

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA  
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Lucca di Giacomo Neves Lemos

**Projeto do conversor DAB aplicado a um carregador de baterias utilizando algoritmos  
genéticos**

Florianópolis

2020

Lucca di Giacomo Neves Lemos

**Projeto do conversor DAB aplicado a um carregador de baterias utilizando algoritmos genéticos**

Trabalho Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. André Luís Kirsten.

Florianópolis

2020

## Ficha de identificação da obra

Lemos, Lucca di Giacomo Neves  
Projeto do conversor DAB aplicado a um carregador de  
baterias utilizando algoritmos genéticos / Lucca di  
Giacomo Neves Lemos ; orientador, André Luis Kirsten, 2020.  
49 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro Tecnológico,  
Graduação em Engenharia Elétrica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Elétrica. 2. Algoritmos genéticos. 3.  
Conversor DAB. I. Kirsten, André Luis. II. Universidade  
Federal de Santa Catarina. Graduação em Engenharia  
Elétrica. III. Título.

Lucca di Giacomo Neves Lemos

**Projeto do conversor DAB aplicado a um carregador de baterias utilizando algoritmos genéticos**

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 18 de dezembro de 2020.



Documento assinado digitalmente  
Jean Vianei Leite  
Data: 21/12/2020 11:32:01-0300  
CPF: 003.474.909-80

---

**Prof. Jean Vianei Leite, Dr.**  
Coordenador do Curso de Graduação em Engenharia Elétrica

**Banca Examinadora:**



Documento assinado digitalmente  
André Luís Kirsten  
Data: 18/12/2020 16:12:04-0300  
CPF: 013.373.690-40

---

**Prof. André Luís Kirsten, Dr.**  
Orientador  
Universidade Federal de Santa Catarina



Documento assinado digitalmente  
Lucas Mondardo Cunico  
Data: 19/12/2020 07:22:14-0300  
CPF: 008.659.309-96

---

**Prof. Lucas Mondardo Cunico, Me.**  
Instituto Federal de Santa Catarina



Documento assinado digitalmente  
Mateus Nava Mezaroba  
Data: 18/12/2020 16:39:45-0300  
CPF: 030.100.130-86

---

**Eng. Eletric. Mateus Nava Mezaroba**

Este trabalho é dedicado à minha família e amigos, que estiveram ao meu lado durante todo esse processo.

## **AGRADECIMENTOS**

Aos meus pais, que me incentivaram e acreditaram em mim até mesmo nos momentos mais difíceis da universidade e à minha irmã que sempre me deu suporte por todos esses anos.

Aos meus amigos, que além de serem boas companhias nos momentos de descontração, sempre foram bons companheiros na hora de estudar e focar na faculdade.

Ao professor André, que aceitou me orientar nesse projeto ao fim de um semestre conturbado e a todo momento me deu o suporte necessário para sua execução.

"The first step is to establish that something is possible; then probability will occur."  
(MUSK, 2018)

## RESUMO

Esse trabalho tem como objetivo apresentar a avaliação da aplicação de metodologias de inteligência artificial em projetos de engenharia. Foi escolhido o uso de algoritmos genéticos, que permitem a solução eficiente de problemas específicos. A motivação do projeto foi que a determinação de componentes e especificações em projetos demanda demasiado tempo, a melhora no aproveitamento de tempo durante esse trabalho pode ter grande valor para pesquisa e desenvolvimento. Nesse contexto, foi apresentado um método de projeto utilizando algoritmos genéticos no desenvolvimento de um conversor DAB aplicado a um carregador de baterias. Dito isso, foi feita a comparação do fator de potência médio durante o processo de carregamento entre soluções apresentadas pelo algoritmo e soluções que foram apresentadas pelo método clássico de desenvolvimento. O algoritmo genético tendo apresentado um aumento no fator de potência médio maior que 1% em relação ao método clássico de projeto. Isso mostrou que há vantagens na utilização de ferramentas de inteligência artificial no desenvolvimento de projetos de engenharia.

**Palavras-chave:** Algoritmo genético. Conversor DAB. Transformador de estado sólido.

## ABSTRACT

This work has the goal of evaluating the application of artificial intelligence methodologies in engineering projects. Genetic algorithm was the chosen methodology, which allows for an efficient solution for specific problems. The project's motivation was the fact that determining components and project specification demands too much time, the increase in the work's efficiency can bring much value for research and development. In this context, a project method will be presented utilizing genetic algorithms in the development of a DAB converter applied into a battery charger. With that said, the average power factors for the charging processes between the solution presented by the genetic algorithm and classic method of development were compared. The genetic algorithm showed positive results, presenting by the end over 1% increase in the average power factor against the one presented by the classic method. This shows there are advantages in using artificial intelligence tools in engineering projects.

**Keywords:** Genetic algorithm. DAB converter. Solid-state transformer.

## LISTA DE FIGURAS

Figura 1 – Sistema de potência de um notebook.....	17
Figura 2 – Esquemático do conversor DAB.....	19
Figura 3 - Formas de onda dos sinais de comando, tensão resultante da ponte do primário e do secundário, e corrente de entrada do conversor DAB.....	21
Figura 4 - Fluxograma de funcionamento de um algoritmo genético .....	24
Figura 5 – Curva de carregamento da bateria NCR 18650PF.....	26
Figura 6 - Curva de carregamento do <i>pack</i> de baterias .....	27
Figura 7 - Curva de carregamento com $\phi_N = 30^\circ$ .....	31
Figura 8 - Curva de carregamento com $\phi_N = 45^\circ$ .....	31
Figura 9- Curva de carregamento com $\phi_N = 60^\circ$ .....	32
Figura 10 - Gráfico dos valores de $\phi_N$ em cada geração para passo de mutação de 40% .....	35
Figura 11 - Gráfico dos valores de $N_s$ em cada geração para passo de mutação de 40% .....	35
Figura 12- Curvas de carregamento da bateria.....	36

## LISTA DE TABELAS

Tabela 1 - Especificações do pack de baterias .....	25
Tabela 2 - Construção do <i>pack</i> de baterias.....	26
Tabela 3 - Resultados para projetos com método clássico.....	30

## LISTA DE QUADROS

Quadro 1 - Gerações com 40% de passo de mutação.....	33
--	----

## LISTA DE ABREVIATURAS E SIGLAS

DAB *Dual Active Bridge*

SST *Solid State Transformer*

CA Corrente Alternada

CC Corrente Contínua

FP Fator de Potência

## SUMÁRIO

<b>1</b>	<b>Introdução .....</b>	<b>15</b>
<b>2</b>	<b>Conceitos iniciais.....</b>	<b>16</b>
2.1	Redes elétricas inteligentes.....	16
2.2	Baterias residenciais .....	17
2.3	Conversor DAB .....	18
<b>2.3.1</b>	<b>Funcionamento do conversor DAB.....</b>	<b>18</b>
<b>2.3.2</b>	<b>Projeto do conversor .....</b>	<b>21</b>
<b>2.3.3</b>	<b>Avaliação da performance do conversor .....</b>	<b>22</b>
2.4	Algoritmo genético .....	22
<b>2.4.1</b>	<b>Conceitos.....</b>	<b>23</b>
<b>2.4.2</b>	<b>O algoritmo .....</b>	<b>24</b>
2.5	Conclusão .....	24
<b>3</b>	<b>Desenvolvimento .....</b>	<b>25</b>
3.1	Pré determinações de projeto.....	25
3.2	Curvas de carregamento de bateria.....	25
3.3	Algoritmo genético para essa aplicação .....	27
3.4	O código .....	28
3.5	Conclusão .....	29
<b>4</b>	<b>Resultados.....</b>	<b>30</b>
4.1	Método clássico .....	30
4.2	Método com algoritmos genéticos.....	32
4.3	Conclusão .....	36
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>38</b>
	<b>REFERÊNCIAS.....</b>	<b>39</b>
	<b>ANEXO A – Datasheet da bateria NCR18650PF .....</b>	<b>40</b>
	<b>ANEXO B – Datasheet do <i>Powerwall</i>.....</b>	<b>41</b>
	<b>APÊNDICE A – Código do projeto .....</b>	<b>42</b>

## 1 INTRODUÇÃO

O estudo da área de inteligência artificial ganhou muita atração nos últimos anos (FUMO, 2017), com as grandes empresas de tecnologia aplicando metodologias de *machine learning* em seus algoritmos e a popularização do tema, um número maior de pesquisadores busca diferentes aplicações dessas metodologias. Entretanto, o estudo de metodologias que permitem um computador solucionar problemas existe desde a década de 1940 com Alan Turing, decifrando os códigos alemães durante a guerra. O termo foi cunhado por John McCarthy em 1956 (COPPIN, 2004) e na atualidade tem como definição “utilizar métodos baseados no comportamento inteligente de humanos e outros animais para solucionar problemas complexos”.

Nesse escopo, no campo da engenharia há diversas metodologias que dependem da experiência do engenheiro e de processos empíricos de teste, fazendo com que aumente o custo e tempo de desenvolvimento. Como por exemplo o projeto de conversores, o engenheiro deve tomar decisões baseadas em seus conhecimentos prévios e testar manualmente cada iteração do projeto, limitando o número de possibilidades que pode testar. Essa metodologia faz com que o projetista tenha um grande *trade-off* entre a eficiência máxima que o projeto pode alcançar e o número de iterações que podem ser testadas.

O propósito da presente pesquisa é fazer um estudo de caso com inteligência artificial, aplicando uma metodologia alternativa de projeto para um problema de engenharia. Será apresentado um projeto e seus objetivos e a partir disso serão comparadas as soluções apresentadas pelo modelo que utiliza ferramentas de inteligência artificial àquelas encontradas pelo método empírico clássico (KIRSTEN, 2014).

Desta maneira, o estudo tem por finalidade apresentar o projeto de um conversor de energia CC-CC utilizando algoritmos genéticos. No capítulo 2 será apresentado o problema, suas especificações e conceitos que serão utilizados ao longo do desenvolvimento deste projeto. Em seguida, no capítulo 3 será demonstrado o processo de execução do trabalho e as metodologias utilizadas a fim de alcançar o resultado esperado. Ao final, o capítulo 4 irá apresentar os resultados obtidos e se a utilização da metodologia discutida é promissora.

## 2 CONCEITOS INICIAIS

Neste capítulo serão apresentados os conceitos de redes elétricas inteligentes, baterias residenciais, algoritmos genéticos e conversores de corrente contínua para corrente contínua (CC-CC), especificamente o conversor *dual active bridge* (DAB), que permite a transferência bidirecional de energia.

