

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

José Wilson Sousa Prado

**ABORDAGEM DE CONVERSORES RESSONANTES CLASSE E QUADRÁTICO
COM APLICAÇÃO DE PSO**

Santa Maria, RS

2021

José Wilson Sousa Prado

**ABORDAGEM DE CONVERSORES RESSONANTES CLASSE E QUADRÁTICO
COM APLICAÇÃO DE PSO**

Dissertação apresentada ao Curso de mestrado do Programa de Pós-Graduação em Engenharia Elétrica, Área de concentração em processamento de energia elétrica, da Universidade Federal de Santa Maria (UFSM), como requisito parcial para obtenção do grau de **Mestre em Engenharia Elétrica**.

Orientador: Prof. Dr. Fábio Ecke Bisogno

Santa Maria, RS

2021

PRADO, José Wilson Sousa
ABORDAGEM DE CONVERSORES RESSONANTES CLASSE E
QUADRÁTICO COM APLICAÇÃO DE PSO / José Wilson Sousa
PRADO.- 2021.
89 p.; 30 cm

Orientador: Fábio Ecke Bisogno
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Engenharia Elétrica, RS, 2021

1. Otimização por enxame de partículas 2. Abordagem
normalizada 3. Eletrônica de potência 4. Conversores
ressonantes I. Bisogno, Fábio Ecke II. Título.

JOSÉ WILSON SOUSA PRADO

**ABORDAGEM DE CONVERSORES RESSONANTES CLASSE E QUADRÁTICO
COM APLICAÇÃO DE PSO**

Dissertação apresentada ao Curso de mestrado do Programa de Pós-Graduação em Engenharia Elétrica, Área de concentração em processamento de energia elétrica, da Universidade Federal de Santa Maria (UFSM), como requisito parcial para obtenção do grau de **Mestre em Engenharia Elétrica**.

Aprovado em 21 de maio de 2021:

Fábio Ecke Bisogno, Dr. (UFSM) - Videoconferência

(Presidente/Orientador)

Vinícius Foletto Montagner, Dr. (UFSM) - Videoconferência

Gustavo Guilherme Koch, Dr. (URGS) - Videoconferência

Santa Maria, RS

2021

AGRADECIMENTOS

Agradeço a Deus por tudo o que me foi concedido para a realização deste trabalho, sobretudo, em relação ao encontro com algumas pessoas que contribuíram e foram essenciais nesta jornada:

- ao meu orientador Professor Dr. Fábio Ecke Bisogno, pela oportunidade e por todo conhecimento repassado, pelo incentivo ao estudo e ao tema desta dissertação;

- à minha esposa Lívia Prado, pelo amor, pelo companheirismo, pela força necessária para o cumprimento desta etapa e pela dedicação.

- ao meu filho Benício Prado pela paciência e compreensão pelas vezes em que tive que abdicar de alguns momentos com ele;

- aos amigos e familiares que ofereceram apoio e incentivo;

- à Universidade Federal de Santa Maria;

- e a todos que de alguma forma contribuíram para este trabalho.

Muito obrigado.

RESUMO

ABORDAGEM DE CONVERSORES RESSONANTES CLASSE E QUADRÁTICO COM APLICAÇÃO DE PSO

AUTOR: JOSÉ WILSON SOUSA PRADO

ORIENTADOR: FABIO ECKE BISOGNO

Esta dissertação apresenta a aplicação do algoritmo de otimização por enxame de partículas (PSO) para obtenção de parâmetros de projeto da etapa inversora de um conversor Classe E quadrático. Este conversor pode ser projetado modelando-se a etapa retificadora como uma carga de impedância equivalente. Quanto à etapa inversora, é feita a decomposição em parâmetros ressonantes, pois permite a normalização do sistema. As equações diferenciais que representam o circuito dependem de um conjunto de variáveis que, de acordo com metodologias da literatura, dependem de valores estimados a partir de figuras de mérito, por exemplo. Como contribuição deste trabalho, um algoritmo PSO é utilizado aqui para selecionar estes valores, retirando a necessidade de arbitrar valores para variáveis e também diminuindo o tempo de interação homem-máquina no projeto das variáveis do circuito. Com base nos dados obtidos, foi feita a simulação e posterior implementação para validação dos resultados teóricos. Os resultados práticos mostraram-se satisfatório e próximos ao do modelo desenvolvido.

Palavras-chave: Otimização, conversor, ressonância, normalização.

ABSTRACT

APPROACH OF CLASS AND QUADRATIC RESONANT CONVERTERS WITH PSO APPLICATION

AUTHOR: JOSÉ WILSON SOUSA PRADO

ADVISOR: FABIO ECKE BISOGNO

This dissertation presents the application of the particle swarm optimization algorithm (PSO) to obtain design parameters for the inverting step of a quadratic Class E converter. This converter can be designed by modeling the rectifier step as an equivalent impedance load. As for the inverting step, the decomposition into resonant parameters is performed, as it allows for the normalization of the system. The differential equations that represent the circuit depend on a set of variables that, according to methodologies in the literature, depend on values estimated from figures of merit, for example. As a contribution of this work, a PSO algorithm is used here to select these values, removing the need to arbitrate values for variables and also reducing the human-machine interaction time in the design of circuit variables. Based on the data obtained, a simulation was carried out and subsequent implementation to validate the theoretical results. The practical results were satisfactory and close to the developed model.

Keywords : Optimization, converter, resonance, normalization.

LISTA DE ABREVIATURAS E SIGLAS

WPT	Transferência de Potência sem Fio (do inglês, <i>Wireless Power Transfer</i>)
ZVS	Zero de tensão na Chave (do inglês, <i>Zero-Voltage Switching</i>)
φ	Ângulo para condução do diodo retificador
\mathbf{P}	Vetor de equivalência
iL_f	Corrente de entrada do conversor
iL_s	Corrente de saída do conversor – etapa inversora
vC_p	Tensão ressonante sobre a chave
vC_s	Tensão sobre capacitor C_s
ϵ	Coeficiente de inércia
Q, Q_1	Fator de qualidade
d	Diodo
I_o	Corrente retificada na carga – Saída do retificador
C_p	Capacitor paralelo a chave ativa
L_f	Indutor de entrada do conversor
C_d	Capacitor paralelo diodo retificador
R_s	Impedância de saída da etapa inversora do conversor Classe E
C_s	Capacitância de rede ressonante de saída do inversor Classe E
L_s	Indutância de rede ressonante de saída do inversor Classe E
D_c	Razão cíclica de operação do conversor
ω	Frequência angular de operação do conversor
ω_i	Frequência i angular de ressonância
R_{dc}	Impedância de entrada do conversor
T_{pot}	Taxa de transferência de potência
a	Razão entre as impedâncias de entrada e saída do conversor, ou inverso da taxa de transferência de potência.
A_i	Frequência i de ressonância normalizada
v_i	Velocidade da partícula
r_1, r_2	Valores randômicos da distribuição uniforme

LISTA DE ABREVIATURAS E SIGLAS

D_d	Razão cíclica do diodo da etapa retificadora
$x_{e(t)}$	Transformação de equivalência
x_i	Posição atual da partícula
it	Iteração
A_N, B_N, C_N, E_N	Matrizes constantes do respectivo estágio N de operação
$\bar{A}_N, \bar{B}_N, \bar{C}_N, \bar{D}_N$	Matrizes normalizadas
P	Partícula
c_1, c_2	Coeficientes cognitivo e social

SUMÁRIO

1 INTRODUÇÃO	12
1.1 MOTIVAÇÃO E JUSTIFICATIVA.....	12
1.2 PROCEDIMENTO METODOLÓGICO.....	13
1.3 ORGANIZAÇÃO DO TRABALHO	14
1.4 CONTEXO HISTÓRICO SOBRE OTIMIZAÇÃO	14
2 CONVERSOR CLASSE E.....	16
2.1 CONVERSOR CLASSE E – ETAPA INVERSORA	16
2.2 DECOMPOSIÇÃO EM PARÂMETROS RESSONANTES	17
2.2.1 Fator de qualidade	17
2.2.2 Frequências de ressonâncias.....	19
2.2.3 Taxa de Transferência de Potência.....	20
2.2.4 Frequência normalizada	20
2.3 CONVERSOR CLASSE E – ETAPA RETIFICADORA	21
2.4 MODELO PARA O CONVERSOR COMPLETO	22
2.5 SISTEMA NORMALIZADO PARA ETAPA INVERSORA	23
2.5.1 Etapas de Operação.....	23
2.6 REPRESENTAÇÃO EM ESPAÇO DE ESTADOS	25
2.6.1 Descrição de um sistema por espaço de estados	25
2.6.2 Normalização do sistema	27
3 MÉTODOS DE OTIMIZAÇÃO	35
3.1 ASPECTOS GERAIS SOBRE O MÉTODO DE OTIMIZAÇÃO MELHOR GLOBAL	35
3.2 PSO.....	36
3.2.1 Conceitos do PSO.....	37
3.2.2 Velocidade e Posição das partículas.....	39
3.3 DESENVOLVIMENTO DO PSO PARA APLICAÇÃO NO CONVERSOR CLASSE E.....	40
3.4 APLICAÇÃO DO PSO NA DETERMINAÇÃO DE PARÂMETROS DO CONVERSOR – ETAPA INVERSORA	41
3.4.1 Parâmetros da etapa inversora	42
4 APLICAÇÃO DA METODOLOGIA	43
4.1 AVALIAÇÃO DA FUNÇÃO	43
4.2 PROJETO DA ETAPA INVERSORA.....	44
4.2.1 Resposta normalizada do sistema e determinação dos valores de projeto da etapa inversora.....	46
4.2.2 Comparação entre resultados teóricos e simulados da etapa inversora	49
4.3 PROJETO DA ETAPA RETIFICADORA	52

4.3.1 Obtenção dos parâmetros e simulação da etapa retificadora	53
4.4 IMPLEMENTAÇÃO E RESULTADOS EXPERIMENTAIS.....	57
5 CONSIDERAÇÕES FINAIS	60
REFERÊNCIAS.....	61
APÊNDICE A – Código utilizado para o PSO.....	63
APÊNDICE B – Código utilizado para a solução dos estado e obtenção dos parâmetros de projeto	71
APÊNDICE C – Principais materiais utilizados e foto da implementação	82
APÊNDICE D – Valores individuais dos parâmetros e da função objetivo para cada execução.....	83
APÊNDICE E – Gráficos das execuções de teste/resultados estatísticos.	85
ANEXO I – Fragmento do artigo <i>Resonant dc/dc Converter with Class-E Inverter ... and Class-E Rectifier</i>	88
ANEXO II – Fragmento do livro RESONANT POWER CONVERTERS.....	90

1 INTRODUÇÃO

Neste capítulo, é apresentado uma breve introdução do tema abordado nessa dissertação, fornecendo uma visão geral sobre os tópicos de cada capítulo. Estes, por sua vez, estão organizados de forma a estruturar o propósito metodológico para aplicação da técnica de otimização na solução de equações de projeto de conversores.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

O processamento de energia detém uma grande parcela da tecnologia de energia, em que, a eletrônica de potência é um ramo dedicado à análise, simulação, projeto, manufatura e aplicação em conversores (Kazin, 2011).

Além disso, a conversão ressonante está no centro de tecnologias aplicadas em energias renováveis, em telecomunicações, em linhas automotivas, em *Wireless Power Transfer* – WPT, instrumentação e indústria de consumo. O motivo da presença destes conversores que operam em ressonância se deve ao fato de que possuem características como alta eficiência, baixo ruído, entre outros.

Entretanto, os estudos e desenvolvimento sobre o processamento de energia envolvendo topologias que operam de forma ressonante já vêm sendo explorados pelo menos, desde a década de 70, como pode ser visto no artigo publicado por Schwarz (1970), em que o autor aplica as condições ressonantes para impor aos componentes, níveis moderados de estresse. Ainda no artigo, no artigo de Miller, o autor descreve o funcionamento de um circuito ressonante utilizado para o processamento de energia em aplicações de tecnologia de navegação marinha (Miller, 1976).

A complexidade da topologia do circuito ressonante contendo elevado número de componentes armazenadores de energia, como indutores e capacitores, originam mais frequências normalizadas, o que implica em mais variáveis a serem determinadas na solução do sistema formado.

Uma metodologia para a solução desses sistemas pode ser obtida pela atribuição arbitrada à alguma dessas variáveis presentes no sistema como, por

exemplo, a razão cíclica de operação, fazendo-se necessária a utilização de figuras de mérito ou ábacos.

