

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE UMA INTERFACE IHM PARA
CONFIGURAÇÃO DO INVERSOR DE FREQUÊNCIA
CFW-11 WEG**

Gutierre Apolinário de Lima

Brasília, Dezembro de 2020



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

IMPLEMENTAÇÃO DE UMA INTERFACE IHM PARA
CONFIGURAÇÃO DO INVERSOR DE FREQUÊNCIA
CFW-11 WEG

Gutierre Apolinário de Lima

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Lélío Ribeiro Soares Júnior, ENE/UnB

Orientador

Prof. Fernando Cardoso Guimarães, ENE/UnB

Examinador interno

Prof. Fernando Cardoso Melo, ENE/UnB

Examinador interno

Brasília, Dezembro de 2020

FICHA CATALOGRÁFICA

LIMA, G.A, (2020). Implementação de uma interface IHM para a configuração do inversor de frequência CFW-11 Weg,

[Distrito Federal] 2020.

XI, 50p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2020). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1. Protocolo de comunicação ModBus

2.Biblioteca *Python*

3. IHM (Interface Homem Máquina)

I. Mecatrônica/FT/UnB

II. Implementação de uma interface IHM para a configuração do

inversor de frequência CFW-11 Weg

REFERÊNCIA BIBLIOGRÁFICA

LIMA, G.A, (2020).Implementação de uma interface IHM para a configuração do inversor de frequência CFW-11 Weg. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°14, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 50p.

CESSÃO DE DIREITOS

AUTOR: Gutierre Apolinário de Lima

TÍTULO DO TRABALHO DE GRADUAÇÃO: Implementação de uma interface IHM para a configuração do inversor de frequência CFW-11 Weg.

GRAU: Engenheiro

ANO: 2020

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Gutierre Apolinário de Lima

Campus Darcy Ribeiro, FT, Universidade de Brasília.

70919-970 Brasília – DF – Brasil.

Dedicatória

Dedico a toda minha família, em especial meu pai, minha querida mãe e minha irmã, pelo empenho e pelas doses de ânimo a cada ligação realizada.

Gutierre Apolinário de Lima

Agradecimentos

Agradeço a minha querida tia Luzia e a meu amado tio José Carlos, agradeço aos meus amados primos Jeremias, Luana, Carlos Henrique, Karla, Alessandro e Maria Divina, que não mediram esforços em me ajudar nos momentos difíceis. Aos meus amigos Felipe, Ricardo, Vitor e Leonardo pela força ao longo do caminho. Aos professores que fizeram de mim uma pessoa mais forte e em especial ao meu orientador, Professor Lélío, pela sua sabedoria, gentileza e prontidão.

Gutierre Apolinário de Lima

RESUMO

O crescimento de forma sustentável vem sendo discutido mundo afora ultimamente. Novas formas renováveis de geração de energia elétrica são muito importantes nesse atual cenário de desenvolvimento. A viabilidade desses modelos energéticos está relacionada com estudos científicos, seja de simulação, levantamento de dados ou até mesmo análises computacionais de desempenho. No Brasil foi possível perceber uma capacidade energética potencial e o aumento dos parques eólicos já é realidade em estados do Nordeste.

Esse trabalho possibilita o estudo dessa tecnologia, por meio da emulação de uma turbina eólica, em um estado com baixo potencial de ventos para implementação de um modelo real de testes. A bancada em questão é formada basicamente por um motor de indução trifásico e um inversor de frequência CFW-11 da Weg. A simulação realizada utiliza de uma interface gráfica, executada em um *notebook*, no qual o usuário entra com as suas preferências. O protocolo de comunicação ModBus é utilizado para troca de mensagens entre *notebook* e inversor de frequência, por meio de bibliotecas *Python*.

Palavras Chave: Comunicação, ModBus, Eólica, *Python*

ABSTRACT

Growth in a sustainable way has been discussed around the world lately. New renewable forms of electricity generation are very important in this development scenario. The feasibility of energy models is related to scientific studies, based on simulation, data collection or even analysis of computational performance. In Brazil, it was possible to perceive potential energy capacity and the increase in wind farms is already a reality in states in the Northeast Region.

This work allows the study of this technology through the emulation of a wind turbine, in a state with low wind potential for the implementation of a real test model. The bench in question is basically formed by a three-phase induction motor and a CFW-11 frequency inverter from Weg manufacturer. The simulation performed uses a graphical interface, executed on a notebook, where the user enters with the best conditions. The ModBus communication protocol is used to exchange messages between notebook and frequency inverter through Python libraries

Keywords: Communication, ModBus, Wind, Python

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	4
1.3	OBJETIVOS DO PROJETO	4
2	Referencial Teórico	5
2.1	ENERGIA EÓLICA	5
2.1.1	A POTÊNCIA DO VENTO	6
2.2	MODELOS DE TURBINAS EÓLICAS	7
2.3	TURBINA EIXO HORIZONTAL	8
2.3.1	A POTÊNCIA DE UMA TURBINA IDEAL	9
2.3.2	A POTÊNCIA DE UMA TURBINA REAL	10
2.3.3	VELOCIDADE DE PONTA	11
2.4	INVERSOR DE FREQUÊNCIA	12
2.4.1	UNIDADE CENTRAL DE PROCESSAMENTO (CPU)	13
2.4.2	INTERFACE HOMEM/MÁQUINA (IHM)	14
2.4.3	INTERFACES	14
2.4.4	SISTEMA DE POTÊNCIA	14
2.4.5	PARÂMETROS	18
2.4.6	FORMAS DE CONTROLE	19
2.4.7	CONTROLE ESCALAR (V/F)	19
2.5	COMUNICAÇÃO SERIAL	22
2.5.1	TAXA DE TRANSMISSÃO	23
2.5.2	BITS DE DADOS	23
2.5.3	BITS DE PARADA	23
2.5.4	PARIDADE	23
2.5.5	RS485	24
2.5.6	COMUNICAÇÃO USB-SERIAL	27
2.6	PROTOCOLO DE COMUNICAÇÃO MODBUS	27
2.6.1	MODBUS-RTU	29
2.6.2	BIBLIOTECA MINIMALMODBUS	32
2.6.3	BIBLIOTECA PYSERIAL	32
2.7	MOTOR DE INDUÇÃO TRIFÁSICO	33

2.7.1	PRINCÍPIO DE FUNCIONAMENTO	33
3	Descrição da bancada	35
3.1	MOTOR DE INDUÇÃO TRIFÁSICO.....	35
3.2	MOTOR DO VENTILADOR.....	36
3.3	INVERSOR DE FREQUÊNCIA	37
3.4	MÓDULO RS-485.....	38
3.5	ADAPTADOR USB-SERIAL	39
4	Comunicação e Controle	40
4.1	CONFIGURAÇÃO DO INVERSOR PARA ACESSO REMOTO	40
4.2	COMUNICAÇÃO VIA <i>Software SuperDrive</i>	40
4.3	COMUNICAÇÃO VIA MINIMALMODBUS E PYSERIAL	42
4.3.1	ESTABELECENDO CONEXÃO	43
4.4	IMPLEMENTAÇÃO DO CONTROLE	45
5	Conclusões.....	49
5.1	PERSPECTIVAS FUTURAS.....	49
	REFERÊNCIAS BIBLIOGRÁFICAS	51
	Anexos.....	54
I	Diagrama elétrico.....	55
II	Código em Python.....	56

LISTA DE FIGURAS

1.1	Composição da matriz energética Brasileira (ABEEÓLICA, 2019).....	2
1.2	Evolução da Capacidade Instalada no Brasil, em MW (ABEEÓLICA, 2019).	3
1.3	Capacidade e número de parques eólicos por estado (ABEEÓLICA, 2019).	3
2.1	Formação dos ventos devidos ao deslocamento das massas de ar (DUTRA, 2019). ...	5
2.2	Evolução dos Aerogeradores (REIS, 2016).	6
2.3	Modelos de turbinas eólicas (STROSK, 2019).....	8
2.4	Esquema básico e componentes constituintes de uma turbina (REIS, 2016).	9
2.5	Fluxo de vento por uma turbina eólica (REIS, 2016).	10
2.6	Curvas relacionando a potência disponível no vento com as potências real e teórica extraídas pela turbina (RICO, 2017).	11
2.7	Inversores de frequência (WEG, 2009).	13
2.8	Componentes do inversor de frequência em blocos (FRANCHI, 2013).....	13
2.9	Interface Homem/Máquina CFW-11 (WEG, 2016).....	14
2.10	Diagrama de um inversor de frequência com alimentação trifásica (FRANCHI, 2013). 15	15
2.11	Fonte de tensão alternada aplicada a um retificador (FRANCHI, 2013).	15
2.12	Diagrama de um inversor de frequência alimentação monofásica (FRANCHI, 2013). 16	16
2.13	Lógica de controle de transistores,para inversor monofásico (FRANCHI, 2013).....	16
2.14	Frequência de chaveamento (FRANCHI, 2013).	17
2.15	Sinal de saída de um PWM (FRANCHI, 2013).....	17
2.16	Sinal de saída inversor de frequência (TOSSI, 2011).	18
2.17	Sinal de saída inversor de frequência (ELECTRONICS, 2020).	18
2.18	Diagrama de blocos do controle V/f (WEG, 2012).	21
2.19	Curva V/f com compensação de escorregamento (WEG, 2012).	22
2.20	Topologia do barramento e daisy chain (FREITAS, 2017).....	24
2.21	<i>Full-Duplex</i> e <i>Half-Duplex</i> , estrutura de transmissão no RS-485 (INSTRUMENTS, 2016).	25
2.22	Especificação de nível de sinal para transmissão RS-485 (INSTRUMENTS, 2016)....	25
2.23	Modelo de cabo de comunicação RS-485, juntamente com suas especificações (INS- TRUMENTS, 2016).....	26
2.24	Taxa de dados (INSTRUMENTS, 2016).....	26
2.25	Conversor industrial USB para Serial DB-9 (PORT, 2015).....	27
2.26	Layout de Hardware adaptador USB para porta (RS422/485) (PORT, 2015).....	27
2.27	Relação mestre-escravo (INSTRUMENTS, 2019).	28

2.28	Transação ModBus sem erro (MODBUS, 2012).	29
2.29	Transação ModBus com erro (MODBUS, 2012).	29
2.30	PDU do ModBus (INSTRUMENTS, 2019).	29
2.31	ADU do ModBus-RTU (INSTRUMENTS, 2019).	30
2.32	ADU do ModBus-RTU (WEG, 2007).	30
2.33	Faixas de tempo para o modo ModBus-RTU (WEG, 2010).	31
2.34	Exemplo uso MinimalModbus (BERG, 2019).	32
2.35	Exemplo uso pySerial (LIECHTI, 2017).	32
2.36	Vista de corte motor de indução (CHAPMAN, 2013).	33
2.37	Campo girante de estator induzindo tensão nas barras do rotor.(CHAPMAN, 2013)	34
3.1	Motor de indução trifásico e ventilador (MENEZES; SATO, 2019).	36
3.2	Inversor de frequência utilizado no projeto (MENEZES; SATO, 2019).	37
3.3	Principais componentes do CFW-11 (WEG, 2016).	38
3.4	Módulo RS-485.(INDUSTRIAL, 2020).....	39
4.1	Configurações estabelecidas no Software SuperDrive.	41
4.2	Acesso aos dados do inversor via Software <i>SuperDrive</i>	42
4.3	Janela principal da interface do usuário.	46
4.4	Janela com mensagem de erro após a tentativa de conexão sem sucesso.	46
4.5	Janela com a seleção do controle escalar.	47
4.6	<i>Hold</i> de rampa(WEG, 2012).	48

LISTA DE TABELAS

2.1	Parâmetros importantes para configuração do inversor de frequência	20
2.2	Taxas de comunicação e tempos envolvidos na transmissão de telegramas (WEG, 2010)	32
3.1	Dados do Motor de Indução Trifásico (IVO, 2017).	35
3.2	Dados do ventilador (MENEZES; SATO, 2019).	36
3.3	Dados do inversor de frequência (MENEZES; SATO, 2019).	37
3.4	Descrição dos componentes CFW-11	38
3.5	Dados do módulo RS-485 (IVO, 2017).	39
3.6	Dados do adaptador USB-Serial (IVO, 2017).	39
4.1	Parâmetros ajustados para acesso remoto.....	40
4.2	Configuração USB Serial Port (COM3).....	41

LISTA DE SÍMBOLOS

Siglas

WWEA	<i>World Wind Energy Association</i>
ABEEólica	<i>Associação Brasileira de Energia Eólica</i>
CEBDS	<i>Conselho Empresarial Brasileiro para o Desenvolvimento Sustentável</i>
ACR	<i>Ambiente de Contratação Regulada</i>
ACL	<i>Ambiente de Contratação Livre</i>
MW	<i>Megawatt</i>
GW	<i>Gigawatt</i>
RPM	<i>Rotações por minuto</i>
PWM	<i>Pulse Width Modulation</i>
CPU	<i>Unidade Central de Processamento</i>
IHM	<i>Interface Homem Máquina</i>
CA	<i>Corrente alternada</i>
CC	<i>Corrente contínua</i>
MIT	<i>Motor de Indução Trifásico</i>
P.I.	<i>Placa de Interface</i>
ICW	<i>Initialization Command Words</i>
OCW	<i>Operational Control Word</i>
RS-485	<i>Recommended Standart-485</i>
EIA	<i>Electronics Industries Association</i>
UTP	<i>Unshield Twisted Pairn</i>
AWG	<i>American Wire Gauge</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
CLP	<i>Controlador Lógico Programável</i>
ADU	<i>Application Data Unit</i>
RTU	<i>Remote Terminal Unit</i>
PDU	<i>Protocol Data Unit</i>
CRC	<i>Cycling Redundancy Check</i>

Capítulo 1

Introdução

1.1 Contextualização

Formas de se obter energia limpa e renovável estão sendo estudadas e desenvolvidas cada vez mais. A preocupação com o meio ambiente nunca foi tão evidenciada como atualmente. A obtenção de energia elétrica com o mínimo possível de poluentes é importante e vista como um grande avanço na matriz energética de qualquer país.

Utilizada há aproximadamente três mil anos pela humanidade, a conversão da energia eólica em outro tipo de energia apropriada para movimentar mecanismos já era presente em moinhos de vento que moíam os grãos e bombeavam água para os mais variados tipos de atividade. O desenvolvimento das navegações e as descobertas realizadas de novos continentes foram propiciados em grande parte pelo emprego da energia dos ventos (MARTINS; GUARNIERI; PEREIRA, 2007).

Um aproveitamento bastante importante que ocorre há alguns anos é a conversão da energia eólica, como fonte alternativa de energia em eletricidade. Segundo (HERBERT; INIYAN; RAJAPADIAN, 2007) pesquisas que envolvem o aproveitamento dessa modalidade de energia são realizadas há mais de cento e cinquenta anos. Nos dias atuais a energia eólica, em conjunto com a solar, vem sendo apontada como uma fonte de energia sustentável promissora a curto prazo, por possuir uma certa viabilidade econômica e possuir baixo custo socioambiental.

Os avanços na utilização dessa modalidade de energia, em nível mundial, são bastante satisfatórios. É possível perceber um crescimento da capacidade instalada. A forma como ocorre esse crescimento geral, pautado na robustez da energia eólica mundial, acompanha de forma satisfatória a diversificação geográfica adicional.

Todas as turbinas eólicas instaladas até o final de 2017 podem cobrir mais de 5% da demanda global de eletricidade. Para muitos países, a energia eólica se tornou um pilar em suas estratégias para eliminar progressivamente a energia fóssil e nuclear. Um número crescente de países alcançou uma participação de dois dígitos na produção de energia eólica.(WWEA, 2017)

Países da Europa e, mais recentemente, da América Latina, representada em sua maioria pelo Brasil, surgem desempenhando um papel assaz importante nesse desenvolvimento dinâmico. De-

vido à sua capacidade de geração, principalmente na região Nordeste, o Brasil permeia entre países desenvolvidos e que possuem matrizes energéticas bastante diferenciadas, quando comparadas com a brasileira, fundamentada em hidroelétricas, conforme gráfico da Figura 1.1.

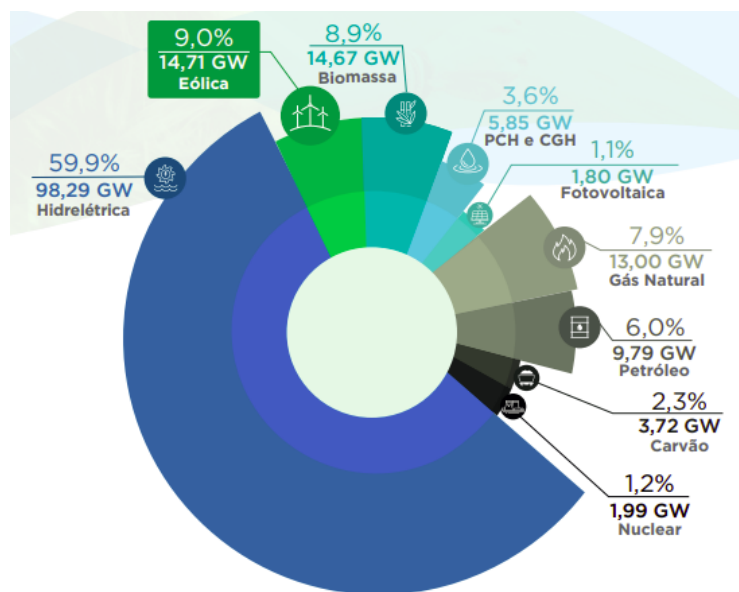


Figura 1.1: Composição da matriz energética Brasileira (ABEEÓLICA, 2019).

Ainda segundo (ABEEÓLICA, 2019) a capacidade instalada de 14,71 GW, referente à 9% da geração Brasileira é formada por 14,44 GW provenientes de parques em operação comercial, e os restantes 0,27 GW provenientes de simulações e testes. Ressalta-se ainda que não existe mais nenhum parque apto a operar, já que os últimos 98,7 MW nessa condição obtiveram disponibilidade de margem de escoamento na transmissão e passaram a operar normalmente a partir de outubro de 2018.

O crescimento do Brasil é considerável, levando-se em conta a magnitude dos investimentos. O Brasil ocupa o 8º lugar no Ranking mundial de capacidade instalada de energia eólica. Em 2012, era o 15º colocado. O gráfico da Figura 1.2 mostra o perfil da evolução em relação à capacidade instalada.

A curva da capacidade instalada da fonte eólica demonstra o crescimento virtuoso da fonte no decorrer dos anos. A composição dos dados é feita por meio da consolidação das capacidades contratadas nos ambientes de contratação livre e regulado, ACL e ACR, respectivamente. Ao final de 2024 serão 21,189 GW instalados em território brasileiro.(ABEEÓLICA, 2019)

De toda essa energia instalada e produzida no Brasil, tem-se que grande parte é de responsabilidade da região Nordeste. O Nordeste responde por 86% da produção de energia eólica no país, segundo dados da Associação Brasileira de Energia Eólica (ABEEólica). O estado do Piauí está entre os maiores produtores de energia limpa, que ocupa o quinto lugar entre os maiores produtores de energia eólica e é o segundo maior produtor de energia solar do Brasil.(CEBDS, 2015)

Esse modelo de distribuição, que prioriza a região Nordeste, ocorre devido aos requisitos ne-

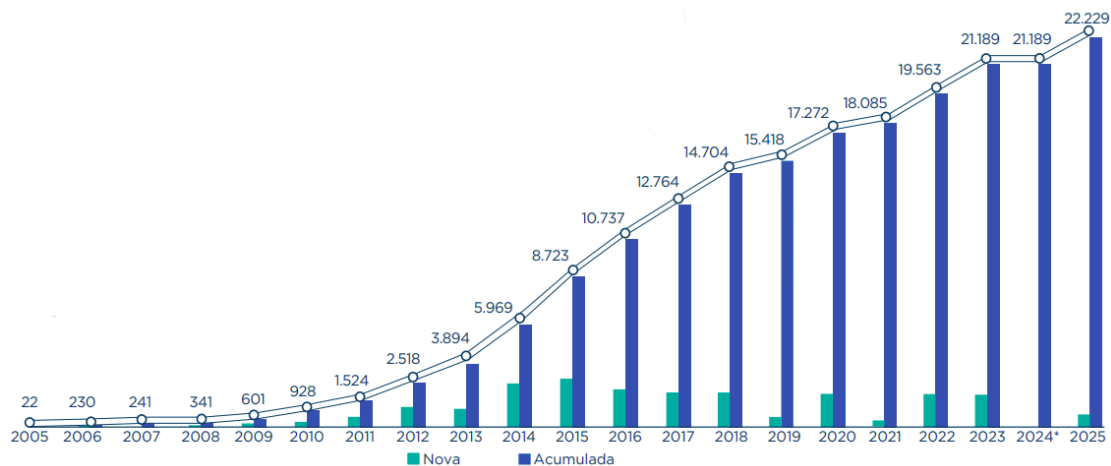


Figura 1.2: Evolução da Capacidade Instalada no Brasil, em MW (ABEEÓLICA, 2019).

cessários para a geração de energia elétrica a partir da energia do vento. Desses requisitos tem-se que as condições ideais de vento para a geração de energia eólica deve ser unidirecional, constante e de velocidade estável, características presentes nos Estados mostrados na Figura 1.3.