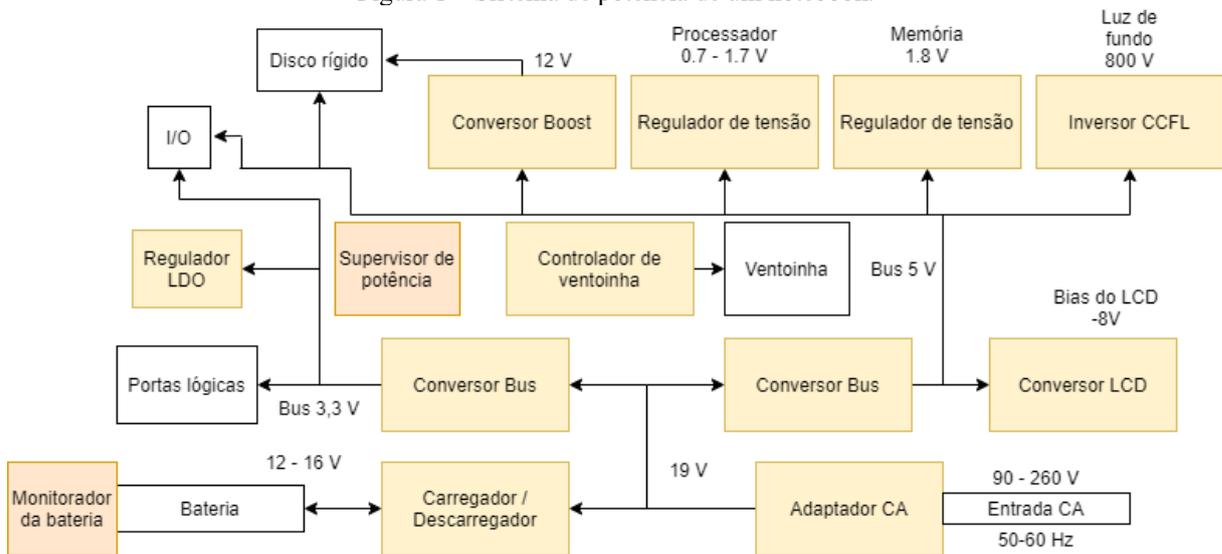
### 2.1 REDES ELÉTRICAS INTELIGENTES

De acordo com Forte, *Smart Grids* são sistemas inteligentes para a distribuição de energia elétrica, utilizando aparelhos de comunicação bidirecional, medidores automáticos e sistemas de computadores especializados (FORTE, 2010). Além disso, ainda de acordo com o autor, esses sistemas irão facilitar a adoção de redes de geração distribuída, incluindo fontes renováveis e veículos elétricos.

Abordando essa temática, para Boroyevich, uma rede inteligente deveria reduzir o número de conversores de energia, “entregando” essa energia pronta para o consumo. Demonstrando assim, que a maior parte dos utensílios domésticos utilizam corrente contínua e apresentam dentro deles conversores CA-CC, além de defender que uma distribuição em corrente contínua seria mais eficiente (BOROYEVICH, 2010).

Nesse escopo, por mais que se tenha distribuição de energia em CC, cada dispositivo apresenta um nível de tensão próprio e ainda se faz necessária a utilização de um conversor. Boroyevich faz um comparativo com a alimentação da placa de um *notebook*, visto na Figura 1, o qual possui apenas um conversor CA-CC e diversos conversores CC-CC para os diferentes níveis de tensão no sistema, a fim de alimentar os componentes como o *disk drive*, periféricos e ventoinhas, além da bateria que será abordada posteriormente. Com base nisso, o sistema se torna mais eficiente e controlável.

Figura 1 – Sistema de potência de um notebook.



Fonte: Boroyevich (2010), adaptada pelo autor via *software* draw.io.

## 2.2 BATERIAS RESIDENCIAIS

Nesse escopo, ao utilizar geradores solares, a produção de energia é maior das 12h às 15h (LAZAR, 2014). Contudo, nesses horários, o consumo residencial é normalmente menor, fazendo assim com que grande parte da produção não seja diretamente utilizada pela unidade produtora, mas sim que seja entregue à rede, gerando um crédito a ser consumido em outro momento (HODGE, 2020). Entretanto, devido à tarifa branca da ANEEL, os créditos são equivalentes ao valor da energia no momento de geração (ANEEL, 2015), ou seja, a residência geradora poderá utilizar uma quantidade menor de kWh do que gerou no momento de maior consumo.

Uma solução para o maior aproveitamento da energia produzida, é o armazenamento por meio do uso de baterias, permitindo o uso dessa energia nos momentos de pico de consumo.

Com o desenvolvimento e redução do custo das células de lítio, a implementação dessas baterias se tornou mais acessível, como no modelo *Powerwall 2* da Tesla, que está descrito no ANEXO B e permite a armazenagem de até 13,5 kWh de energia.

## 2.3 CONVERTOR DAB

Em aplicações como micro geradores e baterias residenciais, é necessário que a energia armazenada em CC seja convertida em corrente alternada (CA), para isso são usados inversores. Tendo a energia em CA é possível abastecer o consumo da residência ou exportar a energia para a rede.

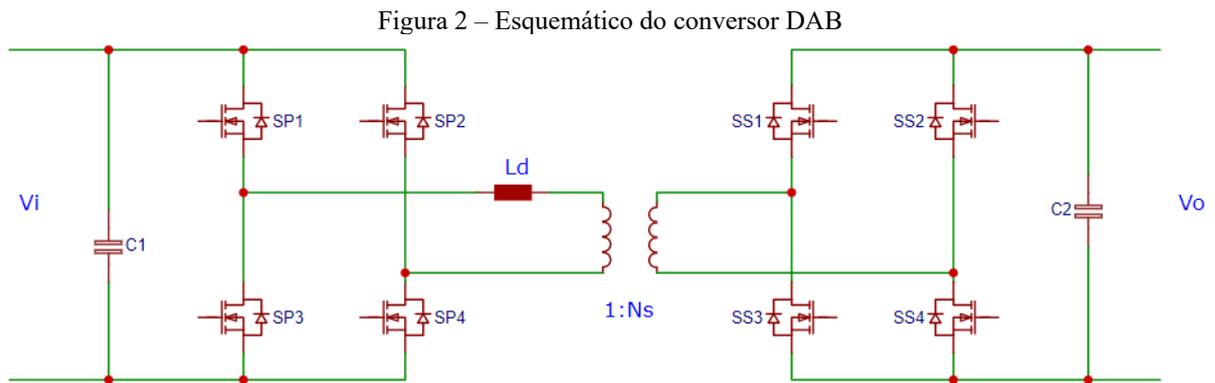
Inversores são importantes no processo de conversão de energia, mas, para que se tenha a melhor eficiência possível, é necessário que o nível de tensão CC já esteja próximo do nível de saída do inversor. Para elevar ou abaixar a tensão das baterias para o nível desejado é necessária a utilização de um conversor CC-CC. Há diversas topologias de conversores CC-CC, porém, por ser uma topologia bidirecional, isolamento galvânica, operação abaixadora e elevadora além de alta eficiência (KIRSTEN, 2014), a solução escolhida para este estudo é a do conversor DAB (*Dual Active Bridge*).

### 2.3.1 Funcionamento do conversor DAB

O conversor DAB consiste em conversores *full-bridge*, nos lados primário e secundário, conectados entre si por um transformador de média ou alta frequência (STIENEKER, 2015).

Essa topologia permite que a energia possa ser transferida em ambas as direções, ou seja, do primário para o secundário e do secundário para o primário. Essa característica é importante ao fazer o uso de uma bateria, pois o mesmo conversor pode ser usado tanto para carregar e utilizar a energia armazenada.

A Figura 2 abaixo apresenta o esquemático de um conversor DAB.



Fonte: Autor (2020) via easyeda.

Na modulação *phase-shift* se utilizam dois sinais de comando, um para os interruptores do primário e outro para os interruptores do secundário, sendo a variável de controle a defasagem angular entre os dois comandos, também chamado de ângulo de carga ( $\phi$ ), consequentemente essa também é a diferença em graus das tensões no primário e secundário das pontes completas de interruptores. A Figura 3 apresenta as formas de onda do acionamento das chaves e como as tensões no primário e secundário respondem a isso, assim como as correntes do sistema durante um período de operação em regime permanente. A transferência de energia entre os lados do conversor se deve a esse deslocamento de fase da comutação entre os lados. Assim, esse ângulo de carga causa uma queda de tensão no indutor auxiliar  $L_d$ , gerando corrente no sistema (KIRSTEN, 2014).

A transferência de potência, em um sistema sem perdas, em função de  $\phi$  é dado por (KIRSTEN, 2014):

$$P_o(\phi) = \frac{V_i^2 d \phi}{\omega_s L_d (\phi_N)} \left( \frac{\pi - |\phi|}{\pi} \right) \quad (1)$$

Sendo,

$P_o$  – Potência de saída do conversor;

$V_i$  – Tensão de entrada do conversor;

$V_o$  – Tensão de saída do conversor;

$N_s$  – Relação de espiras do transformador;

$d$  – Relação de transformação;

$\phi$  – Ângulo de carga;

$L_d$  – Indutor auxiliar;

$\omega$  – Frequência angular em radianos;

$f_s$  – Frequência de comutação

$\omega_s$  – Frequência angular de comutação

$\phi_N$  – Ângulo de carga nominal

Em regime permanente, o sistema se comporta diferentemente em 4 etapas como apresentado abaixo. É válido ressaltar que está sendo considerada uma situação de transferência direta de energia, do primário para o secundário.

Etapa 1:  $0 \leq \omega t < \phi$

$$i_p(\omega t) = (-I_x) + \left( V_i + \frac{V_o}{N_s} \right) \frac{\omega t}{\omega_s L_d} \quad (2)$$

Etapa 2:  $\phi \leq \omega t < \pi$

$$i_p(\omega t) = (I_y) + \left( V_i - \frac{V_o}{N_s} \right) \frac{\omega t - \phi}{\omega_s L_d} \quad (3)$$

Etapa 3:  $\pi \leq \omega t < \pi + \phi$

$$i_p(\omega t) = (I_x) - \left( V_i + \frac{V_o}{N_s} \right) \frac{\omega t - \pi}{\omega_s L_d} \quad (4)$$

Etapa 4:  $\pi + \phi \leq \omega t < 2\pi$

$$i_p(\omega t) = (-I_y) + \left( -V_i + \frac{V_o}{N_s} \right) \frac{\omega t - \pi - \phi}{\omega_s L_d} \quad (5)$$

E a corrente de entrada  $i_{in}$  é dada por:

