1. Tendo-se feito esta transformação, escolheu-se um valor de 0,6 para o degrau de ciclo de trabalho, pois, ao optar-se por um valor próximo do valor central, dentro da faixa de operação, obteve-se a máxima amplitude de atuação.

Assim como feito para a simulação, o intuito da realização do ensaio ao degrau teve como objetivo a obtenção do tempo de acomodação da planta para que após se pudesse realizar o ensaio PRBS. Pois, como mencionado na seção 2.2.2, o ensaio PRBS é mais desejado quando a planta envolve motores.

Para se adquirir o tempo de acomodação, utilizou-se a função *tfest* configurada para a obtenção da estimativa do modelo por uma função de transferência de primeira ordem, com um pólo e nenhum zero. A escolha deste tipo de modelo foi feita considerando-se o rotor do BLDC ideal, sem momento de inércia, $J = 0$, o que, conforme a expressão (2.23), retornou uma função de transferência de primeira ordem. Com a utilização da função *tfest*, obteve-se um modelo com 88,67% de encaixe. As respostas ao degrau do sistema experimental e de seu modelo podem ser vistas na Figura 34, da qual se pode medir um tempo de acomodação de 0,36s para uma variação no sinal de saída menor que 2%.

Assim como feito para o sistema simulado, a geração do sinal PRBS, deu-se pela utilização da função *idinput*. Para a utilização desta função, além de se indicar o tipo, é necessário informar o número de amostras e, no caso do sinal PRBS, também, selecionar os valores de amplitude e a banda de operação. Os valores de amplitude foram escolhidos conforme o conhecimento empírico do projetista sobre o sistema. Realizando-se

Figura 34 – Resposta ao degrau em malha aberta



Fonte: do Autor

experimentos com ensaio PRBS, chegou-se em melhores resultados com a utilização de amplitudes de ciclo de trabalho com valores de 0,45 e 0,75.

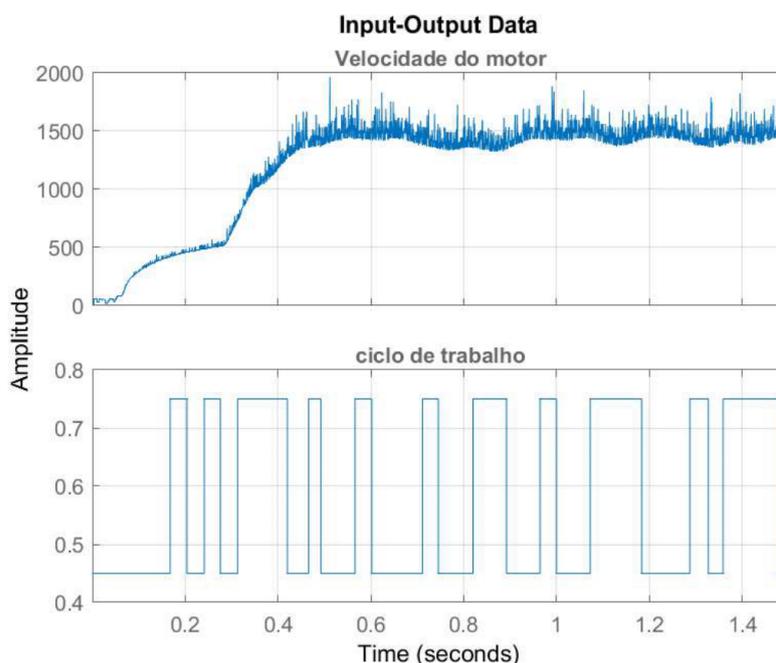
Com o sinal PRBS criado, realizou-se o seu ensaio experimentalmente. Após a realização do ensaio, coletou-se novamente os sinais de saída e entrada. O resultado do ensaio para a saída junto de sua entrada estão expostos na Figura 35.

Tendo-se adquirido os dados de entrada e saída, construiu-se novamente o objeto *iddata*, para utilizá-lo na identificação do sistema, através da função *tfest*.

Para a resposta experimental também buscou-se a melhor estimativa, verificando-se diferentes ordens para o modelo, obtendo-se aproximações para plantas com 2 polos e nenhum zero; 2 polos e 1 zero; 3 polos e 2 zeros; e 4 polos e 3 zeros. Estes modelos retornaram respectivamente os seguintes percentuais de encaixe: 81,04%, 81,04%, 79,33% e 84,06%. Como os modelos obtidos para a planta experimental tiveram performances semelhantes, adotou-se o modelo com dois pólos e nenhum zero para se evitar maiores complexidades.

A comparação da resposta experimental, em conjunto com a resposta do modelo, quando aplicado o sinal PRBS, para o modelo adotado, está exposta na Figura 36.

Figura 35 – Ensaio PRBS experimental



Fonte: do Autor

3.2.2 Experimento do Processo em Malha Fechada com PID

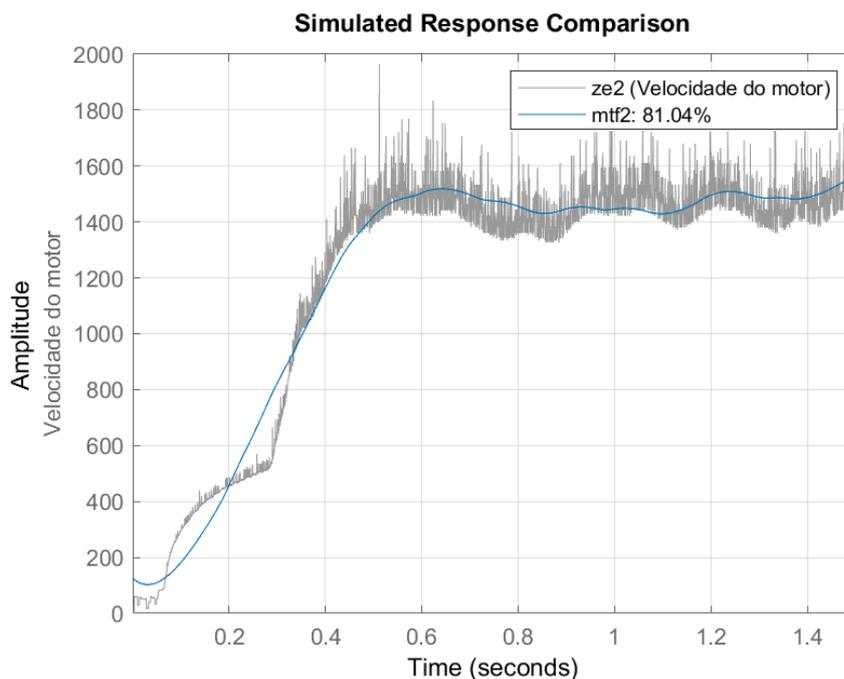
A função de transferência estimada para a planta, definida como tendo dois pólos, cujo encaixe do modelo retornado foi de 81,04%, foi a escolhida para modelar o sistema e executar o projeto do controlador. Os valores encontrados para estes pólos foram de $[-4,65 \pm 4,24i]s^{-1}$. A versão final da função de transferência, com o ganho incluso, é mostrada na expressão (3.30).

$$\frac{1,004 \cdot 10^5}{s^2 + 9,319s + 39,73} \quad (3.30)$$

O projeto do controlador PI foi realizado pelo método do lugar geométrico das raízes com auxílio da ferramenta *rltool*. Para o projeto experimental também se estabeleceu alguns objetivos. Primeiramente, definiu-se que o projeto do controlador deveria entregar um sistema em malha fechada com tempo de acomodação menor que o observado em malha aberta, com margem de fase maior que 45° e larga margem de ganho. O segundo objetivo foi o de garantir a não-saturação do integrador, evitando assim a utilização de um controlador com *antiwindup*.

Para contornar a saturação do atuador, escolheu-se o ganho proporcional do controlador de forma a respeitar a condição (3.31).

Figura 36 – Resposta para a entrada PRBS experimental



Fonte: do Autor

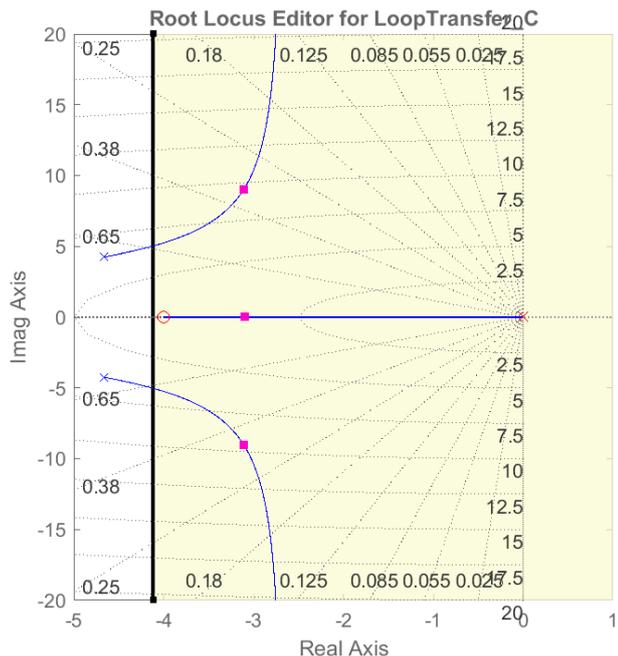
$$K_p r < 1 \quad (3.31)$$

Portanto, ao se aplicar uma referência de $1460 \frac{rad}{s}$ no sistema, teve-se de escolher um K_p menor que $6,85 \cdot 10^4 \frac{s}{rad}$. Com o limite para o ganho proporcional estabelecido, procurou-se escolher um zero real capaz de gerar um sinal de controle com o máximo valor o mais próximo possível da saturação, para que desta forma se pudesse obter o menor tempo de acomodação. Considerou-se que esta condição estava satisfeita após escolher-se um zero com um valor de $4,00 s^{-1}$ e um ganho K_p de $7,00 \cdot 10^4 \frac{s}{rad}$. Aqui, cabe observar que, devido a uma dinâmica mais lenta observada por parte da saída da planta, nos instantes iniciais, propositalmente aumentou-se a atuação do sistema, a fim de agilizar a dinâmica. O LGR projetado está mostrado na Figura 37.

Com os parâmetros do controlador definidos e o modelo da planta conhecido, calculou-se $L(s) = G(s)C(s)$ e verificou-se as margens de estabilidade do sistema. A validação da estabilidade do controlador foi feita por análises gráficas através do diagrama de Bode, de onde se obteve uma margem de fase de $45,30^\circ$ e uma margem de ganho infinita.