Nesse ponto, tem-se a contribuição deste trabalho, por meio da utilização de um algoritmo de otimização aplicado na busca de valores ótimos para variáveis que em geral são arbitradas para tornar viável a solução de sistemas de equações de estados que descrevem o comportamento das topologias de conversores ressonantes. Esse fato implica que mais etapas do projeto fiquem menos dependentes da experiência do projetista. Ademais, o sistema apresenta um nível elevado de complexidade para solução analítica, e o algoritmo de otimização escolhido tem a capacidade de obter os valores ótimos, tornando-se atrativo para a solução deste tipo de problema.

Já sobre os métodos de otimização, Eberhart e Kennedy publicaram um artigo abordando um conceito baseado em enxame de partículas, que se orienta com base em melhores resultados global e individualmente.

Além dessas razões, há a motivação acadêmica para compreensão mais aprofundada da Topologia Classe E, pois este circuito possui uma estrutura física simples, mas as soluções do seu modelamento matemático não é tão trivial, o que conduziu a necessidade de envolver outras ferramentas de projeto, como métodos de otimização.

1.2 PROCEDIMENTO METODOLÓGICO

Para o desenvolvimento desta dissertação, foi adotado um procedimento baseado, primeiramente, em uma pesquisa bibliográfica sobre os principais pontos abordados (conversores ressonantes e métodos de otimização), fundamentada em livros e artigos. Após esse desenvolvimento do referencial teórico, foi modelado o conversor Classe E e feita a decomposição em termos ressonantes. A seguir, um algoritmo com base na lógica do enxame de partículas foi desenvolvido e, por fim, foram realizados os testes por simulação e prototipação de um conversor para aplicações rebaixadora entre níveis utilizados em equipamentos automotivos, aparelhos biomédicos, geração de energia, dentre outros.

1.3 ORGANIZAÇÃO DO TRABALHO

No capítulo 2 será apresentada uma visão geral sobre conversores, células ressonantes, os modos de operação e as principais características. Além disso, a abordagem normalizada pela decomposição em parâmetros ressonantes será demonstrada, aplicada ao conversor Classe E. Também, serão apresentados os equacionamentos e estruturação para representação em espaço de estados.

No capítulo 3, serão abordados os métodos de otimização. Também será realizada uma revisão sobre as principais estruturas para minimizar funções, e o desenvolvimento do algoritmo baseado em enxame de partículas. Neste ponto, o capítulo 2 se complementa, pois, o equacionamento desenvolvido para a representação em espaço de estados é solucionado com o auxílio da otimização por enxame de partículas.

No capítulo 4, é feita a aplicação da metodologia para a determinação de valores para o projeto, a comparação entre os resultados teóricos e de simulação. Além disso, é demonstrada a implementação do conversor Classe E, com o propósito de aplicações mencionadas anteriormente, na condição de rebaixar a tensão e, novamente, uma comparação entre os resultados de simulação e de implementação prática. Por fim, apresentam-se discussões sobre os resultados obtidos, as conclusões e sugestões para trabalhos futuros.

O capítulo 5 traz as considerações finais sobre o trabalho.

1.4 CONTEXO HISTÓRICO SOBRE OTIMIZAÇÃO

Singeresu (2009, p.1) define otimização como o ato de obter o melhor resultado em determinadas circunstâncias. Satybrotto (2011) destaca ainda que o mundo está cada vez mais complexo e competitivo, e em vários campos do conhecimento, os cientistas, como engenheiros e economistas por exemplo, devem tomar decisões de forma otimizada.

Logo, conforme Singeresu (2009), nos últimos anos, os métodos de otimização surgiram como poderosas e populares soluções para resolver problemas complexos de engenharia. Esses métodos incluem algoritmos genéticos, otimização de enxame

de partículas, otimização de colônias de formigas, otimização baseada em rede neural e otimização fuzzy.

O algoritmo genético foi originalmente proposto por John Holland em 1975. O algoritmo de otimização por enxame de partículas imita o comportamento de organismos sociais como uma colônia ou enxame de insetos, um bando de pássaros ou um cardume de peixe. Este algoritmo foi proposto por Eberhart em 1995 (Singeresu, 2009).

A otimização por colônia de formigas é baseada no comportamento cooperativo, em que as formigas são capazes de encontrar o caminho mais curto do seu ninho para uma fonte de alimento, apresentado por Marco Dorigo, em 1992. O método por rede neural é baseado na capacidade computacional do sistema nervoso para resolver problemas de percepção na presença de grande quantidade de dados sensoriais, por meio da sua capacidade de processamento paralelo, apresentado originalmente por Hopfield e Tank em 1985 (Singeresu, 2009).

Por fim, os métodos de otimização fuzzy foram desenvolvidos para resolver problemas envolvendo dados de projeto, função objetivo e restrições assumidas de forma imprecisa. Este método foi apresentado por Rao em 1986.

2 CONVERTOR CLASSE E

Dentre as topologias de conversores, o Classe E se destaca por ter uma eficiência extremamente alta (KAZIMIERCZUK, PUCZKO, 1987), além de possuir apenas uma chave ativa, para o caso do inversor. Ainda, é possível obter as condições de chaveamento suave, tanto em operações ZVS quanto ZCS. Em (Tomoharu et al, 2017), os autores abordaram a topologia para uma aplicação com um acoplamento indutivo, estabelecendo uma isolação entre as partes inversoras e retificadora, a fim de satisfazer critérios de segurança para aplicações em dispositivos médicos, acionamento de led's e conversores para sistemas de energias renováveis. Ainda neste artigo, foi apresentada a definição para o termo "E²" como sendo a conexão em série das etapas inversora e retificadora.

Além disso, o acoplamento indutivo permite que as estruturas das duas etapas do conversor Classe E estejam separadas por uma certa distância, compondo um sistema WPT. Esta aplicação já foi demonstrada (Nagashima et al, 2014), no qual o acoplamento indutivo para esta implementação atingiu uma distância de 30 cm entre as bobinas.

Desta forma, estes artigos demonstram um amplo campo de aplicação do conversor utilizando a topologia Classe E.

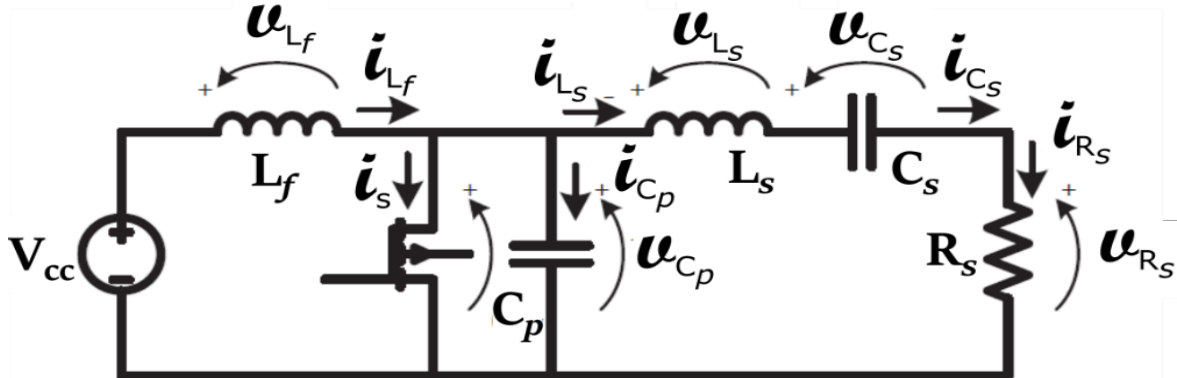
2.1 CONVERTOR CLASSE E – ETAPA INVERSORA

Para a etapa inversora, a topologia que será utilizada nesta dissertação é a apresentada na Figura 1, que possui uma chave ativa. De acordo com Kazimierczuk e Puczek (1987, p.34), esta configuração possui uma rede composta pela associação em série de uma resistência R_s , um indutor L_s e um capacitor C_s , ressonante em uma frequência abaixo da frequência de operação f , e, ainda, que toda potência ac é liberada sobre a resistência R_s .

Nesta fase, a análise assumirá que:

- 1) A chave ativa da etapa inversora é ideal, sem perdas com comutação instantânea;
- 2) Os componentes R_s , L_s , C_s , C_p e L_f são ideais, lineares e sem perdas.

Figura 1 – Circuito Classe E, etapa inversor



Fonte: (Sangoi, 2017)

2.2 DECOMPOSIÇÃO EM PARÂMETROS RESSONANTES

Para decomposição em parâmetros ressonantes, é necessário fazer e estabelecer as devidas definições sobre esses aspectos. Por isso, estas definições serão feitas com base em, por exemplo as apresentadas em artigos e bibliografias e condensadas nos tópicos seguintes.

2.2.1 Fator de qualidade

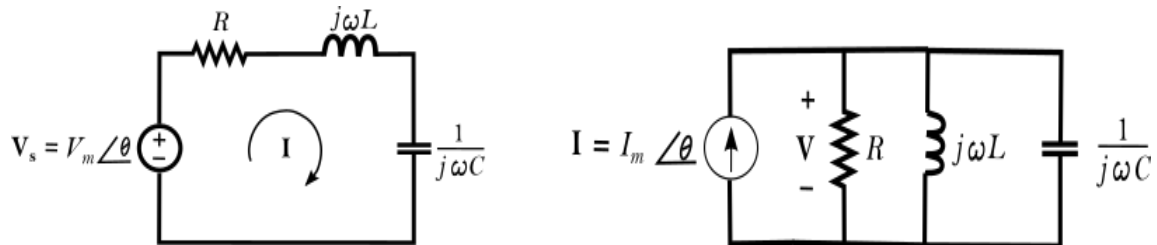
De acordo com Sadiku (2013, p.561) a “Ressonância é uma condição em um circuito RLC no qual as reatâncias capacitivas e indutivas são iguais em módulo, resultando em uma impedância puramente resistiva [...]”.

Ainda de acordo com o mesmo autor, os circuitos ressonantes podem ser série ou paralelo, conforme exemplificado na Figura 2. A impedância dos elementos depende de uma frequência ω_i , a qual pode ser chamada de frequência de ressonância quando satisfizer a condição de que a parte imaginária do circuito da referida figura for nula (Alexander e Sadiku, 2013).

Na ressonância, a energia reativa oscila entre o indutor e o capacitor em um circuito. O fator de qualidade é uma medida quantitativa que relaciona a energia máxima ou de pico armazenada e a energia dissipada no circuito, em um ciclo de operação (Alexander e Sadiku, 2013, p.561).

Em outras palavras, fator de qualidade Q pode ser a relação mostrada em (1), o qual tem uma forma conforme mostrado na Figura 3.

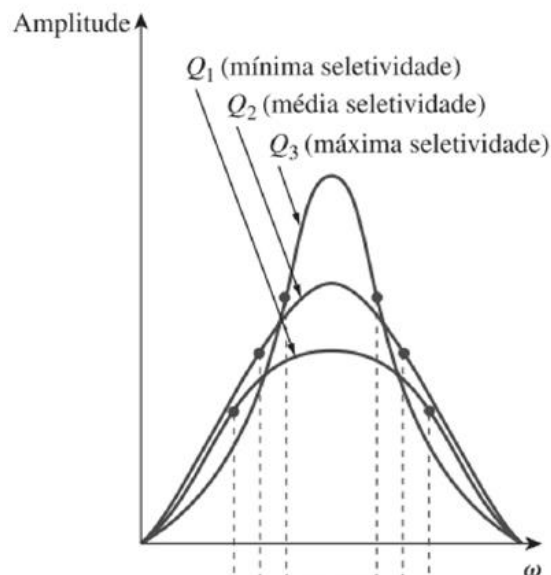
Figura 2 – Exemplos de circuito série e paralelo para composição ressonante



Fonte: (Alexande e Sadiku, 2013)

$$Q = 2\pi \frac{\text{Energia de pico armazenada no circuito}}{\text{Energia dissipada pelo circuito em um período de ressonância}} \quad (1)$$

Figura 3 – Faixa de seletividade, ou fator de qualidade de um circuito ressonante



Fonte: (Alexande e Sadiku, 2013)

Conforme Kazimierczuk e Puczeko (1987), uma alta eficiência do conversor Classe E pode ser obtida assumindo um fator infinito, ou elevado, e obtido por (2) para os circuitos série ressonante e (3) para o paralelo ressonante. Kazimierczuk e Jòzwik

(1989) estabeleceram um Q maior que 5 para obter uma corrente de saída muito próxima de uma forma senoidal, sendo que este fator Q pode ser dado também pelas expressões (2) e (3).

$$Q = \frac{\omega L}{R} = \frac{1}{\omega RC} \quad (2)$$

$$Q = \frac{R}{\omega L} = \omega CR \quad (3)$$

2.2.2 Frequências de ressonâncias

A frequência angular de ressonância ω_i é uma relação definida pela equação (4) conforme Sadiku (2013), e pode ser relacionada para qualquer combinação de componentes armazenadores de energia presente no circuito, o que é identificado pelo subíndice i .