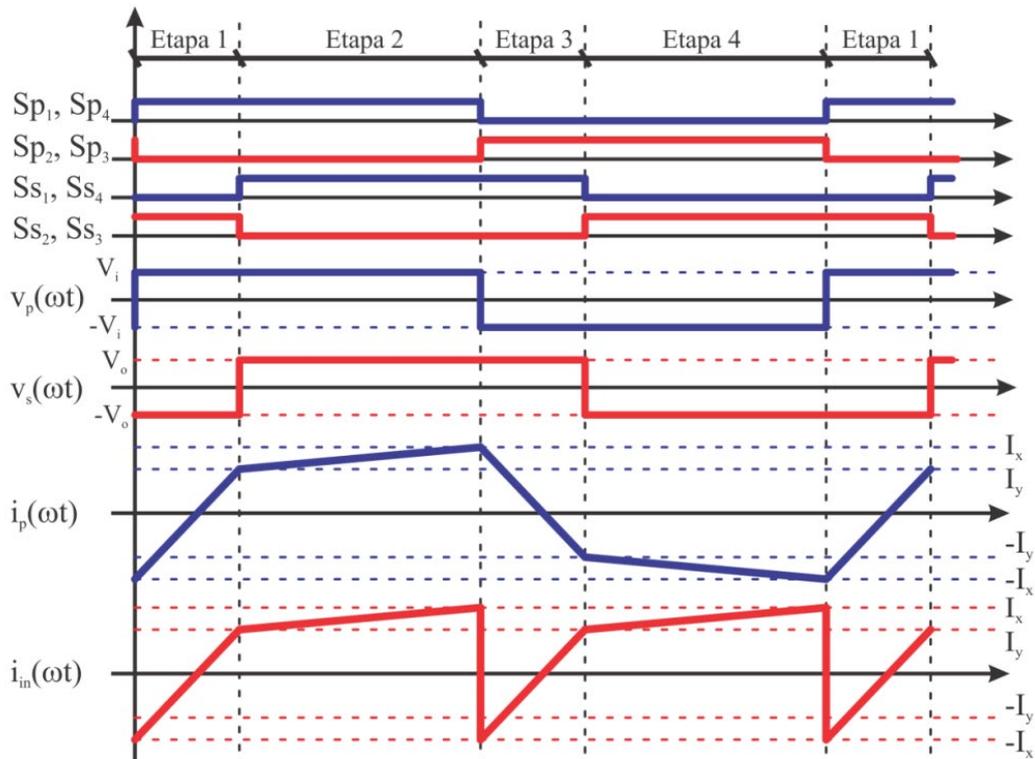
$$i_{in}(\omega t) = \begin{cases} i_p(\omega t) & \text{se } 0 \leq \omega t < \pi \\ -i_p(\omega t) & \text{se } \pi \leq \omega t < 2\pi \end{cases} \quad (6)$$

Onde:

$I_x$  – Corrente no momento  $\pi$ .

$I_y$  – Corrente no momento  $\phi$ .

Figura 3 - Formas de onda dos sinais de comando, tensão resultante da ponte do primário e do secundário, e corrente de entrada do conversor DAB



Fonte: Kirsten (2014)

A Figura 3 mostra ainda a forma de onda da corrente de entrada ( $i_{in}$ ). Nota-se que a corrente em parte do período é negativa. Isso acarreta uma potência aparente superior a potência ativa, ou seja, o sistema terá um fator de potência menor do que a unidade.

### 2.3.2 Projeto do conversor

Para o desenvolvimento do projeto do conversor, se deve levar em consideração as especificações necessárias, por exemplo, a tensão de entrada, a tensão de saída e a transferência máxima de potência.

Com base nisso, as especificações acima irão determinar diversos elementos do projeto. Entretanto, existem duas variáveis que dependem da determinação empírica do projetista: o ângulo de defasagem nominal ( $\phi_N$ ) e a relação de transformação do conversor ( $d$ ).

Para a determinação de  $d$  é utilizada a equação abaixo, utilizando os valores de  $V_i$  e  $V_o$  determinados pela especificação do projeto e  $N_s$  a ser determinado pelo projetista.

$$d = \frac{V_o}{V_i N_s} \quad (7)$$

Assim, após a determinação dessas duas variáveis é possível encontrar o valor de  $L_d$  com a equação abaixo:

$$L_d(\phi_N) = \frac{V_i^2 d\phi_N \eta}{\omega_s P_o} \left( \frac{\pi - |\phi_N|}{\pi} \right) \quad (8)$$

A variável  $\eta$  na equação acima representa a eficiência esperada do conversor, que para esse projeto será determinada como 90%.

Para o projeto físico do conversor DAB também seriam necessários o projeto físico do indutor, do transformador, dos capacitores e das semicondutores. O projeto detalhado de todos esses componentes é demonstrado por Kirsten (2014).

### 2.3.3 Avaliação da performance do conversor

A fim de poder comparar a performance de diferentes conversores será analisado o fator de potência (FP) em diversos pontos de operação. Sendo o cálculo do fator de potência a razão entre a potência ativa de entrada ( $P_i$ ) e a potência aparente ( $S_i$ ), como mostrado na equação abaixo:

$$FP = \frac{P_i}{S_i} \quad (9)$$

Nesse escopo, com o objetivo de facilitar a visualização e comparação dos diferentes fatores de potência os números serão apresentados em porcentagem.

Os cálculos de  $P_i$  e  $S_i$  em diferentes pontos de operações podem ser feitos utilizando as fórmulas abaixo:

$$P_i = \frac{1}{\pi} \int_0^\pi V_i i_{in}(\omega t) d\omega t \quad (10)$$

$$S_i = V_i \sqrt{\frac{1}{\pi} \int_0^\pi i_{in}(\omega t)^2 d\omega t} \quad (11)$$

## 2.4 ALGORITMO GENÉTICO

De acordo com Holland, “adaptação designa qualquer processo pelo qual uma estrutura seja modificada progressivamente para ter melhor desempenho no seu ambiente”. Nesse sentido, foi elaborado um projeto de algoritmo genético (HOLLAND, 1992). Esse, é feito utilizando métodos de busca local, os quais promovem pequenas alterações em soluções potenciais para um problema, até que uma solução ótima ideal seja identificada (COPPIN, 2004).

### 2.4.1 Conceitos

Neste trabalho foi utilizada a representação de algoritmo genético determinada por John Holland. Conforme esse, uma cadeia de bits representa um cromossomo e cada bit ou conjunto de bits representa um gene. Para se converter um gene com valor numérico para uma cadeia de bits é necessário fazer uma conversão de decimal para binário, isso faz com que genes de valor contínuo tenham que ser discretizados, mesmo que se diminua a precisão dos cálculos é um processo necessário para o funcionamento do algoritmo.

Desta forma, um conjunto de cromossomos determina uma população, na qual será aplicado o algoritmo, à fim de evoluir até alcançar a população desejada.

Nesse sentido, outro conceito importante é a aptidão. Sendo essa uma métrica para nivelar os cromossomos de uma população de acordo com sua performance na busca pela resposta do problema determinado e, conseqüentemente, determinar a probabilidade desse cromossomo gerar descendentes na próxima geração.

Com base nos estudos, cruzamento é a combinação dos genes de diferentes cromossomos, com escopo gerar um descendente. Esse cruzamento pode ser feito bit a bit ou em conjuntos de bits, de tal maneira que ao final se gere um cromossomo com as informações dos seus ascendentes e com o mesmo número de genes.

Nesse seguimento, a mutação é a alteração aleatória de genes. No caso, a representação utilizada no projeto é a reversão de um bit. Com probabilidade pré-determinada de ocorrência, a mutação é aplicada para que o sistema encontre diferentes soluções, a fim de apresentar boa performance diante de problemas que apresentam máximos locais.

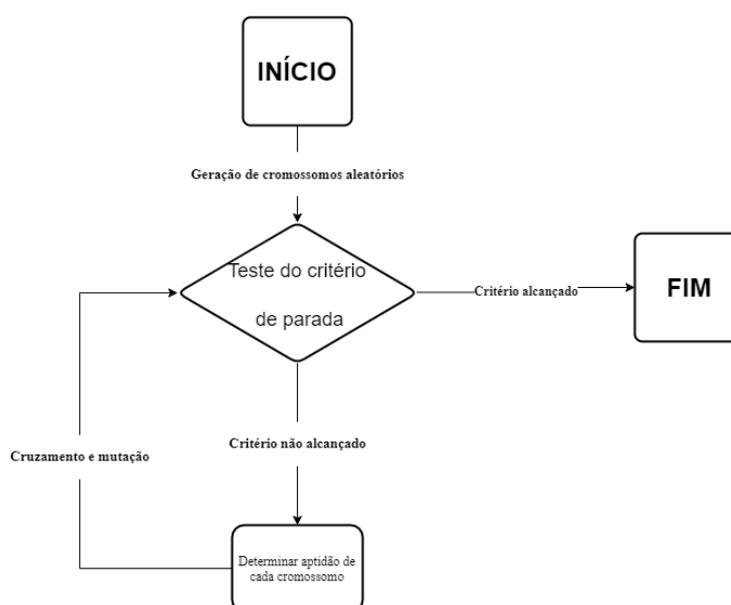
Nessa sequência, critério de terminação determina quando se deve interromper o algoritmo, seja pelo número de repetições, alcance da resposta desejada ou algum erro no sistema. A vista disso, é de suma importância que esse critério seja determinado para que se possa ter controle do algoritmo.

### 2.4.2 O algoritmo

O algoritmo genético ocorre em 5 etapas, as quais podem ser vistas abaixo e no fluxograma da Figura 2:

1. Gerar uma população aleatória de cromossomos.
2. Se o critério de terminação for atendido, interromper o algoritmo. Caso contrário seguir para etapa 3.
3. Determinar a aptidão de cada cromossomo.
4. Aplicar cruzamento e mutação na geração atual a fim de gerar uma nova geração.
5. Retornar à etapa 2.

Figura 4 - Fluxograma de funcionamento de um algoritmo genético



Fonte: Autor (2020) via *software* draw.io.

## 2.5 CONCLUSÃO

Neste capítulo foram apresentados os conceitos básicos por trás da motivação e funcionamento deste trabalho. Com base nisso, esses fundamentos serão utilizados ao longo do desenvolvimento do projeto. Assim, o próximo capítulo irá demonstrar como eles foram aplicados.

### 3 DESENVOLVIMENTO

O propósito do capítulo é apresentar a metodologia de construção do trabalho, as predeterminações, funções e simplificações utilizadas, além do algoritmo utilizado na simulação do projeto.

Nesse escopo, tendo os parâmetros que irão direcionar o projeto e as características do comportamento do sistema, será construído o algoritmo genético associando as variáveis  $N_s$  e  $\phi_N$  aos cromossomos de cada geração.

#### 3.1 PRÉ DETERMINAÇÕES DE PROJETO

Buscou-se as determinações do projeto com base nas especificações do *Powerwall* e da bateria Panasonic NCR18650PF, que estão descritas na Tabela 1. Ademais, na mesma tabela são também apresentadas as especificações que foram determinadas empiricamente.