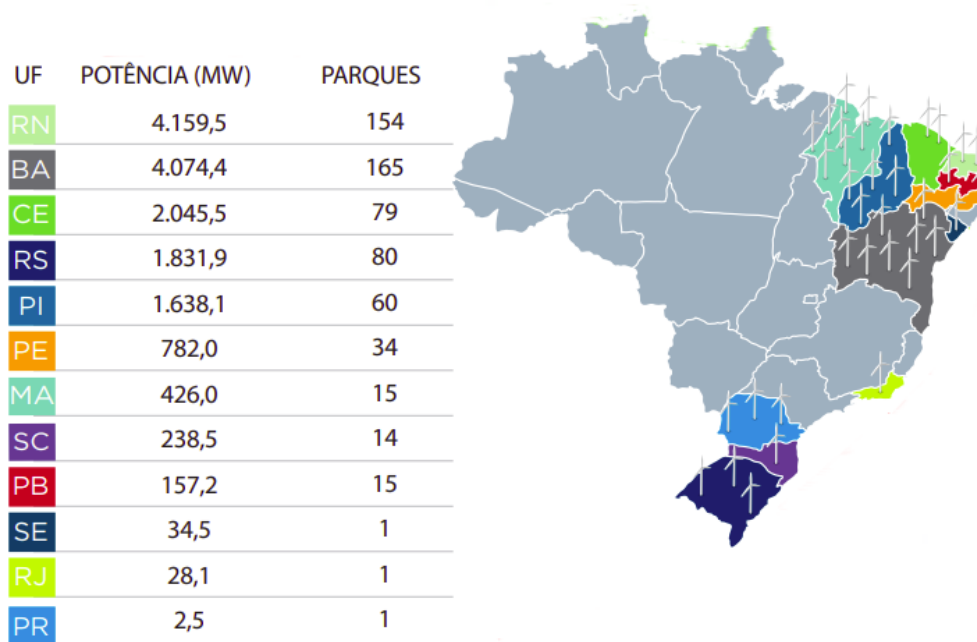


Figura 1.3: Capacidade e número de parques eólicos por estado (ABEEÓLICA, 2019).

Operando em 12 Estados Brasileiros, com ênfase para os estados da Bahia e do Rio Grande do Norte, e com capacidade instalada de 15,4 GW em 619 parques eólicos, usando 7578 aerogeradores, a indústria dos ventos mostra seu tamanho e principalmente seu grande potencial de crescimento no país.

Considerando leilões já realizados e contratos firmados no mercado livre, o Brasil terá cerca de

21,189 GW de capacidade eólica instalada até 2023. Cada MW gerado corresponde à 15 postos de trabalho e 21 milhões de toneladas de CO_2 foram evitadas em 2018, equivalente à emissão anual de CO_2 de aproximadamente 16 milhões de automóveis(ABEEÓLICA, 2019).

1.2 Definição do problema

Conforme visto anteriormente, por meio de dados de instituições especializadas em energia eólica, alguns estados brasileiros não possuem os requisitos básicos para a implantação desse tipo de geração, dentre esses estados encontra-se o Distrito Federal. Por se tratar de uma região pouco propensa à implantação de turbinas eólicas, à título de pesquisa e levantamento de dados, tem-se como solução desenvolver formas alternativas, que possibilitem simular, da melhor maneira possível, especificidades e características da geração eólica.

Por meio de um motor de indução trifásico e um inversor de frequência, as particularidades existentes em um aerogerador podem ser replicadas e analisadas no próprio laboratório. O inversor de frequência possibilita, por meio de seus parâmetros, a implementação de formas de controle que simulem o modo de operação de turbinas eólicas reais.

Utilizando-se desses parâmetros é possível estabelecer variações de velocidade no motor, tornando possível replicar perfis de vento e características comuns, como rajada, brisa e *ripple*, vento cuja velocidade se comporta como uma onda. A parametrização do inversor é possibilitada pelo protocolo de comunicação ModBus, por meio da biblioteca do Python MinimalModbus, que permite a troca de mensagens via comunicação serial. A parametrização do inversor deverá ser facilitada por meio de uma interface de usuário, possibilitando uma integração mais clara entre mestre e escravo.

1.3 Objetivos do projeto

Esse projeto tem como objetivo principal a emulação de uma turbina eólica por meio de um motor de indução trifásico, acionado por um inversor de frequência. O controle do motor será realizado por um inversor de frequência programado via comunicação serial. Os dados com os perfis de velocidade do vento serão a base que indicará a forma de giro do motor. Esses dados serão parametrizados no inversor de frequência via uma interface de usuário, que também deverá garantir a comunicação entre mestre e escravo.

Dados como velocidade do vento, serão inseridos em uma Interface Homem máquina (IHM) e serão submetidos ao inversor de frequência. Os perfis de velocidade do vento e rotação do motor estarão disponíveis ao usuário na forma de gráficos, que poderão ser salvos utilizando a interface do usuário. A interface terá como característica principal a capacidade de receber arquivos com dados coletados por instituições que realizam medições, e então convertê-los para informações relacionadas ao modo de giro do motor.

Capítulo 2

Referencial Teórico

2.1 Energia Eólica

O aquecimento de forma desigual da superfície terrestre é o responsável pela formação dos ventos. A inclinação existente entre o eixo de rotação da Terra em relação ao Sol ocasiona uma menor intensidade de radiação nos polos na região equatorial. Esses fatores, em conjunto com as características de cada superfície, causam movimentação do ar (DUTRA, 2019), conforme mostra a Figura 2.1

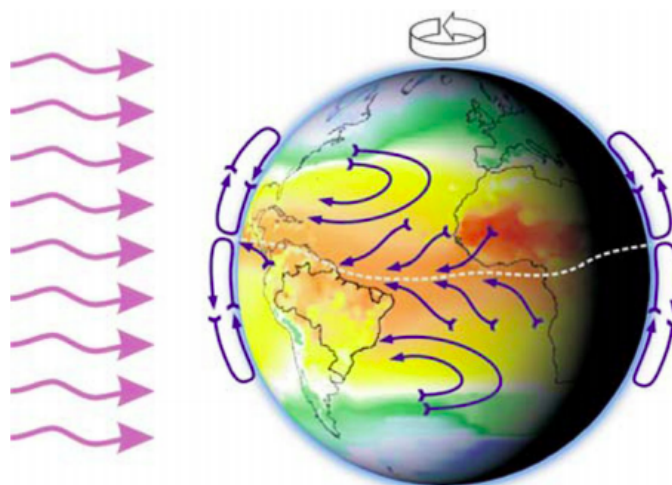


Figura 2.1: Formação dos ventos devidos ao deslocamento das massas de ar (DUTRA, 2019).

Ainda conforme (DUTRA, 2019), o uso dos ventos para beneficiamento da energia elétrica também apresenta, como a maioria das tecnologias energéticas, características ambientais desfavoráveis como impacto visual, ruídos, interferências eletromagnéticas. Mas, por meio de um planejamento adequado e com o uso de tecnologia esses aspectos negativos podem ser minimizados suficientemente. A não necessidade do uso de água como elemento motriz traz um destaque a esse tipo de energia. O fato de não produzir resíduos gasosos ou radioativos e permitir a utilização de 99% das áreas usadas para parques eólicos para outros fins, como agricultura ou pecuária,

também são pontos a favor da energia eólica.

Segundo (PINTO, 2013), no reinado de califa Umar que durou de (633-644), moinhos de vento foram construídos com o intuito de se obter água para usar na irrigação e para a moagem de grãos. A história diz que o conquistador mongol Gêngis Khan (1162-1227) importava moinhos do Oriente Médio para a China, sendo assim a primeira referência de um moinho de vento chinês ocorrida no ano de 1219. Na Europa, a data de 1105 marca a primeira referência à utilização de moinhos de vento, na França. Algo bastante semelhante, datado de 1180 refere-se à Inglaterra.

A Figura 2.2, faz referência aos moinhos de vento e mostra a evolução dos aerogeradores de grande porte. O autor (SILVA, 2014) afirma que os primeiros aerogeradores são alemães. Por volta dos anos 60, o construtor Allgaier, com o intuito de fornecer energia, construiu aerogeradores de pequeno porte com 5 metros de raio e aproximadamente 7 kW. Anos depois mais um avanço surgiu, o dinamarquês Riisager projetou e construiu um aerogerador maior e com uma capacidade de geração de 22 kW.

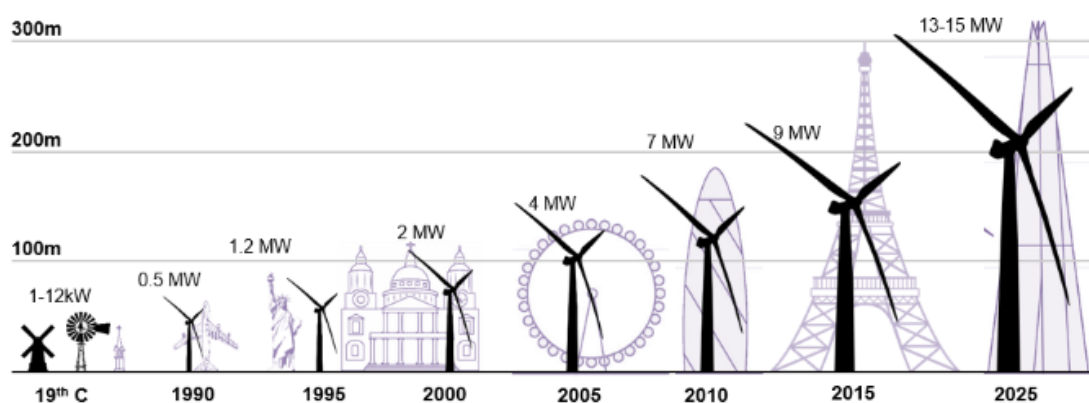


Figura 2.2: Evolução dos Aerogeradores (REIS, 2016).

2.1.1 A potência do vento

A potência do vento pode ser dada em função da altitude e da temperatura ambiente ou em função da sua velocidade. Para esse trabalho, a potência em função da velocidade do vento será de suma importância. Deduzida por (CUSTODIO, 2013), tem-se que a potência dissipada no vento é proporcional ao cubo da velocidade do vento, dada por:

$$P/A = \frac{1}{2}\rho v^3, \quad (2.1)$$

onde:

- P : potência do vento (em W);
- A : área da seção transversal (em m^2);

- ρ : densidade do ar (em kg/m^3);
- v : velocidade do vento (em m/s).

2.2 Modelos de turbinas eólicas

É por meio das turbinas que a energia cinética do vento é convertida em outra modalidade de energia, por meio do giro das pás do rotor. O gerador se aproveita do giro do rotor e transforma essa energia mecânica em elétrica. São instalados na camada superficial da atmosfera de modo a utilizar, em ampla faixa de altura, a energia do vento (HERBERT; INIYAN; RAJAPADIAN, 2007).

Conforme (CRESESB, 2014), os tipos de turbinas usados para geração de energia elétrica são definidos como os que apresentam rotores de eixo vertical e os com rotores de eixo horizontal. A Figura 2.6 mostra os tipos de turbinas existentes. As turbinas de eixo vertical não apresentam mecanismos direcionais e possuem como vantagem o fato de o gerador e as transmissões serem instalados no solo. As Figuras 2.3(a), 2.3(b) e 2.3(c) são exemplos desse tipo de eixo.

Ainda é possível definir cada modelo:

- **GIROMILL**: As turbinas do tipo Giromill são bastante parecidas com o modelo Darrieus, pois possuem lâminas verticais e que rotacionam em seu próprio eixo, fazendo com que o ângulo de ataque do vento seja contínuo, ajudando no processo para se obter mais eficiência no sistema. Porém, esse modelo de turbina necessita de velocidade de vento maior, de modo que ela comece a rotacionar, ou seja, ela tem um alto torque de partida e mais eficiência em ventos turbulentos (KOMORI; FREITAS; PASSOS, 2017);
- **SAVONIUS**: Turbina com torque de partida médio ou alto, para motores do tipo compostos. Possui velocidade de rotor baixa e normalmente é utilizada em processos de amento e moagem. Seu torque pode ser controlado por meio da regulação da passagem de ar entre as pás. Utiliza rotor não estritamente de arraste, mas com característica de pá com grande área de interceptação (CUSTODIO, 2013);
- **DARRIEUS**: Assim como os modelos Savonius e Giromill, não necessitam de mecanismos direcionais. Possui velocidade de rotor alta, possibilitando assim sua utilização na produção de energia elétrica. Nesse modelo o gerador é instalado no solo, facilitando então sua montagem e manutenção (CUSTODIO, 2013);
- **Eixo Horizontal**: Tipo de turbina mais usado atualmente, principalmente em instalações de maior potencial de produção, como os parques eólicos. Possuem como característica principal a capacidade de se manter perpendicularmente à direção do vento, de forma a possibilitar a máxima captura de energia. O gerador é instalado juntamente com a turbina (CUSTODIO, 2013).



(a) GIROMILL (eixo vertical).



(b) SAVONIUS (eixo vertical).



(c) DARRIEUS (eixo vertical).



(d) Eixo horizontal.

Figura 2.3: Modelos de turbinas eólicas (STROSK, 2019).

2.3 Turbina Eixo Horizontal

Por se tratar de um projeto que simula as condições de turbinas em relação à velocidade do vento, o tipo de turbina utilizado nesse trabalho será de Eixo Horizontal, por serem utilizadas em larga escala na produção de energia elétrica.

Segundo (CASTRO, 2009), a grande maioria das turbinas utilizadas em parques eólicos são constituídas por três pás. A energia capturada ao vento é de 3% a 5% maior quando se passa de duas para três pás, mas conseqüentemente esta percentagem diminui à medida que o número de pás é aumentado. Esse fato se deve à maior estabilidade e aerodinâmica proporcionada pelo projeto composto por três pás.

A explicação dada por (CUSTODIO, 2013) para que o número de pás não seja muito grande é que a velocidade do rotor é diminuída por conta de uma pá estar ocupando um espaço já ocupado por outra, aproveitando-se assim de um ar perturbado. No entanto, um número maior de pás aumenta o torque sobre o eixo do rotor. Então, quanto menor o número de hélices, mais rápido o mesmo irá gira. O rotor ideal para aplicações energéticas é o composto por três pás, por possuir

movimento suave e estável, com balanceamento facilitado, menor vibração e redução de ruídos. A Figura 2.4 mostra os detalhes de uma turbina usada atualmente, com rotor de três hélices.

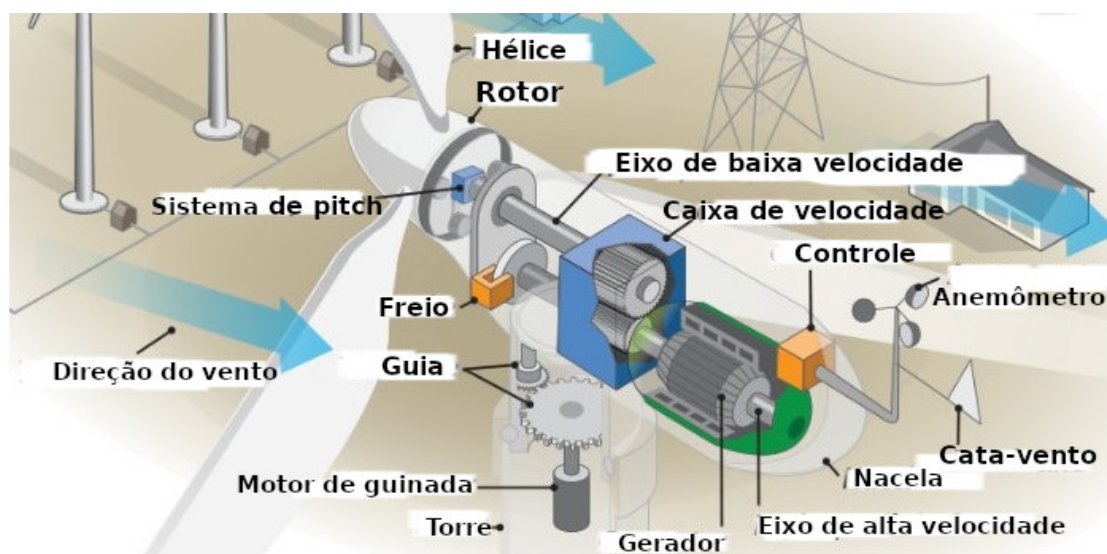


Figura 2.4: Esquema básico e componentes constituintes de uma turbina (REIS, 2016).

2.3.1 A potência de uma turbina ideal

Conforme Bernoulli, a vazão de um fluido é constante para diferentes pontos ao longo de um tubo de vazão. A Figura 2.5 mostra uma ilustração do modelo de fluxo de ar que passa através de uma turbina eólica de eixo horizontal com três pás, em que:

- V_1 : velocidade do vento antes da turbina (em m/s);
- V' : velocidade do vento na seção do tubo de vazão na entrada da turbina (em m/s);
- V_2 : velocidade do vento na seção do tubo de vazão na saída da turbina (em m/s);
- A_1 : área da seção transversal na entrada do tubo (em m^2);
- A_2 : área da seção transversal na saída do tubo (em m^2).

Em seu livro (CUSTODIO, 2013), utilizando características descritas pela Figura 2.5, mostra que existe uma potência máxima que pode ser extraída do vento. Esse valor é conhecido como Máximo de Betz, ou coeficiente de Betz, e diz que:

$$P_{tm\acute{a}x} = \frac{16}{27} \left(\frac{1}{2} \rho A v^3 \right) = \frac{16}{27} P, \quad (2.2)$$

onde:

- $P_{tm\acute{a}x}$: potência máxima extraída do vento por uma turbina ideal (em W);

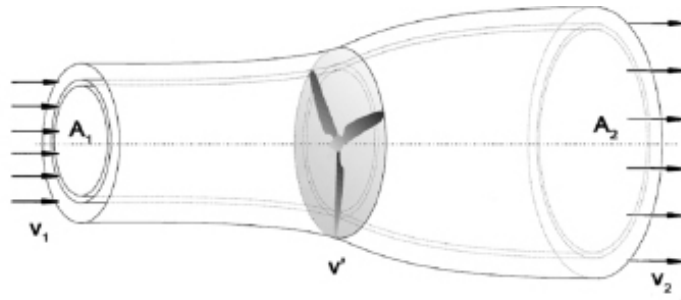


Figura 2.5: Fluxo de vento por uma turbina eólica (REIS, 2016).

- A : área da seção transversal (em m^2);
- ρ : densidade do ar (em kg/m^3);
- v : velocidade do vento (em m/s);
- P : potência disponível no vento (em W), (equação 2.1).

Então, com base nas (equações 2.1 e 2.2), é possível perceber que uma turbina eólica pode extrair no máximo $16/27$ da potência disponível no vento, ou seja, $59,3\%$.

2.3.2 A potência de uma turbina real

Ainda segundo (CUSTODIO, 2013), a potência real de uma turbina eólica será somente uma parte do valor máximo de Betz, isso ocorre por conta de perdas aerodinâmicas ocorridas na conversão da energia. Existe um coeficiente de potência que indica a relação entre a potência real extraída do vento por uma turbina e a potência disponível no vento. Sendo assim:

$$C_p = \frac{P_t}{\frac{1}{2}\rho Av^3}, \quad (2.3)$$

- C_p : Coeficiente de potência (adimensional);
- P_t : potência de uma turbina real (em W);
- ρ : densidade do ar (em kg/m^3);
- v : velocidade do vento (em m/s);
- A : área da seção transversal (em m^2).

Um valor que maximiza C_p pode ser obtido usando estratégias de controle na própria turbina, de modo que a leitura da velocidade do vento permita um ângulo de ajuste que potencialize C_p .

A partir da (equação 2.3), que define o coeficiente de potência C_p , tem-se que a potência de uma turbina eólica real é dada por:

$$P_t = \frac{1}{2} C_p \rho A v^3 \quad (2.4)$$

A potência que é fornecida por uma turbina varia com o cubo da velocidade do vento.

O gráfico da Figura 2.6 mostra as curvas que relacionam a potência disponível no vento, dada pela (equação 2.1), com a potência extraída por uma turbina eólica ideal (equação 2.2) e a potência extraída por uma turbina eólica real (equação 2.4).

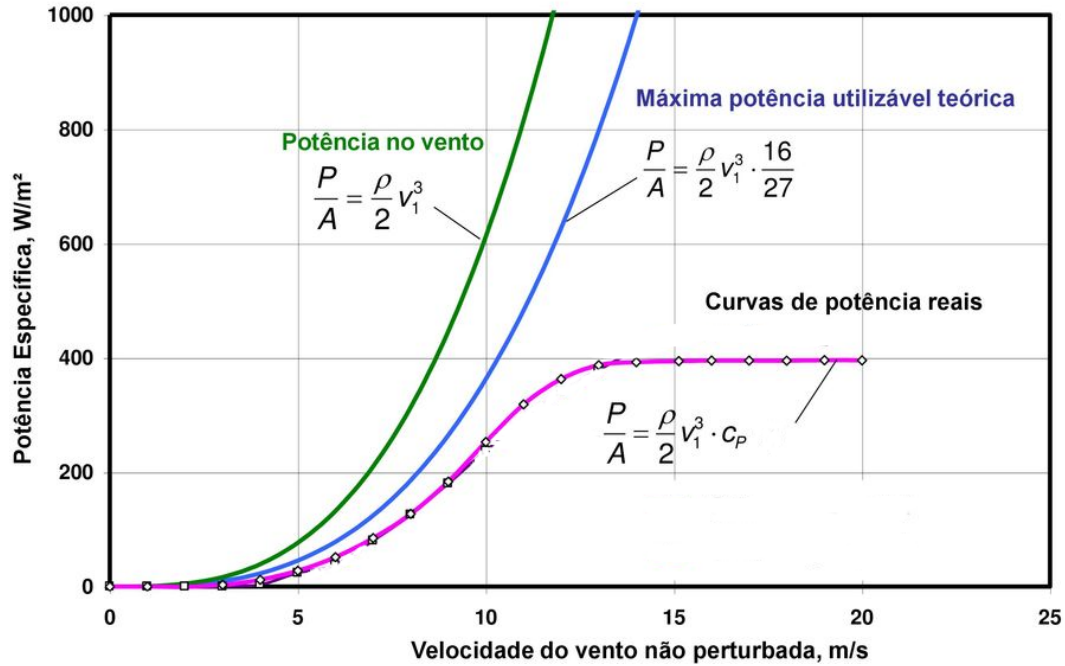


Figura 2.6: Curvas relacionando a potência disponível no vento com as potências real e teórica extraídas pela turbina (RICO, 2017).

