



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Martha Roxana Quispe Ayala

**Estudo de Diferentes Abordagens para o Acionamento do
Motor Síncrono de Ímã Permanente Sem Sensor de Posição**

Rio de Janeiro

2019

Martha Roxana Quispe Ayala

Estudo de Diferentes Abordagens para o Acionamento do Motor Síncrono de Ímã Permanente Sem Sensor de Posição



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Controle e Automação.

Orientador: Prof. Dr. Luís Fernando Corrêa Monteiro.

Orientador: Prof. Dr. Lisandro Lovisolo.

Rio de Janeiro

2019

Martha Roxana Quispe Ayala

Estudo de Diferentes Abordagens para o Acionamento do Motor Síncrono de Ímã Permanente Sem Sensor de Posição

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Controle e Automação.

Aprovado em: 18 de Dezembro de 2019

Banca Examinadora:

Prof. Dr. Luís Fernando C. Monteiro. (Orientador)

Faculdade de Engenharia - UERJ

Prof. Dr. Lisandro Lovisolo. (Orientador)

Faculdade de Engenharia - UERJ

Prof. Dr. Harold Dias de Mello Junior.

Faculdade de Engenharia - UERJ

Prof. Dr. Marcio Zamboti Fortes.

Faculdade de Engenharia - UFF

Rio de Janeiro

2019

AGRADECIMENTO

A Deus, por sempre estar comigo.

A minha família, por oferecer-me todo seu apoio, em especial aos meus pais Marta e Daniel, que sempre têm a palavra certa para me animar e a quem amo com todo meu coração, as minhas irmãs, que sempre me dão um motivo pra sorrir, e aos meus sobrinhos, que cada final de ano aguardam por mim chegar em casa.

Ao meu orientador, Prof. Dr. Luís Fernando, que sempre me deu os melhores conselhos nas horas boas e também nas difíceis, e que contribuiu imensamente ao longo desta trajetória.

Ao meu orientador, Prof. Dr. Lisandro Lovisolo, por não desistir de mim e acreditar que eu poderia alcançar os objetivos deste projeto.

Ao Prof. Cleiton Magalhães, pela pronta ajuda, sempre que precisei.

Aos professores Jorge Amaral e José Paulo Vilela, pela contribuição dos conhecimentos adquiridos nas suas matérias.

A todos os amigos que fiz e me apoiaram neste caminho, em especial, ao Andrés, Celina, Hugo, Nayara, Dayane, Veronica, Dennis, Aline, Victor Hugo, Manuchi, Paulo, Israel, Virginia, Talita, Camila, Marcia.

Obrigada a todos vocês!

Quem nunca errou nunca experimentou nada novo.

Albert Einstein

RESUMO

QUISPE A., Martha R. *Estudo de Diferentes Abordagens para o Acionamento do Motor Síncrono de Ímã Permanente Sem Sensor de Posição*. 087f. Dissertação de mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2019.

Esta dissertação considerou diferentes abordagens de técnicas de controle para acionamento de um Motor Síncrono de Ímã Permanente (MSIP) sem sensores de velocidade. As técnicas foram implementadas com base no controle por orientação de campo (*Field Oriented Control* – FOC) no qual, a título de comparação, considerou-se inicialmente o controle com o uso de um sensor de posição para determinar a velocidade do motor. Esta primeira abordagem permitiu analisar o acionamento do MSIP com base no controle FOC convencional. Em seguida, três diferentes abordagens de técnicas de controle sem sensor de posição foram implementadas. Na segunda (considerando a ordem de apresentação no texto), a velocidade mecânica foi estimada por meio de um circuito de sincronismo denominado IPLL (*Improved Phase-Locked-Loop*), no qual, os sinais de entrada correspondem aos fluxos no estator representados no sistema de coordenadas $\alpha - \beta$. Nas duas restantes, a velocidade mecânica foi estimada por meio de técnicas de inteligência artificial como Redes Neurais, a partir de duas bases de dados diferentes, que consistiu no objetivo maior deste trabalho. A primeira base de dados correspondeu aos sinais adquiridos pelo sensor mecânico utilizado na primeira abordagem. A segunda base de dados foi obtida a partir da velocidade mecânica estimada pelo IPLL. Com base nos resultados de simulação obtidos com as diferentes abordagens foi possível verificar a viabilidade no uso de redes neurais para estimar a velocidade concebida a partir de diferentes bases de dados.

Palavras-chave: Motor Síncrono de Ímã Permanente (MSIP); estimador da velocidade mecânica; redes neurais; controle por orientação de campo (FOC).

ABSTRACT

QUISPE A., Martha R. *Study of different approaches for driving a Permanent Magnet Synchronous Motor without speed sensor.* 087f. Dissertação de mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2019.

This dissertation addressed the study of different approaches for driving a Permanent Magnet Synchronous Motor (MSIP) without speed sensors. Essentially, four different approaches based on Field Oriented Control (FOC) were implemented. The first one was implemented with a position sensor to acquire the frequency and the speed of the motor. In sequence, three different approaches based of sensorless control techniques were implemented. In the second (considering the order of presentation in the text), the speed was estimated through a synchronizing circuit labelled as IPLL (Improved Phase-Locked-Loop). The IPLL input signals were the rotor flux represented in $\alpha - \beta$ coordinate system. To the remaining two approaches, artificial intelligence techniques based on Neural Networks, from two different databases, were applied to estimate. The first database were the acquired signals from the sensor used in the first approach. The second database derived from the estimated speed by the IPLL. Based on the simulation results obtained through different ways of estimating the motor speed, it was verified the feasibility of using neural networks to estimate correctly the motor speed, even through different databases.

Keywords: Permanent Magnet Synchronous Motor; Mechanical Speed Estimator; Neural Networks; Field Oriented Control (FOC).

LISTA DE FIGURAS

Figura 1	Esquemático simplificado do controle de um MSIP com uso de um sensor mecânico.	15
Figura 2	Esquemático simplificado do controle sem sensores de um MSIP com uso de um estimador neural.	16
Figura 3	Rotor de um máquina síncrona convencional (a) e de uma máquina síncrona de ímã permanente (b).	20
Figura 4	Variantes de ímãs de MSIPs.	20
Figura 5	Circuito equivalente de uma máquina síncrona convencional.	21
Figura 6	Circuito equivalente do MSIP com os terminais conectados em estrela.	22
Figura 7	Representação dos fasores no sistema de coordenadas “ $a - b - c$ ” e “ $\alpha - \beta$ ”	24
Figura 8	Representação do fluxo nos sistemas de coordenadas “ $\alpha - \beta$ ” e “ $d - q$ ”. ...	25
Figura 9	Circuito equivalente no eixo direto.	26
Figura 10	Circuito equivalente no eixo quadratura.	26
Figura 11	Esquemático simplificado do circuito com o controle por orientação de campo.	28
Figura 12	Ângulo de posicionamento do rotor θ_m	32
Figura 13	Comparação entre as velocidades ω_m e ω_m^*	32
Figura 14	Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.	32
Figura 15	Correntes do estator tanto de eixo direto (i_d) quanto de quadratura (i_q). ..	33
Figura 16	Esquemático simplificado do circuito com o controle sem sensores IFO.	35
Figura 17	Diagrama de blocos para determinação da posição e velocidade com no controle sem sensores IFO.	36
Figura 18	Diagrama de blocos do Observador de Fluxo convencional.	37
Figura 19	Resposta em frequência do filtro passa altas de segunda ordem com frequência de corte em 5 Hz.	38
Figura 20	Diagrama de blocos do IPLL utilizado para estimar a velocidade ($\omega_{m.est.IFO}$) e a posição angular ($\theta_{m.est.IFO}$).	39
Figura 21	Fluxos λ_α e λ_β no Observador de Fluxo Convencional.	39

Figura 22 Fluxos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$ depois do compensador CC.....	40
Figura 23 Posição angular do rotor θ_m obtida com os fluxos antes da compensação da componente CC (λ_α e λ_β).	40
Figura 24 Posição angular do rotor θ_m obtida com os fluxos após a compensação da componente CC ($\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$).	41
Figura 25 Posição angular do rotor $\theta_{m.est.IFO}$ obtida com o IPLL.	41
Figura 26 Comparação entre a velocidade estimada ($\omega_{m.est.IFO}$) e a de referência (ω_m^*). ..	42
Figura 27 Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.	42
Figura 28 Sensores para aquisição de dados usados neste trabalho: corrente, tensão e posição.	44
Figura 29 Subamostragem Periódica que mantém a proporção de amostras/entradas das diferentes classes.	45
Figura 30 Subamostragem uniforme da velocidade mecânica “ ω_m ”.	46
Figura 31 Forma de onda original do sinal de torque e ondas resultantes das normalizações min-max e z-score.	47
Figura 32 Arquitetura <i>Perceptron</i> de Múltiplas Camadas.....	49
Figura 33 Composição de um neurônio artificial.	49
Figura 34- Função de ativação - Tangente Hiperbólica.	50
Figura 35 Algoritmo de aprendizado <i>Feed-forward and Back-propagation</i>	52
Figura 36 Comportamento do erro em função das épocas de treinamento.	52
Figura 37 Erro da rede neural artificial em função dos pesos das sinapses. As curvas em cores exemplificam como os pesos são ajustados durante o treinamento considerando diferentes taxa de aprendizado “ η ”.	53
Figura 38 Aprendizado com termo do momento.	54
Figura 39 Distribuição dos conjuntos de aprendizado.	55
Figura 40 Técnica de parada antecipada.	56
Figura 41 Esquemático do conversor <i>Buck</i>	57
Figura 42 Forma de onda da corrente desejada “ I_L ” e quantidade de amostras coletadas.	58
Figura 43 Subamostragem periódica dos sinais adquiridos no conversor <i>buck</i>	58
Figura 44 Forma de onda do “ V_o ” original e com a normalização <i>z-score</i>	59

Figura 45 Esquemático do conversor <i>Buck</i>	60
Figura 46 Comparação entre as correntes I_L , I_{LRN} e $I_L^* = 4,5 \text{ amp}$	61
Figura 47 Comparação entre as correntes I_L , I_{LRN} e $I_L^* = 7,8 \text{ amp}$	61
Figura 48- Esquemático simplificado do circuito com o controle sem sensores - Estimador Neural.	63
Figura 49- Diagrama de blocos da implementação do controlador.	64
Figura 50- Diagrama de blocos da operação do controlador.	65
Figura 51- Arquitetura da rede neural -	67
Figura 52- Desempenho da rede neural empregada.	68
Figura 53- Ângulo de posicionamento do rotor θ_m obtido do observador convencional, a partir de $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$	68
Figura 54- Comparação entre a velocidade estimada ($\omega_{m.est.RN}$) com a velocidade de referência (ω_m^*).	69
Figura 55- Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.	69
Figura 56- Arquitetura da rede neural empregada.	71
Figura 57- Desempenho da rede neural empregada.	72
Figura 58- Ângulo de posicionamento do rotor $\theta_{m.est.IFO}$ obtido com o IPLL.	72
Figura 59- Comparação entre a velocidade estimada ($\omega_{m.est.RN.IFO}$) com a velocidade de referência (ω_m^*).	73
Figura 60- Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.	73

LISTA DE TABELAS

Tabela 1	Parâmetros do motor.	30
Tabela 2	Dados nominais do motor.	31
Tabela 3	Descrição dos parâmetros da rede neural.	59
Tabela 4	Descrição dos padrões e alvo da base de dados.	66
Tabela 5	Descrição dos parâmetros da rede neural.	67
Tabela 6	Descrição dos atributos e alvo da base de dados.	70
Tabela 7	Descrição dos parâmetros da rede neural.	71
Tabela 8	Descrição dos parâmetros da rede neural.	74

LISTA DE SIGLAS

MS	Máquina Síncrona
MSIP	Motor Síncrono de Imã Permanente
FOC	Field Oriented Control
DTC	Direct Torque Control
IFO	Improved rotor Flux Observer
IPLL	Improved Phase Locked loop
SMO	Sliding Mode Operation
SOGI	Second-Order Generalized Integrator
ANN	Artificial Neural Network
MLP	Multilayer Perceptro
RMSE	Root Mean Square Error
RBF	Radial Basic Funtion
RNN	Recurrent Neural Networks
FNN	Fuzzy Neural Network

SUMÁRIO

	INTRODUÇÃO	14
1	ACIONAMENTO DO MOTOR SÍNCRONO DE ÍMÃ PERMANENTE COM SENSOR DE VELOCIDADE	19
1.1	Conceitos básicos da máquina síncrona	19
1.2	Modelagem da Máquina Síncrona de Ímã Permanente.....	21
1.2.1	Representação do modelo matemático no Sistema de Coordenadas “ $d - q$ ”.	23
1.3	Controle do Motor Síncrono de Ímã Permanente	26
1.3.1	Controle por Orientação de Campo (FOC)	27
1.3.2	Resultados de simulação com controlador por orientação de campo convencional.....	30
2	ACIONAMENTO DO MSIP SEM SENSOR DE POSIÇÃO COM USO DE UM OBSERVADOR DE FLUXO EM CONJUNTO COM O CIRCUITO DE SINCRONISMO	34
2.1	Controle sem sensores de um MSIP baseado num Observador de Fluxo	34
2.1.1	Observador de Fluxo Convencional	36
2.1.2	Compensador da componente CC	37
2.1.3	<i>Improved Phase Locked loop</i> (IPLL)	38
2.2	Resultados de Simulação	39
3	ESTIMADOR NEURAL	43
3.1	Aquisição de sinais e Geração da base de dados	43
3.2	Pré-processamento de dados	44
3.2.1	Amostragem da base de dados	44
3.2.2	Normalização dos dados	46
3.3	Redes Neurais Artificiais	47
3.3.1	Arquitetura da Rede Neural.....	48
3.3.2	Neurônio artificial.....	48
3.3.3	Funções de ativação.....	50

3.3.4	Algoritmo de aprendizado	50
3.3.5	Taxa de aprendizado “ η ” e Termo do <i>momentum</i> “ α ”	52
3.3.6	Treinamento, Validação e Teste de uma Rede Neural Artificial	54
3.4	Estimador de Corrente via Rede Neural Artificial	56
3.4.1	Aquisição de sinais e Geração da base de dados	57
3.4.2	Pré-processamento de dados	58
3.4.3	Características da Rede Neural empregada	59
3.4.4	Resultados da Generalização do modelo.....	60
4	ACIONAMENTO DO MSIP SEM SENSOR DE POSIÇÃO COM USO DE UM ESTIMADOR NEURAL DE VELOCIDADE	62
4.1	Metodologia de simulação sem sensores	62
4.2	Acionamento do MSIP sem sensores de posição com uso de um estimador neural de velocidade baseado nos dados provenientes do controle com sensores mecânicos	65
4.2.1	Aquisição de sinais e geração da base de dados.....	65
4.2.2	Características da Rede Neural empregada	66
4.2.3	Resultados da Generalização do modelo.....	68
4.3	Acionamento do MSIP sem sensores de posição com uso de um estimador neural de velocidade baseado nos dados provenientes do controle sem sensores - IFO	70
4.3.1	Definição da base de dados.....	70
4.3.2	Características da Rede Neural empregada	71
4.3.3	Resultados da Generalização do modelo.....	72
4.4	Análise e resumo dos resultados	74
	CONCLUSÃO	78
	REFERÊNCIAS	81
	ANEXOS	86
A	COMUNICAÇÃO ENTRE PROGRAMAS	86
B	CÓDIGO FONTE DO ESTIMADOR NEURAL PROPOSTO	87

INTRODUÇÃO

A extração de recursos minerais é uma das atividades que impulsionam a economia de diversos países. De acordo com o Banco Mundial, essa atividade desempenha um papel econômico, social e político importante na vida de 3,5 bilhões de pessoas em 81 países. No Brasil, de acordo com notícia publicada em Junho de 2018 pelo Ministério de Minas e Energia, a atividade extrativista representou 4% do PIB brasileiro de 2017 com superávit de US\$23,4 bilhões e crescimento de 30% no ano, dados que reforçam a importância da atividade no País.

Motores síncronos são amplamente utilizados em diferentes segmentos, incluindo atividades extrativistas. Isso porque tais motores apresentam como principais características a capacidade de operar com velocidade constante tanto em sobrecarga quanto durante a ocorrência de variações momentâneas de tensão, levando em consideração os limites do seu conjugado, e a capacidade de atuar numa ampla faixa de velocidades, com controle através de conversores de potência, independentemente da variação de carga. Além disso, máquinas síncronas apresentam outras características importantes como, por exemplo, a capacidade de correção dinâmica do fator de potência.

Particularmente, os Motores Síncronos de Ímã Permanente (MSIPs) são cada vez mais utilizados no setor de petróleo/gás em Bombas Submersíveis Elétricas [1]- [2], Compressores Submarinos de Gás [3] [4], bem como em Ferramentas Elétricas de perfuração de fundo de poço [5] [6]. Por essas razões, o MSIP é a base para a realização dos controladores e técnicas exploradas neste trabalho.

O MSIP tem algumas vantagens como, por exemplo, a alta densidade de potência (relação potência/volume), alta eficiência e facilidade no uso de controladores convencionais, tais como, o controle orientado por campo. Outro aspecto importante consiste no fato de não haver necessidade de circuito de alimentação para o rotor, o que reduz a ocorrência de manutenção e elimina as perdas no cobre da bobina do rotor [7].

No entanto, o MSIP tem como desvantagem a necessidade de conhecer a posição do rotor, uma vez que essa é a informação fundamental, para realizar o controle de velocidade de um MSIP. Esta informação é usualmente adquirida por sensores que elevam o custo do equipamento. Isso tem motivado a pesquisa e o desenvolvimento de técnicas de estimação da posição do rotor alternativas, de forma a que os sensores de posição possam

ser desconsiderados/descartados.

Durante os últimos anos, foram propostas diferentes abordagens orientadas ao controle para acionamento de um motor síncrono, que abrangem desde o controle convencional até o uso de técnicas mais elaboradas, incluindo o controle preditivo [1]. Seu objetivo é estimar corretamente o torque e a velocidade de operação da máquina com base nos níveis de tensão e corrente para o adequado acionamento do MSIP. Para tal, é necessário o conhecimento prévio do comportamento da máquina a partir ou de modelos matemáticos ou de um banco de dados que representem corretamente o comportamento do motor em função de alguns parâmetros conhecidos, o esquemático simplificado é representado na Figura 1. Neste trabalho, considera-se as tensões e correntes produzidas pelo conversor de potência como parâmetros. De fato, no cenário considerado neste trabalho, a máquina está em local de difícil acesso onde não é possível adquirir informações da velocidade angular do rotor.

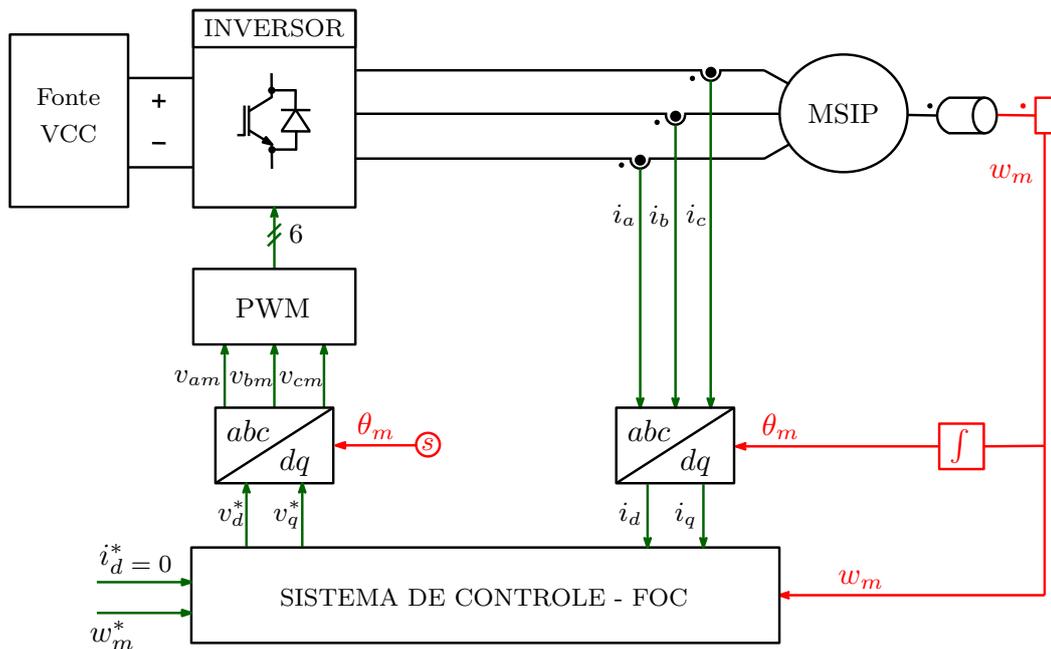


Figura 1: Esquemático simplificado do controle de um MSIP com uso de um sensor mecânico.

Na literatura, observou-se que a maioria das propostas para controladores preditivos necessita de grande capacidade de processamento para atingir desempenho satisfatório. Assim, a complexidade computacional envolvida ou requerida torna-se um dos desafios para o desenvolvimento e implantação de tais sistemas [1]. Conseqüentemente,

a proposta deste trabalho foi implementar um sistema de controle preditivo baseado em Redes Neurais, visando melhorar o desempenho/resposta de algoritmos convencionais, e concomitantemente reduzir a complexidade computacional associada à tarefa, conforme ilustrado na Figura 2.

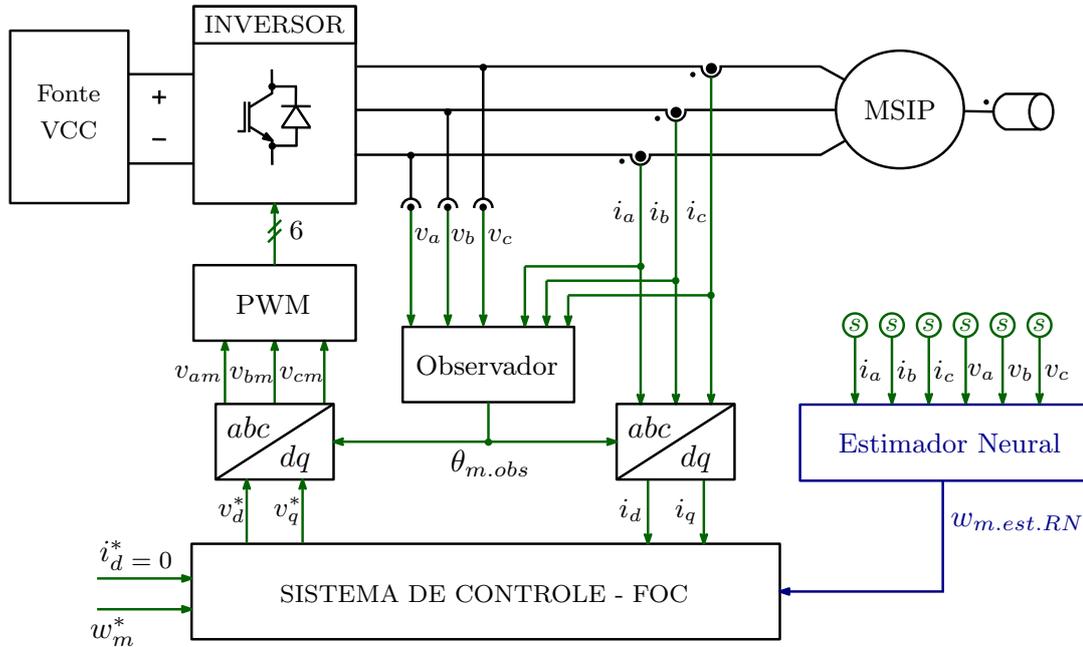


Figura 2: Esquemático simplificado do controle sem sensores de um MSIP com uso de um estimador neural.

Consequentemente, diferentes abordagens de técnicas de controle para o acionamento do MSIP, sem sensores de velocidade, foram estudados nesta dissertação. Técnicas de controle por orientação de campo (*Field Oriented Control* – FOC), no qual a título de comparação, considerou-se inicialmente o controle com o uso de um sensor de posição para determinar a velocidade do motor, foram implementadas. Esta primeira abordagem permitiu realizar análises do funcionamento do acionamento de um MSIP com base no controle FOC.

Em seguida, três diferentes abordagens de técnicas de controle sem sensor de posição foram implementadas. Na segunda (considerando a ordem de apresentação no texto), estima-se a velocidade por meio de um circuito de sincronismo denominado IPLL (*Improved Phase-Locked-Loop*), no qual, os sinais de entrada correspondem aos fluxos no estator representados no sistema de coordenadas $\alpha\beta$. Nas duas restantes, a velocidade foi estimada por meio de técnicas de inteligência artificial como Redes Neurais, a partir de

duas bases de dados diferentes, que consiste no objetivo central desta dissertação. A primeira base de dados corresponde aos sinais adquiridos pelo sensor de posição utilizado na primeira abordagem. A segunda base de dados é obtida a partir da velocidade estimada pelo IPLL.

Com base nos resultados de simulação obtidos usando as diferentes formas de estimar a velocidade foi possível verificar o melhor desempenho com o uso de redes neurais, mesmo quando a base de dados derivou de um estimador de velocidade sem sensores.

Objetivos

Fundamentalmente, este trabalho teve como objetivo central o estudo do uso de redes neurais para estimar a velocidade par o acionamento de um MSIP. A maior contribuição deste trabalho consiste de uma solução híbrida para o acionamento do MSIP. Na solução, a posição é determinada por um observador de estado proposto em [8] com a velocidade estimada por uma rede neural. Para tal, as seguintes atividades foram desenvolvidas:

- Estudo da Máquina Síncrona de Ímã Permanente(MSIP);
- Implementação e simulação do acionamento de MSIPs com base no controlador orientado por campo (FOC), com o uso de sensor da posição angular e da velocidade, e a correspondente análise dos resultados de simulação;
- Eliminação do sensor de posição para a implantação do estimador com base no observador de fluxo em conjunto com circuito de sincronismo IPLL no controlador FOC, e a correspondente análise dos resultados de simulação;
- Estudo de técnicas baseadas em redes neurais para estimação de parâmetros a partir de um conjunto de outros;
- Substituição do IPLL por uma rede neural para estimar a velocidade, isso foi feito empregando duas bases de dados diferentes, e a correspondente análise dos resultados obtidos em simulação;
- Comparação dos resultados obtidos pelas diferentes abordagens de controle de acionamento de MSIPs acima descritas.

Organização

Este trabalho está dividido em 4 capítulos. No Capítulo 1, foram descritas algumas das características de MSIP com apresentação de seus circuitos equivalentes e modelos dinâmicos correspondentes, representados em três diferentes sistemas de coordenadas. Inicialmente, a representação foi feita no sistema de coordenadas “ $a - b - c$ ”. Em seguida, ela foi convertida para o sistema de coordenadas “ $\alpha - \beta$ ” e, posteriormente fez-se uma segunda conversão para o sistema de coordenadas “ $d - q$ ”. Na sequência, foi apresentado o controlador FOC para o acionamento do PSIM. Ainda nesse capítulo foram apresentados os resultados de simulação obtidos com o controlador FOC convencional, com o uso do sensor de posição.