Tabela 1 - Especificações do pack de baterias

#### **Especificações do *Powerwall***

Tensão interna da bateria:	50 V
Capacidade:	14 kWh

#### **Especificações Panasonic NCR18650PF**

Tensão nominal:	3,6 V
Capacidade:	2700 mAh

#### **Especificações do projeto**

Tensão de entrada ( $V_i$ ):	300 V
Potência nominal ( $P_o$ ):	7 kW
Frequência de comutação ( $f_s$ ):	20 kHz

Fonte: Autor (2020)

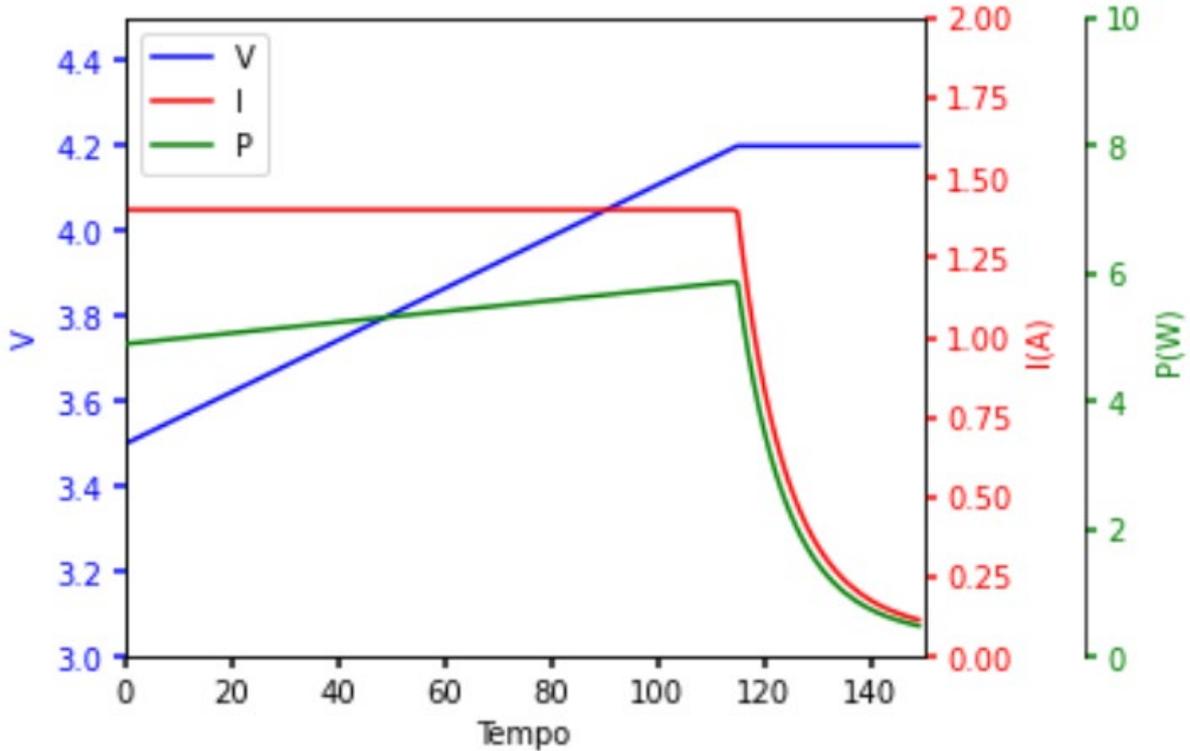
Neste projeto será considerado que não há variação na tensão de saída do conversor.

Com base nessas determinações, é possível simular a curva de carregamento da bateria e encontrar os valores de  $\phi_N$  e  $N_s$  que alcançam o melhor fator de potência.

#### 3.2 CURVAS DE CARREGAMENTO DE BATERIA

A curva de carregamento da bateria NCR18650PF pode ser encontrada em seu *datasheet* apresentado no ANEXO A . Contudo, para a aplicação neste projeto, a curva foi aproximada para a simplificação dos cálculos, não alterando o resultado. Verifica-se assim a curva utilizada na Figura 4 abaixo:

Figura 5 – Curva de carregamento da bateria NCR 18650PF



Fonte: Autor (2020) via *software* JupyterLab

Ao utilizar as células NCR18650PF para simular o pack do *Powerwall* é necessária a utilização de um conjunto como visto na Tabela 2 abaixo:

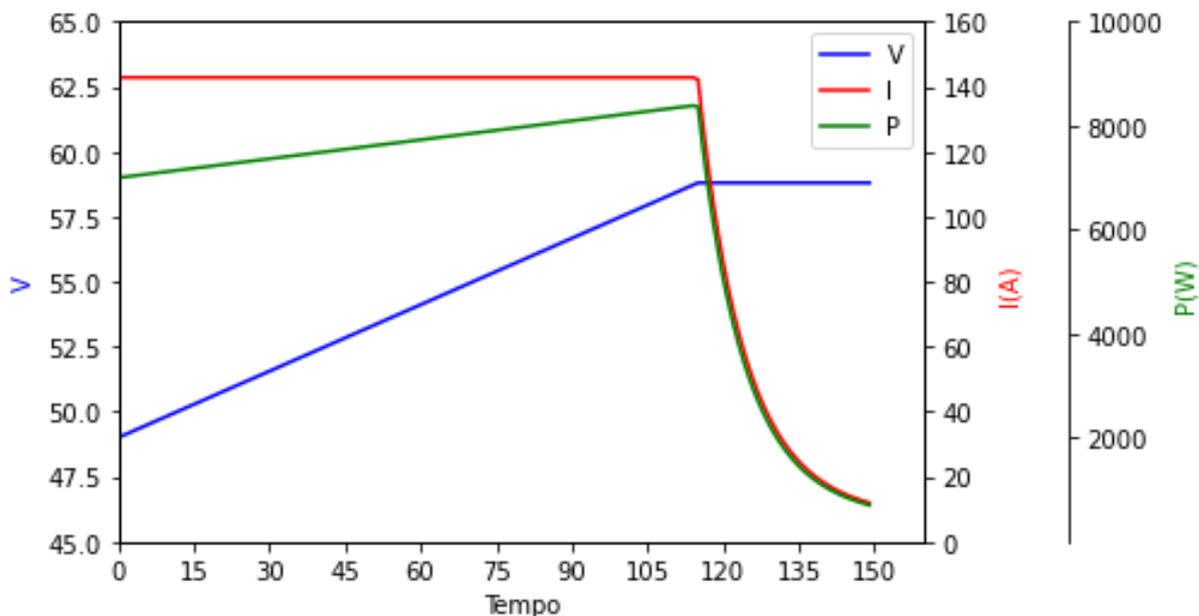
Tabela 2 - Construção do *pack* de baterias

Tensão nominal do <i>pack</i> :	50 V
Tensão nominal da célula:	3,6 V
Número de células em série:	14
Tensão do <i>pack</i> :	50,4 V
Capacidade nominal do <i>pack</i> (kWh):	14 kWh
Capacidade nominal da célula (mAh):	2700 mAh
Capacidade nominal da célula (mWh):	9720 mWh
Número total de células necessárias:	1440
Número de células em paralelo:	102

Fonte: Autor (2020)

A curva final de carregamento do *pack* de baterias pode ser vista na Figura 5 abaixo:

Figura 6 - Curva de carregamento do *pack* de baterias



Fonte: Autor (2020) via *software* JupyterLab

### 3.3 ALGORITMO GENÉTICO PARA ESSA APLICAÇÃO

Para essa aplicação foram utilizadas gerações com diferentes números de cromossomos, com o objetivo de comparar resultados, tendo  $N_s$  e  $\phi_N$  como seus genes. Nesse sentido, a primeira geração é gerada de maneira aleatória utilizando a função *random*. Entretanto, os valores de  $N_s$  foram limitados entre 0,1 e 100, por decisão empírica. Ademais,  $\phi_N$  entre  $1^\circ$  e  $90^\circ$  (KIRSTEN, 2014), com o objetivo do  $\phi$  estar sempre entre 0 e  $\pi/2$ , podendo assim aplicar as funções apresentadas anteriormente. Por mais que esses sejam os valores iniciais para as variáveis, podem ser encontrados valores fora desse limite devido à mutação.

Primordialmente, cada cromossomo dessa geração terá calculado a média do fator de potência (FP) de seu sistema resultante, ao longo do processo de carregamento da bateria, representado pela letra  $\lambda$ .

Ao final, o seu coeficiente de aptidão ( $\rho$ ) da iteração ( $i$ ) e geração ( $g$ ) é dado pela razão entre seu valor de  $\lambda$  e a soma de  $\lambda$  para todos os cromossomos da mesma geração, como pode ser visto em (12).

$$\rho_{i,g} = \frac{\lambda_{i,g}}{\sum \lambda_g} \quad (12)$$

Face ao exposto, tendo os cromossomos elencados pelo seu coeficiente de aptidão, aquele com maior  $\lambda$  é clonado para a geração seguinte. Portanto, os cromossomos remanescentes são gerados a partir da escolha aleatória de  $(x-1)/2$  pares de pais, utilizando seus respectivos coeficientes de aptidão como probabilidade de serem selecionados.

Nesse sentido, ao gerar a nova geração é levado em consideração a mutação, convertendo cada gene em valores binários e fazendo a inversão de cada bit aleatoriamente de acordo com a probabilidade de mutação.

Em conclusão, esse processo é repetido geração a geração até atingir o critério de finalização, determinado pelo número de gerações.

### 3.4 O CÓDIGO

O código deste trabalho, que pode ser encontrado na íntegra no APÊNDICE A, foi feito na linguagem *Python*, utilizando como editor de texto e local de execução o *Jupyter Lab*.

Para a construção da simulação foram primordialmente implementadas as fórmulas de operação do conversor DAB. Onde determinando os valores das variáveis de  $V_i$  nominal, eficiência esperada, frequência de comutação, potência nominal,  $N_s$  e  $\phi_N$ , o programa apresenta as variáveis de apoio e permite o cálculo de  $\lambda$ .

Nesse escopo, para os packs de bateria foram utilizadas as curvas apresentadas anteriormente e determinado que seriam necessárias 14 baterias em série e 102 em paralelo. Logo, a tensão e corrente finais são indicadas pelo fator entre o valor indicado na curva e o número de células em série e paralelo respectivamente, como apresentada na Figura 5 anteriormente.

Nesse contexto, o algoritmo genético foi programado como uma função que executa o processo de uma geração, tendo como variáveis de entrada 2 vetores indicando  $N_s$  e  $\phi_N$ . Após a execução da função é apresentada uma nova geração, como explicado na seção 3.3.

Consequentemente, o programa foi construído para que se utilizasse a função do algoritmo genético em um loop, até que alcançasse o critério de finalização determinado pelo número de gerações.

Em desfecho, o código apresenta os valores de  $N_s$ ,  $\phi_N$  e  $\lambda$  alcançado do melhor cromossomo de cada geração.

### 3.5 CONCLUSÃO

Este capítulo apresentou a metodologia utilizada no desenvolvimento do trabalho e a aplicação do conhecimento obtido. O próximo capítulo irá demonstrar quais foram os resultados do projeto e avaliar a sua efetividade.

## 4 RESULTADOS

Neste capítulo serão apresentados os resultados das simulações feitas ao longo do projeto. Assim, comparando o processo utilizado ao método empírico clássico e indicando a melhor opção ao se realizar um projeto.

### 4.1 MÉTODO CLÁSSICO

A fim de fazer a comparação dos resultados apresentados pelo algoritmo genético, serão apresentadas 3 configurações que seriam utilizadas ao se projetar um conversor pelo método empírico clássico.