2.3.3 Velocidade de ponta

A relação entre a velocidade tangencial na ponta da pá do rotor e a velocidade do vento que incide na turbina é chamada de velocidade de ponta (CUSTODIO, 2013), definida por:

$$\lambda = \frac{u}{v} = \frac{\omega R}{v}, \quad (2.5)$$

onde:

- λ : velocidade de ponta (adimensional);
- u : velocidade tangencial na ponta da pá (em m/s);
- v : velocidade do vento (em m/s);
- R : raio do rotor da turbina (em m);

- ω : velocidade angular da pá da turbina (em rad/s).

O valor de ω , que é a velocidade angular é dado pela (equação 2.6), onde n é a rotação da turbina em RPM (Rotações por minuto).

$$\omega = \frac{2\pi n}{60} \quad (2.6)$$

Relacionando as equações 2.5 e 2.6 é possível se obter uma relação importante que permite transformar a velocidade do vento incidente em uma turbina qualquer, em velocidade de rotação do motor. A equação fica na forma:

$$n = \frac{\lambda v 60}{2\pi R} \quad (2.7)$$

Pela equação 2.7 é possível se obter, somente com as características das pás da turbina e a velocidade do vento, a velocidade de rotação do rotor, dada em RPM. O valor de λ se mantém constante para turbinas com velocidade variável e a rotação do rotor varia com a velocidade do vento (CUSTODIO, 2013).

Em (RAGHEB, 2018) se deduz um valor ótimo para λ . A proporção ótima de velocidade da ponta depende do número de pás do rotor e do vento incidente na turbina. Quanto menor o número de pás, mais rápido a turbina eólica precisa girar para se obter a máxima energia disponível no vento. Então:

$$\lambda_{opt} \approx \frac{4\pi}{n_p}, \quad (2.8)$$

em que n_p é o número de pás do rotor da turbina.

2.4 Inversor de Frequência

O inversor de frequência é um instrumento capaz de gerar um conjunto de tensões trifásicas com flexibilidade no ajuste das tensões eficazes e da frequência, usado para o controle de velocidade dos motores elétricos de indução trifásicos. De acordo com suas características construtivas, podem ser alimentados usando uma rede elétrica monofásica ou trifásica.

Normalmente são programados para os valores máximo e mínimo de frequência de saída, conforme a aplicação. A variação da frequência em que o motor pode ser controlado de modo a prover um ajuste contínuo de velocidade é selecionado com relação à carga utilizada. Os inversores de frequência são equipamentos com a capacidade de operar em vários formatos de controle, mudando suas próprias configurações. Por serem versáteis e por atenderem a diversos requisitos são muito utilizados na indústria (WEG, 2009).

O inversor recebe alimentação tipo corrente alternada (CA), por meio de um retificador converte-a para contínua, manipula a frequência e depois, por meio de um PWM (*Pulse Width Modulation*), gera tensões para a excitação do motor, pelo qual a frequência pode ser ajustada.

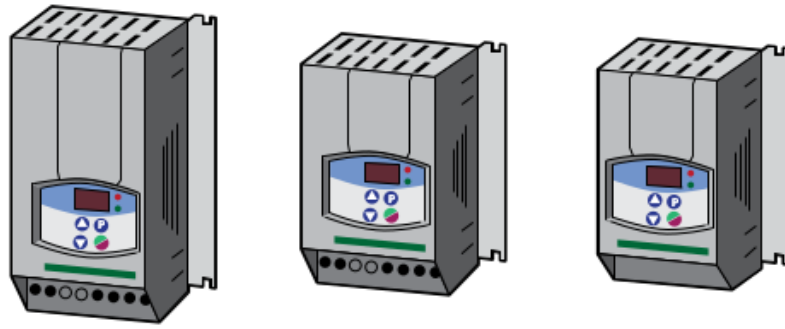


Figura 2.7: Inversores de frequência (WEG, 2009).

Os componentes de um inversor de frequência podem ser vistos na Figura 2.8. São eles: CPU (*Central Process Unit*), IHM (Interface Homem Máquina), Interfaces e Sistema de potência.

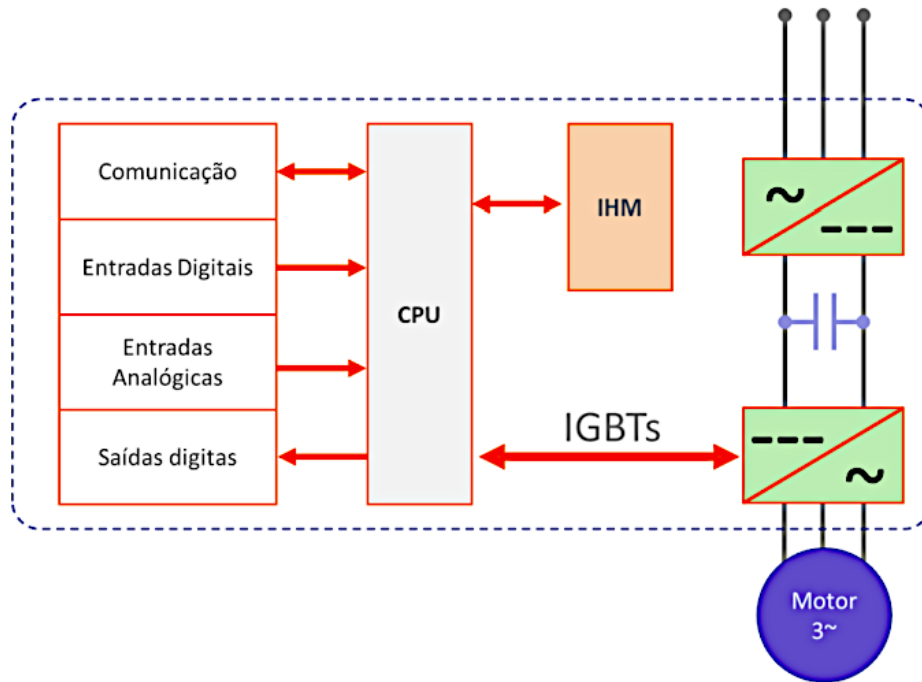


Figura 2.8: Componentes do inversor de frequência em blocos (FRANCHI, 2013).

O autor (FRANCHI, 2013) define cada um deles conforme os seguintes tópicos.

2.4.1 Unidade Central de Processamento (CPU)

A formação da CPU de um inversor de frequência varia de fabricante para fabricante. Alguns utilizam um microprocessador, outros utilizam um microcontrolador. Independentemente do fabricante é nessa unidade que todos os dados, como parâmetros e informações do sistema, estão armazenados, existindo também uma memória integrada. Além de armazenar dados e informa-

ções importantes para o funcionamento do equipamento, a CPU desempenha outro papel muito importante que é o de gerar pulsos de disparo, o que ocorre por meio de uma lógica de controle usada para disparar os IGBT's (*Insulated Gate Bipolar Transistor*).

2.4.2 Interface Homem/Máquina (IHM)

Dispositivo disponível no inversor de frequência com a finalidade de possibilitar a interação entre o usuário e o equipamento. Utilizado para parametrização e visualização do que ocorre durante o funcionamento. Possibilita, por meio dos parâmetros de leitura, a visualização de diferentes tipos de grandeza do motor, como corrente, tensão, frequência, códigos de alarme, entre outras. Permite também, com os parâmetros de escrita, ligar e desligar, inverter o sentido de giro, variar a velocidade da máquina. A Figura 2.9 mostra a IHM do inversor de frequência Weg CFW-11.



Figura 2.9: Interface Homem/Máquina CFW-11 (WEG, 2016).

2.4.3 Interfaces

Grande parte dos inversores de frequência podem ser configurados por meio de sinais analógicos ou digitais. A interface analógica é comumente mais usada, por possibilitar a inversão do sentido de rotação, porém as entradas digitais também são importantes e podem ser habilitadas por meio de parâmetros de programação.

2.4.4 Sistema de potência

Esse sistema é formado basicamente por um retificador, um filtro capacitivo e um inversor. A alimentação pode ser monofásica em 220V, com saída trifásica 220V ou 380V. A Figura 2.10 mostra um inversor de frequência com alimentação de três fases R, S e T.

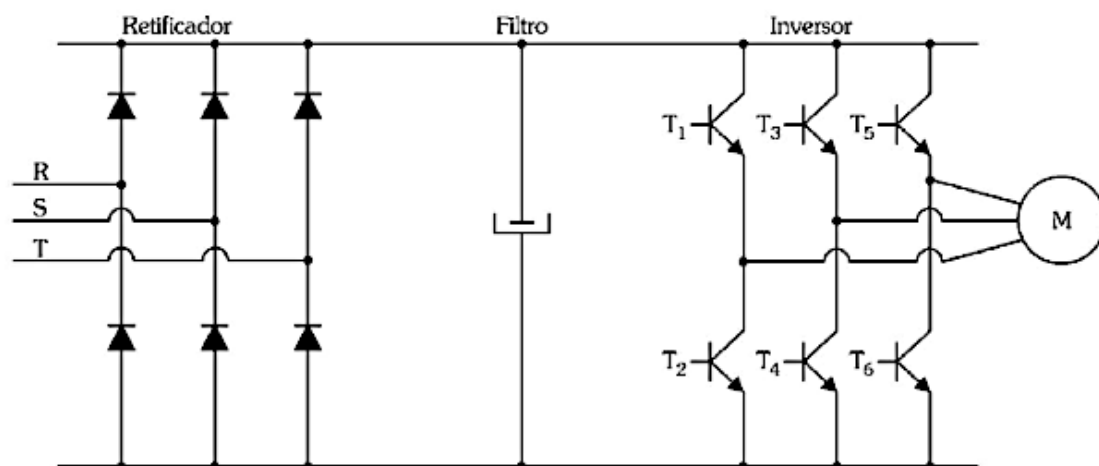


Figura 2.10: Diagrama de um inversor de frequência com alimentação trifásica (FRANCHI, 2013).

Circuito retificador e filtro

O retificador é um conversor com a função de transformar corrente alternada em corrente contínua (CC). É o responsável pela retificação do sinal alternado da rede de alimentação, que possui frequência e tensão constantes (FRANCHI, 2013). A Figura 2.11 ilustra o circuito de retificação recebendo alimentação alternada trifásica.

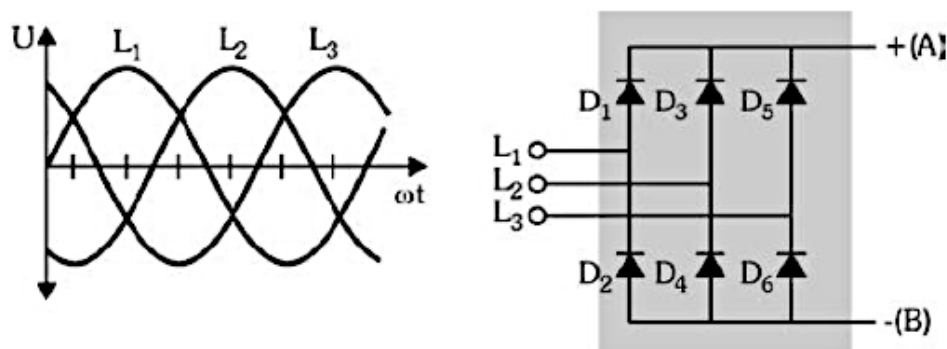


Figura 2.11: Fonte de tensão alternada aplicada a um retificador (FRANCHI, 2013).

O circuito de retificação utiliza diodos, que são componentes semicondutores que permitem a passagem de corrente em apenas um sentido, do ânodo para o cátodo. Quando se tem corrente alternada sobre um diodo, ocorre sua conversão em corrente contínua pulsante, sendo assim, tem-se uma forma de onda retificada pulsante.

A oscilação desse sinal é reduzida utilizando-se um filtro capacitivo. Esse circuito de filtro tem a capacidade de formar uma fonte de corrente contínua simétrica devido a um ponto de terra em relação à referencio denominada barramento CC. Isso ocorre pela regulação da tensão retificada com o armazenamento de energia pelos capacitores.

Circuito inversor

O circuito do inversor é formado basicamente por transistores, controlados por meio de uma unidade de controle. De acordo com a Figura 2.10, a tensão contínua é ligada à saída por meio dos transistores T_1 a T_6 , que por sua vez funcionam no corte ou na saturação, por serem chaves estáticas. Um modelo de inversor monofásico é utilizado para a compreensão do funcionamento do circuito Figura 2.12

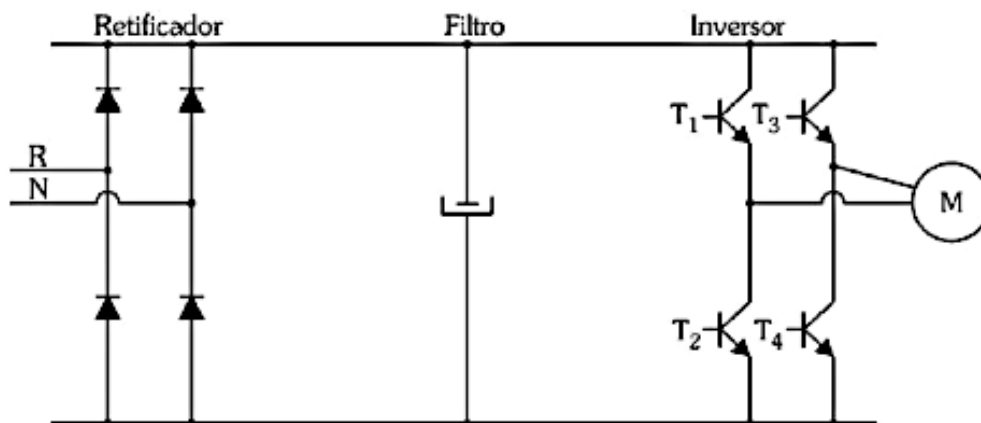
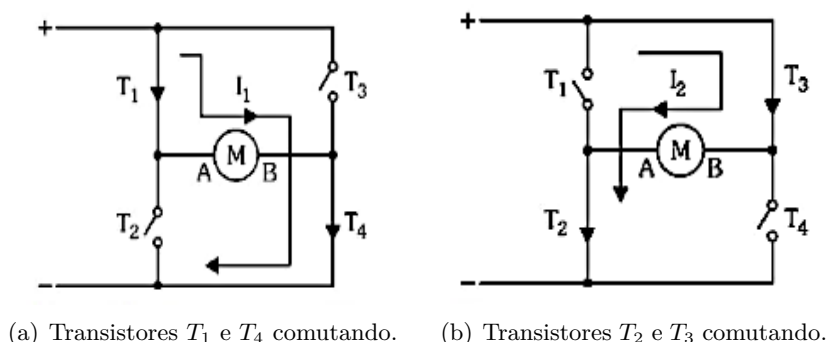


Figura 2.12: Diagrama de um inversor de frequência alimentação monofásica (FRANCHI, 2013).

A conversão CC-CA é realizada por meio de uma lógica de controle que aciona os transistores sempre em pares. Com base no circuito da Figura 2.12, tem-se que em certo momento os transistores T_1 e T_4 estão ligados, enquanto T_2 e T_3 desligados. Nesse momento a corrente circula de A para B, conforme a Figura 2.13(a).

Em seguida ocorre o contrário, os transistores T_1 e T_4 serão desligados e T_2 e T_3 ligados e a corrente circulando de B para A, como mostra a Figura 2.13(b). Baseada nessa sequência de acionamento, ocorre a conversão CC-CA, por meio da inversão no sentido da corrente.



(a) Transistores T_1 e T_4 comutando. (b) Transistores T_2 e T_3 comutando.

Figura 2.13: Lógica de controle de transistores, para inversor monofásico (FRANCHI, 2013).

A frequência do sinal modulador define a velocidade de rotação do motor, por meio da modu-

lação da portadora de alta frequência, como pode ser visto na Figura 2.14, que mostra o exemplo com frequência de chaveamento de 60 Hz e de 30 Hz. A frequência de chaveamento está relacionada com a frequência da portadora, ou seja, do próprio sinal triangular PWM.

O modo de operação dos transistores, trabalhando como chaves, gera uma forma de onda quadrada, sendo assim também a forma da saída do inversor. O motor necessita de uma tensão de entrada senoidal, ou o mais próximo possível disso, então é necessário chavear os transistores modulando a largura de seus pulsos.

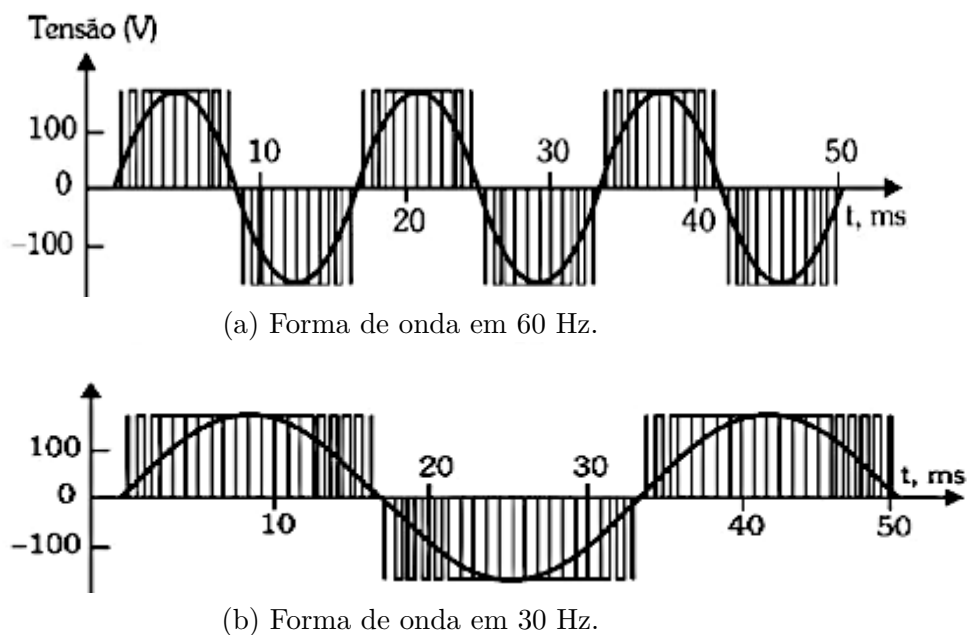


Figura 2.14: Frequência de chaveamento (FRANCHI, 2013).

Modulação por largura de pulso

As funções de um PWM são realizadas através de um microcontrolador. Existem várias técnicas de modulação, porém a mais utilizada em inversores de frequência é a que combina uma onda triangular com uma senoidal e produz a forma de onda da tensão na saída. O sinal triangular diz respeito à frequência de chaveamento, já o sinal senoidal determina a largura dos pulsos, funcionando como uma referência (FRANCHI, 2013). A Figura 2.15 mostra a saída de um gerador PWM.

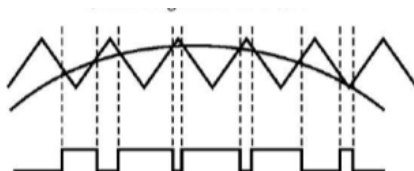


Figura 2.15: Sinal de saída de um PWM (FRANCHI, 2013).

Um circuito controla eletronicamente a tensão e a frequência, a tensão CC é manipulada para produzir tensão e frequência variáveis. A Figura 2.16 mostra, de maneira mais clara as três fases sendo geradas por meio dos IGBT's, controlados pelo PWM e defasadas em 120 °.

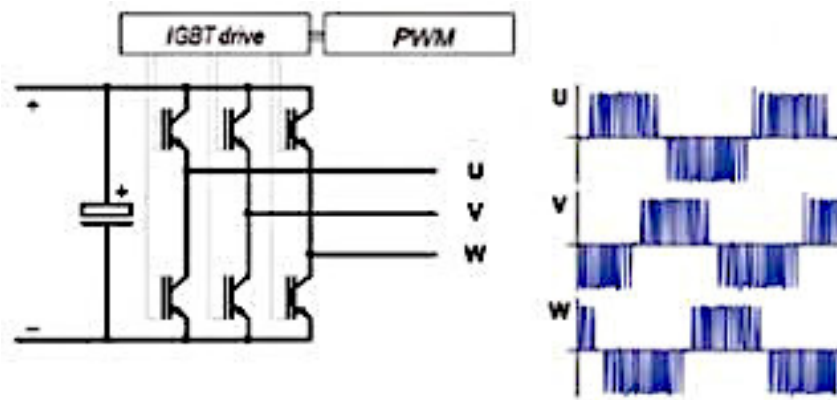


Figura 2.16: Sinal de saída inversor de frequência (TOSSI, 2011).

As três fases da saída do inversor de frequência, com defasagem de 120° , podem ser vistas com mais clareza na Figura 2.17.

