

JOACY DE LIMA FREITAS JR.

**Proposta de um sistema híbrido de produção de energia para
veículos aéreos não tripulados**

Joacy de Lima Freitas Jr.

**Proposta de um sistema híbrido de produção de energia para
veículos aéreos não tripulados**

Tese apresentada à Faculdade de Engenharia
de Guaratinguetá da Universidade Estadual
Paulista, para a obtenção do título de Doutor
em Engenharia Mecânica.

Orientador: Prof. Dr. Teófilo Miguel de Souza
Co-orientador: Prof. Dr. Rodrigo Kuntz
Rangel

F866p	Freitas Junior, Joacy de Lima Proposta de um sistema híbrido de produção de energia para veículos aéreos não tripulados / Joacy de Lima Freitas Junior – Guaratinguetá, 2020 142 f. : il. Bibliografia: f. 113-117 Tese (doutorado) – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2020. Orientador: Prof. Dr. Teófilo Miguel de Souza Coorientador: Prof. Dr. Rodrigo Kuntz Rangel 1. Inteligência artificial. 2. Energia elétrica – Produção. 3. Aeronaves. I. Título. CDU 007.52(043)
-------	--

JOACY DE LIMA FREITAS JÚNIOR

**ESTA TESE FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE
“DOUTOR EM ENGENHARIA MECÂNICA”**

**PROGRAMA: ENGENHARIA MECÂNICA
CURSO: DOUTORADO**

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

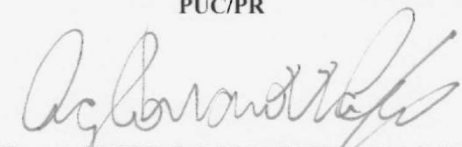

Prof.ª Dr.ª Ivonete Ávila
Coordenadora

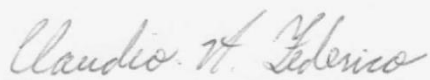
BANCA EXAMINADORA:


Prof. Dr. TEÓFILO MIGUEL DE SOUZA
Orientador / UNESP/FEG


Prof. Dr. JOSÉ RUI CAMARGO
UNITAU


Prof. Dr. OSIRIS CANCIGLIERI JUNIOR
PUC/PR


Prof. Dr. AGNELO MAROTTA CASSULA
UNESP/FEG


Prof. Dr. CLAUDIO ANTONIO FEDERICO
IEAv/DCTA

Fevereiro de 2020

DADOS CURRICULARES

JOACY DE LIMA FREITAS JR.

NASCIMENTO

09.12.1959 – SÃO VICENTE/SP

FILIAÇÃO

JOACY DE LIMA FREITAS

MARLENE PERES FREITAS

1978/1983

Graduação em Engenharia Eletrotécnica e Eletrônica na Faculdade de Engenharia de Barretos/SP.

2003/2005

Pós-Graduação em Engenharia Mecânica, nível de Mestrado na Faculdade de Engenharia do Campus de Guaratinguetá da Universidade Estadual Paulista.

2015/2020

Curso de Pós-Graduação em Engenharia Mecânica, nível de Doutorado na Faculdade de Engenharia do Campus de Guaratinguetá da Universidade Estadual Paulista

AGRADECIMENTOS

Agradeço a Deus acima de tudo.

Meu agradecimento especial à minha esposa, Maria Angélica, aos meus filhos Glaucia e Guilherme, e a toda minha família pela paciência e incentivo.

Agradeço ao meu Orientador, Dr. Teófilo Miguel de Souza, pela dedicação, pelo apoio irrestrito, pelo exemplo e pelo incentivo nos momentos difíceis, meu mais sincero obrigado.

Ao meu Coorientador, Dr. Rodrigo Kuntz Rangel, pelo suporte técnico, e pela ajuda nas diversas ocasiões que foram necessárias.

Ao Eng. Dr. Rosemberg André da Silva, que dedicou seus momentos de lazer para apoiar nas atividades dos softwares desenvolvidos.

À Empresa BRVANT, ao Instituto de Pesquisa IBRV, e aos amigos que neles trabalham, que sempre atenderam minhas solicitações demonstrando grande profissionalismo.

À ANAC, na pessoa do Gerente de Engenharia de Produto, Eng. Nelson Nagamine, que deu total apoio e incentivo.

A todos os funcionários da UNESP, Campus de Guaratinguetá, que direta ou indiretamente ajudaram a concretizar este trabalho.

RESUMO

Uma das grandes limitações hoje existente no uso de sistemas aéreos não tripulados com propulsão elétrica, é a pequena autonomia dos voos, o que estimula a pesquisa e desenvolvimento de meios alternativos que proporcionem uma forma de contornar o problema. Esta tese trata do projeto de um Sistema Híbrido de Produção de Energia, onde o controle desse sistema foi implementado com uma lógica PID e uma outra baseada em redes neurais artificiais. A necessidade imposta por alguns tipos de operações com RPA (*Remotely-Piloted Aircraft*), que exigem uma autonomia mínima, ou ainda que o RPA tenha um sistema alternativo dissimilar de produção de energia para atingir o nível exigido em termos de segurança de voo, são as motivações deste trabalho. Como requisito básico, foi feito uso de materiais e componentes COTS (*Commercial off-the-shelf*) a fim de viabilizar economicamente a proposta. Para testar o conceito, o sistema foi simulado e construído um protótipo em bancada, sendo que para os testes em voo será utilizado uma aeronave de decolagem e pouso vertical - VTOL (*Vertical Take-Off and Landing*) que está em fase de construção. Como parte dos trabalhos para se estudar a sua viabilidade, foram feitas simulações em MATLAB® Simulink. Esse método permite que se determine as características físicas e a eficiência do projeto antes que o sistema seja montado. A arquitetura é composta por um motor de combustão interna acionando um gerador para se obter a energia extra necessária para a expansão da autonomia. Apesar deste modelo não ser inédito e já ter sido estudado e testado em projetos anteriores alguns aspectos acrescentaram novos avanços, como neste caso a instalação de um controle inteligente baseado em dados treinados de uma rede neural, visando melhor rendimento e facilidade de alteração para aplicações futuras. O trabalho apresenta resultados positivos mostrando a factibilidade da proposta e abre caminho para o desenvolvimento de um controle mais sofisticado, com a inclusão do algoritmo de treinamento no próprio código, proporcionando o ajuste dos ganhos em tempo real.

PALAVRAS-CHAVES: Inteligência artificial. RPA. Sistema híbrido de energia. VTOL.

ABSTRACT

Today, one of the great limitations on the use of RPA (Remotely-Piloted Aircraft), is the small autonomy of the flights, which stimulates the research and development of alternative means that provide a way to get around the problem. This thesis deals with the design of a Hybrid Power Generation System, where the control was implemented with a PID logic and another control was based on artificial neural networks. The motivations of this work are the need imposed by the RPA's type of operation for a minimum autonomy, or even if the RPA has an alternative system of energy to reach a level required in terms of flight safety. As a basic requirement, the use of COTS (Commercial off-the-shelf) materials and components was sought to make the proposal economically viable. To test the concept, the system was simulated and tested in bench, and the next step will be the installation in the RPA type VTOL that is being developed. As part of the work to study its viability, simulations were made in MATLAB® Simulink. This method allows to determine the physical characteristics and the efficiency of the design before the system be built. The architecture is composed of an internal combustion engine driving a generator to obtain the extra energy needed to extend the system autonomy. Although this model is not unprecedented and has been studied and tested in previous projects, some aspects will add new advances, as in this case the installation of an intelligent control based on trained data of a neural network, aiming at better performance and ease of change for future developments. The work presents positive results showing the feasibility of the proposal and opens the way for the development of a more sophisticated control, with the inclusion of the training algorithm in the code itself, providing the adjustment of gains in real time.

KEYWORDS: Artificial intelligence. Hybrid energy system. RPA. VTOL.

LISTA DE FIGURAS

Figura 1 – Fontes de energia usadas em RPAs.....	23
Figura 2 - Exemplo de controle em malha fechada.....	28
Figura 3 – Diagrama de blocos da malha de controle PID.....	30
Figura 4- Modelo Computacional Discreto (genérico).....	31
Figura 5 –Aprendizado de máquinas e suas principais ramificações.....	34
Figura 6 - Representação de um neurônio biológico (esquerda) e de um artificial (direita) ...	35
Figura 7 - Exemplos de funções típicas de ativação de neurônios.....	36
Figura 8 - Exemplo de uma arquitetura de rede com uma única camada	37
Figura 9 – Esquema teórico do controle RNA.....	42
Figura 10 – Exemplo de Código do BPA.....	44
Figura 11 - Esquema PID-RNA usado em controles tipo ‘online’	46
Figura 12 – Fluxograma dos passos seguidos para execução do trabalho.....	48
Figura 13 – Gráfico – Limite de atuação Bateria-Gerador.....	49
Figura 14 – Gráfico da Corrente (a) e da Velocidade (b).....	50
Figura 15 – Gráfico da Corrente (a) e Altura do solo (b) nos três voos.....	51
Figura 16 – Gráfico da Corrente (a) x tempo (s).....	52
Figura 17 – Histograma da distribuição de corrente.....	53
Figura 18 – Gráfico do histograma e da distribuição normal... ..	53
Figura 19– Diagrama Simplificado do sistema desenvolvido.....	55
Figura 20 – Fotografia da bancada de testes com os componentes principais instalados.....	56
Figura 21 – Fotografia dos sensores instalados.....	57
Figura 22 – Curvas características de um motor a combustão equivalente, com ignição.....	58
Figura 23 – Fotografias dos motores Titan ZG 62SL (a) e Vulcan VR430 (b).....	60
Figura 24 – Força de Lorentz aplicada a um único fio num campo magnético uniforme.....	62
Figura 25 – Uma bobina de fio em movimento através de um campo magnético em transição.....	63
Figura 26 – Esquema de um MSIP.....	64
Figura 27 – Esquema de uma vista explodida do BLDC.....	65
Figura 28 – Circuito que simboliza de um gerador monofásico.....	67
Figura 29 – Forma de onda utilizada no comando do Servo.....	71
Figura 30 – Esquema elétrico do ACS712.....	71

Figura 31– Placa de controle modelo Mbed NXP LPC1768.....	72
Figura 32 – Pinagem da placa de controle.....	73
Figura 33 - Fotografia da placa utilizada no protótipo.....	74
Figura 34 – Esquema n. 1 de Coleta de Dados.....	78
Figura 35 – Localização dos Sensores no Sistema.....	78
Figura 36 – Gráficos rpm x t e V_g x t.....	79
Figura 37 – Gráfico da I_c x t.....	79
Figura 38 – Gráficos rpm x t e V_b x t.....	80
Figura 39 – Esquema n. 2 de Coleta de Dados.....	81
Figura 40 – Curva da Potência do Motor x Rotação.....	82
Figura 41 – Esquema de planta simulada no Simulink®	84
Figura 42 – Desenho detalhado do sistema.....	85
Figura 43 – Gráficos do primeiro esquema Simulado.....	86
Figura 44 – Desenho do segundo esquema Simulado com Bateria.....	87
Figura 45– Gráficos do segundo esquema Simulado.....	88
Figura 46 – Janela apresentando o worksheet do Matlab®.....	91
Figura 47 – Janela após a importação dos dados do arquivo.....	91
Figura 48 – Janela indicando a topologia utilizada.....	92
Figura 49 – Janela do <i>nn toolbox</i> , detalhando o treinamento da RNA.....	92
Figura 50 – Gráfico da Performance.....	93
Figura 51 – Gráficos da regressão dos dados após treinamento e validação.....	94
Figura 52 – Desenho da RNA com algoritmo MLP.....	96
Figura 53 – Fluxograma do software de controle do motor baseado no Sistema Especialista.....	97
Figura 54 – RNA aplicada no código RNA em C++.....	98
Figura 55 – Janela da ‘toolbox’ do Matlab®.....	99
Figura 56 – Plotagem dos dados de entrada e saída.....	100
Figura 57 – Janela do aplicativo indicando as várias estimativas realizadas com os dados...101	101
Figura 58 – Janela do aplicativo que descreve as estimativas testadas da FT.....101	101
Figura 59 – Curvas obtidas com 62,16% de fidelidade da curva do sinal de saída original...102	102
Figura 60 – Janela do aplicativo.....103	103
Figura 61 – Parâmetros ajustados conforme resposta desejada.....104	104
Figura 62 – Janela com os gráficos das curvas após ajuste.....105	105
Figura 63 – Fluxograma do software de controle do motor baseado em PI.....106	106

Figura 64 – Gráfico com os sinais V_g e I_c com o controle implementado.....	108
Figura 65 – Gráfico com detalhe ampliado.....	108
Figura 66 – Gráfico com os sinais V_g e I_c usando uma bomba d'água como carga.....	109
Figura 67 – Gráficos dos dados em voo.....	118
Figura 68 – Gráficos dos dados em voo.....	119
Figura 69 – Histograma dos dados da Corrente do voo n.3.....	119
Figura 70 – Esquema elétrico da placa de controle.....	122
Figura 71 - Esquema elétrico da placa LPC1768.....	122
Figura 72– Dados coletados da rotação do motor.....	127
Figura 73 – Dados da Corrente na carga e da Tensão no Gerador variando-se a Carga.....	127
Figura 74 – Esquema BLDC atuando como Motor de Partida.....	132

LISTA DE QUADROS E TABELA

Quadro 1 – Revisão Bibliográfica – Comparativo Resumido.....	25
Quadro 2 – Parâmetros de Controle PID.....	29
Quadro 3 – Diferença da Instalação Utilizada e a Final.....	40
Quadro 4 – Massa dos Componentes do Sistema Testado.....	56
Quadro 5 – Características Técnicas do Motor Titan ZG 62PCI-HV.....	59
Quadro 6– Características do motor Vulcan VR-430.....	60
Quadro 7 – Nomenclatura Adotada.....	66
Quadro 8 – Características do Motor/Gerador Mod. PROPDRIVE v2 5050 580KV.....	68
Quadro 9 – Tipos de sensores e limites operacionais.....	69
Quadro 10 – Características do Servo MG 995R.....	70
Tabela 1 – Classificação das Baterias LiPo em relação a tensão de operação.....	75
Quadro 11 – Descrição da Fonte tipo <i>Boost step-up</i>	75

LISTA DE ABREVIATURAS E SIGLAS

BLDC –Brushless DC Motor

BPA -Backpropagation Algorithm

COTS - Commercial off-the-shelf

DAQ - data acquisition

ECU - Electronic Control Unit

FADEC - Full Authority Digital Engine Control

IBRV – Instituto BRVANT de Pesquisa

LMIs - Linear Matrix Inequalities

LQG - Linear–Quadratic–Gaussian

MLP - Multilayer Perceptron

MTOW - Maximum Take-Off Weight

NASA - National Aeronautics and Space Administration

PI – Proporcional-Integral

PID – Proporcional-Integral-Derivativo

PID-RNA - Proporcional-Integral-Derivativo-Redes Neurais Artificiais

PWM – Pulse Width Modulation

RPAS – Remoted Piloted Aircraft System

RNA – Rede Neural Artificial

UAS – Unmanned Aerial System

UAV – Unmanned Aerial Vehicle

UGV - Unmanned Ground Vehicles

USV- Unmanned Surface Vehicles

VANT – Veículo Aéreo Não Tripulado

VTOL - Vertical Take-Off and Landing

LISTA DE SÍMBOLOS

H_{∞} - H-infinito

$G_c(S)$ – Transformada de Laplace do controlador

K_p - Constante proporcional

K_i - Constante integral

K_d - Constante derivativa

$R(S)$ - Transforma de Laplace do sinal de referência de entrada

$U(S)$ – Transformada de Laplace do sinal de saída antes da planta

$E(S)$ - Transformada de Laplace do sinal de erro

$G_c(S)$ - Transformada de Laplace da Função de transferência do controlador

$Y(s)$ – Transformada de Laplace do Sinal de Saída

Σ - Símbolo de somatória

b – Bias da rede *perceptron*

θ – Bias da rede neural

w – Peso na rede *perceptron*

x – Constante da camada de entrada na rede *perceptron*

∂ - Derivada parcial

ζ – Fator de amortecimento

k_e - constante fem

K_m - constante motor

ω_m - velocidade rotação

ω_e - velocidade elétrica

R - Resistência (ohm- Ω)

L - Indutância (H)

T - Torque (N.m)

H - rendimento

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS.....	16
1.1.1	Objetivo Principal	16
1.1.2	Objetivos Específicos.....	16
1.2	CONSIDERAÇÕES SOBRE O USO DE RPA	17
1.3	ORGANIZAÇÃO E ESTRUTURA DA TESE.....	19
2	FUNDAMENTAÇÃO TEÓRICA.....	21
2.1	INTRODUÇÃO.....	21
2.2	REVISÃO BIBLIOGRÁFICA.....	21
2.3	SISTEMA HÍBRIDO DE PRODUÇÃO DE ENERGIA (SHPE).....	24
2.4	SISTEMAS DE CONTROLE	26
2.4.1	Controle Clássico.....	26
2.4.1.1	Controle PID.....	27
2.4.2	Controles Inteligentes.....	32
2.4.2.1	Aprendizado de Máquinas e Redes Neurais Artificiais (RNA).....	33
2.5	TOPOLOGIAS ADOTADAS	39
2.5.1	RNA e o Algoritmo de Treinamento.....	42
3	COMPONENTES UTILIZADOS E MÉTODOS APLICADOS	47
3.1	INTRODUÇÃO.....	47
3.2	COLETA DE DADOS.....	50
3.2.1	Discussão sobre Autonomia de RPA	51
3.3	ARQUITETURA DO SISTEMA.....	54
3.3.1	Motor de Combustão.....	57
3.3.2	Gerador	61
3.3.3	Sensores e Servo.....	68
3.3.4	Placa Microcontroladora	72
3.3.5	Banco de Baterias	74
3.3.6	Fonte Boost.....	75
3.3.7	Retificador Trifásico.....	76
3.4	MÉTODOS APLICADOS	76
3.4.1	Descrição do funcionamento do SHPE.....	76
3.4.2	Coleta de Dados	77

3.4.3	Modelagem e Simulação	82
3.4.4	RNA desenvolvida.....	89
3.4.4.1	Dados de Entrada para RNA Multicamadas.....	89
3.4.5	Software com controle RNA	96
3.4.6	Controle PID	98
4	DISCUSSÃO DOS RESULTADOS	107
5	CONCLUSÃO E RECOMENDAÇÕES	111
5.1	TRABALHOS FUTUROS.....	112
	REFERENCIAS	113
	APÊNDICE A – Gráficos obtidos em voo do RPA do IBRV - Vide tópico 3.2.1....	118
	APÊNDICE B	120
	APÊNDICE C – Sistema Especialista – Vide tópico 3.4.4.	120
	APÊNDICE D – Esquema elétrico da Placa Desenvolvida – tópico 3.3.4.	122
	APÊNDICE E – Código do firmware desenvolvido – RNA.	123
	APÊNDICE F – Dados Coletados – Vide tópico 3.4.2.....	127
	APÊNDICE G – Relatório fornecido com as características da estimativa escolhida – Vide tópico 3.4.6.....	128
	APÊNDICE H – Código do controle tipo PID linguagem C++ – Vide tópico 3.4.6.	129
	APÊNDICE I – Esquema BLDC atuando como Motor de Partida.....	132
	ANEXO A – Folha de dados do motor Titan ZG 62SL.	133
	ANEXO B – Dados do Motor Vulcan VR430H.	134
	ANEXO C – Datasheet do Microcontrolador Mbed NXP LPC1768.....	135
	ANEXO D – Tipos de Algoritmos de Aprendizado de Maquinas.....	138

1 INTRODUÇÃO

O RPA (*Remotely-Piloted Aircraft*) têm-se tornado muito popular nas últimas décadas, por vários motivos, dentre os quais pode-se destacar: avanço dos materiais empregados nas diversas partes do equipamento, como por exemplo novas baterias, com melhores características, motores elétricos de pequena dimensão e grande potência, melhorias nos materiais empregados na fuselagem, desenvolvimento da eletrônica embarcada e de câmeras miniaturizadas de excelente resolução, o que os torna atraentes para missões de monitoramento.

A principal motivação do presente trabalho é a contribuição para a melhoria da autonomia desses sistemas, proporcionando inclusive a possibilidade do uso desses equipamentos em novas aplicações, bem como o aumento na segurança de voo, uma vez que adiciona um sistema dissimilar de fornecimento de energia.

As contribuições mais relevantes neste trabalho estão no desenvolvimento de um controle inteligente para o Sistema de Geração Híbrida, e no uso inédito de uma placa de controle microprocessada como cérebro desse sistema de controle.

Esse controle mostra-se equivalente ao FADEC (*Full Authority Digital Engine Control*) adotado em aeronaves, sendo basicamente uma placa microprocessada, também conhecida como ECU (*Electronic Control Unit*), e seus respectivos sensores que monitoram todos os aspectos principais do controle. Esse controle se aplica a um motor de combustão interna não objetivando a propulsão, mas sim o acionamento de um gerador elétrico, de forma a proporcionar uma alternativa de alimentação do RPA em determinadas condições de demanda.

A relevância se explica pelo amplo uso desses modernos sistemas pilotados remotamente, em situações diversas, possibilitando a ampliação das aplicações em especial as que são dependentes de uma maior autonomia. Com este desenvolvimento expande-se as possibilidades para esta tecnologia, explorando um conceito novo na geração híbrida de energia.

Sistemas Híbridos de Produção de Energia (SHPE) estão sendo estudados em variadas versões, como por exemplo, uso de células combustíveis, painéis solares, e motores a combustão (LAPÉÑA-REY et al., 2017), (KHARE; SINGH, 2012).

A sofisticação desta proposta está em se conseguir aliar um sistema conhecido baseado em motor a combustão, com um controle inteligente alimentado por informações em tempo real das condições de consumo elétrico e ambiental.

1.1 OBJETIVOS

Nos sistemas com geração híbrida existentes comercialmente verifica-se um desperdício de combustível na fonte de energia instalada devido à ausência da capacidade de controle desse sistema. Isso provoca um aumento desnecessário do consumo, já que geralmente o sistema trabalha em máxima capacidade, além de induzir a um desgaste maior. A segurança de voo é outro fator fundamental para se obter a aprovação de projeto junto aos órgãos responsáveis, e isso também é alcançado pela proposta.

1.1.1 Objetivo Principal

O objetivo principal foi desenvolver um Sistema de Geração de Energia convencional que trabalhará em conjunto com o banco de baterias, e por isso foi chamado de Geração Híbrida. O maior diferencial em relação aos sistemas existentes é o tipo de controle desenvolvido, virtualmente inédito neste tipo de aplicação.

Esse trabalho procura explorar essa lacuna, introduzindo um novo conceito, explorando técnicas de controle inteligente, além do clássico controle PID.

1.1.2 Objetivos Específicos

Os Objetivos Específicos são:

- Aumentar da segurança nos veículos não tripulados, com a instalação de um sistema redundante, proporcionando uma alternativa em casos de emergência. Esse é um aspecto importante, que é resolvido implicitamente, e atende aos novos requisitos certificação de aeronavegabilidade tanto nacionais como internacionais. O Regulamento Brasileiro de Aviação Civil Especial nº 94/2017. (ANAC, 2018), é a norma de certificação nacional. Este Regulamento cria classes de acordo com o peso, como será explicado no tópico 1.2.
- Desenvolver um sistema modular capaz de se adequar a diferentes necessidades operacionais e de potência sem ter que alterar totalmente o controle;
- Estudar e implementar os conceitos relacionados com Aprendizado de Máquinas e Redes Neurais Artificiais (RNA);

- Implementar os softwares Weka® e MATLAB® Simulink para mineração de dados, definição da função de transferência e para simulação e modelamento;

1.2 CONSIDERAÇÕES SOBRE O USO DE RPA

Nas últimas quatro décadas, os RPA têm sido amplamente utilizados para operações militares que incluam rastreamento, vigilância, engajamento ativo com armas e coleta de dados aerotransportados (CHEN et al., 2016). A demanda por estes veículos também é maior em relação aos tripulados devido as suas vantagens inerentes. Estas vantagens incluem menores custos de fabricação e operação, flexibilidade na configuração para atender solicitações específicas dos clientes além de não arriscar a vida do piloto.

Para o uso militar, destacam-se, dentre outras, as seguintes aplicações (RANGEL, 2012):

- Monitoramento tático (Vigilância marítima, aérea e terrestre),
- Suporte aéreo de buscas e salvamento (Bombeiros),
- Repetidor de telecomunicações (enlace eletrônico) entre pilotos e,
- Alvo aéreo manobrável para treinamento de pilotos e equipe de solo.

Os números atuais indicam que os RPA para uso civil atualmente constituem 30% do mercado mundial (“The Drone Market and Industry Trends | Toptal”). O mercado de RPA foi de US 13,22 bilhões em 2016 e está projetada para atingir US 28,27 bilhões até 2022 (MarketsandMarkets, 2019).

Outro fator determinante é a demanda crescente de RPA tipo VTOL (*Vertical Take-Off and Landing* - Decolagem e Aterrissagem Vertical) no setor civil e comercial. O mercado é impulsionado ainda mais pela demanda por mapeamento 3D, inspeção de gasodutos, atividades recreativas, segurança e vigilância (REPORT IDST, 2016).

A tecnologia também servirá para atender, dentro de um certo limite, um grande problema se observa hoje nas grandes cidades: a mobilidade urbana. Diversos meios de transporte têm sido estudados, com maior ou menor grau de interesse, objetivando aliviar a dificuldade cada vez maior das pessoas se locomoverem. Enfatiza-se que para um RPA com propulsão elétrica classe 1 o sistema aqui proposto não terá uma aplicação direta, devido à grande capacidade de energia das baterias, mas servirá como embasamento para um estudo do uso do controle inteligente em eventuais aplicações que o RPA necessite.

As grandes cidades são polos atrativos devido à maior oferta de empregos e serviços públicos, e isso por sua vez, agrava ainda mais a já difícil mobilidade das pessoas no dia a dia.

Nesse contexto, várias ideias surgem, mas que geralmente, não são capazes de proporcionar uma solução satisfatória a médio e longo prazo. Os espaços urbanos estão cada vez mais disputados, provocando grandes congestionamentos nas ruas e estradas sendo que os transportes coletivos muitas vezes não têm a capilaridade necessária para atender as necessidades das pessoas que moram cada vez mais longe do trabalho.

Existem projetos que buscam, devido a todos os problemas descritos, atender essa demanda usando o espaço aéreo. Pode-se citar o "*Uber Elevate*", com participação da Embraer (EMBRAER) 2019 , e outras grandes Empresas, como um dos projetos mais ambiciosos. Este programa tem pretensão de entrar em funcionamento já nos próximos anos (HOLDEN; GOEL, 2016).

A tendência da propulsão nos meios de transporte futuros indica que a energia elétrica será predominante, o que faz com que um meio dissimilar e redundante de alimentação seja fundamental. Esta opção pelo uso da energia elétrica proporciona vantagens como por exemplo, menor ruído, menos poluição, além de ser mais atraente em termos de manutenção. Um fator determinante na confiabilidade do projeto e consequentemente na sua aprovação junto as autoridades, seria a necessidade de possuir um sistema de alimentação *backup* capaz de auxiliar em caso de emergência. Uma vez que no VTOL com propulsão elétrica a alimentação dos motores é dependente diretamente da energia elétrica disponível, faz-se necessário aumentar a confiabilidade através de um sistema dissimilar alternativo. Veículos aéreos não tripulados com propulsão elétrica classe 1, que demandam o uso de muita energia, não tem no momento um sistema dissimilar de alimentação.

Nos aspectos técnicos, é fundamental que o novo transporte proporcione uma autonomia mínima para ser viável economicamente (RANGEL; TERRA, 2018), além do atendimento do conceito de segurança exigido pelas atuais normas nacionais e internacionais de certificação, como o Regulamento Brasileiro de Aviação Civil Especial nº 94/2017 (ANAC, 2018). Historicamente, verifica-se uma exigência cada vez maior por parte das autoridades de certificação, devido ao crescimento em tamanho dos RPAs.

A ANAC classifica o RPAS (RPA Sistema) e a RPA de acordo com o peso máximo de decolagem (PMD) da RPA da seguinte maneira (E94.5 Classificação do RPAS e da RPA): classe 1, classe 2 e classe 3.

(1) Classe 1: RPA com peso máximo de decolagem maior que 150 kg;

(2) Classe 2: RPA com peso máximo de decolagem maior que 25 kg e menor ou igual a 150 kg; e

(3) Classe 3: RPA com peso máximo de decolagem menor ou igual a 25 kg.

No SHPE proposto, o objetivo é a sua instalação em RPA classe 2 ou classe 3. Os RPAS Classe 1 devem obter um Certificado de Tipo conforme o RBAC 21, que possui regras bem mais rígidas. O RPA de classe 2, devido ao tipo de operação esperada e a sua dimensão e capacidade, sofre um processo de autorização de projeto ou de certificação mais cuidadoso e demorado do que o de classe 3. Neste sentido pode ser exigido, dependendo de uma série de fatores, sistemas de alimentação redundantes e/ou de emergência que sejam compatíveis com o tamanho da aeronave. Esta avaliação permite dizer que o emprego do SHPE ora proposto atuaria como uma opção válida para solucionar esta possível exigência.

Outros meios de energia para RPA tem sido pesquisado com maior ou menor grau de sucesso, como por exemplo, as células de combustível. Com a tecnologia atual, o único combustível que proporciona correntes de interesse prático é o hidrogênio, apesar de já existirem células que utilizam diretamente metanol como combustível. Mas, neste caso, as correntes obtidas ainda são relativamente baixas. Nos aspectos segurança e custo, esta tecnologia não se mostrou comercialmente viável, sendo complexa a sua manufatura e instalação (OZDEMIR et al., 2014).

Com a proposta ora em tela, fica explicitada sua importância neste cenário, corroborando para a evolução da tecnologia do uso de motores elétricos nos transportes, sendo, portanto, uma alternativa promissora, podendo assumir uma opção estratégica viável.

1.3 ORGANIZAÇÃO E ESTRUTURA DA TESE

Esta tese encontra-se organizada da seguinte forma:

No Capítulo 1 é feita uma introdução geral ao tema, mostrando sua relevância e motivação para o presente desenvolvimento. É acrescentado um breve histórico da tecnologia RPA, mostra sua importância e concluindo com a previsão de um forte aumento no uso desta tecnologia. Em seguida são listados o Objetivo principal e os Objetivos Específicos. Este Capítulo conclui com algumas considerações sobre o uso da tecnologia e uma descrição da estrutura adotada no trabalho.

No capítulo 2 é apresentada a fundamentação teórica dos assuntos. Neste capítulo é feita a Revisão Bibliográfica das tecnologias envolvidas, dando uma síntese das diversas áreas abrangidas, bem como descreve a abordagem adotada para o seu desenvolvimento. Foi

incluído um Quadro onde tem-se uma lista de artigos técnicos que mostram o estado da arte dessa área, comprovando o ineditismo deste trabalho. São feitas considerações sobre os tipos de controles estudados, sua origem, aplicações e sobre a topologia escolhida na execução do protótipo.

No capítulo 3 são descritos os componentes utilizados e métodos aplicados. São detalhados os equipamentos e hardwares que foram selecionados, o motivo das escolhas, os softwares utilizados, bem como a arquitetura do sistema como um todo. Os métodos são apresentados e discutidos. Neste Capítulo também foi incluído um fluxograma explicativo dos principais pontos seguidos na pesquisa, o que facilita a compreensão de cada passo.

No Capítulo 4 são feitas as discussões dos resultados alcançados, e no Capítulo 5 são feitas a Conclusão e as Recomendações, além de se incluir alguns pontos não realizados e julgados como importantes de serem abordados em trabalhos futuros.

Nos Apêndices foram inseridas informações complementares dos testes como gráficos e tabelas de dados coletados e do Sistema Especialista, com os códigos desenvolvidos.

Os anexos contêm documentos fornecidos pelos fabricantes e fornecedores dos componentes principais.

.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 INTRODUÇÃO

Neste capítulo será apresentada a fundamentação teórica que embasa esta Tese, juntamente com a revisão bibliográfica, uma síntese da história do emprego do RPA estendendo para os pontos que justificam este trabalho, uma breve descrição do sistema híbrido escolhido, bem como é feita uma introdução aos sistemas de controle.

2.2 REVISÃO BIBLIOGRÁFICA

O RPA é uma aeronave onde o piloto não está a bordo, mas a controla remotamente de uma determinada interface, que pode ser um computador, um celular, ou um controle específico geralmente eletrônico. Quando é mencionado RPAS, está-se falando do sistema, que é composto pela aeronave e pelo controle no solo.

Algumas Organizações Internacionais usam a nomenclatura VANT (veículos aéreos não tripulados) ou, em inglês *UAV (Unmanned Aerial Vehicles)*. Nesse trabalho será usada a nomenclatura RPA (Aeronaves Remotamente Pilotadas em inglês).

Historicamente, os veículos aéreos não tripulados existem desde o século XIX, tendo originalmente finalidade militar. O uso mais antigo registrado de um RPA remonta ao ano 1849, quando os austríacos atacaram a cidade italiana de Veneza usando balões não tripulados que foram carregados com explosivos (WATTS et al., 2012).

Embora os grandes desenvolvimentos na tecnologia devam-se ao emprego militar, comercialmente para uso civil hoje existem interesses econômicos impulsionando alguns avanços. Tem-se notícias de empresas propondo o seu emprego nas entregas de pacotes, execução de vídeos promocionais, inspeção de linhas de transmissão (SU et al., 2016), mapeamento, vigilância, aerofotografia, laser, finalidades agropecuárias, entre outros.

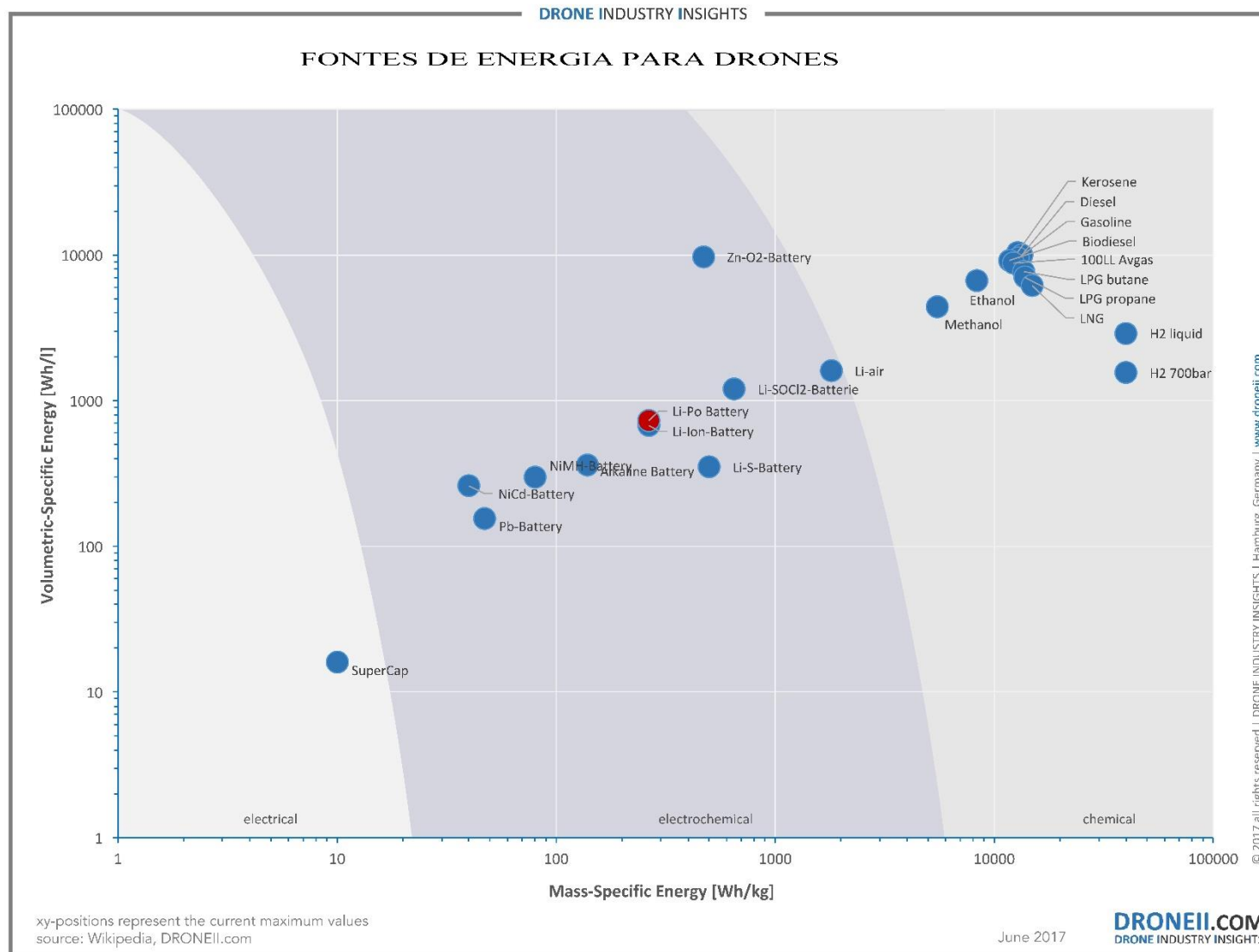
Os RPAs sempre tiveram seu uso pensado em diversos tipos de missões que são potencialmente perigosas podendo levar a algum tipo de risco as pessoas. Diversos avanços foram feitos nos sistemas remotamente pilotados, especialmente no enlace de controle e na forma de propulsão do RPA. As principais formas de propulsão são os motores de combustão interna e os motores elétricos. As fontes de energia para os propulsores acompanharam este desenvolvimento dentro de certo limite.

Avanços tecnológicos, especialmente na área de hardware, software e processamento de dados, vão continuar alavancando o comércio de RPAs (REPORT IDST, 2016). Novos tipos de fontes de energia, motores e materiais estruturais, juntamente com melhorias na área de softwares de controle, impulsionará o uso de RPAS ainda mais (SECRETARIA DE DESENVOLVIMENTO E COMPETITIVIDADE INDUSTRIAL; PLANEJAMENTO, 2017).

Com relação as baterias, as do tipo lítio-polímero (*Lithium-Polymer* (Li-Po)) e lítio-íon (*lithium-íon* (li-íon)) são as mais usadas. Novos compostos estão sendo estudados para se melhorar a relação de energia específica volumétrica (energia [Wh] por unidade de volume). Para ultrapassar os limites do desempenho de voo dos RPAs, as baterias deveriam ficar menores e mais leves.

Na Figura 1 são apresentadas as fontes de energia empregadas nos RPAs, em um gráfico que relaciona a massa específica de energia (Wh/kg) pelo volume específico de energia (Wh/l).

Figura 1 – Fontes de energia usadas em RPAs.



Fonte: Adaptado de Drone Industry Insights (2018).