As 3 configurações serão feitas com o objetivo de manter  $d = 1$ , tendo  $V_i = 300$  V e a tensão nominal do pack de baterias  $V_o = 50$  V, temos  $N_s = 1/6$ . Os 3 valores para  $\phi_N$  serão  $30^\circ$ ,  $45^\circ$  e  $60^\circ$ .

É possível observar nas figuras abaixo, que mostram a potência consumida para o carregamento da bateria ao longo do tempo, o fator de potência no mesmo instante e o valor de  $\lambda$ , que as curvas do fator de potência, aqui representadas em porcentagem para facilitar a visualização, se comportam de maneira muito diferentes entre si, para os ângulos de maior valor a curva segue uma trajetória linear até o ponto onde se inicia a diminuição da transferência de potência, apresentando então certa elevação no seu FP, mas já baixando rapidamente. Contrário a isso, a curva com  $30^\circ$  de  $\phi_N$  variações suaves no seu FP ao longo de toda a curva, apresentando resultados superiores durante grande parte do tempo de operação e não apenas em um ponto específico.

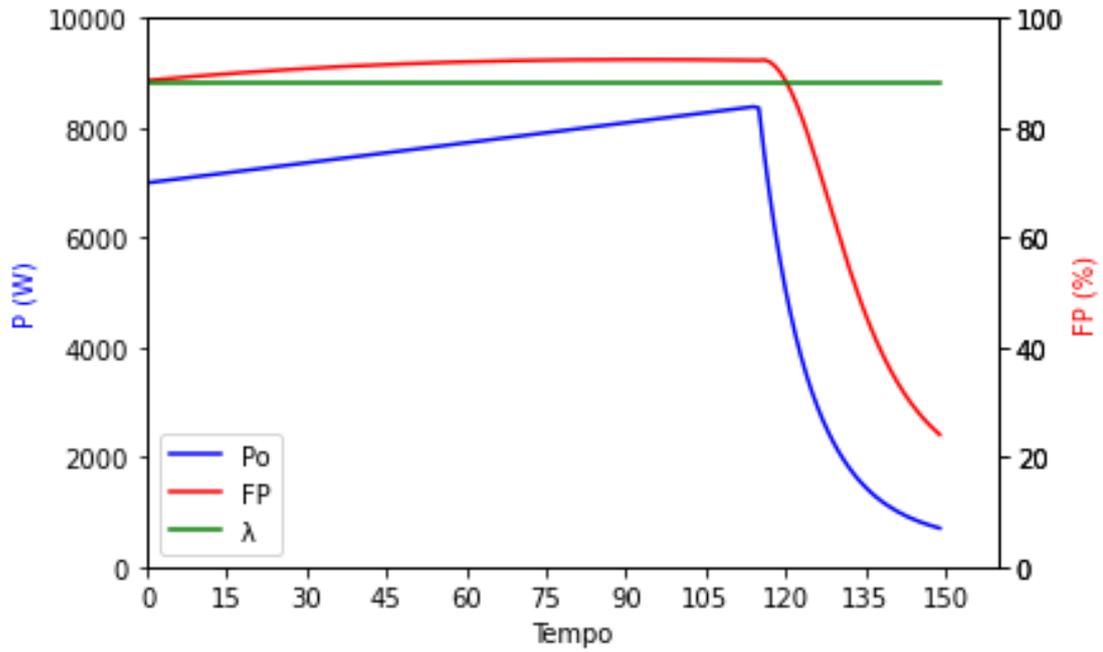
Os valores de  $\lambda$  para cada uma das curvas de carregamento da bateria podem ser vistos na Tabela 3 e nas figuras abaixo:

Tabela 3 - Resultados para projetos com método clássico

$\phi_N$	$\lambda$
$30^\circ$	88,0679%
$45^\circ$	85,4302%
$60^\circ$	81,5452%

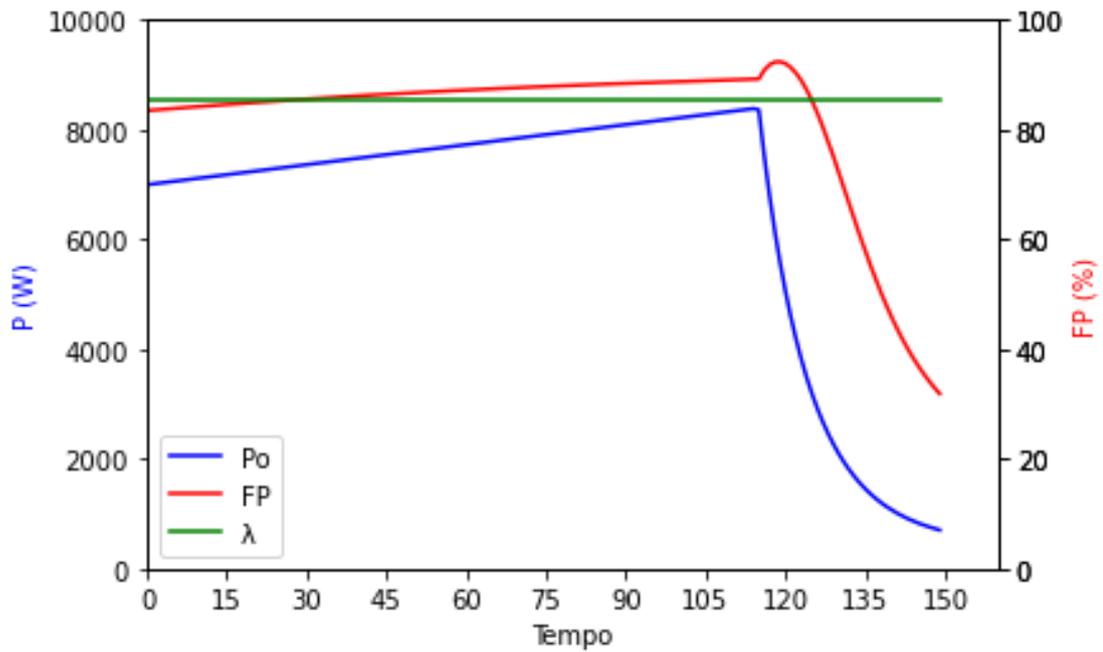
Fonte: Autor (2020)

Figura 7 - Curva de carregamento com  $\phi_N = 30^\circ$

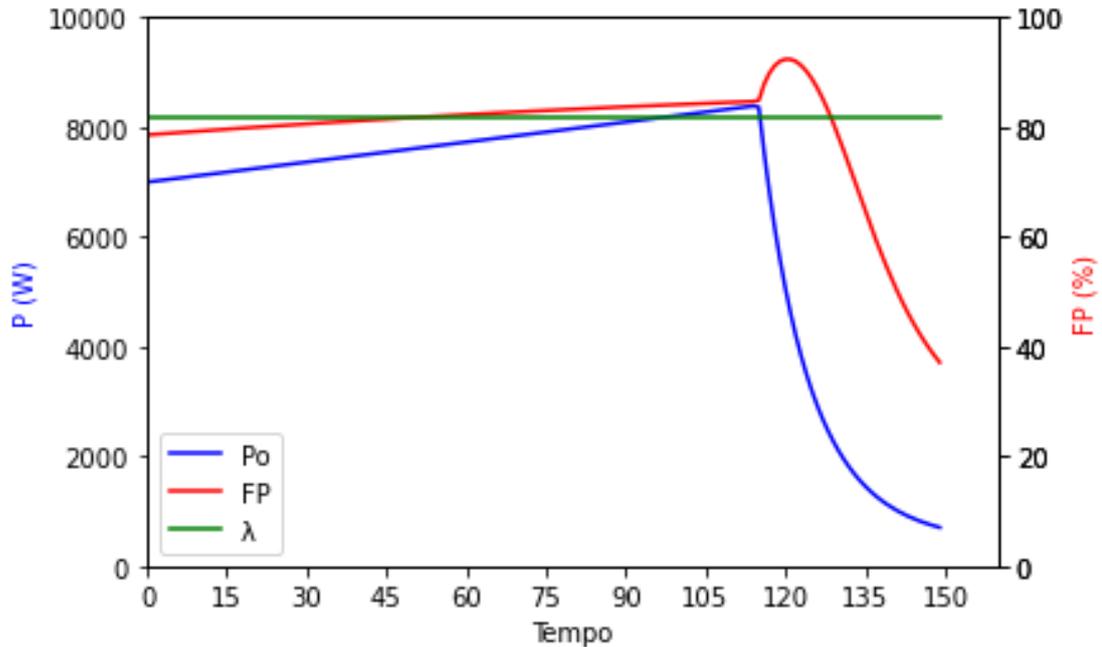


Fonte: Autor (2020) via *software* JupyterLab

Figura 8 - Curva de carregamento com  $\phi_N = 45^\circ$



Fonte: Autor (2020) via *software* JupyterLab

Figura 9- Curva de carregamento com  $\phi_N = 60^\circ$ 

Fonte: Autor (2020) via *software* JupyterLab

## 4.2 MÉTODO COM ALGORITMOS GENÉTICOS

O algoritmo foi construído da maneira apresentada nos capítulos anteriores, foram utilizadas 50 gerações com 21 cromossomos em cada a fim de exaurir o sistema e confirmar que o resultado alcançado se aproximasse ao máximo do resultado ideal.

O passo de mutação, variável que determina a probabilidade de um gene ser modificado entre gerações, permite que o algoritmo avalie possibilidades diferentes daquelas oferecidas pelas gerações anteriores. A importância disso da aleatoriedade em gerações avançadas é evitar a convergência a um máximo regional, algo que acontece quando não se avaliam valores em todo o alcance possível. Para esse projeto foi determinada a utilização de um passo de mutação de 40%.

O Quadro 1 abaixo apresenta as informações de determinadas gerações, possibilitando a observação, que devido aos grandes passos de mutação, os cromossomos seguem bem diversificados até as últimas gerações, entretanto, convergindo para valores próximos do ideal e como, sempre mantivemos o cromossomo de melhor performance para a geração seguinte, o valor máximo de  $\lambda$  segue presente.