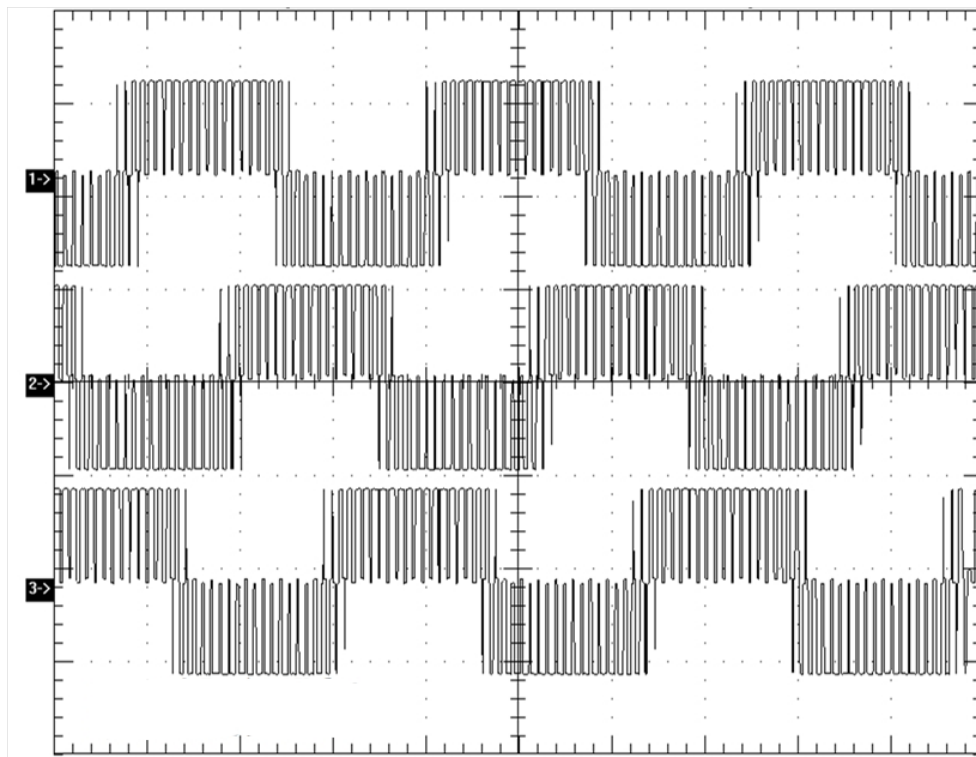


Figura 2.17: Sinal de saída inversor de frequência (ELECTRONICS, 2020).

2.4.5 Parâmetros

Existem os parâmetros de leitura de comunicação e de regulação, ou escrita. Estes parâmetros são utilizados na configuração da fonte de comandos para os modos local e remoto do inversor CFW-11 e para monitoramento do funcionamento do conjunto inversor de frequência, motor de indução trifásico. No manual, os parâmetros são classificados como:

- **RO:** parâmetro somente de leitura;
- **CFG:** parâmetro somente alterado com o motor parado;
- **Net:** parâmetro visível via IHM se o inversor possuir interface de rede instalada – RS232, RS485, CAN, *Anybus-CC*, *Profibus* – ou se a interface USB for conectada;
- **Serial:** parâmetro visível na IHM se o inversor possuir interface RS232 ou RS485 instalada;
- **USB:** parâmetro visível por meio da IHM se a interface USB do inversor for conectada.

Alguns desses parâmetros só podem ser habilitados com a presença de um determinado tipo de comunicação presente no inversor, como no caso de interface RS232 e RS485. Outro parâmetro importante, e que torna o inversor funcional é o P0000 que deve ser configurado com o valor 5, isso para que os demais parâmetros sejam acessíveis. O comando JOG é a associação do comando Gira/Para com uma referência de velocidade via parâmetro, a Tabela 2.1 mostra alguns parâmetros e sua descrição.

2.4.6 Formas de controle

O inversor tem como principal característica o modo como ocorre a alimentação do motor, na forma de tensão, corrente e frequência variáveis, sendo possível então se obter o controle da velocidade do motor com base nesses parâmetros. Os valores que se aplicam ao motor são baseados em uma estratégia de configuração que depende do tipo de controle selecionado e dos ajustes aplicados ao inversor por meio de seus parâmetros. Os modos de controle disponíveis, de acordo com o manual de programação do Inversor de Frequência CFW-11 (WEG, 2012) são:

- **V/f:** controle escalar (modo de controle utilizado no projeto);
- **VVW (*Voltage Vector WEG*):** controle estático de velocidade, mais preciso que o V/f;
- **Vetorial sem sensor:** controle orientado pelo campo e sem sensor de velocidade no motor;
- **Vetorial com encoder:** controle orientado pelo campo, necessita de encoder no motor;
- **Vetorial com encoder para motor PMSM:** necessita de encoder incremental no motor;
- **Vetorial sem sensor para motor PMSM:** sem sensor de velocidade no motor, controle de velocidade na faixa 1:100.

2.4.7 Controle escalar (V/f)

É o modo de controle utilizado no projeto. É um controle simples que se baseia em uma curva que relaciona tensão/frequência de saída. A frequência estabelecida no inversor determina a tensão de saída para o motor. O controle escalar é recomendado para alguns casos específicos, que são:

Tabela 2.1: Parâmetros importantes para configuração do inversor de frequência

Parâmetros	Descrição
P0000	Acesso aos Parâmetros
P0002	Leitura velocidade do motor
P0003	Leitura corrente do motor
P0005	Leitura frequência do motor
P0100	Regulação do aceleração
P0101	Regulação do tempo de desaceleração
P0105	Seleção 1a/2a Rampa
P0202	Seleção tipo de Controle
P0220	Seleção Fonte Local/Remoto
P0221	Seleção Referência Local
P0222	Seleção Referência Remota
P0223	Seleção Giro Local
P0224	Seleção Gira/Pára Local
P0225	Seleção Jog Local
P0226	Seleção Giro Remoto
P0227	Seleção Gira/Pára Remoto
P0228	Seleção Jog Remoto
P0308	Seleção Endereço Serial
P0310	Seleção da taxa de Comunicação Serial
P0311	Configuração dos bytes da interface serial
P0312	Protocolo serial, TP ou Modbus-RTU
P0313	Seleção de ação para erro de comunicação
P0133	Regulação de referência de velocidade mínima
P0134	Regulação de referência de velocidade máxima
P0136	Boost de Torque Manual
P0137	Boost de torque automático
P0138	Compensação de escorregamento
P0400	Tensão nominal motor
P0401	Corrente nominal motor
P0402	Rotação nominal motor
P0403	Frequência nominal motor
P0681	Velocidade do motor em 13 bits
P0682	Palavra de controle via Serial/USB
P0683	Referência de velocidade via Serial/ USB

- Acionamento de vários motores com o mesmo inversor (acionamento multimotor);
- A corrente nominal do motor é menor que 1/3 da corrente nominal do inversor;
- Para propósito de testes, quando o inversor é ligado sem motor ou com um motor pequeno sem carga;
- Aplicações que não exijam resposta dinâmica rápida, precisão na regulação de velocidade ou alto torque de partida.

O inversor alimenta o motor com tensão, corrente e frequência variáveis, através das quais

consegue-se o controle da velocidade do motor. Os valores aplicados ao mesmo seguem uma estratégia de configuração, a qual depende do tipo de controle selecionado e dos ajustes dos parâmetros do inversor.

O inversor opera basicamente como uma fonte de tensão, que tem a capacidade de geração de valores de frequência e tensão de acordo com a relação V/f . O controle e ajuste dessa curva é possível para motores de 50 Hz, 60 Hz ou especiais, usando o controle V/f . Uma das principais características dessa configuração é a sua simplicidade, pois carece de poucos ajustes. A colocação em funcionamento é rápida e simples e o ajuste padrão de fábrica normalmente necessita de poucas modificações.

A Figura 2.18 mostra o diagrama de blocos completo de um esquema de controle escalar. Os blocos *boost* de torque automático, de compensação de escorregamento e de referência são importantes para o funcionamento do controle no inversor de frequência. Já o *boost* de torque manual atua sob velocidades mais baixas, aumentando a tensão de saída do inversor, realizando uma certa compensação à queda de tensão que ocorre na resistência do estator do motor, a fim de manter um torque constante.

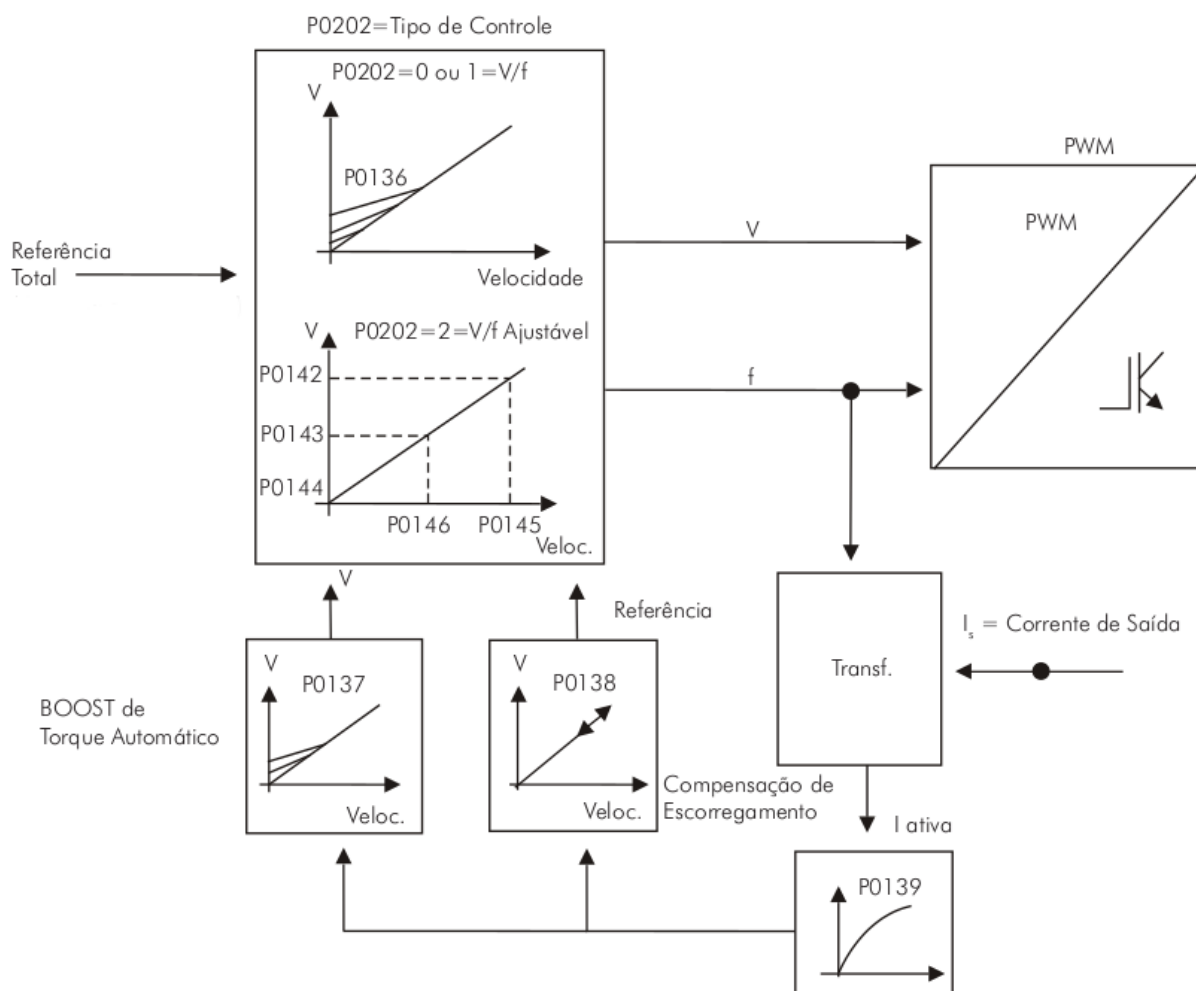


Figura 2.18: Diagrama de blocos do controle V/f (WEG, 2012).

A compensação de escorregamento compensa a queda na rotação devido a aplicação da carga no eixo e incrementa a frequência de saída em função do aumento da corrente ativa do motor. O parâmetro P0138, que efetua a compensação do escorregamento é utilizado na função de compensação de escorregamento do motor, quando ajustado para valores positivos. Neste caso, compensa a queda na rotação devido à aplicação da carga no eixo e incrementa a frequência de saída em função do aumento da corrente ativa do motor.

Para o ajuste do parâmetro P0138 é necessário primeiro acionar o motor a vazio, atingindo aproximadamente o valor equivalente à metade da faixa de velocidade de utilização e então realizar a medição da velocidade do motor ou equipamento com medidor de rotação, depois disso, aplicar uma carga nominal no equipamento e, por fim, incrementar o parâmetro P0138 até que a velocidade atinja o valor a vazio.

A Figura 2.19 mostra a curva de compensação do escorregamento de forma mais detalhada, por meio do ajuste do parâmetro P138. Valores negativos também são utilizados em aplicações especiais, onde é importante que ocorra a redução da velocidade de saída em função do aumento da corrente do motor. Um exemplo é em casos de distribuição de cargas em motores acionados em paralelo.

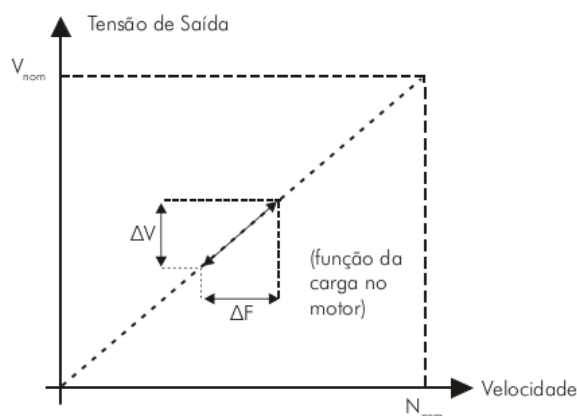


Figura 2.19: Curva V/f com compensação de escorregamento (WEG, 2012).

2.5 Comunicação Serial

A conexão de dois sistemas pode ser realizada de duas formas distintas, seja de forma paralela ou de forma serial. No caso da comunicação paralela, como o próprio nome já diz, os sistemas em questão são conectados fisicamente através de cabos conectados em paralelo, ou seja, um para cada bit de dado. Como, por exemplo, a conexão entre um computador e uma impressora, onde a proximidade física dos equipamentos e as necessidades de alta velocidade, juntamente com a quantidade de dados transferidos, se encaixam nesse modelo de transmissão.

Em casos de sistemas que se encontram fisicamente distantes, a conexão por fios em paralelo pode não ser tão viável. Nesses casos, a conexão do tipo serial é a mais recomendada, que é o caso das conexões via modem, em que as informações são enviadas ou recebidas bit a bit, por meio de

um mecanismo de comunicação mais robusto.

Na transmissão e comunicação serial, que é a mais aplicável nesse caso, podem ocorrer basicamente duas formas: da maneira síncrona e assíncrona. A transmissão síncrona é não somente o transporte dos bits de dados de forma serial, mas também caracteres de sincronismo, os quais, depois de serem detectados no receptor, ajustam seus clock's internos e se tornam aptos a receberem os bits de dados em uma determinada taxa de comunicação.

Na comunicação assíncrona cada caractere é transmitido de forma individual, e para cada um desses caracteres existem os bits de início *Start Bits* e os bits de parada *Stop bits*. As taxas de comunicação devem ser definidas de maneira prévia entre os elementos comunicantes (HIRAKAWA; CUGNASCA, 2014).

Segundo (ENGINEERS, 2020), os pontos de maior importância e que precisam de mais atenção na comunicação serial são a taxa de transmissão, os bits de dados, os bits de parada e a paridade.

2.5.1 Taxa de transmissão

A velocidade com que ocorre o tráfego ou transmissão das informações deve ser rápida, de tal modo que consiga conciliar as capacidades, tanto do receptor, quanto do emissor. Surge então a taxa de transmissão, que é necessariamente o número de mudanças de sinal que ocorre por segundo, ou seja, é a taxa de modulação que ocorre em determinado meio de transmissão, também conhecida como *baud rate*. A taxa de bits por segundo em que determinado dado é transmitido e recebido deve ocorrer da mesma forma para recepção e transmissão (BEZARRA, 2019).

Outro ponto importante é a dependência dos dispositivos envolvidos e a taxa de transmissão, ou seja, a taxa de transmissão entre dois depende de vários fatores, tais como as características dos condutores utilizados, o modo de utilização do canal para transmissão, a capacidade de transmissão máxima dos dispositivos e etc.

2.5.2 Bits de dados

Os bits de dados correspondem efetivamente a quantidade de dados a serem transmitidos, normalmente essa quantidade é de 8 bits, mas pode acontecer de ser 5 ou 7.

2.5.3 Bits de parada

Esses bits são caracterizados pela funcionalidade de indicar, ou marcar, o fim de uma determinada mensagem, ou até mesmo o fim de uma transmissão.

2.5.4 Paridade

A paridade normalmente é utilizada para controlar e caracterizar os erros na transmissão e nas trocas de mensagens. A transmissão pode ser definida como sendo com paridade par ou ímpar, por

meio da quantidade de números "1" da palavra mais o bit de paridade. Esse número de paridade é gerado como sendo zero ou um, de modo a garantir a paridade definida.

Um exemplo, segundo (MECAWEB, 2019), é que se ocorra a transmissão do caractere "A" cujo valor é "65", ou seja, em binário "01000001", e a paridade for par o bit de paridade será "0" para manter a quantidade par de números "1", já se a paridade escolhida for ímpar o bit de paridade será "1" fazendo assim que a transmissão tenha um número ímpar de números "1".

2.5.5 RS485

O padrão RS-485 (*Recommended Standard-485*) foi aprovado em 1983 pela *Electronics Industries Association* (EIA) como um novo meio de transmissão de dados por cabeamento. Foi bastante aceito em grande parte da indústria, como em setores de equipamentos médicos, automação industrial, embarcações, laboratórios, robótica, sistemas de ar condicionado, entre outras aplicações. As características mecânicas dos conectores elétricos não é uma regra do padrão e isso deixou uma certa autonomia aos fabricantes, sendo possível então estabelecer muitos tipos diferentes, de acordo com a necessidade de cada aplicação (FREITAS, 2017).

As topologias que utilizam o padrão RS-485 são a barramento ou a *daisy chain* e permitem a adição e remoção de dispositivos sem influenciar outro que já estejam em operação. Sendo um padrão multiponto, é possível adicionar vários dispositivos na mesma rede, diferente do padrão RS-232, onde apenas é admitida a comunicação entre dois dispositivos. A Figura 2.20 mostra as topologias de barramento e *daisy chain*.

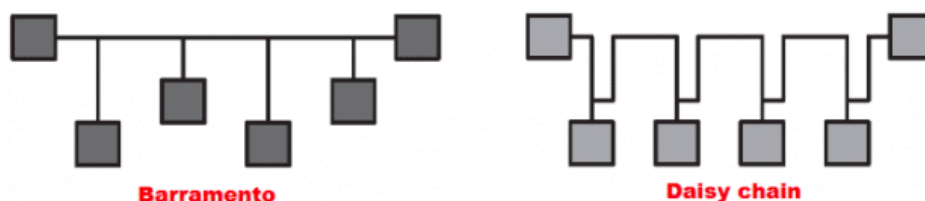


Figura 2.20: Topologia do barramento e daisy chain (FREITAS, 2017).

Esse padrão ainda é caracterizado pelo seu modo de transmissão via barramento, *full-duplex* ou *half-duplex*. A *half-duplex* utiliza um par de cabos padrão e em uma *full-duplex* utiliza-se de dois pares de cabos, Figura 2.21, sendo que em *half-duplex* somente é possível utilizar uma operação por vez no barramento, ou seja, as operações de envio ou recebimento de dados não podem ocorrer em um mesmo período de tempo. Já no modo *full-duplex* é possível realizar o envio, ou recebimento de dados de forma paralela.

No *half-duplex* somente um par de sinais é usado, exigindo então que ocorra um direcionamento, possibilitando, assim, que o recebimento dos dados ocorra em diferentes parcelas. Ambas as implementações necessitam de um controle, de todos os nós por meio de comandos de direção, como sinais de habilitação do *driver*/receptor, para garantir que apenas um *driver* esteja ativo no barramento por vez. (INSTRUMENTS, 2016)

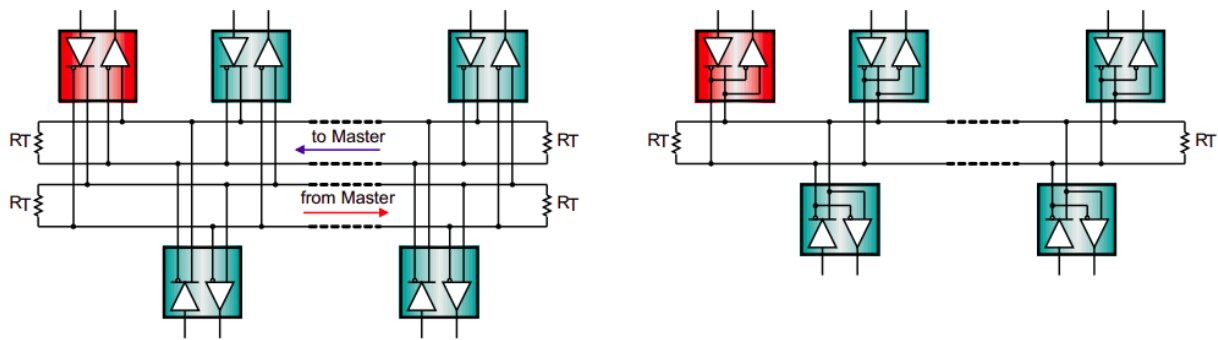


Figura 2.21: *Full-Duplex* e *Half-Duplex*, estrutura de transmissão no RS-485 (INSTRUMENTS, 2016).