No Capítulo 2 foi apresentado o estimador da velocidade e a posição angular a partir de um observador de fluxo, com um circuito de sincronismo denominado IPLL. Na sequência há os resultados de simulação obtidos com a substituição do sensor de posição por este estimador, sendo o primeiro controlador sem sensor implementado e simulado neste trabalho.

No Capítulo 3, fez-se um estudo básico sobre redes neurais, descrevendo as etapas necessárias à concepção da rede neural utilizada neste trabalho. A título de exemplo, no final desse Capítulo há os resultados obtidos com a rede neural utilizada para estimar a corrente produzida por um conversor CC-CC.

No Capítulo 4, a topologia de rede neural apresentada no Capítulo 3 foi utilizada para estimar a velocidade do MSIP para o acionamento do mesmo, usando a rede neural para substituir o IPLL empregado no Capítulo 2. Para tal, a rede foi testada considerando duas bases de dados diferentes. Resultados de simulação foram apresentados, incluindo uma comparação entre as 4 abordagens, considerando diferentes parâmetros, para o controle de acionamento de MSIP implementadas e simuladas.

Por fim, apresentam-se as conclusões derivadas deste trabalho, as dificuldades encontradas e os problemas ainda não resolvidos como propostas para trabalhos futuros.

1 ACIONAMENTO DO MOTOR SÍNCRONO DE ÍMÃ PERMANENTE COM SENSOR DE VELOCIDADE

Este capítulo tem como objetivo apresentar o acionamento do Motor Síncrono (MSIP) de Ímã Permanente com o sensor de posição. Neste contexto, de início foi apresentado resumidamente os tipos de máquinas síncronas, conforme a montagem física do rotor, prosseguindo com a modelagem das equações de estado [7, 9, 10]. Na sequência foi descrito o modelo de acionamento do MSIP com o controle por orientação de campo (FOC - *Field Oriented Control*), com o uso do sensor para obter a informação tanto da velocidade quanto da posição angular. Tal configuração consistiu na primeira abordagem proposta neste trabalho, a qual serviu de base para analisar o desempenho das outras abordagens propostas em que a velocidade e a posição foram estimadas.

1.1 Conceitos básicos da máquina síncrona

De forma semelhante a outras máquinas elétricas, a máquina síncrona tem um estator constituído por bobinas alimentadas por uma fonte de corrente alternada, geralmente trifásica. A diferença frente outras máquinas elétricas está localizada basicamente na construção do rotor, conforme é apresentado na Figura 3. A Figura 3a ilustra a seção transversal da máquina síncrona convencional com o rotor bobinado e alimentado por uma tensão contínua. Por outro lado, na Figura 3b observa-se o emprego de um ímã permanente em vez de um rotor bobinado. O uso do ímã permanente faz com que o fluxo seja autoinduzido no rotor, sem a necessidade de uma fonte de alimentação adicional que o magnetize. Nestas condições, considerando ainda que não ocorre desmagnetização do ímã, o fluxo no rotor é assumido como constante.

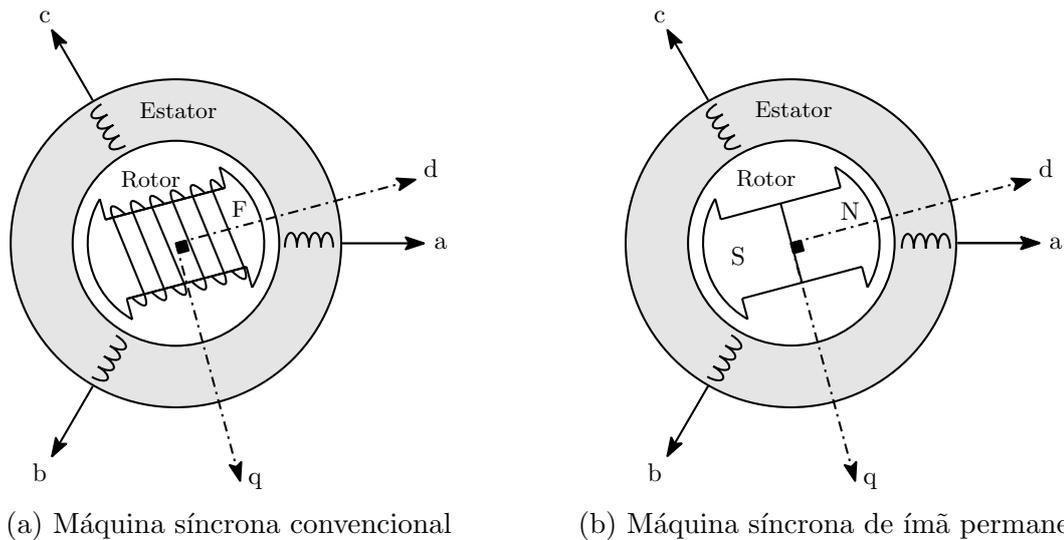


Figura 3: Rotor de um máquina síncrona convencional (a) e de uma máquina síncrona de ímã permanente (b).

Há algumas variações na estrutura e na implementação do ímã do rotor. As variantes permitem classificar tais máquinas em função da estrutura e posição do ímã. Na Figura 4 é ilustrada a seção transversal desse tipo de máquina. É conveniente ressaltar algumas das diferenças mais importantes entre estes dois tipos de MSIP.

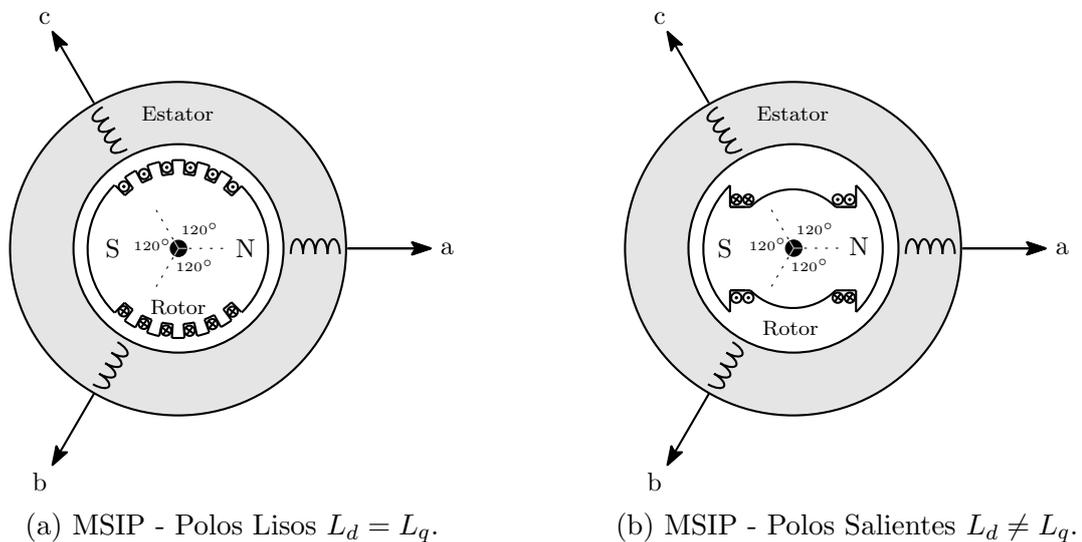


Figura 4: Variantes de ímãs de MSIPs.

O MSIP de polos lisos, ou polos cilíndricos, conforme a Figura 4a, é constituído por estator e ímã no rotor cilíndricos. Este tipo de montagem permite que a distribuição do entreferro seja uniforme, possibilitando operação em alta velocidade. Consequentemente, por operar em velocidades elevadas, esta configuração apresenta um número pequeno de polos, normalmente de 2 a 4 polos.

Por outro lado, o MSIP de polos salientes, apresentado na Figura 4b, apresenta estator cilíndrico com o rotor composto por ímãs salientes. Com isto, a distribuição do entreferro é variável, o que permite operar em baixas velocidades. Consequentemente, esta máquina, normalmente, tem mais de 4 polos.

1.2 Modelagem da Máquina Síncrona de Ímã Permanente

Inicialmente, consideremos a máquina síncrona convencional. A Figura 5 ilustra os circuitos equivalentes do rotor e estator, com as respectivas alimentações independentes. O MSIP é modelado a partir do circuito equivalente da máquina síncrona [7], conforme segue.

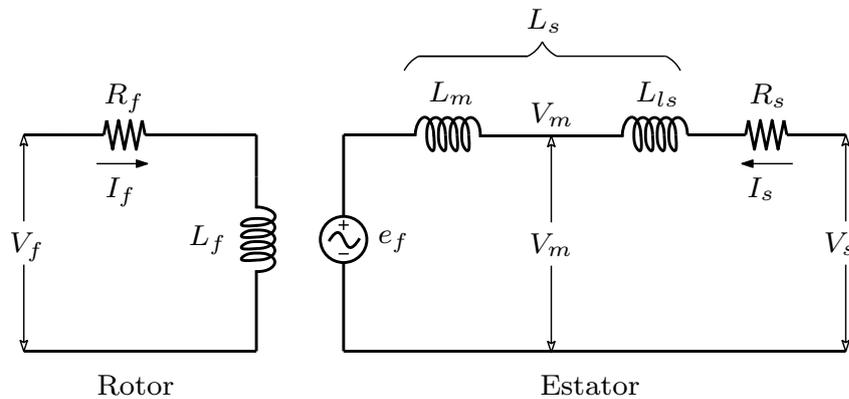


Figura 5: Circuito equivalente de uma máquina síncrona convencional.

O MSIP, por não apresentar circuito de alimentação no rotor, tem o circuito equivalente do rotor simplificado por um fluxo constante, assumindo que o ímã não desmagnetize. Para a modelagem do MSIP neste trabalho considerou-se um rotor cilíndrico com 2 polos. A equação

$$\omega_m = \frac{2}{p} \omega_e, \quad (1)$$

corresponde à relação entre as velocidades elétrica (ω_e) e mecânica (ω_p) do MSIP, onde p é o número de par de polos. Desta forma, sendo p igual a 2, tem-se $\omega_m = \omega_e$. A amplitude da tensão interna no estator induzida pelo rotor, e_f , é constante e dada por $\omega_m L_m I_f$.

Assumindo que o motor está conectado em estrela, as equações

$$v_a = R_a i_a + L_a \frac{\partial i_a}{\partial t} + e_{af}, \quad (2)$$

$$v_b = R_b i_b + L_b \frac{\partial i_b}{\partial t} + e_{bf}, \quad (3)$$

$$v_c = R_c i_c + L_c \frac{\partial i_c}{\partial t} + e_{cf}, \quad (4)$$

descrevem o modelo para o equivalente trifásico do MSIP [10]. Nas equações (2)–(4), v_a , v_b e v_c são as tensões no circuito de armadura do MSIP, i_a , i_b e i_c são as correntes e R_a , R_b e R_c são as resistências próprias das bobinas L_a , L_b e L_c , respectivamente. Por fim, as tensões internas são representadas por e_{af} , e_{bf} e e_{cf} . A Figura 6 apresenta o equivalente trifásico do MSIP com conexão em estrela.

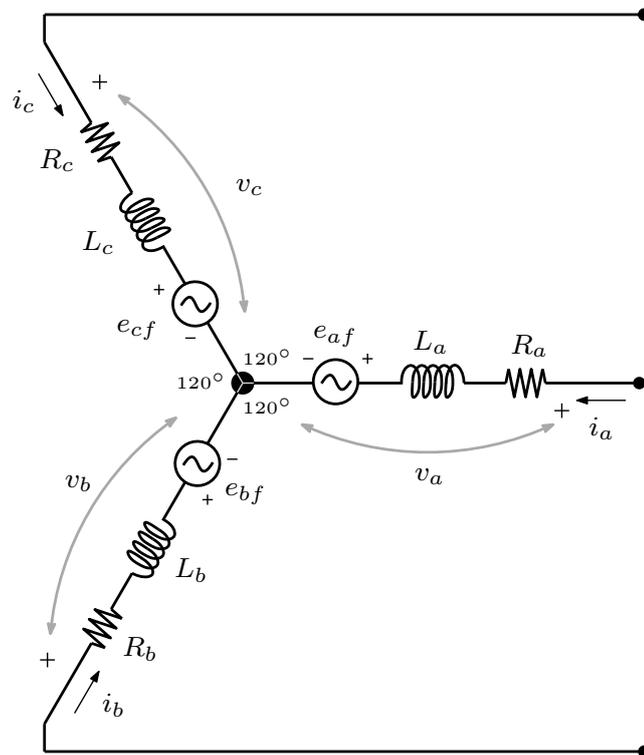


Figura 6: Circuito equivalente do MSIP com os terminais conectados em estrela.

Considerando $R_s = R_a = R_b = R_c$, o conjunto de equações que representam a

dinâmica do MSIP podem ser rearranjadas na forma matricial como

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{\partial}{\partial t} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}, \quad (5)$$

onde os fluxos produzidos por cada bobina estão representados por λ_a , λ_b , λ_c [9]. Estes são obtidos a partir do sistema

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \lambda_m \begin{bmatrix} \cos(\theta_m) \\ \cos(\theta_m - 120) \\ \cos(\theta_m - 240) \end{bmatrix}, \quad (6)$$

no qual λ_m é o fluxo de magnetização e θ_m representa a posição do rotor, o ângulo entre λ_m e a tensão da fase a (v_a). Na equação (6), têm-se as indutâncias próprias $L_{aa} = L_{bb} = L_{cc} = L_{ls} + L_m$, e as indutâncias mútuas $L_{ab} = L_{ba} = L_{ac} = L_{ca} = L_{bc} = L_{cb} = -\frac{L_m}{2}$, sendo L_m a indutância de magnetização e L_{ls} a indutância de dispersão da bobina da armadura.

As equações no sistema de coordenadas “ $a - b - c$ ” podem ser simplificadas usando um sistema de coordenadas composto por 2 eixos girantes, sendo um alinhado com o rotor e o outro atrasado de 90° . Estes 2 eixos girantes correspondem ao sistema de coordenadas “ $d - q$ ”, conforme descrito no item 1.2.1.

1.2.1 Representação do modelo matemático no Sistema de Coordenadas “ $d - q$ ”.

Os controles de acionamento do MSIP são usualmente feitos no sistema “ $d - q$ ” [11,12]. A representação do sistema de coordenadas “ $a - b - c$ ” para o sistema “ $d - q$ ” permite simplificar o modelo do MISP.

Essa transformação foi feita, neste trabalho, usando uma transformação intermediária auxiliar. Inicialmente, faz-se a mudança de coordenadas do sistema “ $a - b - c$ ” para o sistema “ $\alpha - \beta$ ” para então, em seguida, fazer a conversão do sistema “ $\alpha - \beta$ ” para o sistema “ $d - q$ ”. A Figura 7 mostra os fasores dos fluxos representados tanto em “ $a - b - c$ ” quanto em “ $\alpha - \beta$ ”. O mesmo procedimento é válido para a converter tensões e correntes.

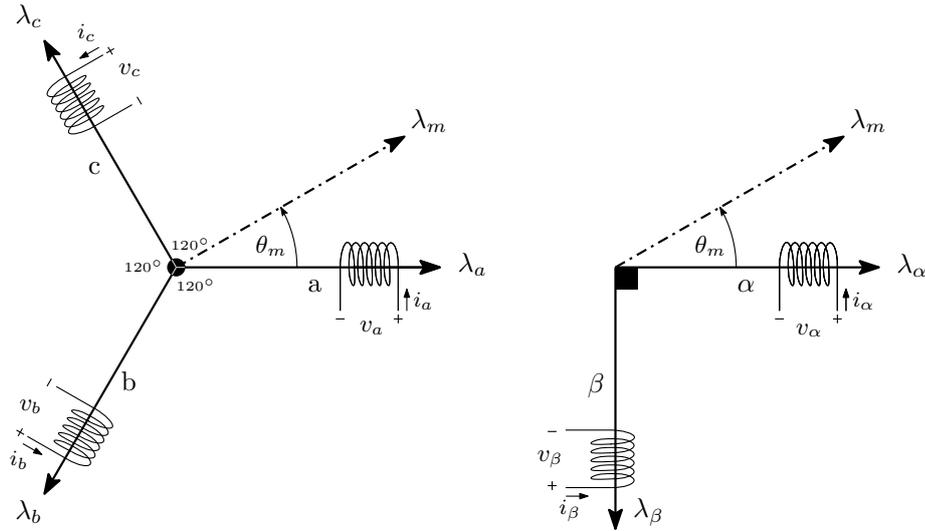


Figura 7: Representação dos fasores no sistema de coordenadas “ $a - b - c$ ” e “ $\alpha - \beta$ ”

Na representação fasorial, observa-se que a componente no eixo “ α ” está alinhada com a fase “ a ”. Por outro lado, a componente no eixo “ β ” está atrasada de 90° . Esta conversão é obtida por meio da solução matricial

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}, \quad (7)$$

que corresponde à solução proposta por Clarke [13]. A título de exemplo, acima, foi apresentada a conversão das tensões, o mesmo procedimento pode ser empregado para converter outros parâmetros como as correntes ou os fluxos. Estendendo esta abordagem, é possível re-escrever as equações dos terminais do estator no sistema de coordenadas “ $\alpha - \beta$ ”, obtendo a solução matricial

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = R_s \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + (L_{ls} + \frac{3}{2}L_m) \frac{\partial}{\partial t} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{\partial}{\partial t} \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix}, \quad (8)$$

onde $\lambda_\alpha = \lambda_m \cos(\theta_m)$, $\lambda_\beta = \lambda_m \sin(\theta_m)$ e a indutância síncrona, L_s , ao ser descrita em “ $\alpha - \beta$ ”, corresponde a $L_{ls} + \frac{3}{2}L_m$.

Após a conversão do sistema de coordenadas “ $a - b - c$ ” para o sistema “ $\alpha - \beta$ ”, aplica-se a conversão para o sistema de coordenadas “ $d - q$ ”. O sistema de coordenadas “ $d - q$ ” é composto por dois eixos girantes, com o eixo “ d ” alinhado com o rotor e o eixo

“ q ” atrasado de 90° . A Figura 8 ilustra o fluxo representado nos sistemas de coordenadas “ $\alpha - \beta$ ” e “ $d - q$ ”. A conversão do sistema de coordenadas “ $\alpha - \beta$ ” para “ $d - q$ ” ocorre por meio de

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \cos(\theta_m) & \sin(\theta_m) \\ -\sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}, \quad (9)$$

onde θ_m corresponde à posição angular do rotor. Esta é uma conversão para um sistema ortogonal girante (sistema de coordenadas “ $d - q$ ”) a partir do sistema ortogonal $\alpha - \beta$, por meio da transformada de Park [14].

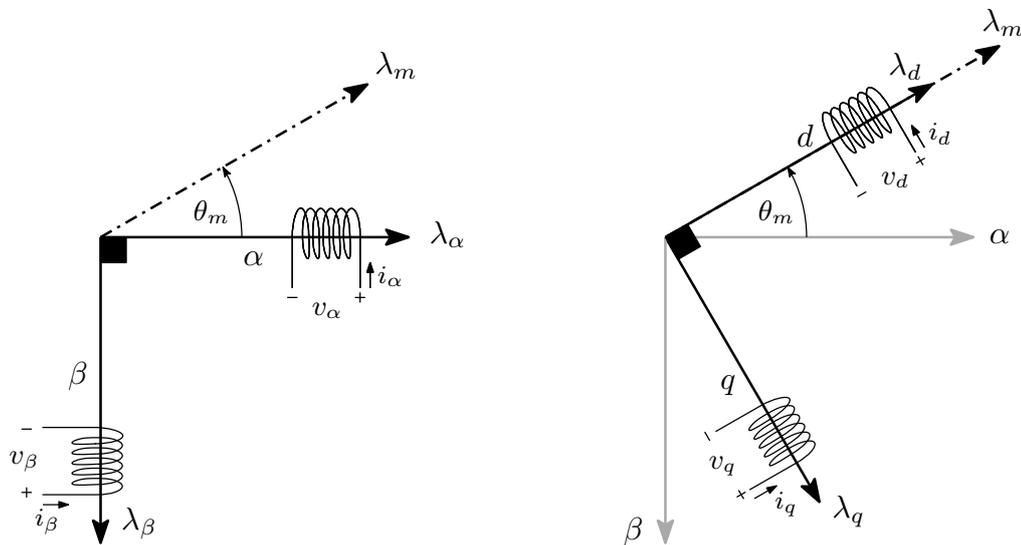


Figura 8: Representação do fluxo nos sistemas de coordenadas “ $\alpha - \beta$ ” e “ $d - q$ ”.

A partir da conversão das tensões e correntes para o sistema de coordenadas “ $d - q$ ”, obtém-se

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s & -\omega_m L_q \\ \omega_m L_d & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \omega_m \lambda_m \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (10)$$

Pelo fato de neste trabalho utilizar-se o MSIP de rotor liso, os parâmetros L_q e L_d são iguais à indutância síncrona (L_s) [8].

Os circuitos equivalentes dos eixos direto e em quadratura estão representados, respectivamente, na Figura 9 e na Figura 10. O fluxo de eixo direto é dado por $\lambda_d = L_d i_d$, enquanto o fluxo do eixo de quadratura é igual a $\lambda_q = L_q i_q$.

Por fim, observa-se que o torque elétrico no sistema de coordenadas “ $d - q$ ” é dado

por

$$T_e = \frac{3}{2} \left(\frac{P}{2} \right) \lambda_m \left[i_q - (L_d - L_q) i_d i_q \right]. \quad (11)$$

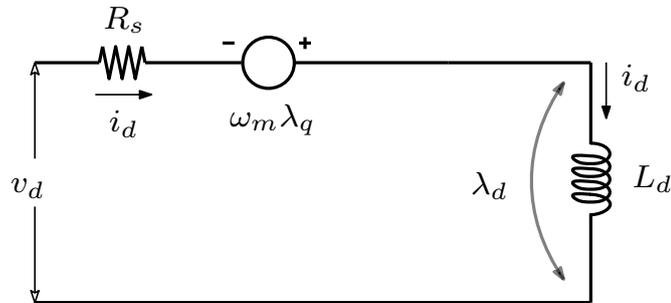


Figura 9: Circuito equivalente no eixo direto.

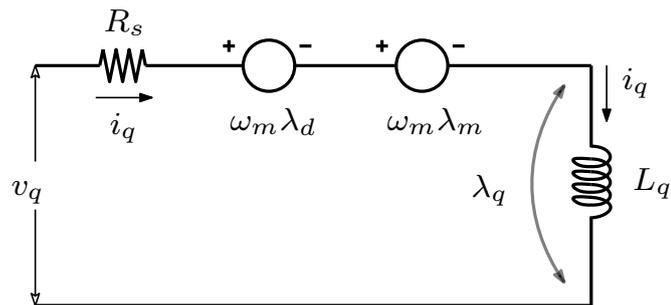


Figura 10: Circuito equivalente no eixo quadratura.

A apresentação simplificada do MSIP teve o intuito de introduzir conceitos e equações que regem esta máquina. A partir das equações base é possível modelar o comportamento dinâmico da máquina síncrona e derivar outros parâmetros importantes como potência e torque. No entanto, o objetivo deste trabalho reside no estudo do acionamento desta máquina por meio do controlador por orientação de campo. Neste sentido, este controle é descrito na próxima seção com a inclusão de resultados de simulação que auxiliam no entendimento da dinâmica da máquina em diferentes situações.

1.3 Controle do Motor Síncrono de Ímã Permanente

Conforme [15, 16], as técnicas de controle do MSIP são usualmente baseadas tanto **i)** na orientação de campo (FOC - *Field Oriented Control*) quanto **ii)** no controle direto

de torque e fluxo (DTC - *Direct Torque Control*) podem ser consideradas apropriadas para o acionamento do MSIP com uso de inversores do tipo fonte de tensão.

No controle FOC, as tensões de referência que o inversor deve produzir são determinadas a partir das projeções ortogonais das correntes do estator no sistema de coordenadas “ $d-q$ ”, em conjunto com a velocidade do motor, ω_m . Assim, o torque e o fluxo são obtidos indiretamente. O objetivo central consiste em controlar as correntes do estator tanto de eixo direto (i_d) quanto de quadratura (i_q), produzindo o torque e o fluxo necessários para controlar a velocidade do motor. Particularmente, a corrente i_d é forçada para ser igual a zero em regime permanente, para não desmagnetizar o MSIP, com o torque controlado pela corrente i_q . Nesta estratégia, devido aos controles das correntes i_d e i_q serem independentes entre si, é possível controlar o motor para um torque específico com a mínima corrente necessária, o que aumenta a eficiência do motor. Além disso, considerando o uso do MSIP de polos lisos, os indutores representados no sistema de coordenadas “ $d-q$ ” (L_d e L_q) são iguais, o que simplifica ainda mais o controle, considerando o cenário com a implementação do controle sem o uso do sensor de posição do rotor.

A técnica DTC utiliza as tensões e correntes do estator no sistema de coordenadas “ $d-q$ ”, em conjunto com a velocidade do motor, ω_m para produzir o sinal de controle. Nesta técnica, o torque e o fluxo são estimados a partir das tensões e correntes do estator, no sistema de coordenadas “ $d-q$ ”. O DTC apresenta como vantagem a possibilidade de eliminar os reguladores de corrente utilizados no FOC. No entanto, tem como desvantagem o difícil controle do torque em baixas velocidades por não usar controladores a partir das correntes do estator.

Do exposto, decidiu-se utilizar o controlador FOC por ser o mais simples de ser implantado.

1.3.1 Controle por Orientação de Campo (FOC)

Essencialmente, a partir da equação (10) determina-se as tensões no sistema de coordenadas (v_d, v_q) moduladas pelo conversor fonte de tensão. Neste sentido, este controlador é realimentado pelos parâmetros variantes no tempo que correspondem às correntes i_d e i_q , em conjunto com a velocidade ω_m . Desta forma, a Figura 11 mostra um esquemático simplificado de circuito para o acionamento do MSIP usando controlador FOC.