Para o caso do circuito do Classe E mostrado na Figura 1, é possível se estabelecer três relações de ressonância entre os elementos do circuito (Bisogno, 2006), considerando a regra de que o número de frequências angulares de ressonância é igual ao número de elementos reativos menos 1. Neste caso, teremos três frequências de ressonância dadas por ω_i .

$$\omega_i = \frac{1}{\sqrt{L_i C_i}} \quad (4)$$

As combinações são escolhidas conforme as equações (5), (6) e (7).

$$\omega_1 = \frac{1}{\sqrt{L_s C_s}} \quad (5)$$

$$\omega_2 = \frac{1}{\sqrt{L_s C_p}} \quad (6)$$

$$\omega_3 = \frac{1}{\sqrt{L_f C_p}} \quad (7)$$

2.2.3 Taxa de Transferência de Potência

Conforme Bisogno (2006), a taxa de transferência de potência é obtida a partir da equação (8), e é um valor numérico necessário para a solução do sistema normalizado. Kazimierczuk e Puczko (1987) também abordam essa relação, estabelecendo-a como uma razão entre a impedância de entrada R_{dc} e a impedância de saída R_s , conforme (9). Analisando a relação estabelecida nesta última equação, a potência de saída dissipada sobre R_s reduz-se de modo inversamente proporcional a R_{dc} .

$$T_{pot} = \frac{1}{2\pi} \int_0^{2\pi} (v_{R_s}^2) d\omega t \quad (8)$$

$$a = \frac{R_{dc}}{R_s} \quad (9)$$

$$a = \frac{1}{T_{pot}} \quad (10)$$

Portanto, ainda de acordo com Bisogno, a razão entre as impedâncias de entrada e saída do conversor mostrada em (9) é o inverso da taxa de transferência de potências, conforme definido em (10).

2.2.4 Frequência normalizada

De acordo com Mendonça (2017), a relação entre a frequência de ressonância ω_i e a frequência de operação ω define a frequência de ressonância normalizada, A_i . Como nessa aplicação foram definidas três frequências em (5), (6) e (7), logo, haverá também três frequências de ressonância normalizadas, as quais estão expressas em (12), (13) e (14).

Portanto, com a introdução desta definição, o projeto do circuito começa a ficar independente dos parâmetros de projeto como, por exemplo, os valores dos componentes. Ainda com base no trabalho de Bisogno (2006), pode-se definir mais uma relação para o inverso da taxa de transferência de potência, desta vez, relacionando a tensão de entrada, a corrente de entrada e a resistência de saída, conforme mostrado em (11).

$$a = \frac{V_{cc}}{R_S I_{cc}} \quad (11)$$

$$A_1 = \frac{\omega_1}{\omega} \quad (12)$$

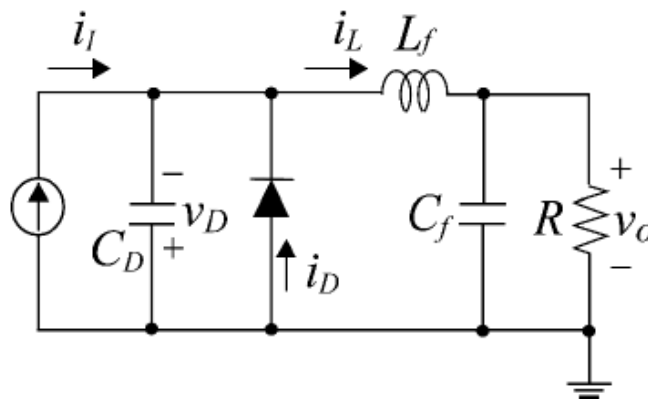
$$A_2 = \frac{\omega_2}{\omega} \quad (13)$$

$$A_3 = \frac{\omega_3}{\omega} \quad (14)$$

2.3 CONVERSOR CLASSE E – ETAPA RETIFICADORA

A etapa retificadora, ou o retificador Classe E é mostrado na Figura 4. Esta topologia, de acordo com Kazimierczuk e Czarkowski (2011), é a homóloga à topologia do inversor Classe E.

Figura 4 – Topologia básica de um retificador Classe E



Fonte: (Asiya, *et al*, 2019)

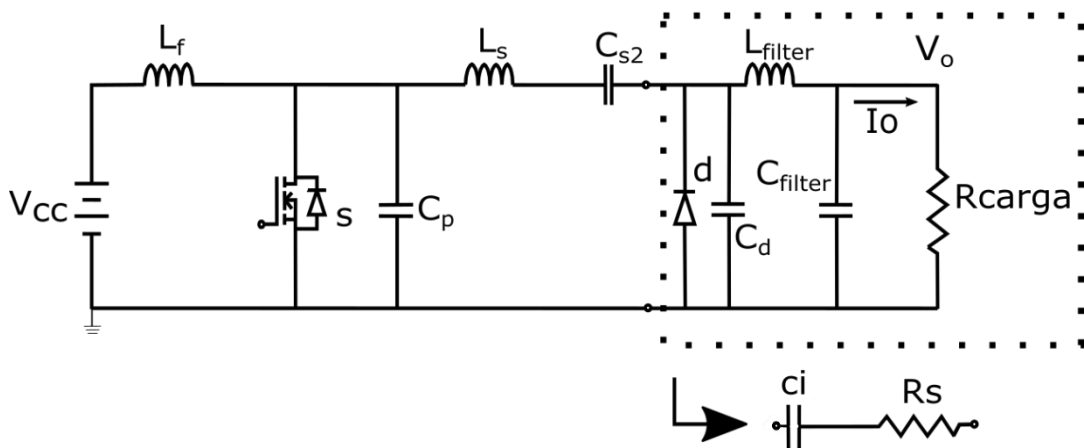
Conforme pode ser analisado da Figura 4, esta topologia apresenta um diodo como chave com um capacitor paralelo e um filtro passa-baixa de segunda ordem na saída. A chave passiva deste retificador é controlada pelo sinal senoidal, que é a forma de onda da fonte de corrente do exemplo da figura acima.

O adequado projeto do capacitor paralelo à chave deste circuito e do filtro de saída possibilita que o diodo também opere em ressonância. A corrente que fluirá pelo indutor do filtro de saída deverá ser essencialmente contínua, e também, ocorrerá uma razão cíclica D_d .

2.4 MODELO PARA O CONVERSOR COMPLETO

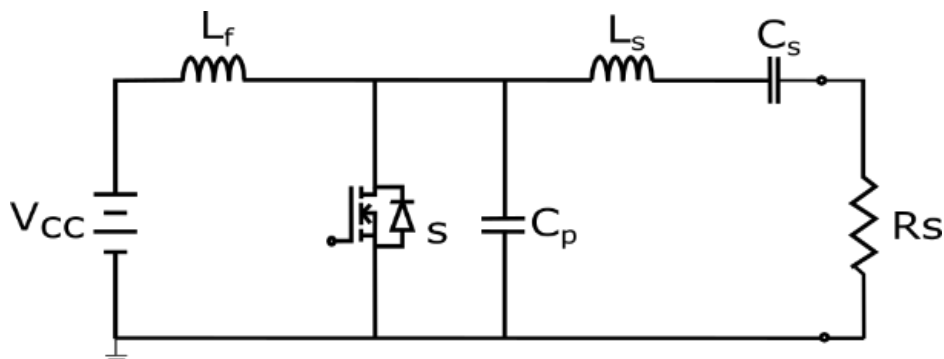
O modelo proposto para aplicação da otimização se baseia na topologia apresentada por Kazimierczuk e Jozwik (1989), conforme apresentado na Figura 5. Neste modelo, a etapa retificadora é substituída por uma impedância equivalente, formada por uma resistência e capacitância em série, como é demonstrado na Figura 6.

Figura 5 – Topologia básica do conversor Classe E quadrático



Fonte: (Kazimierczuk e Jozwik, 1989). Adaptado pelo autor.

Figura 6 – Circuito equivalente para o conversor Classe E



Fonte: (Autor, 2021)

A capacitância da impedância resultante da etapa retificadora fica em série com a capacitância da etapa inversora, o que forma uma capacitância equivalente. Desta forma, o circuito da Figura 6 fica semelhante ao da Figura 1.

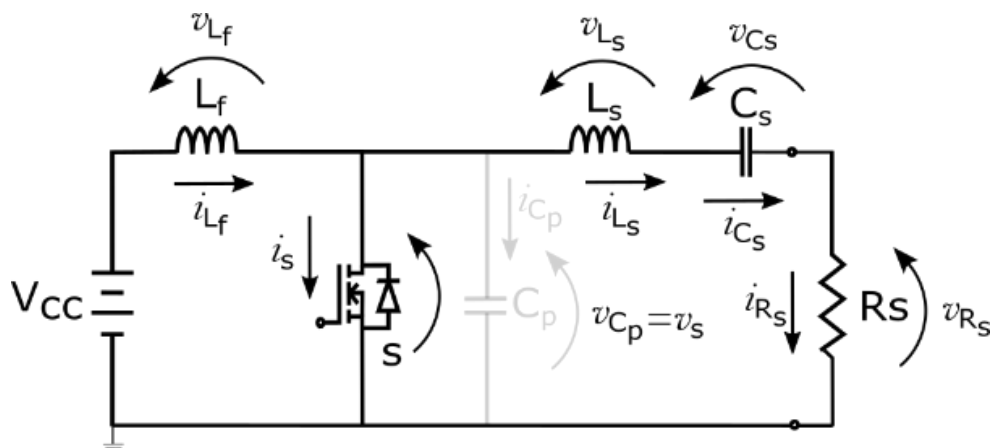
2.5 SISTEMA NORMALIZADO PARA ETAPA INVERSORA

De posse das definições iniciais sobre normalização das variáveis do sistema, pode-se iniciar os equacionamentos do sistema da etapa inversora.

2.5.1 Etapas de Operação

Considere o circuito da Figura 1, que apresenta uma chave ativa, que comuta entre os estados de condução e bloqueio. Primeiramente, considerando o estágio de operação ($0 \leq \omega t < 2\pi D_c$) em que a chave está conduzindo, tem-se a topologia conforme a Figura 7. Durante este intervalo, aplicando as análises de circuito, obtêm-se as equações diferenciais mostradas em (15), (16), (17) e (18).

Figura 7 – Circuito equivalente para o primeiro estágio de operação do conversor Classe E – Etapa Inversora



Fonte: (Autor, 2021)

$$L_f \frac{d}{dt} i_{L_f(t)} = V_{cc} \quad (15)$$

$$L_s \frac{d}{dt} i_{L_s(t)} = -R_s i_{L_s(t)} - v_{C_s(t)} \quad (16)$$

$$C_s \frac{d}{dt} v_{C_s(t)} = i_{L_s(t)} \quad (17)$$

$$C_p \frac{d}{dt} v_{C_p(t)} = 0 \quad (18)$$

Considerando o segundo estágio de operação ($2\pi D_c \leq \omega t < \omega t$), período este em que a chave está em bloqueio, o circuito equivalente torna-se conforme está demonstrado na Figura 8. Nesta etapa, a análise deste circuito fornece as equações diferenciais (19), (20), (21) e (22).