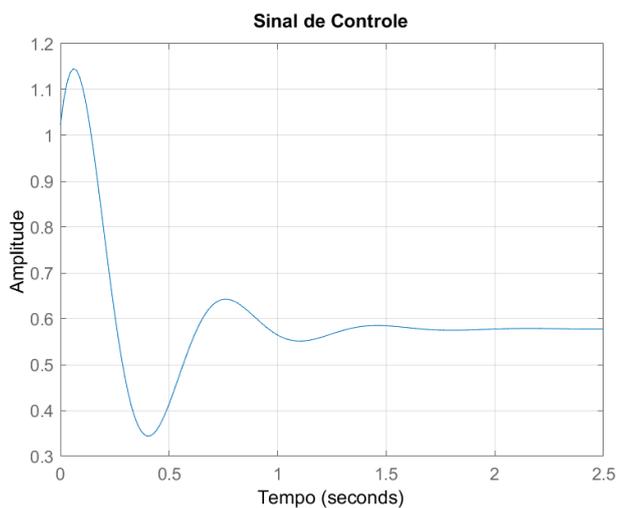
A estimativa do tempo de acomodação e do sobre-passo do modelo foram obtidas de um ensaio ao degrau, retornando valores de $0,90s$ e $17,30\%$, para o tempo de acomodação e o sobre-passo. O sinal de controle ficou conforme se pode ver na Figura 38.

Figura 37 – Projeto do LGR para o PI experimental



Fonte: do Autor

Figura 38 – Sinal de controle PI do modelo experimental



Fonte: do Autor

A implementação do controlador projetado, deu-se em sua forma digitalizada em *firmware*. Esta digitalização foi realizada pela aplicação do método de Backward-Euller, o qual retornou um P igual ao ganho proporcional do controlador PI projetado ($P = K_p$), e I igual ao produto do ganho proporcional pelo zero ($I = K_p zero$), com valor de $28,00m\frac{s}{rad}$. O código utilizado para a realização do controle PID pode ser visto no apêndice D.

3.2.3 Experimento do Processo em Malha Fechada com PLL

De posse do modelo da planta para uma saída de velocidade angular, obteve-se o modelo da planta para uma saída de fase através daquele obtido em 3.30 em série com um integrador de ganho unitário. Desta forma, a função de transferência da planta para a região de trancamento ficou conforme a expressão (3.32).

$$\frac{1,004 \cdot 10^5}{s(s^2 + 9,319s + 39,73)} \quad (3.32)$$

O projeto do filtro PI do sistema PLL, assim como feito para o sistema simulado, foi realizado utilizando-se o método do lugar geométrico das raízes com o auxílio da ferramenta *rltool*.

Para a implementação do sistema experimental também escolheu-se um comparador de fase do tipo IV. Para incluir o seu efeito no sistema, dividiu-se o comportamento do bloco PFD em dois blocos, um bloco somador e outro amplificador. Desta forma, pôde-se deixar a arquitetura do sistema com a forma padrão.

Do mesmo modo que tomou-se cuidado com as não linearidades adicionadas pelo PFD no sistema simulado, fez-se para o sistema experimental, pondo-o para operar em torno de um ponto, e escolhendo um K_d relativamente pequeno, evitando-se assim a saturação do atuador. Para atender este requerimento, escolheu-se um ganho com valor de 0,012.

Com o ganho K_d definido e incluso na planta, utilizou-se a ferramenta *rltool* para encontrar os parâmetros do filtro PI. No projeto LGR, escolheu-se os parâmetros com os seguintes valores: um zero em $1,06s^{-1}$ e um ganho K_p de 0,0687. O projeto do LGR pode ser visto na Figura 39.

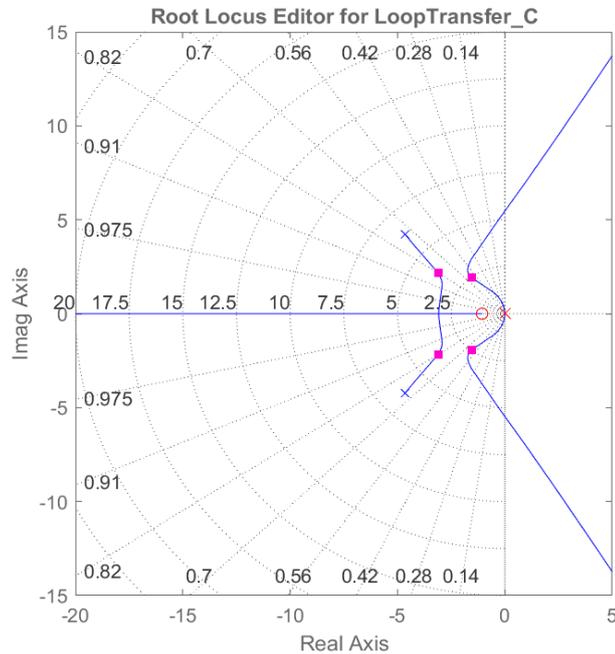
Com o controlador aplicado ao modelo, e com a malha fechada, obteve-se para o sistema um tempo de acomodação de 3,59s, um sobre-passo de 51,30%, uma margem de ganho de 10,6dB, e uma margem de fase de 33,60°. O sinal de controle projetado para a fase ficou como se vê na Figura 40.

A avaliação da estabilidade do sistema fora da região de trancamento para o filtro projetado, deu-se pela análise de Popov. Uma explicação de como aplicar esta análise em um sistema PLL é mostrada em C, onde chega-se em um critério gráfico para estabilidade conhecido como Critério de Popov.

Levando-se em consideração a relação dada em (C), para o sistema projetado, chegou-se em uma função de transferência $GH(s)$ conforme a expressão (3.33).

$$GH(s) = \frac{K_p M[(s+z)]}{2\pi N s(s^2 + as + b)} \quad (3.33)$$

Figura 39 – Projeto do LGR para o PLL experimental



Fonte: do Autor

onde z é o zero e K_p o ganho do filtro PI; a e b são os coeficientes da função de transferência planta; N é o número de divisões do divisor de frequência; e M é o número de ranhuras do *encoder*. Em função destes parâmetros, pôde-se separar $GH(j\omega)$ em suas parcelas reais e imaginária (multiplicada por ω), conforme (3.34) e (3.35), respectivamente.

$$X = \frac{K_p[\omega^2(a - z) + zb]}{\omega^4 + \omega^2(a^2 - 2b) + b^2} \quad (3.34)$$

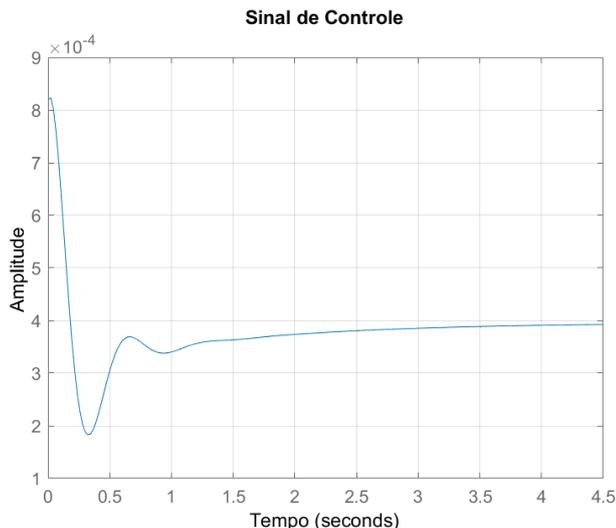
$$Y = \frac{K_p[-\omega^4 + \omega^2(b - az)]}{\omega^4 + \omega^2(a^2 - 2b) + b^2} \quad (3.35)$$

Conhecendo-se estas parcelas, pôde-se reescrever a inequação 3.33 na forma $X - \beta Y + \frac{1}{K}$, com $X = Re\{GH(j\omega)\}$ e $Y = \omega Im\{GH(j\omega)\}$. Dada esta nova interpretação, parametrizou-se X e Y em função da frequência ω , chegando-se na curva mostrada na Figura 41.

Através da curva mostrada em 41, vê-se que o modelo do sistema em malha fechada com o filtro projetado será estável para todas as frequências, desde que este apresente um β menor que 0,63.

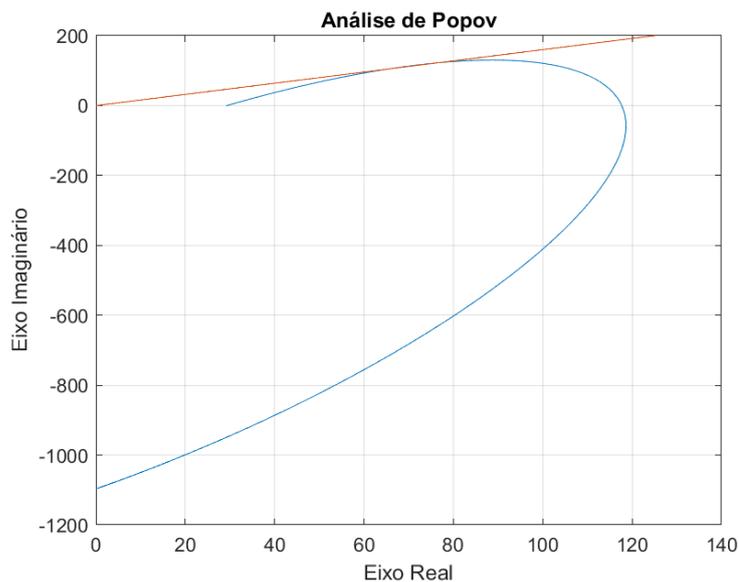
A implementação do sistema com o filtro projetado, deu-se pela descrição dos blocos PFD e filtro PI no microcontrolador. A descrição do bloco PFD foi feita dentro de uma função, implementando o diagrama de estados mostrado na Figura 41. A chamada

Figura 40 – Sinal de controle PLL do modelo experimental



Fonte: do Autor

Figura 41 – Análise de Popov para o sistema experimental



Fonte: do Autor

desta função foi criada para ocorrer a cada borda de subida do sinal de referência ou do sinal do encoder. A sinalização da ocorrência destas bordas foi executada em *firmware* através de interrupções, nas quais as bordas de subida para o sinal de referência foram criadas em interrupções geradas com a frequência de referência, e as bordas de subida do sinal de *encoder* foram criadas a cada 32 interrupções do *timer* conectado ao *encoder*, ou seja, a cada revolução do eixo do rotor.

A implementação do filtro PI, deu-se de forma semelhante à implementação do controlador PI. Executando-se o seu algoritmo em múltiplos do tempo de amostragem adotado. O código utilizado para descrever estes blocos está mostrado no apêndice E.