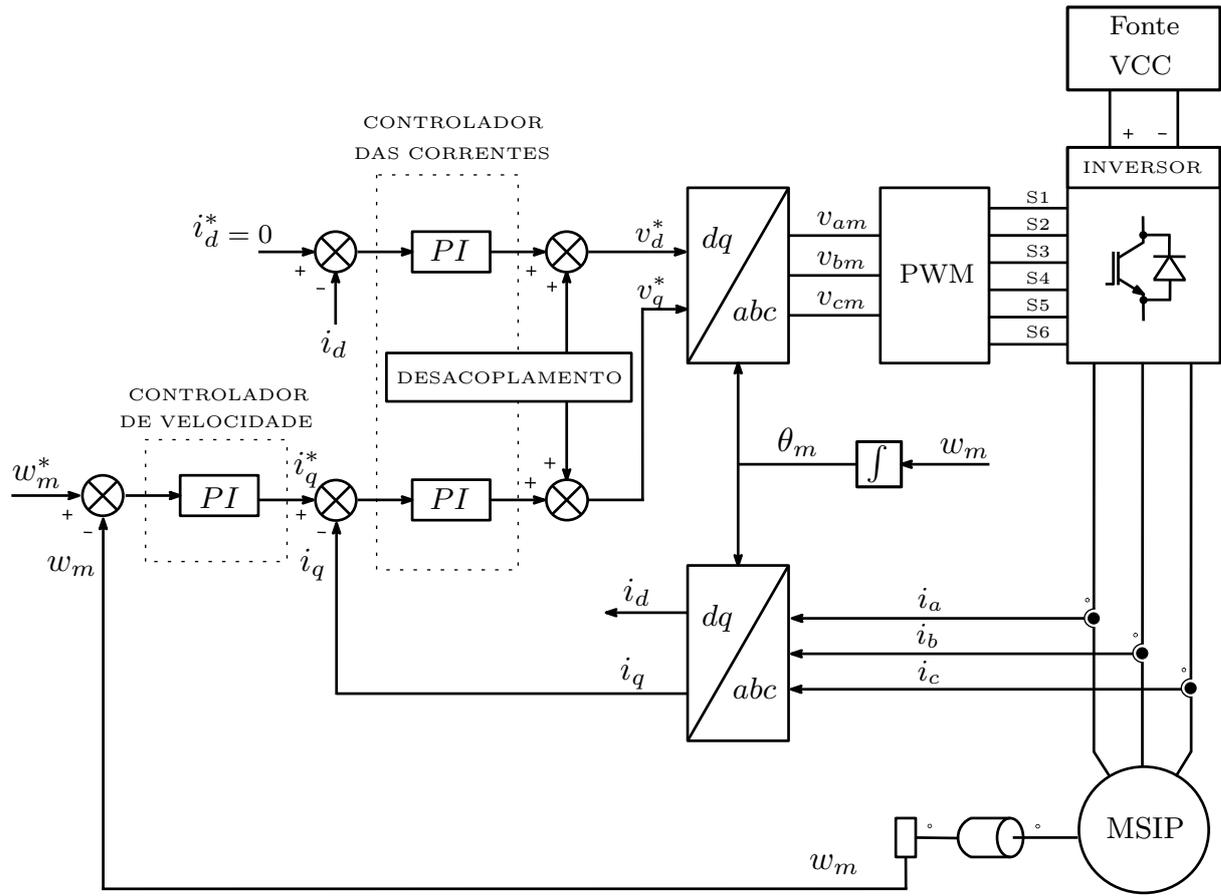


Figura 11: Esquema simplificado do circuito com o controle por orientação de campo.

Neste trabalho foi considerado o uso de MSIP de polos não salientes e, portanto, a corrente de eixo direto, i_d , deve ser igual a zero. A partir disso, pode-se afirmar que a fase da corrente de eixo direto está alinhada com a fase do fluxo magnético, λ_m . Da mesma forma, impõe-se que a fase da corrente de quadratura, i_q , esteja deslocada em 90° da fase de λ_m , de modo que as correntes i_d e i_q sejam ortogonais entre si, conforme ilustrado na Figura 8. Além disso, substituindo $i_d = 0$ na equação (11) é possível simplificar a expressão para o cálculo do torque elétrico (12). Deste modo, considerando tão somente que no MSIP, o fluxo magnético, λ_m , é constante, a corrente de quadratura, i_q , está diretamente relacionada com a produção do torque do motor, que, com as simplificações e considerações anteriores, é dado por:

$$T_e = \frac{3}{2} \left(\frac{p}{2} \right) \lambda_m i_q. \quad (12)$$

Ainda com relação à Figura 11, há um bloco denominado “Desacoplamento”, no qual são calculados os termos complementares utilizados na determinação das tensões v_d

e v_q em função das correntes i_q e i_d . Pode ser observado que o controle FOC é composto por dois laços, sendo o primeiro utilizado para controlar a velocidade do motor e o segundo utilizado no controle das correntes do estator a partir das tensões produzidas pelo conversor.

Inicialmente, utiliza-se um controlador PI no qual o sinal de entrada corresponde ao erro entre a velocidade de referência, ω_m^* , e a velocidade do motor, ω_m . A saída deste controlador PI é a corrente de referência, i_q^* , que o inversor deve produzir. Esta corrente está diretamente relacionada com o torque produzido pelo motor. De fato, o incremento do erro entre ω_m^* e ω_m indica a necessidade do aumento do torque que, neste controle, está associado à corrente de referência i_q^* . A variação de i_q^* tende a zero à medida que ω_m tende para ω_m^* , com o motor operando em regime permanente.

Em seguida, a partir das correntes de armadura do motor i_a , i_b e i_c , obtêm-se as equivalentes no sistema “ $d-q$ ”, com uso da posição angular θ_m . Neste segundo laço, também composto por controladores do tipo PI, são determinadas as tensões de referência que o conversor deve produzir. Para tal, as correntes i_a , i_b e i_c são convertidas para o sistema de coordenadas “ $d - q$ ”, representadas por i_d e i_q . Na sequência, estas correntes convertidas são comparadas com as correntes de referência i_d^* e i_q^* , respectivamente, com os erros destas comparações correspondendo aos sinais de entrada dos controladores PI utilizados neste segundo laço. Particularmente, é importante comentar que como as tensões e correntes de alimentação do motor são compostas, majoritariamente, por uma única componente harmônica, a transformada de Park, neste caso, resulta em correntes compostas por componentes contínuas [14], o que é ajustado para o uso de controladores do tipo PI.

Observa-se que as tensões e correntes do estator, no sistema de coordenadas “ $d-q$ ”, estão relacionadas entre si. Com isso, são calculadas as componentes denominadas por u_d e u_q , que estão relacionadas com as correntes i_q e i_d via

$$u_d = -\omega_m L_q i_q, \quad (13)$$

$$u_q = \omega_m L_d i_d + \omega_m \lambda_m. \quad (14)$$

Conseqüentemente, a partir dos parâmetros mencionados, as tensões de referência

que o conversor deve produzir, no sistema de coordenadas “ d - q ” são dadas por:

$$v_d^* = u_d + K_p \cdot (i_d^* - i_d) + K_i \cdot \int (i_d^* - i_d), \quad (15)$$

$$v_q^* = u_q + K_p \cdot (i_q^* - i_q) + K_i \cdot \int (i_q^* - i_q). \quad (16)$$

Com base nos cálculos feitos para determinar as tensões de referência a serem produzidas pelo inversor é observada a influência que as correntes i_d e i_q têm sobre a dinâmica do motor. Tal fato é observado tanto nas malhas dos controladores PI quanto nas tensões internas do estator, u_d e u_q .

É importante comentar que por tratar-se de controladores em cascata, os ganhos dos controladores foram ajustados de modo que a dinâmica do controlador de velocidade (primeiro laço) fosse mais lenta que a do controle das correntes do estator (segundo laço). Tal metodologia foi empregada em outros trabalhos com o uso de controladores em cascata, conforme apresentados em [17, 18].

1.3.2 Resultados de simulação com controlador por orientação de campo convencional

Os principais parâmetros do MSIP empregados na simulação estão descritos na Tabela 1 e na Tabela 2. Este modelo de MSIP é utilizado nos próximos capítulos desta dissertação.

Tabela 1: Parâmetros do motor.

Descrição	Valor	Unidade
Resistência do estator (R_s)	0,0158	Ω
Indutância ($L_d = L_q$)	4,85	mH
Número de polos (p)	2	
Momento de inercia (J)	0,03	$kg.m.m$
Constante de tempo (T)	4,33	s

Visando a implementação do esquemático apresentado na Figura 11, onde o objetivo principal é controlar a velocidade mecânica ω_m , é necessário considerar (para sintonizar os ganhos dos controladores PI) que, a princípio, a malha de controle das correntes i_d e i_q apresenta dinâmica mais rápida que a da malha de controle da velocidade ω_m . Assim,

Tabela 2: Dados nominais do motor.

Descrição	Valor	Unidade
Potencia nominal (P_N)	20	kW
Velocidade nominal (W_N)	377	rad/s
Corrente nominal (I_N)	40	A
Tensão nominal (V_N)	440	V

os ganhos escolhidos para o controlador das correntes, tanto a corrente do eixo direto i_d , quanto a corrente do eixo de quadratura i_q são, $k = 10$ e a constante de tempo $T = 5 ms$. Por outro lado, o ganho escolhido para o controlador PI da velocidade mecânica é $k = 0,6$ e neste caso $T = 50 ms$. Estes valores foram obtidos a partir de testes preliminares com este circuito simulado.

Na Figura 12 apresenta-se o ângulo de posicionamento do motor θ_m com o motor operando na velocidade nominal de $377rad/s$. Em seguida, na Figura 13 é apresentada uma comparação entre as formas de onda da velocidade mecânica de referência do motor ω_m^* e a velocidade mecânica fornecida pelo sensor de posicionamento ω_m . Basicamente, o teste é iniciado com o motor operando a uma velocidade de $150rad/s$, que corresponde a 40% da velocidade nominal do motor. Na sequência, a velocidade é elevada gradativamente de $150rad/s$ para $377rad/s$ e permanece nesta velocidade por menos de 1 segundo. Por fim ocorre o último transitório que consiste na redução gradativa do sinal de referência da velocidade nominal para $200rad/s$. Conforme é ilustrado neste resultado, a velocidade do motor (ω_m) está em conformidade com a referência, indicando o desempenho satisfatório do controlador FOC. Na Figura 14 apresenta-se o torque mecânico da carga no mesmo intervalo de tempo da (ω_m) . Observa-se a variação no primeiro transitório de $18N.m$ para, aproximadamente, $57N.m$. No segundo transitório o torque foi reduzido para $25N.m$, aproximadamente, quando o motor opera em regime permanente. Na Figura 15 observa-se a forma de onda da corrente de quadratura i_q , a mesma que está diretamente relacionada com a produção do Torque.

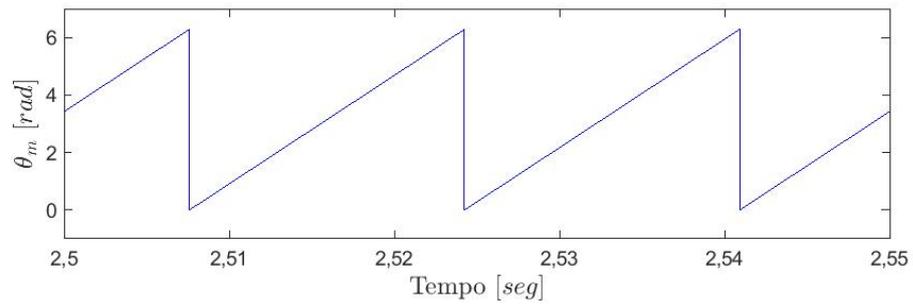


Figura 12: Ângulo de posicionamento do rotor θ_m .

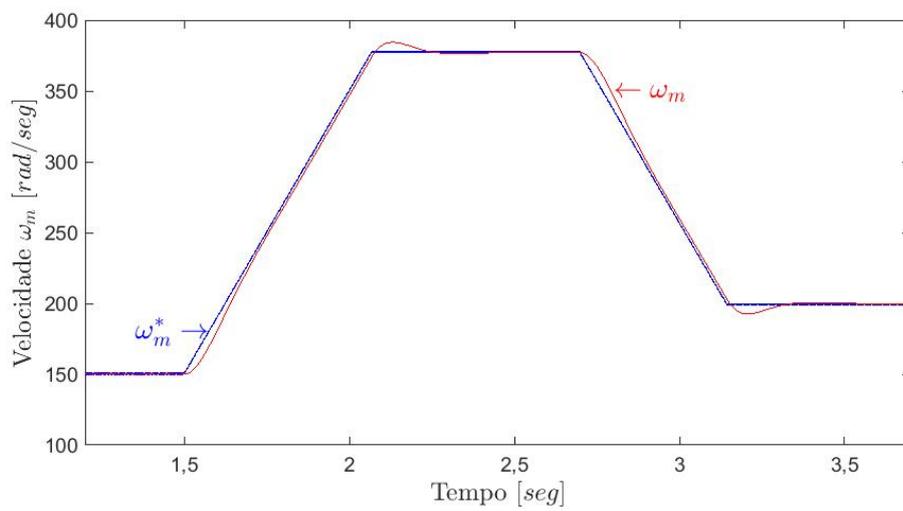


Figura 13: Comparação entre as velocidades ω_m e ω_m^*

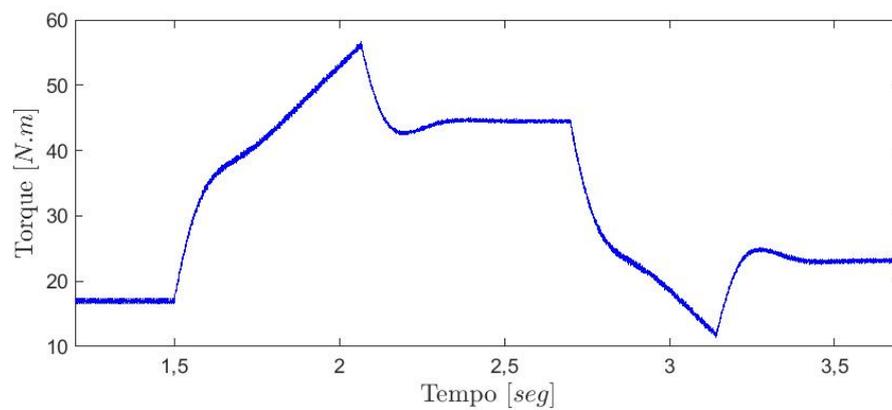


Figura 14: Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.

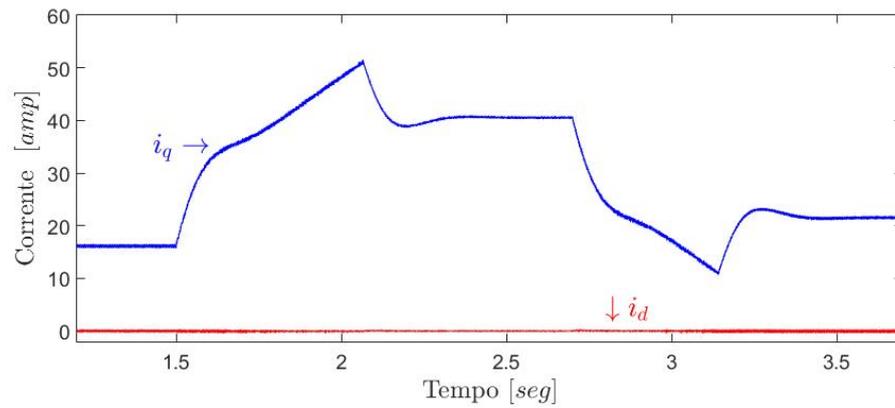


Figura 15: Correntes do estator tanto de eixo direto (i_d) quanto de quadratura (i_q).

Desta forma, neste Capítulo foram apresentados algumas das características do MSIP com o acionamento feito por meio controlador orientador pelo campo. Foi observado que trata-se de um controlador de simples implementação com a velocidade controlada a todo instante durante a simulação.

Continuando com a descrição do estudo feito nesta dissertação, o próximo Capítulo trata do primeiro controlador *sensorless* proposto neste trabalho, que consistiu no estimador da velocidade e da posição a partir de um observador de fluxo em conjunto com um circuito de sincronismo IPLL.

2 ACIONAMENTO DO MSIP SEM SENSOR DE POSIÇÃO COM USO DE UM OBSERVADOR DE FLUXO EM CONJUNTO COM O CIRCUITO DE SINCRONISMO

No Capítulo anterior foi apresentado o controle por orientação de campo da velocidade do motor com uso de sensor de posição. Tais sensores de posição, instalados no eixo de motores, aumentam o custo de instalação e manutenção [19]. Conseqüentemente, há trabalhos na literatura propondo diferentes técnicas para estimar tanto a posição do rotor quanto a sua velocidade, constituindo como técnicas de controle sem sensores (*sensorless motor control*).

Há algumas categorias de técnicas sem sensores para o acionamento e controle do MSIP [20]. Na primeira categoria, têm-se os métodos de controle para baixas velocidades, incluindo velocidade nula. Dentre esses, encontram-se métodos com base na produção de tensões com alta frequência. Basicamente, a produção de tensões de alta frequência nos enrolamentos do motor geram correntes induzidas de alta frequência, onde os sinais correspondentes são processados para estimar a posição do rotor [19, 21, 22].

A segunda categoria compreende técnicas de controle para velocidades médias e altas do motor. O presente trabalho se enquadra nesta categoria, já que o motor opera na faixa entre 150 rad/s e 377 rad/s , aproximadamente. Para isso, encontram-se proposições de observadores de fluxo [8, 23, 24], propostas de observadores com base nos modos deslizantes (SMO - *Sliding Mode Operation*) [25, 26], com Filtros de Kalman [27–29], além de técnicas baseadas em inteligência artificial [30, 31], dentre as quais podemos enquadrar este trabalho.

Na seqüência, é apresentado um controle sem sensores de um MSIP baseado em um observador de fluxo, em conjunto com um circuito de sincronismo [8]. Esse é um exemplo empregado como comparação para o método desenvolvido neste trabalho, dada a sua simplicidade de implantação e por apresentar resposta dinâmica similar à observada para o controlador FOC com o sensor de posição.

2.1 Controle sem sensores de um MSIP baseado num Observador de Fluxo

O esquemático simplificado é representado na Figura 16, vê-se que é empregado um observador de fluxo melhorado [8]. A velocidade e o ângulo são estimados a partir

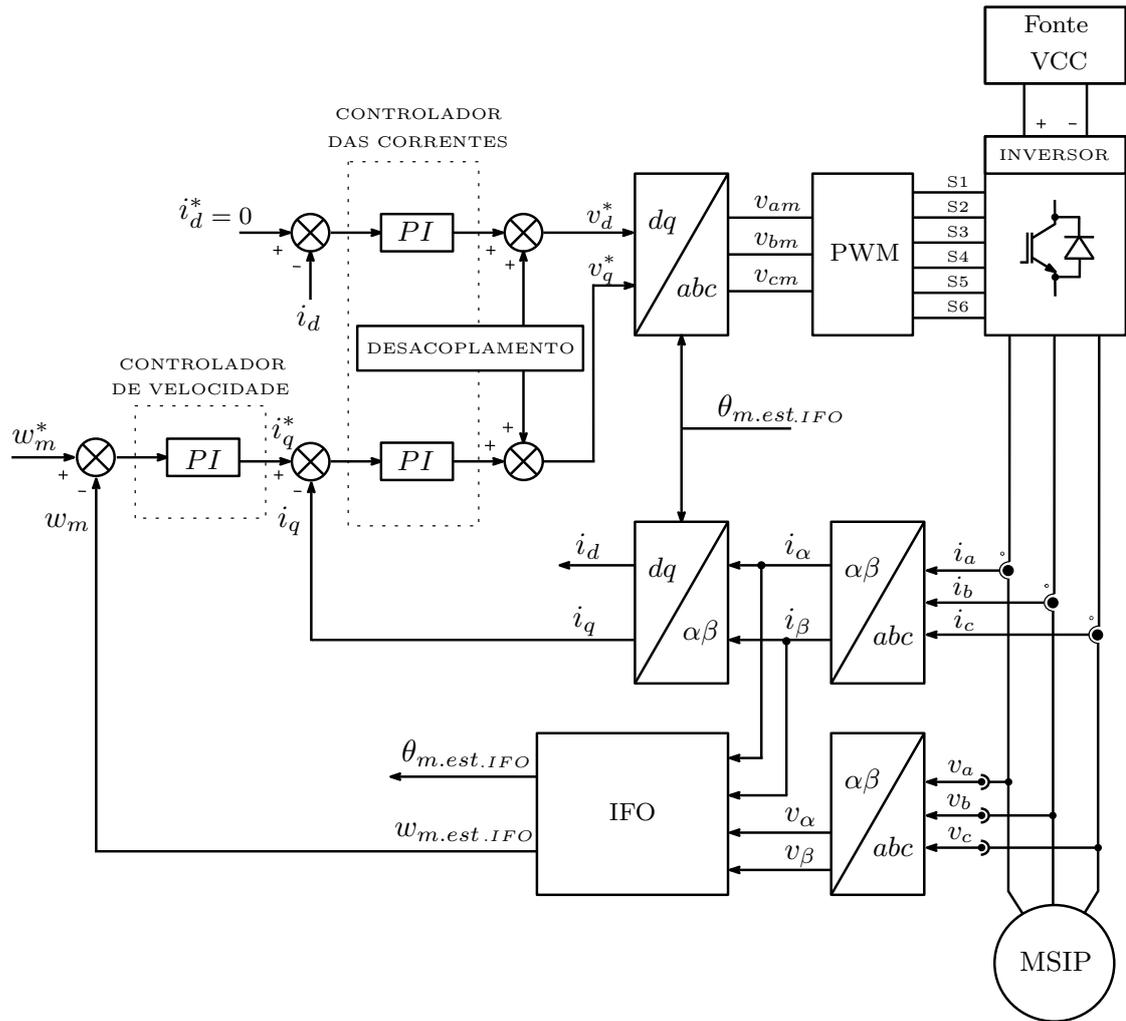


Figura 16: Esquemático simplificado do circuito com o controle sem sensores IFO.

das tensões e correntes na alimentação do motor por v_α , v_β e i_α , i_β , respectivamente.

Na Figura 17 é ilustrado o diagrama de blocos do observador de fluxo IFO. O primeiro bloco (na esquerda) é um observador de fluxo convencional, que a partir das tensões e correntes v_α , v_β e i_α , i_β calcula os fluxos no sistema de coordenadas “ α - β ”, representados por λ_α e λ_β .

Após o observador de fluxo convencional, empregam-se filtros passa-altas para eliminar a componente contínua (componente CC) presente nos fluxos em função da ação dos integradores. Estas componentes CC também podem ser eliminadas a partir de integradores generalizados de segunda ordem (SOGI - *Second-Order Generalized Integrator*), no entanto o uso de filtros passa-altas é mais simples de ser implantado e resultados de simulação indicam que eles são capazes de atenuar muito severamente a componente CC. Por essa razão, utilizamos filtros passa-altas.

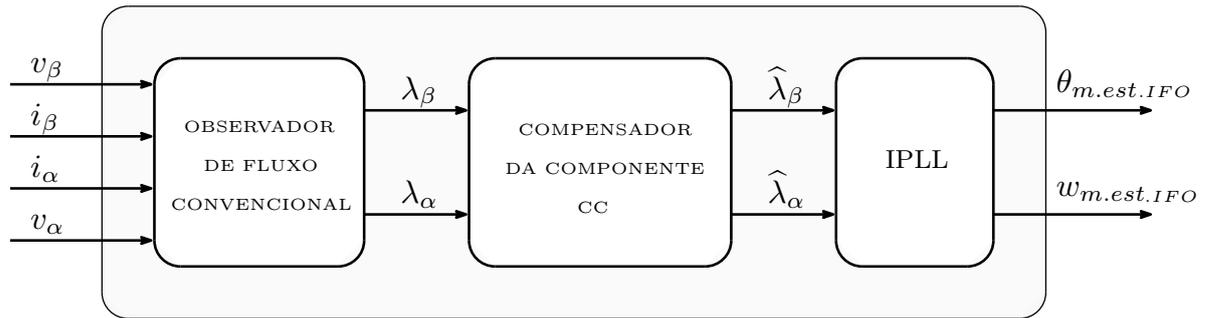


Figura 17: Diagrama de blocos para determinação da posição e velocidade com no controle sem sensores IFO.

Por fim, o último bloco é o IPLL (*Improved Phase-Locked-Loop*), onde a posição e o ângulo são estimados usando um PLL. Suas entradas são os fluxos λ_α e λ_β filtrados $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$, respectivamente.

Os princípios de funcionamento de cada um dos blocos na Figura 17, equações relevantes, sintonia dos ganhos, entre outros aspectos, estão descritos a seguir.

2.1.1 Observador de Fluxo Convencional

A partir do sistema de equações (8) implementou-se um observador de fluxo convencional. Neste observador, para o MSIP de polos lisos, os fluxos λ_α e λ_β são dados por:

$$\lambda_\alpha = \int (v_\alpha - R_s i_\alpha) dt - L_s i_\alpha; \quad (17)$$

$$\lambda_\beta = \int (v_\beta - R_s i_\beta) dt - L_s i_\beta. \quad (18)$$

Nessas equações, L_s representa a indutância do estator.

Aplicando as equivalências $\lambda_\alpha = \lambda_m \cos(\theta_m)$ e $\lambda_\beta = \lambda_m \sin(\theta_m)$ e como θ_m é a posição angular do motor, podemos estimar θ_m através da relação

$$\theta_m = \tan^{-1} \left(\frac{\lambda_\beta}{\lambda_\alpha} \right). \quad (19)$$

Na Figura 18 é apresentado um diagrama de blocos que estima θ_m [8], a partir das tensões e correntes no sistema de coordenadas “ α - β ”, com $R_s=0,1058\Omega$ e $L_s=4,85mH$.

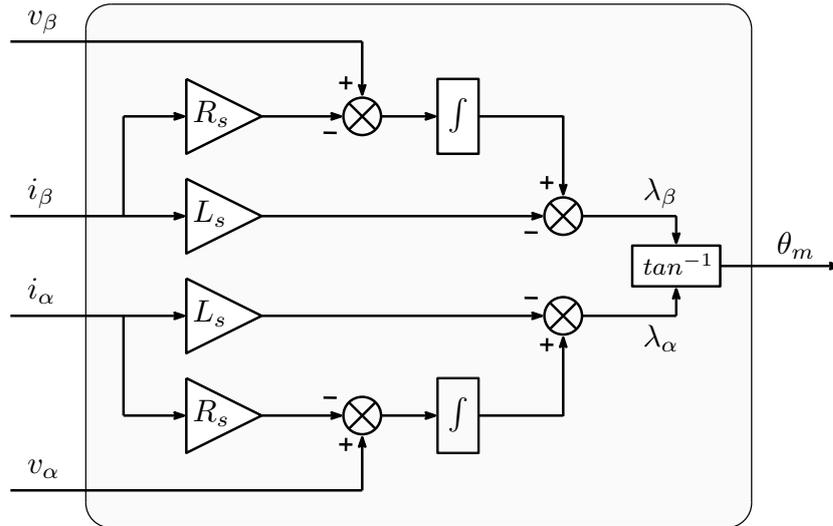


Figura 18: Diagrama de blocos do Observador de Fluxo convencional.