Níveis de sinal e Tipo de cabo

Segundo (INSTRUMENTS, 2016), os *drivers* que trabalham em sintonia com o padrão RS-485 fornecem uma saída do tipo diferencial mínima de 1,5 V por uma carga de 54 Ω , enquanto os receptores em conformidade com o padrão detectam uma entrada diferencial de até 200 mV. Os dois valores fornecem margens que se mostram suficientes para uma transmissão de dados dita como confiável, mesmo quando ocorrer uma grave degradação do sinal ao longo do cabo e dos conectores. Esta robustez é uma marcante característica e uma das principais razões pelas quais o RS-485 é adequado para longas distâncias de rede em ambientes com ruídos.

A Figura 2.22 mostra essa especificação de entradas e saídas dos receptores, em conformidade com o padrão RS-485.



Figura 2.22: Especificação de nível de sinal para transmissão RS-485 (INSTRUMENTS, 2016).

Do tipo de cabo é possível afirmar que as aplicações RS-485 se caracterizam por possuir uma sinalização diferencial em um cabo do tipo par trançado. Por conta de ruídos externos, fontes se acoplam igualmente em ambas as linhas de sinal, como ruídos de modo comum, que são rejeitados pelo diferencial na entrada do receptor. Os cabos RS-485 são do tipo par trançado revestido, não blindado *Unshield Twisted Pair* (UTP), que possui como característica impedância de 120 Ω e 22-24 *American Wire Gauge* (AWG). A Figura 2.23 deixa bem claro as especificações desse tipo de cabo.



Figura 2.23: Modelo de cabo de comunicação RS-485, juntamente com suas especificações (INSTRUMENTS, 2016).

Taxa de transmissão X Comprimento do barramento

O comprimento máximo de um determinado barramento é limitado pelas possíveis perdas na linha de transmissão e o tempo de execução dos pacotes de dados, em uma determinada taxa. A Figura 2.24 traz os dados relacionados ao comprimento do cabo versus a característica de taxa de dados de um cabo RS-485 do tipo convencional para um sinal que possui 10 % de transmissão de pacotes (*jitter*).

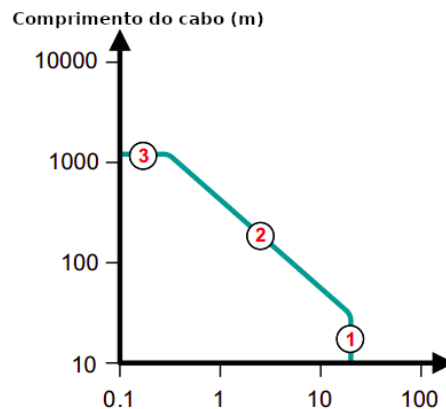


Figura 2.24: Taxa de dados (INSTRUMENTS, 2016).

De acordo com o gráfico da Figura 2.24:

- (1) Essa seção presente no gráfico apresenta a área de altas taxas de dados em cabos curtos. Nesse ponto as perdas da linha de transmissão podem ser desprezadas sem maiores prejuízos;
- (2) Essa seção mostra uma certa transição de pontos entre dados, de curtos para longas distâncias. Nesse caso as perdas das linhas de transmissão já devem ser levadas em consideração, ou seja, com o aumento do comprimento do cabo, a taxa de dados deve ser reduzida;
- (3) Essa seção do gráfico mostra a faixa de frequência mais baixa, local onde a resistência da linha e a ausência de comutação tende a limitar o comprimento efetivo do cabo.

2.5.6 Comunicação USB-Serial

O sinal do computador deve ser emitido e recebido pelo inversor de frequência de forma clara e legível ao inversor. Para que isso aconteça é necessária uma conversão do tipo USB-Serial RS-232/845, que acontece por meio de um conversor industrial USB para Serial DB-9, Figura 2.25



Figura 2.25: Conversor industrial USB para Serial DB-9 (PORT, 2015).

As Figuras 2.26 (a) e (b) mostram alguns detalhes importantes do módulo de conversão USB-Serial.

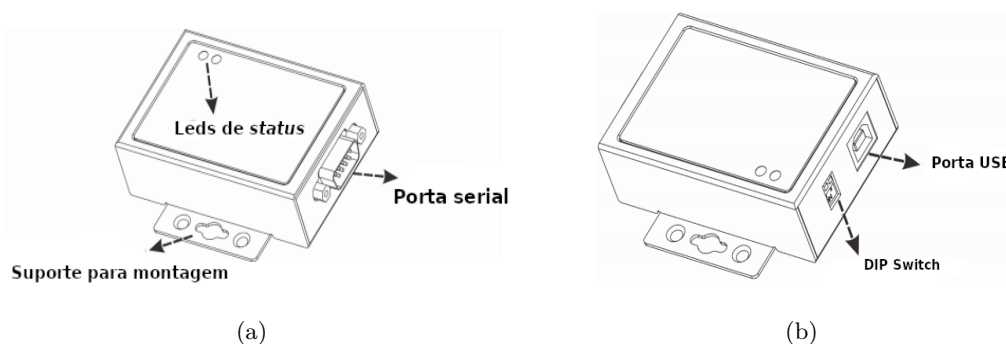


Figura 2.26: Layout de Hardware adaptador USB para porta (RS422/485) (PORT, 2015).

2.6 Protocolo de comunicação ModBus

O protocolo de comunicação Modbus é um protocolo industrial que foi desenvolvido em 1979 com o intuito de tornar possível a comunicação entre dispositivos, principalmente de automação. Foi implementado em um primeiro momento como um protocolo de nível de aplicação destinado a transferir dados por uma camada serial, que posteriormente foi ampliado para incluir implementações em comunicações seriais dos protocolos *Transmission Control Protocol/Internet Protocol* (TCP/IP) e *User Datagram Protocol* (UDP).

O Modbus é conhecido por ser caracterizado como um protocolo de requisição-resposta, que se relaciona na forma mestre-escravo. Um relacionamento tipo mestre-escravo tem como base a comunicação que sempre ocorre em pares, ou seja, um dispositivo tem a função de iniciar a

requisição e então aguardar por uma resposta.

O dispositivo mestre é o responsável por iniciar cada interação, ou seja, pelo envio da mensagem de requisição, disponibilizada pelo escravo posteriormente. Normalmente ocorre que o mestre é uma IHM ou um sistema *Supervisory Control and Data Acquisition* (SCADA), ou um computador. Já o escravo é um sensor, um controlador lógico programável (CLP), ou um inversor de frequência (INSTRUMENTS, 2019). A Figura 2.27 mostra a forma de relacionamento entre um dispositivo mestre e outro escravo.

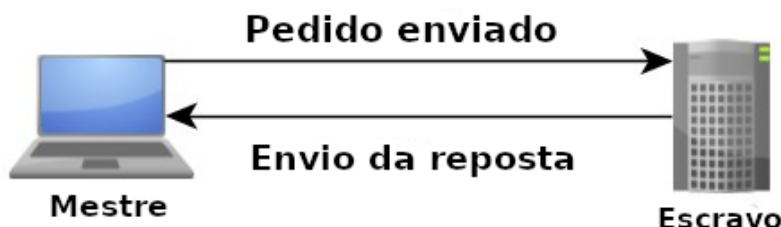


Figura 2.27: Relação mestre-escravo (INSTRUMENTS, 2019).

O conteúdo dessas requisições e respostas e as camadas de rede pelas quais essas mensagens são enviadas são definidos pelas diferentes camadas do protocolo. Os protocolos mais usados são TCP/IP e serial, mas outros também podem ser usados, como o UDP, por exemplo. Para transmitir os dados necessários, o Modbus possui um conjunto de variantes de *Application data unit* (ADU)¹, projetada nas conformidades de cada protocolo de rede.

Os formatos padrões para as ADU's são: TCP, ASCII e *Remote Terminal Unit* (RTU). Os tipos RTU e ASCII são tradicionalmente utilizadas em comunicação do tipo serial, que é o caso do inversor de frequência CFW-11 da fabricante Weg, que utiliza o ModBus RTU. As ADU's dos tipos RTU e ASCII são tradicionalmente utilizadas por uma linha serial, enquanto que o TCP é mais usado em redes TCP/IP ou UDP/IP modernas.(INSTRUMENTS, 2019)

No que diz respeito à transmissão de dados existe uma certa relação entre *Client* e *Server*. Caso a solicitação ocorra sem erros, relacionado à função ModBus que foi solicitada em um ModBus recebido corretamente, então a ADU do campo de dados de uma resposta de um servidor a um cliente contém os dados solicitados, isso de forma satisfatória. Se caso ocorra algum erro relacionado à função ModBus no momento de solicitação, então o campo contendo o código de exceção do servidor será usado para determinar uma possível próxima ação a ser executada (MODBUS, 2012). A Figura 2.28 mostra o esquemático de uma transmissão sem erros.

Para uma resposta de exceção, ou erro relacionado à transmissão ModBus, o servidor retorna um código que é equivalente à função original da *Protocol data unit* (PDU)² de solicitação com seu bit mais significativo definido como lógico em 1. A Figura 2.29 mostra o mesmo esquemático de solicitação ModBus, porém com um erro relacionado.

¹Unidade de dados de aplicação, nas categorias ASCII, RTU e TCP/IP.

²Em cada ADU, há uma PDU (*protocol data unit*) que é o centro do protocolo Modbus.

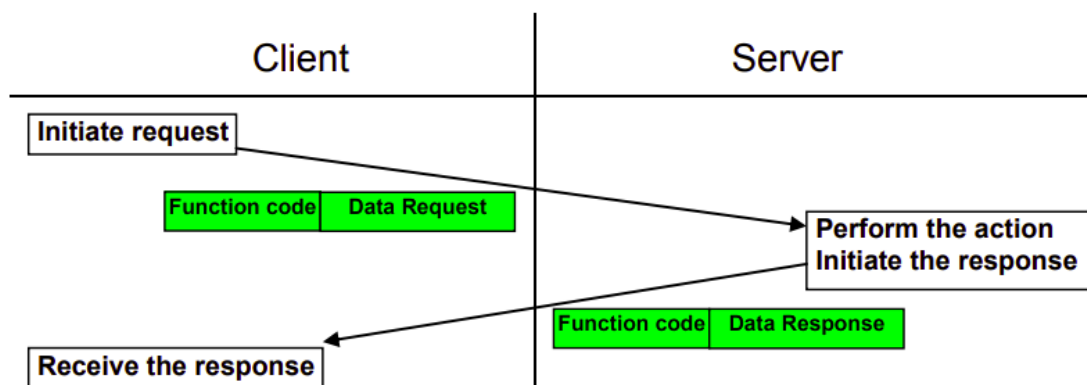


Figura 2.28: Transação ModBus sem erro (MODBUS, 2012).

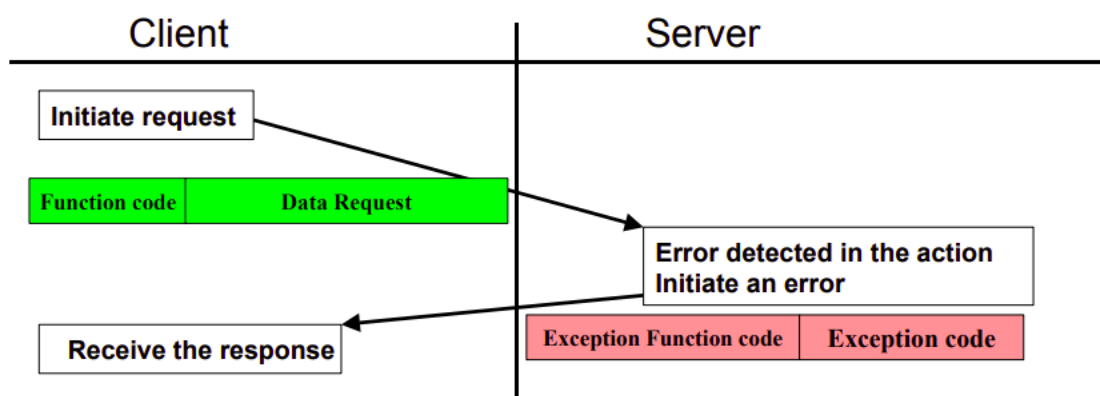


Figura 2.29: Transação ModBus com erro (MODBUS, 2012).

2.6.1 ModBus-RTU

A comunicação ModBus-RTU possui somente duas partes de informação, além da sua (PDU) base, que é composta por um campo função, com 1 byte (8 bits), e outro campo de dados de requisição, com 252 bytes de tamanho, Figura 2.30. O campo da função é o primeiro analisado, e caso o valor de função não seja válido para o dispositivo que recebe a requisição, ele responderá com uma mensagem de exceção. Caso o código do campo função seja aceito, o dispositivo escravo começará a processar os dados, conforme a definição da função.

Como os dados de requisição possuem um certo limite é preciso que os dispositivos sejam também limitados, com relação à quantidade de dados a serem transferidos.

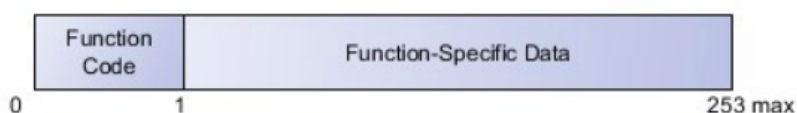


Figura 2.30: PDU do ModBus (INSTRUMENTS, 2019).

Depois de entender o formato da PDU no ModBus é possível mostrar a estrutura de mensagem

utilizada pelo ModBus RTU. Nesse formato um endereço é usado de modo a definir a qual escravo a PDU será enviada. O endereço 0 é conhecido como endereço de difusão, e significa que caso o mestre envie um comando ao endereço 0, necessariamente todos os escravos deverão realizar o processamento da requisição em questão. Além do endereço e da PDU, existe ainda um campo de *Cycling Redundancy Check* (CRC)³ que é a última parte do telegrama. A Figura 2.31 mostra a ADU do ModBus-RTU completo.

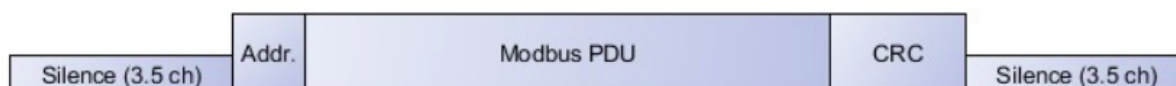


Figura 2.31: ADU do ModBus-RTU (INSTRUMENTS, 2019).

A Figura 2.32 mostra com clareza a forma de envio e recebimento das mensagens, além de especificar a sua estrutura e divisão.

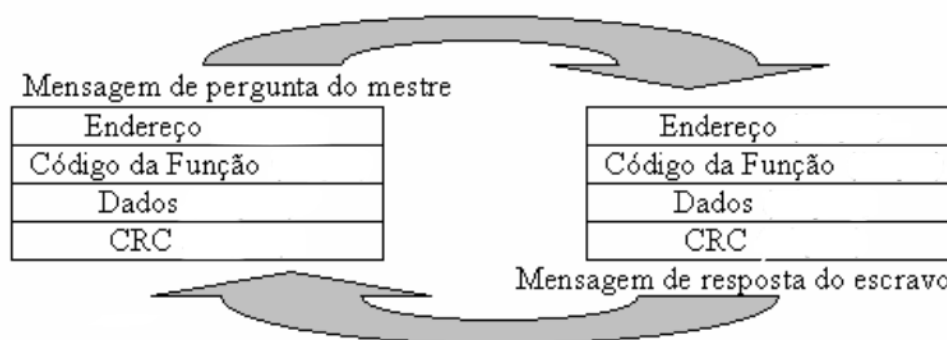


Figura 2.32: ADU do ModBus-RTU (WEG, 2007).

Da descrição completa dos campos, segundo o manual de Comunicação ModBus-RTU (WEG, 2007):

- **Endereço:** O mestre inicia todo o processo de comunicação por meio do envio de um byte, com o endereço do escravo para o qual se destina a mensagem. Após receber a solicitação, o escravo inicia então o telegrama, com o seu próprio endereço, possibilitando que o mestre saiba qual escravo está lhe enviando a resposta;
- **Código da função:** Campo composto por um único byte, em que o mestre descreve o tipo de serviço ou função solicitada ao escravo, pode ser leitura, escrita, etc. Segundo o protocolo de comunicação, cada função tem a capacidade de acessar um tipo específico de dado;
- **Campo de dados:** Alguns autores divergem em relação ao tamanho desse campo, segundo (INSTRUMENTS, 2019) trata-se de um campo com 252 bytes de tamanho, no entanto (WEG, 2007) afirma ser um campo de tamanho variável com n bytes. O formato e conteúdo deste campo dependem da função utilizada e dos valores a serem transmitidos;

³Método de detecção de erros normalmente usada em redes digitais e dispositivos de armazenamento para detectar mudança acidental em cadeias de dados.

- **CRC (Cycling Redundancy Check):** O método utilizado é o CRC-16. Este campo é formado basicamente por dois bytes, em que primeiro é transmitido o byte menos significativo (CRC-), e depois o mais significativo (CRC+).(WEG, 2007)

Tempo entre mensagens

Um ponto importante a ser descrito é o tempo entre mensagens. No modo ModBus-RTU não existe um bit específico com a função de unicamente indicar o início ou o fim de uma mensagem. A forma de se indicar o momento em que uma nova mensagem inicia e quando termina é por meio da ausência de transmissão de dados na rede. Para esse modo de comunicação um tempo mínimo de 3,5 vezes o tempo de transmissão de um byte de dados foi estabelecido.

No caso de uma determinada mensagem tenha iniciado após a decorrência deste tempo mínimo de 3,5 vezes, os elementos da rede tendem a assumir que o primeiro caractere recebido representa o início de um novo telegrama. O fim do telegrama será indicado quando, recebidos os bytes do telegrama, este tempo decorra novamente.

Um ponto importante é que, se durante a transmissão de um telegrama o tempo entre os bytes for maior que este tempo mínimo, o telegrama será considerado inválido, pois o inversor irá descartar os bytes já recebidos e montará um novo telegrama com os bytes que estiverem sendo transmitidos (WEG, 2010). O gráfico da Figura 2.33 ilustra a transmissão de um telegrama no tempo.

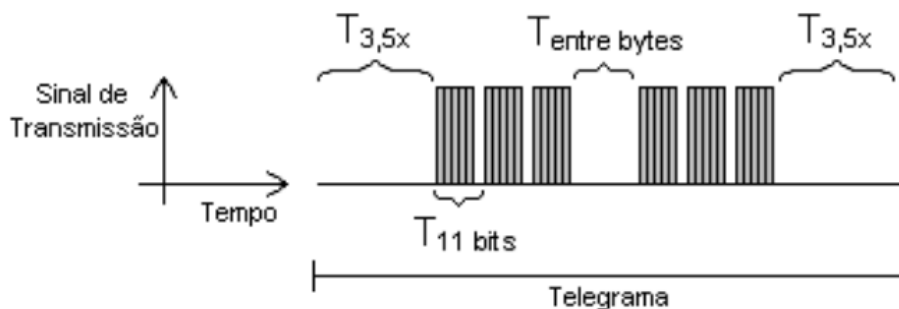


Figura 2.33: Faixas de tempo para o modo ModBus-RTU (WEG, 2010).

A tabela 2.2 abaixo mostra a relação entre algumas taxas de comunicação, seus tempos para transmitir uma palavra do telegrama T_{11bits} e o intervalo mínimo para indicar começo e fim de telegrama $T_{3,5x} = 3,5 \times T_{11bits}$.

Tabela 2.2: Taxas de comunicação e tempos envolvidos na transmissão de telegramas (WEG, 2010)

Tx comunicação	T_{11bits}	$T_{3,5x}$
9600 bits/s	1,146 ms	4,010 ms
19200 bits/s	573 μs	2,005 ms
38400 bits/s	573 μs	2,005 ms
57600 bits/s	573 μs	2,005 ms

2.6.2 Biblioteca MinimalModbus

A MinimalModbus é uma biblioteca Python, de fácil usabilidade e usada para comunicação entre dispositivos mestre-escravo usando o protocolo ModBus. Seu módulo deve ser executado no instrumento mestre. A única dependência do módulo MinimalModbus é o módulo pySerial (também Python puro).

Essa biblioteca Python suporta as versões de comunicação serial ModBus-RTU e ModBus-ASCII do protocolo ModBus, e destina-se ao uso em plataformas Linux, OSX e Windows. É de código aberto e possui a Licença Apache. (BERG, 2019)

```
#!/usr/bin/env python3
import minimalmodbus

instrument = minimalmodbus.Instrument('/dev/ttyUSB1', 1) # port name, slave address (

## Read temperature (PV = ProcessValue) ##
temperature = instrument.read_register(289, 1) # Registernumber, number of decimals
print(temperature)

## Change temperature setpoint (SP) ##
NEW_TEMPERATURE = 95
instrument.write_register(24, NEW_TEMPERATURE, 1) # Registernumber, value, number of
```

Figura 2.34: Exemplo uso MinimalModbus (BERG, 2019).

2.6.3 Biblioteca pySerial

Este módulo tem a funcionalidade de encapsular o acesso à porta serial. Ele fornece *back-ends*⁴ para o Python em execução no Windows, OSX, Linux, BSD e IronPython. A biblioteca serial seleciona automaticamente o *back-end* apropriado.(LIECHTI, 2017)

```
import serial
ser = serial.Serial('/dev/ttyUSB0') # open serial port
print(ser.name) # check which port was really used
ser.write(b'hello') # write a string
ser.close() # close port
```

Figura 2.35: Exemplo uso pySerial (LIECHTI, 2017).

⁴Parte secundária, de suporte ou de retaguarda da linguagem de programação.

2.7 Motor de Indução Trifásico

Um motor de indução é muito semelhante a uma máquina síncrona, ou seja, possuem o mesmo estator com uma construção de rotor diferente. Existem dois tipos diferentes de rotores de motor de indução, o rotor gaiola de esquilo e o rotor bobinado. A Figura 2.36 mostra os dois modelos de rotores usados em motores de indução.