Quadro 1 - Gerações com 40% de passo de mutação

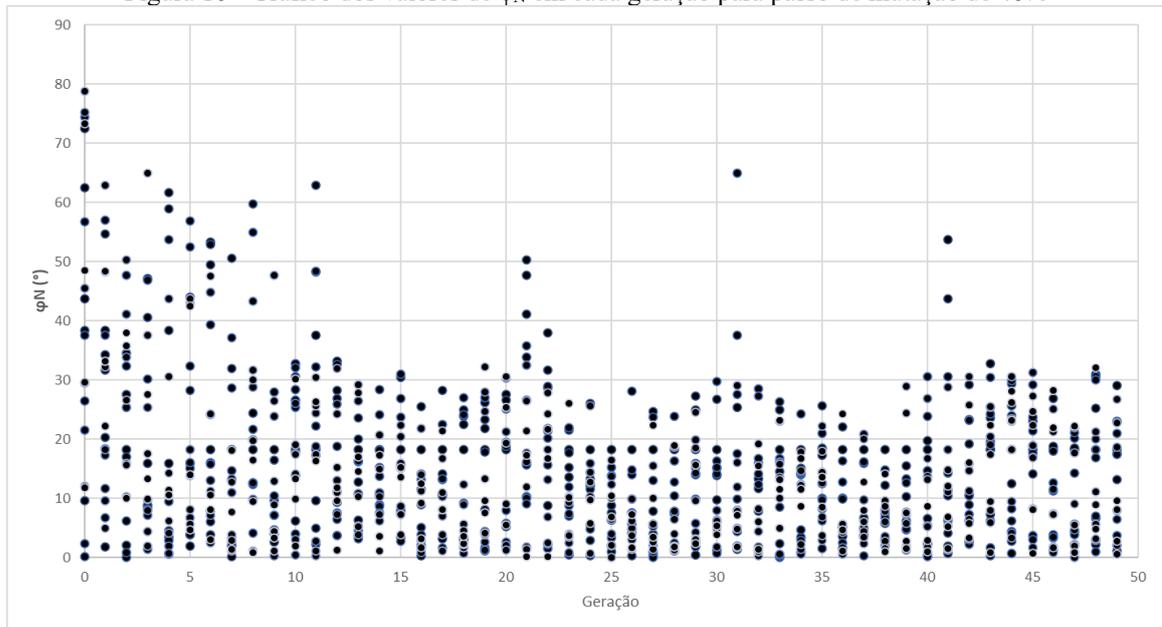
Geração	Iteração	$\phi_N$	$N_s$	$\lambda$
0	0	43,715	64,3420	0,2468%
0	1	72,519	28,6490	0,5543%
0	2	62,51	48,4670	0,3277%
0	3	9,586	78,4120	0,2025%
0	4	38,337	0,6380	22,5890%
0	5	26,402	94,8010	0,1675%
0	6	0,09	32,8140	0,1459%
0	7	37,586	30,3040	0,5240%
0	8	2,316	94,3160	0,1684%
0	9	56,744	78,9300	0,2012%
0	10	21,467	40,0920	0,3961%
0	11	74,388	12,8310	1,2376%
0	12	75,181	45,9360	0,3457%
0	13	45,442	79,3080	0,2002%
0	14	29,511	12,6880	1,2515%
0	15	12,113	96,7320	0,1642%
0	16	78,737	84,7040	0,1875%
0	17	29,652	7,3470	2,1610%
0	18	73,26	95,8900	0,1656%
0	19	11,851	14,4320	1,1003%
0	20	48,439	53,3610	0,2976%
2	0	18,232	0,2490	59,8722%
2	1	2,127	0,0350	0,9549%
2	2	16,958	3,6110	4,3071%
2	3	27,486	0,5260	26,5314%
2	4	6,198	0,4740	15,1250%
2	5	0,036	0,0670	1,1835%
2	6	0,856	1,0180	1,6743%
2	7	32,288	3,0150	5,2428%
2	8	34,587	0,0810	34,4098%
2	9	25,405	0,9960	14,6511%
2	10	47,661	0,4970	28,5815%
2	11	41,094	0,7400	19,7738%
2	12	33,817	5,0810	3,1237%
2	13	50,33	0,0910	51,9881%
2	14	10,234	0,1530	60,0324%
2	15	15,873	0,1890	88,9370%
2	16	26,602	0,6300	22,4456%
2	17	15,658	0,0400	6,4335%
2	18	35,784	0,0260	7,6370%
2	19	10,045	0,9290	14,7028%
2	20	37,932	0,2260	68,1548%
5	0	15,873	0,1890	88,9370%
5	1	1,945	0,1770	42,2564%
5	2	3,807	0,2290	28,3415%
5	3	43,957	0,1080	62,8841%
5	4	15,035	0,1750	88,3740%
5	5	3,828	0,0410	1,8463%
5	6	32,324	0,1170	61,6051%
5	7	28,247	0,0830	30,8605%
5	8	56,792	0,0670	35,0041%
5	9	52,461	0,0280	10,7917%
5	10	16,009	0,1190	40,3671%
5	11	18,246	0,1800	89,7062%
5	12	6,733	0,0030	0,7636%
5	13	13,859	0,1690	84,6412%
5	14	8,127	0,1730	78,9455%
5	15	42,978	0,1220	73,4031%
5	16	5,668	0,0000	0,0000%
5	17	43,648	0,1690	85,4164%
5	18	42,42	0,0360	12,7527%
5	19	4,716	0,0460	2,5636%
5	20	13,952	0,1630	80,0683%
10	0	18,246	0,1800	89,7062%
10	1	6,159	0,2340	38,9408%
10	2	13,907	0,0550	8,8580%
10	3	32,718	0,2080	77,3818%
10	4	26,573	0,1030	43,4062%

10	5	28,353	0,1570	85,9513%
10	6	4,015	0,1660	47,8559%
10	7	30,498	0,1340	74,0707%
10	8	17,639	0,1990	84,7113%
10	9	32,084	0,0220	5,8054%
10	10	25,381	0,2290	70,1092%
10	11	25,99	0,2010	82,4442%
10	12	0,416	0,1790	11,2286%
10	13	3,037	0,1880	62,4197%
10	14	14,077	0,0300	2,0240%
10	15	19,042	0,0590	12,9838%
10	16	17,359	0,2550	57,0571%
10	17	1,927	0,1240	6,3259%
10	18	9,917	0,1910	86,5773%
10	19	13,286	0,1990	83,8123%
10	20	30,172	0,0750	27,4690%
20	0	18,246	0,1800	89,7062%
20	1	27,586	0,2320	68,8891%
20	2	5,21	0,2160	45,6057%
20	3	26,609	0,0410	10,5155%
20	4	5,706	0,0490	3,2959%
20	5	18,796	0,2090	78,9620%
20	6	1,878	0,0560	1,4625%
20	7	19,487	0,1360	62,7179%
20	8	1,945	0,1320	7,8652%
20	9	25,108	0,0410	10,0210%
20	10	7,939	0,0440	3,9106%
20	11	2,548	0,2150	26,2212%
20	12	21,262	0,0500	11,3127%
20	13	30,215	0,0650	22,0229%
20	14	2,034	0,1960	41,8991%
20	15	1,26	0,2210	11,7697%
20	16	8,982	0,0840	11,5939%
20	17	19,286	0,0720	17,8138%
20	18	30,534	0,1750	87,9543%
20	19	5,481	0,1240	17,6086%
20	20	25,398	0,0310	7,1584%
49	0	18,246	0,1800	89,7062%
49	1	29,062	0,1930	84,6052%
49	2	1,155	0,0530	1,0565%
49	3	13,103	0,0480	6,9537%
49	4	17,548	0,0170	2,6635%
49	5	23,07	0,0400	9,0405%
49	6	22,863	0,0360	7,8599%
49	7	17,338	0,1890	88,8348%
49	8	2,031	0,1350	8,9281%
49	9	20,981	0,0410	8,6014%
49	10	3,71	0,0750	4,1351%
49	11	6,479	0,1630	58,8614%
49	12	7,906	0,0170	1,3482%
49	13	18,663	0,1660	86,2743%
49	14	2,001	0,0290	0,8922%
49	15	26,66	0,0530	14,8316%
49	16	22,812	0,0960	33,9133%
49	17	0,491	0,1790	13,4238%
49	18	8,15	0,0660	7,0705%
49	19	2,812	0,2010	43,6261%
49	20	9,547	0,2310	53,3631%

Fonte: Autor (2020)

Dessa maneira, também é possível visualizar na Figura 9 abaixo a distribuição dos valores de  $\phi_N$  entre as 50 gerações, demonstrando a convergência dos valores, mas mantendo certa abrangência devido ao grande passo de mutação.

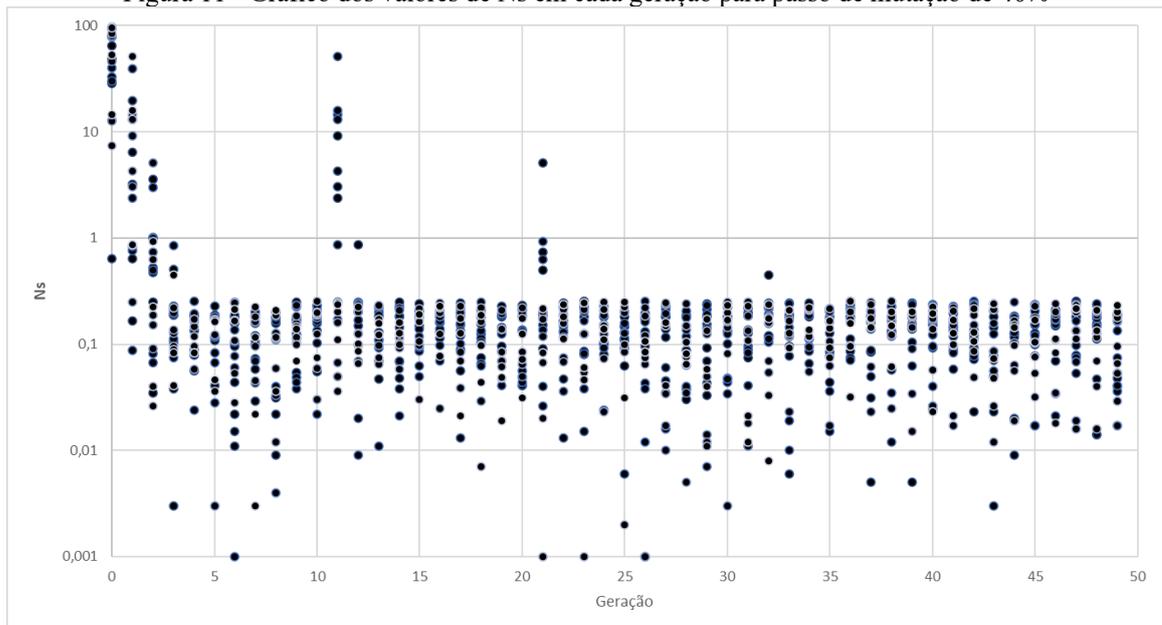
Figura 10 - Gráfico dos valores de  $\phi_N$  em cada geração para passo de mutação de 40%



Fonte: Autor (2020) via Excel

Seguindo o raciocínio, a Figura 10 demonstra em escala linear a distribuição dos valores de  $N_s$  para cada geração. É possível observar uma convergência muito maior que para os valores de  $\phi_N$ , demonstrando que o algoritmo encontrou a região de máximo  $\lambda$  e explorou essa região buscando maximizar os resultados.