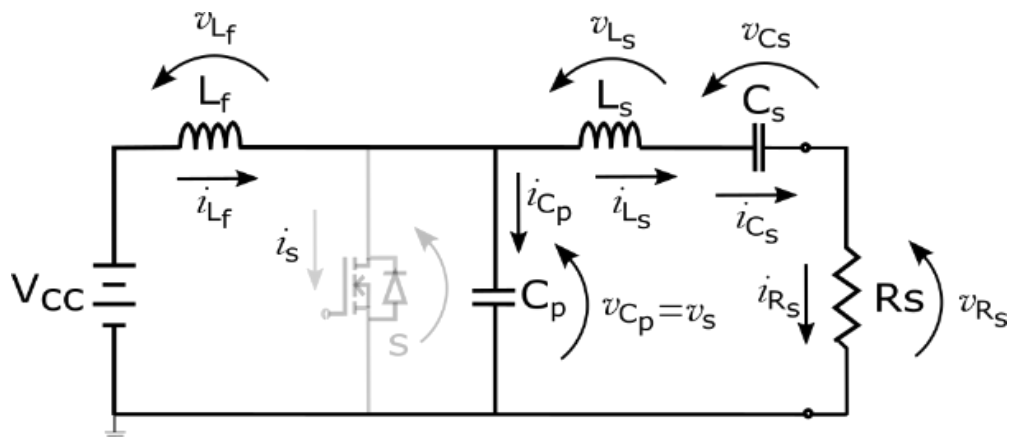
$$L_f \frac{d}{dt} i_{L_f(t)} = -v_{C_p(t)} + V_{CC} \quad (19)$$

$$L_s \frac{d}{dt} i_{L_s(t)} = -R_s i_{L_s(t)} - v_{C_s(t)} + v_{C_p(t)} \quad (20)$$

$$C_s \frac{d}{dt} v_{C_s(t)} = i_{L_s(t)} \quad (21)$$

$$C_p \frac{d}{dt} v_{C_p(t)} = i_{L_f(t)} - i_{L_s(t)} \quad (22)$$

Figura 8 – Circuito equivalente para o segundo estágio de operação do conversor Classe E – Etapa Inversora



2.6 REPRESENTAÇÃO EM ESPAÇO DE ESTADOS

Com as equações diferenciais escritas nas formas matriciais em (23) e (24), pode-se obter a solução do sistema em espaço de estados. Entretanto, é necessário obtê-lo de forma normalizada.

$$\frac{d}{dt} \begin{bmatrix} iL_f(t) \\ iL_s(t) \\ vC_s(t) \\ vC_p(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & -\frac{1}{L_s} & 0 \\ 0 & \frac{1}{C_s} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{Matriz } A_I} \cdot \underbrace{\begin{bmatrix} iL_f(t) \\ iL_s(t) \\ vC_s(t) \\ vC_p(t) \end{bmatrix}}_{\text{Matriz } B_I} + \begin{bmatrix} \frac{1}{L_f} \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot V_{cc} \quad (23)$$

$$\frac{d}{dt} \begin{bmatrix} iL_f(t) \\ iL_s(t) \\ vC_s(t) \\ vC_p(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & -\frac{1}{L_f} \\ 0 & -\frac{R_s}{L_s} & -\frac{1}{L_s} & \frac{1}{L_s} \\ 0 & \frac{1}{C_s} & 0 & 0 \\ \frac{1}{C_p} & -\frac{1}{C_p} & 0 & 0 \end{bmatrix}}_{\text{Matriz } A_{II}} \cdot \underbrace{\begin{bmatrix} iL_f(t) \\ iL_s(t) \\ vC_s(t) \\ vC_p(t) \end{bmatrix}}_{\text{Matriz } B_{II}} + \begin{bmatrix} \frac{1}{L_f} \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot V_{cc} \quad (24)$$

2.6.1 Descrição de um sistema por espaço de estados

De acordo Chen (1998), um sistema linear invariante no tempo pode ser escrito na forma mostrada em (25) e (26), em que A , B , C e E são matrizes constantes. Neste caso, a questão é encontrar uma solução, ou uma resposta $x(t)$ a uma entrada $u(t)$, contendo certas condições iniciais $x(0)$.

As soluções $x(t)$ e $y(t)$ são obtidas de acordo com (27) e (28).

$$\dot{x}_{(t)} = \mathbf{A}x_{(t)} + \mathbf{B}u_{(t)} \quad (25)$$

$$y_{(t)} = \mathbf{C}x_{(t)} + \mathbf{E}u_{(t)} \quad (26)$$

$$x_{(t)} = e^{\mathbf{A}t}x_{(0)} + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u_{(\tau)}d\tau \quad (27)$$

$$y_{(t)} = \mathbf{C}e^{\mathbf{A}t}x_{(0)} + \mathbf{C} \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u_{(\tau)}d\tau + \mathbf{E}u_{(\tau)} \quad (28)$$

Contudo, Chen (1998) ainda demonstra que o sistema pode ser representado por uma transformação de equivalência, a partir de um vetor, ou uma matriz \mathbf{P} , real e não singular, na forma dada em (29). Nesta equação, $x_{e(t)}$ é chamado de transformação de equivalência.

$$\mathbf{P} = x_{e(t)} x_{(t)}^{-1} \quad (29)$$

A partir deste vetor \mathbf{P} , as matrizes constantes das equações (25) e (26) podem ser modificadas de acordo com (30), (31), (32) e (33), ficando este sistema algebricamente equivalente a (34) e (35).

Nesta etapa, o sistema ainda está em função da variável t (tempo), conforme pode ser verificado no subíndice de (27) e (29). Porém, é necessário para a normalização do sistema, que ele fique em função de ωt , ou seja, de um período ou ciclo.

Para tornar o sistema dependente somente do período ωt , pode-se fazer uma modificação multiplicando o operador da derivada do vetor de estados do modelo em espaço de estados por ω . Ou seja:

A multiplicação de matrizes de estados e de entrada, de um modelo de espaço de estados genéricos...por um operador linear, modifica a variável independente do operador derivada e do argumento da função relacionada, proporcionalmente. (MENDONÇA, 2017, p. 47).

Logo, de acordo com o proposto por Mendonça e com a transformação de equivalência de Chen, o sistema passa a ser conforme segue.

$$\bar{\mathbf{A}} = \frac{1}{\omega} \mathbf{PAP}^{-1} \quad (30)$$

$$\bar{\mathbf{B}} = \frac{1}{\omega} \mathbf{PB} \quad (31)$$

$$\bar{\mathbf{C}} = \mathbf{CP}^{-1} \quad (32)$$

$$\bar{\mathbf{E}} = \mathbf{E} \quad (33)$$

$$\dot{\bar{\mathbf{x}}}_{(\omega t)} = \bar{\mathbf{A}} \bar{\mathbf{x}}_{(\omega t)} + \bar{\mathbf{B}} \bar{\mathbf{u}}_{(\omega t)} \quad (34)$$

$$\bar{\mathbf{y}}_{(\omega t)} = \bar{\mathbf{C}} \bar{\mathbf{x}}_{(\omega t)} + \bar{\mathbf{E}} \bar{\mathbf{u}}_{(\omega t)} \quad (35)$$

2.6.2 Normalização do sistema

Agora, algumas definições são assumidas para concluir a normalização do sistema, tais como:

- O vetor de estados fica definido como sendo as correntes nos indutores L_s e L_f , e as tensões sobre os capacitores C_p e C_s ; e
- O vetor de saída do sistema é formado pelas próprias variáveis de estados. Desta forma, as matrizes \mathbf{C}_I e \mathbf{C}_{II} se tornam apenas matrizes identidades, e as matrizes \mathbf{E}_I e \mathbf{E}_{II} ficam nulas.

A partir do vetor de transformação $x_{e(t)}$, que será definido de acordo com o proposto por Mendonça (2017), sendo este vetor constituído por termos compostos pelas raízes quadradas da energia armazenada nos elementos reativos, conforme (36), formando-se novas matrizes.

$$\mathbf{x}_{e(t)} = \begin{bmatrix} \frac{\sqrt{2}\sqrt{L_f}}{2} \dot{i}_{L_f(t)} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}\sqrt{L_s}}{2} \dot{i}_{L_s(t)} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}\sqrt{C_s}}{2} v_{C_s(t)} & 0 \\ 0 & 0 & 0 & \frac{\sqrt{2}\sqrt{C_p}}{2} v_{C_p(t)} \end{bmatrix} \quad (36)$$

$$\overline{\mathbf{A}}_I = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{-R_s}{\omega L_s} & \frac{-1}{\omega \sqrt{C_s} \sqrt{L_s}} & 0 \\ 0 & \frac{1}{\omega \sqrt{L_s} \sqrt{C_s}} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (37)$$

$$\overline{\mathbf{B}}_I = \begin{bmatrix} -\frac{\sqrt{2}}{2\omega \sqrt{L_f}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (38)$$

$$\overline{\mathbf{A}}_{II} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{\omega \sqrt{L_f} \sqrt{C_p}} \\ 0 & \frac{-R_s}{\omega L_s} & \frac{-1}{\omega \sqrt{C_s} \sqrt{L_s}} & \frac{1}{\omega \sqrt{L_s} \sqrt{C_p}} \\ 0 & \frac{1}{\omega \sqrt{L_s} \sqrt{C_s}} & 0 & 0 \\ \frac{1}{\omega \sqrt{C_p} \sqrt{L_f}} & \frac{1}{\omega \sqrt{L_s} \sqrt{C_p}} & 0 & 0 \end{bmatrix} \quad (39)$$

$$\overline{\mathbf{B}}_{II} = \begin{bmatrix} -\frac{\sqrt{2}}{2\omega \sqrt{L_f}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (40)$$

$$\bar{C}_I = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{L_f}} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{\sqrt{L_s}} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{\sqrt{C_s}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (41)$$

$$\bar{C}_{II} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{L_f}} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{\sqrt{L_s}} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{\sqrt{C_s}} & 0 \\ 0 & 0 & 0 & \frac{\sqrt{2}}{\sqrt{C_p}} \end{bmatrix} \quad (42)$$

Com isso, as matrizes A_I , B_I , A_{II} , B_{II} das equações (23) e (24), após as operações descritas de (30) a (33), ficam conforme (37), (38), (39) e (40), ao passo que as matrizes C_I e C_{II} ficam conforme (41) e (42).

Neste ponto, se observa que o primeiro elemento do vetor B_I e do vetor B_{II} é diferente de 1. Logo, deve incluir na equação (29) um termo que torne todos os elementos não nulos desses vetores iguais a 1.

A adequação desenvolvida no método de normalização proposto, é a utilização de um fator de correção K. Este fator deve ser equacionado de forma que o termo diferente de 0 ou 1 se torne 1. Além disso, o fator K é equacionado de forma a normalizar a sistema em relação a entrada através da multiplicação pela fonte de entrada... (MENDONÇA, 2017, p.48)

Desta forma, a equação (29) passa a ser, considerando este fator, conforme a equação (43). Isso implica que as matrizes \bar{A}_I , \bar{A}_{II} , \bar{B}_I , \bar{B}_{II} obtidas em (37), (38), (39),

(40), (41) e (42), respectivamente, também sejam modificadas, passando a ser como expostas em (44), (45), (46), (47), (49) e (50), considerando que o fator de correção K seja conforme (48)

$$\mathbf{P} = K (x_{e(t)} x_{(t)}^{-1}) \quad (43)$$

$$\bar{\mathbf{A}}_I = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{R_S}{\omega L_S} & -\frac{1}{\omega \sqrt{C_S} \sqrt{L_S}} & 0 \\ 0 & \frac{1}{\omega \sqrt{C_S} \sqrt{L_S}} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (44)$$

$$\bar{\mathbf{B}}_I = \begin{bmatrix} \frac{1}{V_{CC}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (45)$$

$$\bar{\mathbf{A}}_{II} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{\omega \sqrt{C_P} \sqrt{L_f}} \\ 0 & -\frac{R_S}{\omega L_S} & -\frac{1}{\omega \sqrt{C_S} \sqrt{L_S}} & \frac{1}{\omega \sqrt{C_P} \sqrt{L_S}} \\ 0 & \frac{1}{\omega \sqrt{C_S} \sqrt{L_S}} & 0 & 0 \\ \frac{1}{\omega \sqrt{C_P} \sqrt{L_f}} & -\frac{1}{\omega \sqrt{C_P} \sqrt{L_S}} & 0 & 0 \end{bmatrix} \quad (46)$$

$$\bar{\mathbf{B}}_{II} = \begin{bmatrix} \frac{1}{V_{CC}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (47)$$

$$K = \frac{\sqrt{2}}{2\sqrt{L_f}\omega} \quad (48)$$

$$\bar{C}_I = \begin{bmatrix} \frac{V_{CC}}{L_f \omega} & 0 & 0 & 0 \\ 0 & \frac{V_{CC}}{\sqrt{L_f} \sqrt{L_s} \omega} & 0 & 0 \\ 0 & 0 & \frac{V_{CC}}{\sqrt{C_s} \sqrt{L_f} \omega} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (49)$$

$$\bar{C}_{II} = \begin{bmatrix} \frac{V_{CC}}{L_f \omega} & 0 & 0 & 0 \\ 0 & \frac{V_{CC}}{\sqrt{L_f} \sqrt{L_s} \omega} & 0 & 0 \\ 0 & 0 & \frac{V_{CC}}{\sqrt{C_s} \sqrt{L_f} \omega} & 0 \\ 0 & 0 & 0 & \frac{V_{CC}}{\sqrt{C_p} \sqrt{L_f} \omega} \end{bmatrix} \quad (50)$$