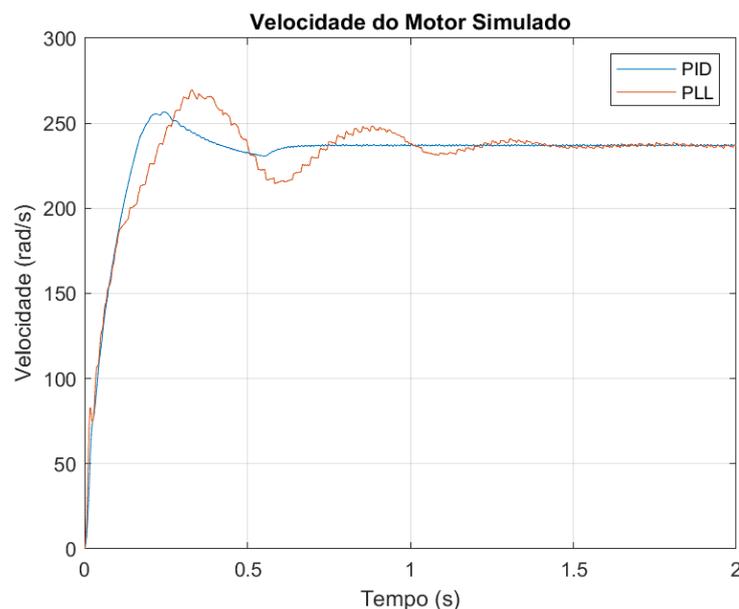
4 RESULTADOS E DISCUSSÕES

4.1 Resultado das Simulações

4.1.1 Resultado das Simulações Sem Carga

Ao simular o ensaio ao degrau, com uma referência de velocidade igual a $237\frac{rad}{s}$, aplicando-se o método de controle PID, e utilizando o controlador PI projetado, obteve-se um sinal de velocidade com valor médio em regime permanente de $237,02\frac{rad}{s}$ com desvio padrão de $0,22\frac{rad}{s}$. Através deste valor, obteve-se um erro menor que $0,01\%$. Ainda, verificou-se um tempo de acomodação de $0,36s$, o qual foi menor que o tempo de $0,46s$ obtido em malha aberta. Um sobre-passo de $8,02\%$ também foi observado. Ao simular o ensaio ao degrau, com uma referência de velocidade igual a $237\frac{rad}{s}$, aplicando-se o método de controle PLL, e utilizando o filtro PI projetado, obteve-se um sinal de velocidade com valor médio em regime permanente de $236,83\frac{rad}{s}$ com desvio padrão de $0,78\frac{rad}{s}$. Através deste valor se obteve um erro menor que $0,1\%$. Ainda, verificou-se um tempo de acomodação de $0,97s$, o qual foi maior que o tempo de $0,46s$ obtido em malha aberta. Um sobre-passo de $13,79\%$ também foi observado. As curvas dos sinais de velocidade mencionados, de onde se obteve os valores citados, estão mostradas na Figura 42.

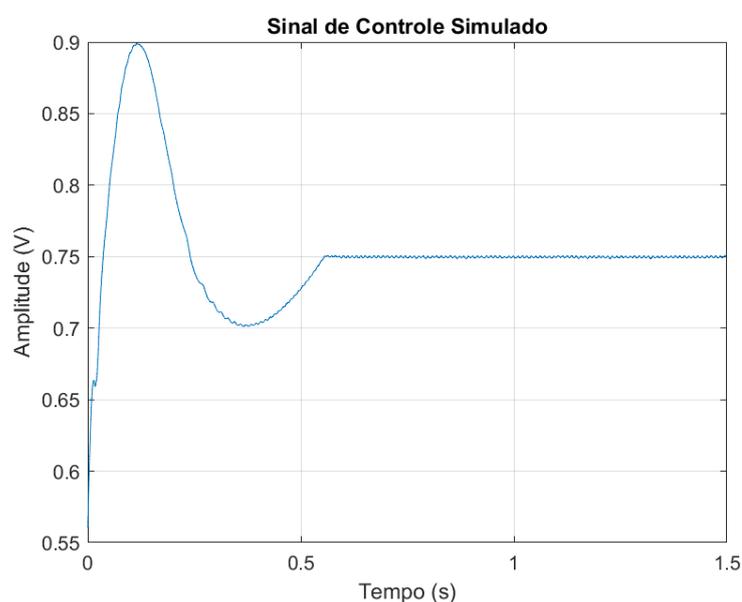
Figura 42 – Resposta das simulações ao degrau



Fonte: do Autor

O sinal de controle, ou ciclo de trabalho, obtido no ensaio ao degrau para o método de controle PID teve um comportamento similar ao projetado. Conforme se pode ver na Figura 25, o sinal de controle projetado teve um pico de 0,86 no instante 0,14s, enquanto que o sinal simulado apresentou o pico de sua curva, no regime transitório, no instante de 0,12s, com valor de 0,9. Portanto o sinal de controle simulado ficou próximo o suficiente do projetado, de maneira a validar o modelo e o projeto do controlador, fazendo com que desta forma se pudesse garantir que o atuador não apresentasse saturação, e tivesse um tempo de acomodação menor do que o de malha aberta. A curva do sinal de controle simulado resultante está mostrada na Figura 43.

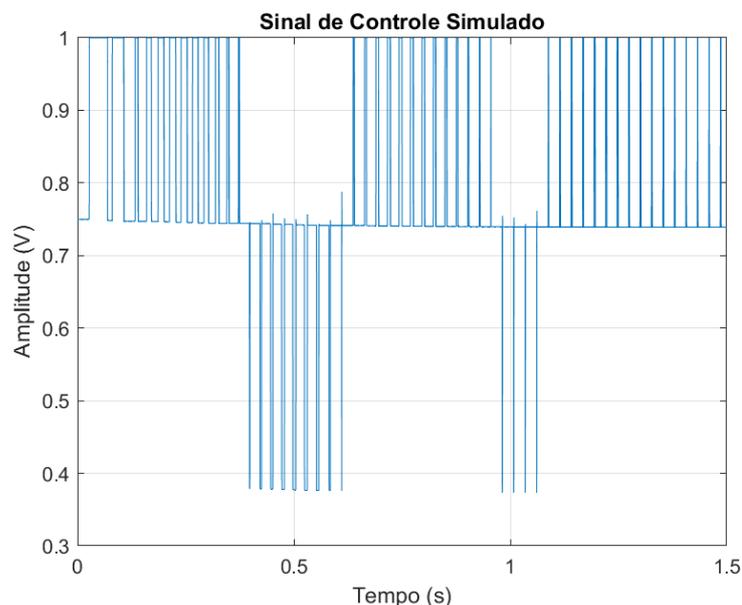
Figura 43 – Sinal de controle do PID simulado



Fonte: do Autor

O sinal de controle obtido para o método de controle PLL teve o seu comportamento conforme o esperado pela teoria, onde o sistema foi posto a operar em uma região de pequenos sinais, mais especificamente ao redor de um ciclo de trabalho de 0,75. O seu comportamento, conforme vê-se pela Figura 44, é o de um sinal periódico com amplitude modulada por largura de pulso, com maiores amplitudes nos instantes iniciais devido a maior necessidade de atuação e pulsos de larguras menores nos instantes finais, onde a necessidade de atuação é menor. Também vê-se que em regime permanente o sinal de saída do filtro adota um comportamento periódico, com sequências de pulsos de espaçamento igual, sendo este um comportamento comumente observado em sistemas PLL operando na região de trancamento.

Figura 44 – Sinal de controle do PLL simulado



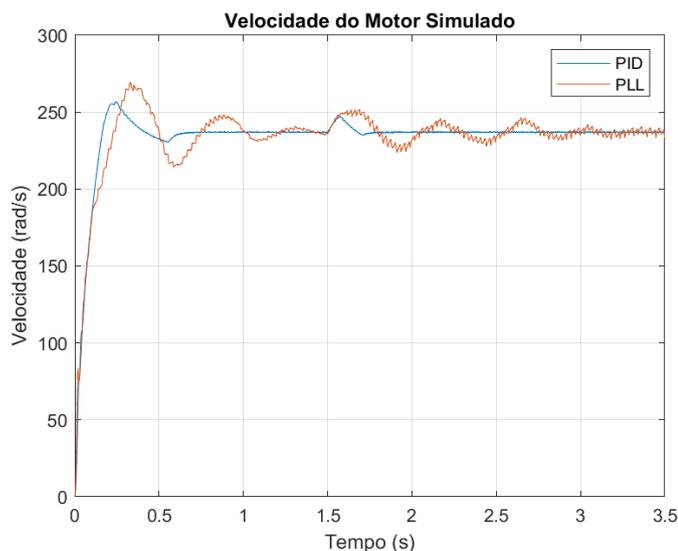
Fonte: do Autor

4.1.2 Resultado das Simulações Com Carga

Para ambos os métodos de controle, avaliou-se a rejeição ao distúrbio aplicando uma carga de $500Nm$ no rotor. Esta carga foi aplicada após o sistema entrar em regime permanente. Para este ensaio, escolheu-se como instante para a aplicação da carga o tempo de um segundo após a aplicação da referência de entrada, tempo suficiente para que os dois sistemas entrassem em regime permanente. Através das curvas para a velocidade do motor, mostradas na Figura 45, pôde-se obter os tempos de rejeição ao distúrbio, com margem de precisão de 2%, para ambos os métodos. Pela Figura 45 vê-se uma rejeição do distúrbio 0,11s após a aplicação da carga para o método de controle PID, e de 1,2s para o método de controle PLL.