Entre as configurações de RPA mais usadas pode-se listar os de asa fixa e os de asa rotativa, com multi-rottores. Esses últimos têm vantagens em algumas situações, uma vez que podem decolar e pousar em área limitada, voar em baixas velocidades e realizar voo pairado. Outro ponto importante, dependendo da aplicação, é a capacidade de ter grande manobrabilidade, permitindo a execução de trajetórias complexas, sendo que normalmente, consomem, mais energia, justamente por possuir mais motores.

Vale destacar que o desempenho do equipamento como um todo depende não somente da fonte de energia, mas também do sistema de propulsão. Um sistema de propulsão pesado não compensa uma fonte de energia eficiente, pois o resultado será inviável economicamente.

Dados sobre operação pretendida, autonomia mínima almejada, são os requisitos de projeto mais relevantes.

2.3 SISTEMA HÍBRIDO DE PRODUÇÃO DE ENERGIA (SHPE)

Esta pesquisa é parte de um projeto maior desenvolvido pelo IBRV. O projeto original que proporcionou este estudo é o desenvolvimento de uma plataforma de monitoramento hidro ambiental multiuso, capaz de servir de posto de observação e monitoramento avançado para a medição da qualidade situacional dos recursos hídricos brasileiros. A proposta é uma ferramenta alternativa, utilizando RPAs e drones aquáticos (*USV- Unmanned Surface Vehicles*), trabalhando juntos para obter informações sobre a qualidade da água, profundidade dos rios e Barragens, e imagens aéreas dessas áreas (RANGEL, 2019).

Considerando-se as grandes dimensões das represas e barragens, este tipo de operação necessita de uma autonomia de voo que não é atendida somente com a opção de uso do RPA com bateria. Com esta limitação evidente é possível dimensionar a importância que esta pesquisa tem, pois visa uma solução que atende este problema.

O RPA a ser construído será um VTOL multirotores, composto de um computador embarcado capaz de processar algumas ações e controlar a carga útil (conceito *Smart Drone*), e possui uma carga dedicada capaz de tirar fotos de alta resolução, usando câmera multiespectral. No Quadro 3 é apresentado o status do trabalho desenvolvido e do trabalho futuro a ser aLfigurlcanção. O RPA será utilizado para mapeamento das barragens e rios antes do uso do USV. Após o voo é criado um mosaico dessas áreas e o operador pode verificar o cenário atual para utilizar o USV.

Considerando as condições geográficas e de relevo, o quadricoptero é utilizado como relé de comunicação para aumentar o alcance operacional do sistema USV. Além disso, o

quadricoptero pode ser utilizado como câmera aérea para ver e controlar o USV, em tempo real (RANGEL, R.K. et al, 2019).

O conceito de se alimentar um RPA com um sistema de produção de energia híbrida está ganhando interesse da comunidade científica mundial. Pode-se verificar que existem vários projetos sendo desenvolvidos, com o foco não somente em RPA, mas também em veículos terrestres e marítimos. Como mencionado na introdução do Capítulo 1, atualmente são estudados e projetados vários tipos de fontes de energia, como por ex. células combustíveis, células fotovoltaicas e outros (LAPENÑA-REY et al., 2017), (KHARE; SINGH, 2012), (OZDEMIR et al., 2014), (OKUMUS et al., 2017). Cada sistema apresenta vantagens e desvantagens, mas o que predominou na escolha da tecnologia para este trabalho foi o fator econômico e segurança.

Observa-se várias Agências internacionais pesquisando sobre este assunto, podendo-se destacar o projeto da NASA (“NASA Flies Hybrid Electric VTOL - AVweb flash Article”), que está desenvolvendo o sistema híbrido *GL-10 Greased Lightning*, o qual, basicamente, através de dois motores diesel alimenta 10 conjuntos motopropulsores elétricos.

No Quadro 1 estão listados alguns artigos científicos que exploram alguns pontos e alternativas ao trabalho desenvolvido nesta Tese. Embora abordando alguns aspectos levantados e utilizados nesta proposta estes artigos confirmam o ineditismo tanto do Controle quanto da Aplicação.

Quadro 1 – Revisão Bibliográfica – Comparativo Resumido.

(continua)

REFERÊNCIA	COMENTÁRIO
(SUI; JIAO, 2015)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(CETIN; IPLIKCI, 2015)	Utiliza auto ajuste de PID através de um modelo preditivo baseado em discretização
(SHARMA et al., 2016)	Utiliza um controle adaptativo em um manipulador robótico
(SRIVIGNESH et al., 2012a)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(KHARE; SINGH, 2012)	Modela e otimiza um SHPE para USVs – com arquitetura diferente da proposta neste trabalho
(SHARMA et al., 2016)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(CHEN; HUANG, 2004)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(BADR, 1997)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho

Quadro 1 – Revisão Bibliográfica – Comparativo Resumido.

(conclusão)

REFERÊNCIA	COMENTÁRIO
(PATEL; KUMAR, 2015)	Controlador Neuro-PID baseado Algoritmo de Propagação Reversa – aplicação diferente da proposta neste trabalho
(SURENDRAN; KUMAR, 2016)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(MOSAAD; SALEM, 2014)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(SHAHRAKI et al., 2009)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(RIVAS-ECHEVERRÍA et al., 2001)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(ZRIBI et al., 2015)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(HERNÁNDEZ-ALVARADO et al., 2016)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(BIDIKLI, 2018)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(SUZUKI et al., 2004)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(ROSALES et al., 2019)	Utiliza RNA na otimização de controle PID mas em aplicação diferente da proposta neste trabalho
(LAPENA-REY et al., 2017)	Descreve sistema híbrido de energia diferente da proposta neste trabalho
(KHARE; SINGH, 2012)	Descreve sistema híbrido de energia diferente da proposta neste trabalho
(AKTAS et al.)	Descreve sistema híbrido de energia diferente da proposta neste trabalho
(OKUMUS et al., 2017)	Descreve sistema híbrido de energia diferente da proposta neste trabalho

Fonte: Produção do próprio autor.

2.4 SISTEMAS DE CONTROLE

2.4.1 Controle Clássico

No mundo real os sistemas de engenharia são passíveis de serem perturbados seja por influência externa seja por ruído de medição, além de haver sempre diferenças entre os modelos matemáticos usados no projeto e no sistema executado. Por essa razão a robustez do controle é fundamental (OGATA, 2010).

Neste raciocínio, o controle é parte integrante do sistema projetado, com o objetivo de satisfazer os níveis de desempenho almejados ainda que em presença de interferências diversas, dinâmicas do sistema não previstas ou variações dos parâmetros de projeto.

Para alcançar este objetivo o modo mais usual é através da realimentação de sinal, apesar de que isso provoca um aumento na complexidade do projeto, bem como introduz a necessidade de sensores, aumentando o custo e introduzindo invariavelmente as preocupações quanto a estabilidade do sistema.

Até 1960, é o chamado período do controle clássico, com entrada única e saída única, baseados em técnicas de resposta em frequência, ajuste de parâmetros do controlador baseado em métodos gráficos como lugar das raízes, diagramas de Bode e Nyquist, ou usando técnicas da transformada de Laplace, para obter-se a performance desejada (OGATA, 2010). Técnicas de controle clássicas como Proporcional (P), Proporcional Derivativo (PD), Proporcional Integral Derivativo (PID) são conhecidas há muitos anos (ZIEGLER; NICHOLS, 1942).

De 1955 a 1980, tem-se a chamada era do Controle Moderno, com múltiplas entradas e múltiplas saídas, LQG (*Linear-Quadratic-Gaussian*), computadores digitais, com programação dinâmica e controle adaptativo (OGATA, 2010).

De 1975 até os dias de hoje, seria a fase pós-moderna, com controle robusto, Controle H_∞ , LMIs (*Linear Matrix Inequalities* - Desigualdades Matriciais Lineares).

Nas últimas décadas o controle robusto tem ganho importância maior exatamente porque os sistemas reais são incertos, variados no tempo, não-lineares e muitas dimensões, além disso, as medições contêm ruídos e erros. Outro ponto importante é que o uso de sistemas digitais para implementação de controladores provoca alguns problemas específicos, o que obriga a discretização das formas de onda (OGATA, 2010).

2.4.1.1 Controle PID

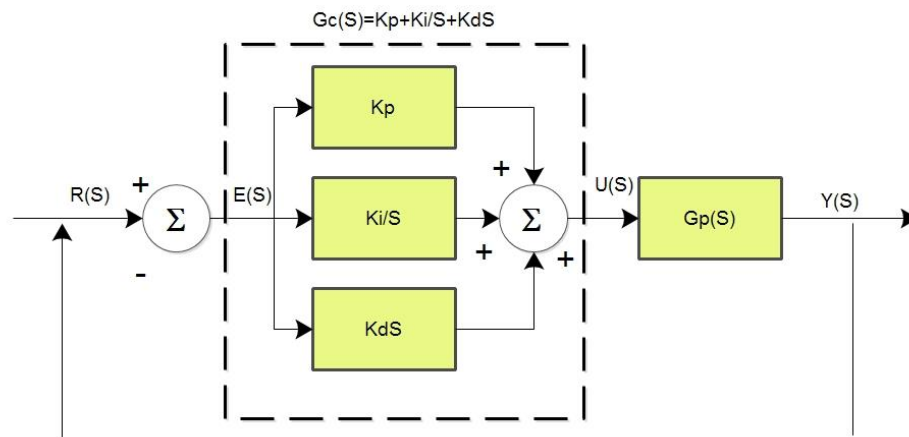
Apesar do surgimento de outros tipos de controle inclusive mais complexos, mais de 95% dos processos industriais ainda são controlados por PID (CHEN; HUANG, 2004).

A popularidade deste tipo de controladores advém da sua simplicidade (no máximo 3 parâmetros para sintonizar) e desempenho robusto. Nota-se que os processos a controlar poderão variar ao longo do tempo, com alterações de diversos tipos, como envelhecimento dos componentes, levando à necessidade de uma nova sintonia do controlador. Com isso em vista, os métodos de sintonia automática de controladores PID revelam-se de grande interesse tanto prático como teórico (CETIN; IPLIKCI, 2015). Esse automatismo é importante quando

se verifica que uma única unidade fabril pode ter centenas desses controles, que tem que ser sintonizados manualmente. Portanto torna-se um fator econômico sensível. Outro ponto de interesse crescente é que os controles automáticos levarão a uma resposta mais robusta, aumentando a qualidade dos processos, bem como a segurança (SRIVIGNESH et al., 2012b).

O diagrama básico de controle está representado na Figura 2.

Figura 2- Exemplo de controle em malha fechada.



Fonte: Produção do próprio autor.

O controlador PID tem três parâmetros, sendo um proporcional um integral e um derivativo, portanto usando-se a Transformada de Laplace, sua função de transferência é apresentada na equação 1:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d(s) \quad (1)$$

Onde:

$G_c(S)$ = a função de transferência do controlador;

K_p = o ganho proporcional;

K_i/s = Integrador e o ganho integral; e

$K_d s$ = Derivador e o ganho derivativo.

O procedimento de seleção dos parâmetros, K_p , K_i e K_d , do controlador PID de modo a serem atendidas as especificações de desempenho, é conhecido como sintonia

do controlador. A escolha adequada destes parâmetros torna possível o ajuste suave e preciso dos controladores.

Neste trabalho os ganhos foram determinados através da toolbox do Matlab PIDtoolbox, como explicado no item 3.4.6, mas a seguir será descrito um método pratico, chamado Ziegler-Nichols (ZN) para se exemplificar um modo alternativo.

Ziegler-Nichols (ZN) desenvolveram dois métodos para a determinação dos ganhos de controladores, um método em malha aberta e um método operando em malha fechada. Os Métodos foram desenvolvidos a partir da sintonia de muitas plantas diferentes, mas com características de respostas semelhantes. Citando o método em malha fechada, o critério de ajuste de parâmetros baseia-se na avaliação da amplitude e frequência das oscilações do sistema no seu limite de estabilidade, colocando o sistema em oscilação harmônica (ZIEGLER; NICHOLS, 1942). A ideia básica é obter valores para os ganhos a partir de experimentos sobre o processo. As regras são muito utilizadas, sempre que o processo a ser controlado permite, por fornecerem um ponto de partida para um ajuste mais fino.

Para um sistema em malha fechada, os valores dos ganhos integral e derivativo (K_i e K_d) são fixados em zero, e aumenta-se o ganho até o limite de estabilidade, a partir do qual o sistema em malha fechada começa a oscilar. Determina-se o valor do ganho K_{cr} (ganho crítico) que torna o sistema oscilatório e determina-se o período de oscilação P_{cr} (período crítico).

Os parâmetros do PID são ajustados conforme as regras mostradas no Quadro 2, a partir de K_{cr} e P_{cr} .

Quadro 2 – Parâmetros de Controle PID

Tipo de controlador	Função de transferência	K_p	T_i	T_d
P	K_p	$0,5K_{cr}$	máximo	0
PI	$K_p(1 + \frac{1}{T_i s})$	$0,45K_{cr}$	$\frac{P_{cr}}{1,2}$	0
PID	$K_p(1 + \frac{1}{T_i s} + T_d s)$	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

Fonte: Produção do próprio autor.

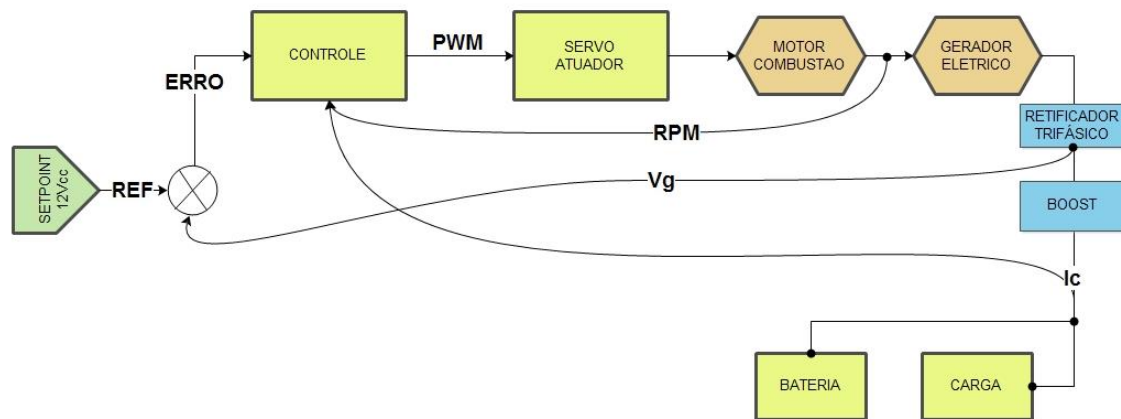
O método ZN tem como vantagens: não requer grande conhecimento técnico, não precisa simular ou modelar o sistema a ser controlado, geralmente é bom para se determinar

os parâmetros iniciais dos ganhos. Como desvantagens, não pode ser considerado um método robusto, que funciona para todos os sistemas e necessita que o sistema seja estável.

Neste trabalho, o controle do gerador é dado pela atuação do servo atuando no motor de combustão, em função principalmente da tensão do gerador (V_g).

O diagrama de blocos da malha de controle pode ser visto na Figura 3.

Figura 3 – Diagrama de blocos da malha de controle PID.



Fonte: Produção do próprio autor.

A dinâmica do controlador no domínio do tempo, considerando a atuação na rotação do gerador, pode ser vista na equação (2).

$$I = Kp * (Vg) + Ki * \int (Vg)dt + Kd * \frac{d(Vg)}{dt} \quad (2)$$

Um modelo computacional discreto (genérico), em linguagem C++ obtido a partir do controlador proposto é representado pela função PID (erro, I_c , K_P , K_I , K_D), e está detalhado na Figura 4.

Esse modelo serviu de base para o código desenvolvido que será visto no tópico 3.4.6. A variável ‘error’ representa o erro entre o estado desejado e o estado atual. A variável *Position* representa o “setpoint” ou posição desejada e as variáveis k_P , k_I e k_D representam os ganhos das malhas.

Figura 4 - Modelo Computacional Discreto (genérico)

```

// Classe PID
class PID{
public:
    double error;
    double sample;
    double lastSample;
    double kP, kI, kD;
    double P, I, D;
    double pid;
    double setPoint;
    long lastProcess;
    PID(double _kP, double _kI, double _kD){
        kP = _kP;
        kI = _kI;
        kD = _kD;
    }
    void addNewSample(double _sample){
        sample = _sample;
    }
    void setSetPoint(double _setPoint){
        setPoint = _setPoint;
    }
    double process(){
        // Implementação PID
        error = setPoint - sample;
        float deltaTime = (millis() - lastProcess) /
1000.0;

        lastProcess = millis();
        //P
        P = error * kP;
        //I
        I = I + (error * kI) * deltaTime;
        //D
        D = (lastSample - sample) * kD / deltaTime;
        lastSample = sample;
        // Soma tudo
        pid = P + I + D;
        return pid;
    }
};

```

Fonte: Simple PID Class for Arduino Projects - GitHub (2018).

Esse tipo de controle é amplamente utilizado na indústria de processos devido às suas boas características e dinâmica de resposta. No entanto, o controle PID também possui algumas limitações. Nos ambientes onde é necessária alta precisão, como robótica e manipulações guiadas, o fator de controle exige atenção e precisão especiais para os resultados desejados. Eles geralmente estabilizam sistemas lineares e precisam de modelos precisos que descrevam a dinâmica do sistema.

Existem sistemas nos quais os modelos são difíceis de se definir, pois as dinâmicas são desconhecidas e imprevisíveis e não podem ser modeladas com precisão, como no caso da planta deste trabalho. Fatores como ruído ambiental, saturação e desvios de parâmetros influem decisivamente no modelamento. Em alguns casos, variações repentinas dos parâmetros também tornam o sistema instável, como no presente estudo, e isso tem que ser adequadamente controlado. Neste trabalho tem-se um motor de combustão atuando ativamente no sistema e é sabido que suas propriedades mecânicas não são lineares.

A utilização de métodos clássicos para o projeto de controladores geralmente baseia-se em modelos matemáticos que usam equações diferenciais, as quais descrevem o comportamento do sistema de forma precisa.

A técnica usada nesta proposta foi de se fazer uma série de medições das grandezas físicas do sistema tanto de entrada como de saída, através de sensores, para se obter a função de transferência adequada (vide item 3.4.6). Usou-se para o presente sistema o software MATLAB®Simulink com a *toolbox* ‘*System Identification*’, determinando-se a função de transferência do sistema dinâmico a partir dos dados medidos.

2.4.2 Controles Inteligentes

Muitas tentativas de pesquisa surgiram com implementações de controles não clássicos, como por exemplo, modelo usando controle adaptativo, PID com controle através de lógica nebulosa (Lógica *Fuzzy*) e PID trabalhando com rede neural (AAMIR, 2013).

Nos últimos anos, a RNA foi aplicada com sucesso em muitos campos da engenharia, como p. ex. aeroespacial, processamento de sinal digital, eletrônica, robótica, visão de máquina, fala, fabricação, transporte, controles e engenharia médica (A. ZILOUCHIAN, 2001). Uma lista parcial de possíveis aplicações industriais das redes neurais inclui controle de temperatura, manipuladores de robótica, controle de servo motor, processos químicos, controle de qualidade em refinarias de petróleo, controles automáticos de voo e controle de pouso em aeronáutica, reconhecimento de imagem e muitos outros (A. ZILOUCHIAN, 2001).

Dentre os ditos ‘inteligentes’ existentes pode-se destacar: lógica fuzzy, redes neurais artificiais, controle genético, Máquinas de Vetores de Suporte (*support-vector machines* SVMs) e o controlador baseado em regras de decisão.

Neste trabalho foram utilizadas técnicas de controle clássico PID, e também técnicas de controle moderno, especificamente as RNA - Redes Neurais Artificiais, aplicando-se o algoritmo *Backpropagation* (YONEYAMA; JR, 2011).

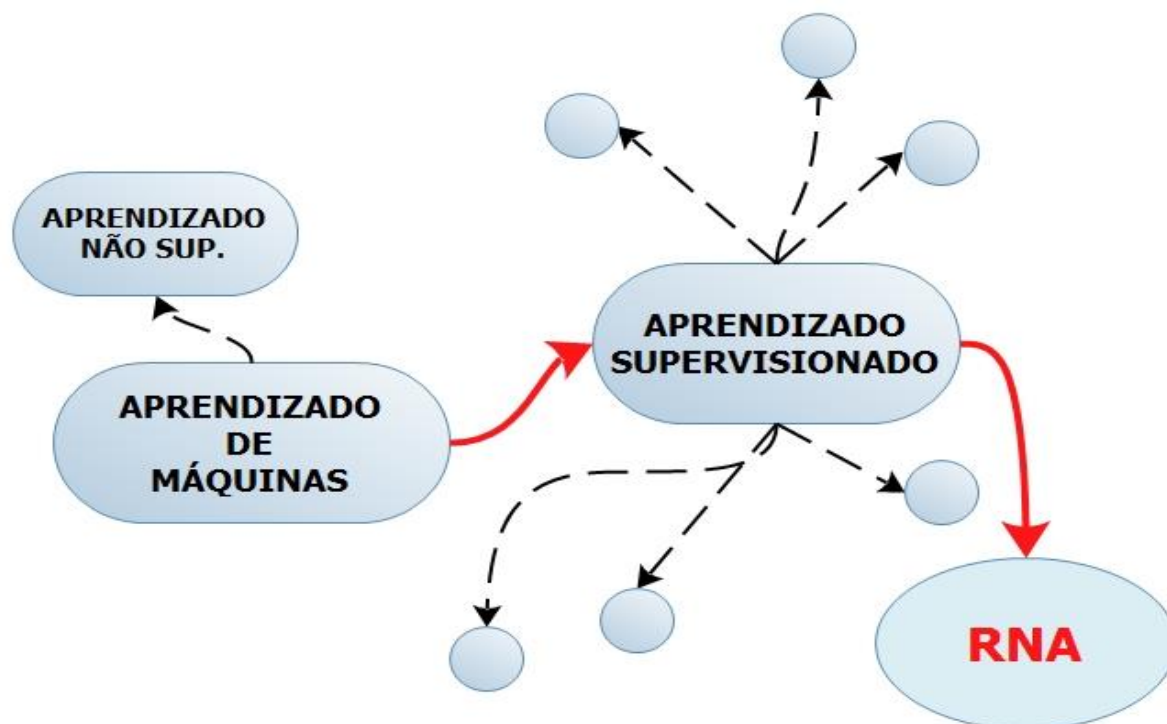
2.4.2.1 Aprendizado de Máquinas e Redes Neurais Artificiais (RNA)

Aprendizado de máquina é um campo da ciência da computação. É uma aplicação de inteligência artificial (IA) que fornece à máquina a capacidade de aprender e melhorar automaticamente a partir da experiência sem ser explicitamente programada. O aprendizado de máquina se concentra no aprimoramento progressivo de programas de computador que podem acessar dados e usá-los para aprender por si mesmos. O objetivo principal do aprendizado de máquina é permitir que os computadores aprendam automaticamente sem intervenção ou assistência humana.

O aprendizado de máquina pode ser categorizado em três tipos: Aprendizagem supervisionada, aprendizagem não supervisionada e por reforço. A variação no aprendizado de máquina é causada pela variação no método de treinamento. Será usado a aprendizagem supervisionada neste trabalho, que é semelhante ao nosso processo de aprendizado. No aprendizado supervisionado, a unidade de dados de treinamento consiste em entrada e saída correta. A entrada são os dados de treinamento e a saída correta é o que o modelo deve produzir para essa entrada específica.

Aprender nesse processo é como uma série de revisões de um modelo. Em cada revisão, a diferença entre a saída da malha e a saída correta é reduzida. Quando a diferença entre a saída da malha e a saída correta se torna insignificante, chama-se ‘treinamento apropriado’. Quanto menor a diferença, melhor a precisão do processo de treinamento. Na Figura 5 está desenhado um fluxograma com os principais modos de aprendizado de máquinas. A intenção é apresentar as subcategorias do aprendizado supervisionado o qual tem várias subáreas, como por exemplo ‘Árvores de decisão’ e outras, e que se optou por utilizar a RNA.

Figura 5 – Aprendizado de máquinas e suas principais ramificações.



Fonte: Produção do próprio Autor.

As redes neurais artificiais têm origem na teoria dos neurônios biológicos. Para entender uma rede neural artificial (RNA), é necessário fazer um paralelo entre um neurônio biológico e o neurônio artificial. As aplicações de RNA podem ser feitas por exemplo para a predição do comportamento de um processo ou para classificações de padrões, entre outras.

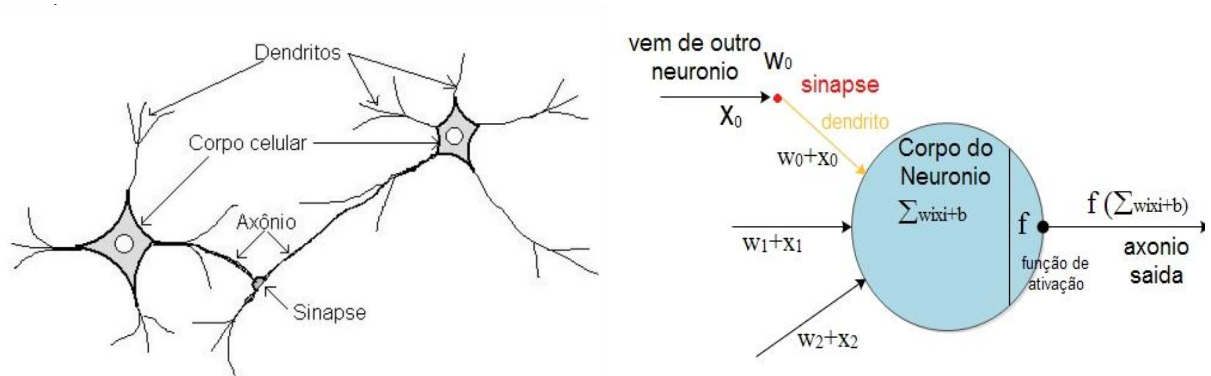
A RNA conta com inúmeros algoritmos para reconhecimento de padrões: Kohonen, *Perceptron*, *Adaline*, *Backpropagation* (BP), e outros, como pode ser visto no ANEXO D, e cada um com sua própria característica.

Um aspecto fundamental das redes neurais artificiais é o uso de simples elementos de processamento que são essencialmente modelos de neurônios no cérebro. Estes elementos são então conectados de maneira bem estruturada, embora a força e a natureza de cada um dos elos de conexão determinam o desempenho operacional geral da rede. Selecionando e modificando os pontos fortes do link de uma maneira adaptativa, pode-se fazer com que a base do aprendizado em rede seja formada ao longo do processo com informações da camada anterior (IRWIN et al., 1995).

Na Figura 6 está representado um neurônio biológico e um neurônio artificial. Um neurônio biológico é formado pelo corpo propriamente dito, pelo axônio que conduz a

informação de um neurônio para outro, por dendritos que são ramificações que comunicam os neurônios com as sinapses, que levam o sinal para o próximo neurônio.

Figura 6: Representação de um neurônio biológico (esquerda) e de um artificial (direita).



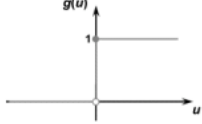
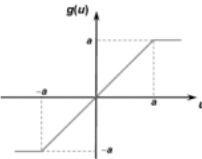
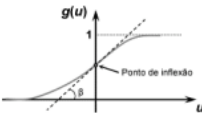
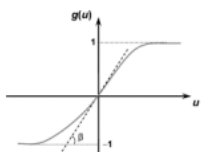
Fonte: Produção do próprio autor.

Com esta analogia, pode-se fazer um paralelo. No neurônio artificial os dendritos são representados pelos valores de entrada, os pesos são as sinapses, a função de soma faz o papel do corpo do neurônio e a função de transferência representa o axônio. A equação (3) é a função de ativação para obter a saída.

$$f(\sum w_i x_i + b) \quad (3)$$

A saída de um neurônio, mapeada por sua função de ativação, pode servir como entrada para outros neurônios, ou servir como resposta final na saída da rede. Existem várias funções de ativação, sendo que as principais estão listadas na Figura 7.

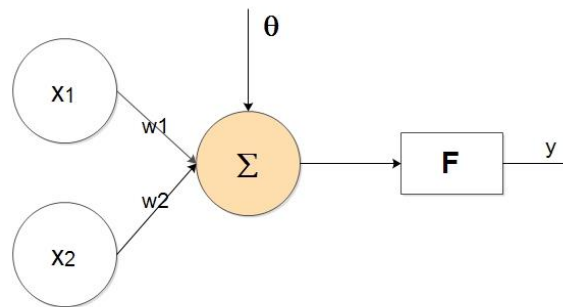
Figura 7: exemplos de funções típicas de ativação de neurônios.

Tipo	Equação	Função
Degrau	$\varphi(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases}$	
Degrau Bipolar	$\varphi(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases}$	
Rampa Simétrica	$\varphi(u) = \begin{cases} 1, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < -a \end{cases}$	
Logística	$\varphi(u) = \frac{1}{1 + e^{-\beta \cdot u}}$	
Tangente Hiperbólica	$\varphi(u) = \frac{1 - e^{-\beta \cdot u}}{1 + e^{-\beta \cdot u}}$	

Fonte: Produção do próprio autor.

Perceptron é um tipo de neurônio artificial de camada única e tem bom desempenho na classificação de dados lineares. Para classificações de dados que não são lineares, o *perceptron* tem um desempenho ruim. Na Figura 8 pode-se observar uma arquitetura *perceptron*, na sua forma mais simples. O *Perceptron* foi proposto por Rosenblatt (ROSEMBLATT, 1958) e *Adaline*, foi apresentada em 1960 por Widrow e Hoff (WIDROW, 1960).

Figura 8: Exemplo de uma arquitetura de rede com uma única camada.



Fonte: Produção do próprio autor.

Nesta Figura 8 é visto um exemplo básico onde o vetor x , é a entrada. A entrada aqui é um parâmetro ou dado que será usado para treinamento e teste. Enquanto $w1$ e $w2$ representam os pesos, o θ representa o viés (*bias*), e y é a saída. A entrada do neurônio é a soma ponderada das entradas mais o termo de polarização, como pode-se ver na equação (10). O *perceptron* usa a função de ativação para obter a saída.

$$y = F(\sum_{i=1}^2 x_i \cdot w_i + \theta) \quad (4)$$

Onde y é a saída da rede.

Em seguida é avaliado com uma regra que pode ser binária (função de ativação degrau-Figura 6):

1 se $net_j > 0$ e

-1 se $net_j \leq 0$.

A saída da rede, portanto, é +1 ou - 1, dependendo da entrada. A rede agora pode ser usada para uma tarefa de classificação: ela pode decidir se um padrão de entrada pertence a uma das duas classes. Se a entrada total for positiva, o padrão será atribuído à classe '+1', se a entrada total é negativa, a amostra será atribuída à classe '-1'. A separação entre as duas classes neste caso é uma linha reta, dada pela equação 5:

$$w_1 x_1 + w_2 x_2 + \theta = 0 \quad (5)$$

A rede de camada única representa uma função discriminante linear.

A equação (5) pode ser escrita como indicado na equação (6):

$$x_2 = \frac{-w_1}{-w_2} x_1 - \frac{\theta}{w_2} \quad (6)$$

É possível observar que os pesos determinam a inclinação da linha e a *bias* determina o quanto a reta está longe da origem. Com a regra de aprendizado *perceptron* tem-se um método para a rede determinar a *bias* e os pesos.

O algoritmo de aprendizagem do *perceptron* é um exemplo de aprendizagem supervisionada. O método de aprendizagem *perceptron* é um procedimento iterativo que ajusta os pesos. Uma amostra de aprendizagem é apresentada à rede. Para cada peso, o novo valor é calculado adicionando uma correção ao valor antigo. O limite é atualizado da mesma maneira como indicado nas equações (7) e (8):

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (7)$$

$$\theta(t+1) = \theta(t) + \Delta \theta(t) \quad (8)$$

Um algoritmo padrão para mostrar o funcionamento do *perceptron* (aprendizado supervisionado) é apresentado a seguir. Redes de apenas uma camada só representam funções linearmente separáveis, porém as redes de múltiplas camadas solucionam essa restrição.

Deve-se estender este raciocínio para uma estrutura mais complexa, para se obter soluções de problemas mais complexos. Uma estrutura multicamada, com número de neurônios diferentes podem ser combinados, formando uma rede *Perceptron Multicamadas* (*Multilayer Perceptron – MLP*).

A profundidade dessa rede, com as camadas ocultas, tem capacidade de aprendizado maior podendo levar a esquemas mais complexos.

No algoritmo listado abaixo está detalhada a lógica adotada no cálculo de erro e novo peso. É uma sequência ilustrativa, que serve de base para o entendimento da lógica envolvida em sistemas mais complexos, com muitas camadas.

A sequência de passos do algoritmo é a seguinte:

1. Declaração de variáveis – $D = [w_i, x_i]$ são dados de treinamento, y é o alvo desejado, peso inicial w , taxa de aprendizado e época máxima;
2. calcule v com a fórmula $v = (x * w)$;
3. calcular a saída com a função de ativação de saída;
4. se a saída = alvo, então continue a calcular o segundo dado;
5. se a saída $\neq$ alvo, então vá para o passo seis;
6. calcular sinal de erro, erro = alvo – saída;
7. calcular novos pesos, novos pesos = peso antigo + (taxa de aprendizado * erro * entrada);
8. de volta para o segundo passo para calcular os próximos dados com novos pesos;
9. parar - se o peso não mudar para todos os dados ou se a máxima época for atingida;

A popularidade do aprendizado on-line para o treinamento supervisionado de *perceptrons* multicamadas foi aprimorada pelo algoritmo de propagação reversa (*Back-propagation algorithm* - BPA).

O BPA foi originalmente introduzido há mais de 40 anos, mas teve sua importância revelada na década de 1980 (RUMELHART et al., 1986). A vantagem deste algoritmo é trabalhar mais rapidamente do que outros que existiam até então, e poder resolver problemas que não tinham solução.(NIELSEN, 2015).

Como pode-se notar, as redes *perceptron* de camada única são limitadas quanto às classificações que podem executar. Observa-se que nas redes multicamadas os trabalhos são computacionalmente completos, mas os primeiros pesquisadores, no entanto, não foram capazes de projetar um algoritmo de aprendizado para seu uso, principalmente pela escassez de recursos computacionais na época (GEORGE F LUGER, 2008).

2.5 TOPOLOGIAS ADOTADAS

Na pesquisa foi usado um motor tipo *Brushless* como gerador, ainda que esses motores e geradores sem escova são altamente não lineares e difíceis de se controlar, devido às suas fortes características eletromagnéticas não lineares. Como vantagens, tem uma estrutura simples e é muito confiável e, o mais importante, sua instalação é relativamente simples. Por isso, tem sido utilizado por um longo tempo.

Com o objetivo de detalhar a instalação construída para os testes desta tese e de se visualizar a instalação final a ser utilizada, criou-se o Quadro 3.

Quadro 3 – Diferença da Instalação Utilizada e a Final.

ITEM	INSTALAÇÃO ATUAL	INSTALAÇÃO FINAL
MOTOR UTILIZADO	VR 430	TITAN G62
POTÊNCIA DO CONJUNTO	1,25 kW	5 kW
INSTALAÇÃO	BANCADA	RPA
TIPO DE CONTROLE	PID E RNA COM TREINAMENTO OFFLINE	GANHOS PID CONTROLADOS PELA RNA
CARGA ELÉTRICA	REOSTATO E RESISTÊNCIAS	MOTORES DE PROPULSÃO
GERADOR	BLDC 2kW	BLDC 5 kW

Fonte: Produção do próprio autor

Foram desenvolvidos dois controles: um controle clássico PID, e um outro controle baseado em aprendizagem de máquina.

Métodos modernos de aprendizagem de máquina aplicados à teoria de controle são muito poderosos e se tornaram uma ferramenta importante para lidar com os sistemas de controle. Hoje essencialmente o que se faz para uma grande parte dos sistemas de controle é se parametrizar através de equações o comportamento dos sistemas.

No controle RNA desenvolvido, usou-se as medições feitas dos sinais através dos sensores para se obter alguns gráficos que representam fielmente o comportamento dinâmico do sistema. Criou-se um banco de dados o qual foi inserido nos softwares de treinamento para se obter o Sistema Especialista desejado. O passo final foi a criação do código com este sistema para uso na placa de controle.

No PID, obteve-se a função de transferência através de coleta de dados do sistema. Com a função de transferência obteve-se a melhor curva de controle no MATLAB®. Com estes ganhos definidos criou-se o código para inserção na placa de controle, como será visto no tópico 3.4.6.

Atualmente o controle moderno de aprendizado de máquina ou orientado a dados, tem sido muito difundido (BALAS, 2019). Considerando-se a evolução que tem havido nos sistemas de informática, hoje observa-se uma explosão de dados através dos recursos atuais de coleta de dados, proporcionando simulações de grande fidelidade, melhores técnicas experimentais de medição, com sensores cada vez mais poderosos, discos rígidos mais baratos e computadores mais rápidos. Ao invés de tentar desenvolver uma equação que represente o fenômeno a ser simulado e controlado, é muito mais fácil o levantamento de dados e começar

a usar as técnicas de aprendizado de máquina para caracterizar o sistema e começar imediatamente a trabalhar com o projeto de leis de controle desse sistema.

Nesse sentido, tudo indica que as técnicas orientadas a dados serão pelo menos uma parte da solução combinada com a teoria clássica dos sistemas dinâmicos no controle, pois não é razoável se desprezar o que se sabe sobre controle e sobre física, mas pode-se evoluir juntando-se a enorme quantidade de dados que hoje são coletados.

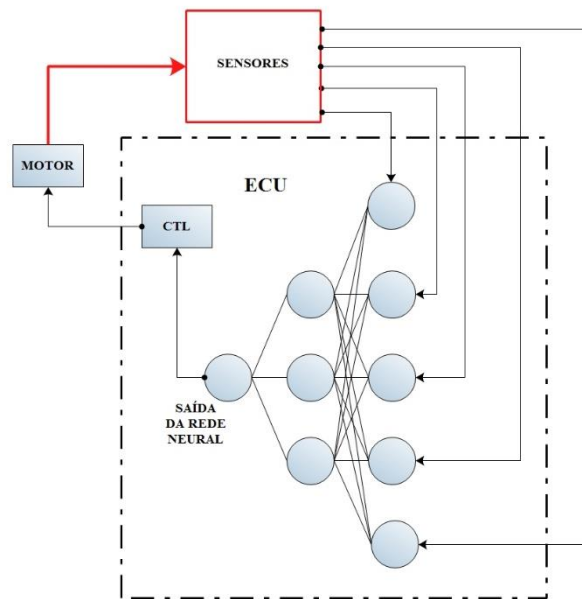
Quando se fala em controle ideal, o que se procura é fazer uma otimização para minimizar a função de custo, devido a alguns distúrbios e restrições que a dinâmica do sistema é forçada a satisfazer. Na maior parte do tempo, o que é feito agora é um controle preditivo do comportamento do modelo. É basicamente essa grande otimização on-line que se consegue com o aprendizado de máquina, que é um conjunto emergente realmente poderoso de técnicas de otimização não-linear. Nesta pesquisa também se usou o aprendizado de máquina para se fazer um dos controles do sistema gerador, buscando-se a otimização de um comportamento não linear, através de modelos orientados a dados. A ideia é ‘aprender’ a partir de dados coletados, de modo a usar isso para o aprendizado de máquina podendo-se modelar o sistema e projetar o controlador.