Após a transformação de equivalência, pode-se aplicar a decomposição em parâmetros ressonantes, aplicando-se alguns arranjos algébricos conforme mostrados nas equações subsequentes. Q_1 é o próprio fator de qualidade.

$$\frac{-R_s}{L_s \omega_1} = \frac{-R_s \omega_1}{L_s \omega \omega_1} = \frac{-A_1}{Q_1} \quad (51)$$

$$\frac{-1}{\sqrt{C_s} \sqrt{L_s} \omega} = \frac{-\omega_1}{\omega} = -A_1 \quad (52)$$

$$\frac{1}{\sqrt{C_s} \sqrt{L_s} \omega} = \frac{\omega_1}{\omega} = A_1 \quad (53)$$

$$\frac{-1}{\sqrt{C_p} \sqrt{L_f} \omega} = \frac{-\omega_3}{\omega} = -A_3 \quad (54)$$

$$\frac{-1}{\sqrt{C_p} \sqrt{L_s} \omega} = \frac{-\omega_2}{\omega} = -A_2 \quad (55)$$

$$\frac{1}{\sqrt{C_p}\sqrt{L_f}\omega} = \frac{\omega_3}{\omega} = A_3 \quad (56)$$

Considerando o circuito equivalente da Figura 7 e o fato de que o conversor opera em ressonância, tem-se por consequência a condição ótima (Bisogno, 2006), além de simplificar a solução dos estados. Esta dedução advém do que foi assumido, de que a condição inicial da corrente do indutor de entrada L_f é igual a condição inicial da corrente do indutor L_s . Portanto, a condição ótima pode ser relacionada conforme a equação (57).

Este conversor opera em ressonância, impondo em ZVS, ficando conforme a equação (58), em que a condição inicial, ou final da tensão sobre o capacitor C_p seja nula. Considere que a tensão sobre este capacitor é a mesma sobre a chave ativa, pois estão em paralelo.

$$\frac{L_f \omega i_{L_f}}{V_{cc}} = \frac{\sqrt{L_f} \sqrt{L_s} i_{L_s}}{V_{cc}} \quad (57)$$

$$v_{C_s(t)} = 0 \quad (58)$$

Desenvolvendo a equação (53), obtém-se a relação entre as correntes iniciais de i_{L_f} e i_{L_s} conforme (59).

$$i_{L_f} = \frac{A_2 i_{L_s}}{A_3} \quad (59)$$

Com essas relações definidas, as matrizes \bar{A}_I , \bar{A}_{II} , \bar{B}_I , \bar{B}_{II} , \bar{C}_I , \bar{C}_{II} , \bar{E}_I e \bar{E}_{II} podem ser representadas completamente por parâmetros ressonantes. Estas matrizes redefinidas estão descritas abaixo.

$$\bar{A}_I = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{A_1}{Q_1} & -A_1 & 0 \\ 0 & A_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (60)$$

$$\bar{\mathbf{B}}_I = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (61)$$

$$\bar{\mathbf{C}}_I = \begin{bmatrix} \frac{A_3^2 A_1 a}{A_2^2 Q_1} & 0 & 0 & 0 \\ 0 & \frac{A_1 A_3 a}{A_2 Q_1} & 0 & 0 \\ 0 & 0 & \frac{A_1 A_3}{A_2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (62)$$

$$\bar{\mathbf{A}}_{II} = \begin{bmatrix} 0 & 0 & 0 & -A_3 \\ 0 & -\frac{A_1}{Q_1} & -A_1 & A_2 \\ 0 & A_1 & 0 & 0 \\ A_3 & -A_2 & 0 & 0 \end{bmatrix} \quad (63)$$

$$\bar{\mathbf{B}}_{II} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (64)$$

$$\bar{\mathbf{C}}_{II} = \begin{bmatrix} \frac{A_3^2 A_1 a}{A_2^2 Q_1} & 0 & 0 & 0 \\ 0 & \frac{A_1 A_3 a}{A_2 Q_1} & 0 & 0 \\ 0 & 0 & \frac{A_1 A_3}{A_2} & 0 \\ 0 & 0 & 0 & A_3 \end{bmatrix} \quad (65)$$

Com isto, a etapa de modelagem matemática do conversor fica completa, podendo-se partir para solução do sistema. Neste ponto, faz-se necessário a aplicação do PSO como ferramenta auxiliar para obtenção desta solução, devido sistema ser formado por um conjunto de equações que são dependentes de 9 variáveis, ou seja, $\dot{x}(t) = \bar{A}_n [A_1, A_2, A_3, Q_1, D_c, i_{L_f(0)}, i_{L_s(0)}, v_{C_s(0)}, v_{C_p(0)}]$. Desta forma, o PSO irá determinar os valores de A_3, Q_1, D_c .

3 MÉTODOS DE OTIMIZAÇÃO

Este capítulo abordará os métodos de otimização, fazendo uma revisão sobre este assunto e demonstrando o algoritmo desenvolvido para obtenção de valores ótimos para os parâmetros ressonantes desejados.

3.1 ASPECTOS GERAIS SOBRE O MÉTODO DE OTIMIZAÇÃO MELHOR GLOBAL

Alguns aspectos gerais devem ser considerados para o desenvolvimento do algoritmo de otimização. Por exemplo, segundo Talukder (2011, p.5) muitos problemas de otimização requerem que algumas variáveis satisfaçam certas condições ou limitações, como ter valores positivos. Nesse caso, são chamados de problemas de otimização restrita. Em contrapartida, ainda segundo este autor, outros problemas não possuem ou não são impostas restrições aos valores que podem ser atribuídos às variáveis do problema, ou seja, são ditos problemas de otimização irrestrita.

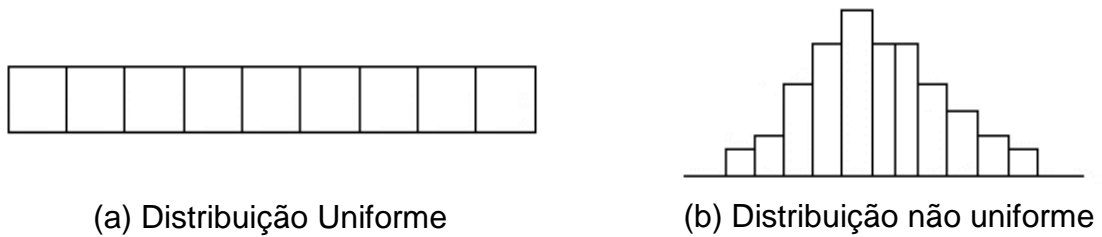
Entretanto em alguns outros problemas de otimização têm funções objetivos que mudam ao longo do tempo, que em consequência ocasionam a mudança da solução ótima para aquelas circunstâncias. Nesses casos, são ditos problemas de otimização dinâmica.

Outro aspecto relevante é acerca da distribuição uniforme, conforme demonstrado na Figura 9(a). Também chamada de distribuição retangular, é uma distribuição onde a probabilidade de ocorrência é a mesma para todos os valores, ou seja, tem uma probabilidade constante. Portanto, seja definido $U(a, b)$, em que a e b são o máximo e o mínimo valor, respectivamente, de um conjunto de valores.

Portanto, se uma distribuição uniforme é dividida em intervalos igualmente espaçados, há um número igual de membros do conjunto de valores em cada intervalo. Não havendo quaisquer dessas uniformidades, é dito uma distribuição não uniforme, conforme Figura 9(b) (Talukder, 2011).

Ainda sobre esse ponto, Talukder ainda apresenta as definições sobre as funções densidade de probabilidade e distribuição cumulativa, conforme demonstradas nas equações (66) e (67), respectivamente, para um intervalo $[a, b]$.

Figura 9 – Exemplo de distribuição uniforme em (a) e não uniforme em (b)



Fonte: (Talukder, 2011)

$$f(x) = \begin{cases} 0 & , \text{para } x < a \\ \frac{1}{b-a} & , \text{para } a \leq x \leq b \\ 0 & , \text{para } x > b \end{cases} \quad (66)$$

$$f(x) = \begin{cases} 0 & , \text{para } x < a \\ \frac{x-a}{b-a} & , \text{para } a \leq x \leq b \\ 1 & , \text{para } x > b \end{cases} \quad (67)$$

Se o conjunto, ou intervalo, a e b for compreendido entre 0 e 1, como consequência, passa-se a ter $U(0,1)$, o qual é chamado de distribuição uniforme padrão.

3.2 PSO

Em (Eberhart e Kennedy, 1995), os autores apresentam o método de otimização por enxame orientado global e localmente. De um modo geral, este método compreende um conceito simples e de fácil implementação em um algoritmo, que requer apenas uma matemática mais elementar, apenas somas e multiplicação e isso implica em menos consumo de processamento computacional.

O método de otimização PSO possui memória, ou seja, cada partícula guarda em sua memória o melhor resultado que já obteve, e isso causa influência em seu deslocamento visando o ponto de ótimo global.

3.2.1 Conceitos do PSO

O PSO é um método de otimização baseado em um comportamento social visto em algumas espécies de animais como revoadas, cardumes etc. A inspiração do algoritmo vem da observação de que o grupo, ou o bando, compartilha o conhecimento individual na busca de alimentos, por exemplo, fazendo como que os demais membros sigam àqueles que forem descobrindo os melhores caminhos.

Conforme Eberhart e Kennedy (1995), o PSO é inicializado com uma população de soluções aleatórias, as partículas, em que cada solução em potencial é atribuída uma posição e uma velocidade aleatória. Cada solução em potencial é uma partícula dessa população, chamada enxame. Elas se movimentam em um espaço de busca visando encontrar o ponto de ótimo global.

Cada partícula se movimento no espaço de busca orientando-se pela memória do melhor resultado que já obteve. Este valor é denominado de melhor individual, ou *pbest*. Esse valor é armazenado na memória dessa partícula, sendo este chamado de melhor individual, *pbest*. O valor de *pbest* é calculado de forma individual para todas as partículas, e a informação sobre o melhor entre todos os valores de *pbest* é compartilhado por todo o enxame, denominado de melhor global, ou *gbest*.

Portanto, a ideia central do PSO consiste em que a cada iteração, a velocidade de deslocamento de cada partícula varie e que ela se desloque considerando as direções ao melhor global e local.

Desta forma, a estrutura apresentada por Eberhart e Kennedy (1995) para o PSO é conforme as etapas abaixo:

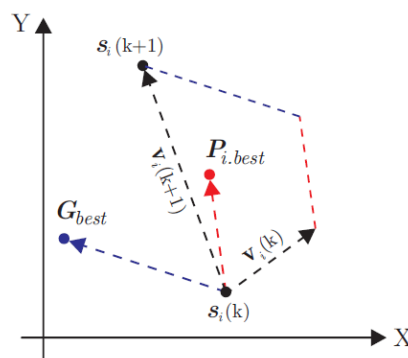
- Inicializar um vetor com N dimensões, e com as partículas em posições aleatórias;
- Avaliar a função a ser minimizada com as variáveis do vetor inicializado;
- Avaliar o valor da função a ser minimizada, comparando o seu valor na iteração atual com o seu valor na iteração anterior, e definindo os melhores valores locais, por meio de uma comparação lógica;
- Avaliar o valor da função a ser minimizada, comparando o seu valor na iteração atual com o seu valor na iteração anterior para definir o melhor valor global, por meio de uma comparação lógica;

- Atualizar o valor da velocidade de deslocamento da partícula;
- Atualizar a posição da partícula; e
- Testar a condição ou critério de parada.

Esta versão é dita PSO do melhor global. Outro detalhe sobre o PSO é acerca das definições de função objetivo e *fitness*. De acordo com Borin (2020, p.20), a função objetivo é uma relação de pontos no espaço de busca com uma medida de qualidade, enquanto a função *fitness* é referente à própria função objetivo associada à posição da partícula. Neste caso, a equação (72) representa a função objetivo.

Portanto, o deslocamento das partículas será guiado para encontrar a melhor posição em que ocorra a minimização dessa função objetivo. Inicialmente, as partículas são distribuídas dentro do espaço de busca, de forma aleatória e começam a se deslocar a uma certa velocidade v_i conforme equação (70) pelas posições x_i , que no caso do conversor Classe E em análise nesta dissertação, é um espaço tridimensional por possuir três variáveis a serem determinadas. A Figura 10 demonstra a evolução das partículas em um espaço de busca de duas dimensões (bidimensional).