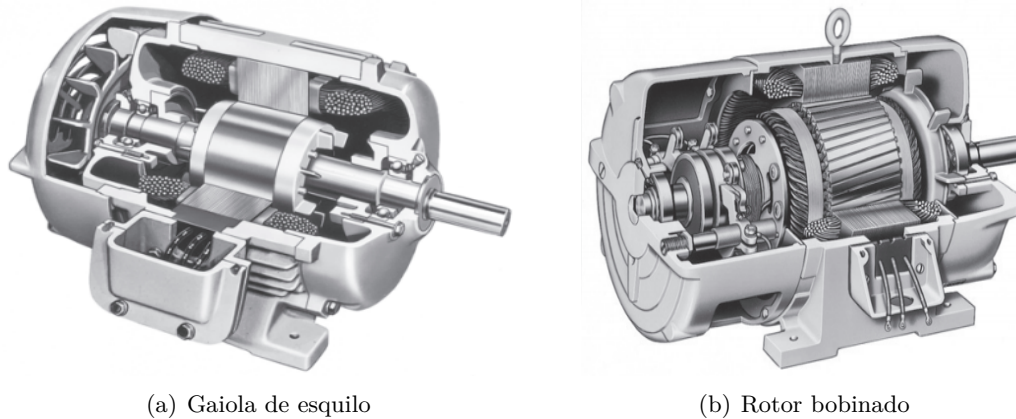


Figura 2.36: Vista de corte motor de indução (CHAPMAN, 2013).

Esse tipo de motor possui alimentação do tipo trifásica, onde as tensões em cada fase, encontram-se defasadas em 120° . Os motores de indução são do tipo assíncronos, existem também os motores síncronos.

2.7.1 Princípio de funcionamento

Em motores desse tipo, a velocidade de rotação n_{sinc} do campo magnético é dada por:

$$n_{sinc} = \frac{120f_{se}}{P} \quad (2.9)$$

onde,

- n_{sinc} : velocidade de rotação (em RPM);
- f_{se} : frequência do sistema aplicada ao estator (em hertz);
- P : número de polos da máquina.

Quando aplicado um conjunto trifásico de tensões defasadas em 120° no estator, ocorre o surgimento de um conjunto trifásico de correntes que circulam o mesmo. Essas correntes produzem um campo magnético chamado B_s , que gira em sentido anti-horário e com velocidade n_{sinc} . Esse campo magnético girante induz tensão nas barras do rotor, ao passar por elas.

A corrente do rotor produz um campo magnético girante chamado B_R , atrasado 90° em relação à própria corrente do rotor. Esse campo magnético B_R interage com o campo B_{Liq} , produzindo

assim um torque anti-horário na máquina, conforme a Figura 2.37 (a) e (b). Essa tensão que é induzida nas barras do motor é dada por:

$$e_{ind} = (vXB)l \quad (2.10)$$

onde,

- v : Velocidade da barra em relação ao campo magnético;
- B : Vetor densidade de fluxo magnético;
- l : Comprimento do condutor dentro do campo magnético.

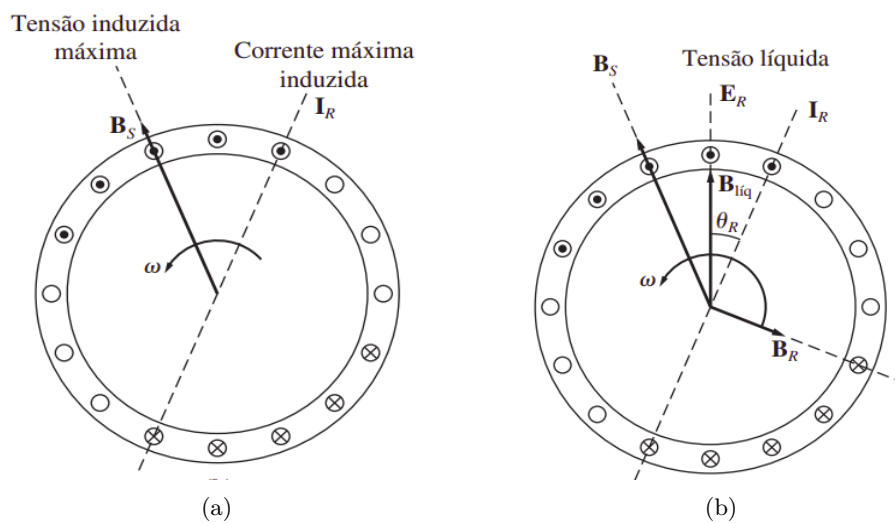


Figura 2.37: Campo girante de estator induzindo tensão nas barras do rotor.(CHAPMAN, 2013)

Em normal funcionamento, ambos os campos magnéticos do rotor e do estator B_R e B_S tendem a girar juntos, na velocidade síncrona n_{sinc} , ao mesmo tempo em que o próprio rotor gira a uma velocidade menor.

Capítulo 3

Descrição da bancada

Neste capítulo, são apresentados os principais componentes utilizados para o projeto. A bancada é composta por um Motor de Indução Trifásico (MIT) de 33 cavalos (cv), por um MIT de 1 cv, usado no ventilador, por um inversor de frequência modelo CFW-11 Weg, por um módulo RS-485 acoplado ao inversor e por um adaptador USB-Serial que permite a comunicação entre o computador e o inversor. Em (ANEXOS I), segue o diagrama elétrico da bancada.

3.1 Motor de Indução Trifásico

O motor em questão é da fabricante Weg. Esse motor apresenta 8 polos e conseqüentemente uma velocidade síncrona de 900 RPM. É um motor de 33 cv (22kW) e possui alimentação trifásica de 380 V. Sua ligação foi realizada em triângulo, considerando recomendações do fabricante. A Tabela 3.1 abaixo mostra as especificações mais detalhadas do motor.

Tabela 3.1: Dados do Motor de Indução Trifásico (IVO, 2017).

Motor de Indução Trifásico (MIT)	
Fabricante	Weg
Linha do produto	2258/M
Frequência	60 Hz
Número de polos	8
Potência	33 cv / 22kW
Rotação nominal	885 RPM
Escorregamento	15 RPM / 1.67%
Tensão nominal (Δ/Y)	380/660 V
Corrente nominal (Δ/Y)	45/25,9 A
Corrente à vazio (Δ/Y)	20,8/12 A
Conjugado nominal	238 Nm
Momento de inércia	0,84722 kgm ²
Carga	50% / 75% / 100%
Fator de potência	0,65 / 0,76 / 0,81
Rendimento	91,7 / 91,9 / 91,7

3.2 Motor do ventilador

Esse motor em específico é de 1 cv e é usado somente para o funcionamento do ventilador. Está acoplado na parte lateral do MIT e possui um acionamento à parte do sistema de partida do outro motor. Sua alimentação também é trifásica 380 V. A Tabela 3.2 mostra seus dados de especificação.

Tabela 3.2: Dados do ventilador (MENEZES; SATO, 2019).

Motor do ventilador	
Marca	Weg
Linha do produto	Blower Motor
Frequência	60 Hz
Potência	1 cv / 0,75 kW
Rotação Nominal	1730 RPM
Número de polos	4
Fator de Serviço	1,15
Tensão nominal	220/380/440 V
Corrente nominal	2,98/1,73/1,49 A
Rendimento	82,6%

A figura 3.1 mostra o motor de indução juntamente com o motor do ventilador utilizados no projeto.



Figura 3.1: Motor de indução trifásico e ventilador (MENEZES; SATO, 2019).

3.3 Inversor de frequência

O inversor de frequência utilizado é o CFW-11 modelo 0058 T405Z de fabricação da Weg. A Tabela 3.3 mostra as especificações desse inversor.

Tabela 3.3: Dados do inversor de frequência (MENEZES; SATO, 2019).

Inversor de frequência CFW-11	
Marca	Weg
Modelo	CFW 11 0058 T4054Z
Frequência de Alimentação	50 / 60 Hz
Tensão de Alimentação	380 / 440 V
Tensão de Saída	380 V
Rendimento Típico	Maior ou Igual a 97%
Fator de Potência Típico	0,94

A figura 3.2, mostra o inversor de frequência utilizado no projeto.

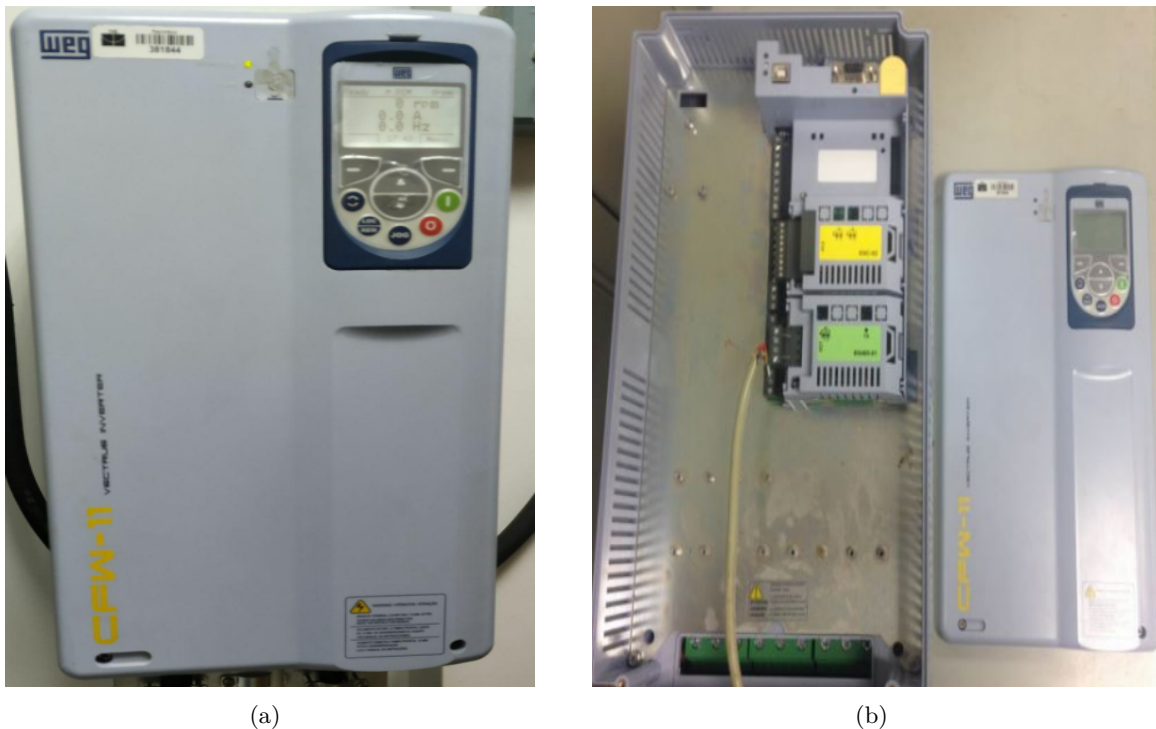


Figura 3.2: Inversor de frequência utilizado no projeto (MENEZES; SATO, 2019).

A figura 3.3 mostra todos os principais componentes do inversor de frequência CFW-11 da Weg e a Tabela 3.4 mostra detalhes de cada componente:

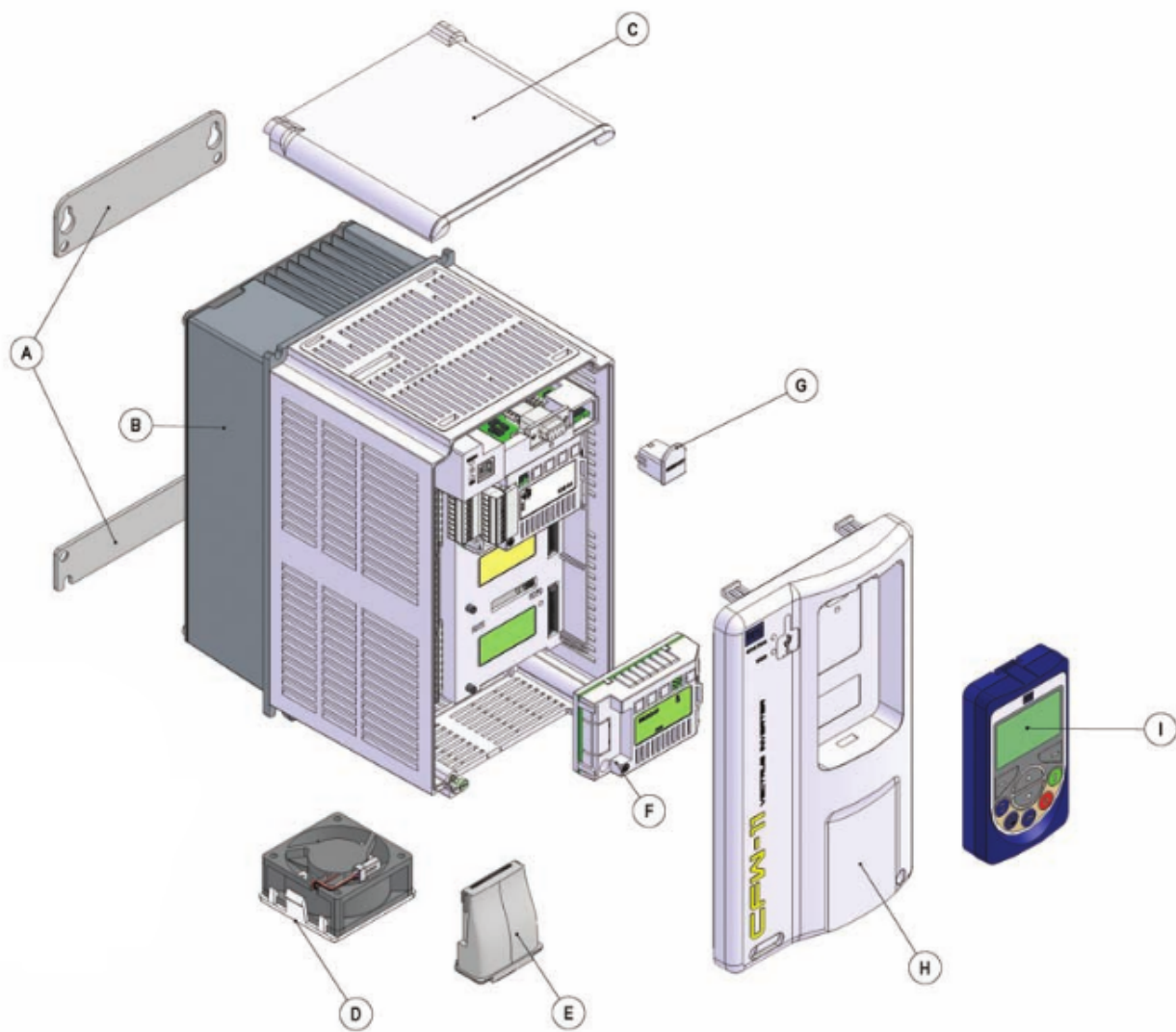


Figura 3.3: Principais componentes do CFW-11 (WEG, 2016).

Tabela 3.4: Descrição dos componentes CFW-11

Componente	Descrição
A	Suportes de fixação
B	Dissipador
C	Tampa superior
D	Ventilador com suporte de fixação
E	Módulo COMM 2 (<i>anybus</i>)
F	Módulo de cartão acessório
G	Módulo de memória FLASH
H	Tampa frontal
I	IHM

3.4 Módulo RS-485

O módulo usado para a comunicação RS-485 possui um conector plug-in de 4 vias (XC7). Suas especificações seguem na Tabela abaixo:

Tabela 3.5: Dados do módulo RS-485 (IVO, 2017).

Pino	Nome	Função
1	A-Line (-)	RxD/TxD negativo
2	B-Line (+)	RxD/TxD positivo
3	GND	0V isolado do circuito RS-485
4	Ground	Terra (blindagem)

A figura 3.4, mostra o módulo de comunicação RS-485 utilizado no projeto.



Figura 3.4: Módulo RS-485.(INDUSTRIAL, 2020)

3.5 Adaptador USB-Serial

O adaptador utilizado no projeto é idêntico ao mostrado anteriormente na figura 2.25. Esse modelo possui uma sequência de ligações para o adaptador serial DB9, mostradas na Tabela 3.6.

Tabela 3.6: Dados do adaptador USB-Serial (IVO, 2017).

Pino	Nome
1	D+
2	D-
3	-
4	-
5	GND

Para configurar o adaptador para utilizá-lo em uma comunicação de dois fios é preciso posicionar as chaves seletoras de forma onde a chave número 1 está ligada e a chave número 2 está desligada.

Capítulo 4

Comunicação e Controle

4.1 Configuração do inversor para acesso remoto

O inversor deve ser configurado inicialmente, de modo a possibilitar acesso do tipo remoto via USB-Serial. Sem esses passos de configuração inicial, via IHM, não é possível realizar a parametrização do equipamento remotamente.

Tabela 4.1: Parâmetros ajustados para acesso remoto.

Valores parâmetros
P0000 \Rightarrow 5
P0307 \Rightarrow 1
P0202 \Rightarrow 0
P0220 \Rightarrow 6
P0221 \Rightarrow 9

- **P000:** Acesso aos Parâmetros (5 =Possibilita alterar o conteúdo dos parâmetros);
- **P0307:** *Start-up* Orientado (1 = Sim);
- **P0202:** Tipo de Controle (0 = V/f 60Hz);
- **P0220:** Seleção Fonte LOC/REM (6 = Serial/USB REM);
- **P0221:** Seleção Referência LOC (9 = Serial/USB).

4.2 Comunicação via *Software* SuperDrive

Uma das partes mais importantes desse capítulo é a comunicação serial entre computador e inversor de frequência, usando o módulo de comunicação USB-Serial. Para tal utilizou-se o *software* disponibilizado pela Weg *SuperDrive* G2, um *software* de parametrização de *drives*, que possibilita a verificação dessa comunicação.

As definições da USB *Serial Port* (COM3) necessárias foram as seguintes:

Tabela 4.2: Configuração USB Serial Port (COM3)

Campo	Dado
Bits por segundo	9600
Bits de dados	8
Paridade	Nenhum
Bits de paridade	1
Controle de fluxo	Nenhuma

Esses campos foram ajustados via computador, acessando a porta COM3.

Com a comunicação estabelecida via porta USB-Serial, foi possível então realizar uma outra comunicação, entre o inversor de frequência e o software Super Drive. Para isso foi necessário manter o inversor no modo remoto (REM) e sincronizar as configurações do inversor com o software, conforme Figura 4.1.



Figura 4.1: Configurações estabelecidas no Software SuperDrive.

Um ponto importante que impossibilitava essa comunicação era o fato de o notebook estar com o carregador conectado à tomada. Nesse caso não era possível estabelecer a comunicação de forma alguma. Esse empecilho foi resolvido após o aterramento correto do inversor de frequência e do motor de indução. Depois dessa realização foi possível estabelecer a comunicação com o *notebook* conectado à tomada de forma satisfatória, sem mais problemas.

Feito isso, realizou-se um *backup* dos parâmetros do inversor, de forma a não perder os dados já salvos e, através do *software*, alguns parâmetros foram alterados, a título de curiosidade, de modo a confirmar a comunicação entre o inversor e o *software*. Percebeu-se um erro de transmissão de dados quando o *notebook* estava conectado à tomada, caso o carregador fosse retirado da tomada

tudo voltava a funcionar corretamente, mas como dito anteriormente, esse problema técnico foi resolvido.

O Software *SuperDrive* dava acesso a todos os parâmetros do inversor, possibilitando uma rápida busca por configurações de parâmetros já existentes no dispositivo. Com isso foi possível realizar uma análise mais detalhada e mais rápida, das configurações preexistentes no inversor de frequência. Alguns parâmetros são desabilitados para escrita via software, impossibilitando assim erros mais graves com comandos de acionamento ou de controle. A Figura 4.2 mostra alguns parâmetros obtidos por meio do *SuperDrive*.

Número	Função	Mínimo	Máximo	Ajuste de Fábrica	Ajuste do Usuário	Unidade
0	Acesso aos Parâmetros	0	9999	0	0	
1	Referência Velocidade	0	18000	0	0	rpm
2	Velocidade do Motor	0	18000	0	0	rpm
3	Corrente do Motor	0	4500	0	0	A
4	Tensão Barram.CC (Ud)	0	2000	0	535	V
5	Frequência do Motor	0	1020	0	0	Hz
6	Estado do Inversor	0	7	0: Ready (Pronto)	0: Ready (Pronto)	
7	Tensão de Saída	0	2000	0	0	V
9	Torque no Motor	-1000	1000	0	0	%
10	Potência de Saída	0	6553,5	0	0	kW
11	cosPhi da Saída	0	1	0	0	
12	Estado DI8 a DI1	00000000b	11111111b	00000000b	00000000b	
13	Estado DO5 a DO1	00000000b	00011111b	00000000b	00000001b	
14	Valor de AO1	0	100	0	0	%
15	Valor de AO2	0	100	0	0	%
16	Valor de AO3	-100	100	0	0	%
17	Valor de AO4	-100	100	0	0	%
18	Valor de AI1	-100	100	0	0	%
19	Valor de AI2	-100	100	0	0	%
20	Valor de AI3	-100	100	0	0	%
21	Valor de AI4	-100	100	0	-100	%
23	Versão de Software	0	655,35	0	3,14	
25	Estado DI16 a DI9	00000000b	11111111b	00000000b	00000000b	
26	Estado DO13 a DO6	00000000b	11111111b	00000000b	00000000b	
27	Config. Acessórios 1	00h	FFFFh	00h	C2h	
28	Config. Acessórios 2	00h	FFFFh	00h	EE40h	
29	Config. HW Potência	00000000b	1111111111111111b	00000000b	1100010101001011b	
30	Temperatura IGBTs U	-20	150	0	25,8	°C
31	Temperatura IGBTs V	-20	150	0	25,8	°C
32	Temperatura IGBTs W	-20	150	0	25,8	°C
33	Temper. Retificador	-20	150	0	25,2	°C
34	Temper. Ar Interno	-20	150	0	40,5	°C

Figura 4.2: Acesso aos dados do inversor via Software *SuperDrive*.