Sendo assim, as Redes Neurais Artificiais (RNAs) têm tido larga aceitação na tarefa de identificar e/ou controlar sistemas não-lineares (NARENDRA; PARTHASARATHY, 1990).

O algoritmo *Backpropagation* para redes neurais (BPNN) é amplamente conhecido por seu robusto mapeamento não-linear e capacidades adaptativas, sendo que sua estrutura e formulação é relativamente simples.

Foi utilizado uma configuração que consiste em se treinar uma RNA (*offline*) para que ela seja capaz de copiar o comportamento de resposta desejada para o sistema. A saída desejada para a rede neural corresponde à saída do controlador, como pode ser visto na Figura 9. Esse esquema representa teoricamente o que acontece no controlador. Na prática são usados os dados treinados da RNA para se formar uma tabela inserida no software, que fornece as respostas para o controlador.

Figura 9 – Esquema teórico do controle RNA.



Fonte: Produção do próprio autor.

As vantagens ocorrem porque o sistema especialista que foi treinado e validado, não sendo portanto necessário um ajuste manual como ocorreria em outros tipos de controle (IRWIN et al., 1995).

2.5.1 RNA e o Algoritmo de Treinamento

As redes neurais artificiais têm infinitas formas de arranjos de seus neurônios.

Dentre as principais propriedades das redes neurais pode-se destacar (WINSTON, 1993):

- **Aprendizado:** a rede é treinada a partir de exemplos ou tabelas.
- **Não linearidade:** um neurônio artificial pode ser linear ou não-linear, sua aplicação é adequada para sistemas físicos que geralmente são não lineares.
- **Capacidade de adaptação:** apresenta a propriedade de se adaptar a novas informações.
- **Capacidade de generalização:** uma rede neural possui uma capacidade de generalização do conhecimento adquirido a partir da existência de dados incompletos ou imprecisos.
- **Tolerância a Falhas:** é capaz de realizar computação robusta sob condições de operação adversas.

De um modo geral as redes neurais artificiais podem ser divididas em camadas.

A primeira camada de neurônios é denominada camada de entrada. Esta camada recebe as informações da entrada da rede, processa conforme seu padrão de funcionamento e sua saída é conectada à segunda camada. A segunda camada é denominada camada oculta. A quantidade de neurônios dessa camada pode variar de acordo com os resultados de testes, e não há um modo de se determinar antecipadamente. Pode haver mais de uma camada intermediária oculta, e a saída é a informação que será enviada para a camada de saída. Boa parte do processamento da rede neural é executado nessa(s) camada(s). A última camada, a de saída possui o número de neurônios necessários para gerar a informação desejada, e no caso da rede utilizada, ela coleta os dados da camada anterior e faz os cálculos de erro para fornecer o sinal de saída.

O processo de aprendizado pode ser supervisionado ou não supervisionado (IRWIN et al., 1995). O processo de aprendizagem não supervisionada consiste da auto-organização dos estados não exigindo os parâmetros de saída desejada, processo este análogo aos sistemas em malha aberta (OGATA, 2003). Tal processo é auto organizado e extrai as características de um dado conjunto de padrões, agrupando-o em classes.

O processo de aprendizagem supervisionado é análogo a um sistema em malha fechada (OGATA, 2003). Consiste em ajustar os pesos e o bias das entradas de tal forma a se obter a classificação desejada. Um padrão de dados de saída (ideal) é mostrado ao neurônio, a fim de se estabelecer um erro entre a saída real e a saída desejada. A aprendizagem é obtida por meio da diminuição (convergência) das distâncias das medidas (IRWIN et al., 1995). Em resumo, o aprendizado supervisionado fornece para a rede o conjunto de informações de entrada e indica a saída desejada para aquele padrão de entradas. A rede então passa a receber os valores na entrada e realiza o ajuste dos pesos dos neurônios para que a saída possa gerar um valor próximo ao desejado, que esteja dentro de uma margem de erro estipulada no projeto do sistema.

Para a previsão do comportamento do sistema de geração escolheu-se uma rede *feedforward* com ativação por função sigmoide. Esta classe de redes neurais é bem conhecida e relativamente bem compreendida além de serem facilmente implementadas sob condições de tempo real. Uma desvantagem é que o esforço de treinamento é geralmente alto, o que dificulta pois consome tempo para explorar as várias estruturas e otimizar a estrutura da rede. Apesar de que as camadas ocultas proporcionam melhor convergência no processo de treinamento optou-se neste trabalho por se usar três camadas: uma camada de entrada, uma camada oculta e uma camada de saída. Foi feita uma análise previa da aplicação de recentes

metodologias e algoritmos e apesar de algumas limitações conhecidas do BPA, como o longo processo de aprendizagem em alguns casos (SIMA; J., 1998) existem métodos para otimizar a taxa de aprendizagem (MIGUEZ, 2012).

Ao se optar por usar algoritmo *Backpropagation* (BPA) tem-se como vantagem a capacidade de trabalhar com multicamadas, sendo capaz de resolver problemas não lineares.

O método de aprendizado com a BP é um procedimento muito usado em controladores neurais (SUZUKI et al., 2004). Os sinais para a camada de entrada são a tensão do gerador, a rotação do gerador, a corrente na carga, a temperatura, a altura indicada.

O funcionamento do BPA baseia-se no cálculo do erro, deve-se recalcular o valor dos pesos do vetor ‘w’ (peso), atualizando todos os pesos das camadas a partir da última até atingir a camada de entrada da rede. Na Figura 10 está descrita um código do BPA, com detalhes dos cálculos envolvidos.

Figura 10 – Exemplo de Código do BPA.

```

function BACK-PROP-LEARNING(examples, network) returns a neural network
  inputs: examples, a set of examples, each with input vector x and output vector y
           network, a multilayer network with L layers, weights  $w_{i,j}$ , activation function g
  local variables:  $\Delta$ , a vector of errors, indexed by network node

  repeat
    for each weight  $w_{i,j}$  in network do
       $w_{i,j} \leftarrow$  a small random number
    for each example (x, y) in examples do
      /* Propagate the inputs forward to compute the outputs */
      for each node i in the input layer do
         $a_i \leftarrow x_i$ 
      for  $\ell = 2$  to L do
        for each node j in layer  $\ell$  do
           $in_j \leftarrow \sum_i w_{i,j} a_i$ 
           $a_j \leftarrow g(in_j)$ 
      /* Propagate deltas backward from output layer to input layer */
      for each node j in the output layer do
         $\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$ 
      for  $\ell = L - 1$  to 1 do
        for each node i in layer  $\ell$  do
           $\Delta[i] \leftarrow g'(in_i) \sum_j w_{i,j} \Delta[j]$ 
      /* Update every weight in network using deltas */
      for each weight  $w_{i,j}$  in network do
         $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$ 
  until some stopping criterion is satisfied
  return network

```

Fonte: Russell; Norvig (2010).

Na equação 9, está indicado o processo de aprendizagem. Cada peso é atualizado proporcionalmente a negativa da derivada parcial do erro com relação ao peso.

$$\Delta w_{ij} = -\mu \frac{\partial e}{\partial w_{ij}} \quad (9)$$

Uma vez calculado o vetor gradiente, atualizou-se cada peso de modo iterativo, diminuindo o erro a ponto de chegar abaixo de algum limiar preestabelecido, ou o número de iterações (*epoch*) atingir um valor máximo, finalizando o algoritmo.

A ideia central por trás dessa solução é que os erros das unidades da camada oculta são determinados pelo retro propagação dos erros das unidades da camada de saída. Por esta razão o método é frequentemente chamado de regra de aprendizado de propagação reversa.

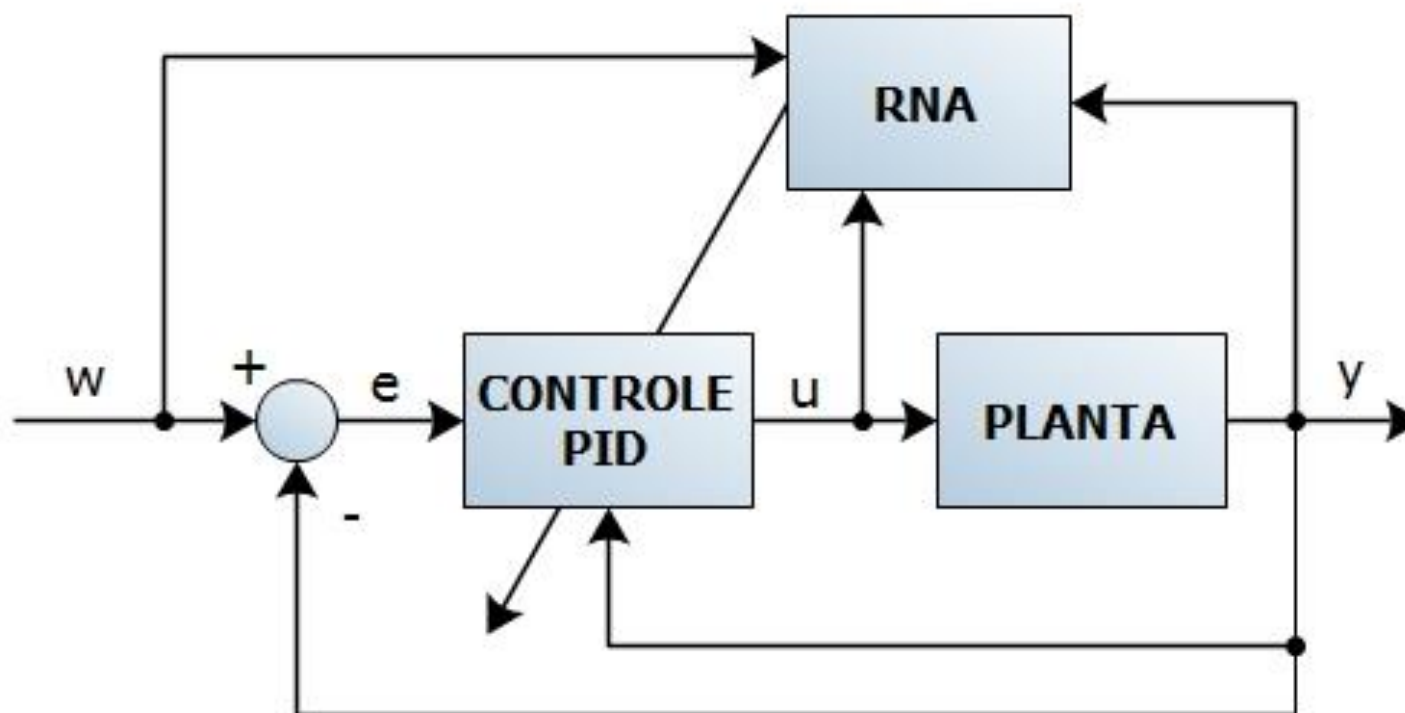
No modelo inteligente usado neste projeto os sinais de saída e de entrada da planta que está sendo controlada servem para formar a tabela que será inserida, após o processo de aprendizagem, no código do programa. Foi usada a função sigmoideal (em forma de S) também chamada ‘logística’. Essa função não linear é o tipo mais comum de ativação usada para se construir as redes neurais. É matematicamente ‘bem comportada’, e diferenciável, como desenhada na Figura 6 (logística).

Outro modelo de controle possível é o controle automático dos ganhos do sistema através de uma RNA. Devido as características de autoaprendizagem, auto-organização e auto adaptação, um controle RNA é capaz de identificar automaticamente os parâmetros do sistema controlado e ajustá-los de acordo com as alterações havidas. Neste tipo de controle tem-se duas ou mais malhas.

Configurações que ajustam online o sinal de controle são vistas com muitas variantes (SRIVIGNESH et al., 2012b), (PATEL; KUMAR, 2015), (CHEN; HUANG, 2004), e todas elas tem como objetivo um controle automatico que permite aumentar a performance do controle PID. O ponto em comum destas variantes tem como principio fundamental o prévio conhecimento da função de transferencia do sistema. Esse modelo não foi usado nesta pesquisa.

Para exemplificar, o diagrama de blocos de um modelo de controle PID-RNA multicamada online, pode ser visto na Figura 11. Esse tipo de controle exige uma definição exata da função de transferencia da planta.

Figura 11: Esquema PID-RNA usado em controles tipo 'online'.



Fonte: Yoneyama; Jr, Nascimento C. (2011).

3 COMPONENTES UTILIZADOS E MÉTODOS APLICADOS

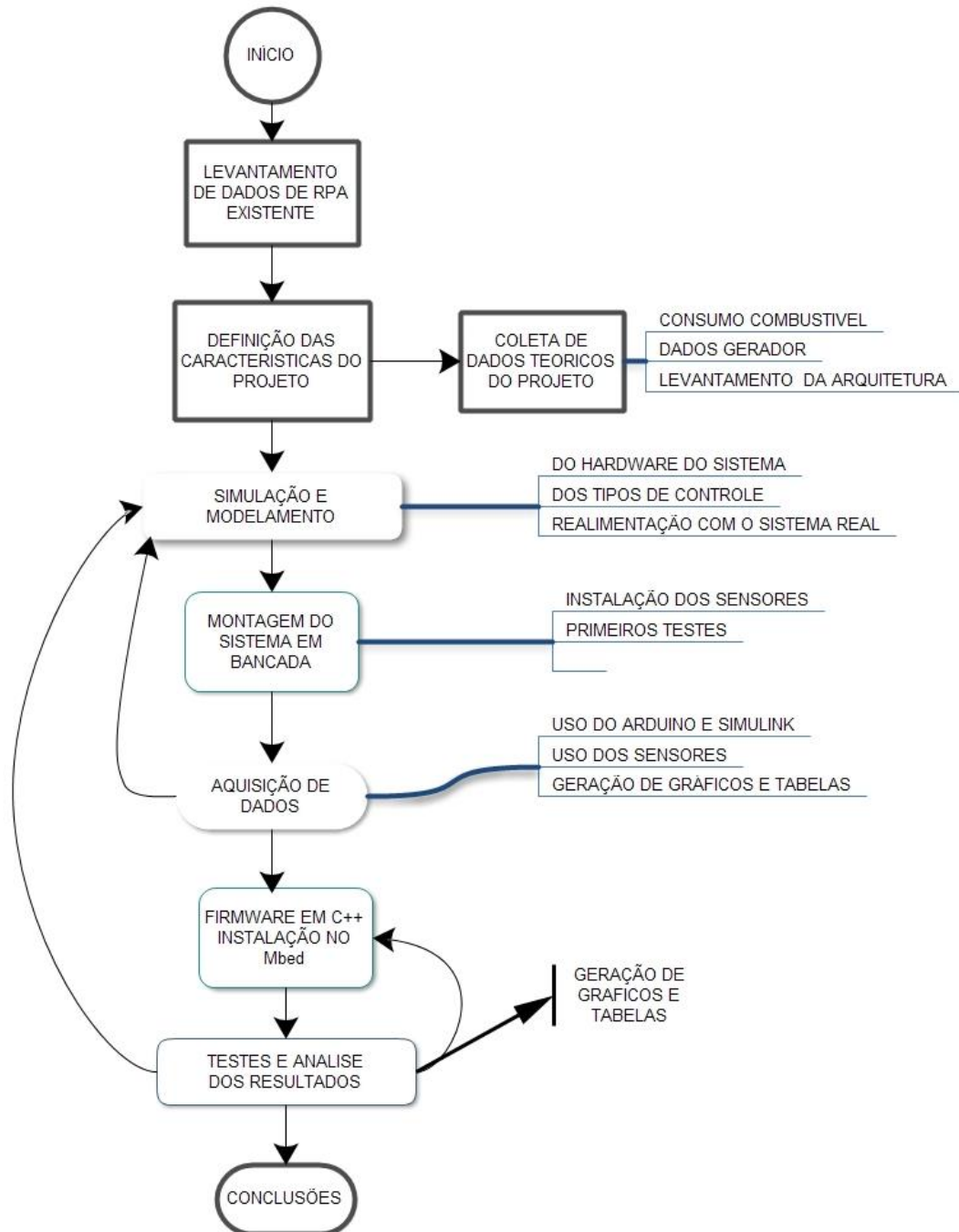
3.1 INTRODUÇÃO

Neste capítulo serão detalhados os Materiais e os Métodos empregados. A premissa básica para o projeto, foi o uso de componentes COTS (*Commercial off-the-shelf*), de forma a atender as necessidades de custo e prazo de execução.

A proposta envolveu também a execução de um protótipo capaz de comprovar a tese. No fluxograma da Figura 12, são indicados os passos principais seguidos desde o início.

Os próximos tópicos estão de acordo com a sequência mostrada neste fluxograma. Pode-se destacar a realimentação entre as fases de simulação e modelamento e entre a coleta de dados e os testes do sistema na bancada. Outro ponto é a realimentação entre as fases da instalação do programa no microcontrolador e os testes e avaliações dos resultados

Figura 12 – Fluxograma dos passos seguidos para execução do trabalho.



Fonte: Produção do próprio Autor.

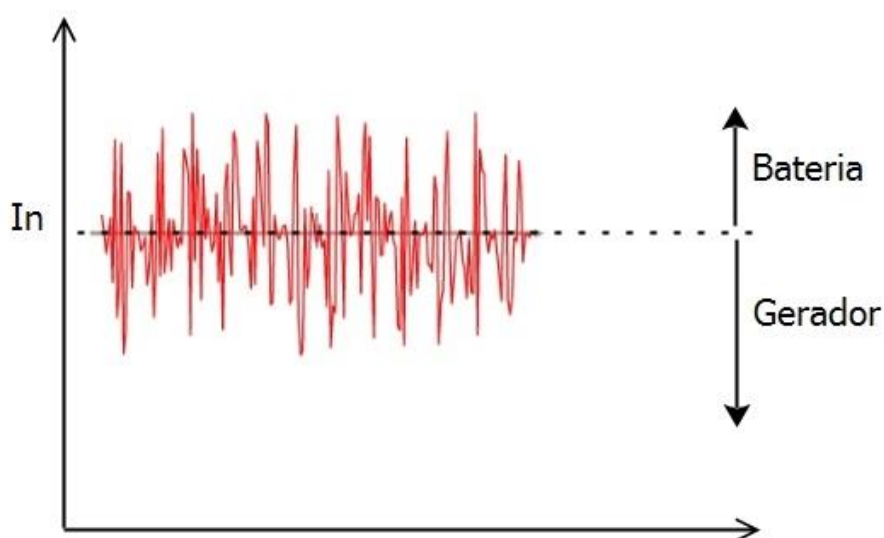
Como esta pesquisa é parte de uma pesquisa mais ampla, foram aproveitados, sempre que possível, os instrumentos e instalações já existentes na sede do Instituto para execução dos testes.

Algumas alternativas foram avaliadas na definição da arquitetura do sistema, como por exemplo, outros tipos de motores de combustão, bem como diferentes tipos de alternadores/geradores.

Optou-se pelo motor de combustão com ignição, a gasolina convencional, devido sua simplicidade e robustez, aliado ao baixo custo. Para a máquina geradora foi escolhido um BLDC atuando como gerador, acoplado a um retificador trifásico. Os sensores usados e o servo mecanismo para atuação no motor de combustão interna são modelos comuns de uso conhecido em robótica.

Na proposta de funcionamento do SHPE, o conjunto de Baterias assume automaticamente quando o valor I_n (corrente nominal pré-estabelecida) é atingido, sendo que abaixo desse valor o Gerador fornece a energia para alimentar as Cargas elétricas, conforme apresentado na Figura 13.

Figura 13: Gráfico – Limite de atuação Bateria-Gerador.



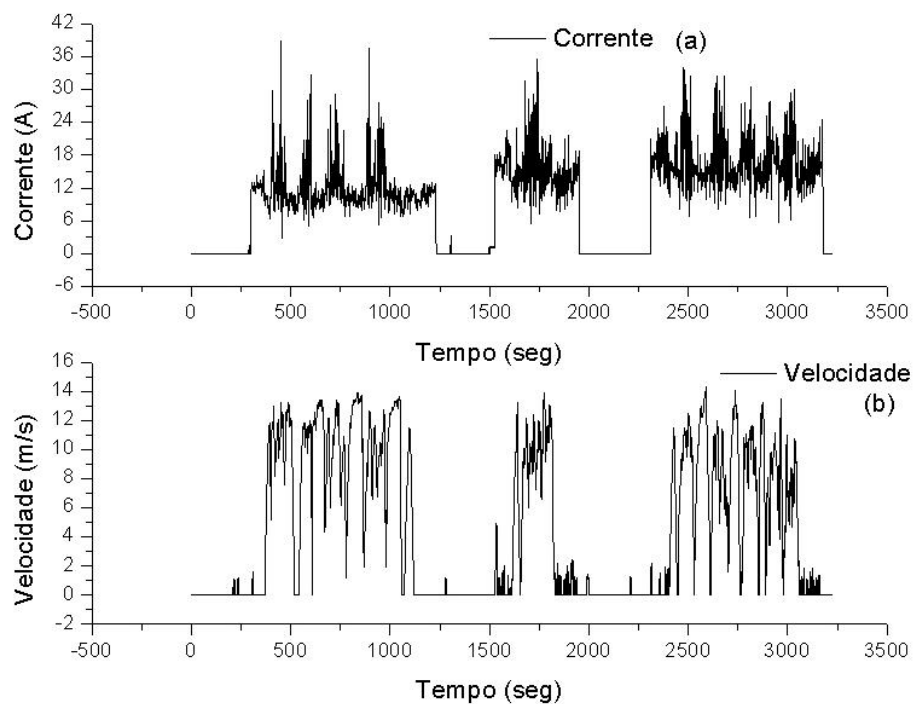
Fonte: Produção do próprio Autor.

3.2 COLETA DE DADOS

Foram feitos alguns voos com um RPA do Instituto IBRV, com o intuito de se coletar dados para a pesquisa. O perfil de consumo de energia é um dado fundamental que embasa e justifica este trabalho. Apesar de que o RPA usado nestes testes não possui as características necessárias para a instalação do sistema desenvolvido, os dados serviram de referência para se calcular a autonomia de um RPA de maior porte.

Os voos foram realizados em 27/08/2019. Na Figura 14 pode-se observar os dados coletados da corrente consumida e a velocidade durante os voos.

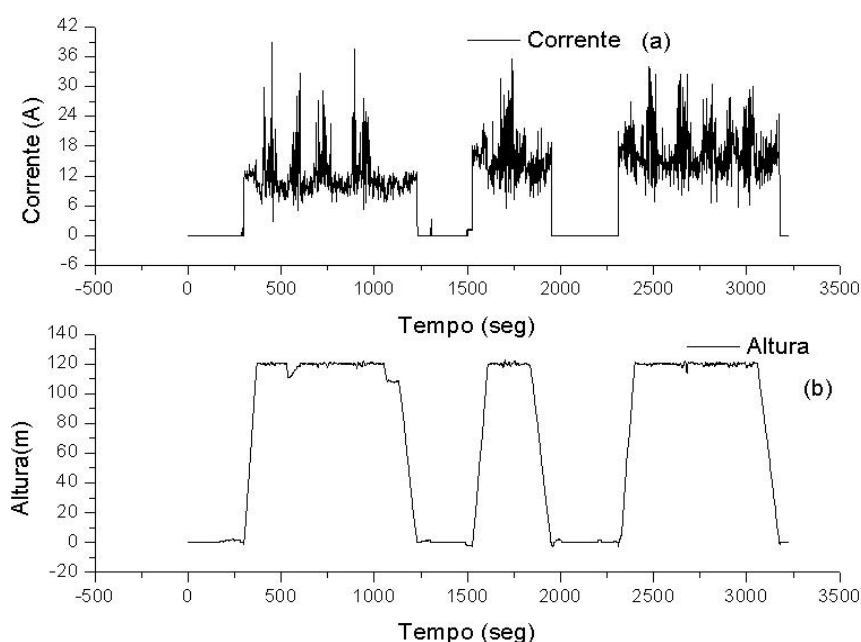
Figura 14 – Gráfico da Corrente (a) e da Velocidade (b).



Fonte: Produção do próprio Autor.

Observa-se nestes gráficos que foram feitos três voos, com perfis parecidos, sendo que o consumo de corrente está diretamente atrelado a velocidade do RPA. Na Figura 15, vê-se os gráficos, com os dados dos mesmos voos, dos dados de corrente e da altura em relação ao solo.

Figura 15 – Gráfico da Corrente (a) e Altura do solo (b) nos três voos.



Fonte: Produção do próprio Autor.

Através destes gráficos e das informações sobre as características técnicas do RPA utilizado no voo, foram tiradas algumas conclusões. Este RPA não possui as características necessárias para fazer os testes em voo com o SHPE, uma vez que tem um MTOW em torno de 3,5 kg, mas mostraram o perfil de consumo padrão. O objetivo da análise dos dados deste RPA é fazer uma pró-análise das características necessárias para uma futura aplicação do gerador híbrido.

Como previsto, a corrente consumida variou constantemente e abruptamente. Estas variações são explicadas pelas condições do vento e aceleração dos motores.

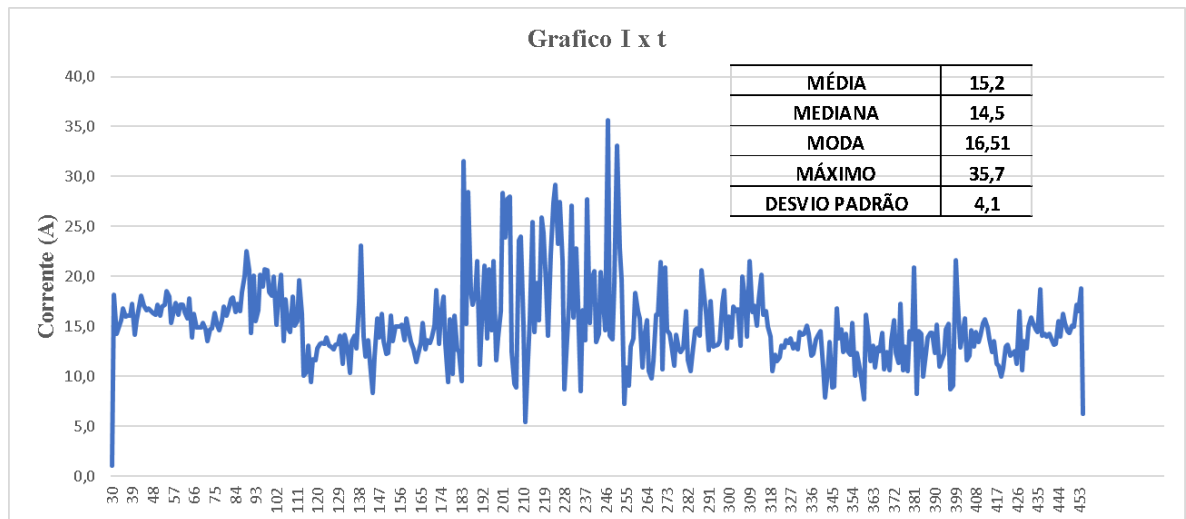
No apêndice A foram inseridos outros gráficos desses voos.

3.2.1 Discussão sobre Autonomia de RPA

Para o cálculo da autonomia estimada, foram considerados os dados do do segundo voo e extrapolados para um RPA maior, com MTOW estimado de 17,5 kg, ou seja, cinco vezes a capacidade do RPA monitorado. Escolheu-se os dados do segundo voo como referência, principalmente porque foi o voo mais curto, portanto com menos dados.

Na Figura 16, expandiu-se os dados de corrente do segundo voo, onde observa-se que a média ficou em 15,2 A e o pico de máxima corrente ficou em 35,7 A.

Figura 16 – Gráfico da Corrente (a) x tempo (s).



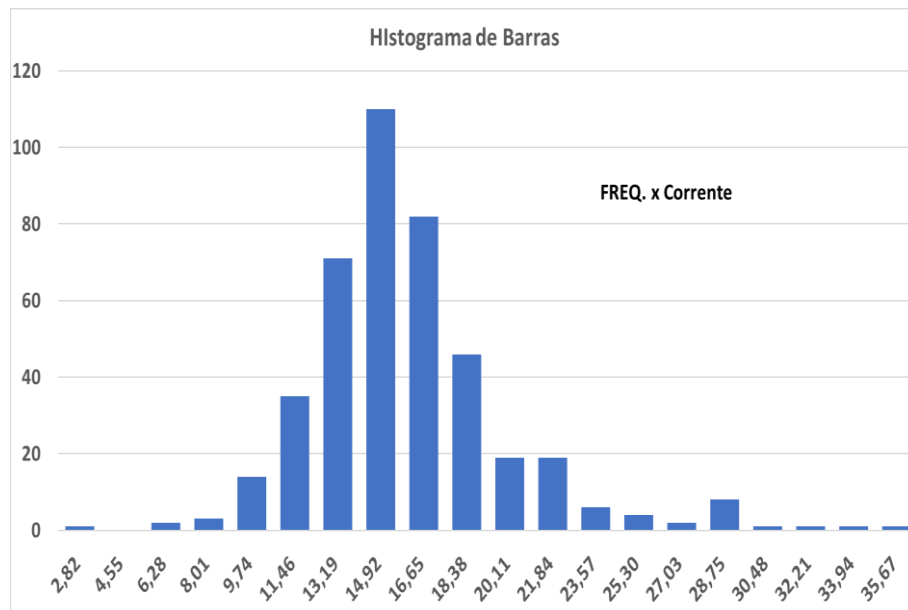
Fonte: Produção do próprio Autor.

Assumindo-se uma corrente média de 15,2 A, e extrapolando-se para RPA estimado a corrente consumida média seria de 76 A com picos de corrente de 178,5 A. Esse cálculo é aproximado pois depende de vários fatores, como já foi mencionado no tópico 3.2.

O aumento na autonomia dependerá de alguns fatores, como por exemplo, das condições de voo, quantidade de combustível condições operacionais.

Na Figura 17, desenhou-se o gráfico do histograma e a distribuição normal do conjunto de dados. Salienta-se novamente que o SHPE a ser construído no futuro deverá ter uma capacidade em torno de 5 kW, o que seria suficiente para alimentar o RPA.

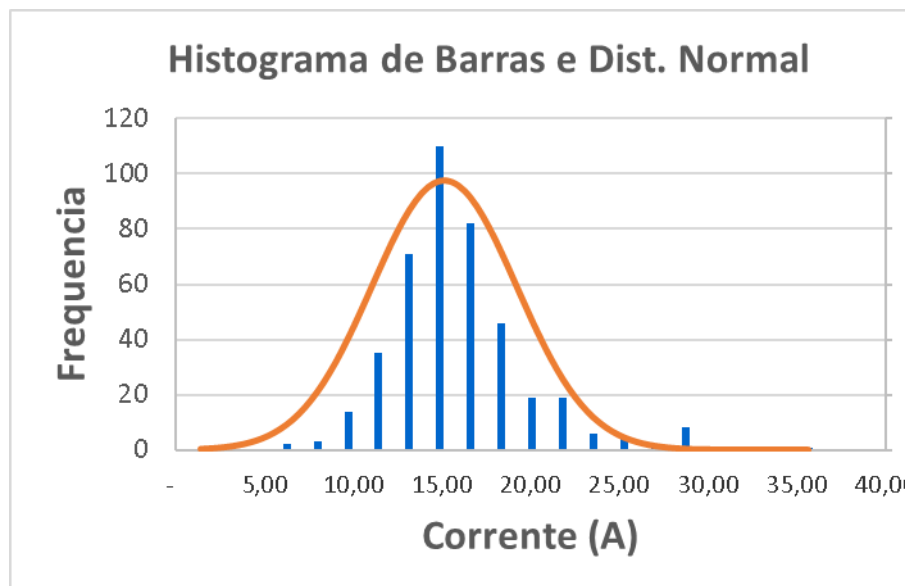
Figura 17 – Histograma da distribuição de corrente.



Fonte: Produção do próprio Autor.

Através da Figura 18, observa-se a distribuição normal desses dados.

Figura 18 – Gráfico do histograma e da distribuição normal.



Fonte: Produção do próprio Autor.

De acordo com as medições efetuadas, a média de consumo de corrente é de 15,16 A e o desvio padrão (σ) ficou em 4,08 A.

No projeto desenvolvido, a tensão nominal do conjunto de células é de 22,2 Vcc. Uma bateria LiPo consiste em várias células. Essas células fornecem aprox. 3,7 Volts cada uma.

A ‘taxa de descarga’ de uma bateria é definida pela letra 'C', encontrada nos manuais e *datasheets*. Normalmente fica entre 25C (nominal) e 40C (máxima). Esta taxa ‘serve como base de cálculo da capacidade de carga e descarga, 1C é a capacidade nominal da bateria. É importante manter-se no nível de descarga nominal ou abaixo dele para preservar a vida futura da bateria. Nem todas as marcas dizem algo sobre a taxa de descarga máxima na própria bateria. A capacidade da bateria é definida em miliampere-hora (mA/hr). A capacidade da bateria, juntamente com a taxa de descarga da bateria LiPo, definirá sua saída de corrente máxima. A combinação de capacidade e taxa de descarga é o fator que interessa na especificação, pois isso indicará quanto de descarga será possível no conjunto.

Como exemplo, no RPA que se está projetando, se for escolhido um conjunto LiPo 6S – 10000 mAh 25C tem-se, para um consumo de corrente constante de 76 A (estimado) o cálculo do tempo de autonomia mínimo indicado na equação (10):

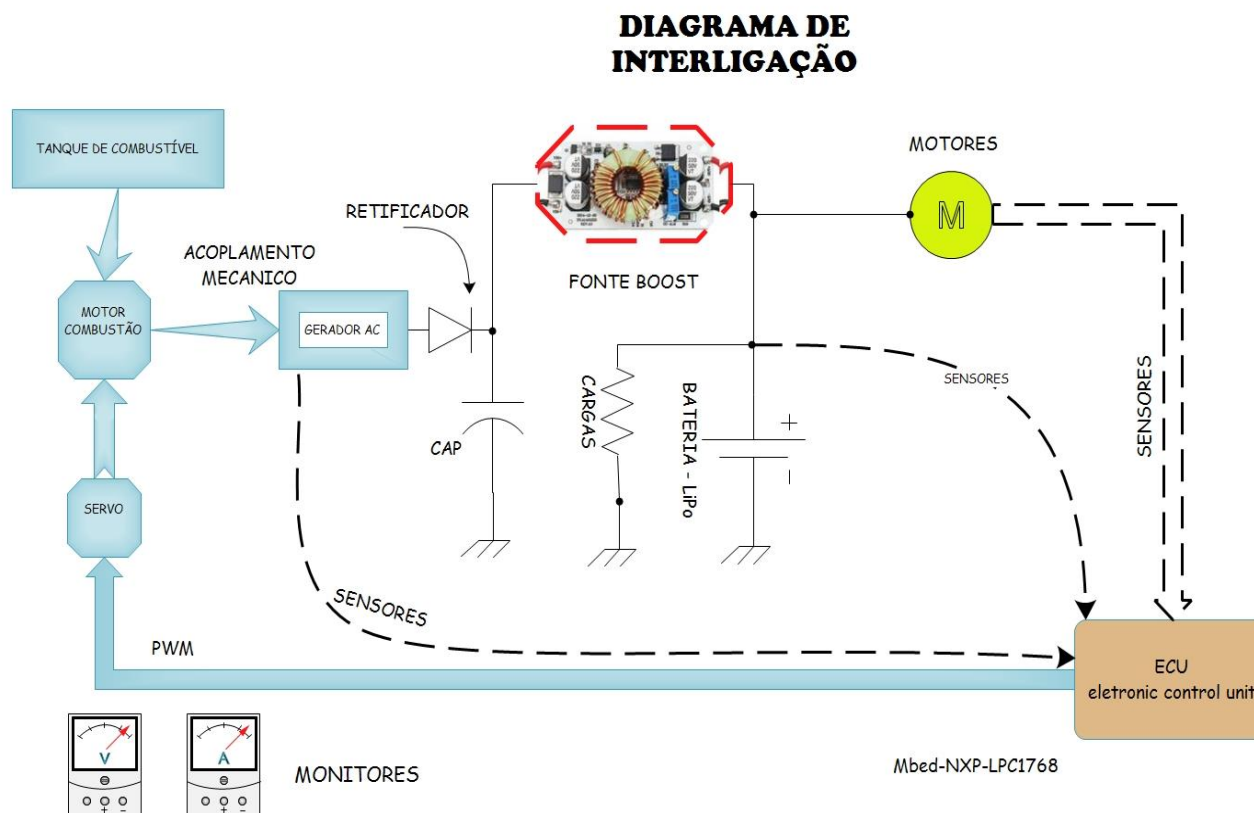
$$(10000 \text{ mAh} / 1000) / 76 \text{ A} = 0,1315 \times 60 = 7,8 \text{ minutos de tempo de voo.} \quad (10)$$

Lembrando-se que o consumo real depende da maneira de voar e de fatores externos, como resistência, peso e vento como visto no voo gravado. Na prática pode-se considerar outros fatores também, como equipamentos eletrônicos obrigatórios e os opcionais incluídos no projeto.

3.3 ARQUITETURA DO SISTEMA

O diagrama com os componentes do sistema está mostrado na Figura 19. Neste diagrama pode-se ver as peças principais do SHPE, e o controle eletrônico representado pelo bloco “ECU”.

Figura 19 – Diagrama Simplificado do sistema desenvolvido.



Fonte: Produção do próprio autor.

A massa de cada componente está no Quadro 4, a menos do banco de baterias.

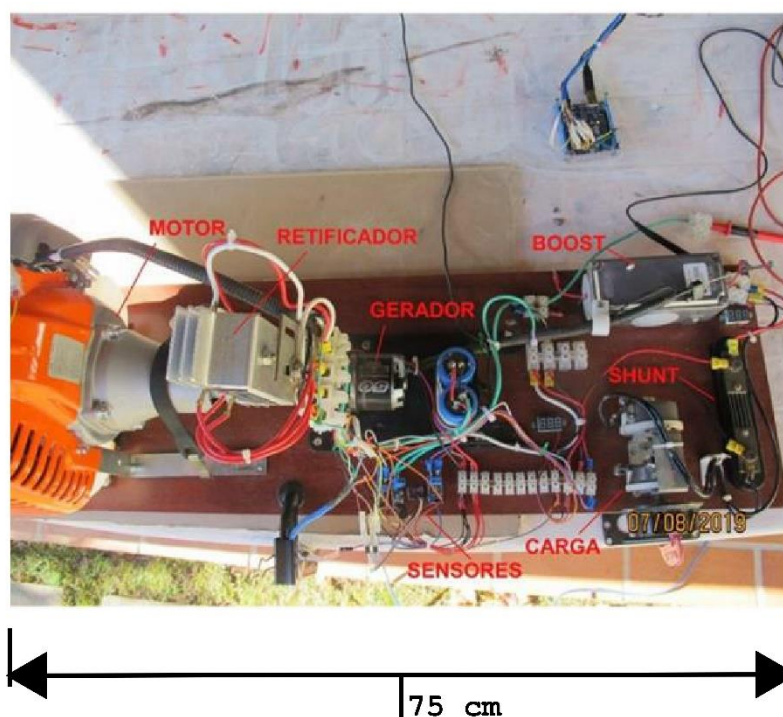
Quadro 4 – Massa dos Componentes do Sistema Testado.