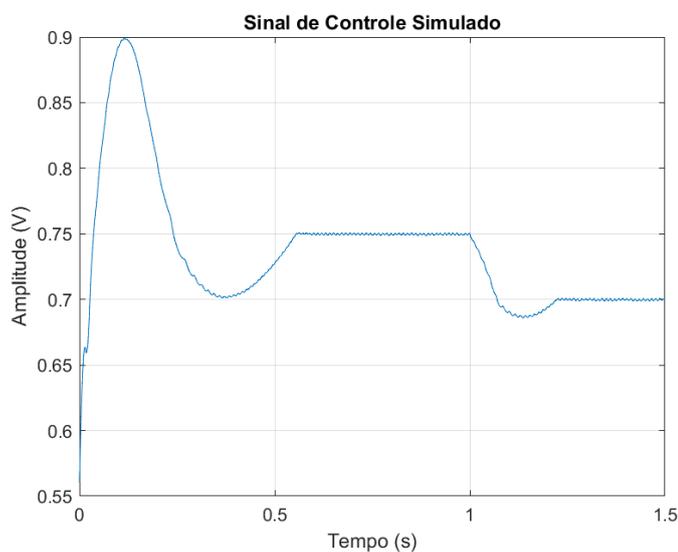
O sinal de controle obtido no ensaio de rejeição ao distúrbio, para o método de controle PID, teve um comportamento condizente ao esperado. Conforme se pode ver na Figura 46, o sinal de controle apresentou o mesmo comportamento que o apresentado no ensaio ao degrau em regime transiente, até o instante em que a carga foi aplicada. Após a aplicação da carga, vê-se que o sinal de controle diminuiu a sua atuação sobre o sistema, já que o torque aplicado teve o mesmo sentido de operação do torque gerado pelo motor, e portanto tendeu a aumentar a velocidade do rotor, o que teve de ser compensado através da redução da potência aplicada ao sistema para que este operasse na mesma velocidade de referência. Também nota-se pela Figura 46 que a dinâmica apresentada pelo PID na rejeição ao distúrbio seguiu o mesmo padrão apresentado no segmento de referência.

Figura 45 – Rejeição ao distúrbio dos métodos simulados



Fonte: do Autor

Figura 46 – Sinal de controle PID na rejeição ao distúrbio simulada

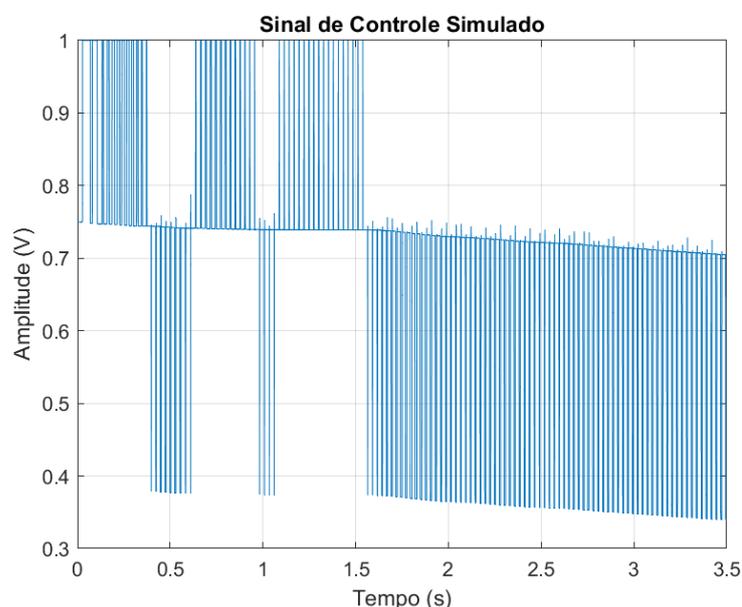


Fonte: do Autor

Como se pode ver pela Figura 47, o sinal de controle obtido para o método de controle PLL na rejeição ao distúrbio teve o seu comportamento conforme o esperado, apresentando o mesmo comportamento obtido no ensaio ao degrau de velocidade até o instante em que a carga foi aplicada. A partir deste instante observou-se uma diminuição do valor médio do sinal de saída do filtro até que a potência entregue ao motor compensasse o aumento de torque do rotor de modo a deixar a velocidade com o seu valor igual à de

referência. Também, pode-se notar a tendência do PFD em atuar no estado -1 devido a maior quantidade de rotações do motor quando aplicada a carga.

Figura 47 – Sinal de controle PLL na rejeição ao distúrbio simulada



Fonte: do Autor

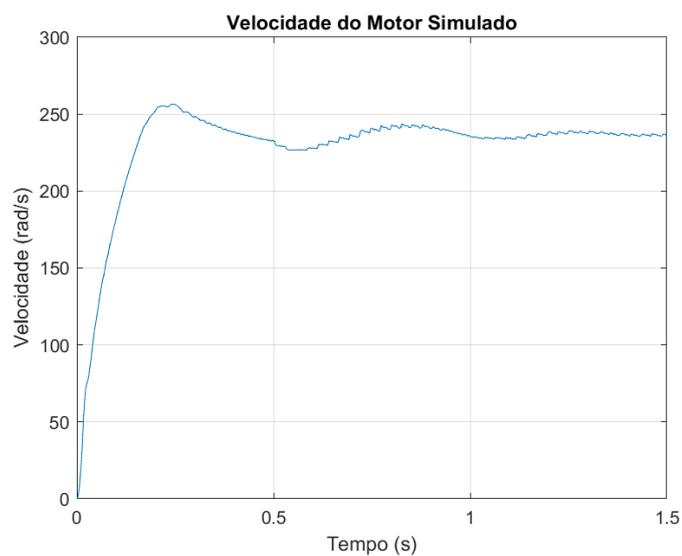
4.1.3 Resultado das Simulações com dual-PLL

Ao simular o ensaio ao degrau, com uma referência de velocidade igual a $237 \frac{rad}{s}$, aplicando-se ambos os métodos de controle, PID e PLL, e utilizando o controlador PI em conjunto do filtro PI projetados, obteve-se um sinal de velocidade com valor médio em regime permanente de $236,99 \frac{rad}{s}$, com desvio padrão de $0,64 \frac{rad}{s}$. Através deste valor, obteve-se um erro menor que 0,01%. Ainda, verificou-se um tempo de acomodação de 0,46s, o qual foi igual ao tempo obtido com controle PI, e um sobre-passo de 8,02%. A curva do sinal de velocidade, de onde se obteve os valores citados, está mostrada na Figura 48.

No ensaio ao degrau, observou-se, pela curva do sinal de controle mostrada na Figura 49, um comportamento híbrido, correspondente a união dos comportamentos dos sinais de controle verificados nos ensaios anteriores, com o sinal de controle, durante o período de atuação PID, atuando de forma idêntica ao observado no ensaio com PID puro, conforme o esperado, devido a escolha do chaveamento temporal, e com o comportamento do sinal de controle apresentando menos variações se comparado ao obtido para o ensaio ao degrau com PLL puro, em regime permanente.

Ainda pela Figura 49, verifica-se que o sinal de controle gerado pela método PLL apresentou os mesmos pulsos periódicos característicos de um sistema PLL atuando em

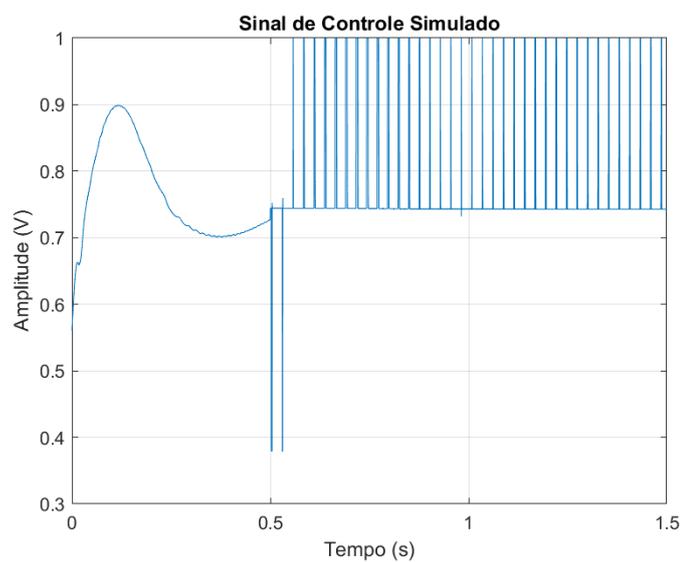
Figura 48 – Ensaio ao degra do dual-PLL simulado



Fonte: do Autor

regime permanente, porém com larguras menores, deixando o sinal de saída mais preciso quando comparado com o método de controle PLL puro.

Figura 49 – Sinal de controle do dual-PLL simulado



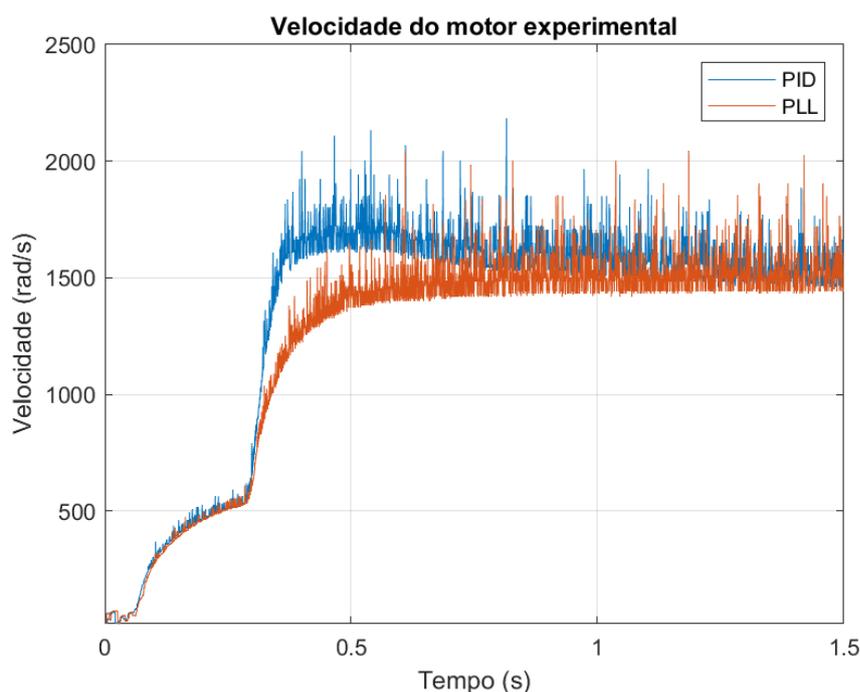
Fonte: do Autor

4.2 Resultado dos Experimentos

4.2.1 Resultado do Ensaio ao Degrau Experimental

Ao realizar o experimento do controle de velocidade do motor BLDC com o método PID, utilizando o ensaio ao degrau para se extrair as análises desejadas, com uma referência de velocidade igual a $1460 \frac{rad}{s}$, e utilizando o controlador PI projetado, obteve-se uma velocidade com valor médio em regime permanente de $1481 \frac{rad}{s}$ e com desvio padrão de $77,81 \frac{rad}{s}$. Através deste valor, chegou-se a um erro de 1,48%. Ao realizar o experimento do controle de velocidade do motor BLDC com o método PLL, utilizando também o ensaio ao degrau para se extrair as análises desejadas, com uma referência de velocidade igual a $1460 \frac{rad}{s}$, e utilizando o filtro PI projetado, obteve-se uma velocidade com valor médio em regime permanente de $1517 \frac{rad}{s}$, com desvio padrão de $88,61 \frac{rad}{s}$. Através deste valor, chegou-se a um erro de 3,93%. As curvas dos sinais de velocidade, de onde se obteve os valores citados, está mostrada na Figura 50.