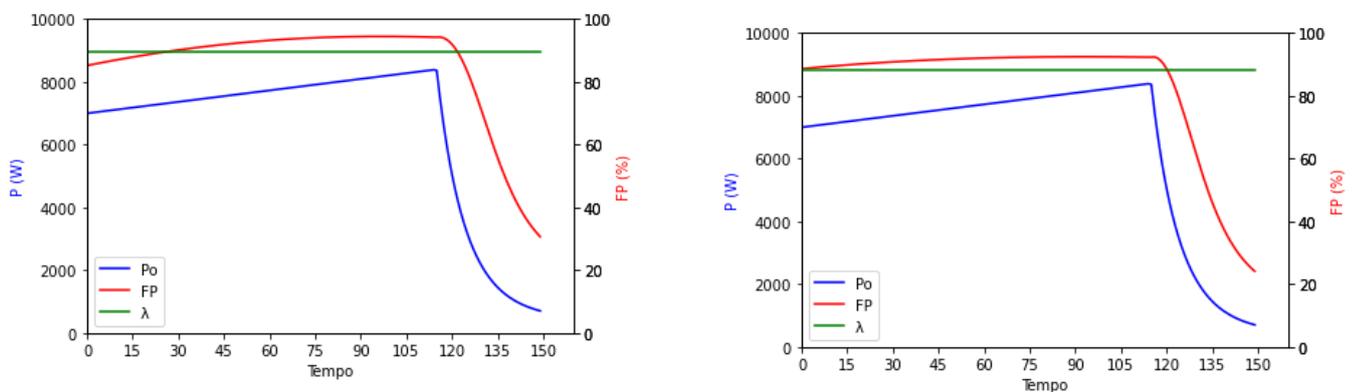
Figura 11 - Gráfico dos valores de  $N_s$  em cada geração para passo de mutação de 40%



Fonte: Autor (2020) via Excel

Nesse escopo, a melhor solução encontrada pelo algoritmo foi no processo com 40% de passo de mutação, tendo  $\lambda = 89,7062\%$ , utilizando  $N_s = 0,18$  e  $\phi_N = 18,246^\circ$ . Abaixo na Figura 11 é possível ver o comportamento do sistema resultante ao lado esquerdo e ao lado direito o resultado apresentado pelo método clássico.

Figura 12- Curvas de carregamento da bateria



Fonte: Autor (2020) via *software* JupyterLab

Como se pode ver Figura 11, por mais que a diferença no valor de  $\lambda$  seja de aproximadamente 1%, as curvas do FP se comportam de maneiras diferentes, a curva apresentada pelo método clássico tem um comportamento mais linear comparado à curva encontrada pelo algoritmo genético. Isso indica que se o sistema fosse avaliado em certos pontos de operação os resultados seriam mais distantes do que o que foi apresentado.

Esse pode não ser um resultado muito expressivo, entretanto ele representa muito mais do que o 1% de variação, o que se pode tirar desse resultado é o fato que um algoritmo em alguns minutos e poucas iterações pôde encontrar um resultado superior que aquele apresentado por um engenheiro com experiência. Não podemos dizer que outro engenheiro, com experiências diferentes e possivelmente mais tempo, não encontraria resultados melhores, entretanto, o uso de um algoritmo remove essa variável da equação.

### 4.3 CONCLUSÃO

Este capítulo demonstrou como a utilização de um sistema de desenvolvimento acompanhado de metodologias de inteligência artificial apresenta resultados com performances

melhores que aqueles feitos pelo método clássico de projeto, abrindo a possibilidade de novos testes, com mais variáveis e abrangências, possivelmente encontrando resultados ainda melhores.

Essa eficiência também se apresenta no quesito trabalho do projetista, que pode transferir a responsabilidade de testes e avaliação de resultados ao algoritmo e focar em outros aspectos do projeto.

É importante ressaltar que a experiência e conhecimento do engenheiro seguem sendo necessárias para o entendimento e implementação dos resultados alcançados, entretanto, o profissional com entendimento profundo de engenharia e dos métodos estatísticos aqui utilizados, pode alcançar resultados melhores que aqueles que alcançaria sem essas ferramentas.

## 5 CONCLUSÃO

Em conclusão, esse trabalho teve como objetivo um estudo de caso com inteligência artificial, oferecendo uma metodologia alternativa de projeto para um problema de engenharia. Foi determinado o uso de algoritmos genéticos em um projeto de um conversor DAB, sendo utilizado para o carregamento de um pack de baterias.

Dessa maneira, o conversor foi modelado na linguagem *Python*, assim como as curvas de carregamento do pack de baterias. O algoritmo genético foi programado utilizando a mesma linguagem, utilizando a modelagem no seu processo de ranqueamento de cada cromossomo dentro de uma geração a fim de utilizar os melhores genes no processo evolutivo.

Ao fim desse trabalho, é possível averiguar que algoritmos genéticos são ferramentas úteis e funcionais a serem aplicadas no desenvolvimento e pesquisas no âmbito de engenharia. Deste modo, a utilização deles possibilita a verificação mais rápida de resultados, bem como de cenários que poderiam ser descartados com o escopo de diminuir o custo computacional.

Os resultados encontrados com a nova metodologia apresentaram mais de 1% no valor de  $\lambda$ , o que pode acarretar um grande aumento de eficiência no projeto final.

Nesse escopo, o objetivo do trabalho foi alcançado, ao encontrar diferentes alternativas ao método clássico de planejamento de projetos. A partir desse ponto, é interessante aumentar o escopo do algoritmo, permitindo que outros aspectos do projeto sejam determinados pelo sistema. Além disso, é necessária a adição de perdas à simulação, isso poderia aumentar ainda mais a disparidade entre os resultados, alcançando possivelmente grandes aumentos de eficiência. Contudo, o maior obstáculo para o uso de algoritmos genéticos para diferentes projetos é a necessidade de modelar a simulação para cada caso, para a popularização do método é importante que seja aplicável de maneira mais simples, como por exemplo a integração com um *software* de simulação SPICE.

Em conclusão, é possível perceber que em projetos futuros será viável aumentar o escopo de aplicação do algoritmo. Assim, reduzindo o trabalho do engenheiro na determinação das mesmas e possibilitando descobertas de metodologias mais eficientes.

## REFERÊNCIAS

FUMO, David, **Why is everyone talking about artificial intelligence?** Disponível em: <https://towardsdatascience.com/why-is-everyone-talking-about-ai-73bab31bf9c1>. Acesso em: 19 dez. 20.

COPPIN, Ben, **Artificial intelligence illuminated**, 1st ed. Boston: Jones and Bartlett Publishers, 2004.

FORTE, Vincent J., **Smart Grid at National Grid**, in: 2010 Innovative Smart Grid Technologies (ISGT), Gaithersburg, MD, USA: IEEE, 2010, p. 1–4.

LAZAR, Jim, **Teaching the “Duck” to Fly**, The Regulatory Assistance Project (RAP), 2014. Disponível em: <https://www.raonline.org/knowledge-center/teaching-the-duck-to-fly-second-edition/#:~:text=Jim%20Lazar%20confirms%20that%20electric,the%202014%20%E2%80%9CTeaching%20the%20Duck>. Acesso em: 29 ago. 20.

BOROYEVICH, Dushan et al, **Future electronic power distribution systems a contemplative view**, in: 2010 12th International Conference on Optimization of Electrical and Electronic Equipment, Brasov, Romania: IEEE, 2010, p. 1369–1380.

STIENEKER, Marco; DE DONCKER, Rik W., **Dual-active bridge dc-dc converter systems for medium-voltage DC distribution grids**, in: 2015 IEEE 13th Brazilian Power Electronics Conference and 1st Southern Power Electronics Conference (COBEP/SPEC), Fortaleza, Brazil: IEEE, 2015, p. 1–6.

HOLLAND, John H., **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**, 1st MIT Press ed. Cambridge, Mass: MIT Press, 1992.

KIRSTEN, André Luís, **METODOLOGIA DE PROJETO DO CONVERSOR DAB APLICADO A TRANSFORMADORES DE ESTADO SÓLIDO**, Universidade Federal de Santa Maria. Santa Maria, 2014.

HODGE, Tyler, **Hourly electricity consumption varies throughout the day and across seasons**, U.S. energy information administration, 2020. Disponível em: <https://www.eia.gov/todayinenergy/detail.php?id=42915#> . Acesso em: 24 ago. 20.

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **RN 687**. Brasília, 2015.

PANASONIC, **NCR 18650PF**. Disponível em: [https://b2b-api.panasonic.eu/file\\_stream/pids/fileversion/3447](https://b2b-api.panasonic.eu/file_stream/pids/fileversion/3447). Acesso em: 12 set. 20.

TESLA. **Powerwall**. Disponível em: [https://www.tesla.com/sites/default/files/pdfs/powerwall/Powerwall%20\\_AC\\_Datasheet\\_en\\_northamerica.pdf](https://www.tesla.com/sites/default/files/pdfs/powerwall/Powerwall%20_AC_Datasheet_en_northamerica.pdf). Acesso em: 12 set. 19.

ANEXO A – DATASHEET DA BATERIA NCR18650PF

Panasonic
Lithium Ion  
NCR18650PF

---

### Features & Benefits

- High energy and power density
- Long, stable, high power
- High safety performance
- Ideal for power assisted bicycles, 2-way radios, medical devices and robotics.

\* At temperatures below 10°C, charge at a 0.25C rate.

### Specifications

Rated capacity <sup>(1)</sup>	Min. 2700mAh
Capacity <sup>(2)</sup>	Min. 2750mAh Typ. 2900mAh
Nominal voltage	3.6V
Charging	CC-CV, Std. 1375mA, 4.20V, 4.0 hrs
Weight (max.)	48.0 g
Temperature	Charge*: 0 to +45°C Discharge: -20 to +60°C Storage: -20 to +50°C
Energy density <sup>(3)</sup>	Volumetric: 577 Wh/l Gravimetric: 207 Wh/kg

(1) At 20°C (2) At 25°C (3) Energy density based on bare cell dimensions

### Dimensions

\*With tube

For Reference Only

---

### Charge Characteristics

Charge: CC-CV 0.5C (max) 4.20V, 100mA cut-off at 25°C

### Cycle Life Characteristics

Charge: CC-CV 0.5 (max) 4.20V, 100mA cut-off at 25°C  
Discharge: CC 1C, 2.5V cut-off at 25°C

---

### Discharge Characteristics (by temperature)

Charge: CC-CV 0.5C (max) 4.20V, 100mA cut-off at 25°C  
Discharge: CC 1C, 2.5V cut-off at each temperature

### Discharge Characteristics (by rate of discharge)

Charge: CC-CV 0.5C (max) 4.20V, 100mA cut-off at 25°C  
Discharge: CC 1C, 2.5V cut-off at 25°C

The data in this document is for descriptive purposes only and is not intended to make or imply any guarantee or warranty.

## ANEXO B – DATASHEET DO POWERWALL

### POWERWALL

Tesla Powerwall is a fully-integrated AC battery system for residential or light commercial use. Its rechargeable lithium-ion battery pack provides energy storage for solar self-consumption, time-based control, and backup.

Powerwall's electrical interface provides a simple connection to any home or building. Its revolutionary compact design achieves market-leading energy density and is easy to install, enabling owners to quickly realize the benefits of reliable, clean power.