	Massa dos Componentes	
TÓPICO	DESCRIÇÃO	MASSA (kg)
1	GERADOR 5050	0,540
2	Motor Combustão + combustível - VULCAN VR430P 43cc 1,7 HP 1,25 kW	8,000
3	retificador trifásico + capacitores	0,100
4	Fonte <i>boost</i>	0,100
5	Shunt (opcional)	0,090
6	Placa de Controle	0,120
7	Outros (placas, mecânica etc.)	0,300
TOTAL		9,250

Fonte: Produção do próprio Autor.

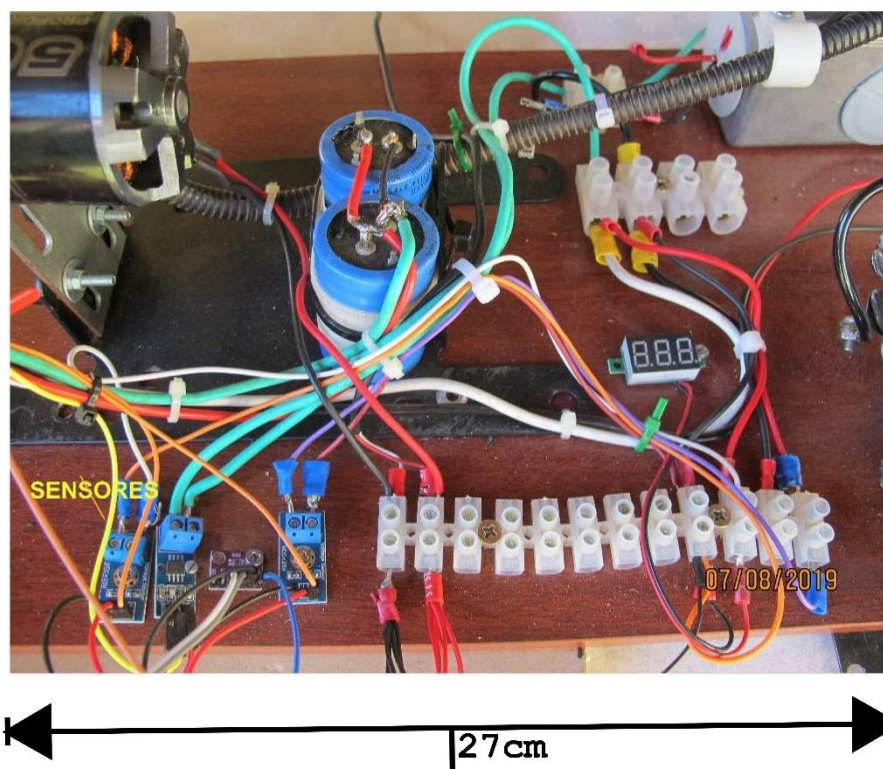
A bancada montada inclui todos os componentes principais e os sensores envolvidos nos testes, e cujas fotografias podem ser vistas nas Figuras 20 e 21.

Figura 20 – Fotografia da bancada de testes com os componentes principais instalados.



Fonte: Produção do próprio autor.

Figura 21 – Fotografia dos sensores instalados.



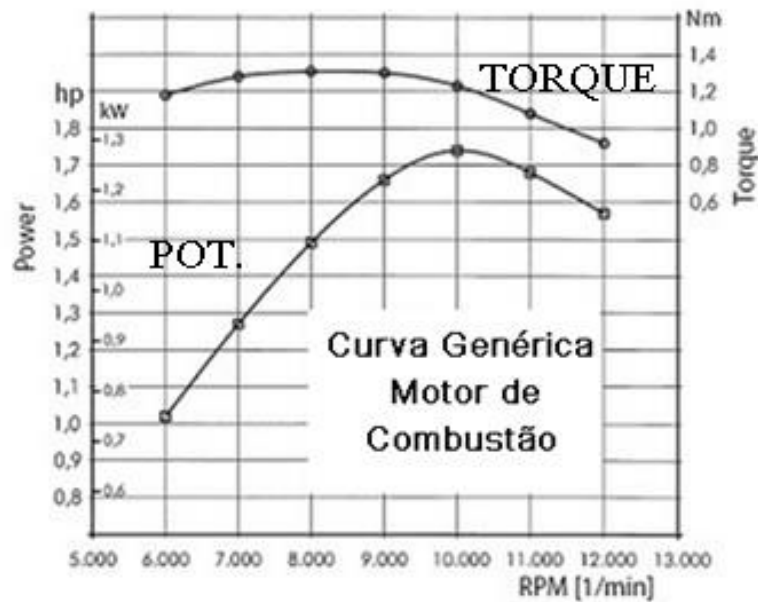
Fonte: Produção do próprio autor.

3.3.1 Motor de Combustão

Na escolha do motor para uso no protótipo de bancada foram levados em conta a potência necessária para acionar o gerador escolhido, o baixo consumo de combustível e a facilidade de manuseio.

Estes critérios o equipamento escolhido e representa uma solução de compromisso entre eles. Outro fator importante é o ciclo de funcionamento, que neste caso é do tipo dois tempos, que geralmente representa motores mais simples e com menos peso. Na Figura 22 estão representadas as curvas características de um motor de combustão genérico.

Figura 22– Curvas características de um motor a combustão equivalente, com ignição.



Fonte: Produção do próprio autor.

O torque é a grandeza física que associa a rotação de um corpo à ação de uma força externa. Ele corresponde ao produto da força externa aplicada no corpo com a distância da mesma em relação a um polo (ponto escolhido para avaliar o torque). Sendo que o torque máximo é atingido antes do pico de potência. A equação (11) relaciona o torque com a Força e com a distância do polo.

$$T = F \times d \quad (11)$$

Onde: T – Torque [Nm]
 F – Força externa [N]
 d – Distância entre a força aplicada e o polo [m]

Potência é a grandeza física que quantifica a energia cedida por uma fonte a cada unidade de tempo. No caso, é calculada a partir do produto entre torque e rotação do motor no instante a ser estudado. A equação (12) relaciona a potência com o torque e a rotação.

$$P = T \times N \times 2 \times \pi / 60 \quad (12)$$

Onde: P – Potência [W]

T – Torque [Nm]

N – Rotação do motor [rpm]

π – 3,14159

A potência depende diretamente do torque, e como este não é linear ao longo das rotações do motor a potência também não é linear. Outro ponto importante é o consumo específico em função da rotação de operação. Estas informações são comuns em motores de maior porte, mas na faixa que se deseja para este trabalho não são fornecidos.

Dentro dos modelos de motores que atendem pré-requisitos, a escolha convergiu para o motor da Zenoah mod. Titan ZG 62SL.

O Titan ZG 62SL é um motor para uso em RPA, seguro, e com ótimas características técnicas. No Quadro 5 são listadas as características técnicas deste motor.

Quadro 5 – Características Técnicas do Motor Titan ZG 62SL.

CARACTERÍSTICAS DO MOTOR	
Aplicação	aeronáutica
Tipo de Cilindro	Anelado
Número de cilindros	único
Massa	2040 g
Tipo de Combustível	gasolina
Dimensões de Montagem	163 x 222 x 197 (mm)
Tipo de operação	2 tempos
RPM	2000–8500
Potência	5 kW
Capacidade	62 cc

Fonte: Svmodelismo (2018).

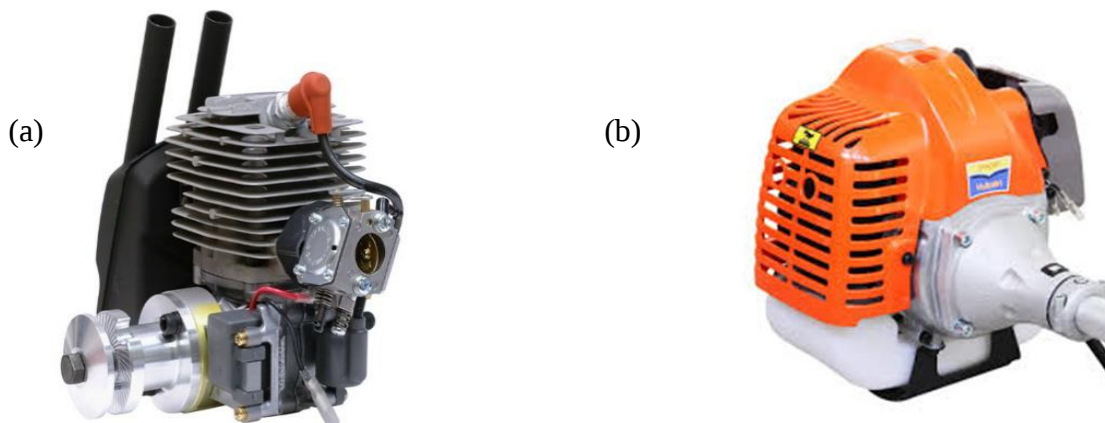
Sendo este motor o mais indicado para esta aplicação, por outro lado não tem o seu acionamento feito de forma trivial, necessitando de técnicas adicionais como o uso de um acionador especial ou pelo giro manual da hélice. No APÊNDICE I foi inserido o esquema a ser usado na partida do Titan ZG 62SL.

No ANEXO A tem-se uma folha com as características do motor Titan ZG 62SL.

Este motor, além de ser mais leve é mais potente que o Vulcan.

O motor selecionado para os testes em bancada foi um Vulcan VR430, de partida mais simples, com acionador por mola, e que possui potência nominal de 1,25 kW. Na Figura 23 tem-se o Titan ZG 62SL (a) e o Vulcan VR430 (b). No ANEXO B, tem-se um Quadro com as características da família deste motor.

Figura 23 – Fotografias dos motores Titan ZG 62SL (a) e Vulcan VR430 (b).



Fonte: Produção do próprio autor.

O motor Vulcan VR430 tem o arranque através de mola retrátil, e tem a rotação inicial do eixo acionada por embreagem, o que permite sua partida sem riscos de giro, que é uma vantagem. Após a partida, o motor tem que ser acelerado até uma rotação mínima, em torno de 5000 rpm, quando começa a girar o eixo. As características do motor Vulcan VR430 encontram-se listadas no Quadro 6.

Quadro 6 – Características do motor Vulcan VR-430

CARACTERÍSTICAS - MOTOR VULCAN VR-430	
tipo de operação	2 tempos
Potência	1,7 HP 1,25 kW - 1,72 CV
Rotação máxima	10.000 rpm
Rotação na lenta	3.000 rpm
Massa	6,9 kg
Tanque do comb.	1,1 Litros
Mistura Combustível	proporção 25:1
Ignição Eletrônica - TCI	
Arranque Manual com mola retrátil	
Carburador Membranas	
Combustível Gasolina com óleo 2 tempos	

Fonte: Vulcan (2019).

O conjunto motor/gerador protótipo tem massa em torno de 10 kgf, como indicado no Quadro 3. A rotação deste motor, entre 3000 rpm e 10000 rpm, está adequada para o acionamento do gerador BLDC. Nas medições realizadas a rotação não ultrapassou 8500 rpm, que é suficiente para a geração de energia solicitada. Com este motor pode-se extrair uma corrente máxima de 52 A em 24 Vcc, mas foi limitada em 30 A.

3.3.2 Gerador

Para a escolha do gerador elétrico foi levado em consideração as características como tipo de tensão entregue (AC ou DC), se monofásico ou trifásico, peso, rotação máxima, mínima e nominal, e potência nominal. Foram testados geradores de 1 a 2 kW de potência nominal, sem escovas (BLDC) com ponte retificadora externa. Este tipo de motor, atuando como gerador se mostrou interessante devido as suas características.

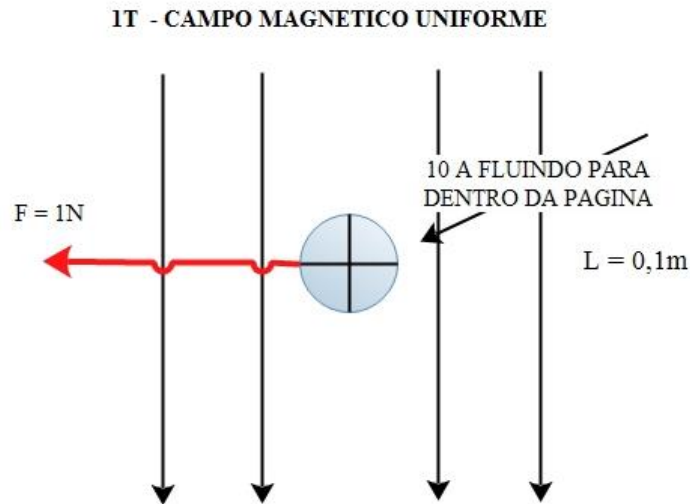
Motores elétricos transformam energia do domínio elétrico para o domínio mecânico usando interação magnética. Em um motor BLDC, essa interação magnética ocorre entre bobinas de fio no estator (parte estacionária), ímãs permanentes no rotor (parte rotativa) e aço estrutura de ambos. Um único fio transportando corrente em um campo magnético uniforme como pode ser visto na Figura 20. Destaca-se a força exercida sobre esse fio chamada ‘Força de Lorentz’.

Na equação (13) está expressada a Força de Lorentz.

$$F = IL . B \quad (13)$$

Onde I é a corrente vetorial que flui através do fio (a direção é importante), L é o comprimento do fio e B é o vetor do campo magnético. A regra do lado direito define a direção da Força de Lorentz, F. Na Figura 24, a força F seria, portanto, 1 N, à esquerda.

Figura 24 – Força de Lorentz aplicada a um único fio num campo magnético uniforme.



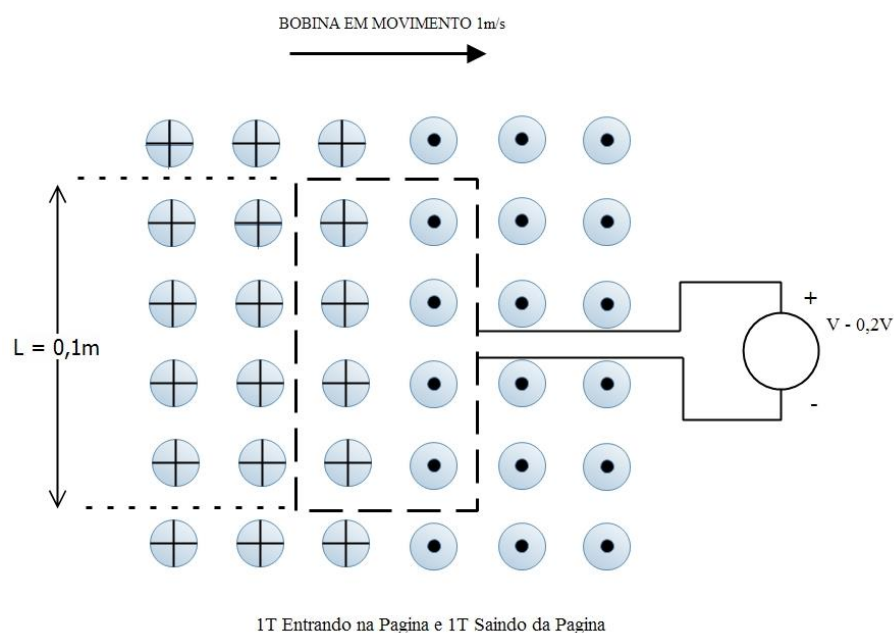
Fonte: Produção do próprio autor.

Outra coisa interessante acontece quando uma bobina de fio se move através de um campo magnético que se altera, mostrado na Figura 25. Nesse caso, uma tensão é induzida com base na taxa de mudança no fluxo magnético que atravessa a bobina. O fluxo é parte integrante do campo magnético sobre área, portanto, devido à geometria, a taxa de mudança de fluxo no cenário representado na Figura 25 é expressa na equação (17):

$$V = \frac{d\phi}{dt} = \Delta B \cdot L \cdot v \quad (17)$$

Onde ‘V’ é a tensão induzida, ‘ ΔB ’ é a mudança no campo magnético, ‘L’ é o comprimento da bobina perpendicular ao movimento e ‘v’ é a velocidade da bobina. O ‘ ΔB ’ é 2‘T’, uma vez que o magnético passa de 1T para a página e 1T para fora da página. Portanto, para a bobina única na Figura 21, a tensão induzida é 0.2 V. Em um motor, isso é chamado de força eletromotriz de retorno (*back EMF*).

Figura 25 – Uma bobina de fio em movimento através de um campo magnético em transição.



Fonte: Produção do próprio autor.

Observa-se que a lei de Lorentz mostra que um motor pode produzir força a partir da corrente.

O relacionamento entre fluxo/tensão mostra que ele pode produzir tensão a partir da velocidade. O detalhe, porém, é que o motor está sempre fazendo os dois no mesmo tempo.

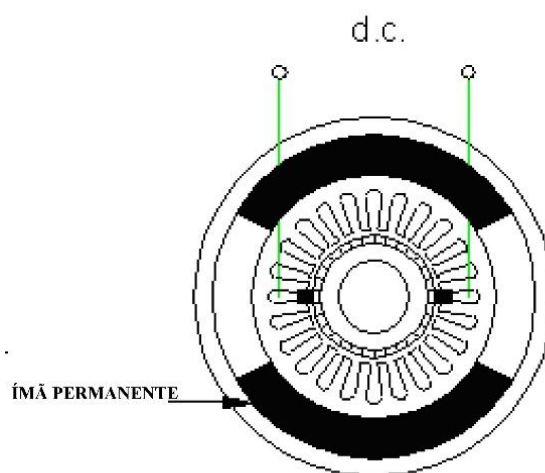
Apenas existe transferência de energia quando todos os componentes - tensão, corrente, força e velocidade - estão presentes ao mesmo tempo. Portanto se um motor for rotacionado internamente ou externamente (*inrunner* ou *outrunner*), tem-se um gerador, como utilizado neste trabalho.

Máquinas DC sem escova utilizando ímãs de neodímio são comumente usadas como geradores para produzir eletricidade e também na função de fornecer torques de frenagem a um eixo (HARTMAN et al.). Os geradores que utilizam ímãs de terras raras de alta energia podem ser muito compactos e produzem bons níveis de energia a partir de rotações relativamente baixas.

O motor de ímã permanente sem escovas se divide em duas classes principais: motor CC sem escova (BLDC) e motor síncrono de ímã permanente (MSIP). O MSIP teve seu desenvolvimento com a substituição do enrolamento do campo da máquina síncrona por um ímã permanente.

Os MSIPs são alimentados com correntes de fase senoidais e operam com base no princípio de campo magnético rotativo. Os enrolamentos, cada um no seu momento, conduzem corrente defasadas. As estruturas de um MSIP são mostradas na Figura 26.

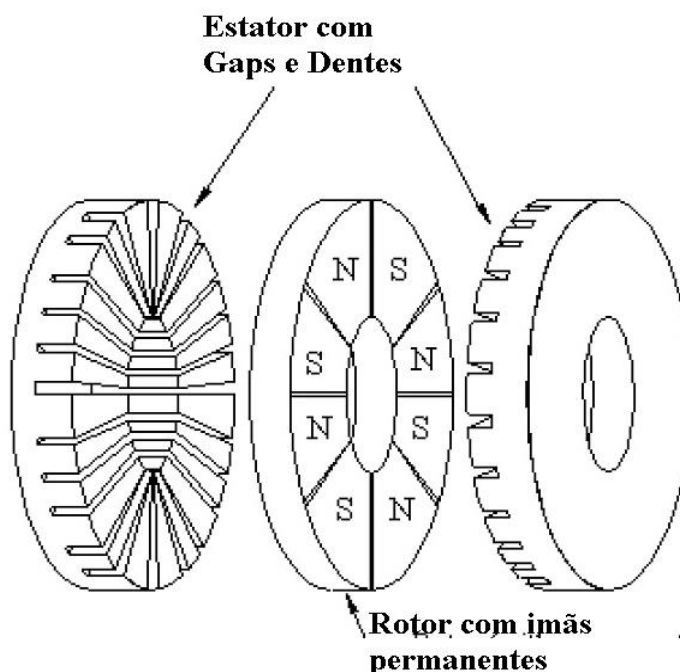
Figura 26 – Esquema de um MSIP.



Fonte: Produção do próprio autor.

O BLDC deve sua origem a uma tentativa de inverter a máquina com escovas DC para remover a necessidade do comutador e da engrenagem da escova. O BLDC é alimentado com formas de onda de corrente de formato retangular ou trapezoidal deslocadas eletricamente por 120° entre elas. A comutação eletrônica é feita pelo sensor de posição do rotor e o controlador eletrônico onde a corrente de armadura é precisamente sincronizada com a frequência do rotor e a posição instantânea. A cada instante apenas duas fases estão conduzindo. Na Figura 27 está esquematizada a vista explodida de uma máquina BLDC.

Figura 27 – Esquema de uma vista explodida do BLDC.



Fonte: Produção do próprio autor.

Até recentemente, os motores convencionais DC eram o sistema de acionamento dominante porque eles forneciam condições mais fáceis de controlar a velocidade, com aceleração rápida, fácil controle de posição e menor custo. Entretanto, vários avanços técnicos nos materiais de ímãs permanentes, em tecnologia de transistores e semicondutores, e em sensores de posição, fizeram do BLDC uma alternativa viável. As aplicações também tiveram grande impulso devido as melhorias em materiais magnéticos, especialmente as terras raras. Estes motores são menores e mais leves pois não tem enrolamentos de campo nem comutadores e escovas, e tem maior eficiência e maior densidade de potência se comparados com os tradicionais motores DC. Além disso, o seu design oferece uma faixa de velocidade maior, pois não é limitada pelo comutador que existe nos motores DC com escovas. Sua aplicação é portanto, importante em ambientes limpos e propensos a explosão, como náutica, robótica, veículos elétricos, e indústrias químicas. Devido ao advento de ímãs permanentes de alta energia e circuitos eletrônicos de controle sofisticados, o motor BLDC está se tornando cada vez mais popular em uma ampla gama de aplicações, que incluem máquinas-ferramentas, robótica, aeroespacial, geradores, atuadores, acionamentos industriais e veículos elétricos. Esse tipo de motor tem aplicação também em freio magnético, que é um freio

auxiliar em veículos elétricos, uma vez que a relutância magnética aumenta na proporção do fornecimento de corrente de retorno para a geração.

Os geradores BLDC são comumente caracterizados por sua constante da FEM (força eletromotriz), K_e e sua resistência de enrolamento, R_b . Esses parâmetros determinam a constante do motor, K_m , uma figura de mérito independente do enrolamento (HARTMAN et al.). Nas equações 19 e 20 tem-se expressa estas grandezas.

$$V = K_e \omega_m = K_e \frac{2}{p} \omega_e \quad (18)$$

$$K_m = K_e / \sqrt{R_b} \quad (19)$$

A simulação de um gerador apenas por esses parâmetros é insuficiente para explicar o comportamento observado do torque. A indutância L deve ser considerada.

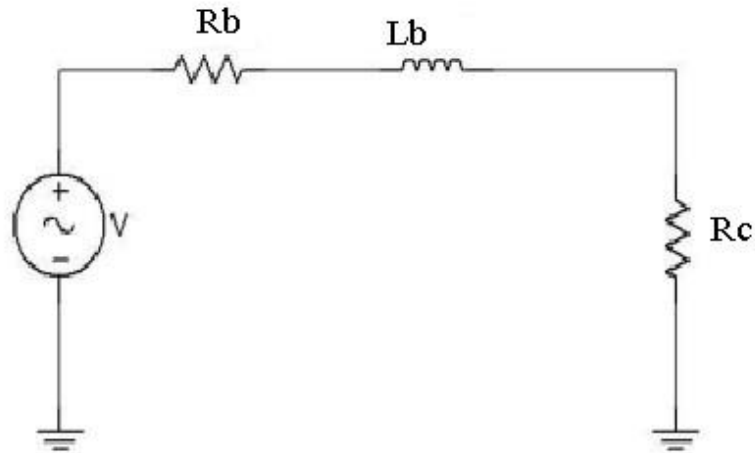
O Quadro 7 contém um resumo da nomenclatura usada. Um gerador monofásico pode ser representado pelo circuito mostrado na Figura 28.

Quadro 7 – Nomenclatura Adotada.

SÍMBOLO	GRANDEZA	UNIDADE
k_e	cte fem	v/rad/s
K_m	cte motor	Nm/watt
ω_m	veloc. rotação	rad/s
ω_e	veloc. elétrica	rad/s
R	resistência	ohm
L	indutância	henry
p	#	polos
I	corrente	amp
P	potência	watts
T	torque	Nm
η	rendimento	#

Fonte: Produção do próprio autor.

Figura 28 – Circuito que simboliza de um gerador monofásico.



Fonte: Produção do próprio autor.

A equação (20) descreve o comportamento do circuito.

$$V(t) = Ke\omega_m \sin(\omega_e t) = IR + L \frac{dI}{dt} \quad (20)$$

Onde:

$R = R_b + R_c$ (resistência da bobina + resistência da carga).

Com o aumento da velocidade, as amplitudes da corrente e as formas de onda de potência aumentam e se aproximam de um limite superior. O torque, no entanto, aumenta até o máximo, depois começa a cair. O torque máximo ocorre na velocidade crítica. A Potência na velocidade crítica é exatamente metade do seu valor no limite de alta velocidade; a corrente é 0,5 vezes do seu limite de alta velocidade.

No Quadro 8, estão detalhadas as características técnicas do modelo escolhido, que foi projetado para ser usado como motor propulsor de RPA.

Quadro 8 – Características do Motor/Gerador Mod. PROPDRIVE v2 5050 580KV

MOTOR/GERADOR PROPDRIVE v2 5050 580KV	
massa	0,400 kg
Kv(rpm/v)	580
Max Corrente (A)	90
Max. Rotação	12876 rpm
Max Tensão (V)	22.2
Pot. (W)	2000

Fonte: Spinner (2019).

O acoplamento entre eixos foi feito de forma direta, sem redução, através de um conector de medição, próprio para haste de aterramento.

3.3.3 Sensores e Servo

Em geral, os sensores podem ser usados em sistema de controle de várias maneiras:

1. Medir os resultados da planta para controle de *feedback*;
2. Medir os sinais de entrada da planta (desejáveis e de perturbação) para controle tipo *feedforward*;
3. Para medir sinais de saída para monitoramento do sistema, diagnóstico, avaliação, ajuste de parâmetros e controle de supervisão;
4. Para medir sinais de entrada e saída para teste do sistema e modelagem experimental (GALIP ULSOY et al., 2012).

Os sensores foram selecionados de acordo com os níveis dos sinais a serem monitorados.

O Quadro 9 lista os limites das grandezas físicas a serem monitoradas e o tipos de sensores instalados.

Quadro 9 – Tipos de sensores e limites operacionais.

Tópico	Sensor	Grandeza	Limites
1	Corrente na Carga (<i>Hall Sensor</i>)	Amperes (A)	0 – 30 A
2	Rotação (<i>Hall Sensor</i>)	rpm	0-10000 rpm
3	Tensão gerador	Vcc (volts)	0-25 Vcc
4	Tensão <i>boost</i>	Vcc (volts)	0-25 Vcc
5	Temperatura	°C	-20 – 80°C
6	Altura do solo	Metros (m)	0- 3000 m

Fonte: Produção do próprio autor.

São os seguintes os modelos usados:

- Sensor de Corrente mod. ACS712 – 30 A
- Sensor de RPM (tipo Hall) mod. KY-003
- Sensor de temperatura, de umidade, e de pressão, mod. BME280
- Sensor de tensão mod. 0-25 Vcc (2 Un.)

Todos estes sensores estão aptos a serem conectados a placa de controle mediante o uso de divisores resistivos, quando necessário, para limitar as tensões nas entradas analógicas da placa Mbed. A tensão de operação nas portas da placa é de 3,3 Vcc.

O Sensor BME280 é um componente de precisão, fabricado pela *Bosh*, que pode medir a umidade relativa do ar de 0 a 100% com precisão de $\pm 3\%$. A pressão barométrica tem um alcance de 300 Pa a 1100 hPa com precisão absoluta de ± 1 hPa e mede temperatura numa faixa de -40°C a 85°C com precisão de $\pm 1,0^{\circ}\text{C}$. Seu funcionamento também é com alimentação de 3,3 Vcc, utilizado na placa de controle.

O sensor de tensão utilizado foi desenvolvido de forma simplificada e em pequenas dimensões, e praticamente faz uma redução por divisor resistivo do valor da tensão. Tem uma relação de 5:1, ou seja, transforma qualquer valor na entrada até o limite de 25 Vcc num valor proporcional na saída.

O servo motor adotado é o modelo TowerPro MG995, com dados técnicos listados no Quadro 10, No ANEXO C tem-se o *Datasheet* deste componente.

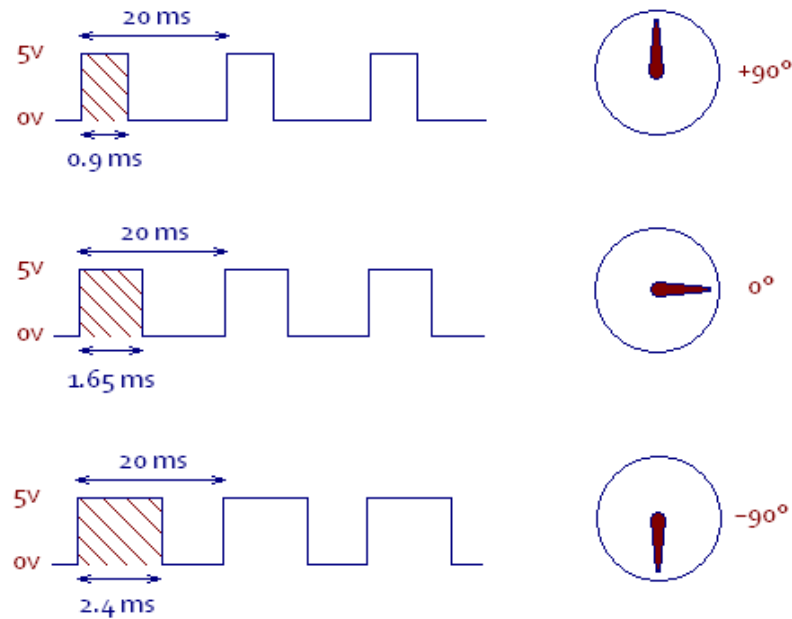
Quadro 10 – Características do Servo MG 995R.

Dados técnicos - MG 995R:	
Descrição	MG 995R TowerPro
Tensão de Alimentação	4,8 - 7,2 Vcc
Corrente de Operação	500 mA – 900 mA
Corrente ‘Stall’	2,5 A
Temperatura de Operação	0 ~ 55°C
Conector	JR (Universal)
Comprimento do cabo	24,5 cm
Velocidade	0,16 s/60° (6 Vcc)
Torque a 4.8 Vcc	8,5 kg-cm
Torque a 6 Vcc	10 kg-cm
Dimensões	40.7x19.7x42.9 mm
Massa	0,055 kg

Fonte: Futabausa (2019).

A forma de onda PWM utilizada no controle está na Figura 29. No controle do motor VR430 tem-se um ponto morto no início do giro do servo, devido a necessidade de se atingir uma certa rotação para a embreagem atuar. Essa diferença é retirada via software.

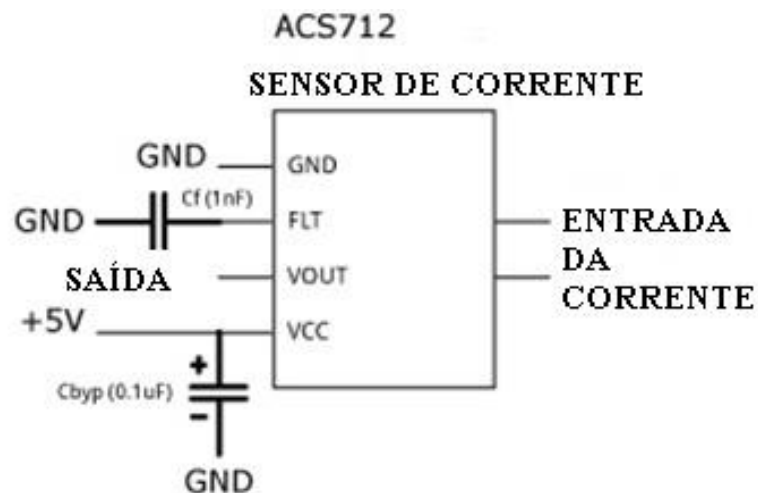
Figura 29 – Forma de onda utilizada no comando do Servo.



Fonte: Produção do próprio autor.

O ACS712 é um dispositivo de medição de corrente com efeito *Hall*, que produz uma tensão analógica em resposta ao fluxo de corrente através de um condutor conectado a ele. Este sensor está disponível nas versões de 5, 20 e 30 Amperes e detecta as correntes CA e CC (positiva ou negativa). O conceito do sensor *Hall* permite que fique o sensor isolado eletricamente da corrente medida. O ACS712 tem uma largura de banda máxima de 80 kHz. Na Figura 30, tem-se o esquema elétrico deste sensor.

Figura 30 – Esquema elétrico do ACS712.



Fonte: Adaptado de ACS712 datasheet (2018).

O pino rotulado "Filtro" é conectado à saída do primeiro *opamp* do ACS712. Um capacitor de 1 nF conectado a este terminal e o terminal de referência serve como um filtro passa-baixos e desvia o ruído de alta frequência. Diferentes valores de capacitância alteram a frequência de corte deste filtro simples.

O ACS712 é um dispositivo de 5 Vcc e, como a entrada analógicas da placa trabalham na faixa de 0 a 3,3 Vcc, o ACS712 foi usado com um divisor de tensão para integrar-se ao Mbed. A relação do divisor resistivo deve ser escolhida para escalar 5 Vcc a 3,3 Vcc, uma taxa de $3,3 / 5$. Resistores na faixa de 100 k Ω , servem bem nesta aplicação.

3.3.4 Placa Microcontroladora

A ECU (*Electronic Control Unit*) usada é uma placa de desenvolvimento da Mbed modelo NXP LPC1768. Na Figura 31 é apresentada a placa. No Apêndice D, Figura 70 está desenhado o esquema do projeto de controle final.

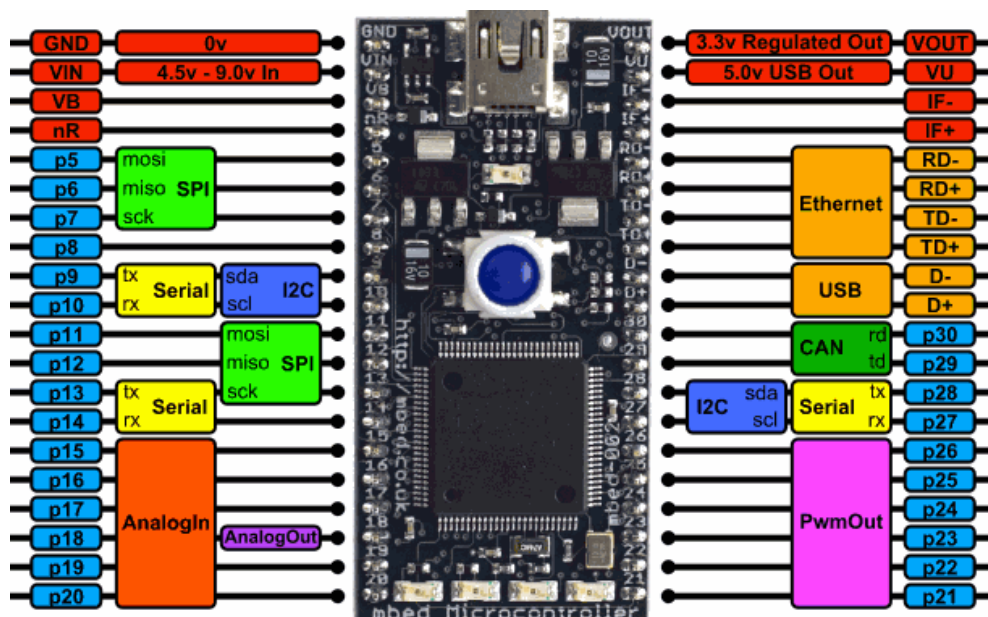
Figura 31– Placa de controle modelo Mbed NXP LPC1768.



Fonte: MBED (2018).

Na Figura 32 tem-se a pinagem e interligações dessa placa de desenvolvimento microprocessada. Observa-se a saída PWM já incorporada no dispositivo (pinos 21-26).

Figura 32– Pinagem da placa de controle.



Fonte: MBED (2019).

A placa Mbed NXP LPC1768 possui várias bibliotecas de controle, o que facilita a criação do *firmware*.

Os microcontroladores Mbed são uma série de módulos de prototipagem baseados no Mbed HDK. O Mbed NXP LPC1768 foi escolhido devido à sua capacidade de trabalhar em tempo real com a adição de sensores no controle de sistema embarcados. Ele é baseado num núcleo ARM Cortex-M3 de 32 bits rodando a 96 MHz incluindo uma memória 512 KB FLASH, e 32 KB de memória RAM. Possui *Ethernet* integrada, *host* e dispositivo USB, CAN, SPI, I2C, ADC, DAC, PWM e outras interfaces de E/S. Todas estas características levaram à escolha deste microcontrolador (MBED, 2019).

Esta placa tem 8 entradas ADC de 12 bits. Esta resolução significa que o conversor consegue ter sensibilidade de 0,006104 Vcc para um sinal de 25 Vcc de entrada. É importante lembrar que a máxima tensão suportada nas entradas analógicas é de 3,3 Vcc.

Calculando o valor da menor resolução de tensão para uma entrada de 25 Vcc (12 bits).

$$S = 25 \times 1/2^{12} (V_{cc})$$

S = 0,006104 Vcc é a menor resolução que pode ser lida pelo sensor.

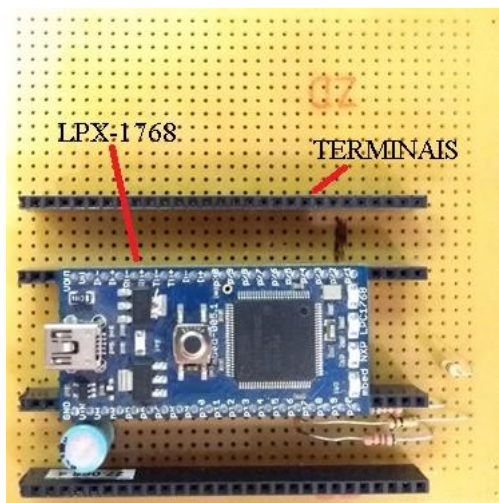
Portanto:

Tensão de entrada do sensor: 0 a 25 Vcc;

Faixa de detecção: 0,006104 até 25 Vcc;

No ANEXO C foi incluído o *Datasheet* deste microcontrolador e no APÊNDICE D está desenhado o esquema elétrico da placa com os sensores. A Figura 33 representa a placa genérica desenvolvida para instalação no protótipo.