4.3 Comunicação via MinimalModbus e pySerial

Depois do teste que confirmava o funcionamento do drive, com a utilização do *software SuperDrive*, foi estabelecido outro objetivo, o de realizar a comunicação serial, agora utilizando as bibliotecas MinimalModbus e pySerial. A programação que possibilitou essa conexão foi realizada em linguagem de programação Python, concatenando as informações das bibliotecas pySerial e MinimalModbus.

Alguns conceitos sobre a forma de envio da mensagem e o modelo de resposta foram obtidos por meio dos manuais de programação e comunicação da Weg. As informações a respeito das bibliotecas MinimalModbus (BERG, 2019) e pySerial (LIECHTI, 2017) foram obtidas em seus sites, que descrevem o uso e funcionamento de forma detalhada. O que se priorizava na conexão entre o programa em Python e o inversor de frequência era a forma como a biblioteca pySerial seria usada juntamente com a MinimalModbus, de forma a possibilitar a leitura ou escrita de parâmetros no inversor de frequência.

4.3.1 Estabelecendo conexão

Primeiramente, foi necessária a instalação das bibliotecas, de forma que pudessem ser importadas durante a programação. Foi possível o *download* nos sites (BERG, 2019) e (LIECHTI, 2017). O primeiro passo para a comunicação foi importar as bibliotecas pySerial e a MinimalModbus, por meio dos comandos, *import serial* e *import minimalmodbus*.

Com isso foi possível obter os dados de comunicação do instrumento por meio de alguns comandos, também obtidos no site MinimalModbus.

Comando esse: `minimalmodbus.Instrument('COM3',1,'rtu')`, onde COM3 é o nome da porta, "1" é a informação de endereço e `rtu` é o modo de comunicação.

A biblioteca pySerial teve de ser incorporada à biblioteca MinimalModbus, possibilitando assim a troca de mensagens entre o computador e o inversor de frequência, como leitura e alteração de parâmetros. Como dito anteriormente, algumas informações tiveram de ser concatenadas às bibliotecas já importadas. Informações essas:

- **baudrate:** taxa de transmissão como 9600 ou 115200 bps;
- **bytesize:** número de bits de dados. Valores possíveis: FIVE BITS, SIX BITS, SEVEN BITS, EIGHT BITS;
- **parity:** ativar verificação de paridade.
Valores possíveis: PARITY_NONE, PARITY_EVEN, PARITY_ODD, PARITY_MARK, PARITY_SPACE;
- **stopbits:** número de stopbits.
Na forma: STOPBITS_ONE, STOPBITS_ONE_POINT_FIVE, STOPBITS_TWO;
- **time out:** tempo limite.

Como as bibliotecas devem estar ligadas, o modo como o comando é dado é importante, a comunicação deve ser atribuída a um nome, que depois será relacionado com a porta serial. O exemplo do uso para as configurações da serial do inversor, atribuindo:

- `instr = minimalmodbus.Instrument('COM3',1,'rtu')`, é dado na forma:

```
instr.serial.baudrate = 9600
instr.serial.bytesize = serial.EIGHTBITS
instr.serial.parity = serial.PARITY_NONE
instr.serial.stopbits = serial.STOPBITS_ONE
instr.serial.timeout = 1
```

O código abaixo mostra como a comunicação foi configurada:

```

1 #Comunicação Serial Port
2 import serial
3 import minimalmodbus
4 instr = minimalmodbus.Instrument('COM3',1,'rtu')
5 instr.mode = 'rtu'
6 instr.debug = True #Mostra dados sobre a leitura
7 #Configuracao dos parametros conforme dados do inversor:
8 instr.serial.baudrate = 9600
9 instr.serial.bytesize = serial.EIGHTBITS
10 instr.serial.parity = serial.PARITY_NONE
11 instr.serial.stopbits = serial.STOPBITS_ONE
12 instr.serial.timeout = 1

```

Listing 4.1: Trecho do código usado para comunicação.

A forma de leitura e escrita de parâmetros segue a mesma linha de implementação. O comando para leitura é do tipo `read_register(registeraddress, number_of_decimals=0, functioncode=3/4, signed=true/False)`. Para esse caso permite-se a leitura de um inteiro de 16 bits no escravo e os argumentos são:

- **registeraddress (int)**: o endereço do registrador no escravo (uso de números decimais, não hex);
- **number_of_decimals (int)**: o número de decimais para conversão de conteúdo;
- **functioncode (int)**: ModBus functioncode. Pode ser 3 ou 4;
- **signed (bool)**: se os dados devem ser interpretados como signed ou unsigned.

Já para escrita de parâmetros, o comando é dado na forma `write_register(registeraddress, value, number_of_decimals=0, functioncode=6/16, signed=True/False)`. Para esse caso permite-se a escrita de um inteiro de 16 bits no escravo e os argumentos são:

- **registeraddress (int)**: o endereço do registrador no escravo (uso de números decimais, não hex);
- **value (int or float)**: o valor a ser armazenado no registrador escravo;
- **number_of_decimals (int)**: o número de casas decimais para conversão de conteúdo;
- **functioncode (int)**: ModBus functioncode. Pode ser 6 ou 16;
- **signed (bool)**: se os dados devem ser interpretados como signed ou unsigned.

O código abaixo mostra os comandos de leitura e escrita, conforme usados no programa.


```

1 #Write_register:
2 #instr.write_register(registeraddress, value, number_of_decimals=0,
   functioncode=16, signed=False)
3 instr.write_register(682, 0x0015, number_of_decimals=0, functioncode= 6,
   signed=False)
4
5 #read_register:
6 #parameter = instr.read_register(registeraddress, number_of_decimals=0,
   functioncode=3, signed=False)
7 parameter = instr.read_register(100,1,number_of_decimals=0, functioncode=
   3, signed=False)
8 print(parameter)

```

Listing 4.2: Escrita e leitura de parâmetros.

4.4 Implementação do controle

Para a implementação do controle do inversor de frequência utilizou-se a biblioteca Tkinter⁵, que fornece uma interface do usuário bastante intuitiva e que facilita o entendimento do projeto. A parte de emulação da turbina eólica, com a implementação dos perfis de vento e a finalização da interface do usuário não foi possível ser finalizada, por conta da restrição de acesso ao laboratório de Qualidade da energia elétrica. Contudo uma grande parte da interface foi desenvolvida com sucesso e alguns perfis de velocidade foram implementados também.

A Figura 4.3 mostra a página principal. É possível realizar a conexão por meio do botão *Connect*. Caso a conexão seja realizada com sucesso uma mensagem de confirmação aparecerá para o usuário, além disso, um campo com a mensagem "conectado" será ativado na parte inferior da janela. Caso a conexão não seja estabelecida, uma mensagem de erro, com a descrição do tipo de erro, será mostrada na tela, conforme a Figura 4.4 e uma mensagem de não conexão será mostrada também na parte inferior da janela, conforme mostra a Figura 4.5, em *Connection Status*.

Essa mensagem de status de conexão foi possível por meio de uma exceção, realizada em linguagem Python, que identifica o ponto do código que falha caso ocorra algum problema de conexão e, nesse ponto específico, mostra uma mensagem, definida pelo programador. Para que a interface fique acessível ao usuário basta realizar a execução do programa em Python. A IDE de programação utilizada foi a *Sublime text*, ideal para programação nessa linguagem específica, e a versão foi o Python 3.6 usando o sistema operacional *Windows 8.1*.

Outros pontos importantes também foram adicionados à interface de usuário, como por exemplo, o botão CFW-11 que mostra dados técnicos sobre o inversor de frequência utilizado no projeto. Algo de relevância que pode ser implementado futuramente é o controle vetorial.

⁵Fornece um kit de ferramentas de janela robusto e independente de plataforma, usando o pacote tkinter e sua extensão, os módulos tkinter.tix e tkinter.ttk.

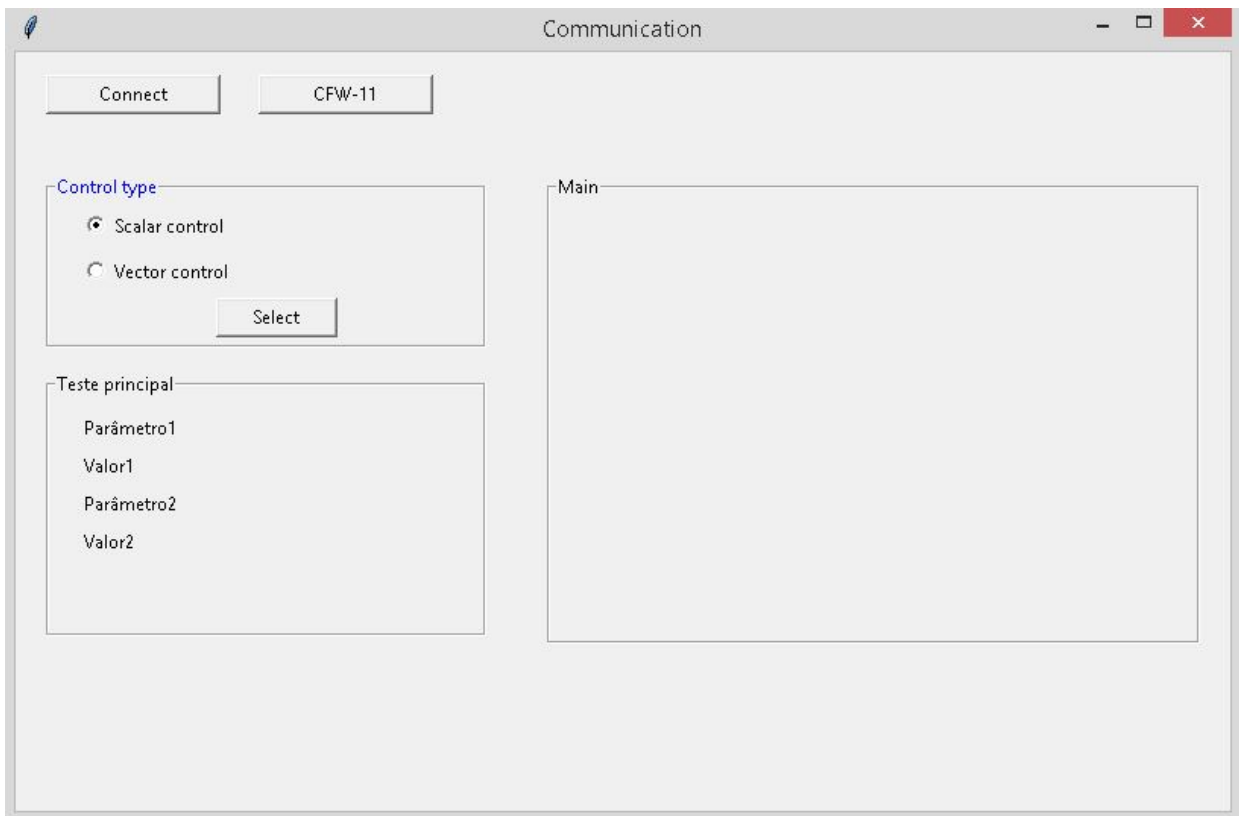


Figura 4.3: Janela principal da interface do usuário.

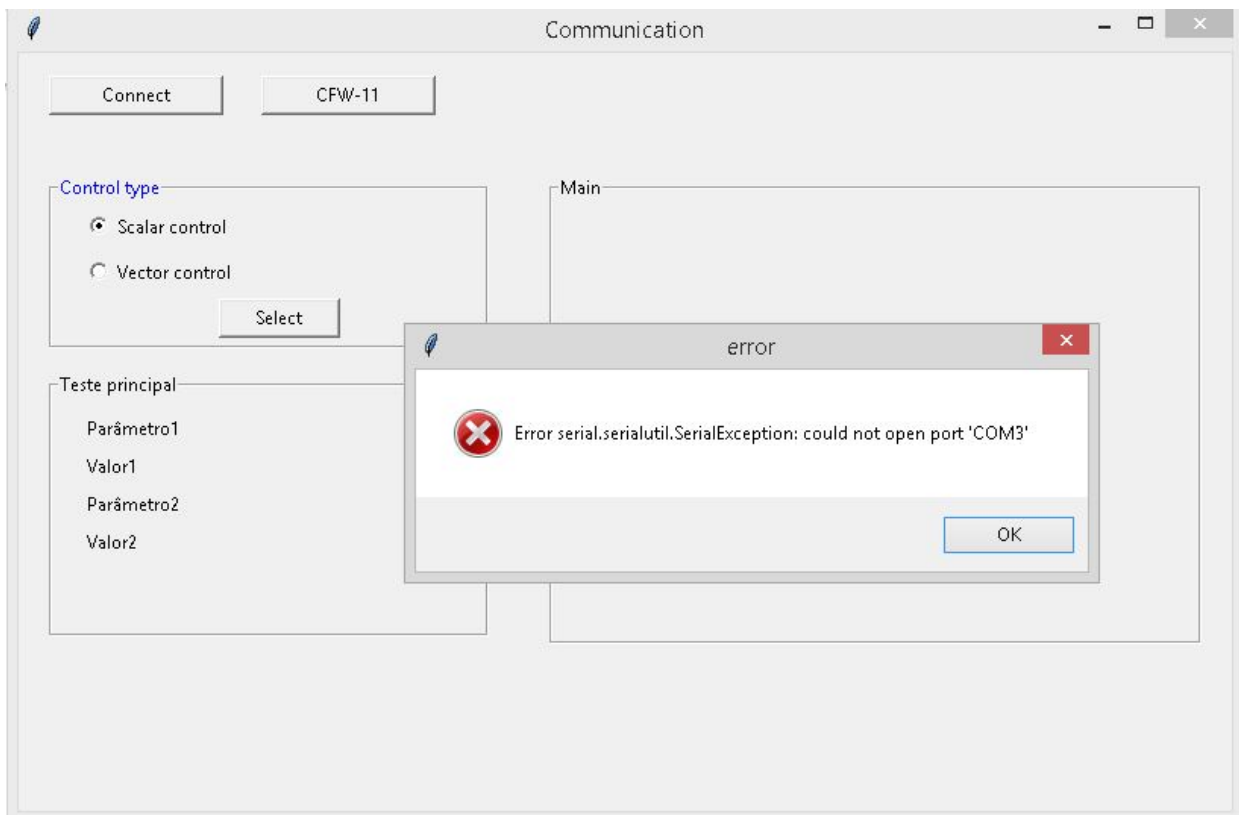


Figura 4.4: Janela com mensagem de erro após a tentativa de conexão sem sucesso.

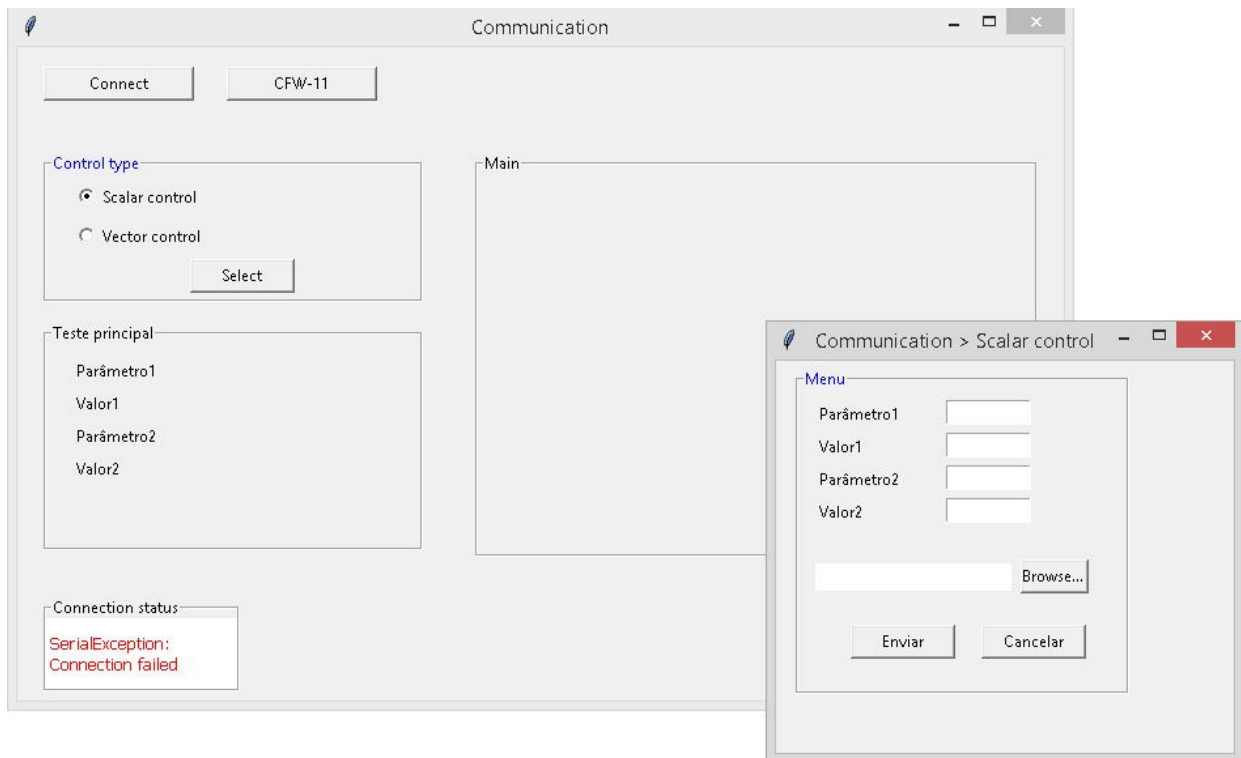


Figura 4.5: Janela com a seleção do controle escalar.

A Figura 4.5 mostra a janela resultante da operação de escolha do tipo de controle em *Control type* » *Scalar control*.

O perfil de velocidade criado seguiu a lógica de cálculo da aceleração, ou desaceleração, baseados no tempo definido e na velocidade a ser alcançada, em RPM. Por meio da (equação 4.1):

$$a_n = \frac{\Delta V}{\Delta t} \quad (4.1)$$

onde,

- a_n é a aceleração, ou desaceleração (em RPM/s²);
- ΔV é a variação de velocidade (em RPM/s);
- Δt é a variação de tempo (em s).

Os valores de aceleração e desaceleração devem ser escritos nos parâmetros P0100, para aceleração e P0101, para desaceleração, conforme manual de programação. A Figura 4.6 mostra de forma melhor como os perfis são criados, considerando velocidade, aceleração, desaceleração e tempo. Os valores de aceleração e desaceleração devem ser escritos nos parâmetros P0100, para aceleração e P0101, para desaceleração, conforme manual de programação. Os valores são ajusta-

dos, em código, com o auxílio do *time.sleep()*, que possibilita uma pausa na execução do programa, por determinado período de tempo, em segundos.

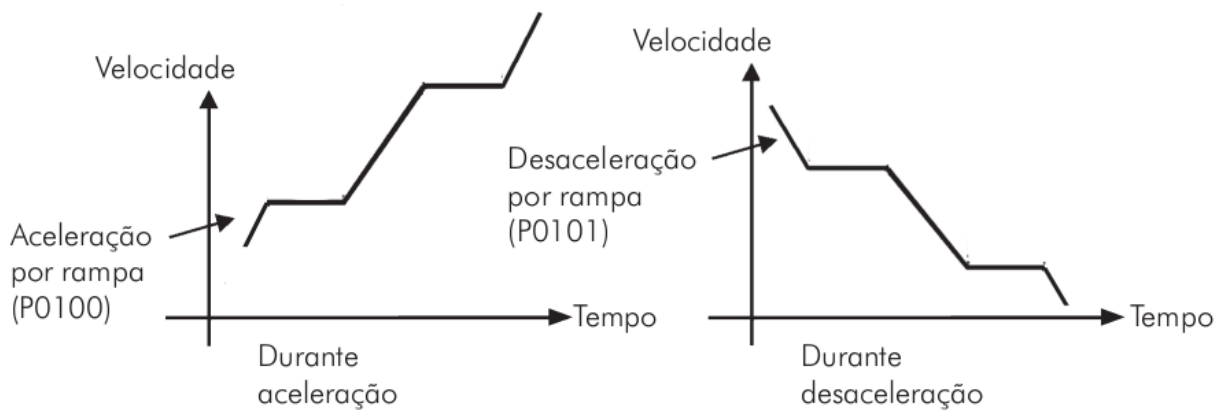


Figura 4.6: *Hold* de rampa(WEG, 2012).

Capítulo 5

Conclusões

Em alguns trabalhos anteriores a esse, alguns pontos ficaram pendentes e sem solução, como a falha causada na comunicação com o inversor de frequência caso o *notebook* estivesse conectado à tomada. Isso foi solucionado com o aterramento correto do motor de indução e do inversor de frequência, segundo especificações do manual.

A realização da comunicação utilizando as bibliotecas Python de forma conjunta, possibilitou o acompanhamento dos resultados de forma bastante clara, pois a biblioteca Tkinter permitiu o monitoramento da conexão por meio de um comando de exceção que identifica o ponto de erro e mostra uma mensagem caso isso ocorra. Por meio da documentação Weg, foi possível estabelecer o funcionamento e tornar o inversor de frequência pronto para ser acessado de forma remota, possibilitando assim que as configurações de controle fossem implementadas de forma satisfatória.

As bibliotecas: MinimalModbus, pySerial e Tkinter foram ferramentas fundamentais na conversação entre mestre e escravo, funcionando como ponto entre linguagem de programação de alto nível e dispositivos de controle. A descrição do protocolo juntamente com a descrição do equipamento, e a programação fizeram-se possíveis graças aos manuais de programação e comunicação, disponibilizados pelo fabricante do inversor de frequência.