2.1.2 Compensador da componente CC

Devido aos fluxos λ_α e λ_β apresentarem componentes contínuas, utilizaram-se filtros do tipo passa-altas para eliminá-las. A função de transferência do filtro passa-altas utilizado é igual a

$$G(s) = \frac{ks^2}{s^2 + 2\zeta\omega_c s + \omega_c^2}, \quad (20)$$

com $k = 1$, frequência de corte, $f_c = (\omega_c/2\pi) = 5H_z$ e constante de amortecimento, $\zeta = 0,7$. Desta forma, os fluxos sem a componente CC, representados por $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$, são obtidos a partir das equações dadas por:

$$\begin{bmatrix} \hat{\lambda}_\alpha \\ \hat{\lambda}_\beta \end{bmatrix} = \frac{ks^2}{s^2 + 2\zeta\omega_c s + \omega_c^2} \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix}. \quad (21)$$

A Figura 19 apresenta a resposta em frequência do filtro passa-altas descrito acima. Observa-se que na faixa de atuação do estimador, entre 150 rad/s a 377 rad/s , o filtro apresenta ganho unitário com pequeno desvio de fase. No entanto, este desvio de fase não comprometeu o desempenho do estimador conforme observado nos resultados de simulação apresentados neste Capítulo.

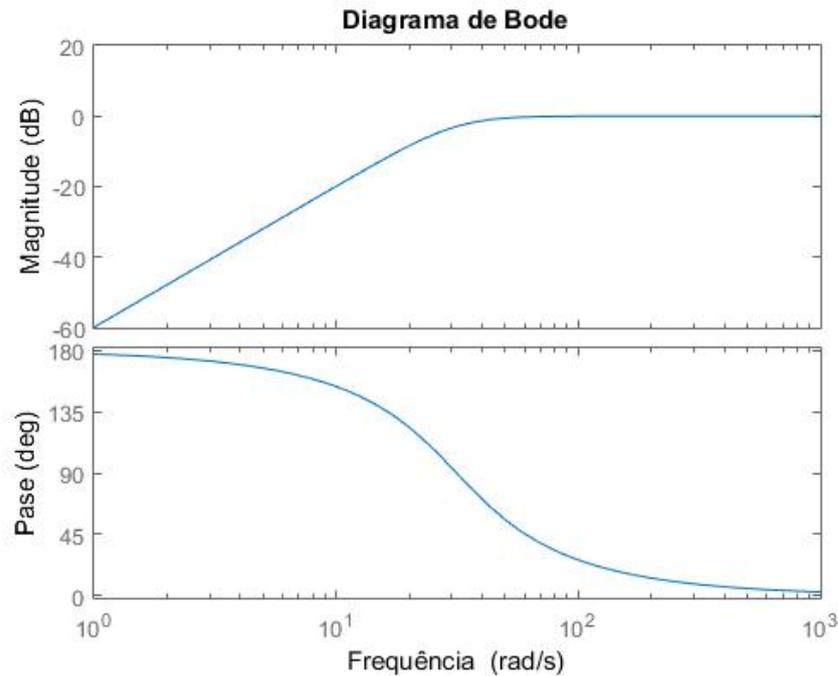


Figura 19: Resposta em frequência do filtro passa altas de segunda ordem com frequência de corte em 5 Hz.

2.1.3 Improved Phase Locked loop (IPLL)

O IPLL determina a frequência e o ângulo de fase, $\omega_{m.est.IFO}$ e $\theta_{m.est.IFO}$, respectivamente, a partir dos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$. Conforme ilustrado na Figura 20, a partir da soma dos produtos dos fluxos normalizados pelos sinais de realimentação obtém-se o sinal de erro (e). O IPLL opera em regime permanente quando o valor médio do sinal de erro (e) for igual a zero, situação esta em que os sinais de realimentação estão sincronizados com os sinais de entrada.

A frequência angular estimada, representada na Figura 20 por $\omega_{m.est.IFO}$, é utilizada no laço de controle da velocidade indicada. Para implementação do IPLL foram sintonizados os ganhos do controlador PI a partir de testes em simulação. A função de transferência do controlador PI é dada pela expressão

$$G_{(S)PI} = k * \frac{(1 + ST)}{ST}, \quad (22)$$

com o ganho proporcional igual a $k = 30$ e a constante de tempo (T) igual a $100\mu s$.

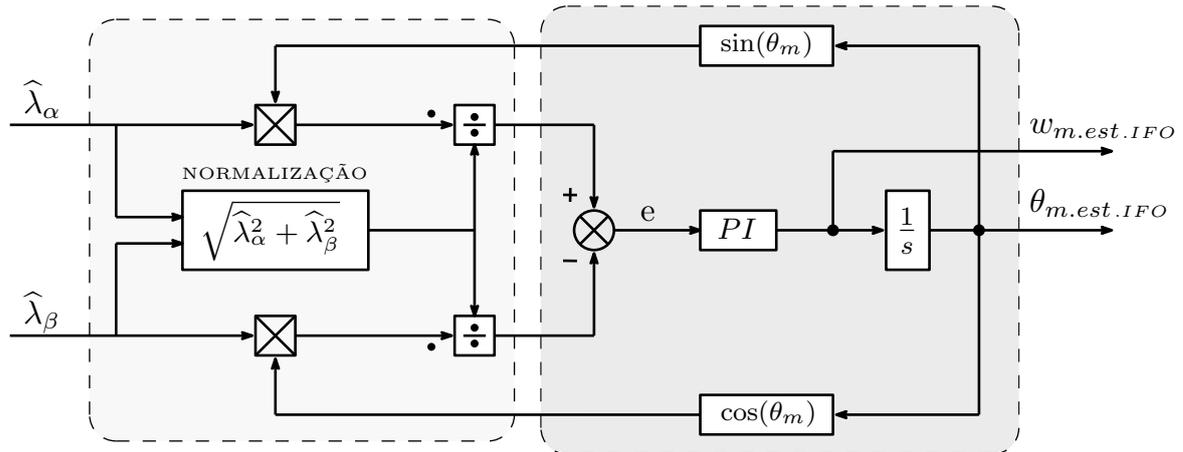


Figura 20: Diagrama de blocos do IPLL utilizado para estimar a velocidade ($w_{m.est.IFO}$) e a posição angular ($\theta_{m.est.IFO}$).

2.2 Resultados de Simulação

Nesta seção, apresentam-se os resultados de simulação para o mesmo caso teste usado no Capítulo 1. No entanto, agora tanto a velocidade mecânica quanto a posição angular são determinados a partir do observador do fluxo com o auxílio do IPLL, conforme ilustrado na Figura 16.

Na Figura 21 são apresentados os fluxos λ_α e λ_β , antes da compensação da componente contínua (componente CC). As formas de onda dos fluxos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$, após da compensação da componente CC, estão ilustradas na Figura 22. Observa-se a efetiva compensação da componente contínua pelo filtro passa-altas de 2ª ordem, mantendo-se entre os fluxos, um ângulo de deslocamento de $\pi/2$ antes e após da compensação CC.

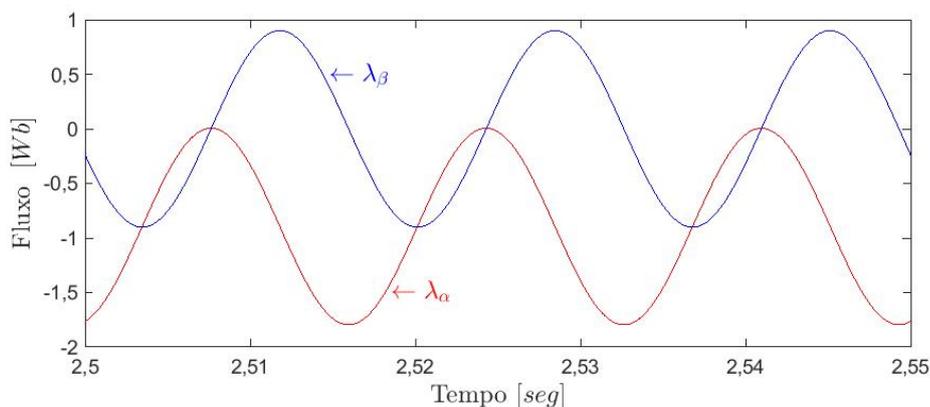


Figura 21: Fluxos λ_α e λ_β no Observador de Fluxo Convencional.

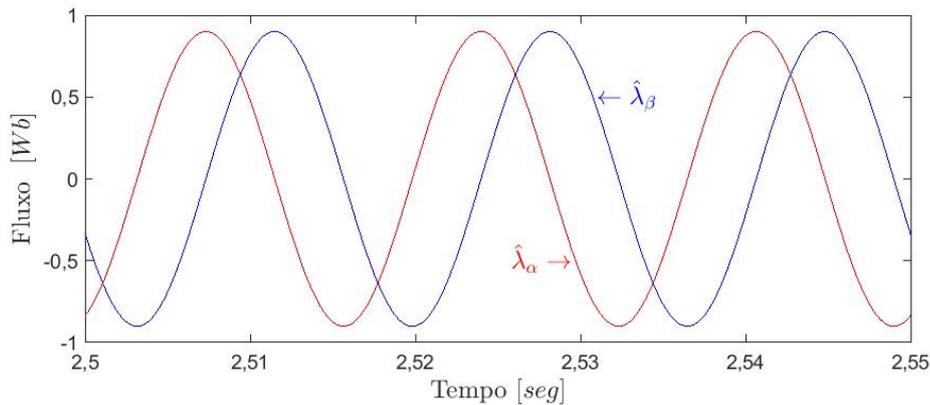


Figura 22: Fluxos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$ depois do compensador CC.

Na Figura 23 é ilustrada a posição angular calculada a partir do observador de fluxo convencional, usando os fluxos λ_α e λ_β sem compensação de componente contínua CC. Na Figura 24 tem-se a posição angular calculada a partir dos fluxos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$ após da compensação CC. Nota-se que o ângulo calculado após a compensação da componente CC varia entre 0 a 2π “linearmente” em cada segmento, diferentemente do que ocorre na Figura 23, na qual há variações abruptas da posição angular quando θ_m é forçado para ser igual a 2π a cada fim de ciclo. Na Figura 25, apresenta-se a posição angular determinada pelo IPLL $\theta_{m.est.IFO}$. É importante comentar que as posições angulares obtidas tanto a partir do observador convencional após a filtragem da componente CC quanto a partir do IPLL estão em fase, indicando que o IPLL sincronizou corretamente com o ângulo de fase dos fluxos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$.

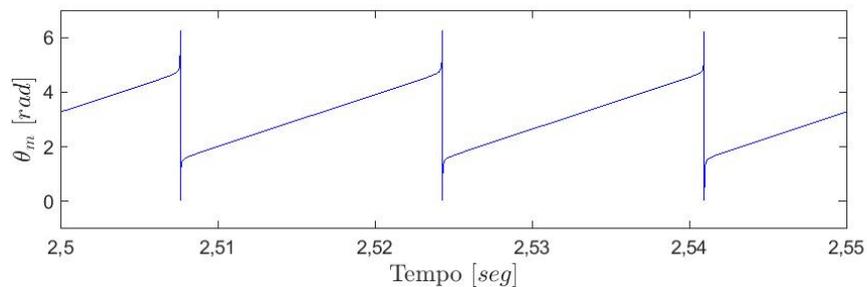


Figura 23: Posição angular do rotor θ_m obtida com os fluxos antes da compensação da componente CC (λ_α e λ_β).

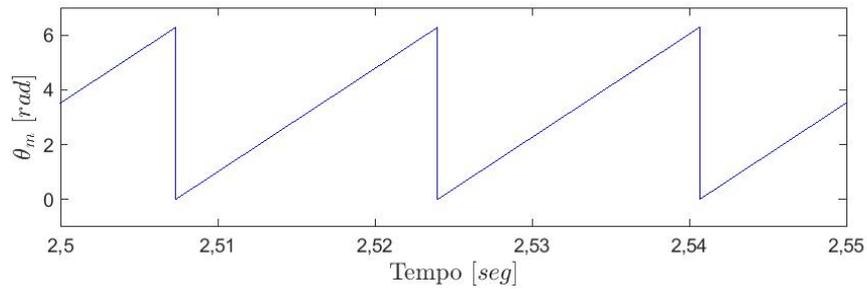


Figura 24: Posição angular do rotor θ_m obtida com os fluxos após a compensação da componente CC ($\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$).

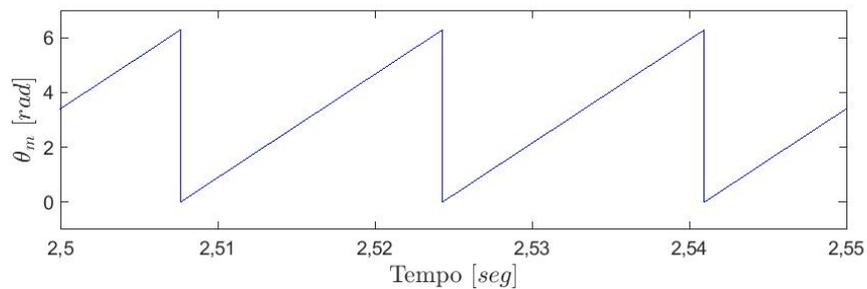


Figura 25: Posição angular do rotor $\theta_{m.est.IFO}$ obtida com o IPLL.

Na Figura 26 há uma comparação entre a velocidade mecânica estimada a partir do IPLL ($\omega_{m.est.IFO}$) e, a velocidade de referência ω_m^* . Este teste foi similar ao feito no Capítulo 1, no qual a velocidade de referência, ω_m^* , começa com $150rad/s$ sendo gradativamente aumentada para $377rad/s$, que corresponde à velocidade nominal da máquina. Em seguida, ω_m^* foi reduzida gradativamente para $200rad/s$. Neste caso teste, a velocidade estimada ($\omega_{m.est.IFO}$) pelo observador de fluxo com o circuito IPLL, apresenta *overshoots* ao final de cada transitório, o que é esperado em função da dinâmica da máquina. Em regime permanente, nas três velocidades, o erro entre ω_m^* e $\omega_{m.est.IFO}$ foi reduzido a zero, indicando desempenho satisfatório do controlador com o estimador apresentado.

Na Figura 27, tem-se o torque mecânico na carga no mesmo intervalo de tempo da Figura 26. Observa-se uma variação do torque durante o primeiro transitório de $19N.m$ para, aproximadamente, $57N.m$. No segundo transitório o torque foi reduzido para $24N.m$, aproximadamente, quando o motor opera em regime permanente. Pode ser observado que a forma de onda do torque obtido com este controlador foi similar ao apresentado no Capítulo 1, quando o controlador do MSIP fez uso do sensor de posição para determinar a velocidade.

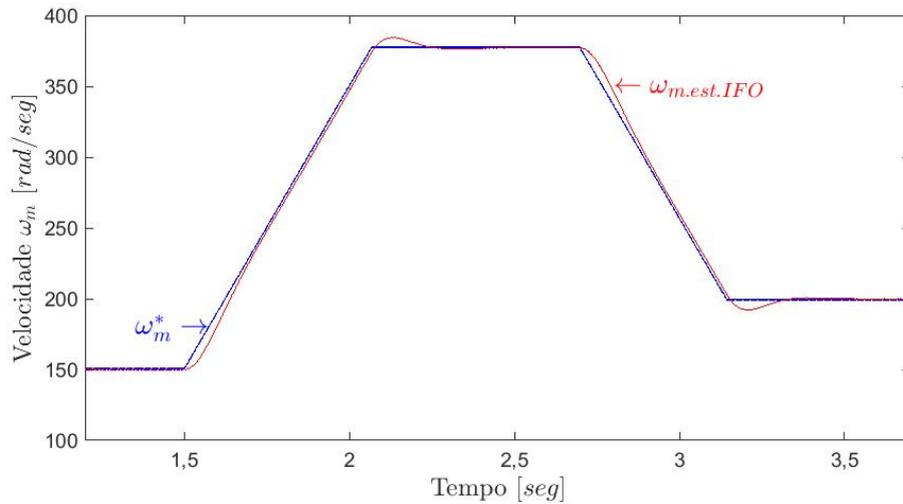


Figura 26: Comparação entre a velocidade estimada ($\omega_{m.est.IFO}$) e a de referência (ω_m^*).

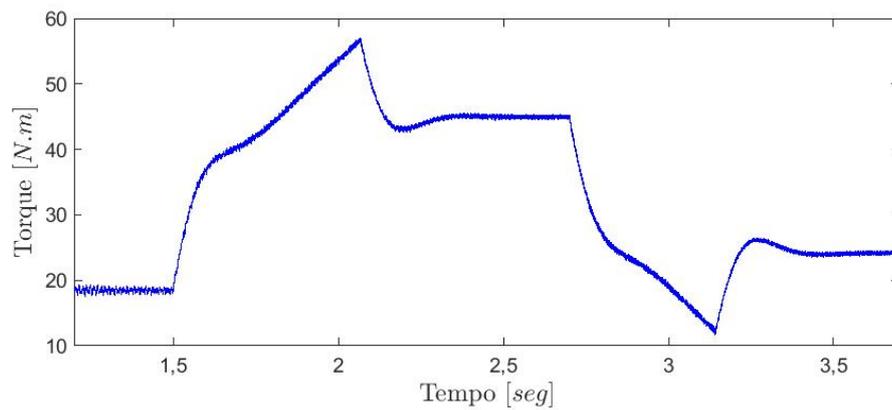


Figura 27: Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.

O estimador de velocidade com base no observador de fluxo com um circuito de sincronismo (IPLL) tem uma implementação relativamente simples, com bom desempenho para velocidades relativamente baixas. De fato, inicialmente a velocidade mecânica correspondia a 150rad/s o que equivale a 40% da velocidade nominal do motor. É importante comentar que o desempenho do controlador com esse estimador foi similar ao obtido quando do uso do sensor de posição para extrair a informação da velocidade mecânica. Tal fato evidencia o desempenho satisfatório deste estimador. A seguir vemos como realizar a estimativa da posição angular usando um estimador neural.

3 ESTIMADOR NEURAL

Neste capítulo são apresentados os principais conceitos no desenvolvimento de um estimador neural de velocidade mecânica para um MSIP. Para tal, inicialmente foram adquiridos sinais referentes às tensões e correntes no estator do motor a partir de simulações do circuito de potência.

Após da obtenção dos sinais correspondente, os dados são submetidos a um processo de condicionamento, também denominado pré-processamento, que consiste do tratamento, da seleção e da modificação da distribuição original dos dados. Assim foi realizada a subamostragem periódica uniforme e a normalização dos dados.

A seguir, são apresentados os conceitos fundamentais das redes neurais do tipo *Perceptron* Multicamadas, também conhecidas como MLP (*Multilayer Perceptron*). MLPs são empregadas neste trabalho visando realizar a estimação da velocidade do MSIP.

Por fim apresentou-se um exemplo que consiste no controlador de corrente baseado num estimador neural, para controle de corrente num conversor *Buck*. Este exemplo teve como finalidade avaliar o estudo do estimador neural proposto.

3.1 Aquisição de sinais e Geração da base de dados

Os recentes avanços tecnológicos na indústria permitiram tanto um aumento da quantidade de dados armazenados quanto a capacidade de transmitir e adquirir dados em maior quantidade com menor tempo. Basicamente, pode-se definir a aquisição de sinais como o registro de um fenômeno físico ou de sua transdução para o domínio elétrico e; a partir desses extraem-se informações visando modelar o comportamento dos fenômenos observados [32]. Particularmente, neste trabalho, foram adquiridos sinais referentes às correntes e tensões produzidas pelo conversor de potência que alimenta o MSIP e, dependendo da estratégia de controle utilizada, as informações adquiridas a partir da medição da velocidade mecânica do rotor do MSIP também foram utilizadas.

Em diferentes aplicações industriais é necessário empregar um sistema de aquisição de dados, o qual está conformado por sensores que são os responsáveis por medir as diferentes grandezas físicas. Uma vez adquiridos os sinais correspondentes, estes passam por outras etapas do processo como, por exemplo, condicionamento, armazenamento e processamento. Para o sistema proposto neste trabalho, a coleta dos dados foi realizada

nas simulações do circuito de potência, o qual foi implementado no ambiente de simulação PSIM. Na Figura 28 são ilustrados os sensores envolvidos nas diferentes simulações ao longo deste trabalho, tais como sensores de corrente, sensores de tensão e um sensor de posição atrelado ao eixo do motor. Após da obtenção dos sinais correspondentes, os dados são submetidos a um processo de condicionamento, também denominado pré-processamento, para o posterior processamento e atualização das tensões de referência que o conversor deve produzir. A seguir, explica-se o pré-processamento dos sinais adquiridos.

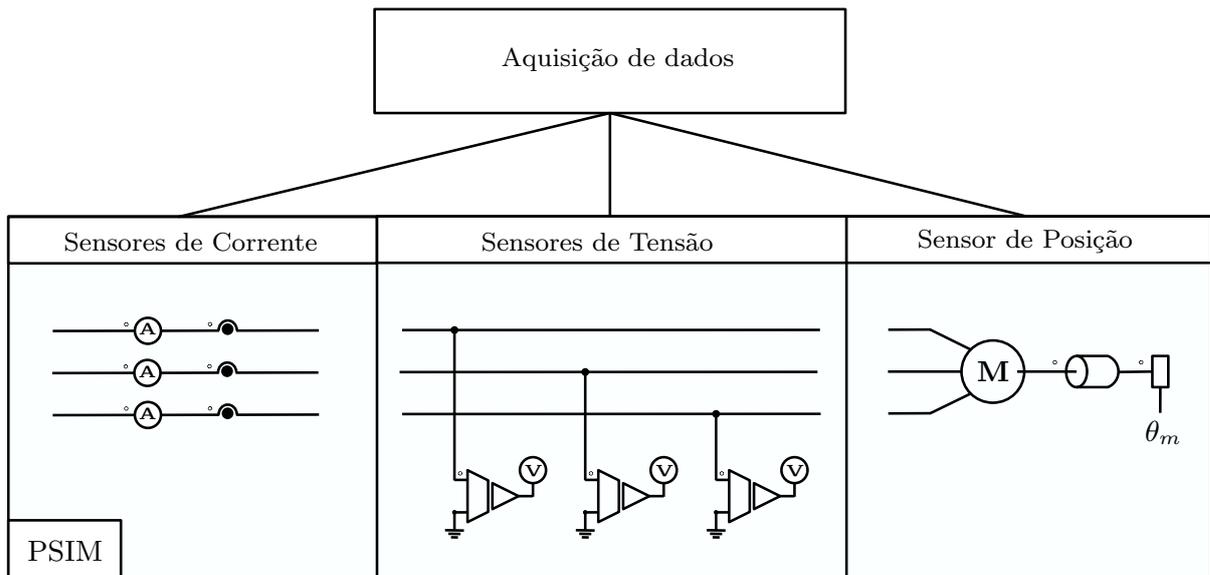


Figura 28: Sensores para aquisição de dados usados neste trabalho: corrente, tensão e posição.

3.2 Pré-processamento de dados

Essencialmente, neste trabalho, o pré-processamento consiste do tratamento, da seleção, da limpeza e da modificação da distribuição original dos dados, no objetivo que estes sejam utilizados adequadamente nas etapas posteriores do processamento. Entre os principais métodos aplicados neste trabalho podem ser mencionados a “amostragem”, onde realiza-se um aumento ou redução da quantidade de amostras e a normalização, que visa minimizar problemas causados pelos sinais adquiridos terem diferentes dispersões.

3.2.1 Amostragem da base de dados

A amostragem visa equalizar a quantidade de amostras de diferentes ‘categorias’, classes ou níveis no banco de dados. Para isso, pode-se realizar uma subamostragem, que

consiste em remover amostras das classes majoritárias da base de dados. Outro método corresponde à super-amostragem, que consiste no aumento da quantidade das amostras minoritárias da base de dados. É possível também fazer uma amostragem mista, a partir da combinação dos dois métodos, realizando a redução e o aumento das amostras das classes apropriadas [33].

A subamostragem mais simples consiste em reduzir a quantidade de entradas da base de dados em vez de alterar a distribuição de probabilidades. Na Figura 29 pode-se observar esta ideia. Há diferentes abordagens para fazer a subamostragem. Neste trabalho, empregou-se a abordagem periódica uniforme ilustrada na Figura 29. Para isso, retém-se uma amostra a cada 10 amostras da base de dados original.

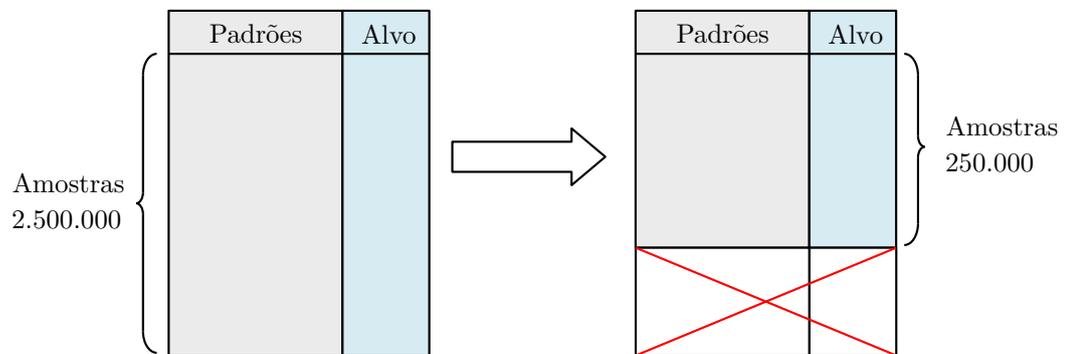
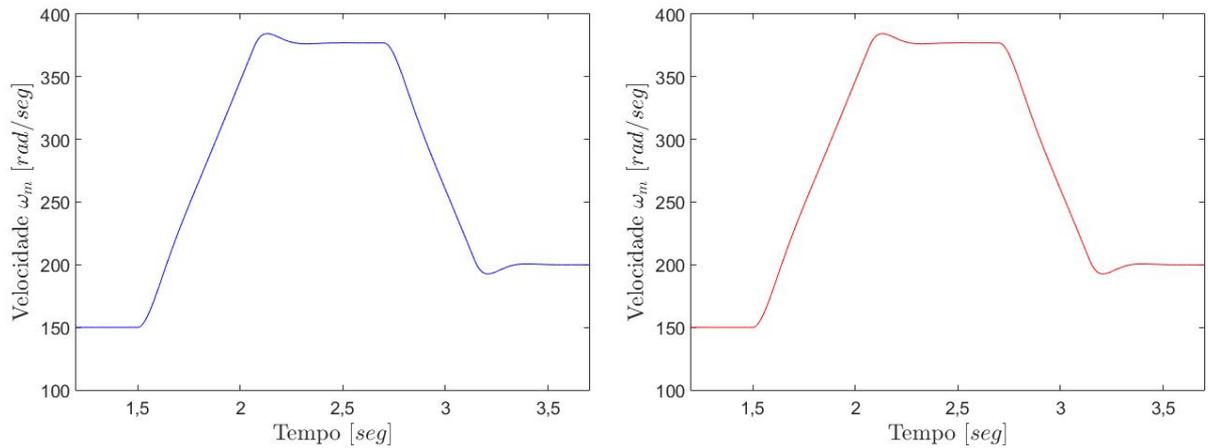


Figura 29: Subamostragem Periódica que mantém a proporção de amostras/entradas das diferentes classes.