Figura 33 - Fotografia da placa utilizada no protótipo.



Fonte: Produção do próprio autor.

3.3.5 Banco de Baterias

Para alguns testes no protótipo foram instaladas duas baterias de 12Vcc/7A cada, em série, perfazendo um total de 24 Vcc na saída. As baterias instaladas são do tipo gel, para uso em *nobreaks*.

Para uso no RPA serão usadas baterias tipo LiPo (Lítio-Polímero), que possuem menor volume e maior capacidade. Na Tabela 1 estão relacionados os níveis de tensão para cada nível de carga nas baterias tipo LiPo.

As baterias de Polímeros de Lítio (LiPo) é um tipo de desenvolvimento mais recente das baterias de Lítio. A diferença está no eletrólito, que neste caso é sólido, ao contrário do tradicional. Isso facilita a fabricação e aumenta a segurança, proporcionado a criação de um formato mais simples. Apesar disso, alguns cuidados são importantes no uso e armazenamento deste tipo de bateria.

Tabela 1 – Classificação das Baterias LiPo em relação a tensão de operação

TENSÃO DAS CÉLULAS	1S	2S	3S	4S	5S	6S	8S	10S	12S
MÍNIMO	3	6	9	12	15	18	24	30	36
DESCARREGADA	3,4	6,8	10,2	13,6	17	20,4	27,2	34	40,8
NOMINAL	3,7	7,4	11,1	14,8	18,5	22,2	29,6	37	44,4
ARMAZENADA	3,8	7,6	11,4	15,2	19	22,8	30,4	38	45,6
CARREGADA	4,2	8,4	12,6	16,8	21	25,2	33,6	42	50,4

MÍNIMO	BAIXA TENSÃO COM GRANDE RISCO DE DANIFICAR PERMANENTEMENTE A BATERIA
DESCARREGADA	BAIXA TENSÃO SEGURA
NOMINAL	TENSÃO NOMINAL DO CONJUNTO
ARMAZENADA	TENSÃO DE ARMAZENAMENTO
CARREGADA	TENSÃO PARA USO

Fonte: Produção do próprio autor.

3.3.6 Fonte Boost

O esquema projetado, é composto por uma fonte tipo *boost step-up*. O BLDC atuando como gerador tem uma tensão de saída variável, de acordo com a rotação aplicada no rotor.

Com o objetivo de se manter a tensão na saída em torno de 24 Vcc foi instalada esta fonte. Composta por duas placas capazes de regular 20 A cada. As características técnicas estão descritas no Quadro 11.

Quadro 11 – Descrição da Fonte tipo *Boost step-up*.

FONTE TIPO BOOST STEP-UP - 600W	
Tensão de entrada	DC 10-60 Vcc
Corrente de entrada	20 A (MAX)
Tensão de Saída	12 a 80 Vcc ajustável
Eficiência	96%
Tamanho do módulo	70 mm x 90 mm

Fonte: Produção do próprio autor.

Foram instaladas duas fontes em paralelo, aumentando para 1200 W a capacidade de regulação total. Nominalmente, qualquer tensão acima de 10 V_{cc} a saída da fonte regula no valor pré-ajustado. Os ajustes de tensão de corrente são feitos por potenciômetros multivoltas.

3.3.7 Retificador Trifásico

Foi utilizado uma ponte retificadora trifásica de 100 A/400 V, na saída do gerador BLDC. Este componente já possui um dissipador apto a proporcionar seu funcionamento a plena capacidade.

3.4 MÉTODOS APLICADOS

Foi feita uma modelagem do sistema em MATLAB®Simulink, com o objetivo de se analisar a melhor arquitetura e o seu comportamento.

Testes também foram feitos, em bancada, com os principais componentes do sistema. As cargas foram substituídas por um reostato de potência, e o banco de baterias instalado foi do tipo Gel. Vários testes com diferentes variações de corrente foram feitos para se avaliar o comportamento dinâmico do sistema e para se definir a função de transferência através do toolbox *System Identification* do MATLAB®Simulink.

Os dados coletados através do sistema de coleta foram gravados e usados para eventuais correções no modelo teórico. O sistema de coleta está detalhado no tópico 3.4.1.

3.4.1 Descrição do funcionamento do SHPE

Após o acionamento do motor de combustão, através da mola de partida, é feito um monitoramento do sistema através das leituras dos sensores. Até o limite de corrente de carga pré-ajustado o gerador alimenta a carga, e a partir daí a alimentação das cargas principais (motores e ESCs) fica por conta das baterias, inclusive nos picos de consumo, que pode atingir até 5 vezes o valor limite.

No sistema embarcado esse valor máximo suportado pelo sistema deverá se limitar a 80 A. A lógica do sistema impõe um monitoramento contínuo dos sensores, sendo que a corrente na carga é o parâmetro principal. Uma vez dentro das condições de operação, o sistema obedece a uma sub-rotina onde os dados previamente treinados são usados para acionamento do servo que comanda o motor. Quando a corrente for maior do que a I_n (corrente nominal

selecionada) ou quando os outros sensores saírem dos limites programados, o servo comanda o motor para a rotação mínima, transferindo a carga para o banco de baterias. Portanto assim que o sistema de monitoramento detectar alguma condição anormal, como por exemplo, altitude ou temperatura fora dos parâmetros normais ou uma corrente superior a aquela que foi ajustada como limite o banco de baterias assume.

Essa condição permanecerá até que o monitoramento considere que o gerador pode assumir a alimentação. Em voo normal estas comutações são rápidas e ocorrem constantemente, já que essas alterações no consumo decorrem dos ajustes de rotação dos motores de propulsão em função principalmente dos variação dos ventos.

3.4.2 Coleta de Dados

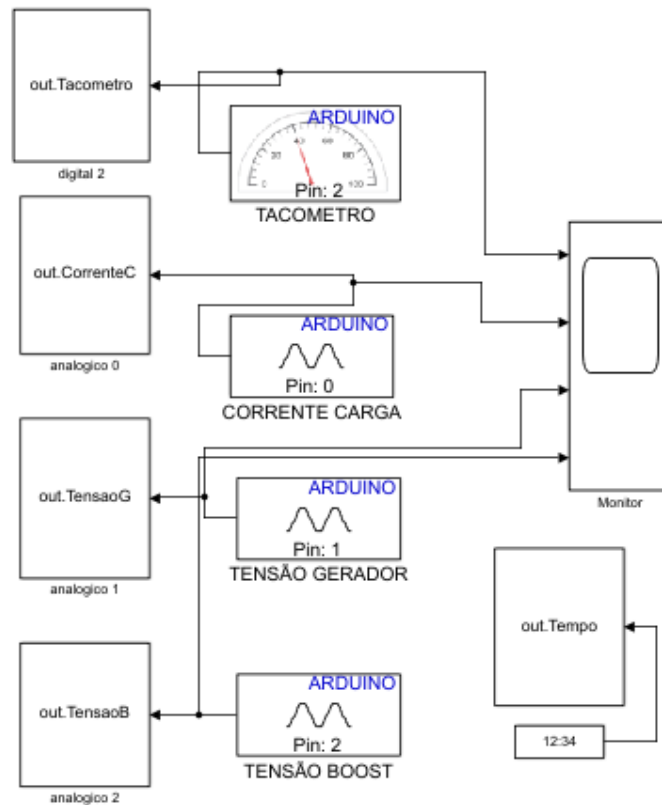
Os sistemas de coleta de dados têm por objetivo coletar dados através de medições de grandezas físicas oriundas de sensores, tais como temperatura, pressão, tensão e corrente, com o objetivo de monitoramento, controle e registro.

As grandezas elétricas e físicas oriundas dos sensores ou transdutores passam por condicionadores de sinais e são disponibilizadas para o sistema de coleta. Este condicionamento tem por objetivo a filtragem, amplificação, isolamento e transformação dos sinais para os níveis e características apropriadas para serem aproveitadas pelo sistema. Neste trabalho utiliza-se a plataforma Arduino® juntamente com o MATLAB®Simulink como módulo de coleta dos dados do protótipo.

O primeiro esquema utilizado nas medições através do MATLAB®Simulink está representado na Figura 34. Foram registrados os sinais de quatro sensores: rotação no eixo do motor, tensão na bateria, tensão no gerador e corrente na carga.

Outros sensores, como a temperatura e altura tiveram seus valores medidos externamente.

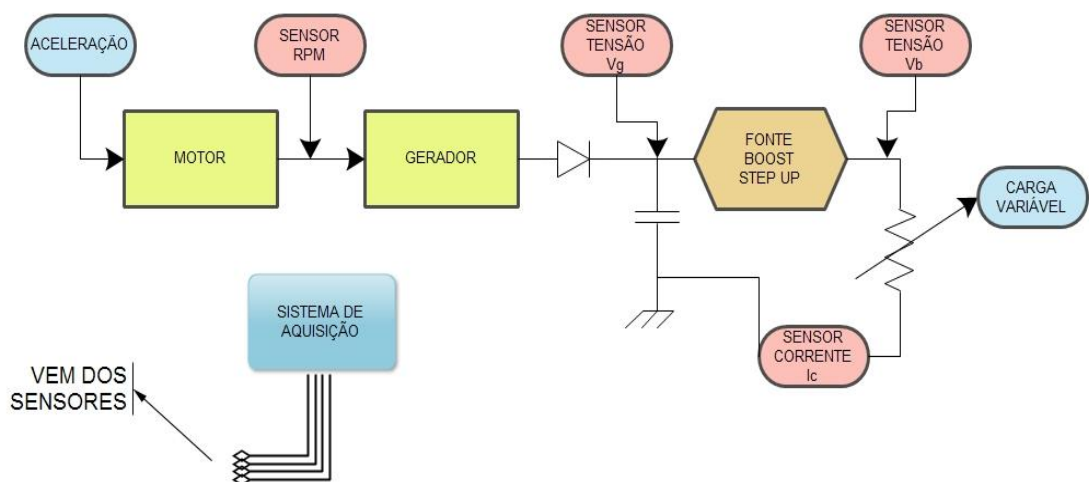
Figura 34 – Esquema n. 1 de Coleta de Dados.



Fonte: Produção do próprio autor.

Foram usados esses quatro sensores para se obter os primeiros gráficos. Na Figura 35 estão indicadas as localizações dos sensores no sistema.

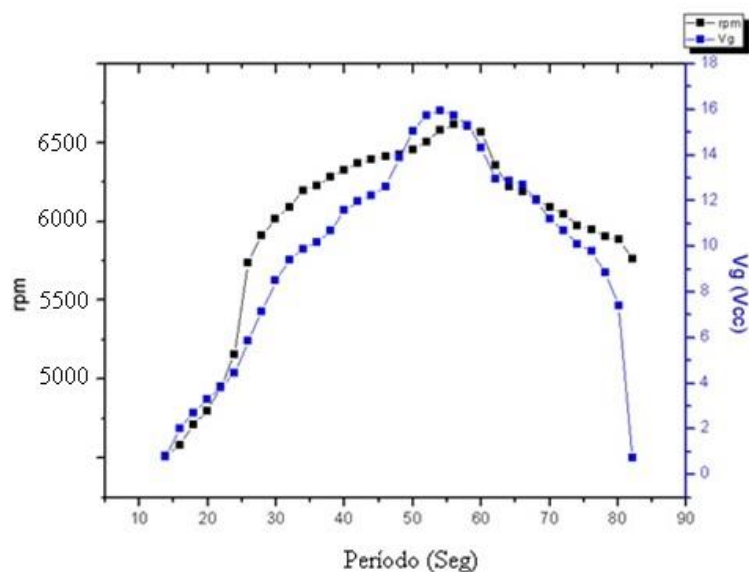
Figura 35 – Localização dos Sensores no Sistema



Fonte: Produção do próprio autor.

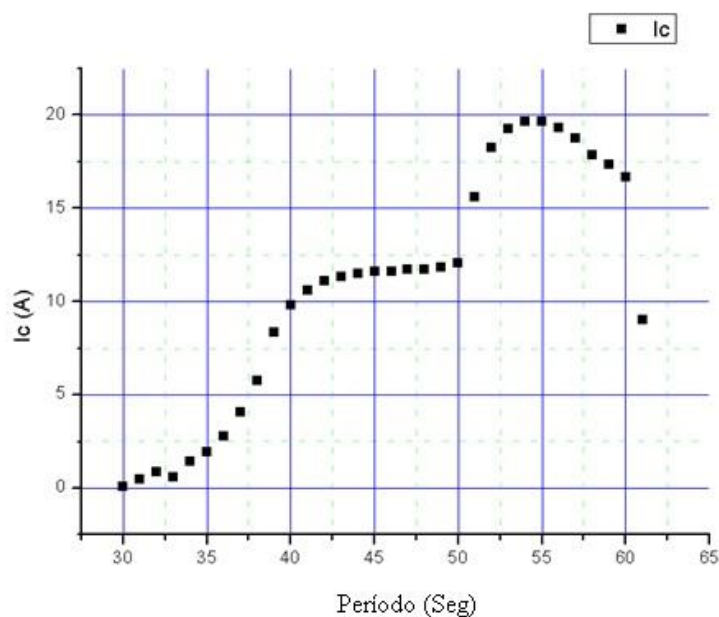
Nas Figuras 36, 37 e 38 estão reproduzidos alguns dos gráficos obtidos a partir deste sistema de coleta. A carga foi alterada de forma aleatória, com o objetivo de se testar o comportamento do conjunto. Foi feita uma atuação manual do aumento da corrente e da aceleração do motor, de forma a se compensar a queda de rotação.

Figura 36 – Gráficos rpm x t e Vg x t.



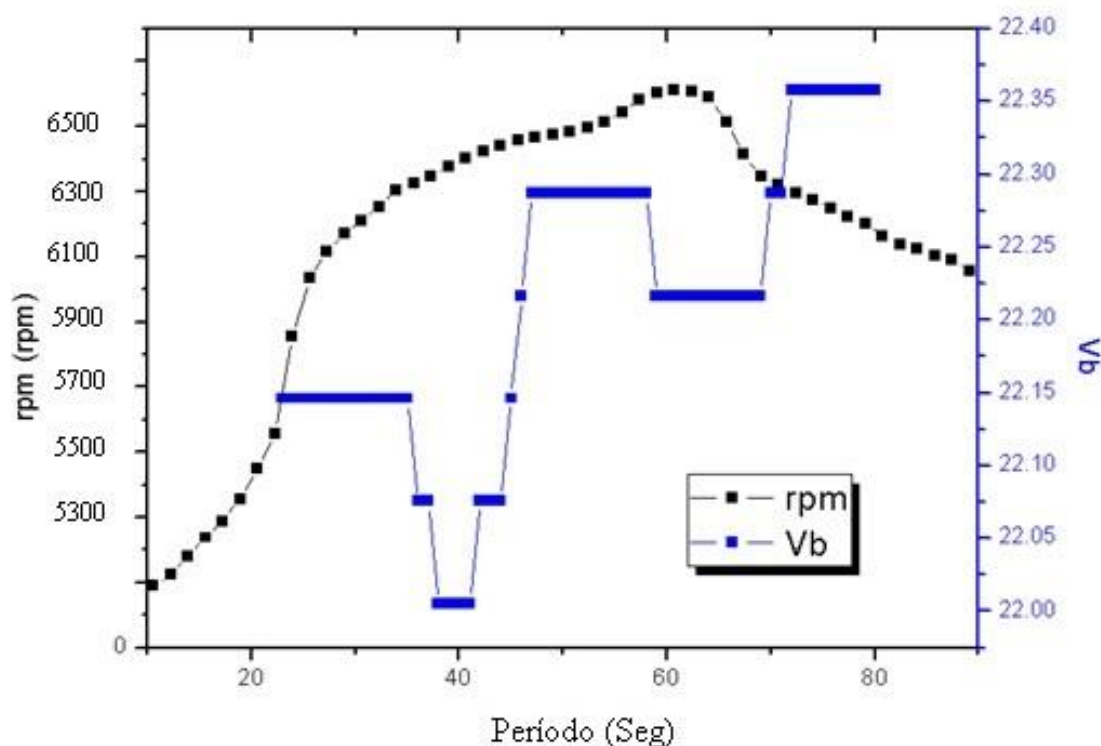
Fonte: Produção do próprio autor.

Figura 37 – Gráfico da Ic x t



Fonte: Produção do próprio autor.

Figura 38 – Gráficos rpm x t e Vb x t



Fonte: Produção do próprio autor.

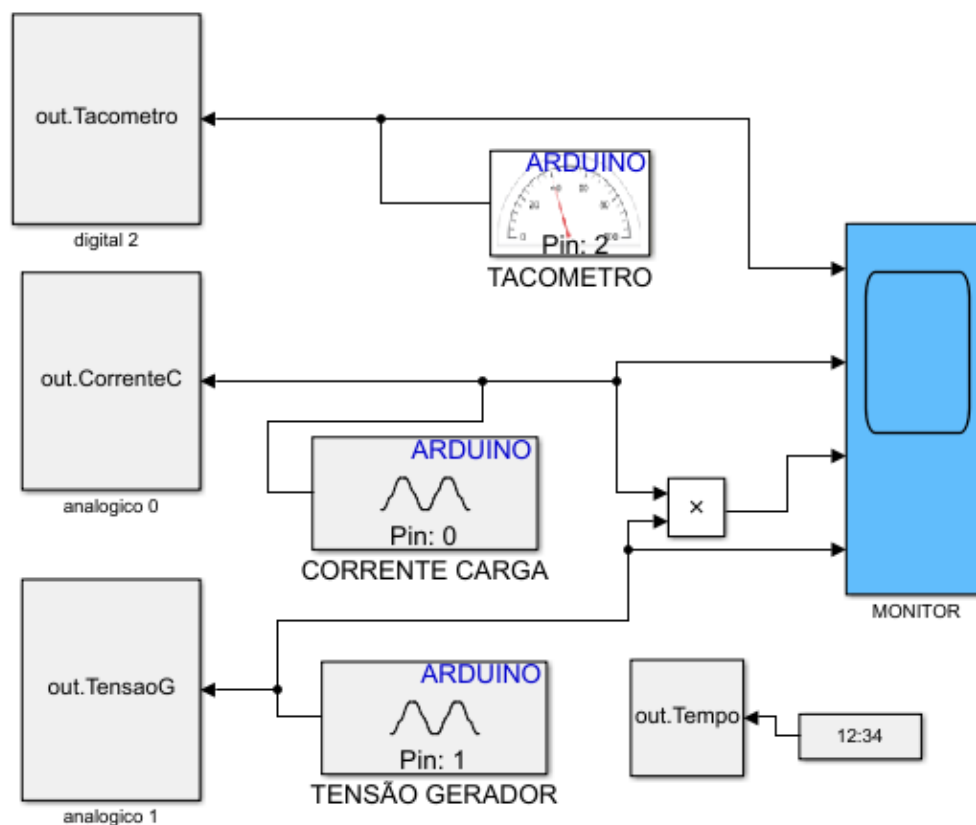
Com base nas informações obtidas dos gráficos, conclui-se que:

- De acordo com os dados medidos, partir de 10 A o motor a combustão tem que trabalhar acima de 6000 rpm;
- A variação na saída do *boost* a partir da rotação mínima não é significativa, para qualquer nível de corrente da carga.
- A margem que se tem que trabalhar no controle da rotação fica entre 6000 rpm e 8500 rpm, com este motor.

Foram obedecidas também as características nominais do motor Vulcan VR430. Para exemplificar, num dos testes executados foi obtido o gráfico plotado na Figura 35, cuja tabela está inserida no Apêndice B. Estes testes se basearam na variação da resistência da carga alimentada pelo gerador para várias rotações, com os sensores indicados na configuração mostrada no tópico 3.4.2. Como padrão do procedimento, foi medida a corrente máxima que se pode extrair de uma determinada rotação. A montagem do sistema apresentou alguns

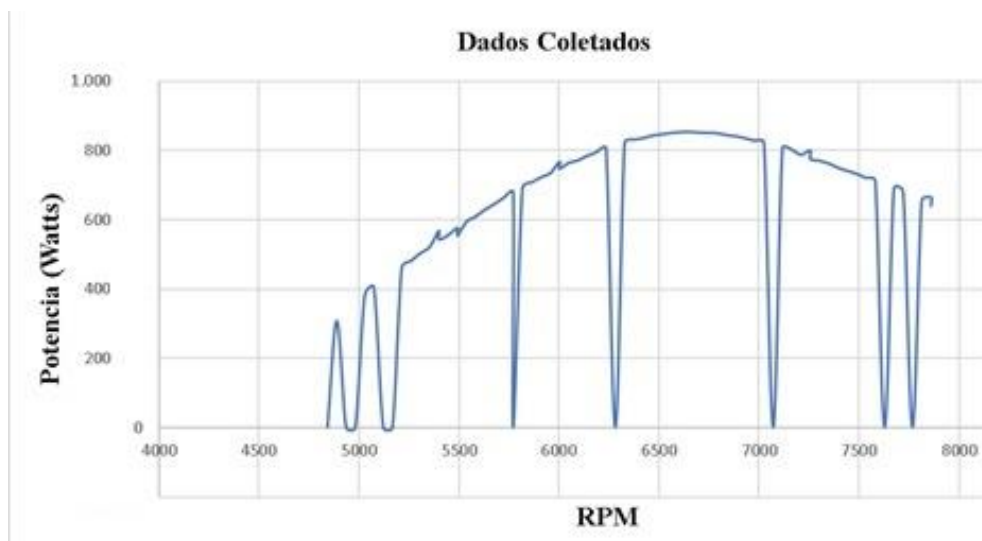
problemas como por exemplo alta vibração na fixação do gerador, o que provocou a soltura dos parafusos depois de alguns minutos. Correções na montagem foram então executadas. Os dados não foram alterados/corrigidos para completar as eventuais falhas observadas, conforme indicado no gráfico. Na Figura 39, está desenhado outro diagrama do Simulink® utilizado nessa medição, tendo como placa coletora o Arduino Uno®. Note que neste caso a potência é resultado direto dos dados da corrente pela tensão, e que o pico ficou perto de 6700 rpm como pode ser visto na Figura 40.

Figura 39 – Esquema n. 2 de Coleta de dados.



Fonte: Produção do próprio autor.

Figura 40 – Curva da Potência do Motor x Rotação.



Fonte: Produção do próprio autor.

Outras medições feitas em bancada, estão apresentadas no Apêndice F, onde a Figura 72 tem plotada a rotação do motor e na Figura 73 foi inserido o gráfico da Corrente e da Tensão no Gerador. Estes dois gráficos não foram incluídos na mesma figura para não prejudicar a visualização. Devido à grande quantidade de dados (alta taxa de amostragem), a tabela com estes valores não foi incluída.

Devido à alta relutância do gerador, pequenas alterações na corrente provocam modificações instantâneas na rotação do conjunto.

3.4.3 Modelagem e Simulação

A modelagem de sistemas e componentes e a consequente observação do funcionamento tem evoluído juntamente com a capacidade computacional. Existem muitas vantagens inerentes ao processo, destacando-se o estudo do comportamento de um sistema sem construí-lo, a grande precisão em comparação com o modelo analítico, auxílio em se prever algum fenômeno inesperado, e ser capaz de uma análise "*What-If*", para se verificar o comportamento geral do sistema sob certas condições.

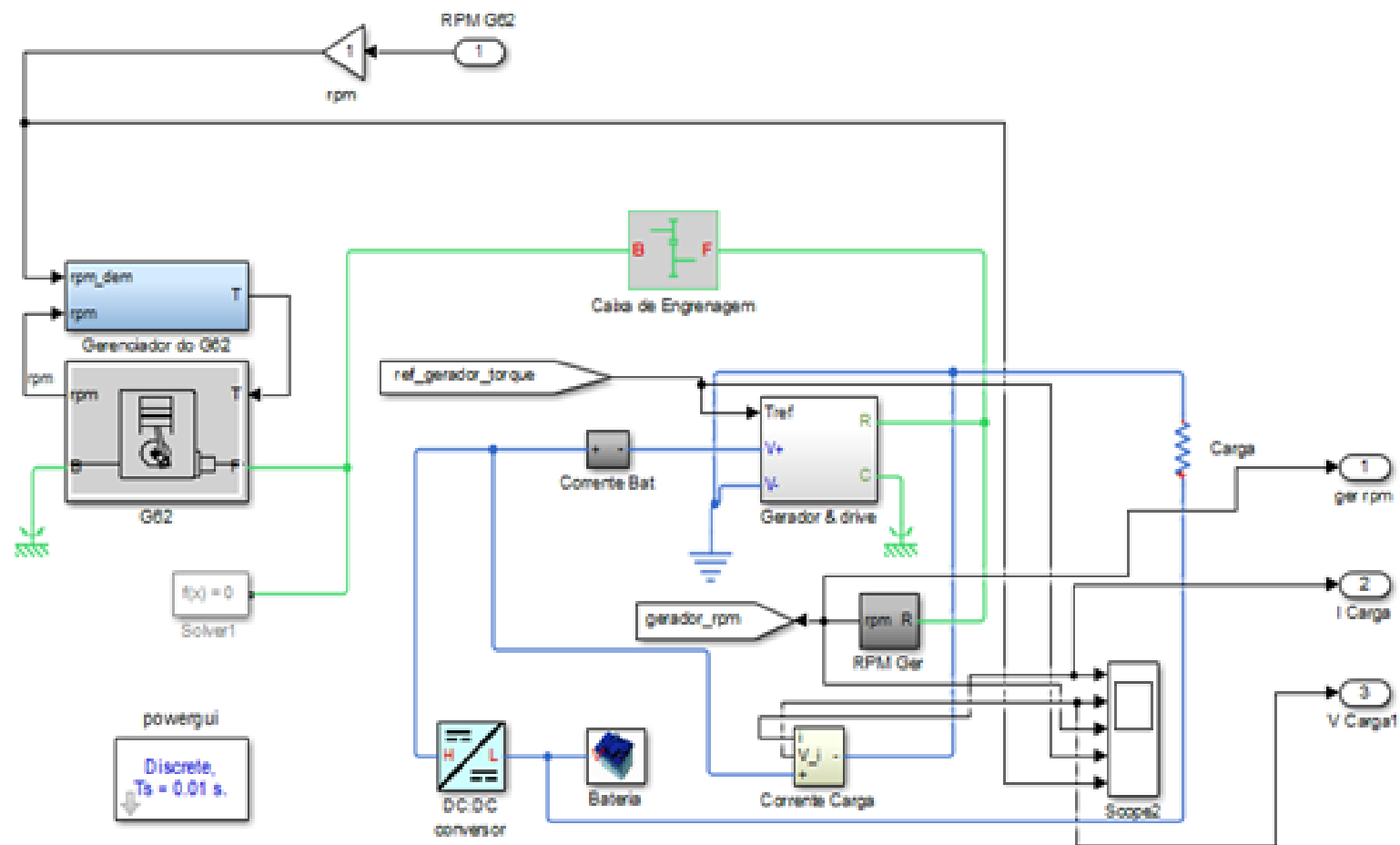
A Figura 41 exibe um dos modelos desenvolvidos. No esquema é possível ver que as quatro saídas de sinais que são consideradas fundamentais são a corrente e a tensão na carga, tensão do gerador, a rotação no gerador.

As simulações feitas em computador objetivaram analisar o comportamento do Sistema sem controle. Foram feitas várias alterações na arquitetura, com valores diferentes nos blocos

pré-existent da ferramenta. Modificações nestes valores foram testadas tentando-se aproximar das características dos componentes reais montados em bancada. O motor de combustão foi representado por um elemento genérico.

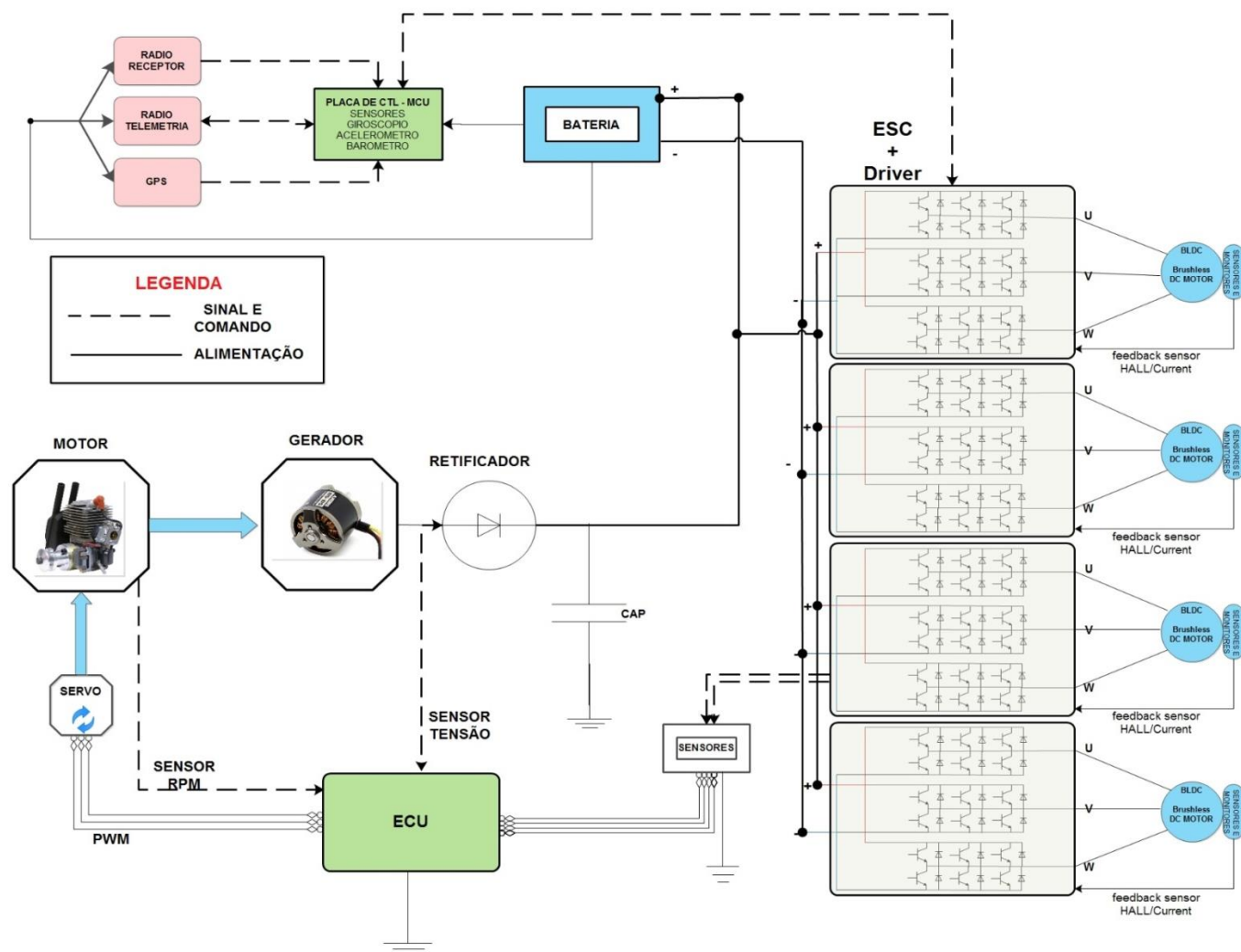
Na Figura 42 tem-se o esquema detalhado do SHPE, onde observa-se todos os blocos principais que comporão o sistema a ser instalado no RPA. Tem-se neste esquema os principais componentes do RPA, e como estão interligados. Os sensores dos motores de propulsão e os sinais dos ESCs servem como sinais de realimentação para a placa de controle de voo.

Figura 41 – Esquema de planta simulada no Simulink®.



Fonte: Produção do próprio autor.

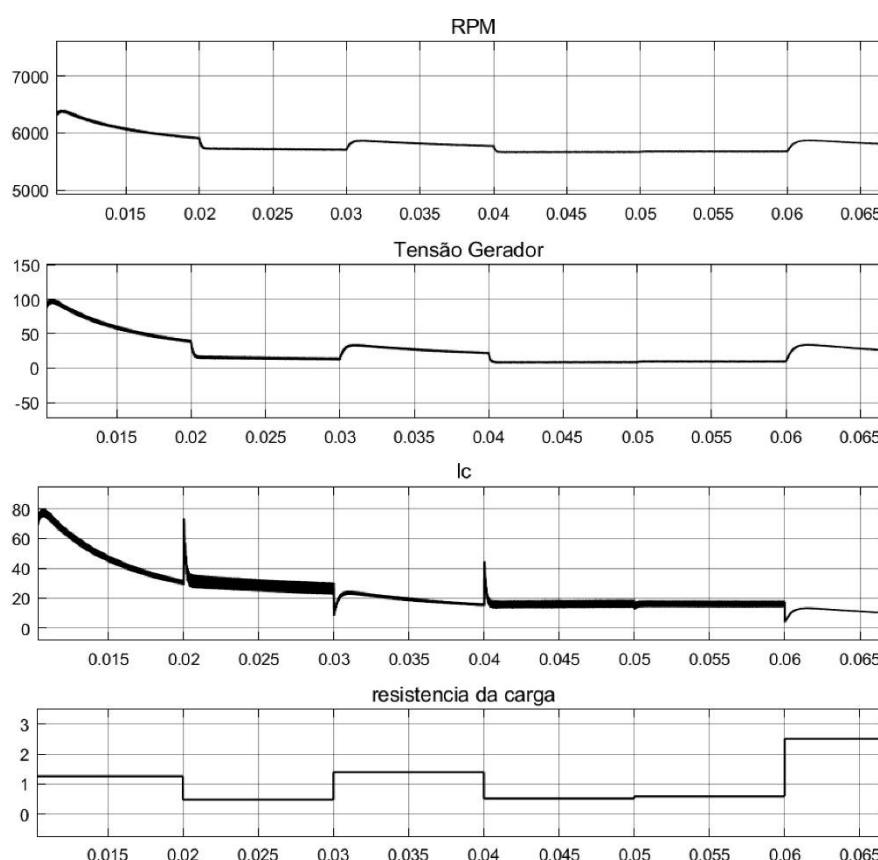
Figura 42 – Desenho detalhado do sistema.



Fonte: Produção do próprio autor.

Na Figura 43, estão desenhados os gráficos que foram coletados deste segundo esquema. Nestas simulações foi adotado um chaveamento aleatório da carga. A ideia é verificar o comportamento do conjunto através da variação abrupta da carga, para representar fielmente os acionamentos rápidos dos motores elétricos no RPA.

Figura 43 – Gráficos do primeiro esquema Simulado.



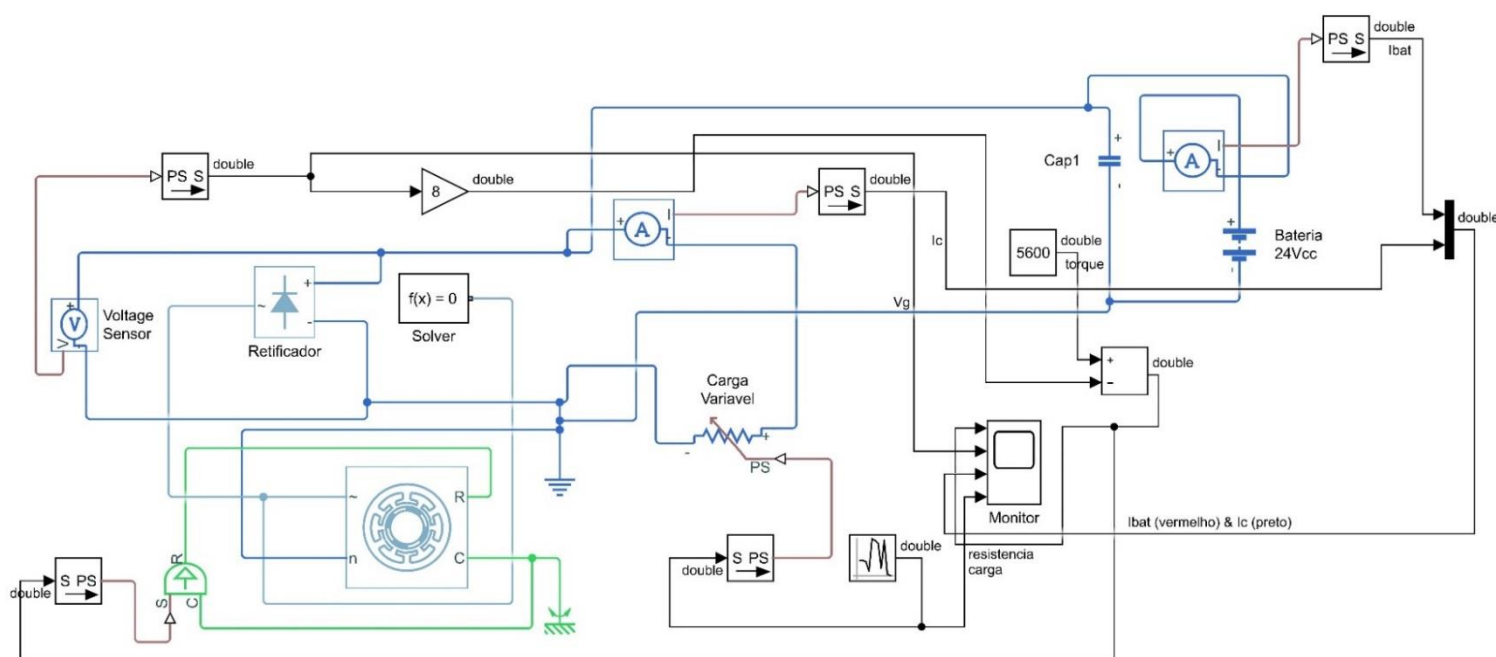
Fonte Produção do próprio autor.

Nota-se no gráfico da rotação do gerador, a variação da rotação em função da energia solicitada pela carga, representando a reação desse tipo de máquina.

No projeto do sistema o controle é o responsável pela correção automática deste efeito na rotação, sendo que fora dos limites propostos o responsável pelo fornecimento de energia passará a ser o banco de baterias

Na Figura 44, está desenhado outro esquema simulado, agora com o banco de baterias, sem o controle.