Figura 50 – Resposta do PID experimental ao degrau



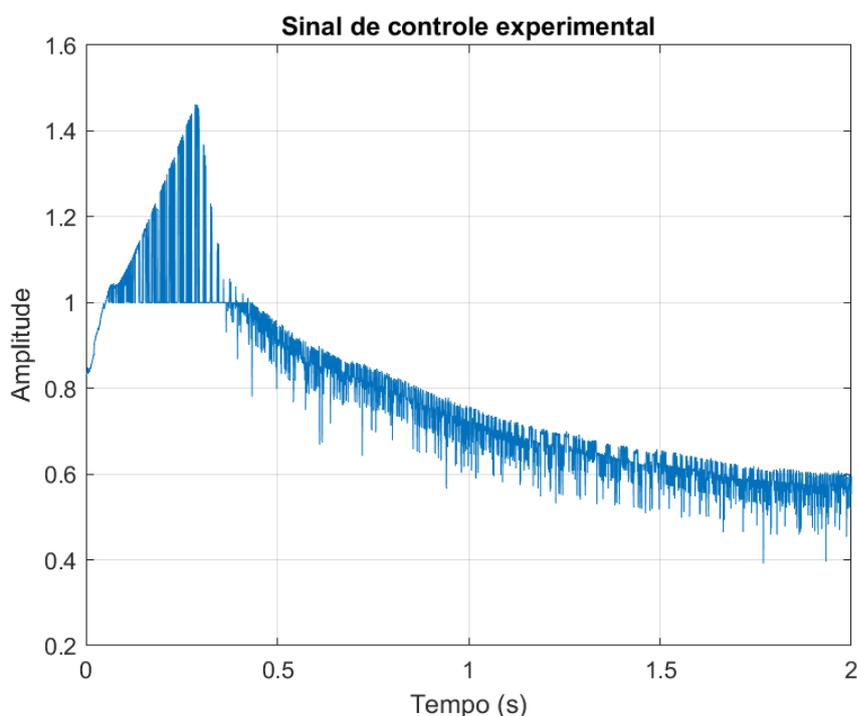
Fonte: do Autor

Após alguns experimentos, observou-se que, durante o regime transitório, o acionamento não alterou o seu padrão de controle inicial, independente do valor de ciclo de trabalho enviado, apresentando um comportamento semelhante a um *soft-start*. Este comportamento pôde ser observado na Figura 50, onde se pode ver um ponto de inflexão que delimita o fim do controle por *soft-start* e o começo do controle por PID ou PLL. O

efeito do *soft-start* também pode ser percebido pela Figura 51, onde verifica-se a saturação do sinal de controle provocada durante o período em que ele está ativo, tendo este comportamento ocorrido por causa da baixa velocidade entregue por este controle.

Ainda pela Figura 50, vê-se que, após o término do ponto de inflexão, existe uma tendência do sinal de velocidade em seguir a referência, estabelecendo-se em um valor de aproximadamente $1460 \frac{rad}{s}$, verificando-se, portanto, o comportamento padrão no regime permanente. Por estas razões, evitou-se nos experimentos a avaliação da performance temporal no regime transitório por parte dos métodos de controle propostos. Entretanto, ainda assim, em cada método de controle, verificou-se a precisão no segmento de referência, já que estas avaliações são obtidas em regime permanente.

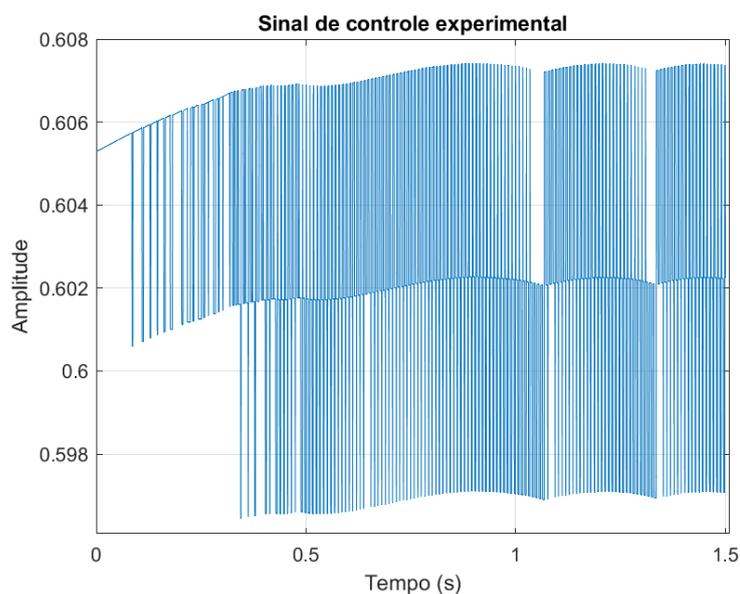
Figura 51 – Sinal de controle do PID experimental



Fonte: do Autor

O sinal de controle obtido para o método PLL experimental, teve o seu comportamento conforme o esperado pela teoria, onde o sistema foi posto a operar em uma região de pequenos sinais, mais especificamente ao redor de um ciclo de trabalho de 0,6. O seu comportamento, conforme vê-se pela Figura 52, é o de um sinal periódico com amplitude modulada por largura de pulso, com maiores amplitudes nos instantes iniciais, devido a maior necessidade de atuação. Da mesma forma que ocorrido para o experimento com controlador PID, as características temporais do método PLL não foram analisadas devido a presença do *soft-start*.

Figura 52 – Sinal de controle do PLL experimental



Fonte: do Autor

4.3 Discussões

Através dos ensaios ao degrau, verificou-se em simulação que o controle com ação PI teve êxito em diminuir o tempo de acomodação, conforme o almejado, e, em simulação e experimento, que este foi capaz de entregar uma saída mais precisa, ao custo de apresentar um sobre-passo que não existia. Também verificou-se, em simulação, que o método de controle PLL deixou o sistema com uma resposta mais lenta do que a obtida em malha aberta, e com um sobre-passo maior que o obtido no ensaio com o controle PID. Entretanto, verificou-se, em simulação e experimento, a vantagem deste método no seguimento de referência, o qual foi próximo do obtido com o PID. Porém, estes resultados estão em desacordo com a premissa da melhor precisão do seguimento de velocidade do método PLL, em relação ao método PID, como mostrado em outros trabalhos, como, por exemplo, em (CAMPOS, 1991). Ainda, não alcançou-se um resultado experimental de mesma ordem de grandeza na precisão do segmento de referência como em (PLL. . . , 2014), onde se obteve um erro experimental menor que 0,1%, o qual seria inferior ao erro de 1,48% encontrado pelo PID neste trabalho e inferior ao erro de 3,93% retornado pelo método PLL. Embora tenha-se usado, no referido trabalho, um PFD contendo 10 estados, e um *encoder* com mais ranhuras, ao invés dos 3 estados e ranhura única, utilizados neste trabalho.

À nível de teoria, pelo teorema do valor final, tem-se que ambos os métodos apresentam erro nulo no seguimento de referência, em regime permanente, mas, em nível de simulação, pequenas oscilações ao redor da referência foram observadas, e, à nível de experimento, estas foram ainda maiores. Esta discrepância se deve ao fato de que os

modelos no domínio de Laplace não são capazes de incluir todos os comportamentos do sistema, os quais apresentam não-linearidades, faixas de operação distintas, resoluções imprecisas e etc. Neste sentido, espera-se que os erros ocorridos no seguimento de referência dos sistemas propostos surjam de suas características físicas, e não teóricas.

Para se comparar ambos os métodos a nível físico, é importante primeiramente escolher configurações que tenham o melhor potencial para se seguir a referência, desta forma, sistemas de controle de velocidade com sensores diretos, como os *encoders*, devem apresentar melhor desempenho. Por esta razão utilizou-se-os neste trabalho. Portanto, ao se comparar os referidos sistemas, com os métodos de controle PID e PLL, com a utilização do *encoder* como sensor, espera-se que a diferença da capacidade de ambos em seguir a referência venha da geração do sinal de erro e da aquisição dos sinais de frequência e fase.

Tanto o bloco subtrator como o bloco PFD, utilizados pelos métodos de controle PID e PLL respectivamente, possuem a capacidade de entregar erro nulo para o controlador, desde que as saídas de fase e frequência sejam iguais a referência. Entretanto, há uma fidelidade maior na quantificação do valor do erro por parte do bloco subtrator por este gerá-lo diretamente, diferentemente do PFD que gera um sinal de erro derivado da média de outro sinal, sendo este um ponto a favor do método de controle por PID. Porém, quando se compara a aquisição dos sinais de frequência e de fase, tem-se uma vantagem para o sistema PLL, pois a aquisição do sinal de fase é obtida diretamente, enquanto que a aquisição do sinal de frequência é dependente da obtenção do tempo decorrido entre bordas de subida do sinal do encoder. Desta forma, tem-se que, aumentando o número de ranhuras do *encoder*, para um controle PLL, aumenta-se a precisão deste sinal. Já para o controle PID, ao aumentar o número de ranhuras, tende-se a perder a precisão na medida do tempo, piorando portanto o sinal de erro. Por estas razões, a capacidade de ser mais preciso no seguimento de referência está diretamente relacionada ao número de ranhuras do encoder e da capacidade de se calcular intervalos de tempo de forma precisa, consequentemente, tem-se que o método de controle por PLL possui um maior potencial para gerar um sinal de erro mais preciso.

Por este motivo, em trabalhos futuros, seria interessante analisar a precisão de ambos os métodos utilizando-se mais ranhuras, diferentemente do feito neste trabalho, onde utilizou-se somente uma ranhura em todas as análises, desfavorecendo o método PLL.

Com relação a capacidade dinâmica destes sistemas, procurou-se avaliar a estabilidade, e a velocidade da resposta do sinal de saída. Estas avaliações foram feitas em simulação medindo-se o tempo de acomodação e sobre-passo em ensaios ao degrau, e também observando-se a capacidade de rejeição ao distúrbio. Nestes quesitos, observou-se uma superioridade do método PID, o qual apresentou menos oscilações e sobre-passo, e um seguimento de referência mais rápido em relação ao método PLL. Ainda, devido às não-linearidades mais acentuadas do método PLL, teve-se de trabalhar em regiões de

operação mais reduzidas. Por estes motivos, conclui-se que o método de controle por PID tende a ser mais robusto, sendo esta uma característica importante em aplicações que exigem o seguimento de referências variadas, como, por exemplo, no controle de veículos motorizados.