Figura 10 – Deslocamento de uma partícula s em um espaço de busca bidimensional



Fonte: (Borin, 2021)

Em relação a este trabalho, um destaque importante se faz necessário ao fato de que a função objetivo busca a otimização com base na solução simultânea de 4 funções em um espaço de busca em tridimensional.

Tendo em vista em peculiaridade, também se justifica a escolha desse caminho de solução meta-heurística em vez de uma solução analítica.

3.2.2 Velocidade e Posição das partículas

O PSO melhor global é uma versão em que cada partícula é influenciada pela partícula de melhor ajuste em todo o enxame. Neste método, cada partícula individual i com $i \in [1, 2, \dots, n]$, com $n > 1$, tem sua posição atual x_i , sua velocidade atual v_i e sua melhor posição pessoal no espaço de busca $P_{best,i}$. Essa melhor posição pessoal ou local corresponde a posição no espaço de busca onde a partícula obteve o menor valor para a *função objetivo* f . O melhor $P_{best,i}$ entre todos gera o G_{best} . De acordo com Talukder (2011), considerando que o objetivo seja minimizar uma função e que os valores $P_{best,i}$ e G_{best} sejam os atualizados, resultantes da iteração atual it , na próxima iteração $it + 1$ irá resultar em:

$$P_{best,i}^{it+1} = \begin{cases} P_{best,i}^{it} & , se f(x_i^{it+1}) > P_{best,i}^{it} \\ x_i^{it+1} & , se f(x_i^{it+1}) \leq P_{best,i}^{it} \end{cases} \quad (68)$$

$$G_{best} = menor\{P_{best,i}^{it}\} \quad (69)$$

Com base em $P_{best,i}$ e G_{best} a velocidade da partícula v_i é atualizada conforme a equação (70) e a posição da partícula pela equação (71), ambas dadas por Talukder (2011) nas seguintes formas:

$$v_i^{it+1} = \epsilon v_i^{it} + c_1 r_1^{it} [P_{best,i}^{it} - x_i^{it}] + c_2 r_2^{it} [G_{best} - x_i^{it}] \quad (70)$$

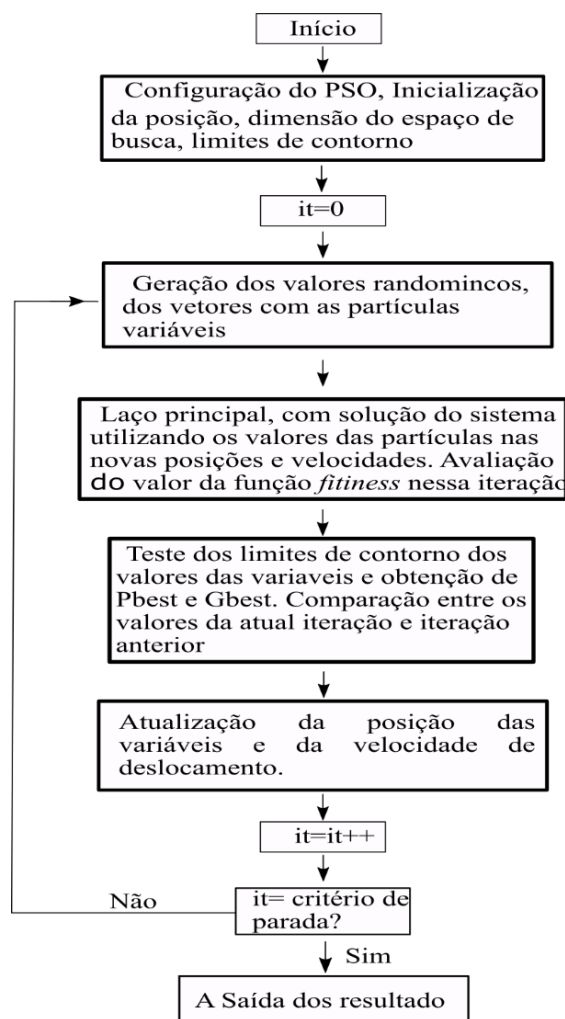
$$x_i^{it+1} = x_i^{it} + v_i^{it+1} \quad (71)$$

Nota-se, pela equação (70), que a velocidade de cada partícula pode ser influenciada por dois termos c_1 , e c_2 , os quais podem ser arbitrados valores para darem mais pesos à parcela individual $P_{best,i}$ ou coletiva G_{best} . Esses termos são constantes chamadas de coeficiente cognitivo e social, respectivamente. Além desses termos, o movimento aleatório é gerado por r_1 e r_2 em uma distribuição uniforme no intervalo $[0;1]$ (Borin, 2020).

O coeficiente de inércia ϵ é um termo para a desaceleração da partícula (Eberhart, 1999) para evitar que saia do espaço de busca e fique nas vizinhanças do melhor resultado encontrado.

3.3 DESENVOLVIMENTO DO PSO PARA APLICAÇÃO NO CONVERSOR CLASSE E

Figura 11 – Fluxo do PSO melhor global desenvolvido e utilizado neste trabalho



Fonte: (Autor, 2021)

Tendo em vistas as matrizes obtidas e escritas em (60)-(65), pode-se aplicá-las nas equações de estados. Entretanto, O PSO será aplicado para a obtenção de

valores ótimos de 3 incógnitas do sistema formado, a saber, Q_1 , A_3 e D_c . As incógnitas restantes são obtidas pela solução das equações (34) e (35).

Desse modo, o algoritmo segue a lógica representada na Figura 11, iniciando pela configuração, ou parâmetros para definição do espaço de busca do PSO. Neste caso, os valores são conforme mostrados na Tabela 1.

Tabela 1 – Parâmetros do PSO

Parâmetro	Descrição	Valor Mínimo	Valor máximo
D_c	Razão cíclica	0,3	0,8
Q_1	Fator de qualidade	5	100
A_3	Frequencia angular de ressonância normalizada	1	1,7
ϵ	Coefficiente de inércia, para desaceleração das partículas	0,4	0,9
c_1	Peso do coeficiente pessoal	0,5	0,5
c_2	Peso do coeficiente global	0,5	0,5
it	Número de iterações	30	30
P	Número de partículas	30	30

Fonte: (Autor, 2021)

O PSO melhor global escolhido para este trabalho é devido intenção de se obter uma otimização mais rápida e com menos gastos de processamento computacional, pois a inclusão de outros critérios de escolha ou de orientação para o deslocamento das partículas pode implicar em mais necessidade de processamento.

3.4 APLICAÇÃO DO PSO NA DETERMINAÇÃO DE PARÂMETROS DO CONVERSOR – ETAPA INVERSORA

O espaço de busca fica limitado pelos valores que determinam os máximos e mínimos valores para as incógnitas Q_1 , A_3 e D_c .

O espaço de busca foi definido com base em valores típicos encontrados na literatura disponível para o caso de A_3 . Para D_c , são os valores decimais possíveis que estão compreendidos entre 0 e 1. Para o fator de qualidade Q_1 , o limite inferior é

estabelecido com base na afirmação de Kazimierczuk (1987) para que a forma de onda da corrente seja senoidal, e o limite superior ficou limitado ao valor numérico de 100, pois impacta em uma penalização à calibração da função objetivo, por aumentar o tamanho do espaço de busca para o deslocamento das partículas, o que irá requerer em mais tempo de processamento.

3.4.1 Parâmetros da etapa inversora

A aplicação do conversor apresentado na Figura 5 se dará para rebaixar o nível de tensão V_{cc} para V_o , ou seja, 12V para 5V. As características de operação estão descritas na Tabela 2. Valores nesta ordem de grandeza utilizadas em aplicações automotivas, em aparelhos eletrônicos de comunicação móvel, equipamentos de áudio, entre outro.

Tabela 2 – Parâmetros do conversor para e Etapa inversora e retificadora

Parâmetro	Descrição	Valor	Unidade
P_{R_s}	Potência sobre resistor R_s	3	Watt
V_{cc}	Tensão de entrada	12	Volt
f_s	Frequência de operação	40.000	Hertz
V_o	Tensão retificada na carga - R_{carga} - Definição	5 V	
$P_{R_{carga}}$	Potência entregue na carga - R_{carga}	3 W	

Fonte: (Autor, 2021)

4 APLICAÇÃO DA METODOLOGIA

Com as definições apresentadas no capítulo 3, o algoritmo é aplicado para obtenção dos valores otimizados de Q_1 , A_3 e D_c .

O PSO irá otimizar os valores fazendo o deslocamento das partículas pelo espaço de busca, de forma que as posições das partículas devem satisfazer os menores valores das variáveis de estado, na forma expressa na equação (72). Essa equação pode ser modificada para atender a outros parâmetros específicos. Esta é uma das vantagens da aplicação do método de otimização. A função objetivo faz o somatório dos valores de pico das variáveis que foram escolhidas, sendo que no algoritmo elaborado já são considerados esses valores em módulo. Esta é a razão do somatório da função objetivo não utilizar termos quadráticos.

A escolha desses parâmetros como uma medida qualitativa para definir a função objetivo é para minimizar os estresses nos componentes, principalmente na chave, pois, conforme o resultado obtido e demonstrado na Figura 18, este valor pode ser superior a três vezes a tensão da fonte de entrada.

$$\sum (i_{L_{fpico}}, i_{L_{spico}}, v_{C_{spico}}, v_{C_{ppico}}) = 0 \quad (72)$$

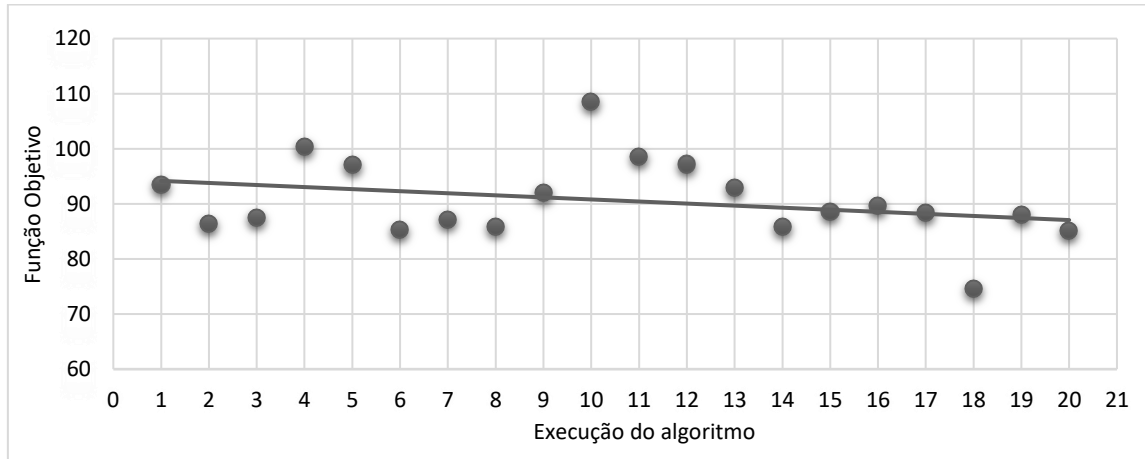
4.1 AVALIAÇÃO DA FUNÇÃO

Algumas medidas estatísticas para avaliação dos resultados do PSO foram obtidas calculando-se a média, a mediana e o desvio padrão dos resultados gerados de uma série de execuções do algoritmo desenvolvido. Na Tabela 3, estão organizados os resultados para 20 execuções do algoritmo.

No Apêndice D, estão organizados os dados individuais de cada iteração. O maior desvio padrão está relacionado ao parâmetro

Em relação à função objetivo, na Figura 12, está demonstrada a linha de tendência dos resultados das execuções do algoritmo. No Apêndice D, estão os gráficos de cada execução do algoritmo.