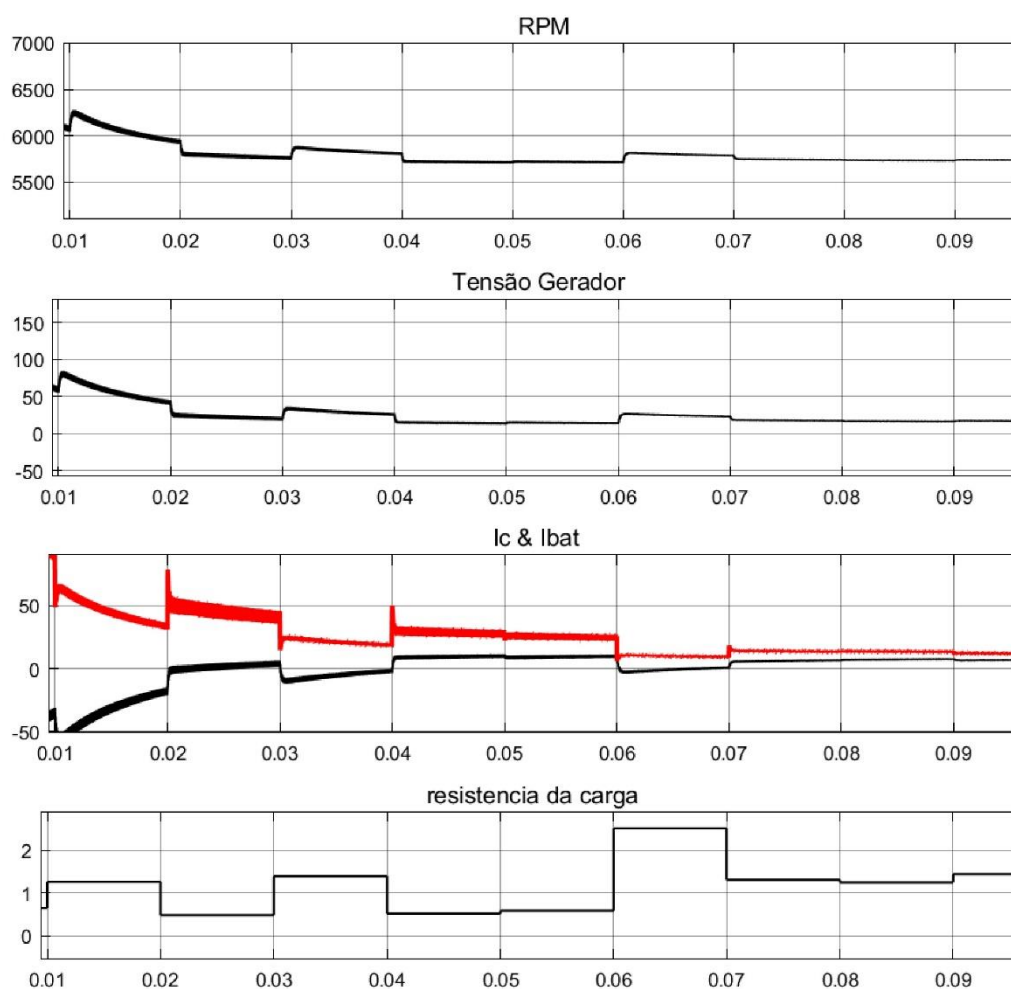
Figura 44 – Desenho do segundo esquema Simulado com Bateria.



Fonte: Produção do próprio autor.

Na Figura 45, estão desenhados os gráficos considerando o acréscimo do banco de baterias, com tensão nominal de 24 Vcc. Vê-se que no instante inicial a bateria recebe uma corrente. Após este período a bateria fornece energia, juntamente com o sistema gerador, dependendo da demanda da carga, como era esperado.

Figura 45 – Gráficos do segundo esquema Simulado.



Fonte: Produção do próprio autor.

Apesar de algumas diferenças em relação ao protótipo, foi útil para avaliar algumas reações em função de alterações nos componentes, e desse modo levantar algumas curvas para se direcionar o trabalho.

3.4.4 RNA desenvolvida

O fluxo de trabalho para o processo geral de criação de uma rede neural possui sete etapas principais:

- Coletar dados
- Criar a rede
- Configurar a rede
- Inicializar os pesos e vieses
- Treinar a rede
- Validar a rede (análise pós-treinamento)
- Usar a rede

Foram usados dois softwares para o desenvolvimento da RNA. O primeiro software foi o Matlab® juntamente com a *Toolbox 'Neural Network'*. O segundo software usado foi o WEKA® versão 3.8, que é um software de código aberto emitido sob a GNU '*General Public License*'. Este software é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Ele contém ferramentas para preparação, classificação, regressão, '*clustering*' (mineração de regras de associação) e visualização de dados.

3.4.4.1 Dados de Entrada para RNA Multicamadas

A Normalização e o Condicionamento dos dados de entrada para posterior processamento se fazem necessários porque muitas vezes os dados disponíveis para treinamento da RNA não estão padronizados de forma adequada.

A utilização de redes neurais com função de transferência *sigmoide* usada neste trabalho, exige que estes dados estejam normalizados no intervalo [0,1]. Neste projeto, os dados foram gravados em arquivo Excel®, em unidades e faixas de valores distintos entre os parâmetros e posteriormente foi salvo em '.cvs' ou em '.arff', para uso no Matlab® e no WEKA®. Outros aplicativos, como por exemplo o uso do EXCEL® com *solver* específico, foram usados como apoio, mas não foram incluídos neste texto.

Com o intuito de corrigir o problema das faixas de valores distintos, faz-se necessário que dados multivariados sejam normalizados, padronizados em uma mesma escala. O

propósito da normalização é minimizar os problemas oriundos do uso de unidades e dispersões distintas entre as variáveis. As variáveis podem ser normalizadas segundo a amplitude ou segundo a distribuição.

Foi desenvolvido um script no ambiente computacional Matlab®, usando as funções de normalização internas do programa ($N = \text{normalize}(A, \text{dim})$) para implementar esta técnica de normalização linear no intervalo $[0,1]$, de acordo com as equações (22) e (23):

$$X_n = \frac{X_o - X_{min}}{X_{max} - X_{min}} \quad (22)$$

Que resulta em:

$$X_o = X_n(X_{max} - X_{min}) + X_{min} \quad (23)$$

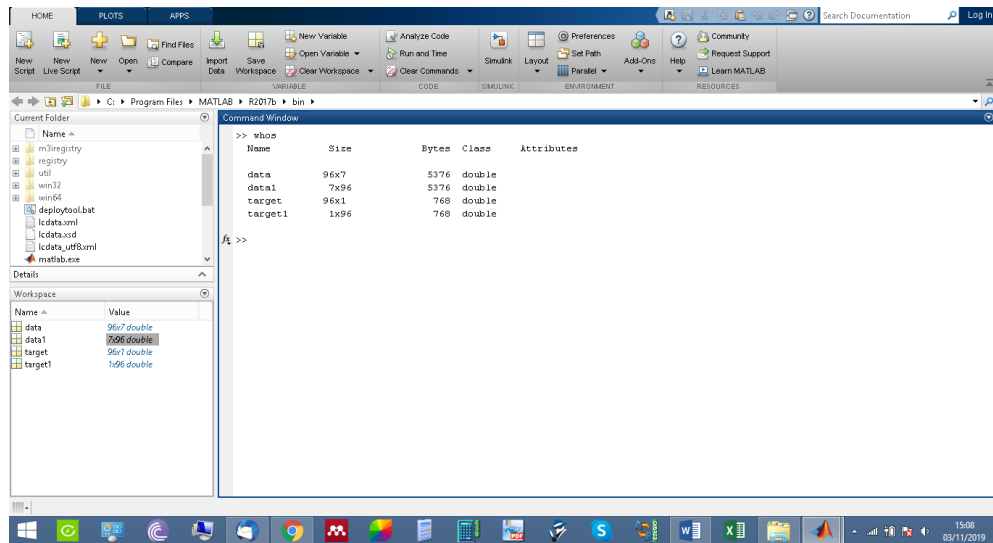
sendo: X_o : x original; X_n : x normalizado; X_{max} : maior valor de x por parâmetro; X_{min} : menor valor de X por parâmetro.

O script do Matlab®, é o seguinte:

```
I = [ 1 2 4; 3 2 4]; % Alguma matriz n x m com valores quaisquer.
scaledI = (I-min(I(:))) ./ (max(I(:))-min(I(:)));
min(scaledI(:)) % the min is 0
max(scaledI(:)) % the max 1
scaledI
```

A Tabela apresentada no Apêndice C serviu de base para o uso na toolbox ‘*nntool*’ do Matlab®. Na Figura 46 é indicada a janela com o ‘*worksheet*’, e com dados de entrada e saída, para criação da RNA.

Figura 46 – Janela apresentado o worksheet do Matlab®.

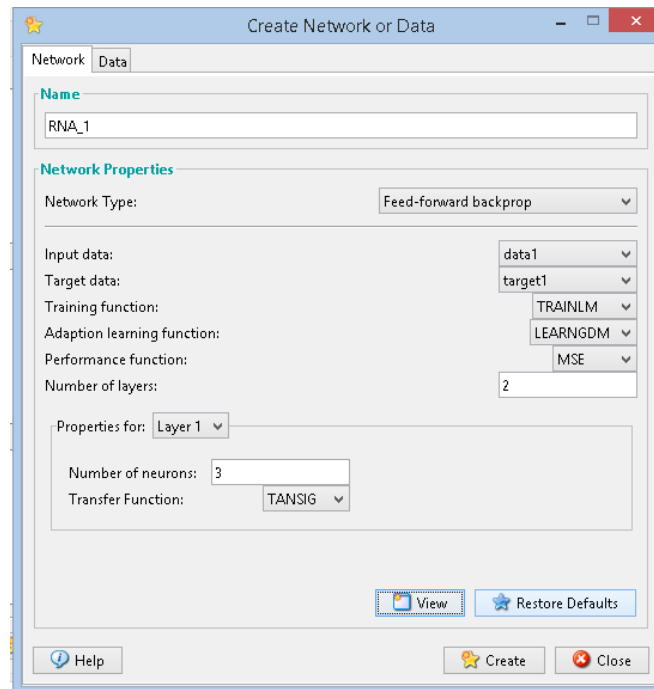


Fonte: Produção do próprio autor.

Na Figura 47 é mostrada a próxima janela, já com os dados importados, tanto os atributos como também a saída desejada.

Existem várias outras funções de treinamento disponíveis na *nn toolbox* do Matlab®.

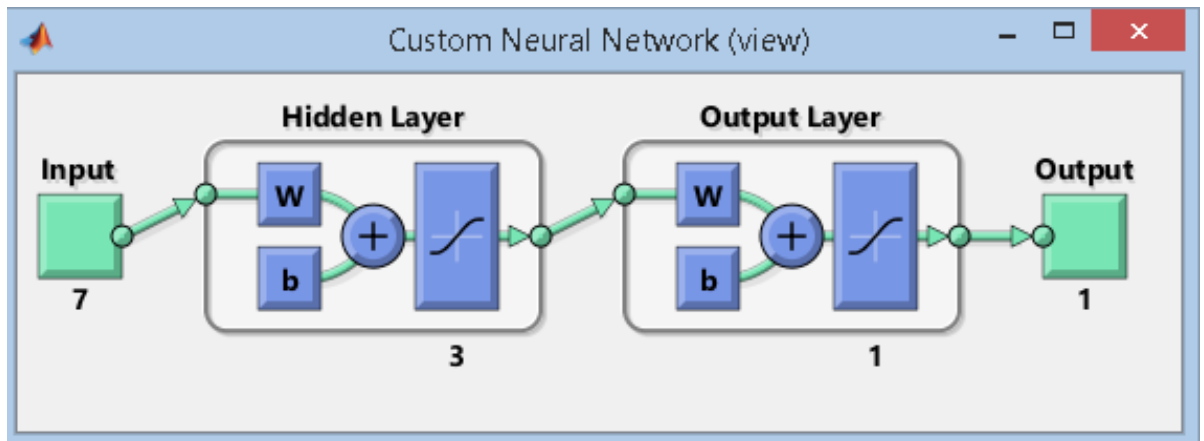
Figura 47 – Janela após a importação dos dados do arquivo.



Fonte: Produção do próprio autor.

Na Figura 48, está indicada a janela, após treinamento, da RNA criada, com 3 neurônios na camada oculta, e com uma camada de saída.

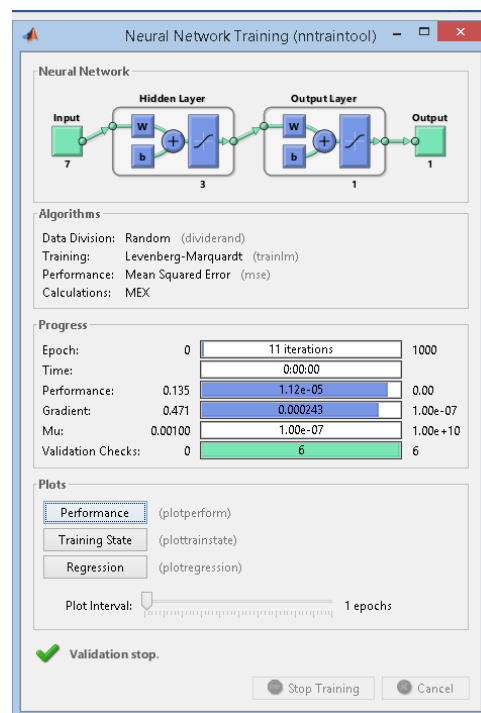
Figura 48 – Janela indicando a topologia utilizada.



Fonte: Produção do próprio autor.

Na Figura 49 está a janela do treinamento executado da *nn toolbox*. Pode-se observar que após 26 interações foi atingido o valor de erro mínimo adotado, não sendo necessário a conclusão do número de épocas escolhido.

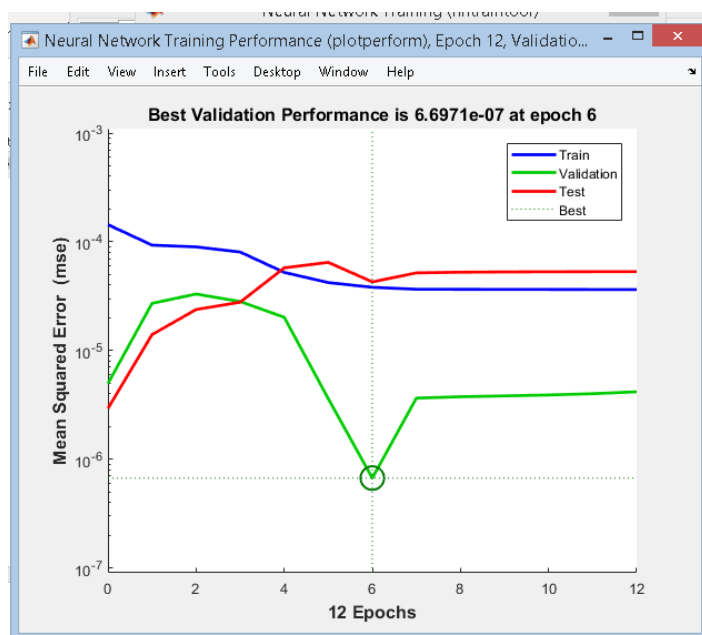
Figura 49 – Janela do *nn toolbox*, detalhando o treinamento da RNA.



Fonte: Produção do próprio autor.

Após o uso e avaliações feitas no *app*, o treinamento foi exportado para o *workspace* do Matlab®. Com isso pode-se treinar outros dados ou usar os dados já treinados para execução do código. Vê-se que na *época* 6 se atingiu o melhor nível de performance na validação. Na Figura 50 está a janela onde é mostrado os dados de performance alcançados.

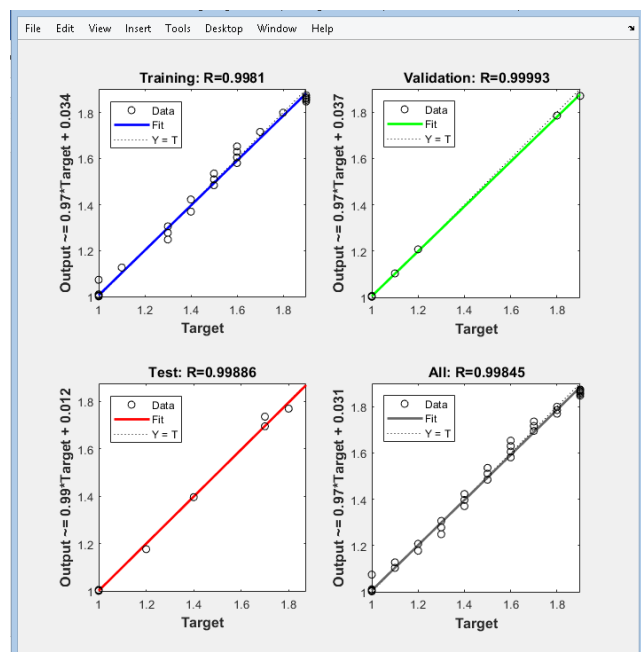
Figura 50 – Gráfico da Performance.



Fonte: Produção do próprio autor.

Os dados plotados da regressão para o treinamento, a validação e o teste executado estão indicados na Figura 51. Foram feitas algumas variações nos valores principalmente de configuração da camada oculta, para se avaliar alguma alteração na performance, mas este foi o melhor resultado alcançado. Observa-se nos gráficos de teste da Figura 51 o sucesso conseguido no treinamento dos dados coletados.

Figura 51 – Gráficos da regressão dos dados após treinamento e validação.



Fonte: Produção do próprio autor.

Outra ferramenta de software aplicada foi o WEKA®, conforme já mencionado no tópico 3.4.4. A mesma tabela de dados (APÊNDICE C) foi importada no aplicativo. Este software tem aplicação em mineração de dados e possui os principais algoritmos empregados em aprendizado de máquina, como referenciado nas págs. 469 a 471 do livro ‘*Data Mining*’ (IAN H. WITTEN, EIBE FRANK, 2011).

O procedimento continua o mesmo, faz-se o treinamento, a validação e o teste com os dados coletados. Foi feito um treinamento com 66% dos dados e com 1000 épocas. O teste foi executado com o restante da tabela. Aplicando-se a topologia *multilyer perceptron* neste software, e variando-se as configurações no final obteve-se o seguinte relatório, gerado de forma automática que mostra tanto o resultado do treinamento como o de teste, o qual indica a diferença do valor real para o previsto:

=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
 Relation: planilha
 Instances: 96
 Attributes: 6
 Temp
 Altura
 SensorIc
 sensorIcN

rpm
PWM
Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

Linear Node 0

Inputs Weights
Threshold 0.5191566454932228
Node 1 -1.784438976410133
Node 2 2.1369262388073964
Node 3 -1.3815978437238028

Sigmoid Node 1

Inputs Weights
Threshold -1.0326883140735024
Attrib Temp -0.11425310169024167
Attrib Altura -0.19824530016323155
Attrib SensorIc 1.592784175579158
Attrib sensorIcN -1.553208795701764
Attrib rpm 0.08257156967579692

Sigmoid Node 2

Inputs Weights
Threshold 0.4096792612406332
Attrib Temp -0.11042763013383215
Attrib Altura -0.1392349907341234
Attrib SensorIc -1.6221729968353755
Attrib sensorIcN 1.4829758059266736
Attrib rpm 0.18277828055334747

Sigmoid Node 3

Inputs Weights
Threshold -1.00821040554496
Attrib Temp -0.10452376041285204
Attrib Altura -0.015202035681904959
Attrib SensorIc 1.114112033407984
Attrib sensorIcN -1.3080458907902752
Attrib rpm 0.25618186260472636

Class

Input
Node 0

Time taken to build model: 0.04 seconds

=== Evaluation on test split ===

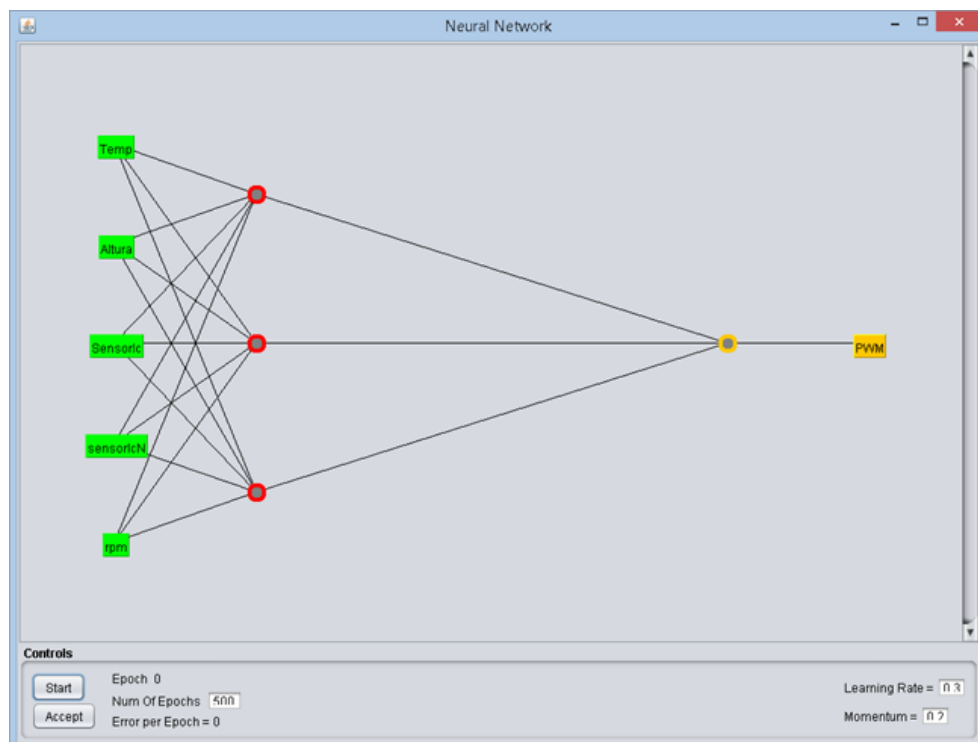
Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient	0.9999
Mean absolute error	0.0897
Root mean squared error	0.1443
Relative absolute error	1.1119 %
Root relative squared error	1.5192 %
Total Number of Instances	33

Na Figura 52, está desenhada a janela com o desenho da RNA, fornecido pelo WEKA®.

Figura 52 – Desenho da RNA com algoritmo MLP.



Fonte: Produção do próprio autor.

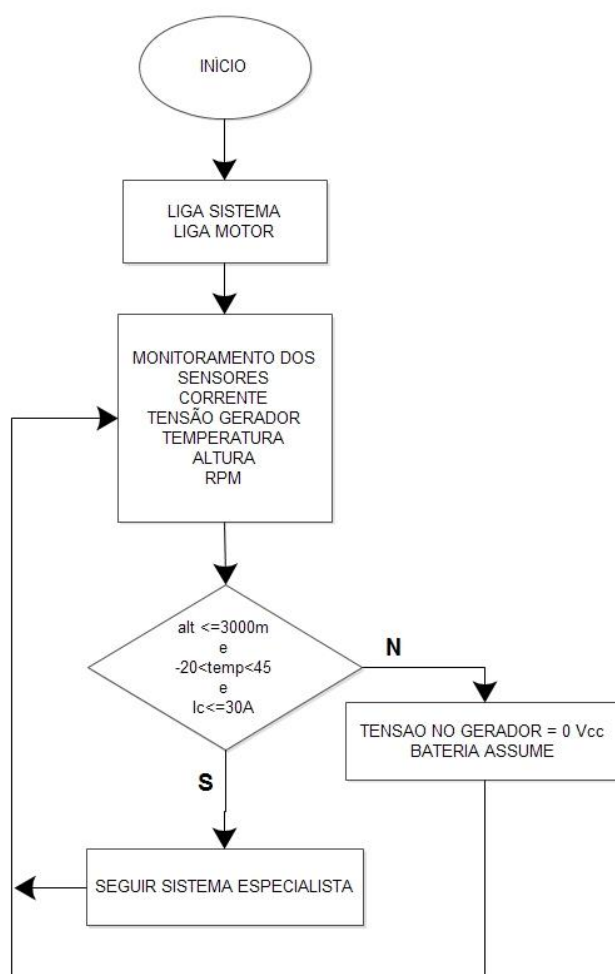
O erro relativo alcançado chegou perto de 1,5% nesta classificação. Outros algoritmos foram testados antes de se decidir pela configuração final.

Como foi feito no Matlab[®], diversas outras configurações foram testadas, mas o resultado desta última foi o escolhido. Serviu de base para o sistema especialista os resultados do WEKA[®].

3.4.5 Software com controle RNA

Foi desenvolvido o *firmware* na placa do microcontrolador Mbed em linguagem C++. O código sofreu vários aprimoramentos após testes em bancada e teve como base o fluxograma da Figura 53.

Figura 53 – Fluxograma do software de controle do motor baseado no Sistema Especialista.



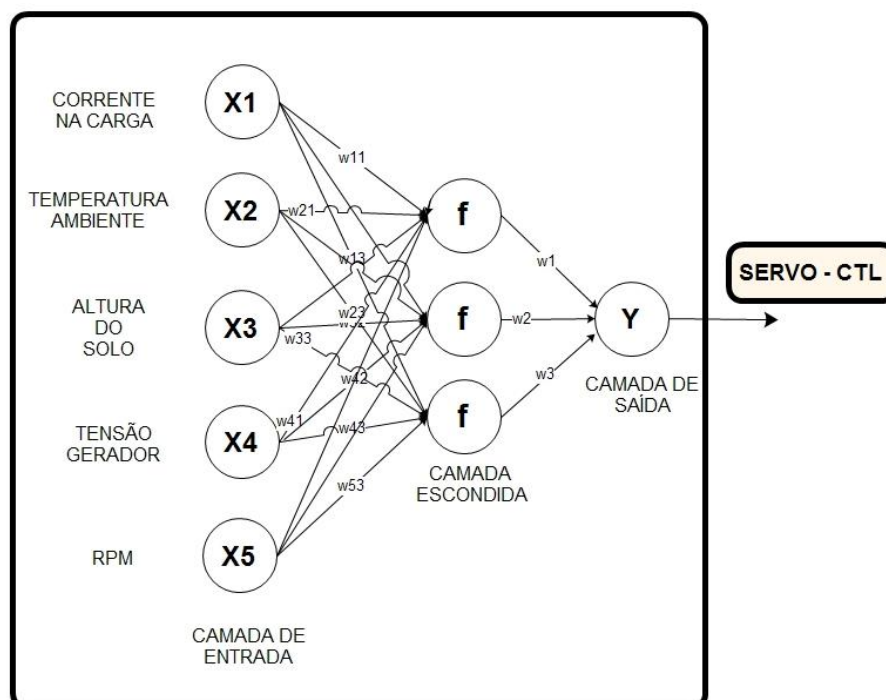
Fonte: Produção do próprio autor.

Neste fluxograma pode-se destacar o bloco que encaminha o controle para o Sistema Especialista. Este Sistema se refere aos dados coletados do sistema após seu treinamento, de acordo com as variáveis de entrada.

O Sistema Especialista que foi incluído no código foi criado com o apoio das ferramentas citadas. Seguiram-se vários testes em bancada. Esse procedimento foi usado para se observar o comportamento final do conjunto.

A Figura 54 contém o esboço da rede que foi usada no projeto, indicando as respectivas entradas.

Figura 54 – RNA aplicada no código RNA em C++.



Fonte: Produção do próprio autor.

A temperatura e a altura do solo são sinais que não foram coletados, mas estão incluídos na tabela final, e no programa desenvolvido. A potência indicada foi resultado dos sinais da multiplicação dos sinais oriundos dos sensores de tensão após o *boost* pela corrente na carga.

O código final foi inserido no Apêndice E.

3.4.6 Controle PID

Neste tópico será descrito todos os procedimentos feitos para definição e execução do firmware do controle clássico. Nos últimos anos, foram desenvolvidos vários métodos de determinação dos parâmetros do controlador PID. Um modelo paramétrico pode ser obtido através da modelagem (utilizando-se as equações e leis físicas) ou através de um procedimento experimental conhecido com identificação de sistemas dinâmicos, em que sinais de excitação adequados são aplicados ao processo para se calcular os parâmetros do processo.

Na execução do controle foi utilizado o seguinte método:

- Coleta de dados do protótipo instalado;

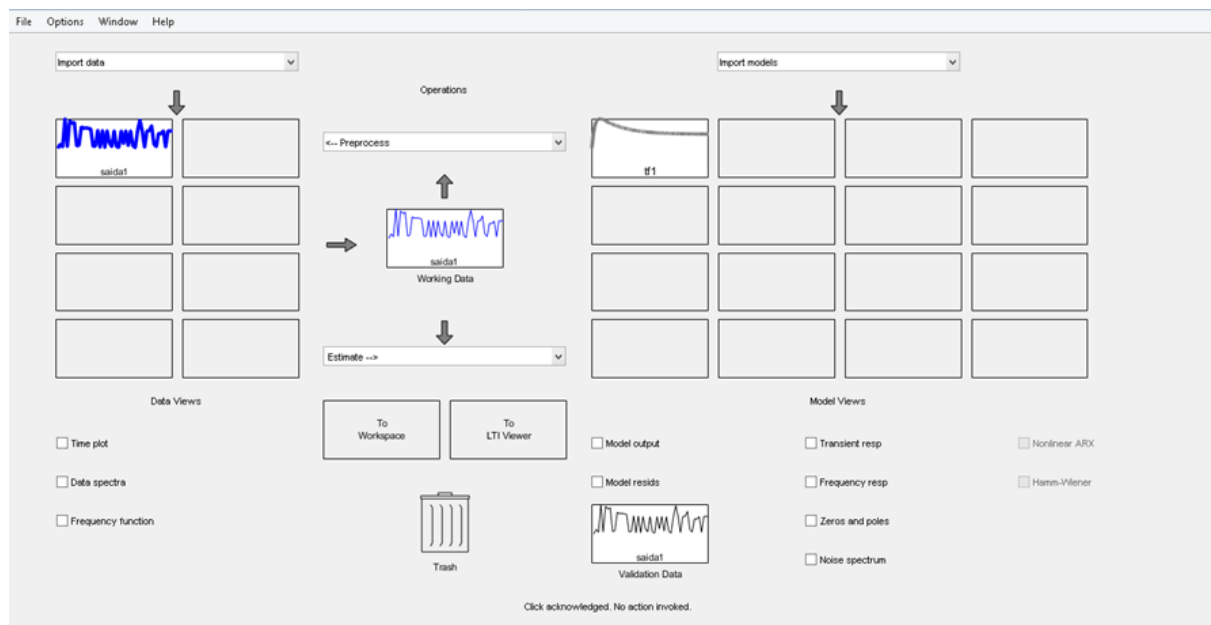
- Uso do módulo ‘*System Identification*’ do MATLAB®Simulink para se determinar a FT (função de transferência) do sistema;
- Simulação do sistema em malha fechada com um controle PI ou PID;
- Determinação das características ótimas do controle;
- Inclusão dos valores definidos no firmware na placa do microcontrolador;
- Testes e conclusões.

A seguir serão descritos os principais passos adotados na execução desta etapa dos trabalhos.

Passo 1: Inserção dos dados coletados no Matlab®.

As tabelas com os dados de rotação e tensão no gerador, coletados pelo sistema de coleta, foram inseridas no programa no formato .txt. Em seguida, como os dados estão no formato ‘*array*’, foi retirada a coluna ‘tempo’ da tabela, ficando somente com a coluna dos dados do sensor. Como dado de entrada usou-se a rotação do gerador ‘rot’ e como dado de saída a Tensão ‘Vg’. Neste ponto é acessado o ‘*toolbox*’ ‘*System Identification*’, onde são importados os dados no domínio do tempo. A Figura 55 representa a janela da ‘*toolbox*’.

Figura 55 – Janela da ‘*toolbox*’ do Matlab®.

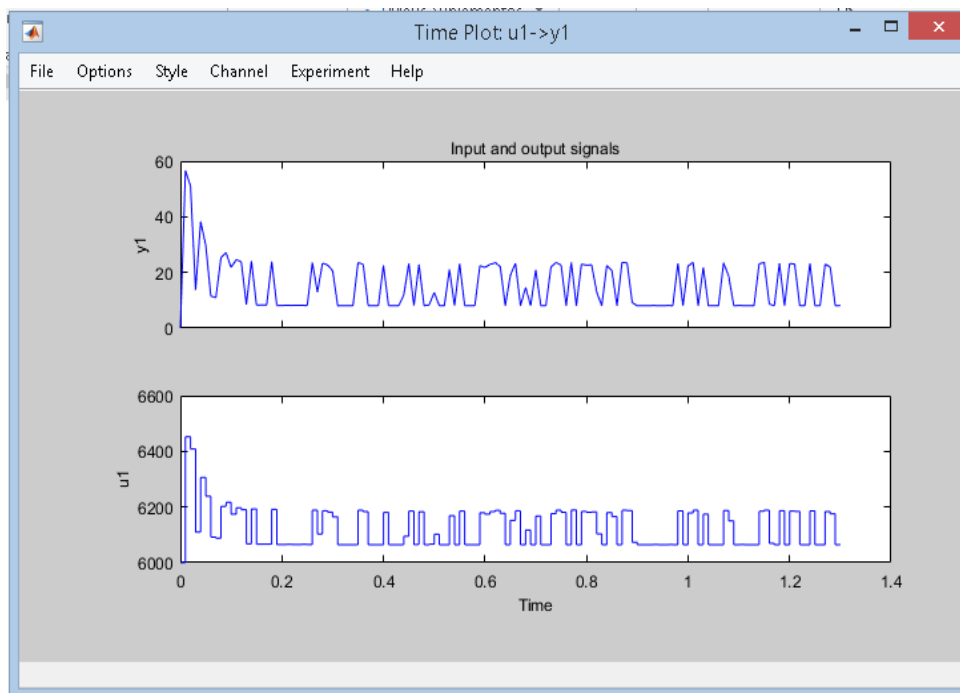


Fonte: Produção do próprio autor.

Passo 2: Inserção dos dados de entrada e saída na ‘toolbox’.

Nessa etapa obtém-se a plotagem dos dados em função do tempo, após a inserção dos dados de entrada e da saída no domínio do tempo, conforme aparece na figura 55.

Figura 56 – Plotagem dos dados de entrada e saída.

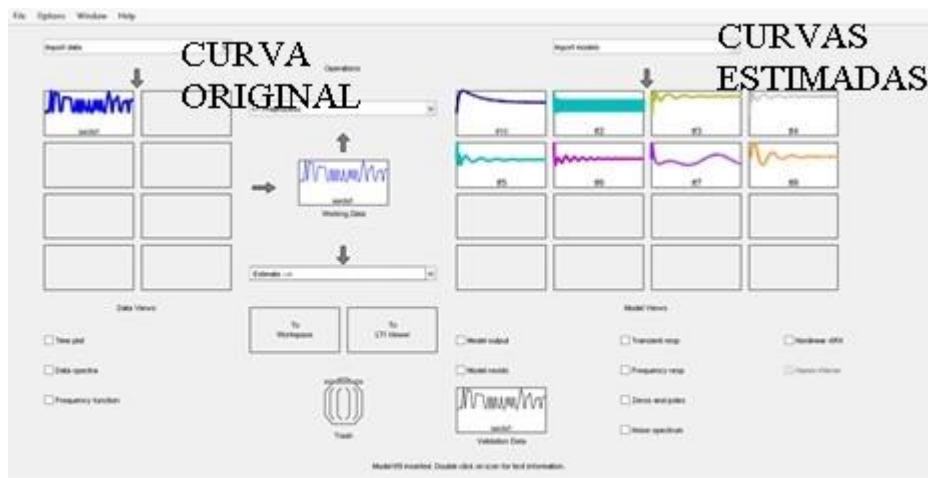


Fonte: Produção do próprio autor.

Passo 3: Processo de estimativa da Função de Transferência.

Nesse passo diversas tentativas de definição e procura da função de transferência que melhor represente o sistema são feitas. Cada estimativa produz um gráfico, que fica guardado na memória para comparação posterior. Na Figura 57 está reproduzida a janela na qual é possível verificar as estimativas feitas com a ferramenta.

Figura 57– Janela do aplicativo indicando as várias estimativas realizadas com os dados.

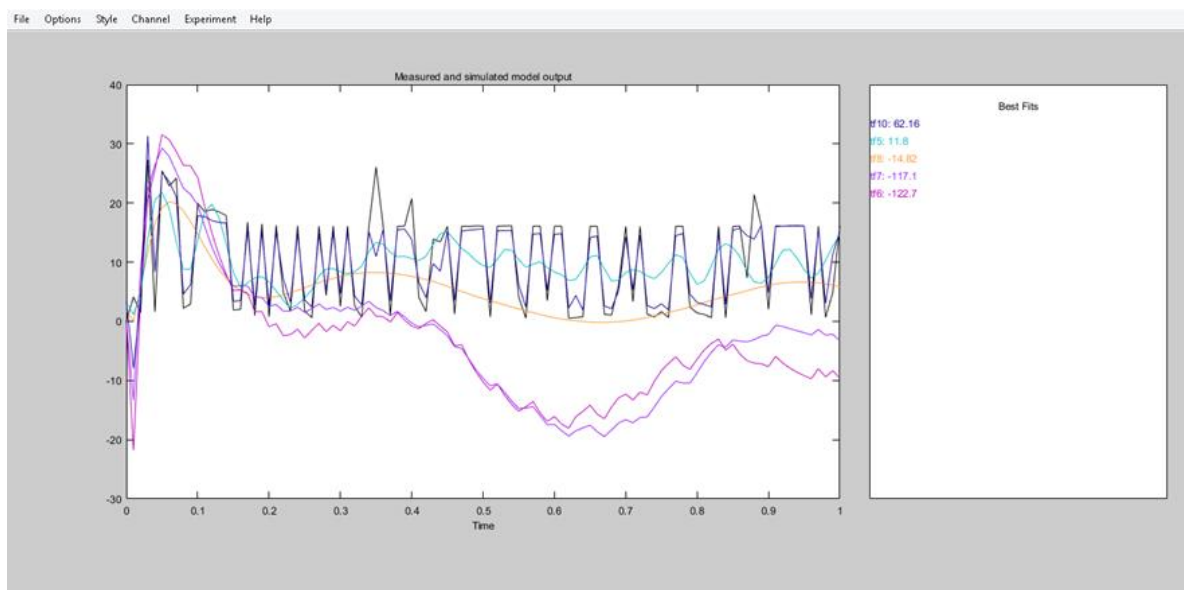


Fonte: Produção do próprio autor.

Passo 4: Nessa etapa é definida a melhor estimativa.

Como definido pelo próprio aplicativo, uma função de transferência com reprodução maior ou igual a 50% da forma de onda original já é considerada satisfatória. Neste caso como pode ser visto na Figura 58, conseguiu-se uma fidelidade de 62,16% em relação a curva original.