Neste contexto, uma consideração importante ao definir o número mínimo de amostras consiste na necessidade de manter a natureza de cada um dos padrões. Na Figura 30a pode-se observar, a variável alvo da base de dados em função do tempo, “ ω_m ”, com 2.500.000 amostras. Em seguida, na Figura 30b é apresentada a sua distribuição subamostrada com 250.000 amostras.



(a) Forma de onda de $\omega_m(t)$ com a distribuição original. (b) Forma de onda de $\omega_m(t)$ sub-amostrada na fração 1/10.

Figura 30: Subamostragem uniforme da velocidade mecânica “ ω_m ”.

3.2.2 Normalização dos dados

De acordo com [33], a normalização de dados é uma transformação que consiste em modificar os valores dos atributos de modo a alterar sua faixa de valores (um mínimo e um máximo, normalmente 0 e 1) porém mantendo a forma da distribuição de probabilidades no novo intervalo. A normalização é particularmente útil para algoritmos baseados em redes neurais, pois traz imunidade a possíveis diferenças de energias médias de diferentes atributos. Existem diferentes métodos de normalização, dentre os mais usados podem ser mencionados:

- Normalização **min-max**: aplica-se uma transformação linear ao tributo, de tal forma que seus valores fiquem confinados aos intervalos ente -1 e 1, ou entre 0 e 1. Isso é obtido através de

$$A_{Nor} = \frac{A - A_{min}}{A_{max} - A_{min}}, \quad (23)$$

na qual, A_{min} e A_{max} são os valores mínimo e máximo do atributo A , e A_{Nor} é o valor do mesmo normalizado.

- Normalização **z-score**: faz-se um mapeamento dos valores do atributo em função da média e do desvio padrão de sua distribuição, com o objetivo de obter uma

distribuição com media zero e desvio-padrão unitário. Isso é conseguido via

$$A_{Nor} = \frac{A - \bar{A}}{\sigma_A}, \quad (24)$$

na qual, \bar{A} e σ_A são a média e o desvio padrão do atributo A , respectivamente. Este método é considerado muito útil quando trabalha-se com aproximadores de funções [34].

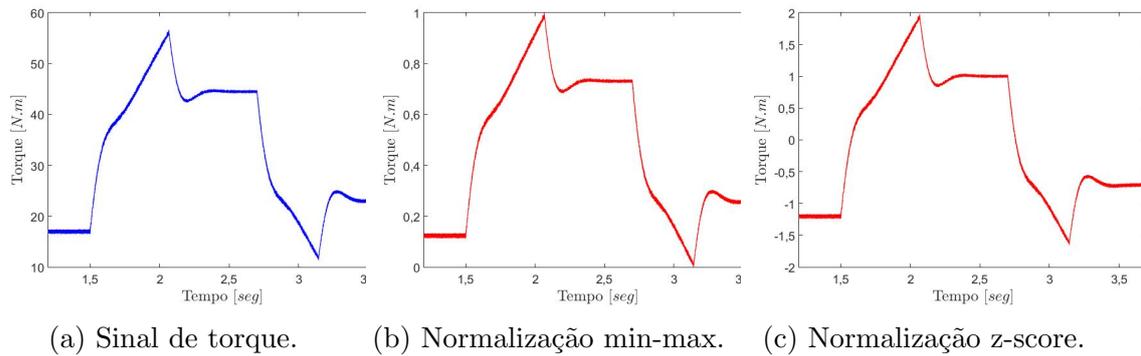


Figura 31: Forma de onda original do sinal de torque e ondas resultantes das normalizações min-max e z-score.

A Figura 31 ilustra o efeito dessas normalizações no sinal de torque. A Figura 31b mostra a forma de onda do torque após a normalização min-max e a Figura 31c após a normalização z-score.

3.3 Redes Neurais Artificiais

Nesta seção são apresentados os conceitos fundamentais das redes neurais do tipo *Perceptron* Multicamadas, também conhecidas como MLP (*Multilayer Perceptron*). MLPs são empregadas neste trabalho visando realizar a estimação da velocidade do MSIP.

A princípio, uma rede neural artificial baseia seu funcionamento na estrutura dos neurônios dos seres vivos [35, 36]. Desta forma, num processo iterativo, este tipo de modelo tem a capacidade de aprender a partir dos exemplos de treinamento, procurando reproduzir a nossa habilidade de generalizar a informação aprendida.

Em função da capacidade de generalização que uma determinada rede neural artificial atinge, esta tem a capacidade de fornecer uma resposta mais, ou menos, aproximada do alvo desejado. Essa aproximação permite inferir o alvo para dados originalmente desconhecidos. Assim, pode se definir uma rede neural artificial como um modelo matemático

não-linear que permite realizar relacionamentos complexos entre os dados de entrada e as saídas desejadas.

Pela sua grande aplicabilidade, as redes neurais são empregadas para solucionar diferentes problemas existentes nas áreas da engenharia, como classificação, agrupamento, previsão, aproximação de funções, entre outras.

Alguns aspectos relevantes para o desempenho da rede são a sua composição, estrutura e regras de aprendizado. Assim, com o objetivo de planejar adequadamente o que usar para uma aplicação alvo, além de dispor de uma base de dados adequada, devemos entender e escolher o modelo de neurônio, a arquitetura de rede e o algoritmo de treinamento.

3.3.1 Arquitetura da Rede Neural

Basicamente, as redes neurais utilizadas neste trabalho apresentam a arquitetura MLP. Tal arquitetura pode conter uma ou mais camadas intermediárias, também conhecidas como camadas escondidas, as quais são compostas basicamente de processadores simples (perceptrons). Na Figura 32 [35], tem-se um exemplo de topologia de redes MLP. Essa rede é formada por uma camada de entrada de n variáveis, duas camadas escondidas constituída de n_1 e n_2 neurônios respectivamente, e, por fim há uma camada neural de saída, composta de n_3 neurônios, sendo que o número de neurônios da camada de saída corresponde ao número de saídas da rede.

3.3.2 Neurônio artificial

Uma rede neural é conformada por um conjunto de unidades de processamento mais simples, com uma alta taxa de conectividade entre si, conferindo a capacidade intrínseca de operar paralelamente. Vejamos os principais elementos que conformam essa unidade de processamento, o Neurônio Artificial, conforme apresentado por [35].

O neurônio artificial é o principal componente do processamento da rede neural. A Figura 33 ilustra um neurônio artificial, ele é composto por sete elementos: *i*) o conjunto de sinais de entrada (x_1, x_2, \dots, x_n) , *ii*) o conjunto de pesos sinápticos (w_1, w_2, \dots, w_n) , *iii*) o combinador linear (\sum_1) , *iv*) o limiar de ativação (θ) , *v*) o potencial de ativação (u) , *vi*) a função de ativação $g(\cdot)$, e *vii*) o sinal de saída y .

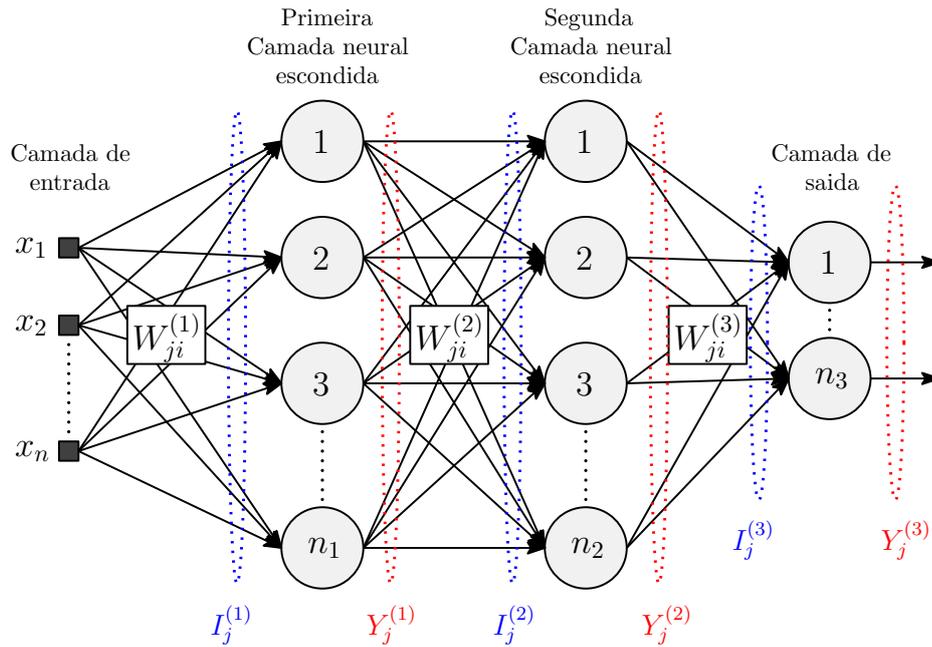


Figura 32: Arquitetura *Perceptron* de Múltiplas Camadas.

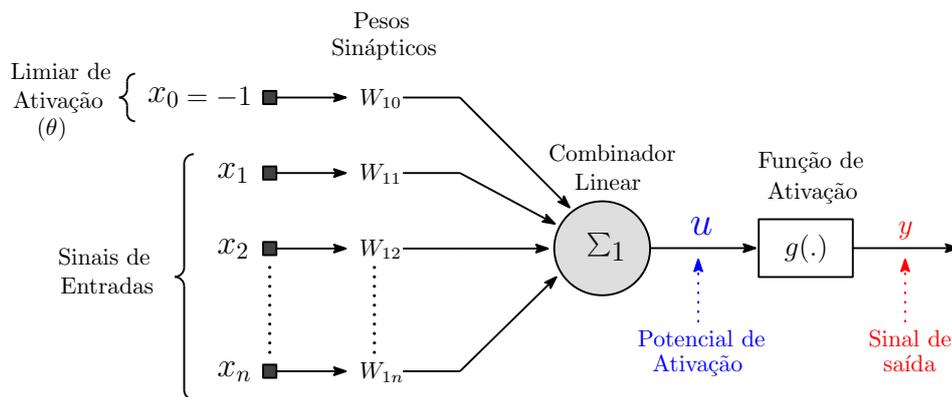


Figura 33: Composição de um neurônio artificial.

Essencialmente, o processamento realizado por essa unidade de processamento (o neurônio artificial) é: **1**) os valores de entrada, sinais provenientes do meio externo, normalmente caracterizados por um vetor do tipo (x_1, x_2, \dots, x_n) , são apresentados ao neurônio; **2**) cada uma dessas entradas é multiplicada por um peso sináptico pré-estabelecido (w_1, w_2, \dots, w_n) correspondente; **3**) os produtos (ponderações) são somados (Σ) a fim reproduzir o potencial de ativação dado por $(u = \sum w_n x_n - \theta)$; **4**) o potencial de ativação é limitado apropriadamente por uma função de ativação $g(\cdot)$ e, com isso, gera-se o sinal de saída $y = g(u)$.

3.3.3 Funções de ativação

As funções de ativação definem a saída do neurônio em termos de níveis de atividade. Entre as principais funções de ativação mais utilizadas, tem-se: a função linear, a função Gaussiana, a função logística, e a função tangente hiperbólica. A última é empregada no estimador neural neste trabalho. Na Figura 34 ilustra-se a função tangente hiperbólica, onde a saída assume valores reais entre -1 e 1, e a sua expressão matemática é dada pela relação

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}}, \quad (25)$$

na qual, β é o nível de inclinação da função em relação ao ponto de inflexão.

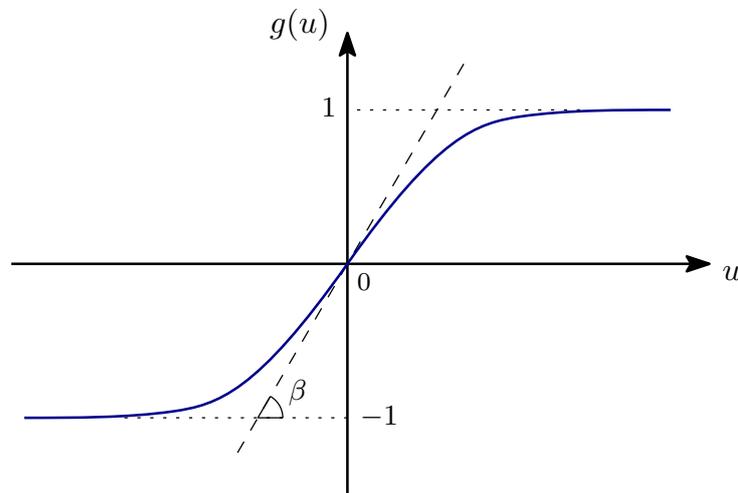


Figura 34: - Função de ativação - Tangente Hiperbólica.

3.3.4 Algoritmo de aprendizado

Define-se como algoritmo de aprendizado ao conjunto de regras para a solução do problema de aprendizado. Particularmente, o algoritmo de aprendizado empregado no presente trabalho é o algoritmo de retro-propagação do erro, também conhecido como “*Feed-forward and Back-propagation*”, muito empregado para o aprendizado por redes MLP. Este algoritmo é dividido em duas etapas bem definidas chamadas de propagação (*forward*), e retro-propagação (*backward*).

Na Figura 35 é ilustrada a arquitetura de rede MLP ajustada para o presente trabalho, composta por uma camada escondida e um neurônio na camada de saída. A partir desta configuração pode se descrever as duas etapas do algoritmo de aprendizado.

Na primeira etapa, a propagação (*forward*), os sinais de entrada são apresentados à rede neural e propagados, camada por camada, até chegar-se à camada de saída. Gera-se o sinal de saída correspondente, que é comparado com a saída ou sinal desejado. Este é um treinamento supervisionado pois compara-se a saída a uma referência ou alvo. Nesta estratégia é necessário dispor de uma tabela de entradas e saídas correspondentes. A partir dessas informações, comparando as saídas geradas às desejadas, a rede neural formula as hipóteses sobre o que deve ser aprendido.

O erro resultante, a comparação entre saída produzida pela rede e a saída desejada, é usado para a retro-propagação (*backward*). O erro resultante é empregado para ajustar os pesos das camadas intermediárias $w_{ji}^{(2)}$. Esse ajuste, a retro-propagação de erro, é aplicado sucessivamente por todas as camadas até chegar-se à primeira camada intermediária $w_{ji}^{(1)}$.

Com a finalidade de avaliar o desempenho, utilizou-se o RMSE (*RMSE - Root Mean Square Error*)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{p} \sum_{k=1}^p (d(k) - y(k))^2}, \quad (26)$$

sendo p o número de amostras do conjunto de alvo (treinamento ou teste), $d(k)$ o valor desejado e, por fim, $y(k)$ é a saída rede neural.

Para fim de interpretação, a Figura 36 mostra o comportamento da medida de generalização em função do número de épocas de treinamento. Essencialmente, cada época corresponde a uma iteração na qual a rede neural realiza um processo completo de treinamento.

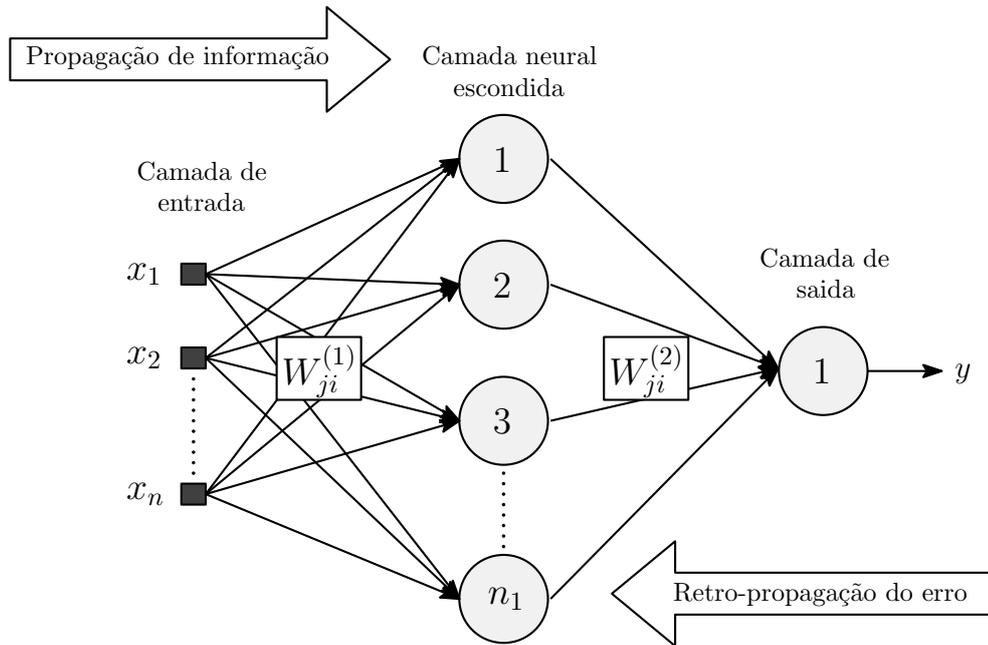


Figura 35: Algoritmo de aprendizado *Feed-forward and Back-propagation*.

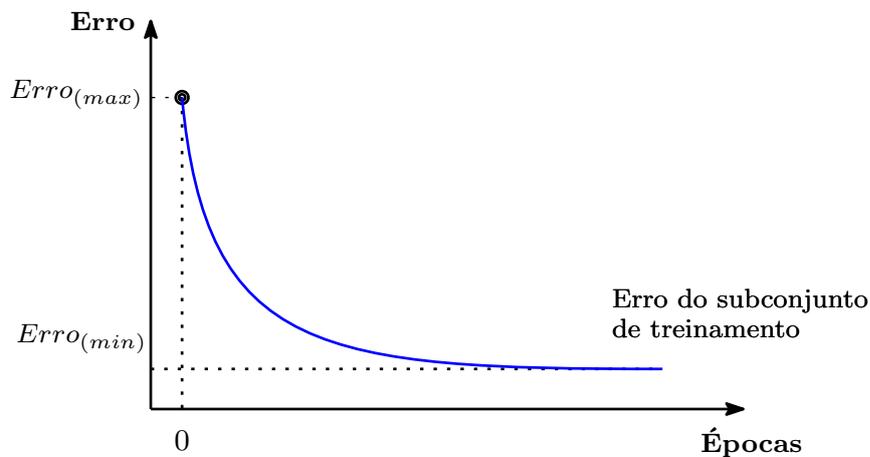


Figura 36: Comportamento do erro em função das épocas de treinamento.

3.3.5 Taxa de aprendizado “ η ” e Termo do *momentum* “ α ”

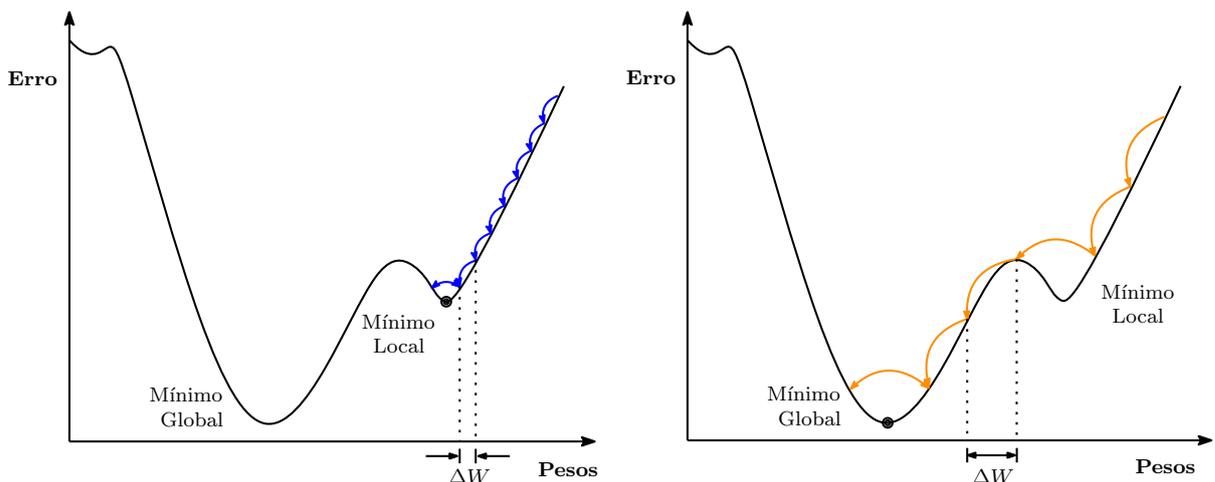
No processo de aprendizado, emprega-se a retro-propagação para realizar o ajuste dos pesos visando minimizar o erro entre a saída produzida pela rede e a de referência. Nesse processo, alguns parâmetros têm grande influência como, por exemplo, a taxa de aprendizado e o *momentum*.

A taxa de aprendizado “ η ” é um parâmetro que interfere na convergência do erro durante o aprendizado. É preciso estabelecer um compromisso. Por uma lado, “ η ” não

pode ser muito baixo porque isso poderia acarretar um treinamento muito lento e ainda levar a um mínimo local, conforme ilustrado na Figura 37a. Por outro lado, um valor grande de “ η ” produz maiores mudança nos pesos, o que aumenta a velocidade de aprendizado mas pode ocasionar oscilações no entorno do mínimo global, sem necessariamente permitir estabilizar os pesos no ponto de mínimo, a Figura 37b exemplifica esta condição. Em seguida apresenta-se a expressão matemática

$$W_{ji}^{(L)}(t+1) = W_{ji}^{(L)}(t) + \eta \delta_j^{(L)} y_i^{(L-1)}, \quad (27)$$

que descreve a atualização dos pesos na L -ésima camada escondida, levando em conta a retro-propagação do erro advinda a partir da camada de saída [35], na qual $\delta_j^{(L)}$ vem referido ao erro do j -ésimo neurônio da camada L .



(a) Convergência do erro durante o treinamento para η pequeno. (b) Convergência do erro durante o treinamento para η grande.

Figura 37: Erro da rede neural artificial em função dos pesos das sinapses. As curvas em cores exemplificam como os pesos são ajustados durante o treinamento considerando diferentes taxa de aprendizado “ η ”.

Conseqüentemente, para melhorar a assertividade na busca do ponto mínimo global, utiliza-se outro parâmetro: o termo de *momentum* α . Este parâmetro tem como objetivo acelerar a convergência dos pesos das sinapses da rede neural sem gerar oscilações no entorno do mínimo global e conseguir assim convergir para o ponto de mínimo global. Na Figura 38 apresenta-se o efeito do termo de *momentum* na aprendizagem da

rede. Deste modo, a atualização dos pesos sinápticos é modificada e determinada por

$$W_{ji}^{(L)}(t+1) = W_{ji}^{(L)}(t) + \alpha (W_{ji}^{(L)}(t) - W_{ji}^{(L)}(t-1)) + \eta \delta_j^{(L)} y_i^{(L-1)}, \quad (28)$$

onde basta inserir um único parâmetro α , tentando ponderar o quão os pesos sinápticos foram alterados entre duas iterações, no instante (t) e $(t-1)$ [35].

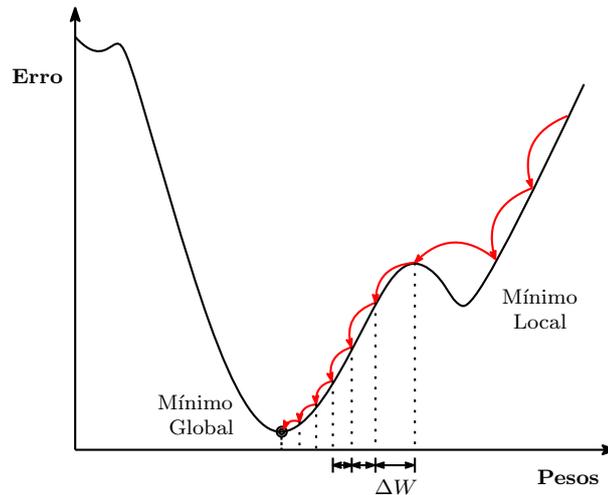


Figura 38: Aprendizado com termo do momento.

Usualmente, os valores desses parâmetros devem ser $0,05 \leq \eta \leq 0,75$ e $0 \leq \alpha \leq 0,9$ para o treinamento de uma rede neural [35]. Obviamente, eles devem ser ajustados no processo de aprendizado.

3.3.6 Treinamento, Validação e Teste de uma Rede Neural Artificial

O conceito de aprendizado [35, 36] implica que uma rede neural não somente deve estimar respostas corretas no processo de treinamento, como também fornecer respostas aproximadas e coerentes quando dados novos lhe são apresentados, isto é, deve ser capaz de generalizar. No objetivo de assegurar isso, divide-se da base de dados em 3 grupos: conjunto de treinamento, conjunto de validação, e conjunto de teste. Estes grupos normalmente apresentam proporções de 60%, 20% e 20% do total de amostras utilizadas, conforme ilustrado na Figura 39.

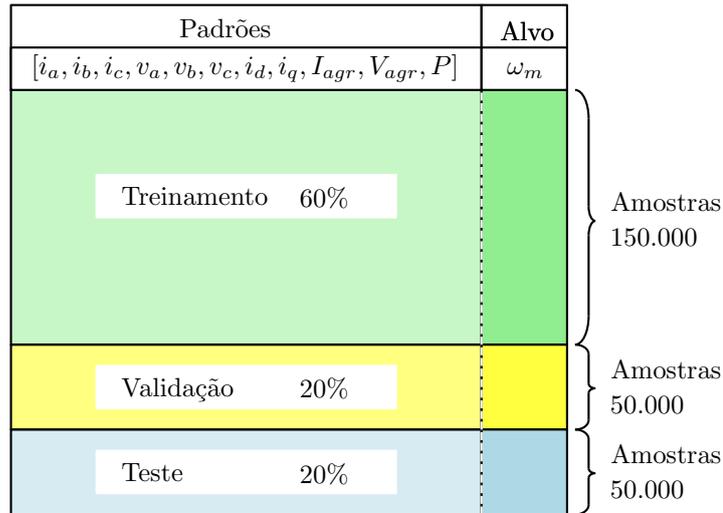


Figura 39: Distribuição dos conjuntos de aprendizado.

O objetivo do treinamento é calcular os pesos/ponderações da rede, o que obtido a partir de saídas conhecidas para os padrões de entrada empregados durante o treinamento (supervisionado). Obviamente, pretende-se que com isso a rede atue de forma satisfatória ao extrapolar para entradas inicialmente não testadas. Quanto maior o conjunto de dados de treinamento, a princípio, melhor. Porém, empregar um grande número de amostras pode resultar em um super-treinamento fazendo com que a rede perca a sua capacidade de generalização.

Assim, após cada ajuste dos pesos numa época, faz-se a validação da rede, com outra fração dos dados disponíveis. Com este grupo de amostras é observado como evolui o erro da rede para padrões de entrada desconhecidos, isto é, não empregados no treinamento, visando evitar assim o super-treinamento. Buscando obter melhor generalização pela rede neural, a validação visa identificar o melhor ponto de parada, técnica conhecida como parada antecipada (*Early Stopping*). Este método consiste em acompanhar a evolução do aprendizado nas curvas correspondentes aos erros da rede para os conjuntos de treinamento e validação, conforme observado na Figura 40. O treinamento é interrompido quando o gradiente do erro para o conjunto de validação muda de sentido, o que, na Figura 40, corresponde ao ponto de mínimo do erro para o subconjunto de validação.