Para avaliar uma solução que combinasse a melhor característica de ambos os métodos, analisou-se o dual-PLL. Pelas simulações, pôde-se observar indícios de que a união dos métodos tende a se comportar bem, apresentando erro no seguimento de referência ligeiramente inferior ao do método PID, sendo este um desempenho melhor do que o obtido por um PLL puro, por este entrar em atuação com o sistema já mais bem comportado, exigindo-se menor robustez de sua capacidade de atuação.

Por fim, em trabalhos futuros, buscando-se alcançar um método de controle com ambas as características, seria interessante testar novos métodos de controle por dual-PLL em situações nas quais o controlar PLL pudesse operar ao se atingir o regime permanente, e que o controlador PID operasse nas situações de partida do motor, na rejeição de distúrbios e na troca de referência. Portanto, seria interessante a realização de um chaveamento baseado em intervalos de velocidade, e talvez também incluir uma troca no valor do divisor de frequências, de forma a favorecer o método de controle em operação.

5 CONCLUSÕES

Neste trabalho buscou-se analisar as principais características de um sistema de controle de velocidade para um motor BLDC utilizando um método de controle pouco difundido, neste caso, o método de controle por PLL.

Como parâmetro de comparação, escolheu-se o método de controle por PID, por este ser, nestes tipos de aplicações, o mais utilizado e reconhecido na indústria. Dentre as diversas formas de se projetar estes sistemas, optou-se pela análise de sistemas com maior potencial para se obter um seguimento de referência preciso. Neste sentido, optou-se por avaliar sistemas que utilizassem *encoders*.

A avaliação do desempenho de ambos os métodos de controle, deu-se inicialmente por simulação. Os primeiros passos das simulações realizadas foram feitos para se obter a função de transferência da planta. Para simular a planta, utilizou-se o software Simulink[®] em conjunto com o Matlab[®], sendo as simulações dos blocos dos componentes da planta feitas no Simulink[®] e os *scripts* para geração e coleta dos sinais de entrada e saída feitas no Matlab[®]. Ainda, utilizou-se o Matlab[®] para processar os sinais gerados e se obter as funções de transferências para os modelos de controle, tanto para um sinal de saída do tipo fase como frequência.

De posse destes modelos, utilizou-se ferramentas de cálculo do Matlab[®], como o *rltool*, para se projetar o filtro do sistema PLL e o controlador do sistema PID. Optou-se por realizar estes projetos utilizando-se o método do Lugar Geométrico das Raízes, por este ser o que melhor se encaixa no trabalho proposto.

Com os controladores projetados, fechou-se a malha de ambos os métodos, e simulou-se novamente os sistemas no Simulink[®], gerando-se e coletando-se os sinais de entrada e saída com o Matlab[®]. Ainda, realizou-se ensaios para se verificar a capacidade dos sistemas em rejeitar o distúrbio e os resultados gerados por uma união de ambos os métodos.

Com os sinais de saídas obtidos, comparou-se o desempenho dos métodos utilizando-se métricas como tempo de acomodação, sobre-passo e tempo de rejeição ao distúrbio. Através destes resultados, observou-se um melhor desempenho do método de controle PID para contornar variações de operação, apresentando menores sobre-passo, tempo de acomodação e rejeição ao distúrbio. Quando avaliou-se a capacidade de exatidão no seguimento de referência, observou-se uma capacidade um pouco inferior para o método de controle PLL, em desacordo com observado em outros trabalhos. Entretanto, as simulações foram realizadas com *encoders* com uma única ranhura, desfavorecendo este método.

Em seguida, também avaliou-se o desempenho através de experimentos. Os primeiros experimentos foram realizados com o intuito de se obter o modelo da planta. Para a composição da estrutura da planta, escolheu-se um motor BLDC do tipo A2212/6T 2200KV com o seu eixo acoplado a um motor CC do tipo Buehler Optically encoded servo motor, e um acionador do tipo ESC UBEC 30A. Os sinais de entrada foram gerados por um microcontrolador STM32F407VET6, que também ficou responsável por medir os sinais de velocidade angular do rotor através da leitura de um sinal de *encoder* Q9898. De posse destes dados, utilizou-se um *script* do Matlab[®] para calcular o modelo da planta, tanto para uma saída de frequência como para fase.

Da mesma forma que feito para a simulação, utilizou-se ferramentas de cálculo do Matlab[®], como o *rltool*, para se projetar o filtro do sistema PLL e o controlador do sistema PID.

Com os controladores projetados, fechou-se a malha para ambos os métodos, implementando-os em *firmware*, para que em seguida se realizasse os experimentos. Após a realização dos experimentos, coletou-se os sinais de entrada e saída, os quais foram analisados através do Matlab[®].

Através da análise gráfica dos resultados, comparou-se o desempenho dos métodos. Como o acionamento não pôde operar sem um *soft-start*, não se avaliou o desempenho dinâmicos dos controladores, no regime transitório, entretanto, ainda pôde-se avaliá-los com respeito a precisão no seguimento de referência, em regime permanente. Quando avaliou-se a capacidade de exatidão no seguimento de referência, assim como ocorrido em simulação, observou-se uma capacidade inferior para o método de controle PLL, até mais acentuada em experimento, em desacordo com observado em outros trabalhos.

Em trabalhos futuros, ainda há espaço para novas investigações, principalmente investigações que utilizem mais ranhuras nos *encoders*, buscando extrair o máximo potencial de um controle por PLL. Há espaço também para testes com cargas experimentais, que não puderam ser executadas neste trabalho devido a baixa capacidade de geração de potência dos instrumentos utilizados, sendo então interessante o teste em sistemas com motores e geradores de maior porte. Por fim, existe também a possibilidade de se realizar mais investigações de sistemas operando com os métodos híbridos, que, conforme indícios das simulações, podem ser capazes de entregar um controle com as melhores características de ambos, resultando em um método de controle superior aos encontrados atualmente, ao custo de uma maior complexidade de projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE, L. A. *INTRODUÇÃO À IDENTIFICAÇÃO DE SISTEMAS: TÉCNICAS LINEARES E NÃO-LINEARES APLICADAS A SISTEMAS REAIS*. [S.l.]: Editora UFMG, 2007. ISBN 978-85-7041-584-4. Citado 2 vezes nas páginas 26 e 27.

BEST, R. E. *PHASE-LOCKED LOOPS: DESIGN, SIMULATION AND APPLICATIONS*. [S.l.]: McGraw-Hill Companies, Inc., 2003. ISBN 0-07-141201-8. Citado 19 vezes nas páginas 14, 30, 31, 33, 34, 35, 36, 37, 38, 41, 44, 45, 46, 47, 48, 49, 50, 61 e 62.

CAMPOS, P. R. B. de. *CONTROLE DE VELOCIDADE DE UM MOTOR DC ATRAVÉS DE UM CONTROLADOR PLL-DUAL*. Dissertação (Mestrado) — UNIVERSIDADE ESTADUAL DE CAMPINAS, 1991. Citado 5 vezes nas páginas 14, 15, 36, 88 e 103.

CHANG-LIANG, X. *PERMANENT MAGNET BRUSHLESS DC MOTOR DRIVES AND CONTROLS*. [S.l.]: John Wiley Sons Singapore Pte. Ltd., 2012. ISBN 978-1-118-18833-0. Citado 8 vezes nas páginas 16, 18, 19, 21, 22, 24, 25 e 28.

FRANKLIN F. GENE; POWELL, J. D. E.-N. A. *Sistemas de Controle para Engenharia*. [S.l.]: Bookman Companhia Editora Ltda, 2013. ISBN 9780136019695. Citado na página 102.

HOBBYKING, I. *30A UBEC USER MANUAL*. 2022. Disponível em: <<https://hobbyking.com/media/file/712477887X38755X54.pdf>>. Citado na página 69.

HOBBYKING, I. *HK-7X USER MANUAL*. 2022. Disponível em: <<https://www.manualslib.com/manual/823116/Hobbyking-Hk-7x.html?page=2#manual>>. Citado na página 70.

KRAUSE PAUL C.; WASYNCZUK, O.; SUDHOFF, S. D. *ANALYSIS OF ELECTRIC MACHINERY AND DRIVE SYSTEMS*. [S.l.]: John Wiley Sons Singapore Pte. Ltd.; IEEE Press Power Engineering Society, Sponsor, 2002. ISBN 978-85-4301-375-6. Citado 2 vezes nas páginas 18 e 20.

MATHWORKS, I. T. *Mixed-Signal Blockset*. 2022. Disponível em: <<https://www.mathworks.com/help/msblks/>>. Citado na página 51.

MATHWORKS, I. T. *Simscape Block Libraries*. 2022. Disponível em: <<https://www.mathworks.com/help/physmod/simscape/ug/introducing-the-simscape-block-libraries.html>>. Citado na página 51.

MATHWORKS, I. T. *System Identification Toolbox*. 2022. Disponível em: <<https://www.mathworks.com/help/ident/>>. Citado na página 54.

MODELING of BLDC Motor with Ideal BackEMF for Automotive Applications. Citado na página 13.

OGATA, K. *ENGENHARIA DE CONTROLE MODERNO*. [S.l.]: Pearson Education, Inc., 2011. ISBN 978-85-4301-375-6. Citado na página 26.

PLL Control System Based on FPGA for Brushless DC Motor. Citado 4 vezes nas páginas 14, 15, 36 e 88.

SATHYAN ANAND; MILIVOJEVIC, N. L. Y.-J. K. M.; EMADI, A. An fpga-based novel digital pwm control scheme for bldc motor drives. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 56, NO. 8, 2009. Citado 2 vezes nas páginas 13 e 14.

SCHREIBER, A.; ISERMANN, R. Identification methods for experimental nonlinear modeling of combustion engines. *2006 IFAC*, 2006. Citado na página 26.

SILVA, D. F. *ANÁLISE DE UM MOTOR BLDC PELO MÉTODO DOS ELEMENTOS FINITOS*. 2018. Citado na página 13.

TIBOR BALOGH; FEDAK, V.; ĎUROVSKY, F. Modeling and simulation of the bldc motor in matlab gui. *978-1-4244-9312-8/11/26.00 ©2011 IEEE*, 2011. Citado na página 13.