Com o melhoramento da IHM, será possível a criação de uma bancada dedicada a simulações envolvendo turbinas eólicas, o que pode agregar muito valor aos equipamentos já existente no laboratório de Qualidade da Energia Elétrica.

5.1 Perspectivas Futuras

O código em Python (ANEXOS II), está bem estruturado e seu entendimento não é muito complicado. Levando em consideração que a comunicação está funcionando, é possível realizar o melhoramento do controle escalar e a implementação do controle vetorial, tendo em vista a disponibilidade de um encoder no laboratório.

Por conta das limitações de acesso ao laboratório, causadas pela pandemia, não foi possível realizar o controle da maneira pretendida, com a implementação de perfis de velocidade dos mais

variados, com a obtenção dos gráficos em pdf e a leitura de dados de um arquivo externo, como por exemplo .csv excel. Todos esses pontos podem ser realizados em trabalhos futuros, aproveitando a estrutura de comunicação funcional, junto com a IHM.

REFERÊNCIAS BIBLIOGRÁFICAS

ABEEÓLICA. *Dados ABEEólica*. 2019. <<http://abeeolica.org.br/dados-abeeolica/>>, acessado 25/02/2020.

DUTRA, R. *Energia Eólica: Princípios e tecnologias*. 2019. <<http://www.cresesb.cepel.br/>>, acessado 27/02/2020.

REIS, P. *Como funciona um aerogerador*. 2016. <<https://www.portal-energia.com/funcionamento-de-um-aerogerador/>>, acessado 26/02/2020.

STROSK, P. N. *turbinas Eólicas*. 2019.

RICO, O. C. *Tecnologia em aerogeradores*. 2017.

WEG. *Módulo 2. Variação de velocidade*. 2009. (Weg), Jaraguá do Sul.

FRANCHI, C. M. *Inversores de Frequência. Teoria e aplicações- Segunda Edição*. São Paulo: Editora Érica, 2013. 192 p. (Livro, 238).

WEG. *Manual do inversor de frequência. CFW-11*. 2016. <<https://static.weg.net/medias/downloadcenter/hab/hc0/WEG-cfw11-manual-do-usuario-200-400v-mec.-a-a-d-10000062964-manual-portugues-br.pdf>>.

TOSSI, L. *Aplicação de UPS dupla conversão em missão crítica – Funcionamento geral e detalhamento do inversor (PWM)*. 2011. <http://www.osetoreletrico.com.br/wp-content/uploads/2011/04/Ed62_fasc_condicionamento_capIII.pdf>, acessado 22/04/2020.

ELECTRONICS, A. *Inversor de frequência- O que é e como funciona*. 2020. <<https://athoselectronics.com/inversor-de-frequencia/>>, acessado 05/03/2020.

WEG. *Manual de programação CFW-11*. 2012. <<https://www.controldrives.com.br/upload/arquivos/weg-cfw11-manual-inversor-de-frequencia.pdf>>.

FREITAS, C. M. *Redes de comunicação em RS-485*. 2017. <<https://www.embarcados.com.br/redes-de-comunicacao-em-rs-485/>>, acessado 10/11/2020>.

INSTRUMENTS, T. *The RS-485 Design Guide*. 2016. <<https://www.ti.com/lit/an/slla272c/slla272c.pdf?ts=1606513991614>>, acessado 11/11/2020>.

PORT, C. i. F. *Conversor USB para 1, 2 ou 4 portas Seriais RS422/485 Manual do Usuário*. 2015. <<https://www.flexport.com.br/Content/upload/produtos/arquivos/84/f5411e-f5421e-f5441e-port.pdf>>, acessado 11/11/2020>.

INSTRUMENTS, N. *O protocolo Modbus em detalhes*. 2019. <<https://www.ni.com/pt-br/innovations/white-papers/14/the-modbus-protocol-in-depth.html>>, acessado 11/11/2020>.

- MODBUS. *MODBUS APPLICATION PROTOCOL SPECIFICATION*. 2012. <https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf>, acessado 20/11/2020.
- WEG. *Manual da Comunicação Modbus-RTU*. 2007. <<http://www.gigawattsistemas.com.br/file/1-15618inteligente.pdf>,acessado11/11/2020>.
- WEG. *RS232 / RS485 Serial Communication Manual*. 2010. <<https://static.weg.net/medias/downloadcenter/h58/h90/WEG-cfw11-rs232-and-rs485-manual-0899.5741-en.pdf>,acessado12/11/2020>.
- BERG, J. *MinimalModbus*. 2019. <<https://minimalmodbus.readthedocs.io/en/stable/readme.html>>, acessado 26/02/2020.
- LIECHTI, C. *pySerial's documentation*. 2017. <<https://pyserial.readthedocs.io/en/latest/>,acessado15/11/2020>.
- CHAPMAN, S. J. *Fundamentos de Máquinas elétricas*. São Paulo: Editora Techbooks, 2013. 698 p. (Livro, 100).
- MENEZES, M. H. D.; SATO, V. Y. *Desenvolvimento de Interface de Comunicação Baseada no Protocolo Modbus para Conexão de um Computador (PC) a um Inversor de Frequência*. 2019. Monografia (Graduação em Engenharia), UnB (Universidade de Brasília), Brasil.
- INDUSTRIAL, C. A. *Módulo de comunicação serial RS485 (Modbus) WEG - CFW11*. 2020. <<https://csaautomacao.com.br/store/produtos/inversores-de-frequencia/inversor-de-frequencia-cfw11/acessorios-cfw11/modulo-de-comunicac-o-serial-rs485-modbus-weg-cfw11.html>>, acessado 20/11/2020.
- IVO, B. M. *Controle de um Motor de Indução trifásico para Emulação de uma Turbina Eólica*. 2017. Monografia (Graduação em Engenharia), UnB (Universidade de Brasília), Brasil.
- MARTINS, F.; GUARNIERI, R.; PEREIRA, E. *O aproveitamento da energia eólica*. 2007. <<http://www.scielo.br/pdf/rbef/v30n1/a05v30n1.pdf>>, acessado 26/02/2020.
- HERBERT, J.; INIYAN, S.; RAJAPADIAN, S. A review of wind energy technologies. *ScienceDirect*, v. 11, n. 6, p. 1117–1145, 2007.
- WWEA. *Wind Power Capacity reaches 546 GW, 60 GW added in 2017*. 2017. <<https://wwindea.org/blog/2018/02/12/2017-statistics/>>, acesso 24/02/2020.
- CEBDS. *Região Nordeste é responsável por 86% da produção de energia eólica brasileira*. 2015. <<https://cebds.org/>>,acessado 24/02/2020.
- PINTO, M. *Fundamentos de Energia Eólica*. São Paulo: Editora LTC, 2013. 392 p. (Livro, 100).
- SILVA, J. R. C. da. *Otimização da posição de aerogeradores em parque eólico*. 2014. Monografia (Mestre em Engenharia), UnB (Universidade de Brasília), Brasil.
- CUSTODIO, R. dos S. *Energia eólica para produção de energia elétrica, segunda edição*. Rio de Janeiro: Synergia Editora, 2013. 340 p. (Livro, 157).
- CRESESB. *Tipos de Aerogeradores para Geração de Energia Elétrica*. 2014. <http://www.cresesb.cepel.br/index.php?section=com_content&lang=pt&cid=231>,acessado 27/02/2020.
- KOMORI, J.; FREITAS, G. H. S.; PASSOS, W. *Análise de uma turbina Giromill*. 2017. <<http://eventos.ufgd.edu.br/enepex/anais/arquivos/2279.pdf>>, acessado 02/03/2020.

CASTRO, R. M. G. *Introdução à Energia Eólica: Energias Renováveis e Produção Descentralizada*. 2009. (Instituto Superior Técnico), Lisboa. Segunda Edição.

RAGHEB, M. *Optimal Rotor Tip Speed Ratio*. 2018.

HIRAKAWA, A. R.; CUGNASCA, P. S. *COMUNICAÇÃO SERIAL*. 2014. <http://sites.poli.usp.br/d/pcs2529/index_arquivos/2529e062014.pdf>, acessado 02/10/2020.

ENGINEERS. *RS-485 Serial Communication General Concepts*. 2020. <<http://www.ni.com/white-paper/11390/en/>>, acessado 03/10/2020.

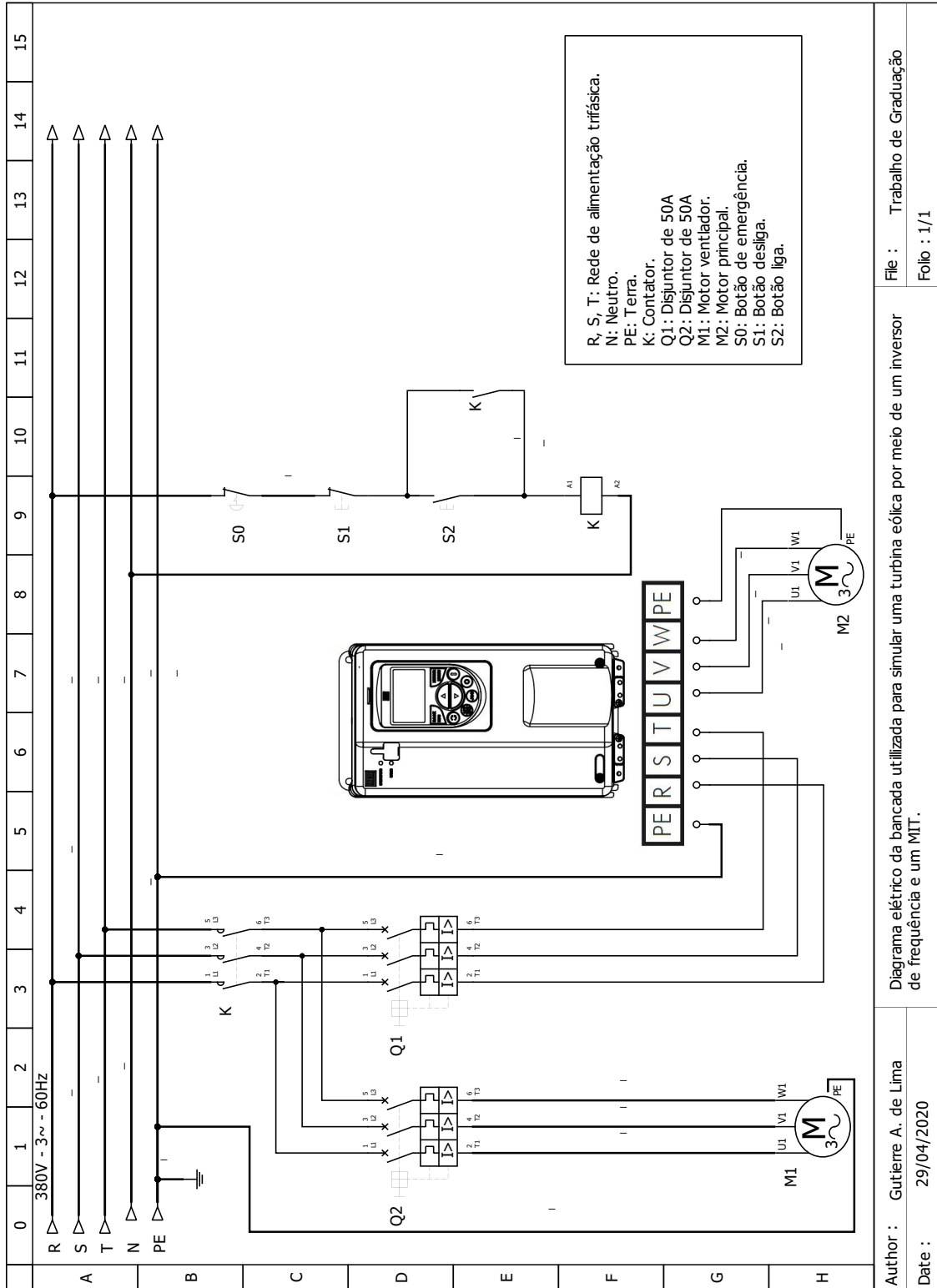
BEZARRA, M. anderson A. *Protocolos de comunicação aplicados em FPGAs*. 2019. <<http://www.dee.ufc.br/anexos/TCCs/2013.2/MARCUS20ANDERSON20ALMEIDA20BEZERRA.pdf>>, acessado 03/10/2020.

MECAWEB. *Porta Serial*. 2019. <http://www.mecaweb.com.br/electronica/content/e_serial>, acessado 10/11/2020.

ANEXOS

Neste capítulo, segue o diagrama elétrico (ANEXOS I), que descreve a bancada e o código de parametrização e configuração do inversor de frequência (ANEXOS II), desenvolvido em linguagem de programação Python.

I. DIAGRAMA ELÉTRICO



Author : Gutierre A. de Lima

Date : 29/04/2020

Diagrama elétrico da bancada utilizada para simular uma turbina eólica por meio de um inversor de frequência e um MIT.

File : Trabalho de Graduação

Folho : 1/1

II. CÓDIGO EM PYTHON

```
1 #Used libraries:
2 import serial
3 import minimalmodbus
4 import tkinter
5 from tkinter import * #c digo base
6 from tkinter import messagebox
7 from tkinter import Entry
8 import time
9 from tkinter import filedialog
10 import matplotlib.pyplot as plt
11 import csv
12 import tkinter.filedialog as fdlg
13 import matplotlib.animation as animation
14
15 #Communication through the Tkinter library:
16 def click():
17     try :    #Get error exception.
18         instr = minimalmodbus.Instrument('COM3',1,'rtu')
19         instr
20         instr.mode = 'rtu'
21         instr.debug = True #Shows information about reading.
22         instr.serial.baudrate = 9600
23         instr.serial.bytesize = serial.EIGHTBITS
24         instr.serial.parity = serial.PARITY_NONE
25         instr.serial.stopbits = serial.STOPBITS_ONE
26         instr.serial.timeout = 1
27         messagebox.showinfo('Communication','Connection successful.')
28         caixa=Frame(janela, borderwidth=3, relief=GROOVE)
29         caixa=LabelFrame(text='Connection status')
30         caixa.place(x = 20, y = 420)
31         lb2 = Label (caixa,width=20,height=3, font=("Tahoma", 10),
32                     background="white",fg="blue", text="SerialException:\
33                     nConnected", anchor=W, justify=LEFT)
34         lb2.pack()
35     except EnvironmentError:
36
37         messagebox.showerror("error","Error serial.serialutil.
38             SerialException: could not open port 'COM3'")#Mensagem de
39             erro
40         caixa=Frame(janela, borderwidth=3, relief=GROOVE)
41         caixa=LabelFrame(text='Connection status')
42         caixa.place(x = 20, y = 420)
```

```

39         lb2 = Label (caixa,width=20,height=3, font=("Tahoma", 10),
40                     background="white",fg="red", text="SerialException:\
41                     nConnection failed", anchor=W, justify=LEFT)
42         lb2.pack()
43
44     return
45
46 #Main window, CFW-11 button:
47 def click2():
48     janela2 = Tk()
49     janela2.title("CFW-11")
50     janela2.geometry("800x500+220+135")
51     caixa4=Frame(janela2,borderwidth=3, relief=GROOVE)
52     caixa4=LabelFrame(janela2,text='Menu')
53     caixa4.place(x = 15, y = 5)
54     lb23 = Label (caixa4,width=35,height=15, background = 'white', text='
55     Modelo: CFW11 0058 T4\nModelo: CFW11 0058 T4\nModelo: CFW11 0058
56     T4')#, font=("Tahoma", 10),fg="red",anchor=W, justify=LEFT)
57     lb23.pack()
58     janela2.mainloop()
59
60     return
61
62 #Selection button and secondary windows:
63 def select():
64     global selection
65     a=10
66     selection = var.get()
67     janela3 = Tk()
68     janela3.geometry("350x300+450+200")
69     def cancel(): #Cancel. Go back to the previous page.
70         janela3.destroy()
71         janela.grab_release()
72     def search(): #Enables file and graphic plot search:
73         global x
74         global y
75         x = []
76         y = []
77         file = fdlg.askopenfilename()
78         with open(file,'r') as csvfile:
79             plots = csv.reader(csvfile, delimiter=',')
80             for row in plots:
81                 x.append(float(row[0])) #Vector x
82                 y.append(float(row[1])) #Vector y
83             print(file)
84         l1= Label(janela3, text=file, bg="red", fg="white").place(x =
85             30, y = 155, width=150)

```

```

81
82     return
83
84 #control through files
85     def submit():
86         global tempo
87         global n
88         print(x)
89         print(y)
90         instr = minimalmodbus.Instrument('COM3',1,mode='rtu', debug=True)
91             #Defini o de instr
92         instr.write_register(682, 0x0015, number_of_decimals=0,
93             functioncode= 6, signed=False) #Desabilita
94         time.sleep(0.05)
95         instr.write_register(682, 0x0016, number_of_decimals=0,
96             functioncode= 6, signed=False) #Habilita
97         time.sleep(2)
98         instr.write_register(683, 6376, number_of_decimals=0,
99             functioncode= 6, signed=False)
100         time.sleep(10)
101         instr.write_register(101, 20, 1)
102         instr.write_register(683, 4554, number_of_decimals=0,
103             functioncode= 6, signed=False)
104         time.sleep(10) #Fica por 5.55s
105         instr.write_register(682, 0x0016, number_of_decimals=0,
106             functioncode= 6, signed=False)
107         time.sleep(2)
108         print('FIM')
109         messagebox.askquestion('Speed Profile','Generate speed profile?')
110         fig = plt.figure()
111         ax = fig.add_subplot(1, 1, 1)
112         xs = []
113         ys = []
114         Temp = [10,10,20,20,30,40,40,60,70] #test
115         temperatura=80
116         tempo=0
117         n=0
118
119     return
120
121 if selection == 0:#Radio button Controle escalar
122     janela3.title("Communication > Scalar control")
123     caixa3=Frame( borderwidth=3, relief=GROOVE)
124     caixa3=LabelFrame( janela3, text='Menu', fg="blue")
125     caixa3.place(x = 15, y = 5)
126     lb23 = Label (caixa3, width=35, height=15)
127     lb23.pack()

```

```

122     Parameter = Label(janela3, text = "Par metro1").place(x = 30, y =
        30)
123     Valor = Label(janela3, text = "Valor1").place(x = 30, y = 55)
124     Parameter2 = Label(janela3, text = "Par metro2").place(x = 30, y
        = 80)
125     Valor2 = Label(janela3, text = "Valor2").place(x = 30, y = 105)
126     Label_File = Label(janela3, bg="white").place(x = 30, y = 155,
        width=150)
127     entry1 = Entry(janela3)
128     entry1.place(x = 130, y = 30, width = 65)
129     entry2 = Entry(janela3)
130     entry2.place(x = 130, y = 55, width = 65)
131     entry3 = Entry(janela3)
132     entry3.place(x = 130, y = 80, width = 65)
133     entry4 = Entry(janela3)
134     entry4.place(x = 130, y = 105, width = 65)
135     cancelar = Button(caixa3, width=10, text = "Cancelar", command =
        cancel).place(x = 140, y =180)
136     Envio = Button(caixa3, width=10, text = "Enviar",command=submit).
        place(x = 40, y = 180)#    submit
137     Browse = Button(caixa3, width=6, text = "Browse... ",command=
        search).place(x = 170, y = 130)
138     elif selection == 1:#R dio button, controle vetorial
139         janela3.title("Communication > Vector control")
140     janela3.mainloop()
141
142     return
143
144 #Main window:
145 janela = Tk()
146 bt = Button(janela, width=15, text = "Connect", command=click)
147 bt.place(x=20, y=15)
148 bt = Button(janela, width=15, text = "CFW-11",command=click2)
149 bt.place(x=160, y=15)
150 janela.title("Communication")
151 janela.geometry("800x500+200+100")
152 caixa=Frame(janela, borderwidth=3, relief=GROOVE)
153 caixa=LabelFrame(text='Control type', fg = "blue")
154 caixa.place(x = 20, y = 80)
155 lb2 = Label (caixa,width=40,height=6)#, font=("Tahoma", 10),fg="red",
        anchor=W, justify=LEFT)
156 lb2.pack()
157 var = IntVar()
158 R1 = Radiobutton(caixa, text='Scalar control', variable=var, value=0)
159 R1.place(x = 20, y = 5)
160 R2 = Radiobutton(caixa, text='Vector control', variable=var, value=1)
161 R2.place(x = 20, y = 35)

```

```

162 Enter = Button(caixa,width=10, text = "Select", command=select).place(x =
    110, y = 65)
163 caixa=Frame(janela, borderwidth=3, relief=GROOVE)
164 caixa=LabelFrame(text='Teste principal')
165 caixa.place(x = 20, y = 210
    )
166 lb2 = Label (caixa,width=40,height=10)#, font=("Tahoma", 10),fg="red",
    anchor=W, justify=LEFT)
167 lb2.pack()
168 Parameter = Label(caixa, text = "Par metro1").place(x = 20, y = 10)
169 Valor = Label(caixa, text = "Valor1").place(x = 20, y = 35)
170 Parameter2 = Label(caixa, text = "Par metro2").place(x = 20, y = 60)
171 Valor2 = Label(caixa, text = "Valor2").place(x = 20, y = 85)
172 caixa=Frame(janela, borderwidth=3, relief=GROOVE)
173 caixa=LabelFrame(text='Main')
174 caixa.place(x = 350, y = 80
    )
175 lb2 = Label (caixa,width=60,height=19)#, font=("Tahoma", 10),fg="red",
    anchor=W, justify=LEFT)
176 lb2.pack()
177 janela.mainloop()

```

Listing II.1: Código em Python.