Figura 12 – Linha de tendência dos resultados do PSO para 20 execuções do algoritmo



Fonte: (Autor, 2021)

Tabela 3 – Dados estatísticos do algoritmo elaborado

Parâmetro	Média	Mediana	Desvio padrão	Menor valor	Maior valor
<i>Função Objetivo</i>	90,61984	88,5205	7,25010	74,546	108,543
Q_1	5,194425	5,008	0,55694	5	7,45
D_c	0,4104	0,4030	0,016004	0,4	0,4585
A_3	0,37335	0,3202	0,15209	0,1615	0,7375

Fonte: (Autor, 2021)

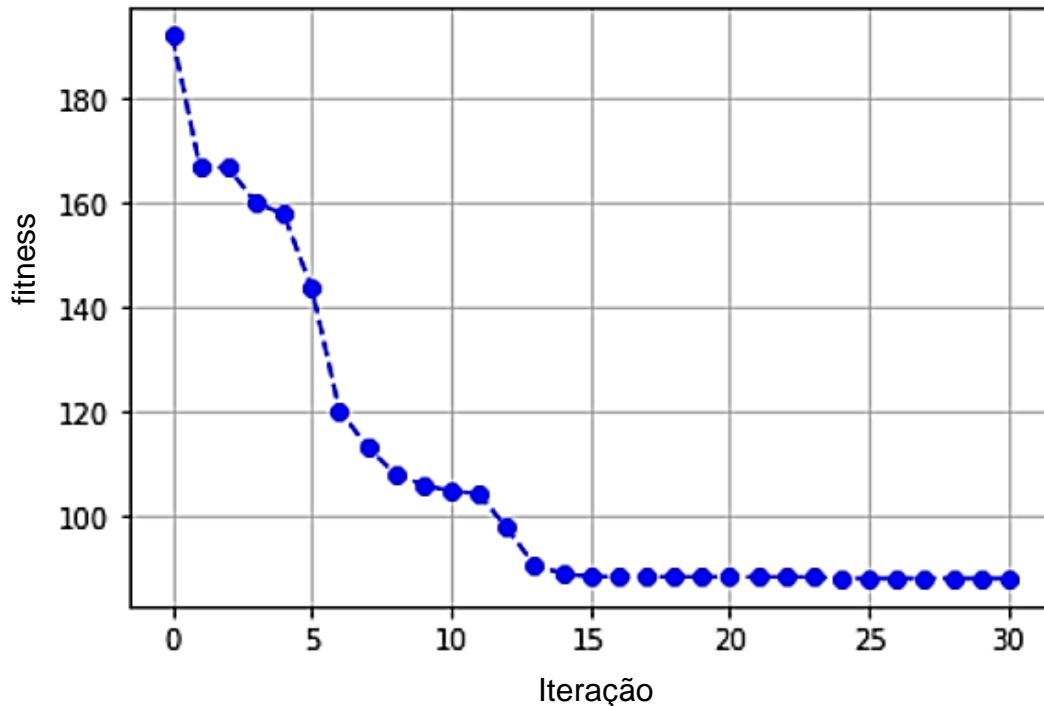
4.2 PROJETO DA ETAPA INVERSORA

Com base no critério estabelecido na função objetivo (72), o algoritmo obteve, em sua primeira aplicação, os valores otimizados $Q_1 = 5,00645$, $A_3 = 0,2766$ e $D_c = 0,40289$.

O resultado mostrado na Figura 13 evidencia a coerência com a equação (72), em que ela deve, de fato, tender à minimização. Na primeira iteração, ocorre o máximo valor, superior a 180, devido à aleatoriedade relativa ao posicionamento de cada

partícula. Após cada iteração, os valores vão reduzindo até que, a partir da décima quinta iteração, aproxima-se de uma convergência para os valores ótimos para Q_1 , A_3 e D_c . O valor final do somatório foi próximo do valor numérico de 85.

Figura 13 – Resultado da primeira execução do algoritmo



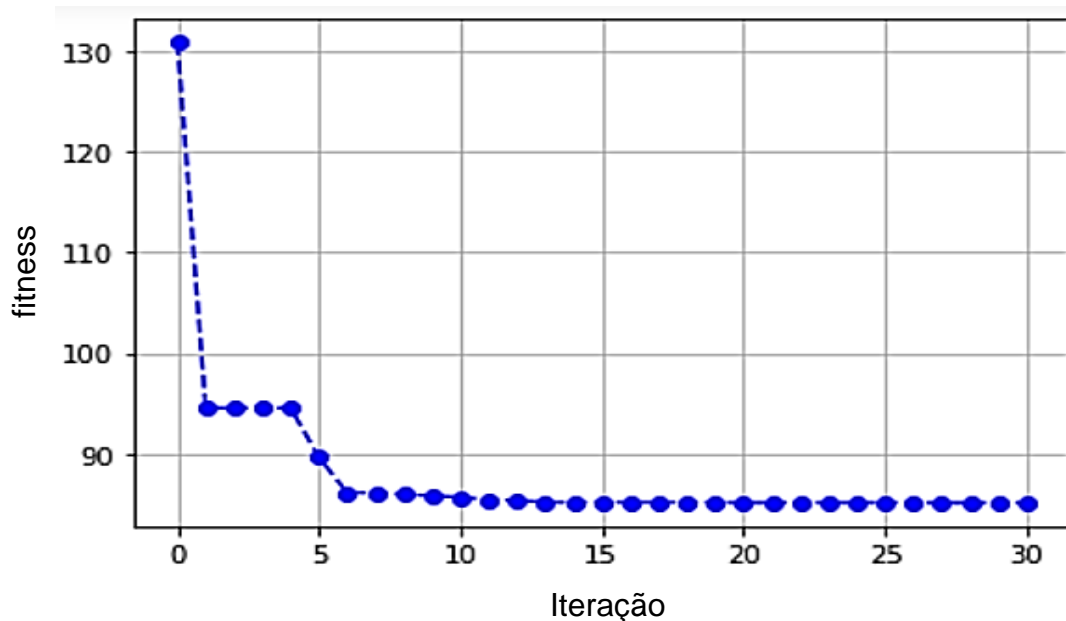
Fonte: (Autor, 2021)

Em uma segunda aplicação do PSO, houve o resultado mostrado na Figura 14. Novamente houve a convergência para minimização da equação (72).

Quanto aos valores otimizados, obteve-se $Q_1 = 5,00496$, $A_3 = 0,2874$ e $D_c = 0,4004$, próximos dos obtidos anteriormente, o que demonstra boa repetibilidade do método. O tempo de execução do algoritmo foi, em ambos os casos, 45 segundos e o valor numérico final do somatório foi de 85,15, semelhante ao obtido na primeira aplicação.

Os valores otimizados obtidos para Q_1 , A_3 e D_c na segunda aplicação serão os utilizados para o restante do projeto da etapa inversora.

Figura 14 – Resultado da PSO adotado



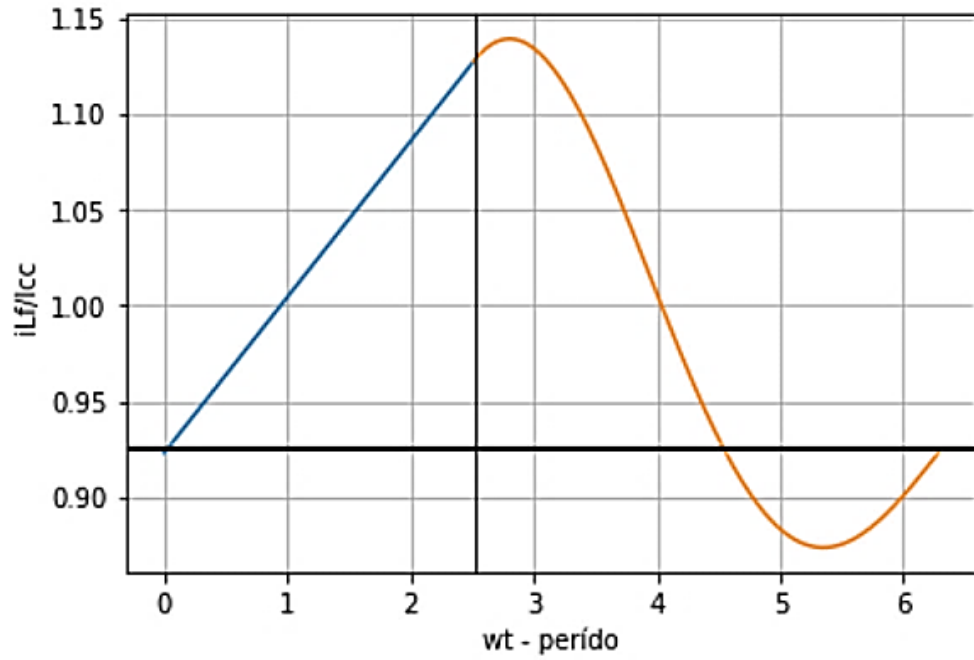
Fonte: (Autor, 2021)

4.2.1 Resposta normalizada do sistema e determinação dos valores de projeto da etapa inversora

Utilizando os valores otimizados obtidos para Q_1 , A_3 e D_c , e retomando as equações (34) e (35), torna-se possível a solução do sistema com a aplicação destes valores. Estas equações são aplicadas nas duas etapas de operação, considerando, segundo Engleitner (2011, p.47), que as condições iniciais da primeira etapa são iguais às condições finais da segunda etapa, e que o circuito continua operando ciclicamente. Com isso, as respostas normalizadas do sistema em termos dos parâmetros ressonantes ficam conforme demonstrados nas Figuras 15, 16, 17 e 18 são referentes à corrente de entrada iL_f , à corrente de saída iL_s , à tensão sobre o capacitor vC_s e sobre o capacitor ressonante vC_p , respectivamente. Essas respostas são referentes a um período de operação.

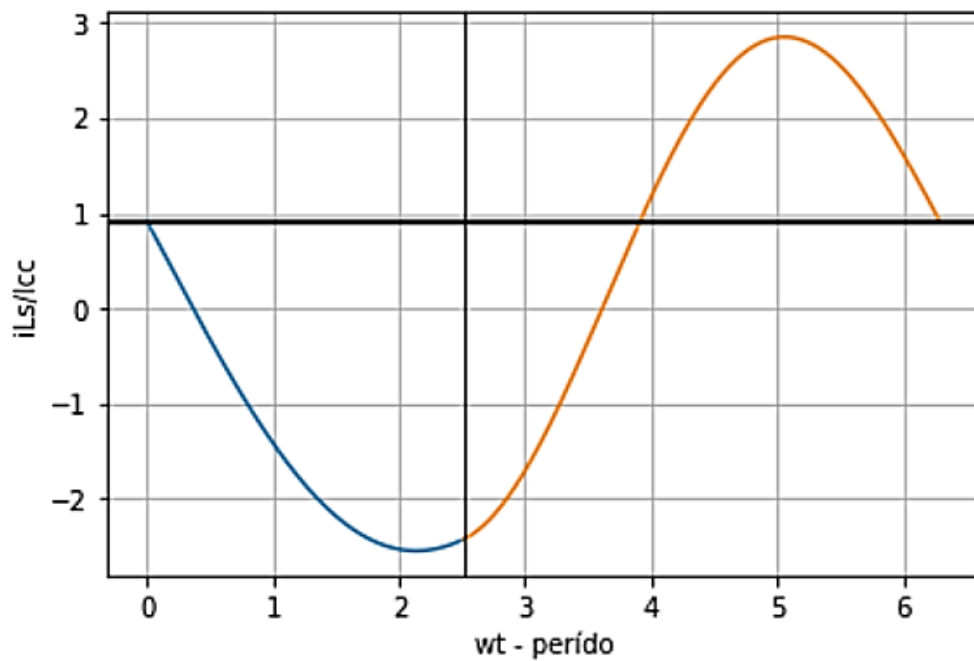
Os valores obtidos a partir da solução do sistema normalizado foram $iL_{s(0)} = 4.11385$, $vC_{s(0)} = 12,53092$, $A_1 = 0,83588$ e $A_2 = 0,83588$. Destaca-se novamente que $vC_{p(0)} = 0$ por ser uma condição de operação imposta, no caso, ZVS. E quanto a $iL_{f(0)} = \frac{A_2}{A_3} iL_{s(0)}$, foi uma relação definida com base na primeira etapa de operação.