Apêndices

APÊNDICE A – SCRIPT SPLL COMO LPLL

```

f=100e3;
N = 25e6; % numero de amostras
w = 2*pi*f; % frequencia angular de referencia
T = 1/f;
%%=====
%% %%% PROJETO DO LPLL %%%
%%=====
damp = 1/sqrt(2); % coeficiente de amortecimento
TL = 0.33; % tempo de acomodacao
wn = 2*pi/TL; % frequencia natural
Nd = 1; % divisor de frequencia
% wn << K0*Kd
Ub = 5; % alimentacao charged pumped
Kd = Ub/(2*pi); % ganho PD
K0 = 2*pi*1000/Kd; % ganho VCO
tau1 = K0*Kd/(Nd*wn^2); % constante filtro 1
tau2 = 2*damp/wn^2; % constante filtro 2
delta_f0 = 3;
f0 = f - delta_f0;
w0 = 2*pi*f0;
TP = pi^2*(2*pi*delta_f0)^2/(16*damp*wn^3)
%%=====
%% Condicoes iniciais
%%=====
n = 2;
u2(n) = 0;
phi2(n) = 0;
ud(n-1) = 0;
uf(n-1) = 0;
Ti = T/20; % tempo de interrupcao
a1 = -1;
b0 = Ti/(2*tau1)*[1+1/tan(Ti/(2*tau2))];
b1 = Ti/(2*tau1)*[1-1/tan(Ti/(2*tau2))];

```

```
%%=====
%% CODIGO LPLL
%%=====

for n = 2:N

    u1(n) = sin(w*(n*Ti));
    ud(n) = Kd*u1(n)*u2(n);
    uf(n) = -a1*uf(n-1) + b0*ud(n) + b1*ud(n-1);
    phi2(n+1) = phi2(n) + [w0+K0*uf(n)]*Ti;

    if phi2(n+1)>pi
        phi2(n+1) = phi2(n+1) - 2*pi;
    end
    if phi2(n+1)>=0
        u2(n+1) = 1;
    else
        u2(n+1) = -1;
    end

    ud(n-1) = ud(n);
    uf(n-1) = uf(n);
    phi2(n) = phi2(n+1);
    u2(n) = u2(n+1);

end
```

APÊNDICE B – SCRIPT SPLL COMO DPLL

```

N = 2e4; % numero de amostras
f=100e3;
w = 2*pi*f; % frequencia angular do motor
Tw = 1/f;
delta_f0 = 3;
f0 = f - delta_f0;
w0 = 2*pi*f0;
Ti = Tw/2; % tempo de interrupcao
%%=====
%% Parametros do Filtro
%%=====
damp = 1/sqrt(2);
TL = 1/100;
wn = 2*pi/TL;
Nd = 1;
% wn << K0*Kd
Ub = 2; %Ub saida do charged pump varia de -1 a 1V
Kd = Ub/(4*pi);
K0 = 2*pi*1000/Kd;
tau2 = 2*damp/wn-1/(K0*Kd);
tau1 = K0*Kd/(Nd*wn)-tau2;
%%=====
%% Condicoes Iniciais
%%=====
n = 2;
t(n-1) = 0;
tcross(n-1) = 0;
t_u1(n-1) = 0;
t_u2(n-1) = 0;
uc(n-1) = 0;
phi2(n-1) = 0;
Q(n-1) = 0;
u1(n-1) = 0;
uf(n-1) = 0;

```

```
%%=====
%%Codigo
%%=====
fs = w/(2*pi);
Ti = 1/fs;

for n=2:N

    T(n) = Ti/2;
    u1(n) = 0.5 + 0.5*(-1)^n;
    phi2(n) = phi2(n-1) + (w0+K0*uf(n-1))*T(n);

    if(phi2(n) > 2*pi)
        phi2(n) = phi2(n) - 2*pi;
        phi2(n-1) = phi2(n-1) - 2*pi;
    end

    if((phi2(n)>=0) && (phi2(n-1)<0))
        tcross(n) = (0-phi2(n-1))*T(n)/(phi2(n)-phi2(n-1));
    else
        tcross(n) = 0;
    end

    %PFD

    switch Q(n-1)

        case 1

            if(tcross(n)==0)
                t_h(n) = T(n);
                t_l(n) = 0;
                Q(n) = 1;
            else
                t_h(n) = tcross(n);
                t_l(n) = 0;
                Q(n) = 0;
            end

        case 0

            if(u1(n-1)>0.5)
```

```
        if(tcross(n)==0)
            t_h(n) = T(n);
            t_l(n) = 0;
            Q(n) = 1;
        else
            t_h(n) = tcross(n);
            t_l(n) = 0;
            Q(n) = 0;
        end
    else
        if(tcross(n)==0)
            t_h(n) = 0;
            t_l(n) = 0;
            Q(n) = 0;
        else
            t_h(n) = 0;
            t_l(n) = T(n)-tcross(n);
            Q(n) = -1;
        end
    end
end

case -1

    if(u1(n-1)>0.5)
        if(tcross(n)==0)
            t_h(n) = 0;
            t_l(n) = 0;
            Q(n) = 0;
        else
            t_h(n) = 0;
            t_l(n) = T(n) - tcross(n);
            Q(n) = -1;
        end
    else
        t_h(n) = 0;
        t_l(n) = T(n);
        Q(n) = -1;
    end
end

end
```

```
% FILTRO
```

```
if(t_h(n)>0)
    uc(n) = uc(n-1) + t_h(n)*(Ub-uc(n-1))/(tau1+tau2);
    uf(n) = uc(n) + (t_h(n)/T(n))*tau2*(Ub-uc(n))/(tau1+tau2);
else
    if(t_l(n)>0)
        uc(n) = uc(n-1)*(1-t_l(n)/(tau1+tau2));
        uf(n) = uc(n)*(1-tau2*t_l(n)/(T(n)*(tau1+tau2)));
    else
        uc(n) = uc(n-1);
        uf(n) = uc(n);
    end
end

end

%ATUALIZACAO

Q(n-1) = Q(n);
phi2(n-1) = phi2(n);
uc(n-1) = uc(n);
uf(n-1) = uf(n);

end
```

APÊNDICE C – CRITÉRIO DE POPOV

APLICADO A UM PLL

Considerando-se que uma região de operação de um sistema possa ser linear de tal forma que possa ser descrita como nas equações (C.1) e (C.2)

$$\bar{x}' = \bar{A}\bar{x} + \bar{b}u \quad (\text{C.1})$$

$$y = \bar{C}^T \bar{x} \quad (\text{C.2})$$

e que este tenha uma região não-linear que possa ser isolada e descrita em termos de uma função não-linear $\phi(t)$, que recebe um sinal de erro $e(t)$ e gera um sinal $u(t)$, como em (C.3),

$$u = \phi(e(t)) \quad (\text{C.3})$$

então, segundo (FRANKLIN F. GENE; POWELL, 2013), tem-se que este pode ser tratado como um sistema do tipo Lur'. Ainda conforme (FRANKLIN F. GENE; POWELL, 2013), é possível para estes sistemas encontrar critérios gráficos de estabilidade, sendo o mais conhecido destes o critério do círculo. Para que se possa aplicar este critério, primeiramente, assume-se que a entrada de referência do sistema é nula. Quando isto acontece, as equações de estado do sistema podem ser descritas como em (C.4).

$$\bar{x} = \bar{A}\bar{x} - \bar{b}\phi(y) \quad (\text{C.4})$$

Com o sistema operando neste estado, define-se a condição de setor $k_1 \geq \frac{\phi(e)}{e} \geq k_2$, que basicamente informa que o gráfico de $\phi(e)$ deve estar no primeiro quadrante, mais precisamente, entre duas retas de inclinação k_1 e k_2 , podendo estas serem retas com inclinação de zero até noventa graus. Desta forma se garante que o sistema tem uma estabilidade assintótica global. Portanto uma investigação sobre quais condições levam um sistema Lur' a respeitar esta regra deve ser feita.

Para verificar se este sistema é estável na origem, quando aplicada uma referência nula, faz-se a análise de Lyapunov. A análise de Lyapunov consiste em encontrar uma função que represente a energia do sistema, para que desta forma possa se verificar se o sistema tende a perder a energia, como um sistema estável, ou se tende a acumulá-la de

forma indefinida, como um sistema instável. Para que esta função se demonstre estável, é necessário que ela cumpra quatro requisitos: retornar um valor nulo quando sua entrada é nula; retornar um valor positivo para qualquer entrada não nula; ser uma função contínua com derivadas contínuas e apresentar uma variação negativa com relação a variável de entrada. Como feito em (CAMPOS, 1991), e também em diversas outras aplicações, considera-se como uma função Lyapunov para um sistema PLL com motor no laço de realimentação uma função de Lyapunov composta por dois termos, um que descreve a energia potencial e outro que descreve a energia cinética no estado atual. Uma forma comum para esta função é mostrada na relação dada em (C.5),

$$V(x) = \frac{1}{2} \bar{x}^T \bar{P} \bar{x} + \beta \int_0^y \phi(\xi) d\xi \quad (C.5)$$

onde \bar{P} é considerada uma matriz positiva contendo a solução para um sistema com energia quadrática, conforme a equação de Lyapunov $\bar{A}^T \bar{P} + \bar{P} \bar{A} + \bar{Q} = 0$. Onde $\bar{Q} > 0$ e $\bar{\beta} > 0$. Após definida a equação Lyapunov, verifica-se se a sua quarta condição é verdadeira. Para tal, aplica-se a derivada na equação (C.5), resultando na expressão (C.6).

$$V(x)' = \frac{1}{2} \bar{x}'^T \bar{P} \bar{x} + \frac{1}{2} \bar{x}^T \bar{P} \bar{x}' + \beta y' \phi(y) \quad (C.6)$$

Substituindo-se (C.4) em (C.6) e calculando a derivada da saída em (C.2), chega-se em (C.7).

$$V(x)' = \frac{1}{2} \bar{x}^T (\bar{A}^T \bar{P} + \bar{P} \bar{A}) \bar{x} - \frac{1}{2} \phi(y) \bar{b}^T \bar{P} \bar{x} - \frac{1}{2} \bar{x}^T \bar{P} \bar{b} \phi(y) + \beta \bar{C}^T \phi(y) \bar{A} \bar{x} - \beta \bar{C}^T \phi(y)^2 \bar{b} \quad (C.7)$$

Aplicando-se a equação de Lyapunov, chega-se na equação (C.8), a qual pode ser reescrita na forma matricial como em (C.9).