#### PERFORMANCE SPECIFICATIONS

AC Voltage (Nominal)	120/240 V
Feed-In Type	Split Phase
Grid Frequency	60 Hz
Total Energy	14 kWh
Usable Energy	13.5 kWh
Real Power, max continuous	5 kW (charge and discharge)
Real Power, peak (10 s, off-grid/backup)	7 kW (charge and discharge)
Apparent Power, max continuous	5.8 kVA (charge and discharge)
Apparent Power, peak (10 s, off-grid/backup)	7.2 kVA (charge and discharge)
Maximum Supply Fault Current	10 kA
Maximum Output Fault Current	32 A
Overcurrent Protection Device	30 A
Imbalance for Split-Phase Loads	100%
Power Factor Output Range	+/- 1.0 adjustable
Power Factor Range (full-rated power)	+/- 0.85
Internal Battery DC Voltage	50 V
Round Trip Efficiency <sup>1</sup>	90%
Warranty	10 years

<sup>1</sup>Values provided for 25°C (77°F), 3.3 kW charge/discharge power.

<sup>2</sup>In Backup mode, grid charge power is limited to 3.3 kW.

<sup>3</sup>AC to battery to AC, at beginning of life.

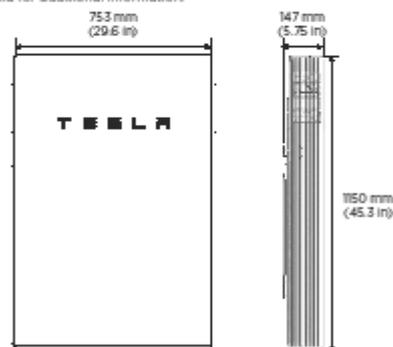
#### COMPLIANCE INFORMATION

Certifications	UL 1642, UL 1741, UL 1973, UL 9540, IEEE 1547, UN 38.3
Grid Connection	Worldwide Compatibility
Emissions	FCC Part 15 Class B, ICES 003
Environmental	RoHS Directive 2011/65/EU
Seismic	AC156, IEEE 603-2005 (high)

#### MECHANICAL SPECIFICATIONS

Dimensions <sup>1</sup>	1150 mm x 755 mm x 147 mm (45.3 in x 29.6 in x 5.75 in)
Weight <sup>1</sup>	114 kg (251.3 lbs)
Mounting options	Floor or wall mount

<sup>1</sup>Dimensions and weight differ slightly if manufactured before March 2019. Contact Tesla for additional information.



#### ENVIRONMENTAL SPECIFICATIONS

Operating Temperature	-20°C to 50°C (-4°F to 122°F)
Recommended Temperature	0°C to 30°C (32°F to 86°F)
Operating Humidity (RH)	Up to 100%, condensing
Storage Conditions	-20°C to 30°C (-4°F to 86°F) Up to 95% RH, non-condensing State of Energy (SoE): 25% initial
Maximum Elevation	3000 m (9843 ft)
Environment	Indoor and outdoor rated
Enclosure Type	NEMA 3R
Ingress Rating	IP67 (Battery & Power Electronics) IP56 (Wiring Compartment)
Wet Location Rating	Yes
Noise Level @ 1m	< 40 dBA at 30°C (86°F)

TESLA

TESLA.COM/ENERGY

## APÊNDICE A – CÓDIGO DO PROJETO

```

# Cálculo parâmetros DAB # Cell 1
import numpy as np import math as math
import scipy.integrate as integrate import sympy as sp
from scipy.optimize import fsolve from scipy.optimize import fminbound
import matplotlib.pyplot as plt import pandas as pd
import random import datetime

pi = np.pi

#Funções que são calculadas apenas uma vez

# Cálculo Ns baseado nos valores nominais desejados
def func_Ns(Vi_nominal, Vo_nominal): Ns = Vo_nominal/Vi_nominal return(Ns)

# Cálculo do indutor auxiliar baseado nos valores nominais
def func_Indutor_auxiliar(eficiencia, phi_nominal_rad, Vi_nominal, Vo_nominal, Po_nominal, omega_s): Ld =
(Vi_nominal**2*phi_nominal_rad*(eficiencia/100))/(omega_s*Po_nominal)*((pi-abs(phi_nominal_rad
))/pi)
return Ld

# Cálculo da corrente de saída em função de wt
def func_i_out(wt, phi_rad, Ns, d): phi_rad = float(phi_rad)
if 0 <= wt < phi_rad or pi <= wt < (pi+phi_rad): i_out = (-1/Ns)*func_ip(wt,phi_rad,Ns,d)[0]
elif phi_rad <= wt < pi or (pi+phi_rad) <= wt < (2*pi): i_out = (1/Ns)*func_ip(wt,phi_rad,Ns,d)[0]
else:
print("Error, wt = ", wt, " phi rad = ", phi_rad)
return(i_out)

#Funções que devem ser calculadas continuamente # Cálculo d baseado nas tensões
atuais e Ns
def func_d(Vi, Vo, Ns): d = float(Vo/(Vi*Ns)) return(d)

# Cálculo de phi para uma potencia determinada
def func_phi_pot(Vi, d, omega_s, Ld, Po, Ns):

Vi = Vi d = d
omega_s = omega_s Ld = Ld
Po = Po Ns = Ns

def phi_finder(phi_rad):

Pot_o = abs(Po - (Vo/pi)*integrate.quad(func_i_out,0,pi,args = (phi_rad, Ns, d))[0])
return(Pot_o)

phi_rad = fminbound(phi_finder,0,pi/2)
return phi_rad

def func_Pot_out(Vo, phi_rad, Ns, d):
Pot_o = (Vo/pi)*float(integrate.quad(func_i_out,0,pi,args = (phi_rad, Ns, d))[0])
return(Pot_o)

def func_S_in(Vi, phi_rad, Ns, d):
S_in = Vi*math.sqrt((1/pi)*float(integrate.quad(func_i_in_2, 0, pi, args = (phi_rad, Ns, d))[0]))
return(S_in)

def func_P_in(Vi, phi_rad, Ns, d):
P_in = (Vi/pi) * float(integrate.quad(func_i_in, 0, pi, args = (phi_rad, Ns, d))[0])
return(P_in)

# Cálculo de Qin - não é usado no algoritmo
def func_Q_in(Vi,phi_rad, Ns, d): P_in = func_P_in(Vi,phi_rad,Ns,d) S_in =
func_S_in(Vi,phi_rad,Ns,d)

```

```

Q_in = math.sqrt((S_in**2)-(P_in**2))
    return(Q_in)

    # Cálculo da Corrente no enrolamento primário em função de wt
def func_ip(wt, phi_rad, Ns, d):
    Ix = (Vi/(omega_s*Ld))*((2*phi_rad*d) - (pi*(d-1)))/2)
    Iy = (Vi/(omega_s*Ld))*((2*phi_rad) + (pi*(d-1)))/2)
    if 0 <= wt < phi_rad:
        ip = -Ix + (((Vi + (Vo/Ns)))*(wt/(omega_s*Ld)))
            elif phi_rad <= wt < pi:
                ip = Iy + (Vi - (Vo/Ns))*( (wt-phi_rad) / (omega_s*Ld) )
    elif pi <= wt < (pi+phi_rad):
        ip = Ix - (Vi + (Vo/Ns))*((wt-pi) / (omega_s*Ld) )
    elif pi+phi_rad <= wt < 2*pi:
        ip = -Iy - (-Vi+(Vo/Ns))*((wt-phi_rad-pi)/(omega_s*Ld))
            else:
                print("Something is wrong") ip = [ip,Ix,Iy]
    return ip

    # Cálculo da Corrente de entrada em função de wt
def func_i_in(wt, phi_rad, Ns, d):
    if 0 <= wt < pi:
        i_in = func_ip(wt,phi_rad,Ns,d)[0]
            elif pi <= wt < 2*pi:
                i_in = - func_ip(wt,phi_rad,Ns,d)[0]
            else:
                print("Something is wrong") print("wt = ", wt)
    return(i_in)

    # Cálculo da Corrente de entrada ao quadrado em função de wt (usado para calcular Sin)
def func_i_in_2(wt, phi_rad, Ns, d):
    i_in_2 = float(func_i_in(wt,phi_rad, Ns,d)**2)
    return(i_in_2)

    #Início dos cálculos

    # Cálculos da bateria

    # Tensão em função do tempo em minutos (uma célula)
def func_Voltage_bateria(tempo):
    if tempo < 115:
        V_bateria = 6.087*(10**(-3))*tempo + 3.5
            else:
                V_bateria = 4.2
    return(V_bateria)

    # Corrente em função do tempo em minutos (uma célula)
def func_Current_bateria(tempo):
    if tempo < 115:
        I_bateria = 1.4
            else:
                I_bateria = 0.06647457 + (2285330 - 0.06647457)/(1 + (tempo/36.77715)**12.59453)
    return(I_bateria)

    # Potência instantanea do pack de bateria
def func_potencia_pack(Nserie, Nparalelo, tempo): V_pack = Nserie * func_Voltage_bateria(tempo)
    I_pack = Nparalelo * func_Current_bateria(tempo) Potencia_pack = V_pack*I_pack
    return(Potencia_pack)

    # Energia gasta para carregamento do pack de baterias em função do tempo e tamanho do pack
def func_energia_bateria(To, Tf, Num_series, Num_parallel, Vin):
    global Vi Vi = Vin
    def func_interna(tempo):
        global Vo

```

```

P_bat = func_potencia_pack(Num_series,Num_paralel,tempo) Vo =
Num_series*func_Voltage_bateria(tempo)
d = func_d(Vi,Vo,Ns)
phi_rad = func_phi_pot(Vi,d,omega_s,Ld, P_bat, Ns) P_in = func_P_in(Vi, phi_rad, Ns,d)
S_in = func_S_in(Vi, phi_rad, Ns,d)
    return(P_in,S_in)

    #energia_total = integrate.quad(func_interna,To, Tf)
ENERGIA_PIN = [] ENERGIA_SIN = []
for tempo in range(To,Tf,1): ENERGIA_PIN.append(func_interna(tempo)[0])
    ENERGIA_SIN.append(func_interna(tempo)[1])
energia_total_P = sum(ENERGIA_PIN) energia_total_S = sum(ENERGIA_SIN) eficiencia =
energia_total_P/energia_total_S return(eficiencia)

    #Função de mutação binária (número a ser mutado, probabilidade de mutação)
def binary_mutation(integer, mutation_prob):
    #converte o inteiro em binario
    bin_num = bin(integer) num_str = ""
    #para o comprimento do numero, se um valor aleatorio for menor que a probabilidade de mutação irá trocar o bit
    for bit in range(2, len(bin_num)):
        if random.random()<mutation_prob:
            if bin_num[bit] == "0": new_bit = "1"
                else:
                    new_bit = "0"
                else:
                    new_bit = bin_num[bit] num_str = num_str + new_bit
            #print(num_str) #print(bin_num)
            #converte de volta para inteiro mutated_integer = int(num_str,2)
            #print(mutated_integer) return(mutated_integer)

print("Completed")

```

Fonte: Autor (2020)