Por fim, testou-se a rede neural artificial com o conjunto de teste - o restante das amostras, aquelas não utilizadas nem treinamento, nem na validação, com o objetivo de verificar o desempenho da rede. Com isso, verificou-se a capacidade de generalização da rede.

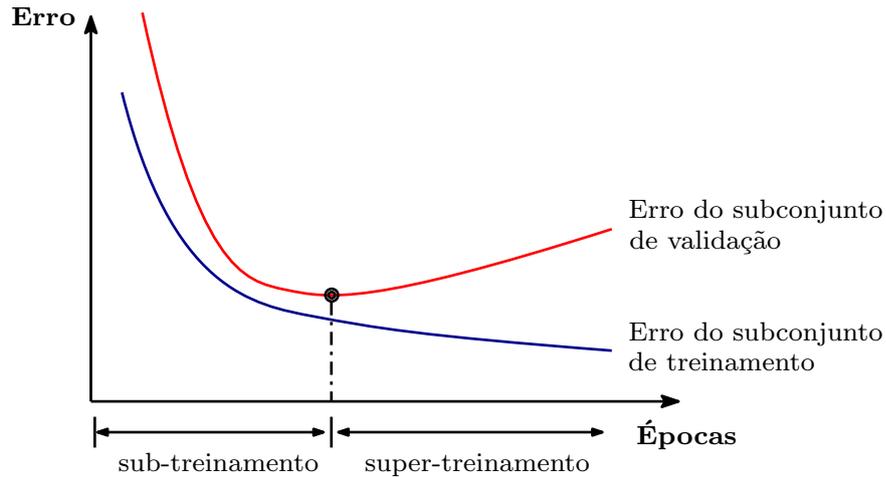


Figura 40: Técnica de parada antecipada.

Na sequência, apresentou-se, a título de exemplo, o controlador de corrente baseado num estimador neural para um conversor CC-CC abaixador de tensão (conversor *buck*).

3.4 Estimador de Corrente via Rede Neural Artificial

Nesta seção, empregou-se uma rede neural artificial para estimar a corrente no indutor do conversor *buck*. Estas redes foram utilizadas nesta dissertação para implementar sistema de acionamento de motor síncrono de ímã permanente sem sensor de posição.

Iniciamos com os resultados obtidos com o conversor operando com a realimentação da corrente I_L , conforme ilustrado na Figura 41. Ainda com relação à este exemplo, a corrente de referência é representada por I_L^* . Nesta configuração, quando o IGBT conduz, a corrente I_L apresenta derivada positiva e, ao contrário, quando o IGBT está aberto, a derivada de I_L é negativa. Neste sentido, um controlador do tipo proporcional-integral foi utilizado para ajustar o ciclo de trabalho em que o conversor deve operar, o qual é dinamicamente modificado enquanto o valor médio da diferença entre I_L^* e I_L for diferente de zero. Os sinais V_0 e d serão as entradas da rede neural artificial que obteremos, enquanto a corrente I_L é o alvo, a estimativa que a rede deverá produzir .

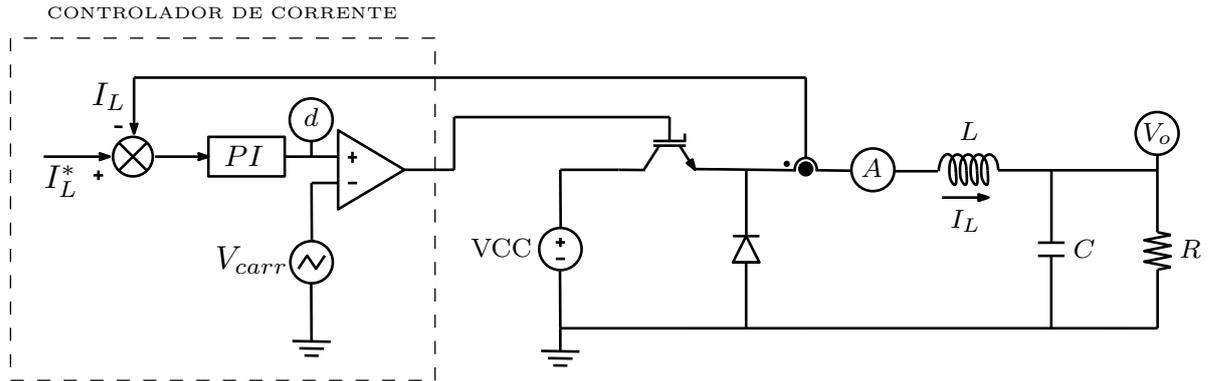


Figura 41: Esquemático do conversor *Buck*.

3.4.1 Aquisição de sinais e Geração da base de dados

Neste exemplo, a coleta dos dados foi realizada a partir de três parâmetros diferentes. O primeiro corresponde aos sinais adquiridos pelo sensor que mede a tensão na carga, V_0 . O segundo é o ciclo de trabalho d , que, neste exemplo, é comparado com a portadora triangular para produzir os sinais que comandam o IGBT. Por fim, o terceiro, consiste nos sinais adquiridos pelo sensor de corrente conectado em série com o indutor.

A princípio, a corrente de referência I_L^* varia em rampa crescente entre 0.5 amp e 9.5 amp durante 1 seg . Como consequência, o ciclo de trabalho, d , apresenta variação linear de $0.05V$ até $0.95V$. A tensão da carga, V_0 , varia de $5V$ a $95V$ e, a corrente I_L , varia de 0.5 amp até 9.5 amp . Coletou-se $1.000.000$ de amostras, posteriormente condicionadas. Na Figura 42a são ilustradas os níveis das correntes de referência I_L^* e a corrente desejada I_L e na Figura 42b o total de amostras coletadas. Os sinais V_0 e d são as entradas da rede neural artificial, enquanto a corrente I_L é o alvo, a estimativa que a rede deverá produzir.

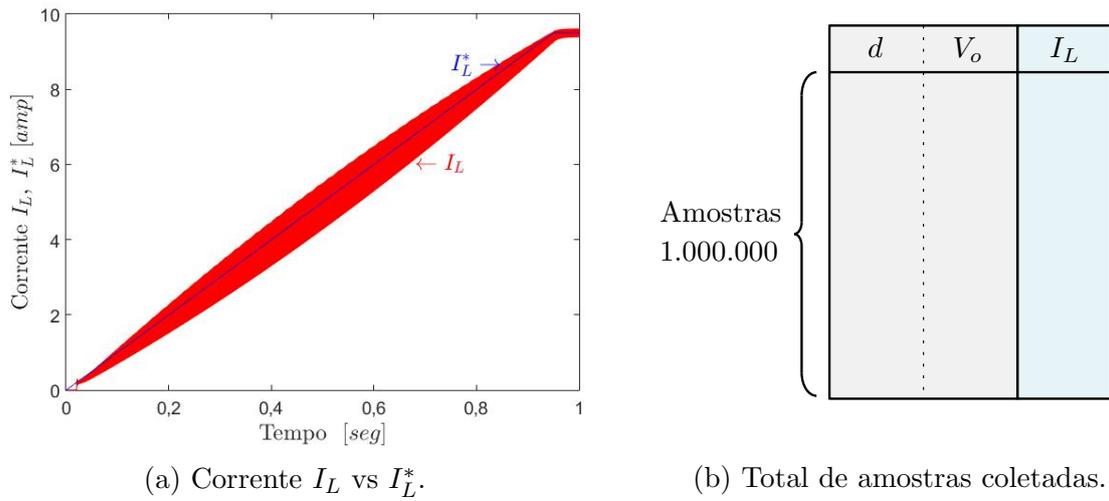


Figura 42: Forma de onda da corrente desejada “ I_L ” e quantidade de amostras coletadas.

3.4.2 Pré-processamento de dados

Após a obtenção dos sinais, a subamostragem e a normalização foram aplicadas, sequencialmente. Na Figura 43 apresenta-se o efeito da subamostragem uniforme, onde a base de dados resultante foi composta por 100.000 amostras.

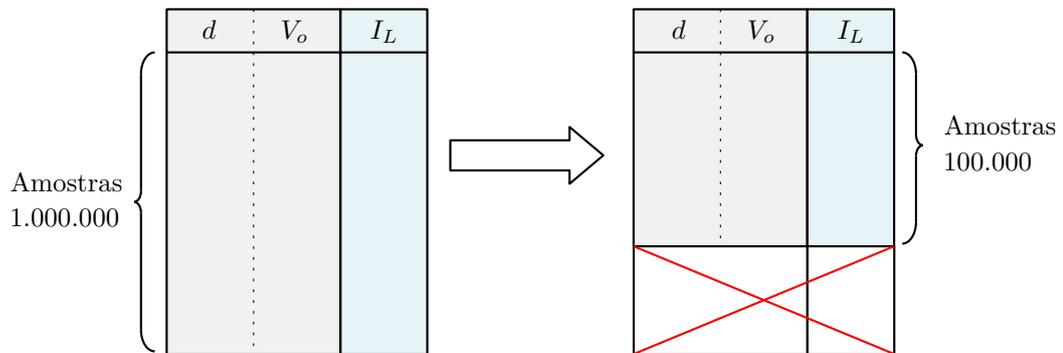
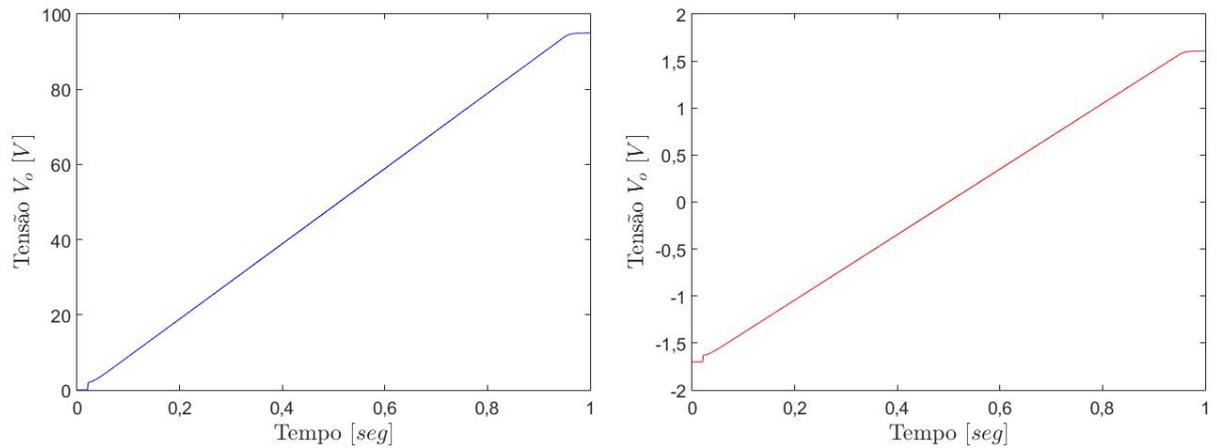


Figura 43: Subamostragem periódica dos sinais adquiridos no conversor *buck*.

Em seguida, a normalização z-score foi aplicada em cada atributo da base de dados. Como exemplo, considerou-se a forma de onda da tensão na carga, V_0 , mostrada na Figura 44.



(a) Forma da onda original, isto é, com a distribuição original. (b) Forma de onda após normalização da distribuição.

Figura 44: Forma de onda do “ V_o ” original e com a normalização z -score.

3.4.3 Características da Rede Neural empregada

Diferentes possibilidades foram testadas para o treinamento da rede neural. Na Tabela 3 são apresentados as duas melhores configurações. Pode-se observar que a configuração I apresenta a melhor medida de generalização RMSE (*Root Mean Squared Error*). Este melhor resultado está destacado em negrito na Tabela 3, razão esta para o uso da configuração I para estimar a corrente do indutor.

Tabela 3: Descrição dos parâmetros da rede neural.

Descrição dos parâmetros	Configuração I	Configuração II
Arquitetura da rede	Perceptron multicamadas	
Processo de treinamento	Supervisionado	
Função de ativação da camada de escondida	Tangente hiperbólica	
Função de ativação da camada de saída	Tangente hiperbólica	
Número de atributos	2	
Número de neurônios na camada escondida	3	3
Taxa de aprendizado	0,5	0,1
Termo do <i>momentum</i>	0,8	0,5
Numero de épocas	5000	5000
Medida de generalização (RMSE teste)	0,2648	0,2659

3.4.4 Resultados da Generalização do modelo

Após o treinamento, implementa-se o conversor *buck*, com uso da corrente estimada pela rede neural I_{LRN} , enquanto o restante da malha de controle permanece inalterado, conforme ilustrado na Figura 45. O “estimador neural” utiliza como entradas a tensão da carga, V_0 e o ciclo de trabalho, d , para estimar a corrente na carga, representada por I_{LRN} .

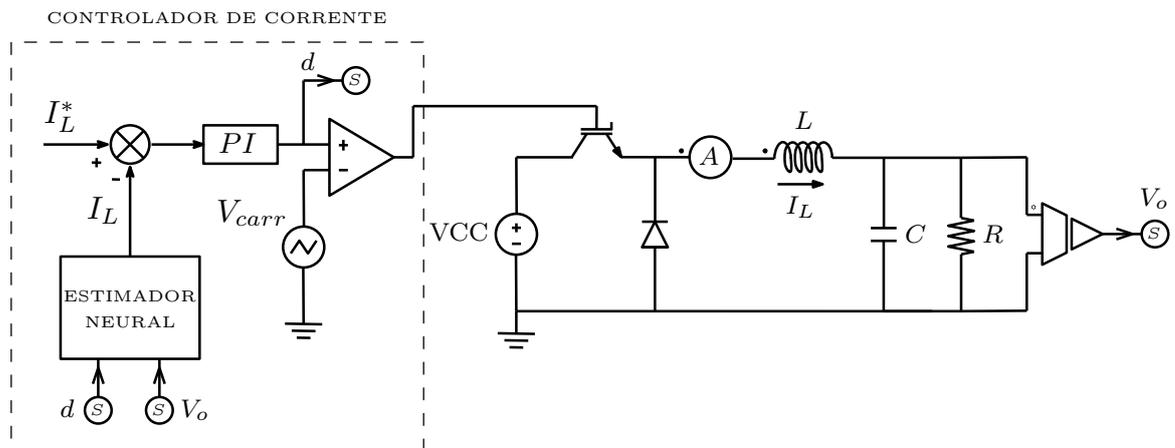


Figura 45: Esquemático do conversor *Buck*.

Na Figura 46 e na Figura 47 pode-se comparar as correntes I_L , I_{LRN} e I_L^* . Na primeira, a corrente de referência foi constante e igual à 4,5 amp, e na segunda ela foi de 7,8 amp. De acordo com os resultados obtidos, observou-se que a corrente estimada, I_{LRN} , correspondeu à componente média da corrente no indutor, I_L .

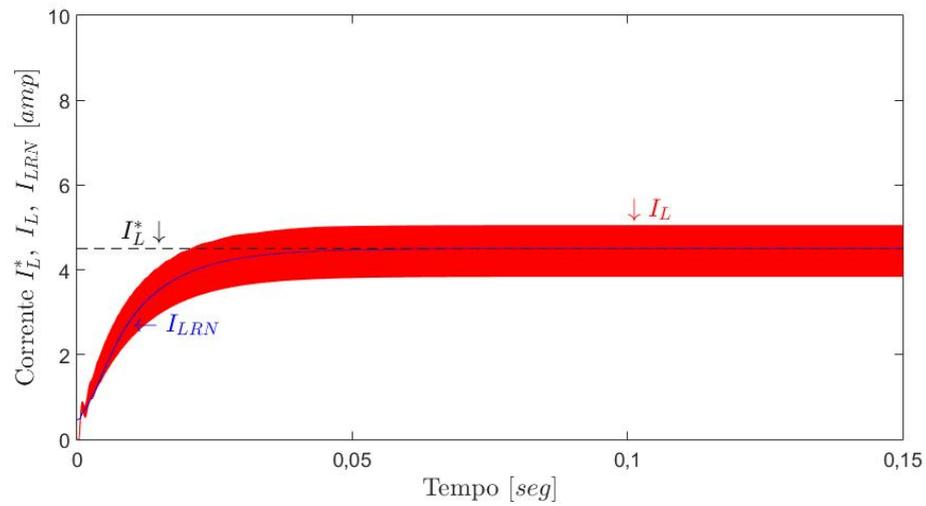


Figura 46: Comparação entre as correntes I_L , I_{LRN} e $I_L^* = 4,5 \text{ amp}$.

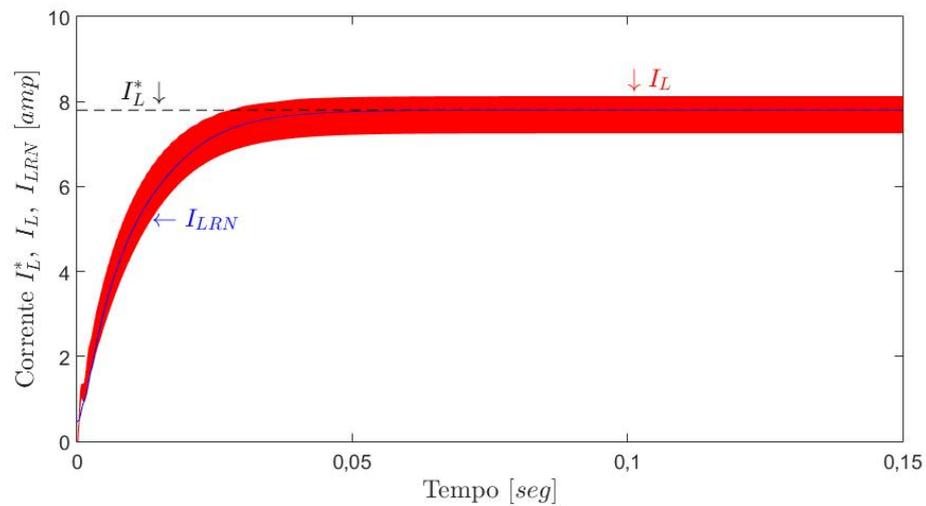


Figura 47: Comparação entre as correntes I_L , I_{LRN} e $I_L^* = 7,8 \text{ amp}$.

Este capítulo apresentou os princípios para a concepção de uma rede neural para uso como estimador de uma variável, tanto a topologia, aspectos de treinamento relevantes com o desempenho foram vistos. A partir deste estudo, no capítulo a seguir, foram apresentados e desenvolvidos controladores de velocidade sem sensor de posição, onde a rede neural foi utilizada em duas situações diferentes.

4 ACIONAMENTO DO MSIP SEM SENSOR DE POSIÇÃO COM USO DE UM ESTIMADOR NEURAL DE VELOCIDADE

Neste Capítulo é apresentado o acionamento sem sensores de um MSIP. Para tal, a velocidade, (ω_m) , foi estimada a por meio de redes neurais artificiais a partir de duas bases de dados diferentes. O primeiro estimador neural de velocidade, $(\omega_{m.est.RN})$, foi obtido a partir da base de dados que corresponde aos sinais adquiridos pelo sensor mecânico utilizado na simulação no Capítulo 1. O segundo estimador neural de velocidade, $(\omega_{m.est.RN.IFO})$, está correlacionado à base aos dados adquiridos a partir da velocidade estimada pelo IPLL na simulação do Capítulo 2.

O esquemático simplificado é apresentado na Figura 48. Neste esquemático, em ambos os casos, a velocidade foi estimada geralmente a partir das tensões e correntes na alimentação do motor: v_a, v_b, v_c e i_a, i_b, i_c , respectivamente. O ângulo de posicionamento usado no primeiro estimador neural foi o estimado pelo observador de fluxo, representado por $\theta_{m.obs}$.

Com o objetivo de simular o circuito baseado no estimador neural de velocidade, foi proposta uma metodologia de simulação com uso combinado dos programas PSIM e MATLAB. No primeiro programa foi simulado o circuito de potência e no segundo programa a generalização do melhor modelo para estimação da velocidade.

Por último, este Capítulo contém uma apresentação dos resultados obtidos com as quatro abordagens com descrição dos resultados e análise qualitativa dos mesmos.

4.1 Metodologia de simulação sem sensores

Para a abordagem da metodologia de simulação foram utilizados os programas PSIM e Matlab/Simulink de forma combinada. Na Figura 49 é ilustrado um desenho do diagrama de blocos da implementação proposta do controlador, que é baseado em um modelo de estimativa usando redes neurais.

Para esse fim, inicialmente foram extraídos os dados do circuito de potência simulado no PSIM. Estes sinais consistem nas tensões e correntes de armadura e na velocidade do motor. Lembrando que para a primeira base de dados a informação da velocidade foi extraída do sensor de posição, utilizado na primeira abordagem, para a segunda base de dados a informação da velocidade foi estimada pelo IPLL, utilizada na segunda aborda-

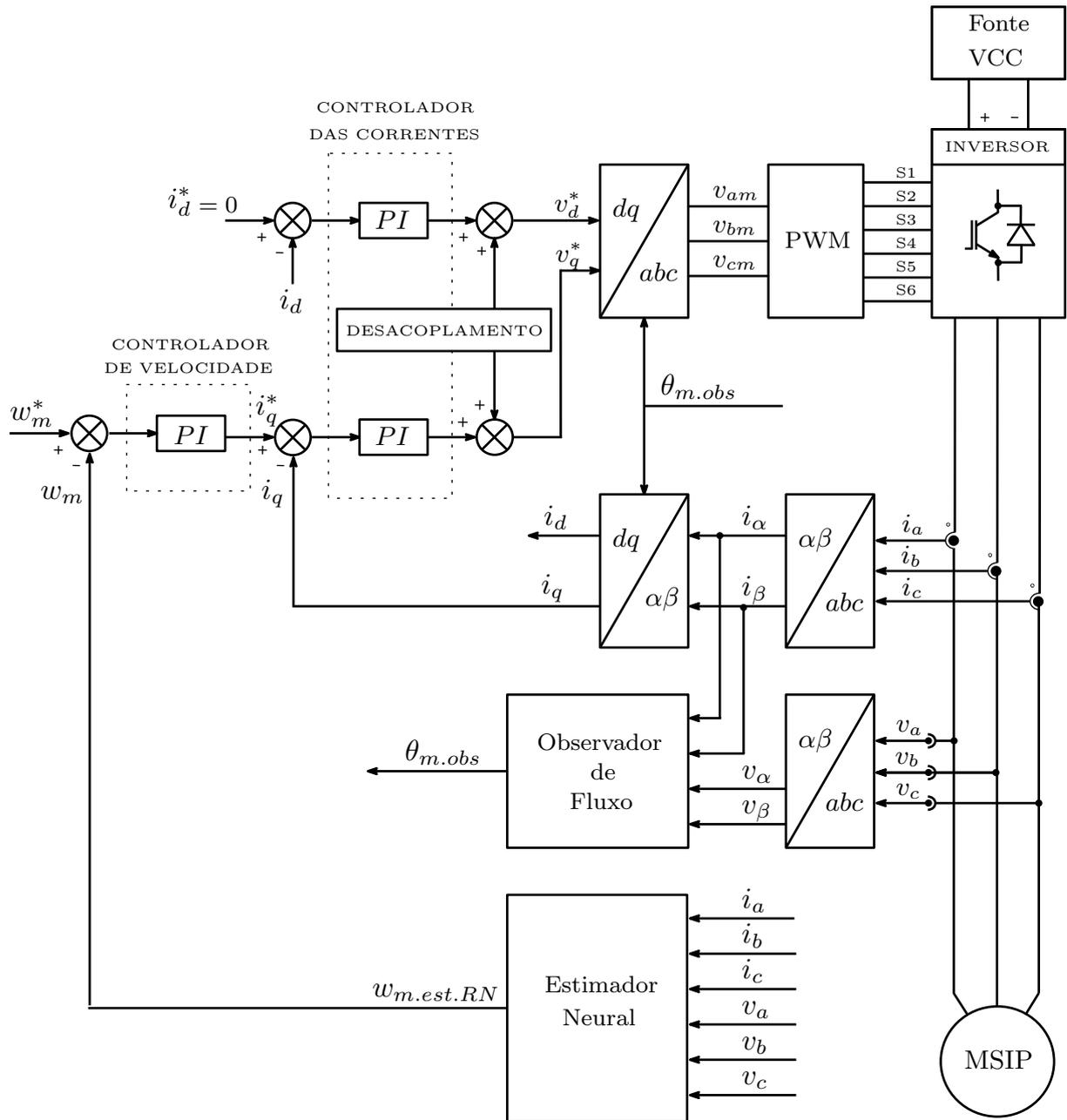


Figura 48: - Esquemático simplificado do circuito com o controle sem sensores - Estimador Neural.

gem.

Na sequência, utilizou-se o MATLAB para fazer o pré-processamento do conjunto de dados extraídos da simulação do circuito de potência no PSIM. Em seguida, foi concebido o melhor modelo para estimar a velocidade mecânica do motor " $w_{m.est.RN}$ ". Uma vez concebido o modelo do estimador, este foi testado com um novo conjunto de amostras que não pertenciam à base de dados, isto no objetivo de obter o estimador com o melhor desempenho de generalização do modelo.

Finalmente, o modelo generalizado para estimar a velocidade do motor baseado em redes neurais, “ $\omega_{m.est.RN}$ ”, foi simulado conforme o diagrama de blocos ilustrado na Figura 50. Nesta simulação há uma comunicação entre o MATLAB e o PSIM. Basicamente, neste arranjo, os resultados produzidos pelo circuito implantado no PSIM são transmitidos e processados no MATLAB. Em seguida, ocorre a transmissão da informação da velocidade estimada no MATLAB para o PSIM, onde a simulação é atualizada. É importante indicar que durante o período em que ocorrem as transmissões das informações e processamento no MATLAB, a simulação no PSIM não é atualizada. Para maior informação sobre o desenvolvimento da metodologia pode se referir ao Anexo A.

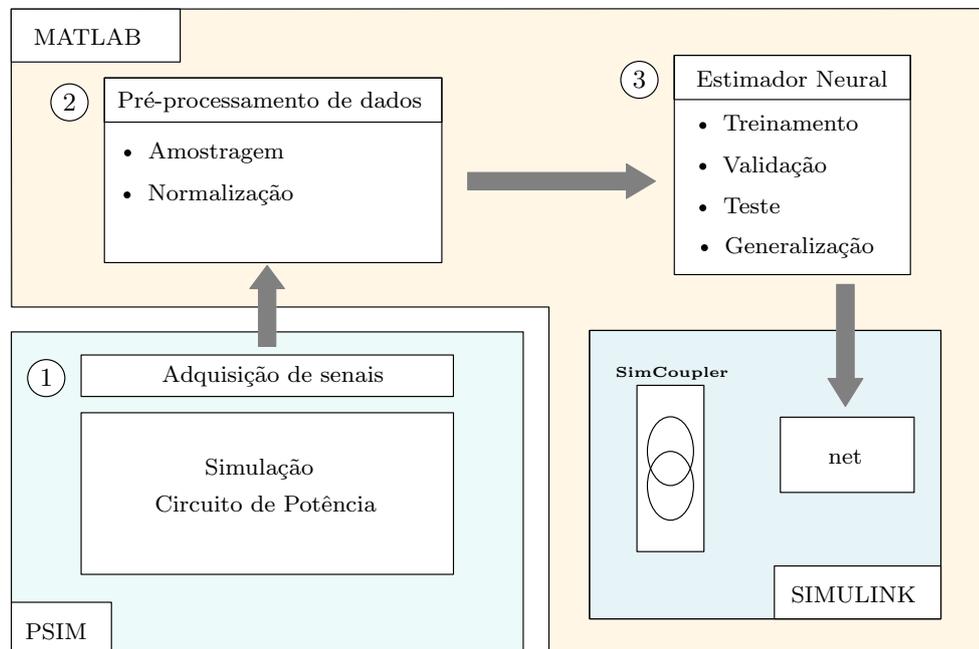


Figura 49: - Diagrama de blocos da implementação do controlador.