$$V(x)' = \frac{1}{2} \bar{x}^T (\bar{Q}) \bar{x} - \frac{1}{2} \phi(y) \bar{b}^T \bar{P} \bar{x} - \frac{1}{2} \bar{x}^T \bar{P} \bar{b} \phi(y) + \beta \bar{C}^T \phi(y) \bar{A} \bar{x} - \beta \bar{C}^T \phi(y)^2 \bar{b} \quad (C.8)$$

$$V(x)' = - \begin{bmatrix} \bar{x}^T & \phi(y) \end{bmatrix} \begin{bmatrix} \frac{1}{2} \bar{Q} & \frac{1}{2} (\bar{P} \bar{b} - \beta \bar{A}^T \bar{C}) \\ \frac{1}{2} (\bar{P} \bar{b} - \beta \bar{A}^T \bar{C})^T & \beta \bar{C}^T \bar{b} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \phi(y) \end{bmatrix} \quad (C.9)$$

Se a derivada da função de Lyapunov resultar em valores negativos, o sistema será estável, mas se o valor for positivo não haverá informação o suficiente para se analisar

a estabilidade. Para contornar este problema supõe-se que existem duas matrizes que respeitam as condições $\bar{M} > 0$ e $\bar{P} > 0$, e que há os vetores \bar{q} e \bar{h} e o escalar ψ que satisfazem as equações (C.10), (C.11), (C.12) e (C.13).

$$\bar{A}^T \bar{P} + \bar{P} \bar{A} = -\bar{Q} = -\bar{q} \bar{q}^T - \bar{M} \quad (\text{C.10})$$

$$\bar{P} \bar{b} - \bar{h} = \sqrt{\psi} \bar{q} \quad (\text{C.11})$$

$$\frac{1}{2} \psi = \beta \bar{C}^T \bar{b} + \frac{1}{K} \quad (\text{C.12})$$

$$\bar{h} = \beta \bar{A}^T \bar{C} + \bar{C} \quad (\text{C.13})$$

Substituindo-se estas equações em (C.9), chega-se na equação matricial mostrada em (C.14).

$$V(x)' = - \begin{bmatrix} \bar{x}^T & \phi(y) \end{bmatrix} \begin{bmatrix} \frac{1}{2}(\bar{M} + \bar{q} \bar{q}^T) & \frac{1}{2}(\sqrt{\psi} \bar{q} + \bar{C}) \\ \frac{1}{2}(\sqrt{\psi} \bar{q} + \bar{C})^T & \frac{1}{2} \psi - \frac{1}{K} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \phi(y) \end{bmatrix}$$

(C.14)

Calculando-se a equação (C.14) efetuando-se os produtos e realizando as devidas substituições, chega-se na equação (C.15) para a derivada de Lyapunov, de onde se vê que se cumpre o quarto critério.

$$V(x)' = -\frac{1}{2} \bar{x}^T \bar{M} \bar{x} - \frac{1}{2} (\bar{x}^T \bar{q} + \sqrt{\psi} \phi(y))^2 - y \phi(y) \left(1 - \frac{\phi(y)}{K y}\right) \quad (\text{C.15})$$

Para que a equação (C.15) seja definida como não nula para todas as entradas, com exceção da origem, deve-se garantir que o sistema seja completamente observável. Esta condição pode ser confirmada utilizando-se do lema de Kalman-Yakubovich.

O lema de Kalman-Yakubovich declara que um sistema com uma matriz \bar{A} e um escalar β controláveis, com vetores \bar{h} , \bar{q} e um escalar ϕ reais, e com matrizes \bar{P} e \bar{M} positivas é capaz de satisfazer as equações (C.16) e (C.17)

$$\bar{A}^T \bar{P} + \bar{P} \bar{A} = -\bar{q} \bar{q}^T - \bar{M} \quad (\text{C.16})$$

$$\overline{Pb} - \overline{h} = \sqrt{\psi\overline{q}} \quad (\text{C.17})$$

somente se a função de transferência dada em C.18 for real e positiva.

$$H(s) = \frac{1}{2}\psi + \overline{h}^T (s\overline{I} - \overline{A})^{-1}\overline{b} \quad (\text{C.18})$$

Substituindo-se as equações C.11, C.12 e C.13 em C.18, chega-se em C.19

$$H(s) = \beta\overline{C}^T (\overline{I} + \overline{A}(s\overline{I} - \overline{A})^{-1})\overline{b} + \overline{C}^T (s\overline{I} - \overline{A})^{-1}\overline{b} \quad (\text{C.19})$$

Por fim, considerando-se a função de transferência da planta $G(s) = \overline{C}^T (s\overline{I} - \overline{A})^{-1}\overline{b}$, chega-se na expressão (C.20) para $H(s)$.

$$H(s) = (1 + \beta s)G(s) + \frac{1}{K} \quad (\text{C.20})$$

Para se garantir que $H(s)$ seja real positiva, a inequação (C.21) deverá ser respeitada. Sendo esta condição chamada de Critério de Popov.

$$Re \{H(j\omega)\} = Re \left\{ (1 + \beta j\omega)G(j\omega) + \frac{1}{K} \right\} \geq 0 \quad (\text{C.21})$$

APÊNDICE D – CÓDIGO CONTROLADOR PID

```

%%*****
%% UFRGS
%%-----
%% ARQUIVO: main.c
%% AUTOR: Samuel
%% VERSAO: 1.00
%% DATA: 06/08/2022
%%*****
#include "stm32f4xx.h"
#include "stm32f4xx_hal.h"
#include "stdlib.h"
#include "main.h"
#include "inicializacao.h"
#include "funcoes.h"
%%=====
%% VARIAVEIS EXTERNAS
%%=====
volatile uint32_t msTicks;
%%=====
%% VARIAVEIS INTERNAS
%%=====
uint32_t cont = 0;
float w_motor = 0;
float ciclo_trabalho = 0.0;
%%=====
%% MAIN
%%=====
int main()
{
    float erro;
    float referencia = 1460;
    float tempo_amostragem = 0.0005;
    float P;
    float I;

```

```

float Kp = 0.00047;
float Ki = 0.0028;
Config_SystemClock();
Config_Carga();
Config_TIM4_PWM();
Config_TIM2_us();
Config_TIM3_Encoder();
ciclo_trabalho = 0.0;
aciona_motor();

while(1)
{
    erro = referencia - w_motor;
    P = Kp*erro;
    I = I + Ki*erro*tempo_amostragem;
    ciclo_trabalho = P + I;
    if(ciclo_trabalho >= 1.0)
        ciclo_trabalho = 1.0;
    if(ciclo_trabalho <= 0.1)
        ciclo_trabalho = 0.1;
    PWM(ciclo_trabalho); // faixa de 0.1 a 1
    Delay_us(500);
}
}

%%=====
%% INTERRUPTOES
%%=====
%%-----
%% INTERRUPTAO:
%% Descr: Interrupcao gerada a cada 32 bordas de subida do sinal de encoder
%%-----
void TIM3_IRQHandler(void)
{
    w_motor = 3.14/(16*periodo_motor()); %%Calculo da velocidade a cada 32
    ranhuras
    TIM3->SR &= ~(TIM_SR_UIF|TIM_SR_CC1IF); %%Limpa o flag de interrupcao e
    de comparacao do channel1
}

```

APÊNDICE E – CÓDIGO CONTROLADOR PLL

```

%%*****
%% UFRGS
%%-----
%% ARQUIVO: main.c
%% AUTOR: Samuel
%% VERSAO: 1.00
%% DATA: 06/08/2022
%%*****
#include "stm32f4xx.h"
#include "stm32f4xx_hal.h"
#include "stdlib.h"
#include "main.h"
#include "inicializacao.h"
#include "funcoes.h"
%%=====
%% VARIAVEIS EXTERNAS
%%=====
volatile uint32_t msTicks;
%%=====
%% VARIAVEIS INTERNAS
%%=====
#define u1 1
#define u2 2
uint32_t cont = 0;
uint32_t cont_encoder = 0;
float w_motor = 0;
float ciclo_trabalho = 0.0;
float Ub = 0.075;
float Ud = 0;
int ESTADO_PFD = 0;
%%=====
%% FUNCOES INTERNAS
%%=====
void PFD(int u);

```

```
%%=====
%% MAIN
%%=====
int main()
{
    float tempo_amostragem = 0.0005;
    float P;
    float I;
    float Kp = 0.0687;
    float Ki = 0.0729;

    Config_SystemClock();
    Config_Carga();
    Config_TIM4_PWM();
    Config_TIM2_us();
    Config_TIM3_Encoder();
    ciclo_trabalho = 0.0;
    aciona_motor();

    while(1)
    {
        P = Kp*Ud;
        I = I + Ki*Ud*tempo_amostragem;
        ciclo_trabalho = P + I + 0.6;

        if(ciclo_trabalho >= 1.0)
            ciclo_trabalho = 1.0;
        if(ciclo_trabalho <= 0.1)
            ciclo_trabalho = 0.1;

        PWM(ciclo_trabalho);%% faixa de 0.1 a 1
        Delay_us(500);
    }
}
```

```
%%=====
%% INTERRUPTOS
%%=====
%%-----
%% INTERRUPTO:
%% Descr: Interrupcao gerada a cada 32 bordas de subida do sinal de encoder
%%-----
void TIM3_IRQHandler(void)
{
    w_motor = 3.14/(16*periodo_motor()); %%Calculo da velocidade a cada 32
        ranhuras
    cont_encoder++;
    if(cont_encoder >= 32)
    {
        cont_encoder = 0;
        PFD(u2);
    }
    TIM3->SR &= ~(TIM_SR_UIF|TIM_SR_CC1IF);%%Limpa o flag de interrupcao e
        de comparacao do chanel1
}
%%-----
%% INTERRUPTO:
%% Descr: Interrupcao chamada com a frequencia de referencia
%%-----
void TIM5_IRQHandler(void)
{
    PFD(u1);
    TIM5->SR &= ~(TIM_SR_UIF);%%Limpa o flag de interrupcao
}
}
```

```
%%-----  
%% FUNCAO:  
%% Descr: Diagrama de estados de um PFD  
%%-----  
void PFD(int u)  
{  
    switch(ESTADO_PFD)  
    {  
        case 1:  
            if(u == u1){  
                ESTADO_PFD = 1;  
                Ud = Ub;  
            }  
            if(u == u2){  
                ESTADO_PFD = 0;  
                Ud = 0;  
            } break;  
        case 0:  
            if(u == u1){  
                ESTADO_PFD = 1;  
                Ud = Ub;  
            }  
            if(u == u2)  
            {  
                ESTADO_PFD = -1;  
                Ud = -Ub;  
            } break;  
        case -1:  
            if(u == u1){  
                ESTADO_PFD = 0;  
                Ud = 0;  
            }  
            if(u == u2){  
                ESTADO_PFD = -1;  
                Ud = -Ub;  
            } break;  
    }  
}
```