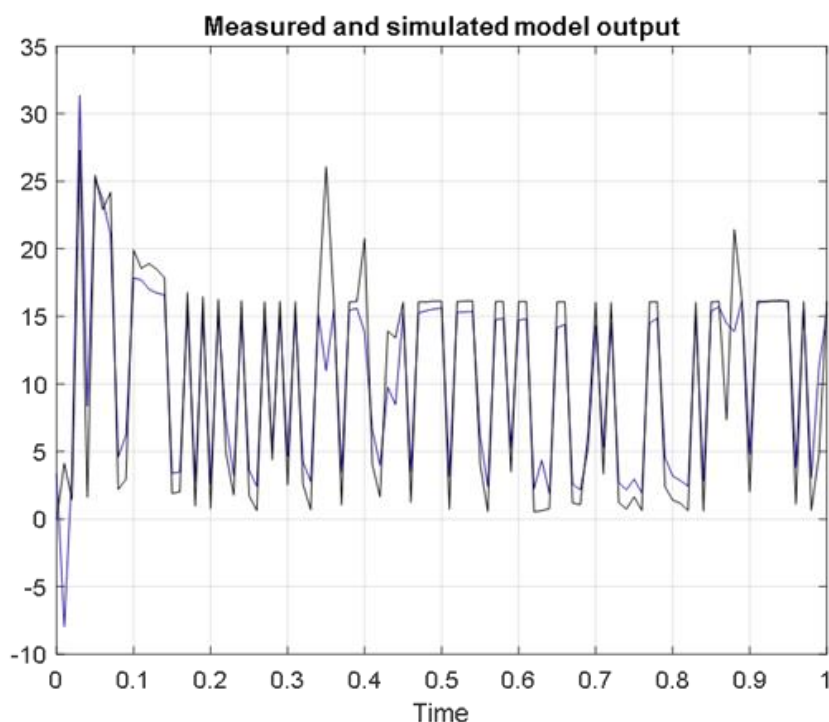
Figura 58 – Janela do aplicativo que descreve as estimativas testadas da FT.



Fonte: Produção do próprio autor.

Na Figura 59, está mostrada a estimativa escolhida juntamente com a curva de saída original, que tem suas características descritas no relatório apresentado no Apêndice G.

Figura 59 – Curvas obtidas com 62,16% de fidelidade da curva do sinal de saída original.



Fonte: Produção do próprio autor.

Passo 5: Função de Transferência obtida.

Nesta etapa a FT é validada com outros dados obtidos em bancada, para ver se é representativa e depois será implementada no Simulink® para se definir as características do controle PID.

A expressão (21) representa a FT obtida:

$$\frac{y}{u} = \frac{-0,8187 \times S^3 - 116,5 \times S^2 - 2498 \times S + 3471}{S^3 + 139,3 \times S^2 + 2637 \times S + 5291} \quad (21)$$

O resultado da validação, feita com outros dados no próprio aplicativo, foi positivo, com 61,34% de fidelidade, portanto o próximo passo é a execução de uma simulação no Simulink® para se concluir o controle.

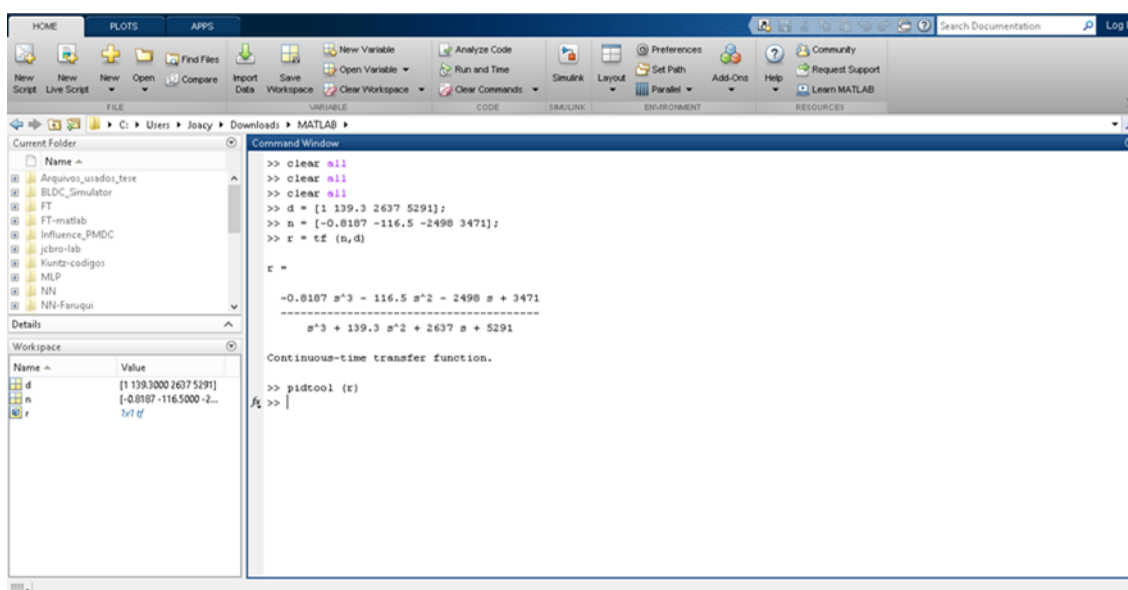
Passo 6: Desenvolvimento do Controle.

A FT é útil para se determinar a estabilidade do sistema, para se conseguir a resposta em frequência, projetar um filtro com base nas informações e para se projetar um controle para este sistema.

Nesse último passo a FT foi inserida no Matlab® e trabalhada no *toolbox PIDtool*, para uma rápida definição das melhores características do controle em malha fechada. Este processo permite uma rápida análise dos parâmetros que melhor se adequam ao controle, como tempo de subida, *overshoot*, comportamento estável ou não estável, ganhos e definição do melhor tipo de controle.

Nesse processo chegou-se à conclusão que o controle *PI*, que normalmente é o mais usado, também neste caso ele se aplica. Na Figura 60 está apresentada uma janela do *workspace* do Matlab®.

Figura 60 – Janela do aplicativo.



Fonte: Produção do próprio autor.

Ajustando-se na *toolbox* os parâmetros do controle *PI*, para que ele tenha comportamento que se espera da resposta do sistema obteve-se os valores dos parâmetros mostrados na Figura 61.

Figura 61 – Parâmetros ajustados conforme resposta desejada.

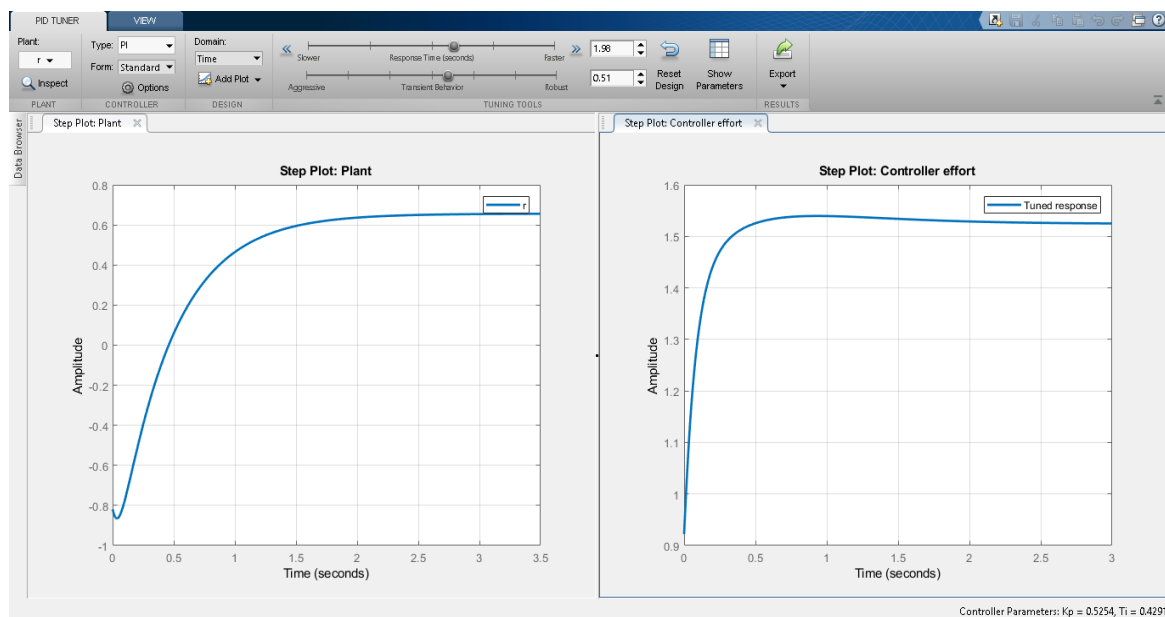
Controller Parameters	
	Tuned
Kp	0.70859
Ki	1.1263
Kd	n/a
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	1.16 seconds
Settling time	2.9 seconds
Overshoot	0 %
Peak	1.91
Gain margin	2.61 dB @ 13.5 rad/s
Phase margin	60 deg @ 1.01 rad/s
Closed-loop stability	Stable

Fonte: Produção do próprio autor.

O controlador *PI*, é um dos mais utilizados por ser simples e eficaz. O projeto do controlador busca uma resposta transitória rápida com baixo *overshoot* (no caso=0), além de que o erro em regime permanente é reduzido. O canal *P*, proporcional ao erro, age imediatamente, sempre que houver alteração da referência ou da perturbação, enquanto o fator integral, acumula o erro (passado), reduzindo o erro em regime permanente.

As margens de ganho e fase indicam a quantidade de incerteza que um sistema de realimentação pode tolerar antes de se tornar instável. A margem de ganho está relacionada à mudança de ganho (processo ou controlador) que pode ser tolerada, enquanto a margem da fase está relacionada à alteração no atraso de tempo que pode ser suportada. Na prática, é recomendado uma margem de ganho de pelo menos 2,5 (no caso=2,61) e margem de fase de pelo menos 60 ° (no caso=60 °) para garantir robustez. Na Figura 62 está a janela do aplicativo com as curvas de sinal de entrada tipo degrau e sinal ajustado pelo controle *PI*.

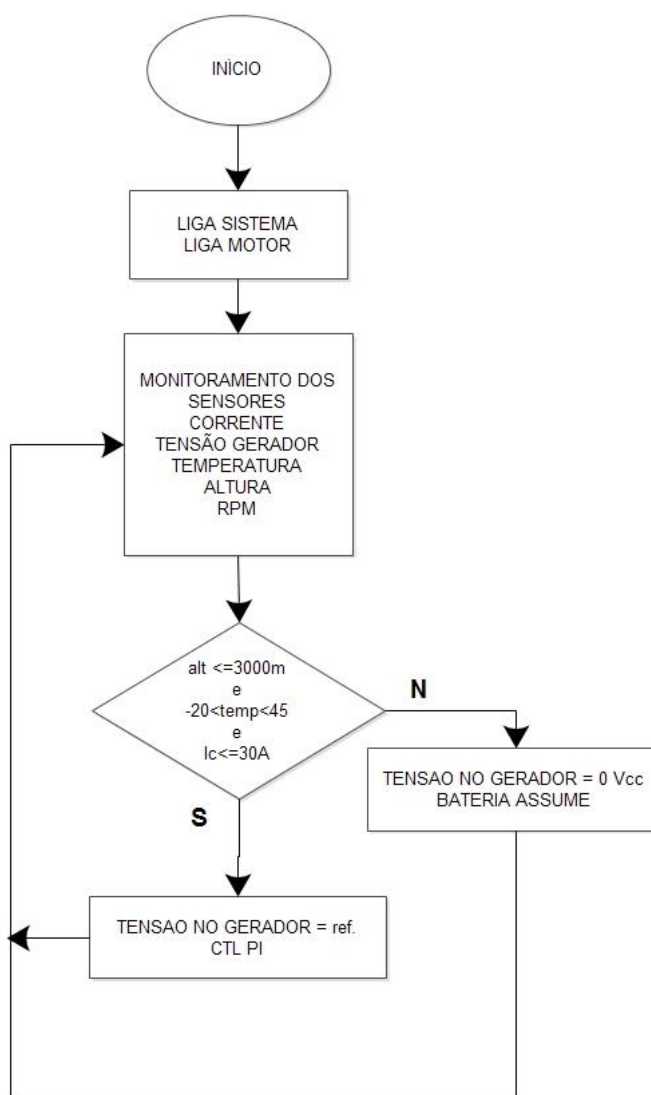
Figura 62 – Janela com os gráficos das curvas após ajuste.



Fonte: Produção do próprio autor.

Com este ajuste nos ganhos do software testes foram feitos e obteve-se um comportamento adequado para o controle. Esse segundo código (*firmware*) de controle desenvolvido, foi baseado no *PI* clássico. No fluxograma apresentado na Figura 63, são detalhados os passos necessários incluídos no código.

Figura 63 – Fluxograma do software de controle do motor baseado em PI.



Fonte: Produção do próprio autor.

A Sintonia de um controlador é um processo de otimização que envolve um critério de desempenho relacionado com a forma de resposta do controlador e com o erro entre a variável de processo e o *setpoint*.

Neste trabalho foi elaborado um código em linguagem C++ com sintonia dos ganhos e do tempo conforme abordado neste tópico. Cada ajuste nos ganhos provocou a reinserção do *firmware* na placa. O código final foi inserido no Apêndice H.

4 DISCUSSÃO DOS RESULTADOS

O presente trabalho procurou acrescentar alguns pontos ainda não explorados na parte de controle e na utilização de aprendizado de máquinas em microcontroladores da nova geração.

Aplicações de outras fontes alternativas de energia para uso tanto como backup como instalado em paralelo estão sendo pesquisadas para uso em veículos não tripulados, o que indica que é uma fronteira do conhecimento a ser explorada.

Como pontos positivos pode-se destacar:

1. As simulações do sistema no MATLAB® Simulink, que se mostraram uteis no estudo das várias configurações possíveis implementadas no hardware, possibilitando a escolha prévia dos componentes instalados no protótipo.
2. A criação do Código com o Sistema Especialista previamente treinado apresentou resultados satisfatórios, conforme pode ser visto na Figura 64.
3. A configuração adotada com o controle através da placa Mbed NXP LPC1768 mostrou-se adequada.

Foram considerados pontos negativos no desenvolvimento:

1. O código desenvolvido para o Controle PID não obteve os resultados esperados, principalmente devido aos ganhos obtidos. A definição da função de transferência através do MATLAB® Simulink precisa ser mais bem trabalhada, mas fornece um método interessante que precisa ser mais pesquisado.
2. O uso de cabos não blindados nas ligações dos sensores captou muito ruídos que interferiram nos dados coletados.
3. O gerador tipo *outrunner* utilizado apresenta mais dificuldade na sua instalação.

Na Figura 64 tem-se uma amostra do controle obtido com o código RNA. Essa Figura mostra a variação da tensão do gerador (V_g) com a variação da corrente (I_c) consumida. O controle com o sistema especialista implementado já demonstra ser adequado para se manter a tensão de saída da fonte *boost* no valor pré-ajustado.

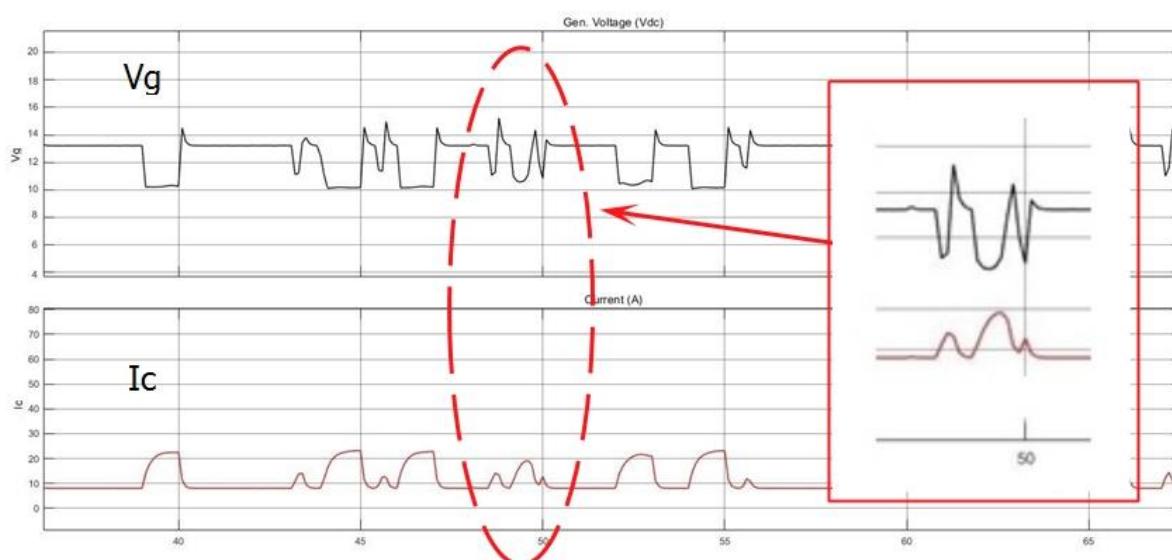
Figura 64 – Gráfico com os sinais V_g e I_c com o controle implementado.



Fonte: Produção do próprio autor.

Na Figura 65 tem-se o detalhe de uma transição do consumo de corrente na carga. Observa-se que há um ligeiro atraso no sinal da tensão que tenta acompanhar a variação de corrente, mas isso não implicou em prejuízo no resultado após o *boost*.

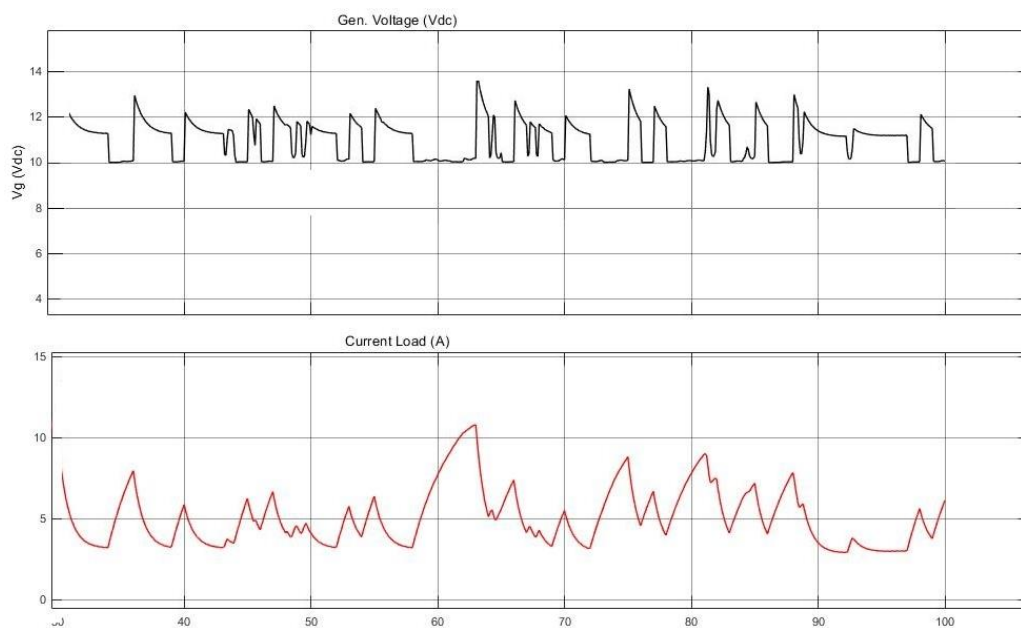
Figura 65 – Gráfico com detalhe ampliado.



Fonte: Produção do próprio autor.

Na Figura 66 tem-se uma amostra dos sinais de tensão e corrente com o sistema alimentando na saída do *boost* uma bomba d'água de 24Vcc, onde se provocou variações na vazão. Ainda que a carga não seja puramente resistiva, o sistema se comportou satisfatoriamente.

Figura 66 – Gráfico com os sinais V_g e I_c usando uma bomba d'água como carga.



Fonte: Produção do próprio autor.

Os testes na bancada com motores elétricos de propulsão não foram possíveis de serem feitos. Devido ao alto risco grande no uso desses motores com as hélices em decidiu-se fazer os testes apenas em voo.

Como pontos a serem melhorados podemos listar os seguintes:

1. O uso de outras formas de interligação entre motor e gerador, de forma a se obter um acoplamento mais flexível.
2. O uso de *vibra-stop* no conjunto deve proporcionar menos vibração e dessa forma não deve influir negativamente no voo.
3. O código PID deve passar por revisão juntamente com uma melhor definição da função de transferência do conjunto.

Mesmo se considerando a simplicidade das curvas usadas para a criação do sistema especialista, a proposta serve de base para um estudo mais profundo desta técnica, permitindo que em outros veículos ou em outras aplicações, mais complexas, as vantagens proporcionadas por este método sejam mais relevantes.

A análise dos resultados dos testes em bancada e nas simulações em computador, indica que o uso de uma rede neural artificial para ajustes dos parâmetros do controle se justifica, pois, além da simplicidade alcançada pelo método alterações devido a instalação em outros veículos seriam restritas praticamente a troca do sistema especialista incluso no código.

5 CONCLUSÃO E RECOMENDAÇÕES

O objetivo principal foi desenvolver um Sistema de Geração de Energia controlado o que se conseguiu com o protótipo construído.

Os objetivos específicos também foram alcançados. O sistema proposto aumenta a segurança nos veículos não tripulados, com a instalação de um sistema redundante, proporcionando uma alternativa em casos de emergência.

Outro objetivo foi o de desenvolver um sistema modular capaz de se adequar a diferentes necessidades operacionais e de potência sem ter que alterar totalmente o controle. O módulo de controle foi construído de maneira a propiciar esta versatilidade.

Foram implementados os conceitos relacionados com Aprendizado de Máquinas e Redes Neurais Artificiais (RNA).

Foram utilizados neste trabalho os softwares Weka[®] e MATLAB[®] Simulink para mineração de dados, definição da função de transferência e para simulação e modelamento.

O trabalho aqui desenvolvido propôs o uso e desenvolvimento de um sistema de geração de energia para proporcionar o aumento da autonomia e da segurança de voo de um veículo aéreo não tripulado. O ponto inovador dessa pesquisa se resumiu na ideia de se pensar e construir um controle a base de RNA, o que torna o inédito neste tipo de aplicação. Outro controle também foi desenvolvido, do tipo clássico PID.

Com relação ao controle tipo PID, novos testes e variações nos parâmetros do código estão sendo realizados e avaliados. Medições foram realizadas mas o sistema ainda não se obteve um resultado satisfatório ocorrendo algumas oscilações em certas condições. Uma função de transferência que represente o sistema de forma mais fiel está sendo buscada. O código que está sendo utilizado nos testes foi inserido no Apêndice H.

Considerando-se os gráficos do tópico 3.2.1, onde são abordados os dados de consumo de corrente, durante o voo indicado, e cálculos prévios com o uso das duas fontes de energia pode-se prever, apesar dos vários parâmetros influenciadores que será possível, obter-se voos de mais de duas horas e trinta minutos de duração com este sistema (tanque de um litro de combustível). É uma estimativa que será avaliada nos voos a serem realizados no protótipo de RPA que está sendo construído.

5.1 TRABALHOS FUTUROS

A continuação do trabalho deverá ter como foco uma otimização na arquitetura do sistema, procurando-se principalmente uma diminuição no peso do conjunto. Nesta direção, deve-se testar diferentes tipos de componentes, notadamente os componentes principais que tem um peso maior. O objetivo seria um aumento na eficiência e, por conseguinte, uma melhora na autonomia final. Outras medições serão realizadas, com e sem o sistema, com banco de baterias de capacidades variadas, e também variando-se o *threshold* entre um sistema e outro.

O próximo passo será a validação de uma nova função de transferência do sistema para se obter um controle adaptativo através dos ajustes dos ganhos PID em tempo real.

Será feita a implementação do algoritmo *backpropagation* no código C++ de forma a se fazer os cálculos dos ganhos no próprio *firmware*. Outro ponto importante será a inclusão do controle também na mistura ar/combustível, aumentando bastante a sua complexidade com o objetivo de se estender os limites de operação do RPA propiciando outros tipos de operações.

As futuras redes de comunicação 5G, que são esperadas para breve, farão que aplicações em tempo real via rede sejam otimizadas e possíveis de se construir. No uso deste controle seria possível algo mais complexo de forma a se atuar com dados online, inclusive nos cálculos dos ganhos em parâmetros usados no controle clássico.

Outra opção viável, a ser pesquisada, utilizando as características desta plataforma Mbed, seria a aplicação da técnica Controle Preditivo do Modelo (*Model Predictive Control* - MPC) que é um método avançado de controle de processo usado para satisfazer um conjunto de restrições (*constraints*) (SEBORG, DALE E., THOMAS F. EDGAR , DUNCAN A. MELLICHAMP, 2011).

REFERÊNCIAS

- DATASHEET. **ACS712**. Disponível em: <https://datasheetspdf.com/pdf/570845/AllegroMicroSystems/ACS712/1> Acesso em: 12 out. 2018.
- AAMIR, M. On replacing PID controller with ANN controller for DC motor position control. **International Journal of Research Studies in Computing**, v. 2, n. 1, p. 21–29, 2013.
- AKTAS, A. *et al.* **Development of a hybrid energy storage system composed battery and ultracapacitor supplied from photovoltaic power source for 3-phase 4-wire smart micro grid structure**, 2014. Disponível em: https://www.researchgate.net/publication/272424335_Development_of_a_Hybrid_Energy_Storage_System_Composed_Battery_and_Ultracapacitor_Supplied_from_Photovoltaic_Power_Source_for_3-phase_4-wire_Smart_Micro_Grid_Structure. Acesso em: 10 jul. 2018
- ANAC. **Agência Nacional de Aviação Civil**. Disponível em: <http://www2.anac.gov.br/rpas/>. Acesso em: 06 ago. 2019.
- BADR, A. Z. Neural network based adaptive PID controller. **IFAC**, v. 30, n. 6, p. 251–257, 1997. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1474667017433738>. Acesso em: 02 mar. 2018.
- BALAS, V. E. **Handbook of deep learning applications**. Springer Switzerland AG, 2019.
- BIDIKLI, B. A self-tuning PID control method for multi-input-multi. **Output Nonlinear Systems**, v. 18, n. 2, p. 218–226, 2018.
- CETIN, M.; IPLIKCI, S. A novel auto-tuning PID control mechanism for nonlinear systems. **ISA Transactions**, v. 58, p. 292–308, 2015. Disponível em: <http://dx.doi.org/10.1016/j.isatra.2015.05.017>. Acesso em: 15 mar. 2018.
- CHEN, J.; HUANG, T. C. Applying neural networks to on-line updated PID controllers for nonlinear process control. **Journal of Process Control**, v. 14, n. 2, p. 211–230, 2004.
- CHEN, S.; LAEFER, D. F.; MANGINA, E. State of technology review of civilian UAVs. **Recent Patents on Engineering**, v. 10, n. 3, p. 160–174, 2016. Disponível em: <http://www.eurekaselect.com/openurl/content.php?genre=article&issn=1872-2121&volume=10&issue=3&page=160>. Acesso em: 19 maio 2018.
- FUTABAUSA. **MG 995R**: dados técnicos. Disponível em: https://futabausa.com/?gclid=EAIaIQobChMItMmHhcub5AIVCwyRCh3T0Q57EAAYASA AEgIMv_D_BwE. Acesso em: 02 out. 2018.
- DRONE INDUSTRY INSIGHTS. **Drone energy sources**: pushing the boundaries of electric flight. Manufacturer, Market, Reseach, Technology, 2017. Disponível em: <https://www.droneii.com/drone-energy-sources>. Acesso em: 06 jun. 2018.
- EMBRAER. **Notícias**. 2019. Disponível em: <https://embraer.com/br/pt/noticias/?slug=1206596-embraerx-atech-e-harris-corporation-colaboram-para-criar-uma-visao-e-trazer-um-novo-paradigma-de-gerenciamento-de-trafego->

aereo-para-a-mobilidade-aerea-urbana. Acesso em: 03 ago. 2019.

GALIP U., A.; PENG, H.; ÇAKMAKCI, M. **Automotive control systems**. Cambridge: Cambridge University Press, 2012. 604 p.

LUGER, G. F. **Artificial intelligence: structures and strategies for complex problem solving**. 6th ed. London: Pearson Education, Inc., 2008. 792 p.

HARTMAN, A.; LORIMER, W.; MAGNETICS, P. I. **Analysis of bldc generators**. Disponível em:

http://performancemagnetics.com/images/BLDC_Generator_Design_Methodology_Performance_Magnetics_revI-2.pdf. Acesso em: 16 jul. 2018.

HERNÁNDEZ-ALVARADO, R. *et al.* Neural network-based self-tuning PID control for underwater vehicles. **Sensors Switzerland**, v. 16, n. 9, p. 1–20, 2016.

HOLDEN, J.; GOEL, N. Fast-forwarding to a future of on-demand urban air transportation. **VertiFlight**, p. 1–98, 2016. Disponível em: <https://www.uber.com/elevate.pdf>. Acesso em: 13 jul. 2018.

WITTEN, IAN H.; EIBE FRANK, M. A. H. **Data mining practical machine learning tools and techniques**. 3rd ed. Morgan Kaufmann Publishers, 2011. 664 p.

IRWIN, G. W.; WARWICK, K.; HUNT, K. J. Neural network applications in control. **IEE Control Engineering**, Stevenage, Institution of Electrical Engineers Neural network, 1995. (series 53).

KHARE, N.; SINGH, P. Modeling and optimization of a hybrid power system for an unmanned surface vehicle. **Journal of Power Sources**, v. 198, p. 368–377, 2012. Disponível em: <http://dx.doi.org/10.1016/j.jpowsour.2011.09.080>. Acesso em: 23 mar. 2018.

LAPEÑA-REY, N. *et al.* A fuel cell powered unmanned aerial vehicle for low altitude surveillance missions. **International Journal of Hydrogen Energy**, v. 42, n. 10, p. 6926–6940, 2017.

MBED. **Mbed**. Disponível em: <https://www.mbed.com/en/>. Acesso em: 22 jan. 2019.

MIGUEZ, G. A. **Otimização do algoritmo de backpropagation pelo uso da função de ativação bi-hiperbólica**, 2012. Disponível em: <https://www.marinha.mil.br/spolm/sites/www.marinha.mil.br/spolm/files/88683.pdf>. Acesso em: 27 jan. 2019

MOSAAD, M. I.; SALEM, F. LFC based adaptive PID controller using ANN and ANFIS techniques. **Journal of Electrical Systems and Information Technology**, Electronics Research Institute (ERI), v. 1, n. 3, p. 212–222, 2014. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S2314717214000403>. Acesso em: 02 nov. 2018.

NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. **IEEE Transactions on Neural Networks**, v. 1, n. 1, p. 4–27, 1990. Disponível em: <http://ieeexplore.ieee.org/document/80202/>. Acesso em: 29 set. 2019.

NASA. **Flies hybrid electric VTOL**. Disponível em:

<https://www.avweb.com/avwebflash/news/NASA-Flies-Hybrid-Electric-VTOL-222636-1.html>. Acesso em: 05 ago. 2018.

NIELSEN, M. A. **Neural networks and deep learning**, 2015. Disponível em: <http://neuralnetworksanddeeplearning.com/about.html>. Acesso em: 24 jan. 2019.

OGATA, K. **Engenharia de controle moderno**. 4. ed. Rio de Janeiro: Prentice-Hall, 2003. 794 p.

OKUMUS, E. *et al.* Development of boron-based hydrogen and fuel cell system for small unmanned aerial vehicle. **International Journal of Hydrogen Energy**, v. 42, n. 4, p. 2691–2697, 2017. Disponível em: <http://dx.doi.org/10.1016/j.ijhydene.2016.09.009>. Acesso em: 24 jan. 2019.

OZDEMIR, U. *et al.* Design of a commercial hybrid VTOL UAV system. **Journal of Intelligent and Robotic Systems: theory and applications**, v. 74, n. 1–2, p. 371–393, 2014.

PATEL, R.; KUMAR, V. Multilayer neuro PID controller based on back propagation algorithm. **Procedia Computer Science**, v. 54, p. 207–214, 2015. Disponível em: <http://dx.doi.org/10.1016/j.procs.2015.06.023>. Acesso em: 20 jan. 2019.

RANGEL, R. K. *et al.* Development of a multipurpose hydro environmental tool using swarms, UAV and USV. In: IEEE AEROSPACE CONFERENCE, 3., 2019, Big Sky, Montana. **Abstracts** [<https://ieeexplore.ieee.org/document/8741624>]. p.1-15, 2019.

Rangel, Rodrigo Kuntz. **Sistema VANT de monitoramento aéreo multiuso com funcionalidades bio-inspiradas**. Orientador: Karl Heinz Kienitz. 2012. 187f. Tese (Doutorado Engenharia Eletrônica e Computação – Área de Informática) - Instituto Tecnológico de Aeronautica - ITA, São José dos Campos, 2012.

RANGEL, R. K.; TERRA, A. C. Development of a surveillance tool using UAV's. **IEEE Aerospace Conference Proceedings**, v. 2018, n. mar., p. 1–11, 2018.

REPORT IDST. **Global UAV Market growth to driven by civil , commercial , military & security sectors**. Disponível em: https://www.reportlinker.com/market-report/Drone/664459/Drone?utm_source=adwords4&utm_medium=cpc&utm_campaign=Aerospace_&_Defense&utm_adgroup=Drone_Market_Reports&gclid=EAIaIQobChMI4bi8o-KF6AIVewWRCh2xNQhIEAAYASAAEgIHXPB_BwE. Acesso em: 20 jul. 2019.

RIVAS-ECHEVERRÍA, F.; RÍOS-BOLÍVAR, A.; CASALES-ECHEVERRÍA, J. Neural network-based auto-tuning for PID controllers. **Neural Network World**, v. 11, n. 3, p. 277–284, 2001.

ROSALES, C.; SORIA, C. M.; ROSSOMANDO, F. G. Identification and adaptive PID control of a hexacopter UAV based on neural networks. **International Journal of Adaptive Control and Signal Processing**, v. 33, n. 1, p. 74–91, 2019.

ROSEMBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533–536, 1986. Disponível em:

<http://www.nature.com/articles/323533a0>. Acesso em: 24 jan. 2019.

RUSSELL; NORVIG . Artificial intelligence: a modern approach. 3. ed. New Jersey: Prentice Hall Editora Upper Saddle River, 2010.

SEBORG, D. E. et al. **Process dynamics and control**. 3. ed. Danvers: John Wiley & Sons Inc., 2011. 464 p.

SECRETARIA DE DESENVOLVIMENTO E COMPETITIVIDADE INDUSTRIAL; **Diálogo Setorial União Europeia-Brasil**: estudo sobre a indústria brasileira e europeia de veículos aéreos não tripulados, p. 92, 2017. Disponível em: www.dialogossetoriais.org. Acesso em: 28 jan. 2019.

SHAHRAKI, F.; FANAEI, M. A. A.; ARJOMANDZADEH, A. R. R. Adaptive system control with PID neural networks. **Chemical Engineering Transactions**, v. 17, n. 1989, p. 1395–1400, 2009. Disponível em: <http://profdoc.um.ac.ir/paper-abstract-1010869.html>. Acesso em: 13 jan. 2019.

SHARMA, R. *et al.* An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload. **ISA Transactions**, v. 62, p. 258–267, 2016. Disponível em: <http://dx.doi.org/10.1016/j.isatra.2016.01.016>. Acesso em: 17 fev. 2019.

J. SIMA. **Introduction to neural networks**: technical report V-755, ICS CAS, Prague, 1998. Disponível em: <https://www.cs.cas.cz/~sima/>. Acesso em: 28 mar. 2019.

SPINNER - PROPDRIIVE. **v2 5050**. Disponível em: https://hobbyking.com/en_us/propdrive-v2-5050-580kv-brushless-outrunner-motor.html. Acesso em: 12 out. 2018.

SRIVIGNESH, N. *et al.* Design of neural based PID controller for nonlinear process. **Procedia Engineering**, v. 38, p. 3283–3291, 2012a. Disponível em: <http://dx.doi.org/10.1016/j.proeng.2012.06.380>. Acesso em: 18 out. 2018.

SRIVIGNESH, N. *et al.* Design of neural based PID controller for nonlinear process. **Procedia Engineering**, v. 38, p. 3283–3291, 2012b. Disponível em: <http://dx.doi.org/10.1016/j.proeng.2012.06.380>. Acesso em: 11 maio 2018.

SU, Y. *et al.* Understanding information diffusion under interactions. **IJCAI International Joint Conference on Artificial Intelligence**, v. 2016-Janua, n. 9, p. 3875–3881, 2016.

SUI, D.; JIAO, Z. Application of neural network in optimization of PID controller. **Metallurgical and Mining Industry**, v. 7, n. 7, p. 2–7, 2015. Disponível em: <https://pdfs.semanticscholar.org/8965/00ed78b132fa79f1a762357ba43d6d0e6f96.pdf>. Acesso em: 05 jun. 2018.

SURENDRAN, S.; KUMAR, V. Neural network based PI controller parameter calculation on a boiler drum level system. **Procedia Technology**, v. 24, p. 1616–1622, 2016. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S2212017316302535>. Acesso em: 11 jun. 2018.

SUZUKI, M.; YAMAMOTO, T.; TSUJI, T. A design of neural-net based PID controllers with evolutionary computation. **IEICE Transactions on Fundamentals of Electronics**,

Communications and Computer Sciences, v. E87-A, n. 10, p. 2761–2768, 2004.

THE DRONE market and industry trends: relatório. Disponível em: <https://www.toptal.com/finance/market-research-analysts/drone-market>. Acesso em: 21 jan. 2019.

WATTS, A. C.; AMBROSIA, V. G.; HINKLEY, E. A. Unmanned aircraft systems in remote sensing and scientific research: classification and considerations of use. **Remote Sensing**, v. 4, n. 6, p. 1671–1692, 2012.

WINSTON, P. H. **Artificial intelligence**. 3.ed. Reading, MA: Addison-Wesley, 1993. 640 p.

NASCIMENTO JR., C. L.; YONEYAMA, T., **Inteligência artificial em controle e automação**. São Paulo: Edgard Blucher Ltda., 2011. e FAPESP 228p.