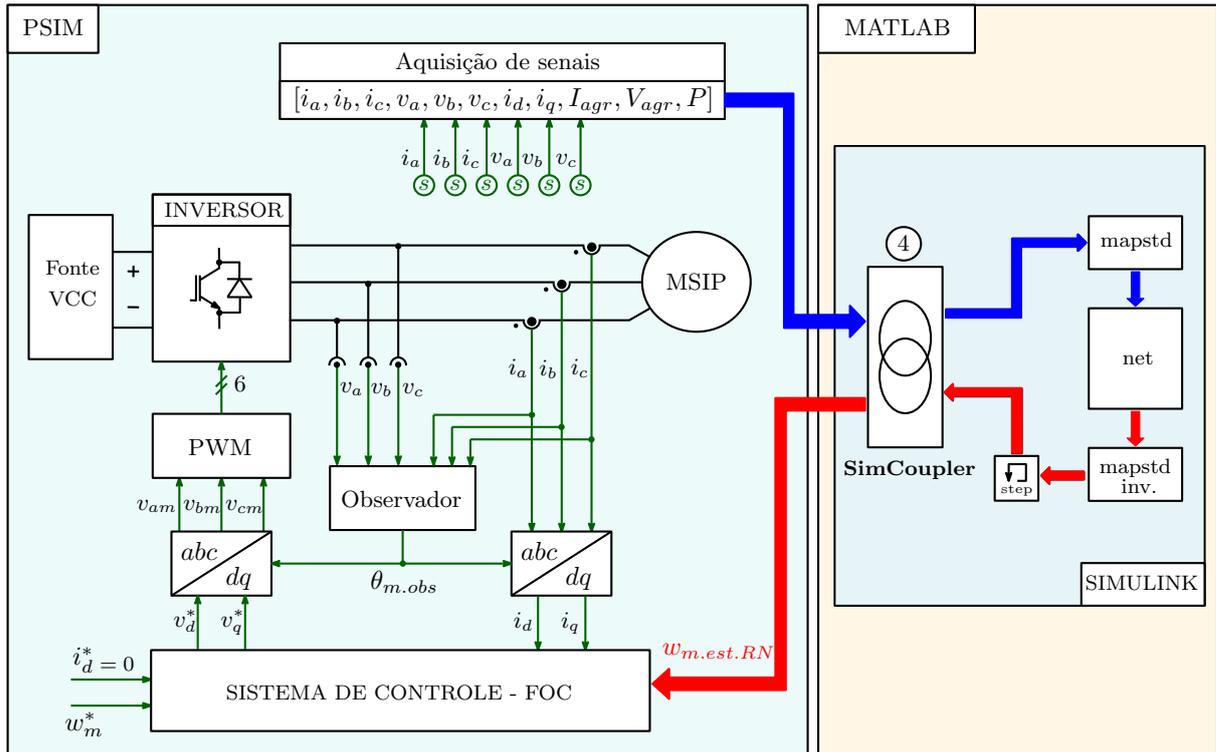


Figura 50: - Diagrama de blocos da operação do controlador.

4.2 Acionamento do MSIP sem sensores de posição com uso de um estimador neural de velocidade baseado nos dados provenientes do controle com sensores mecânicos

Esta seção apresenta como o estimador neural proposto foi obtido a partir dos sinais extraídos do sensor de posição conforme a simulação feita no Capítulo 1. Nesta simulação, a velocidade do motor ω_m foi adquirida por um sensor de posicionamento instalado no eixo do motor. O código fonte pode ser obtido no Anexo B.

4.2.1 Aquisição de sinais e geração da base de dados

A coleta de dados foi realizada considerando 12 parâmetros, dos quais, 11 foram definidos como atributos, e um como o alvo de estimação desejado, conforme descritos na Tabela 4. Os 11 atributos correspondem as correntes e tensões do estator representados por i_a, i_b, i_c , e v_a, v_b e v_c , as correntes do estator convertidas no sistema de coordenadas

“ $d - q$ ”, i_d e i_q , os valores da corrente agregada,

$$I_{agr} = \sqrt{0,6666 * (i_a^2 + i_b^2 + i_c^2)}, \quad (29)$$

e da tensão agregada,

$$V_{agr} = \sqrt{0,6666 * (v_a^2 + v_b^2 + v_c^2)}, \quad (30)$$

e a potência ativa consumida pelo MSIP, (P). O alvo desejado corresponde à velocidade (ω_m). Para fins de esclarecimento, para os cálculos dos parâmetros tanto de I_{agr} quanto de V_{agr} foram utilizados os valores instantâneos das correntes e tensões, respectivamente. Estes parâmetros representam, dinamicamente, os valores das amplitudes das correntes e tensões no estator da máquina.

Na etapa de pre-processamento de dados, de forma semelhante que no exemplo do capítulo 3, foram aplicados os processos de subamostragem uniforme e a normalização z-core. Assim, na Tabela 4 estão descritos os 11 atributos e a saída desejada. Cada atributo está conformado por um total de 2.500.000 amostras, as quais após da subamostragem resultam em uma base de dados composta por 250.000 amostras.

Tabela 4: Descrição dos padrões e alvo da base de dados.

Definição de Atributos	Simbologia
Correntes de fase	i_a, i_b, i_c
Tensões de fase	v_a, v_b, v_c
Correntes dos eixos direito e quadratura	i_d, i_q
Correntes agregada	I_{agr}
Tensão agregada	V_{agr}
Potencia ativa	P
Definição do Alvo	Simbologia
Velocidade mecânica	ω_m

4.2.2 Características da Rede Neural empregada

O processo de aprendizado da rede neural foi realizado a partir de diferentes testes, onde as variações nos seus parâmetros foram realizadas com o fim de encontrar a melhor

configuração do modelo para a etapa de generalização. Assim, a Tabela 5 apresenta o resumo das três melhores configurações, após de serem testadas aproximadamente um total de 168 configurações diferentes da rede neural. Também é importante destacar que, não necessariamente um maior número de neurônios apresenta a melhor generalização. Neste caso, a melhor estimativa da rede neural foi obtida com 5 neurônios na camada escondida, com a principal vantagem de que isso diminuiu a complexidade da rede na hora do processamento.

A Figura 51 mostra a arquitetura da rede neural que está implementada a partir da melhor configuração, que na Tabela 5 apresenta a medida de generalização mas próxima de zero, $RMSE = 0,0567$. Na Figura 52, verifica-se a capacidade de generalização da rede, onde o desempenho é aproximado nos 3 conjuntos que envolvem o processo de aprendizado.

Tabela 5: Descrição dos parâmetros da rede neural.

Descrição dos parâmetros	Config.I	Config.II	Config.III
Arquitetura da rede	Perceptron multicamadas		
Processo de treinamento	Supervisionado		
Função de ativação da camada de escondida	Tangente hiperbólica		
Função de ativação da camada de saída	Tangente hiperbólica		
Número de atributos	11		
Número de neurônios na camada escondida	5	8	10
Taxa de aprendizado	0,3	0,5	0,7
Termo do <i>momentum</i>	0,9	0,9	0,9
Número máximo de iterações (épocas)	5000	5000	5000
Medida de generalização (RMSE teste)	0,0567	0,0596	0,0597

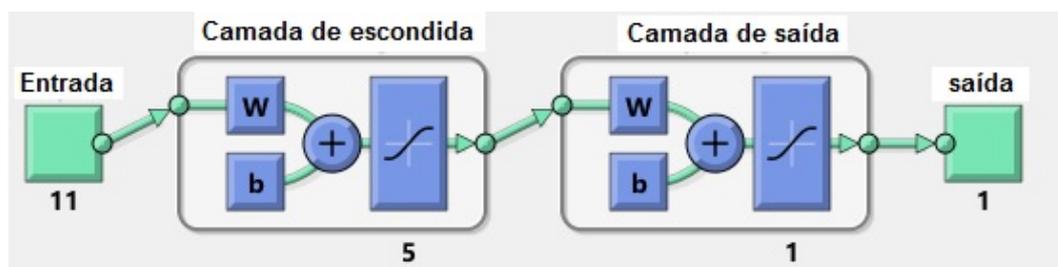


Figura 51: - Arquitetura da rede neural - .

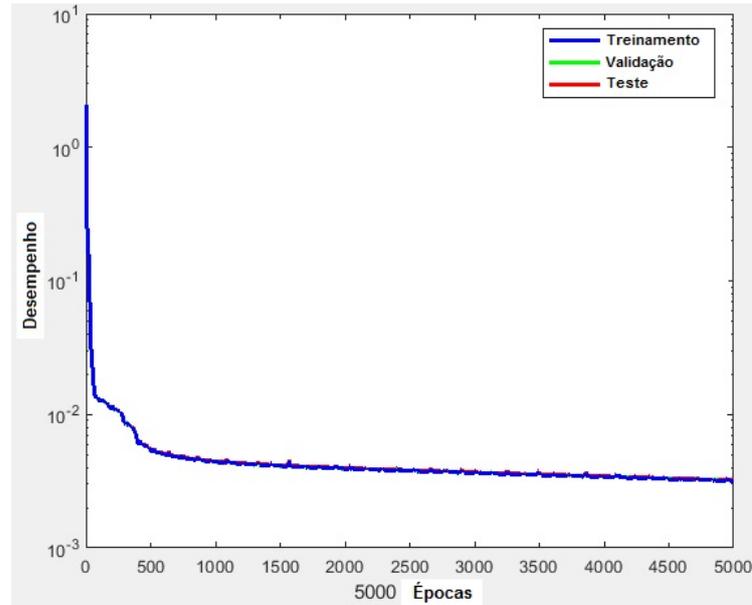


Figura 52: - Desempenho da rede neural empregada.

4.2.3 Resultados da Generalização do modelo

Em seguida, apresenta-se os resultados da estimativa da velocidade com a implementação do estimador neural. Para isso foi considerado o ângulo de posicionamento da Figura 53 nesta simulação, o qual foi calculado no capítulo 2 a partir do observador convencional baseado nos fluxos $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$.

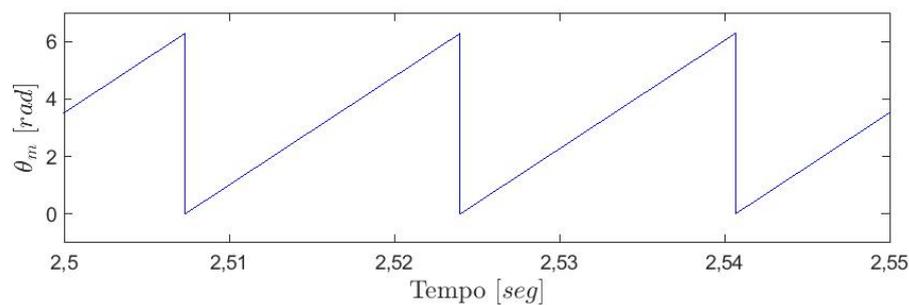


Figura 53: - Ângulo de posicionamento do rotor θ_m obtido do observador convencional, a partir de $\hat{\lambda}_\alpha$ e $\hat{\lambda}_\beta$.

Na Figura 54 apresenta-se uma comparação entre a velocidade estimada a partir do estimador neural ($\omega_{m.est.RN}$) e, a velocidade de referência (ω_m^*). Este teste foi similar aos feitos tanto no Capítulo 1 quanto no Capítulo 2, considerando a variação gradativa da velocidade de referência, (ω_m^*), inicialmente igual a $150rad/s$ e, em seguida, foi

gradualmente incrementada até a velocidade nominal de 377rad/s . Após de alcançar a velocidade nominal, a velocidade de referência, (ω_m^*), diminuiu gradativamente até o valor de 200rad/s . Neste processo, em ambos transitórios, a resposta da velocidade ($\omega_{m.est.RN}$), atinge corretamente as variações indicadas pela velocidade de referência (ω_m^*).

Na Figura 55, apresenta-se o torque mecânico na carga. Observa-se uma variação do torque durante o primeiro transitório de 15N.m para, aproximadamente, 58N.m . No segundo transitório o torque foi reduzido para 24N.m , aproximadamente, quando o motor opera em regime permanente.

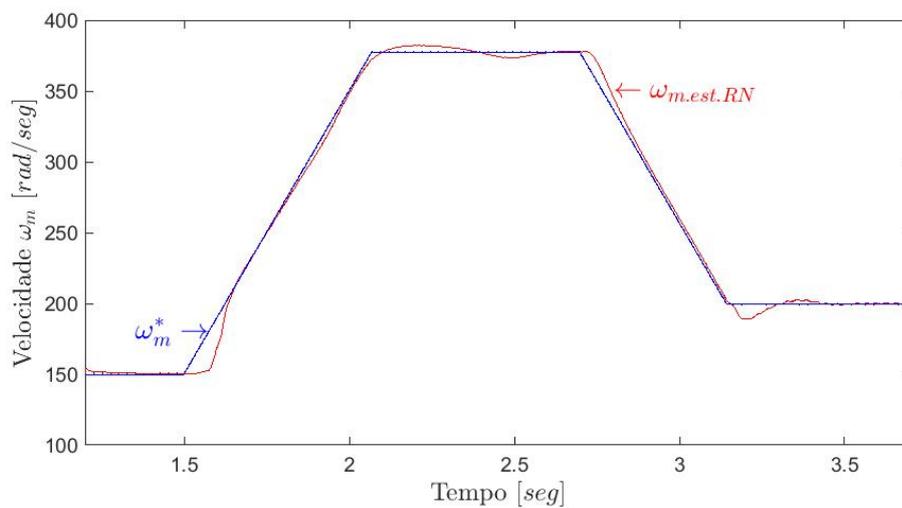


Figura 54: - Comparação entre a velocidade estimada ($\omega_{m.est.RN}$) com a velocidade de referência (ω_m^*).

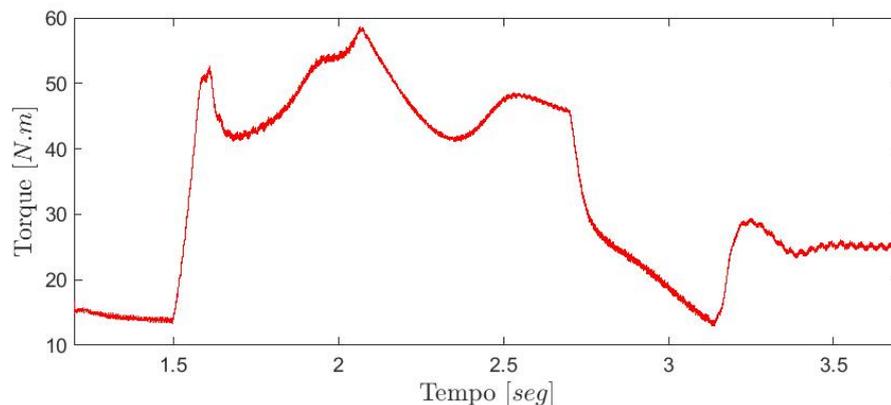


Figura 55: - Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.

4.3 Acionamento do MSIP sem sensores de posição com uso de um estimador neural de velocidade baseado nos dados provenientes do controle sem sensores - IFO

No estimador neural proposto nesta seção, os dados foram extraídos a partir da simulação feita no capítulo 2, onde a velocidade do motor $\omega_{m.est.IFO}$ foi calculada de forma determinística a partir de um IPLL (*Improved Phase Locked loop*). Assim, de forma semelhante no estimador da seção anterior, o objetivo principal foi estimar $\omega_{m.est.RN.IFO}$ a partir de parâmetros baseados nas leituras de correntes e tensões do estator. O código fonte pode ser obtido no Anexo B.

4.3.1 Definição da base de dados

Neste estimador neural de velocidade, a coleta de dados foi realizada considerando os 12 parâmetros, conforme descrito na Tabela 6; destes parâmetros, 11 foram definidos como atributos de entrada, e a diferença com a abordagem da seção anterior está na definição do alvo desejado, que neste teste corresponde ao parâmetro $\omega_{m.est.RN.IFO}$. Assim, considerando o mesmo tempo de simulação, foram coletadas 2.500.000 amostras, as quais após da subamostragem, resultou uma base de dados composta por 250.000 amostras.

Tabela 6: Descrição dos atributos e alvo da base de dados.

Definição de Atributos	Simbologia
Correntes de fase	i_a, i_b, i_c
Tensões de fase	v_a, v_b, v_c
Correntes dos eixos direto e quadratura	i_d, i_q
Correntes agregada	I_{agr}
Tensão agregada	V_{agr}
Potencia Total	P_T
Definição do Alvo	Simbologia
Velocidade mecânica	$\omega_{m.est.IFO}$

4.3.2 Características da Rede Neural empregada

A Tabela 7 apresenta um resumo das três melhores configurações neste processo de aprendizado. O melhor desempenho corresponde a aquela da primeira configuração, o qual apresenta a medida de generalização mais baixa $RMSE = 0,0620$.

Na Figura 56, apresenta-se a arquitetura da rede neural implementada a partir da melhor configuração, onde a camada escondida vem conformada por 6 neurônios, e na Figura 57 verifica-se a capacidade de generalização do estimador neural.

Tabela 7: Descrição dos parâmetros da rede neural.

Descrição dos parâmetros	Config.I	Config.II	Config.III
Arquitetura da rede	Perceptron multicamadas		
Processo de treinamento	Supervisionado		
Função de ativação da camada de escondida	Tangente hiperbólica		
Função de ativação da camada de saída	Tangente hiperbólica		
Número de atributos	11		
Número de neurônios na camada escondida	6	5	4
Taxa de aprendizado	0,3	0,1	0,3
Termo do <i>momentum</i>	0,7	0,9	0,9
Número máximo de iterações (épocas)	5000	5000	5000
Medida de generalização (RMSE teste)	0,0620	0,0630	0,0633

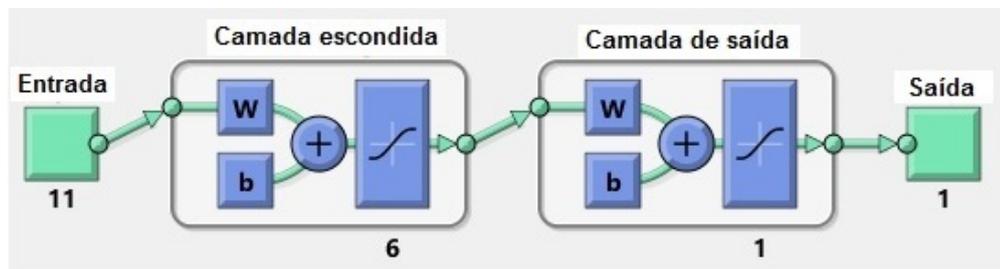


Figura 56: - Arquitetura da rede neural empregada.

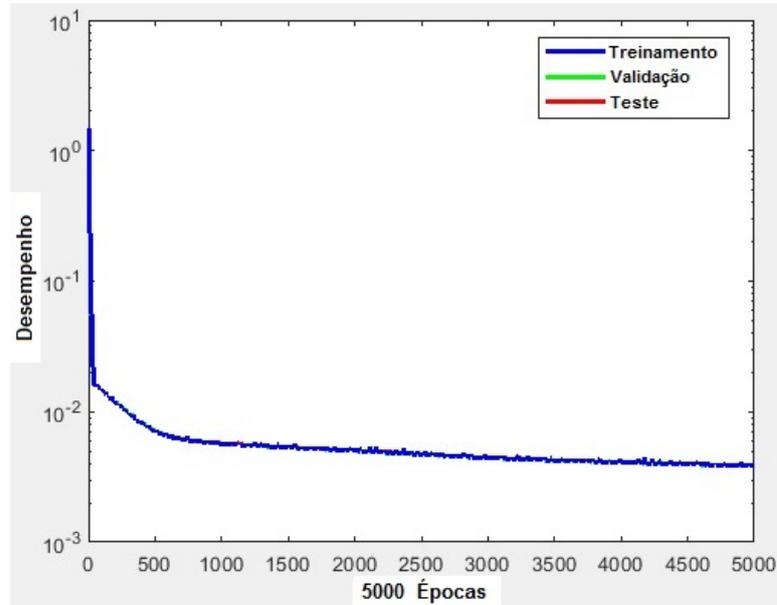


Figura 57: - Desempenho da rede neural empregada.

4.3.3 Resultados da Generalização do modelo

Em seguida, apresentam-se os resultados da estimativa da velocidade com a implementação do estimador neural ($\omega_{m.est.RN.IFO}$). Para isso foi considerado o ângulo de posicionamento da Figura 58 nesta simulação, o qual foi calculado no capítulo 2 a partir do sinal de saída gerado pelo IPLL, $\theta_{m.est.IFO}$.

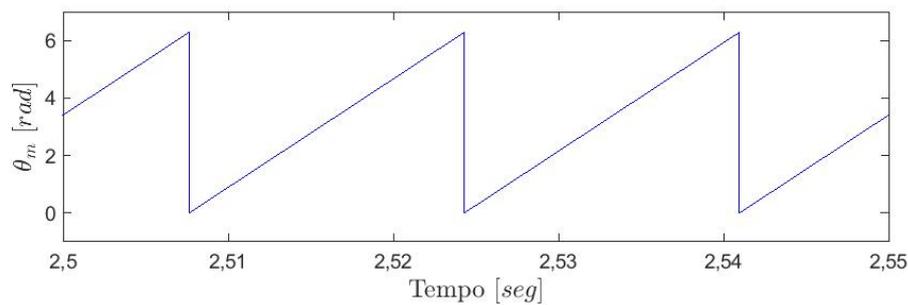


Figura 58: - Ângulo de posicionamento do rotor $\theta_{m.est.IFO}$ obtido com o IPLL.

Na Figura 59 apresenta-se a comparação entre a velocidade estimada a partir do estimador neural ($\omega_{m.est.RN.IFO}$) e, a velocidade de referência (ω_m^*). Este teste foi similar ao feito no Capítulo 1 e no Capítulo 2, considerando a variação gradativa da velocidade de referência, (ω_m^*), inicialmente entre os níveis de 150rad/s e 377rad/s , após de alcançar a velocidade nominal, a velocidade de referência, (ω_m^*), diminui para alcançar

uma velocidade 200rad/s . Neste processo, em ambos transitórios, a resposta da velocidade ($\omega_{m.est.RN}$), atinge corretamente as variações indicadas pela velocidade de referência (ω_m^*).

Na Figura 60, apresenta-se o torque mecânico na carga. Observa-se uma variação do torque durante o primeiro transitório de 15N.m para, aproximadamente, 63N.m . No segundo transitório o torque foi reduzido para 24N.m , aproximadamente, quando o motor opera em regime permanente.

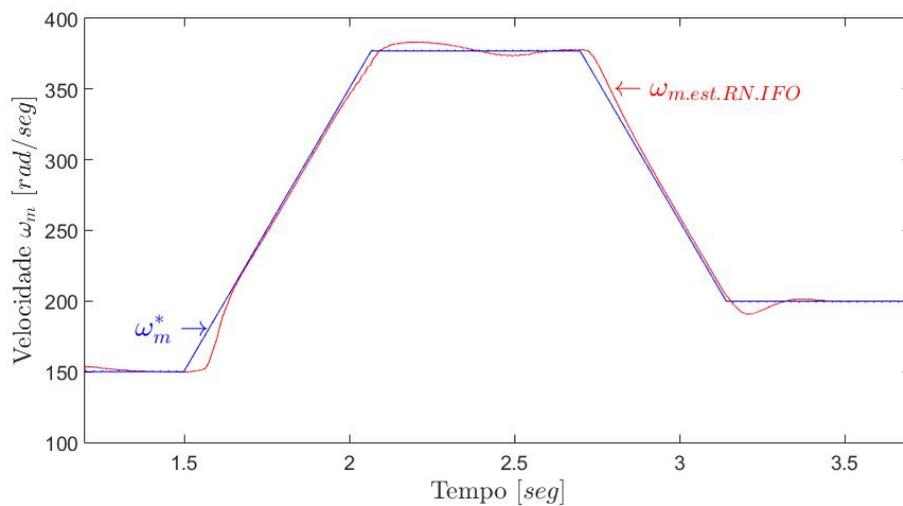


Figura 59: - Comparação entre a velocidade estimada ($\omega_{m.est.RN.IFO}$) com a velocidade de referência (ω_m^*).

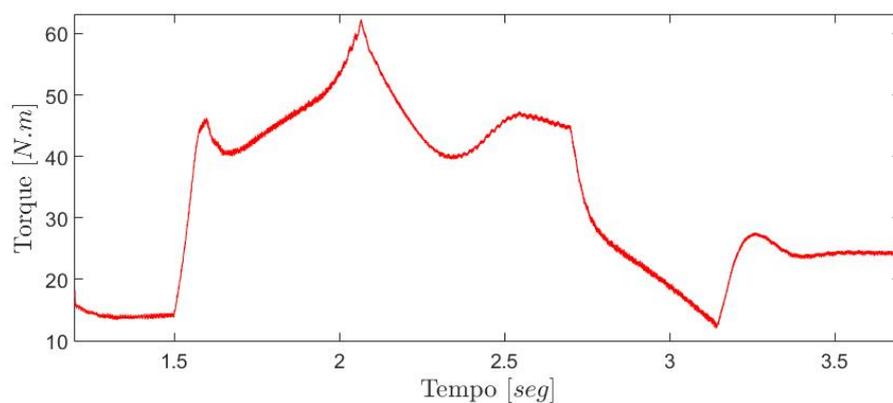


Figura 60: - Torque mecânico da máquina durante os transitórios em que a velocidade mecânica foi alterada.

4.4 Análise e resumo dos resultados

A Tabela 8 apresenta o resumo das duas melhores configurações que foram simuladas nas seções anteriores. Observa-se que o menor erro de generalização, RMSE, foi apresentado no estimador neural I, devido a que o sinal a ser estimado está fornecido pelo comportamento de um sensor mecânico. A diferença do estimador neural II, onde o sinal a ser estimado foi calculado a partir de um controle sem sensores. Para efeitos de comparação, foram utilizados os mesmos atributos com mesma quantidade de amostras para os estimadores obtidos a partir das duas bases de dados.

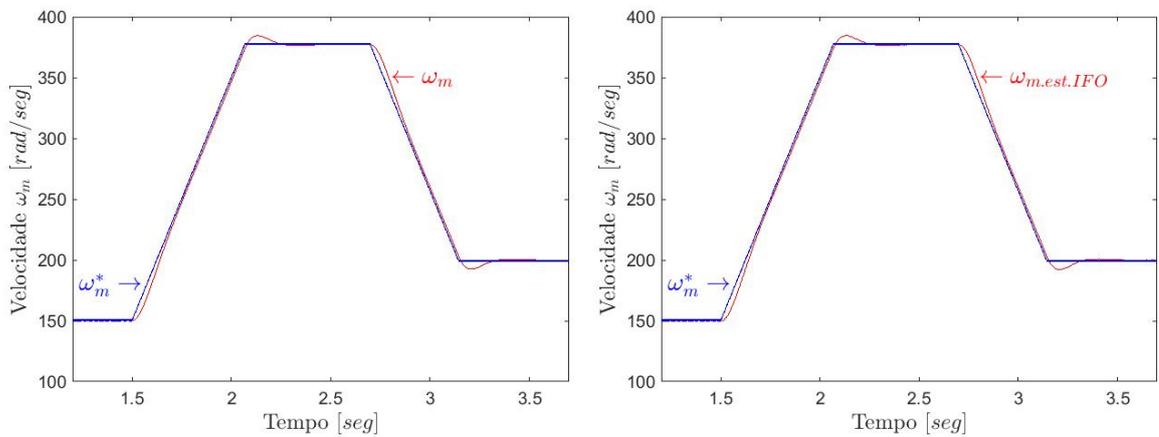
Tabela 8: Descrição dos parâmetros da rede neural.

Descrição dos parâmetros	Estimador neural I	Estimador neural II
Arquitetura da rede	Perceptron multicamadas	
Processo de treinamento	Supervisionado	
Função de ativação da camada de escondida	Tangente hiperbólica	
Função de ativação da camada de saída	Tangente hiperbólica	
Número de atributos	11	
Número de neurônios na camada escondida	5	6
Taxa de aprendizado	0,3	0,3
Termo do <i>momentum</i>	0,9	0,7
Número máximo de iterações (épocas)	5000	5000
Medida de generalização (RMSE teste)	0,0567	0,0620

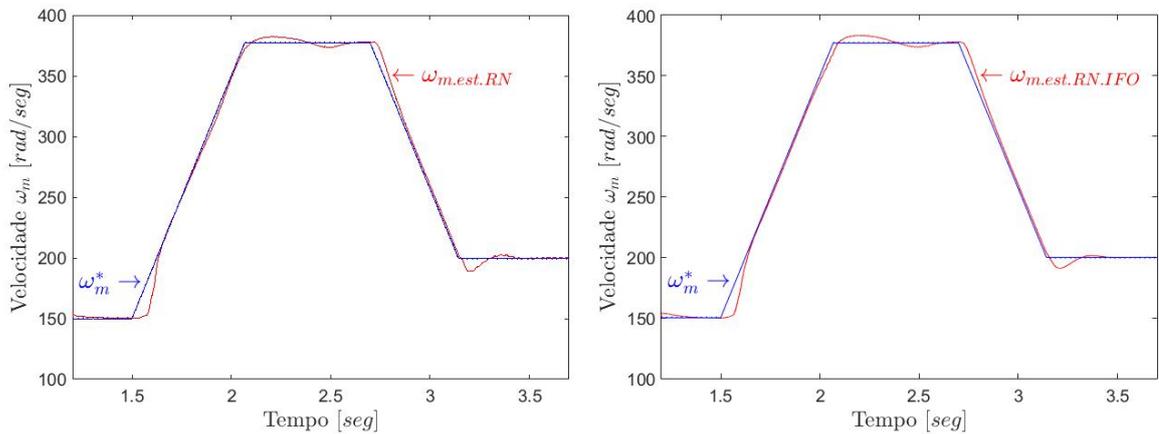
Na Figura 61 apresenta-se a comparação da forma de onda da velocidade resultante nos 4 cenários de simulação. Na Figura 61a apresenta-se a velocidade com o uso do sensor no eixo do motor. Na Figura 61b apresenta-se a velocidade $\omega_{m.est.IFO}$ estimada por meio de um circuito de sincronismo IPLL, que foi a primeira abordagem de controle sem sensores. Nas Figura 61c e Figura 61d apresentam-se a velocidade com o uso do estimador neural. A primeira foi baseada nos sinais fornecidos pela simulação como uso do sensor mecânico e na segunda os sinais foram fornecidos da simulação com uso de um IPLL.

É importante comentar que, com o uso da rede neural, apesar da capacidade de rastreamento da velocidade de referência, é observado uma dinâmica mais lenta de convergência, quando comparado com os resultados obtidos nas duas primeiras abordagens. No entanto, por outro lado, também vale ressaltar o fato do controlador apresentar o

mesmo comportamento a partir de ajustes nos parâmetros da rede utilizada.



(a) Velocidade (ω_m) com uso do sensor no eixo do MSIP. (b) Velocidade ($\omega_{m.est.IFO}$) com uso do estimador IFO.



(c) Velocidade ($\omega_{m.est.RN}$) com uso do estimador neural, com base de dados fornecidos pela simulação com sensor no eixo do MSIP. (d) Velocidade ($\omega_{m.est.RN.IFO}$) com uso do estimador neural, com base de dados fornecidos pela simulação com controlador - IFO .

Figura 61: Comparação da forma de onda da velocidade com os 4 cenários simulados.

Na Figura 62 apresenta-se a comparação da forma de onda do torque resultante nos 4 cenários de simulação. Na Figura 62a apresenta-se o torque com o uso do sensor no eixo do motor. Na Figura 62b apresenta-se o torque calculado por meio de um circuito de sincronismo IPLL. Na Figura 62c apresenta-se o torque calculado quando foi usado o primeiro controlador neural estimando $\omega_{m.est.RN}$. Finalmente a Figura 62d apresenta o torque a partir do segundo estimador neural estimando ($\omega_{m.est.RN.IFO}$).

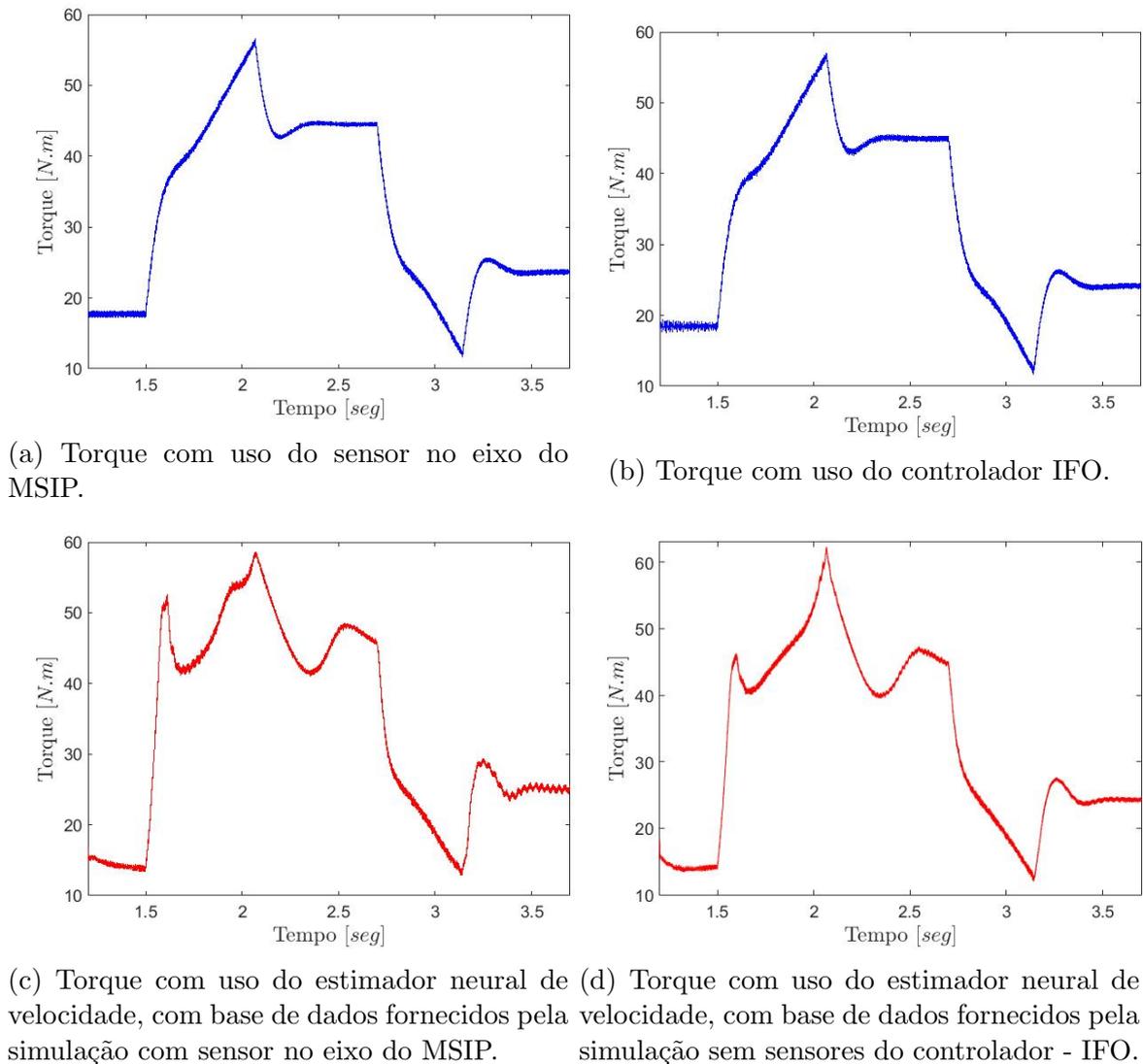


Figura 62: Comparação da forma de onda do torque com os 4 cenários simulados.

A dinâmica mais lenta com o uso da rede neural resultou em uma maior oscilação do torque durante o período em que ω_m^* foi igual a 377rad/s . Para a busca de um melhor desempenho é necessário o estudo de outros métodos de estimação como, por exemplo, o uso de filtros Kalman [37, 38], a definição de outros atributos, outras técnicas de subamostragem, entre outras alternativas. Além disso, há redes neurais com outras arquiteturas como as redes de funções de base radial RBF (*Radial Basis Function*) [39, 40], redes neurais recorrentes RNN (*Recurrent Neural Networks*) [41], ou mesmo o uso da técnica de inteligência artificial baseada em lógica difusa em conjunto com as redes neurais artificiais FNN (*Fuzzy Neural Network*) [42, 43].

No entanto, apesar desta dinâmica mais lenta, é importante destacar a capacidade de estimar a velocidade a partir de uma rede neural com arquitetura simples de duas

camadas, o que foi o objetivo maior deste trabalho. De fato, para a escolha do arranjo da rede neural deve ser estabelecido um compromisso entre o uso de um menor número de camadas para uma resposta mais rápida ou o uso de um número maior de camadas para ter uma resposta mais assertiva. Neste trabalho, em particular como se trata do uso de controladores com capacidade de ação em tempo real, foi escolhida a abordagem mais simples possível sem comprometer a capacidade em estimar a velocidade da máquina. Entretanto, em [44] foi apresentado o controlador para o MSIP com o estimador baseado na rede *perceptron* com um maior número de camadas intermediárias. Neste sentido, como continuidade deste estudo, pretende-se implantar a rede *perceptron* com um número maior de camadas e, em seguida, fazer uso das técnicas mencionadas para estimar a velocidade.

CONCLUSÃO

Este trabalho apresentou uma abordagem alternativa para estimar a velocidade do MSIP (Motor Síncrono de Ímã Permanente) baseado em técnicas da inteligência artificial. Para tal foram usadas as redes neurais artificiais do tipo *Perceptron* multicamadas MLP (*Multilayer Perceptron*), em conjunto com o controlador orientado por campo FOC (*Field Oriented Control*). A partir dos resultados obtidos em simulação foi verificada a capacidade da rede em estimar corretamente a velocidade, onde o controlador apresentou um desempenho similar ao observado nas primeiras abordagens.

Neste contexto, o projeto do estimador neural de velocidade foi dividido em três abordagens. Na primeira foi avaliado o funcionamento do acionamento do MSIP com base no controlador orientado por campo FOC. A escolha desta técnica de controle se deu pela sua implementação simples com a capacidade de manter a velocidade controlada, a todo instante, durante a simulação. Inicialmente foi implementado o controle FOC na sua versão mais simples com uso de um sensor de posição para extrair a informação da posição angular e a velocidade.

A validação desta técnica ocorreu por meio de um caso teste com o motor operando, inicialmente, a uma velocidade de 150rad/s . Em seguida a velocidade de referência (ω_m^*) foi elevada gradativamente para 377rad/s e permanece nesta velocidade, sendo esta a velocidade nominal. O segundo transitório consistiu na redução gradativa de ω_m^* para 200rad/s . Conforme os resultados obtidos nestas simulações, a velocidade do motor (ω_m) está em conformidade com a referência, indicando o desempenho satisfatório do controlador FOC.

Em seguida foi avaliado o comportamento do controlador FOC com o primeiro estimador de velocidade. Nesta abordagem, o sensor de posição foi eliminado para a implantação do estimador com base no observador do fluxo, em conjunto com o circuito de sincronismo IPLL (*Improved Phase Locked Loop*). A avaliação desta abordagem ocorreu com mesmo caso teste utilizado na primeira simulação, onde ω_m^* apresentou os mesmos transitórios. Conforme aos resultados obtidos nas simulações, a velocidade estimada pelo IPLL esteve em conformidade com as variações da referência durante toda a simulação. Além disso, verificou-se um comportamento similar do controlador com este estimador, quando comparado com o observado na primeira abordagem, onde foi utilizado um sensor

de posição para obter a informação da velocidade.

Continuando, foi utilizado a rede neural para estimar a velocidade. Para tal os parâmetros da rede neural foram ajustados a partir de duas bases de dados diferentes. A primeira base de dados consistiu nos sinais extraídos (velocidade e posição angular) do sensor de posição utilizado na primeira simulação. O segundo estimador neural foi obtido com base na velocidade estimada pelo IPLL. A posição angular permaneceu calculada pelo observador de fluxo. Esta abordagem consistiu na contribuição deste trabalho que entendemos ser original, visto que não foi observada na literatura uma proposta similar à apresentada.

Na sequência foram feitas simulações considerando o mesmo caso teste utilizado nas duas primeiras abordagens. O objetivo foi avaliar a capacidade de generalização da rede neural em estimar a velocidade. Em ambos casos, tem-se a capacidade de rastreamento da velocidade de referencia. No entanto, ambos apresentam uma dinâmica mais lenta de convergência, quando comparado com os resultados obtidos nas duas primeiras abordagens.

A dinâmica mais lenta com o uso da rede neural resultou em uma maior oscilação do torque durante o período do primeiro transitório onde a velocidade mecânica varia de 150rad/s para a velocidade nominal 377rad/s . No entanto, apesar desta dinâmica mais lenta, verificou-se a capacidade de estimar a velocidade a partir de uma rede neural com arquitetura simples de duas camadas.

A escolha pela arquitetura MLP se deu por ser um aproximador de funções universal que tem um melhor desempenho quando comparado com outras arquiteturas mais simples como, por exemplo, a ADALINE (*Adaptive Linear Element*) ou mesmo a arquitetura *Perceptron* com um neurônio.

No Capítulo 4 foi apresentado um resumo das três melhores configurações com a rede neural utilizada, após de serem testadas, aproximadamente, um total de 168 configurações diferentes da rede neural. Deste modo, é importante destacar que não necessariamente um maior numero de neurônios apresenta a melhor generalização. De fato, a partir destes testes, a melhor estimação da rede neural ocorreu com 5 neurônios na camada escondida, onde a principal vantagem obtida nesta configuração se deu pela menor esforço computacional. Como consequência, considerando a aplicação com controladores em tempo real, este aspecto é importante para não comprometer o desempenho do

controlador em questão.

Além disso, conforme descrito no Capítulo 4, há outros métodos observados na literatura que foram utilizados para estimar a velocidade do motor em condições similares às apresentadas neste trabalho. Há também a possibilidade em fazer uso da rede neural com a arquitetura MLP com um maior número de camadas conforme apresentado em [44]. Neste sentido, como trabalhos futuros, pretende-se dar a continuidade explorando estas alternativas com objetivo de obter um estimador capaz de apresentar um desempenho similar ou melhor quando comparado com a abordagem tradicional.

REFERÊNCIAS

- [1] J. Liu, T. A. Nondahl, P. B. Schmidt, S. Royak, and T. M. Rowan. “Generalized Stability Control for Open-loop Operation of Motor Drives”. *IEEE Transactions on Industry Applications*, 53(3):2517–2525, May 2017.
- [2] Xiaodong Liang. “Innovative design and feasibility study for a subsea electrical submersible pump system”. In *2016 IEEE/IAS 52nd Industrial and Commercial Power Systems Technical Conference (I CPS)*, pages 1–10, May 2016.
- [3] B. Monsen, K. S. Rongve, T. Lægveid, and C. Gutscher. “Åsgard subsea gas compression—technology qualification testing with high-speed vsd and very long step-out cable”. *IEEE Transactions on Industry Applications*, 50(2):1552–1561, March 2014.
- [4] G. Da Cunha, A. J. Rossa, J. A. Alves, and E. Cardoso. “Control of Permanent Magnet Synchronous Machines for Subsea Applications”. *IEEE Transactions on Industry Applications*, 54(2):1899–1905, March 2018.
- [5] Z. Li, Q. Zhang, H. Luo, H. Wang, J. Wang, F. Han, A. Wang, X. Liu, X. Yu, and L. Zhou. “Sensorless Starting Control of Permanent Magnet Synchronous Motors with Step-up Transformer for Downhole Electric Drilling”. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 689–694, Oct 2018.
- [6] James E. Hipp and New Iberia. ”High torque, low speed mud motor for use in drilling oil and gas well.”, April 18, 2000. US Patent 6,050,346.
- [7] Bimal K. Bose. “*Modern Power Electronics and AC Drives*”. Prentice Hall PTR, 2002.
- [8] W. Xu, L. Wang, Y. Liu, and F. Blaabjerg. “Improved Rotor Flux Observer for Sensorless Control of PMSM with Adaptive Harmonic Elimination and Phase Compensation.”. *CES Transactions on Electrical Machines and Systems*, 3(2):151–159, June 2019.

- [9] Stephen J. Chapman. “*Fundamentos de Máquinas Eléctricas*”. AMGH Editora Ltda., 2013.
- [10] Stephen D. Umans. “*Máquinas Eléctricas de Fitzgerald e Kingsley*”. AMGH Editora Ltda., 2014.
- [11] T. Qiu, X. Wen, and F. Zhao. Adaptive-linear-neuron-based dead-time effects compensation scheme for pmsm drives. *IEEE Transactions on Power Electronics*, 31(3):2530–2538, March 2016.
- [12] D. T. Vu, N. K. Nguyen, E. Semail, and T. J. dos Santos Moraes. Control strategies for non-sinusoidal multiphase pmsm drives in faulty modes under constraints on copper losses and peak phase voltage. *IET Electric Power Applications*, 13(11):1743–1752, 2019.
- [13] W. C. Duesterhoeft, M. W. Schulz, and E. Clarke. Determination of instantaneous currents and voltages by means of alpha, beta, and zero components. *Transactions of the American Institute of Electrical Engineers*, 70(2):1248–1255, July 1951.
- [14] R. H. Park. Two-reaction theory of synchronous machines generalized method of analysis-part i. *Transactions of the American Institute of Electrical Engineers*, 48(3):716–727, July 1929.
- [15] Y. Yan, S. Wang, C. Xia, H. Wang, and T. Shi. “Hybrid Control Set-Model Predictive Control for Field-Oriented Control of VSI-PMSM”. *IEEE Transactions on Energy Conversion*, 31(4):1622–1633, Dec 2016.
- [16] R. Krishnan. “*Permanent Magnet Synchronous and Brushless DC Motor Drives*”. CRC Press, 2010.
- [17] E. A. Rodriguez, C. M. Freitas, M. D. Bellar, and L. F. C. Monteiro. Mppt algorithm for pv array connected to a hybrid generation system. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pages 1115–1120, June 2015.
- [18] L. F. C. Monteiro, C. M. Freitas, and M. D. Bellar. Improvements on the incremental conductance mppt method applied to a pv string with single-phase to three-phase

- converter for rural grid applications. *Advances in Electrical and Computer Engineering*, 19(1):63–71, 2019.
- [19] S. Kim and S. Sul. “High Performance Position Sensorless Control Using Rotating Voltage Signal Injection in IPMSM”. In *Proceedings of the 2011 14th European Conference on Power Electronics and Applications*, pages 1–10, Aug 2011.
- [20] D. Xu, B. Wang, G. Zhang, G. Wang, and Y. Yu. “A Review of Sensorless Control Methods for AC Motor Drives”. *CES Transactions on Electrical Machines and Systems*, 2(1):104–115, March 2018.
- [21] P. L. Xu and Z. Q. Zhu. “Novel Square-Wave Signal Injection Method Using Zero-Sequence Voltage for Sensorless Control of PMSM Drives”. *IEEE Transactions on Industrial Electronics*, 63(12):7444–7454, Dec 2016.
- [22] S. Kim, J. Im, E. Song, and R. Kim. “A New Rotor Position Estimation Method of IPMSM Using All-Pass Filter on High-Frequency Rotating Voltage Signal Injection”. *IEEE Transactions on Industrial Electronics*, 63(10):6499–6509, Oct 2016.
- [23] Y. Lee, Y. Kwon, and S. Sul. “Comparison of Rotor Position Estimation Performance in Fundamental-Model-Based Sensorless Control of PMSM”. In *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 5624–5633, Sep. 2015.
- [24] J. Liu, T. A. Nondahl, P. B. Schmidt, S. Royak, and M. Harbaugh. “Rotor Position Estimation for Synchronous Machines Based on Equivalent EMF”. *IEEE Transactions on Industry Applications*, 47(3):1310–1318, May 2011.
- [25] V. Repecho, D. Biel, and A. Arias. “Fixed Switching Period Discrete-Time Sliding Mode Current Control of a PMSM”. *IEEE Transactions on Industrial Electronics*, 65(3):2039–2048, March 2018.
- [26] H. Kim, J. Son, and J. Lee. “A High-Speed Sliding-Mode Observer for the Sensorless Speed Control of a PMSM”. *IEEE Transactions on Industrial Electronics*, 58(9):4069–4077, Sep. 2011.
- [27] N. K. Quang, N. T. Hieu, and Q. P. Ha. “FPGA-Based Sensorless PMSM Speed Control Using Reduced-Order Extended Kalman Filters”. *IEEE Transactions on Industrial Electronics*, 61(12):6574–6582, Dec 2014.

- [28] H. Li and Z. Wang. “Sensorless Control for PMSM Drives Using the Cubature Kalman Filter Based Speed and Flux Observer”. In *2018 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC)*, pages 1–6, Nov 2018.
- [29] Y. Li, M. Yang, J. Long, Z. Liu, and D. Xu. “Current Sensorless Predictive Control Based on Extended Kalman Filter for PMSM Drives”. In *2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, pages 1–6, Aug 2017.
- [30] G. Zhang, G. Wang, D. Xu, and N. Zhao. “ADALINE-Network-Based PLL for Position Sensorless Interior Permanent Magnet Synchronous Motor Drives”. *IEEE Transactions on Power Electronics*, 31(2):1450–1460, Feb 2016.
- [31] O. Elbeji and M. Ben Hamed. “Sensorless Vector Control of Permanent Magnet Synchronous Motor Using Neural Network Observer”. In *2017 International Conference on Green Energy Conversion Systems (GECS)*, pages 1–6, March 2017.
- [32] J. Park and S. Mackay. *“Practical Data Acquisition for Instrumentation and Control Systems”*. Elsevier, 2003.
- [33] J. Han, M. Kamber, and J. Pei. *“Data Mining Concepts and Techniques”*. Elsevier, 3 edition, 2011.
- [34] L. A. Shalabi and Z. Shaaban. Normalization as a preprocessing engine for data mining and the approach of preference matrix. In *2006 International Conference on Dependability of Computer Systems*, pages 207–214, May 2006.
- [35] I. Nunes, D. Hernane, and R. Andrade. *“Redes Neurais Artificiais para engenharia e ciências aplicadas”*. Artliber, 2010.
- [36] S. Haykin. *“Redes Neurais: princípios e prática”*. Bookman, 2010.
- [37] M. S. Termizi, J. M. Lazi, Z. Ibrahim, M. H. N. Talib, M. J. A. Aziz, and S. M. Ayob. Sensorless pmsm drives using extended kalman filter (ekf). In *2017 IEEE Conference on Energy Conversion (CENCON)*, pages 145–150, Oct 2017.

- [38] D. G. Pillai, A. Vivek, and V. Srikanth. Non-linear state estimation of pmsm using derivative-free and square-root cubature kalman filter. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, pages 126–131, July 2017.
- [39] H. Feifei, W. Zhonghua, L. Yueyang, and H. Tongyi. Sensorless speed control of permanent magnet synchronous motor based on rbf neural network. In *2015 34th Chinese Control Conference (CCC)*, pages 4325–4330, July 2015.
- [40] P. Nguyen and M. Nguyen. Design and implementation of a rbf-based pi controller for pmsm drives. In *2017 International Conference on System Science and Engineering (ICSSE)*, pages 103–106, July 2017.
- [41] Y. Wang, Y. Hoole, and K. Haran. Position estimation of outer rotor pmsm using linear hall effect sensors and neural networks. In *2019 IEEE International Electric Machines Drives Conference (IEMDC)*, pages 895–900, May 2019.
- [42] L. Sheng, G. Xiaojie, and Z. Lanyong. Robust adaptive backstepping sliding mode control for six-phase permanent magnet synchronous motor using recurrent wavelet fuzzy neural network. *IEEE Access*, 5:14502–14515, 2017.
- [43] Q. Liu and X. Chang. Position ip control of a permanent magnet synchronous motor based on fuzzy neural network. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 1081–1086, June 2018.
- [44] J. Gao, S. Cao, and S. Zhang. The neural network control approach for pmsm based on a high gain observer. In *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 1066–1071, May 2018.

A COMUNICAÇÃO ENTRE PROGRAMAS

O manual para o desenvolvimento da comunicação entre os programas Simulink/MATLAB e PSIM está disponível na plataforma GitHub através da url: https://github.com/MarthaRoxana/PEL_UERJ_Martha_2019/blob/master/MatlabPSIM.zip.

B CÓDIGO FONTE DO ESTIMADOR NEURAL PROPOSTO

O código fonte das simulações do estimador neural para o acionamento do MSIP está disponível na plataforma GitHub através da url: https://github.com/MarthaRoxana/PEL_UERJ_Martha_2019/blob/master/Estimador_Neural.zip.