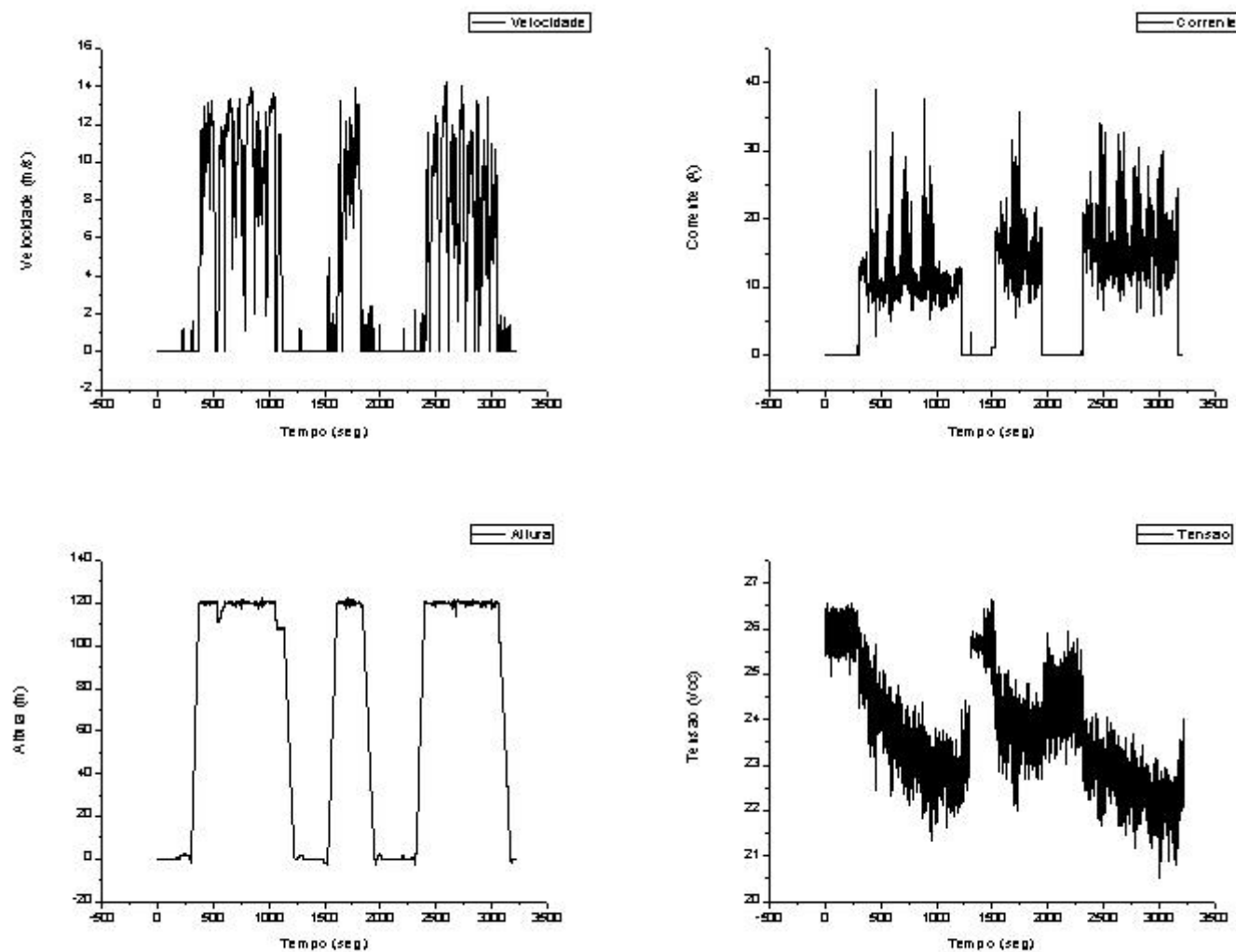
ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. **ASME**, v. 64, p. 759–768, 1942.

ZILOUCHIAN, A.; M. JAMSHIDI. **Intelligent control systems using soft computing methodologies**. Boca Raton, Florida: CRC Press, 2001.

ZRIBI, A.; CHTOUROU, M.; DJEMEL, M. **A new PID neural network controller design for nonlinear processes**, p. 1–10, 2015. Disponível em: <http://arxiv.org/abs/1512.07529>. Acesso em: 09 jul. 2018.

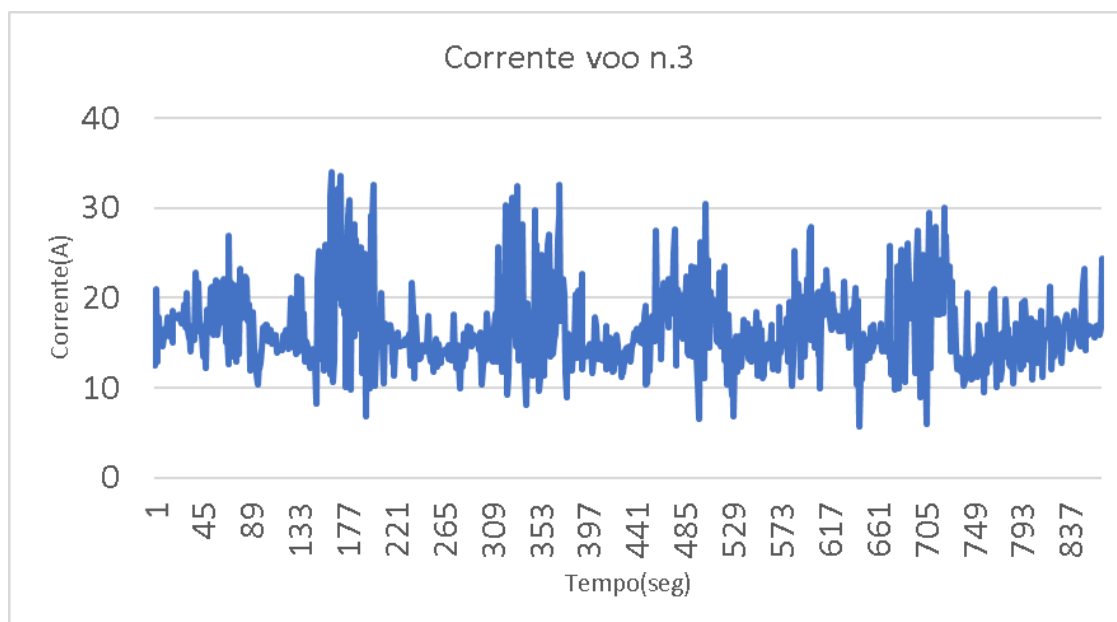
APÊNDICE A – Gráficos obtidos em voo do RPA do IBRV - Vide tópico 3.2.1

Figura 67 – Gráficos dos dados em voo.



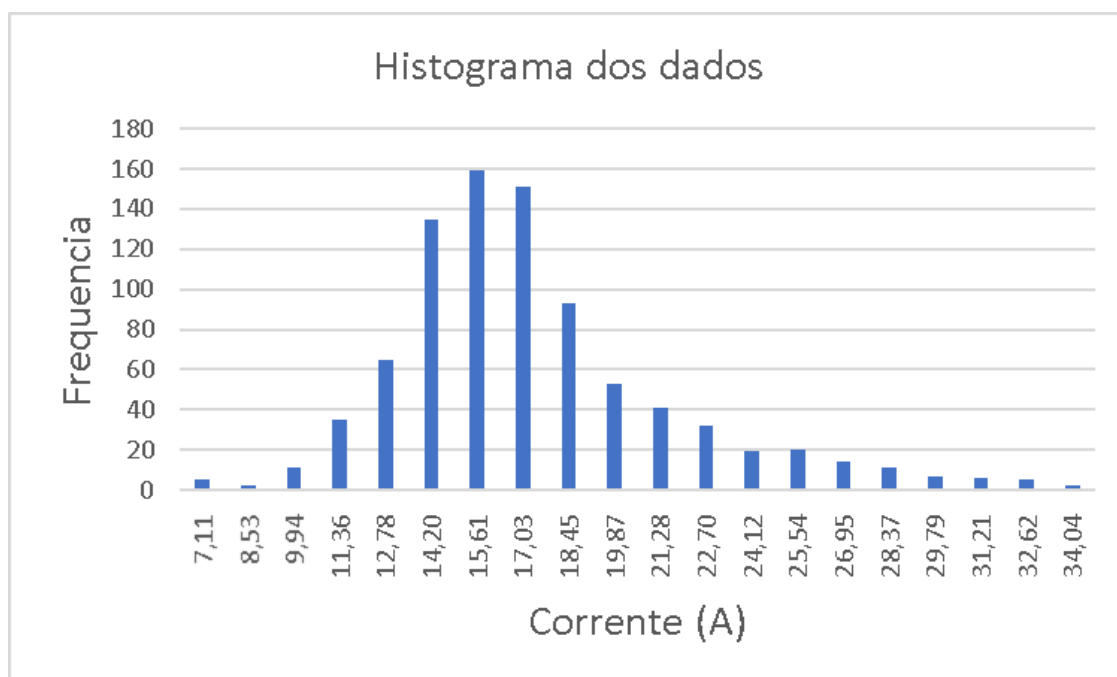
Fonte: Produção do próprio autor.

Figura 68 - Gráficos dos dados em voo.



Fonte: Produção do próprio autor.

Figura 69 – Histograma dos dados da Corrente do voo n.3.



Fonte: Produção do próprio autor.

APÊNDICE B – Tabela com os Dados Coletados - Vide tópico 3.4.2.

RPM	POTENCIA (w)
4287,1	300
4287,1	300
4333,48	300
4379,86	300
4426,24	300
4472,62	300
4519	300
4565,37	300
4611,75	300
4658,13	300
4704,51	300
4750,89	300
4797,27	300
4843,65	300
4843,65	300
4890,03	307.175
4936,4	336,45
4982,78	361,15
5029,16	379.447
5075,54	403.538
5121,92	421,53
5168,3	445,62
5214,68	460.868
5261,05	480.384
5307,43	503.255
5353,81	520.332
5400,19	567.903
5400,19	542.288
5446,57	555.096
5492,95	575.832
5492,95	553.876
5539,33	594.434
5585,71	609.376
5632,08	629.502
5678,46	645.055
5724,84	663.046
5771,22	678.599
5771,22	670,06
5817,6	691.406
5863,98	706.959
5910,36	722.206
5956,73	733.794
6003,11	766.728
6003,11	747.211
6049,49	763.678
6095,87	770.692
6142,25	784.415
6188,63	794.478
6235,01	806.066
6281,39	812,47
6327,76	821.008
6374,14	830.461
6420,52	835.036
6466,9	842.354
6513,28	846.014
6559,66	849.978
6606,04	852.113
6652,41	853.027
6698,79	850.893
6745,17	850.283
6791,55	849.063
6837,93	843.574
6884,31	840.525
6930,69	834.731
6977,06	827.107
7023,44	821.923
7069,82	811,25
7116,2	806.371
7162,58	801.492
7208,96	786.854
7255,34	799.357
7255,34	774.656
7301,72	769.472
7348,09	761.544
7394,47	749.041
7440,85	740.198
7487,23	731.964
7533,61	720.071
7579,99	711.228
7626,37	697,81
7672,74	689.577
7719,12	680.428
7765,5	669,45
7811,88	654.508
7858,26	664.571
7858,26	639.566

APÊNDICE C – Sistema Especialista – Vide tópico 3.4.4.

Temperature range	ground height range	Normalized Current	Generator Voltage (Vdc)	rpm	PWM (ms)
1	1	0	0	5000	1
1	1	0,0333	11	5000	1
1	1	0,0667	11	5300	1,1
1	1	0,1	11	5300	1,1
1	1	0,1333	11	5800	1,2
1	1	0,1667	11	5800	1,2
1	1	0,2	11	6000	1,3
1	1	0,2333	11	6000	1,3
1	1	0,2667	11	6000	1,3
1	1	0,3	11	6500	1,4
1	1	0,3333	11	6500	1,4
1	1	0,3667	11	6500	1,4
1	1	0,4	11	6900	1,5
1	1	0,4333	11	6900	1,5
1	1	0,4667	11	6900	1,5
1	1	0,5	11	7100	1,6
1	1	0,5333	11	7100	1,6
1	1	0,5667	11	7100	1,6
1	1	0,6	11	7100	1,6
1	1	0,6333	11	7300	1,7
1	1	0,6667	11	7300	1,7
1	1	0,7	11	7300	1,7
1	1	0,7333	11	7500	1,8
1	1	0,7667	11	7500	1,8
1	1	0,8	11	7500	1,8
1	1	0,8333	11	8300	1,9
1	1	0,8667	11	8300	1,9
1	1	0,9	11	8300	1,9
1	1	0,9333	11	8300	1,9
1	1	0,9667	11	8300	1,9
1	1	1	11	8300	1,9
1	0	0	0	0	1
1	0	0,0333	0	0	1
1	0	0,0667	0	0	1
1	0	0,1	0	0	1
1	0	0,1333	0	0	1
1	0	0,1667	0	0	1
1	0	0,2	0	0	1
1	0	0,2333	0	0	1
1	0	0,2667	0	0	1
1	0	0,3	0	0	1
1	0	0,3333	0	0	1
1	0	0,3667	0	0	1
1	0	0,4	0	0	1
1	0	0,4333	0	0	1
1	0	0,4667	0	0	1
1	0	0,5	0	0	1
1	0	0,5333	0	0	1
1	0	0,5667	0	0	1
1	0	0,6	0	0	1
1	0	0,6333	0	0	1
1	0	0,6667	0	0	1
1	0	0,7	0	0	1
1	0	0,7333	0	0	1
1	0	0,7667	0	0	1
1	0	0,8	0	0	1
1	0	0,8333	0	0	1
1	0	0,8667	0	0	1
1	0	0,9	0	0	1
1	0	0,9333	0	0	1
1	0	0,9667	0	0	1
1	0	1	0	0	1
0	1	0	0	0	1
0	1	0,0333	0	0	1
0	1	0,0667	0	0	1
0	1	0,1	0	0	1
0	1	0,1333	0	0	1
0	1	0,1667	0	0	1
0	1	0,2	0	0	1
0	1	0,2333	0	0	1
0	1	0,2667	0	0	1
0	1	0,3	0	0	1
0	1	0,3333	0	0	1
0	1	0,3667	0	0	1
0	1	0,4	0	0	1
0	1	0,4333	0	0	1
0	1	0,4667	0	0	1
0	1	0,5	0	0	1
0	1	0,5333	0	0	1
0	1	0,5667	0	0	1
0	1	0,6	0	0	1
0	1	0,6333	0	0	1
0	1	0,6667	0	0	1
0	1	0,7	0	0	1
0	1	0,7333	0	0	1
0	1	0,7667	0	0	1
0	1	0,8	0	0	1
0	1	0,8333	0	0	1
0	1	0,8667	0	0	1
0	1	0,9	0	0	1
0	1	0,9333	0	0	1
0	1	0,9667	0	0	1
0	1	1	0	0	1
1	1	0	0	0	1
0	1	0	0	0	1
1	0	0	0	0	1

APÊNDICE D – Esquema elétrico da Placa Desenvolvida – tópico 3.3.4.

APÊNDICE E – Código do firmware desenvolvido – RNA.

```
#include "mbed.h"
#include "BME280.h"

// Fator a multiplicar a tensão de entrada para obter valor da grandeza
// medida
#define FATOR_UMIDADE 1
#define FATOR_PRESSAO 1
#define FATOR_CORRENTE 1
#define FATOR_TEMPERATURA 1
// Limites entre os quais se utiliza a tabela aprendida
// para gerar referência pwm
#define ALTITUDE_SUPERIOR 3000.0
#define ALTITUDE_INFERIOR 0.0
#define TEMPERATURA_SUPERIOR 45.0
#define TEMPERATURA_INFERIOR -20.0
#define CORRENTE_SUPERIOR 4.48
#define CORRENTE_INFERIOR 2.5
//Referência constante de pwm: pulso com 1ms de largura
#define ROTACAOMIN 0.05
//Referência constante de pwm: pulso com 2ms de largura
#define ROTACAOMAX 0.10
// Comando para desligar motor
#define DESLIGA 0
// Número de pares {corrente, pwm} da tabela aprendida
#define TAMANHO_TABELA 12
// Período do pwm
#define P20MS 0.020
const float tabela[][2] = {
// {corrente, pwm}
{ sistema especialista
};
```

```

// Sensor BME280
#if defined(TARGET_LPC1768)
BME280 sensor(p28, p27);
#else
BME280 sensor(I2C_SDA, I2C_SCL);
#endif

float pressao;
float umidade;
float temperatura;
float pressaoLida;
float umidadeLida;
float temperaturaLida;
float altitude;
// Sensor BME280

PwmOut motor(p21);
AnalogIn corrente(p17);
float correnteLida;

// Função que gera a referência de pwm a partir da corrente de
// entrada, utilizando a tabela aprendida.
float controleDaRotacaoDoMotor (float corrente)
{
    // referência para pwm
    float saida;

    // valor a ser adicionado a limite inferior do intervalo dentro do qual
    // a corrente de entrada se encontra
    // em virtude da interpolação linear
    double delta;

```

```

saida = 0;

if (corrente < CORRENTE_INFERIOR) {
    // Corrente abaixo do mínimo previsto. Limitar saída em 1.
    saida = ROTACAOMIN;
} else if (corrente > CORRENTE_SUPERIOR) {
    // Corrente acima do máximo previsto. Limitar saída em 2.
    saida = ROTACAOMAX;
} else {
    for (int i = 0; i < TAMANHO_TABELA-1; i++) {
        // Corrente dentro da faixa prevista. Utilizar valores tabelados.
        // Para valores intermediários, fazer interpolação linear
        if (corrente >= tabela[i][0] && corrente < tabela[i+1][0]){
            // delta = ((y[i+1]-y[i])/(x[i+1]-x[i]))*(corrente-x[i])
            delta = corrente - tabela[i][0];
            delta = delta * (tabela[i+1][1]- tabela[i][1]);
            delta = delta / (tabela[i+1][0]- tabela[i][0]);
            saida = tabela[i][1] + delta;
            break;
        }
    }
}
return saida;
}

```

```

float obterAltitude (float pressao)
{
    float A = pressao/101325;
    float B = 0.190263;
    float C = pow(A,B);
    C = 1.0 - C;
    C = C * 44330.8;
    return C;
}

```

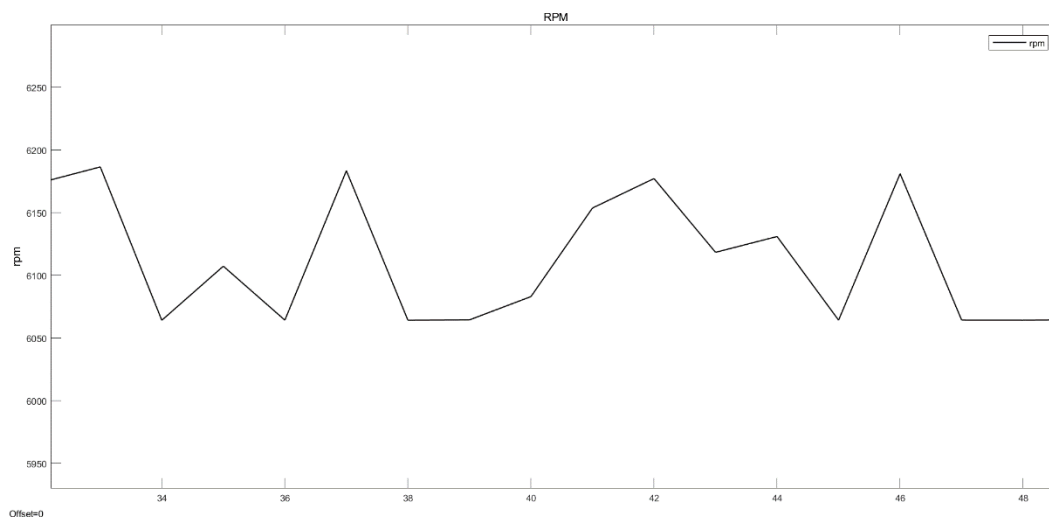
```

int main() {
    // Liga_Sistema()
    // Liga_Motor()
    motor.period_ms(20); // Período PWM configurado com 20 ms
    while(1) {
        pressaoLida    = (sensor.getPressure())*FATOR_PRESSAO;
        correnteLida    = corrente.read()*FATOR_CORRENTE;
        temperaturaLida = (sensor.getTemperature())*FATOR_TEMPERATURA;
        umidadeLida     = (sensor.getHumidity())*FATOR_UMIDADE;
        altitude = obterAltitude(pressaoLida);
        if (
            (correnteLida <= CORRENTE_SUPERIOR) &&
            (temperaturaLida >= TEMPERATURA_INFERIOR && temperaturaLida <=
TEMPERATURA_SUPERIOR ) &&
            (altitude    >=    ALTITUDE_INFERIOR    &&    altitude    <=
ALTITUDE_SUPERIOR)
        )
        {
            motor = controleDaRotacaoDoMotor(correnteLida);    // utiliza tabela de
valores aprendidos para estabelecer largura do pwm em segundos
        }
        else
        {
            motor = ROTACAOMIN; // estabelecer largura do pwm com valor de
ROTACAO1 em segundos
        }
        wait(0.01);
    }
}

```

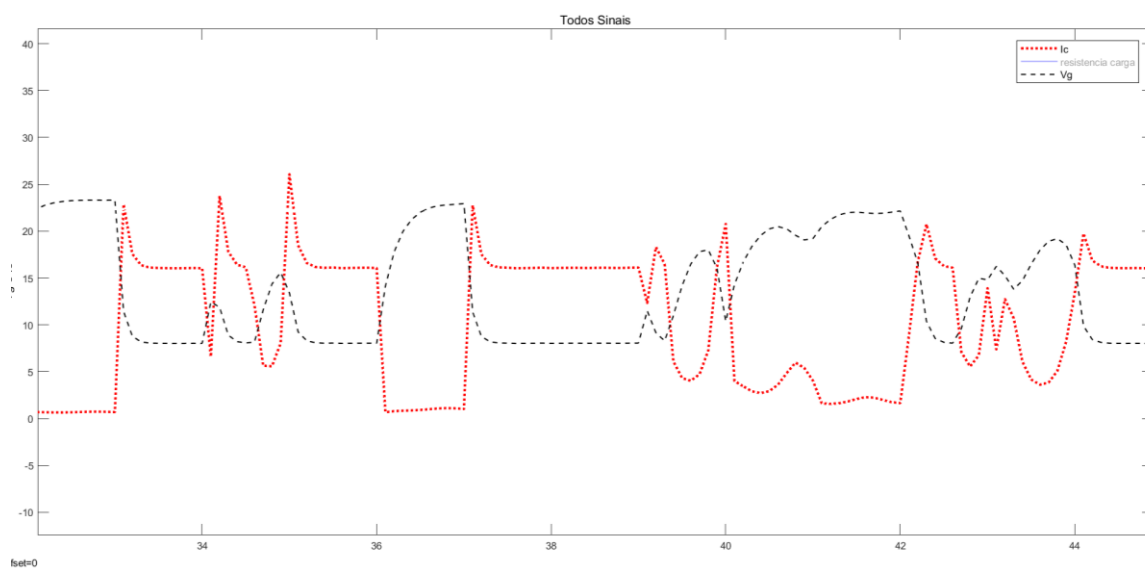
APÊNDICE F – Dados Coletados – Vide tópico 3.4.2.

Figura 72– Dados coletados da rotação do motor.



Fonte: Autoria Própria.

Figura 73 – Dados da Corrente na carga e da Tensão no Gerador variando-se a Carga.



Fonte: Autoria Própria.

APÊNDICE G – Relatório fornecido com as características da estimativa escolhida –

Vide **tópico 3.4.6.**

```
>> step(tf10)
```

```
tf10 =
```

```
tf4 =
```

From input "u1" to output "y1":

$0.1239 (+/- 0.001168) s^2 + 3.868 (+/- 0.9744) s + 0.1988 (+/- 0.1257)$

$s^2 + 31.41 (+/- 8.159) s + 59.62 (+/- 37.56)$

Name: tf4

Continuous-time identified transfer function.

Parameterization:

Number of poles: 2 Number of zeros: 2

Number of free coefficients: 5

Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Termination condition: Maximum number of iterations reached.

Number of iterations: 20, Number of function evaluations: 142

Estimated using TFEST on time domain data "saida1".

Fit to estimation data: 62.54% (stability enforced)

FPE: 12.47, MSE: 11.2

More information in model's "Report" property.

```
>>
```

APÊNDICE H – Código do controle tipo PID linguagem C++ – Vide tópico 3.4.6.

```

/* PI Ctl Tese_Joacy

Permite sintonizar via porta serial.

Movimenta o servo via pino 21.

v 1.5

MAIO-2019

*/

#include "mbed.h"

// servo largura de pulso em microsegundos

#define SERVO_MIN 1000

#define SERVO_MID 1500

#define SERVO_MAX 2000

// cria objetos periféricos

PwmOut servo(p21);

LocalFileSystem local("local");

void initServo();

void moveServo(float angle);

float getVg();

float updatePI(float current,float target,float dt);

// variaveis globais

float kp = 0.7;

float ki = 1.12; // PI co-eficientes

float previousError = 0.0; // variavel armazena erro

float P,I; // proporcional, integral erros;

float Vg = 10.0; // objetivo 10vcc

int main()

{

    float dt = 0.2; // update PI a cada 0.1 s

    float servoAngle = 0.0; // angulo = 0 no servo

    float output; // saida do PI

```

```

initServo();
I = 0.0; // integral em zero
while(1) {
    Vg = servoAngle;

    servoAngle += output; // update servo angulo
    if (servoAngle > 90.0)
        servoAngle = 90.0;
    if (servoAngle < -90.0)
        servoAngle = -90.0;
    if (Vg > targetVg + 1.0)
    else if (Vg < targetVg -1.0)
        moveServo(servoAngle);
    wait(dt);
}

float updatePID (float current,float target,float dt)
{
    P = target - current; // erro proporcional
    I += P*dt; // erro integral
    previousError = P; // salva erro atual
    if (I > 10.0)
        I = 10.0;
    if (I < -10.0)
        I = -10.0;
    float output = P*kp + I*ki; // calcula PI output

    return output;
}

```

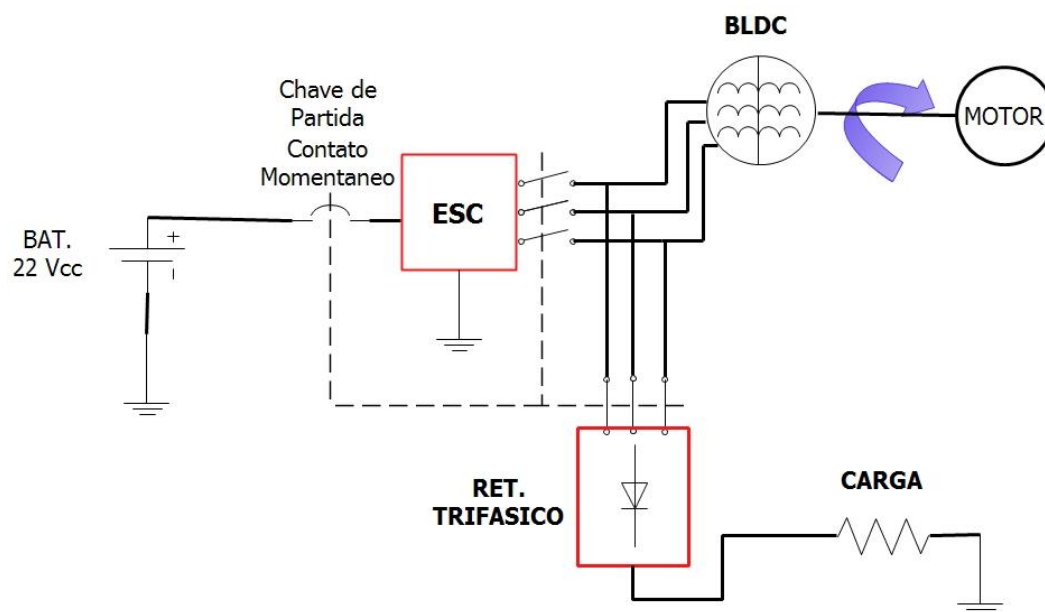
```
// function to setup servo

void initServo()
{
    servo.period(1.0/50.0); // servo PWM 50 Hz
}

void moveServo(float angle)
{
    angle+=90.0; // shift angle to 0 to 180
    float microsPerDegree = (SERVO_MAX - SERVO_MIN) / 180.0;
    int pulsewidth = SERVO_MIN + int (angle*microsPerDegree
    servo.pulsewidth_us(pulsewidth);
}
```

APÊNDICE I – Esquema BLDC atuando como Motor de Partida.

Figura 74 - Esquema BLDC atuando como Motor de Partida



ANEXO A – Folha de dados do motor Titan ZG 62SL.

Titan ZG 62SL

Specification:

Capacity:	62 ccm
Bore:	47,5 mm
Stroke:	35 mm

Power:

With Tuned Pipe:	6,6 hp
Open Exhaust:	5,0 hp
Standard Silencer:	4,3 hp

Weight:	2040 g
without silencer	

Fuel:

Lead free Petrol with BEL RAY H1R
at a mixing ratio of 50:1.

Item #6500
[▶ Webshop](#)
**5 Years Guarantee!**

- **New Silencer**
- **40 mm Prop Hub is now Standard**

The Titan ZG 62 was first produced in 1985 being designed so as to fit into a chain saw and at the same time being able to be used for model purposes without modification. The sales volume with the ZG 38 for models made this possible.

In 1990 Komatsu reworked the cylinder specially for model aircraft purposes this S cylinder had the transfer passages altered as well as the combustion chamber. This resulted in an increase of 200 rpm in the 6,500 range. Fit a tuned pipe and the rpm is considerably increased, giving a power of 6,6 hp at 8,400 rpm.

The newest Titan, the 62SL, has a diecast aluminium crankcase weighing 50 grams less than the previous sand cast version.

Fonte: Zenoah-engines (2019)

ANEXO B – Dados do Motor Vulcan VR430H.

Motor	260H	330H	430H
Potência	0,7 kw	1.0 kw	1,25kw
Cilindrada	26cc	33cc	43cc
Rotação máxima	10000 rpm	10000 rpm	10000 rpm
Rotação na lenta	3000 rpm	3000 rpm	3000 rpm
Peso Líquido	5,8 Kg	7,5 Kg	8,35Kg
Ferramentas de corte	Lâminas 2 pontas furo 1" polegada e fio de nylon		
Ignição	Eletrônica - TCI		
Arranque	Manual com mola retrátil		
Carburador	Membranas		
Combustível - Gasolina x óleo	Gas x óleo 30:1	Gas x óleo 25:1	Gas x óleo 25:1
Tanque do comb.	450 ml	1000 ml	1,100 ml
Autonomia	aprox. 1h	aprox. 1h	aprox. 1h
Guidão	Bi Partido	Bi Partido	Bi Partido
*Nível de Ruído	116 dB	116 dB	116 dB
**Vibração	10,004 m/s ²	10,004 m/s ²	10,004 m/s ²
Registro no CREA/PR	62314		
As especificações estão sujeitas a alterações sem aviso prévio.			
*metodologia utilizada para calibração: K=2,5 dB(A) ISO22868			
**metodologia utilizada para calibração: K=1,5 M/S ² ISO22867			

Fonte: Vulcanequipamentos (2018).

ANEXO C – Datasheet do Microcontrolador Mbed NXP LPC1768.

(continua)



mbed NXP LPC1768
prototyping board

Rapid prototyping for the LPC1768 MCU

This board, which works with the groundbreaking mbed tool suite, lets you create a functioning prototype faster than ever. The tightly coupled combination of hardware and software makes it easy to explore designs quickly, so you can be more adventurous, more inventive, and more productive.

Features

- ▶ Convenient form-factor: 40-pin DIP, 0.1-inch pitch
- ▶ Drag-and-drop programming, with the board represented as a USB drive
- ▶ Best-in-class Cortex-M3 hardware
 - 100 MHz ARM with 64 KB of SRAM, 512 KB of Flash
 - Ethernet, USB OTG
 - SPI, I²C, UART, CAN
 - GPIO, PWM, ADC, DAC
- ▶ Easy-to-use online tools
 - Web-based C/C++ programming environment
 - Uses the ARM RealView compile engine
 - API-driven development using libraries with intuitive interfaces
 - Comprehensive help and online community

Benefits

- ▶ Get started right away, with nothing to install
- ▶ Get working fast, using high-level APIs
- ▶ Explore, test, and demonstrate ideas more effectively
- ▶ Write clean, compact code that's easy to modify
- ▶ Log in from anywhere, on Windows, Mac or Linux

The mbed NXP LPC1768 board lets you create prototypes without having to work with low-level microcontroller details, so you can experiment and iterate faster than ever.

Designers compose and compile embedded software using a browser-based IDE, then download it quickly and easily, using a simple drag-and-drop function, to the board's NXP Cortex-M3 microcontroller LPC1768.

Engineers new to embedded applications can use the board to prototype real products incorporating microcontrollers, while experienced engineers can use it to be more productive in early stages of development. The mbed tools are designed to let you try out new ideas quickly, in much the same way that an architect uses a pencil and paper to sketch out concepts before turning to an advanced CAD program to implement a design.



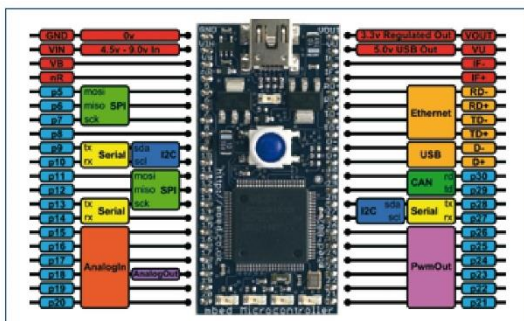
ANEXO C – Datasheet do Microcontrolador Mbed NXP LPC1768.

(continuação)

Elegant simplicity

The mbed tool has been designed for the best trade-off between versatility and immediate connectivity. The LPC1768, housed in an LQFP package, is mounted on the mbed board, which uses a 40-pin DIP with a 0.1-inch pitch. The convenient form factor works seamlessly with solderless breadboards, stripboards, and PCBs.

There is no software to install – everything, even the compiler, is online. The compiler and libraries are completely modular, so they're easy to use, yet powerful enough to take on complex, real-world applications.



Pinout diagram of mbed NXP LPC1768 board

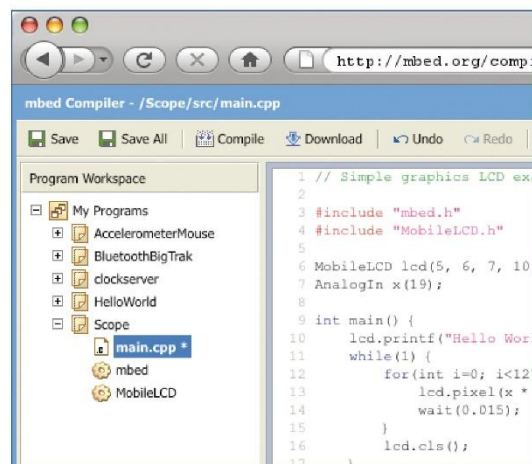
Hassle-free startup

Getting started is as simple as using a USB Flash drive. Simply connect the mbed NXP LPC1768 board to a Windows, Mac or Linux computer and it will appear as a USB drive. Follow the link on the board to connect to the mbed website, where you can sign up and begin designing. There are no drivers to install or setup programs to run. It's so easy, in fact, that you can have a "Hello World!" program running in as little as five minutes.

Online compiler

The mbed Compiler lets you write programs in C++ and then compile and download them to run on the mbed NXP LPC1768 microcontroller. There's no need to run an install or setup program, since the compiler runs online. Supported browsers include Internet Explorer, Firefox, Safari, or Chrome running on a Windows, Mac, or Linux PC. You can log in from anywhere and simply pick up where you left off. And, since you're working with a web-based tool, you can be confident that it's already configured and will stay up-to-date.

The compiler uses the ARM RealView compile engine, so it produces clean, efficient code that can be used free-of-charge, even in production. Existing ARM application code and middleware can be ported to the LPC1768 microcontroller, and the mbed tools can be used alongside other professional production-level tools, such as Keil MDK.



The mbed Compiler

Peripheral libraries

The mbed Library provides an API-driven approach to coding that eliminates much of the low-level work normally associated with MCU code development. You develop code using meaningful peripheral abstractions and API calls that are intuitive and already tested. That frees you up to experiment, without worrying about the implementation of the MCU core or its peripherals. You can work faster and be more creative, and can concentrate on exploring and testing the options for your design.

Rather than simply providing examples, mbed focuses on reusable library functionality, with clear interfaces and solid implementations. The core mbed Library supports the main LPC1768 peripherals, and the libraries already contributed by the mbed design community include USB, TCP/IP, and HTTP support. It's also possible to add third-party and open-source stacks.

The libraries comply with the ARM EABI and are built on the Cortex Microcontroller Software Interface Standard (CMSIS),

ANEXO C – Datasheet do Microcontrolador Mbed NXP LPC1768.

(conclusão)

Programs

AccelerometerMouse

BluetoothBigTrak

main.cpp

Motor.cpp

Motor.h

mbed

AnalogIn

AnalogOut

BusIn

BusOut

Debug

DigitalIn

DigitalOut

I2C

LocalFileSystem

PwmOut

SPI

SPI3

Serial

Ticker

SPI

class SPI : public Base

A SPI Master, used for communicating with SPI slave devices. The default format is set to 8-bits, mode 0, and a clock of 100 kHz. Most SPI devices will also require Chip Select and Resistor.

Example

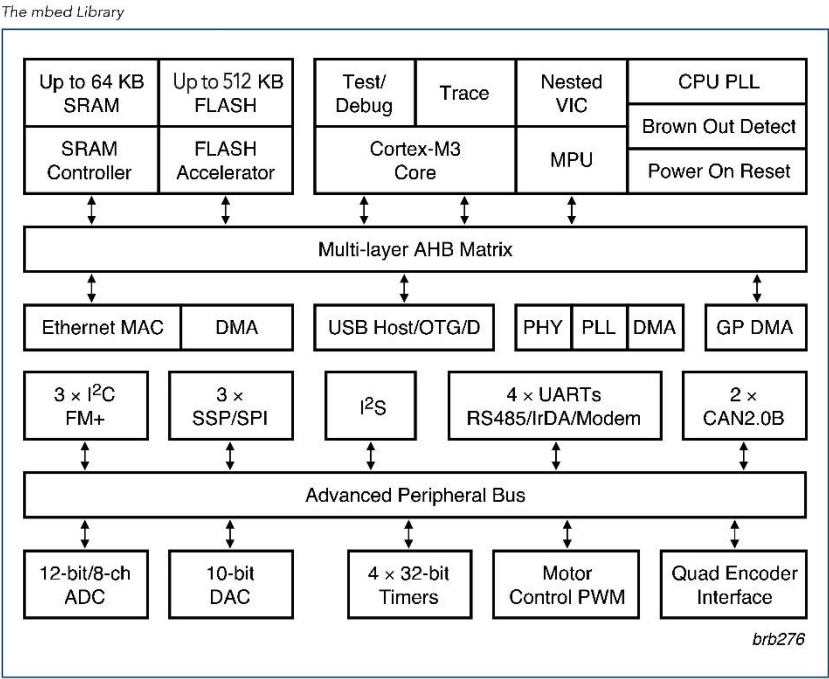
// Send a byte to a SPI slave, and read the response.
#include "mbed.h"
SPI device(5, 6, 7); // mosi, miso, sclk

int main() {
 int response = device.write(0xFF);
}

making it possible to migrate to other toolchains or implement custom code for peripheral interfaces.

LPC176x microcontrollers

The NXP microcontroller family LPC176x is a series of cost-effective, low-power Cortex-M3 devices that operate at up to 100 MHz. They feature best-in-class peripheral support, including Ethernet, USB 2.0 host/OTG/device, and CAN 2.0B. There are 512 KB of Flash memory and 64 KB of SRAM. The architecture uses a multi-layer AHB bus that allows high-bandwidth peripherals such as Ethernet and USB to run simultaneously, without impacting performance. The family is pin-compatible with NXP's 100-pin LPC236x series of ARM7-based microcontrollers.



LPC1768 block diagram

Fonte: MBED (2019).

ANEXO D – Tipos de Algoritmos de Aprendizado de Maquinas.



Fonte: Machinelearningmastery (2019).