Figura 15 – Corrente no indutor L_f normalizada pela corrente de entrada I_{cc}



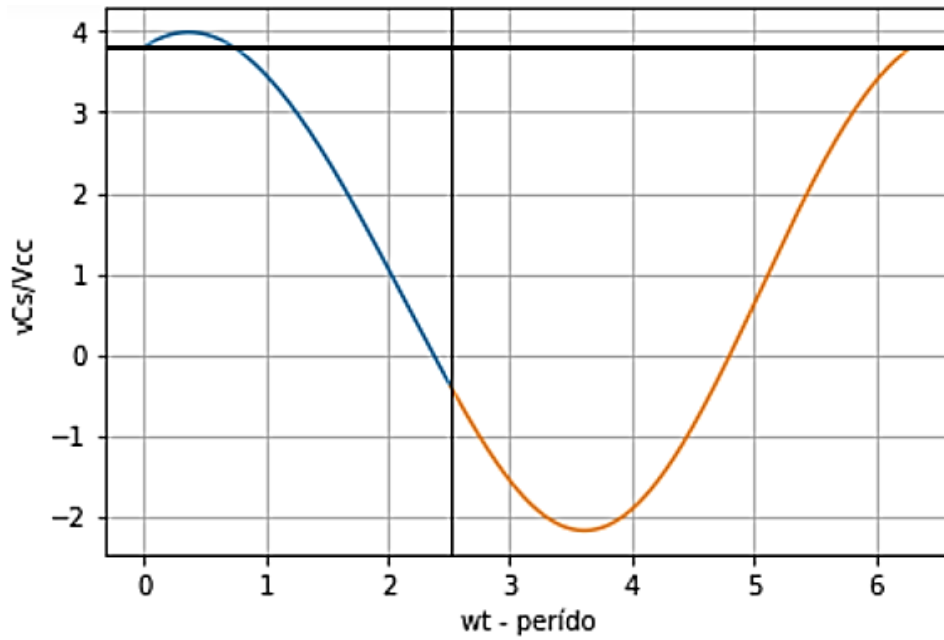
Fonte: (Autor, 2021)

Figura 16 – Corrente no indutor L_s (saída) normalizada pela corrente de entrada I_{cc}



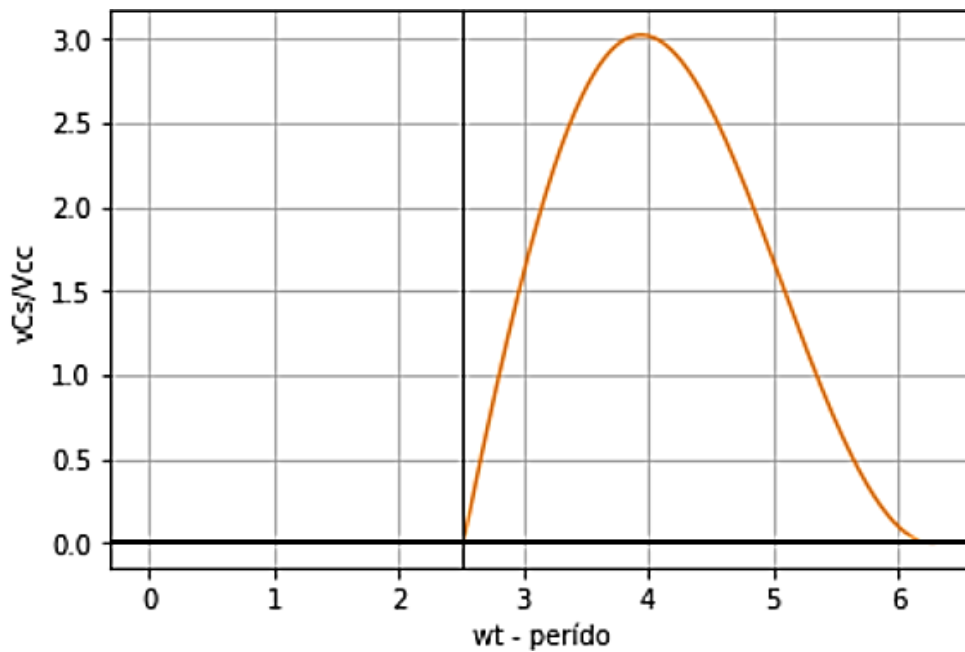
Fonte: (Autor, 2021)

Figura 17 – Tensão sobre capacitor C_s (saída) normalizada pela tensão de entrada V_{cc}



Fonte: (Autor, 2021)

Figura 18 – Tensão sobre capacitor C_p (ressonante) normalizada pela tensão V_{cc}



Fonte: (Autor, 2021)

Esse conjunto de valores forma o vetor $x_{(0)}$ das equações (27) e (28), conforme mostrado em (73).

$$x_{(0)} = \begin{bmatrix} iL_f(0) \\ iL_s(0) \\ vC_s(0) \\ vC_p(0) \end{bmatrix} \quad (73)$$

Portanto, com base na análise da Figura 18, percebe-se que a tensão sobre o capacitor, ou sobre a chave, já que estão em paralelo, chega a 3,01 vezes a tensão de entrada.

4.2.2 Comparação entre resultados teóricos e simulados da etapa inversora

Considerando os valores definidos para o projeto conforme Tabela 3, as tensões esperadas sobre a chave e o capacitor de saída C_s , seriam 36,12 V e 48 V. respectivamente. A corrente I_{cc} pode ser obtida pelo resultado numérico do inverso da taxa de transferência de potência. Para isso, é necessário ter o valor da resistência R_s para aplicar a equação (11).

Neste ponto, além do valor de R_s , os valores de L_f , L_s , C_s e C_p também já podem ser obtidos com o auxílio das equações (4)-(14), tendo em vista que os valores de A_1 , A_2 , A_3 e Q_1 foram determinados. Estes resultados estão organizados na Tabela 4.

Com isso, e com o auxílio do Software Psim®, o circuito equivalente mostrado nas Figuras 7 e 8 foi simulado e os resultados desta análise estão apresentados nas figuras subsequentes.

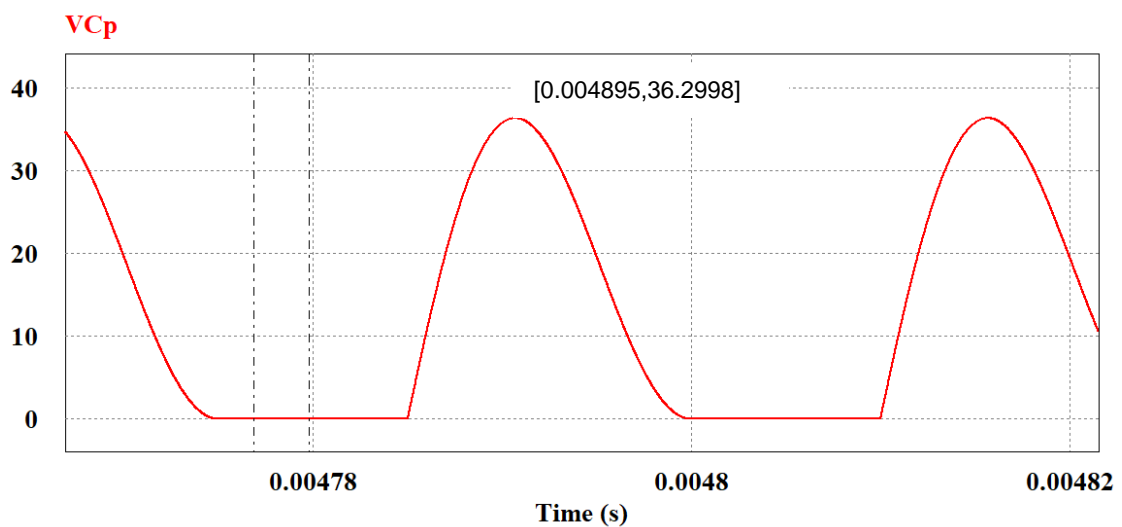
Nota-se, a partir da Figura 19, que o máximo valor foi coerente com o valor deduzido teoricamente. Como complemento, seguem os resultados simulados para iL_f , iL_s , e vC_s . Além disso, a forma da onda da Figura 19 demonstra a ocorrência da ressonância.

Essa corrente mostrada na Figura 21 tem a forma senoidal (muito semelhante) conforme desejado para a etapa inversora do conversor. Além disso, quanto às amplitudes dos valores obtidos, a corrente I_{cc} calculada foi de 0,25 A e considerando esse valor, torna-se também coerente o resultado com o da Figura 16.

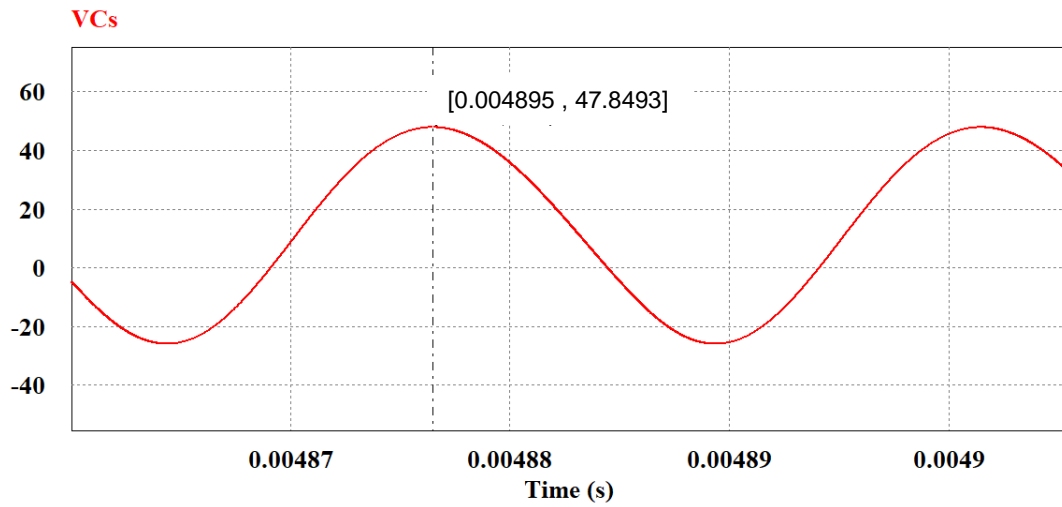
Tabela 4 – Parâmetros do circuito Classe E – Etapa Inversora

Parâmetro	Descrição	Valor
L_f	Indutor de entrada	2,348 mH
L_s	Indutor de saída	308,594 μ H
C_s	Capacitor de saída	73,423 nF
C_p	Capacitor paralelo a chave	81,554 nF
R_s	Resistência da carga	12,953 Ω
A_1	Frequência angular de ressonância normalizada em relação a $L_s C_s$	0,83588
A_2	Frequência angular de ressonância normalizada em relação a $L_s C_p$	0,7931
A_3	Frequência angular de ressonância normalizada em relação a $L_f C_p$	0,28748
Q_1	Fator de qualidade	5,00496
a	Inversor da taxa de transferência de potência	3,70565
I_{cc}	Corrente CC de entrada do conversor	0,25

Fonte: (Autor, 2021)

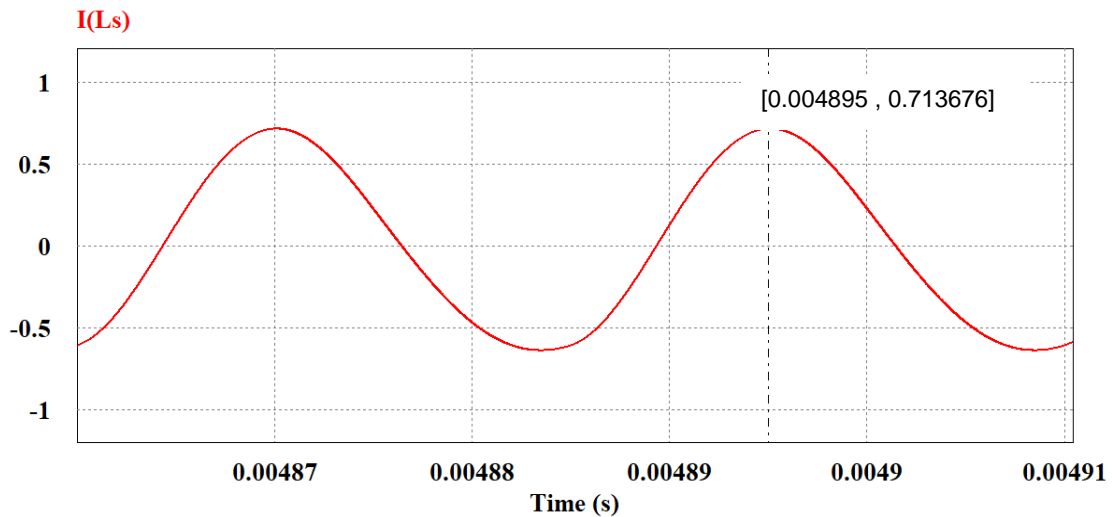
Figura 19 – Tensão sobre capacitor C_p (ressonante) simulada

Fonte: (Autor, 2021)

Figura 20 – Tensão sobre capacitor C_s 

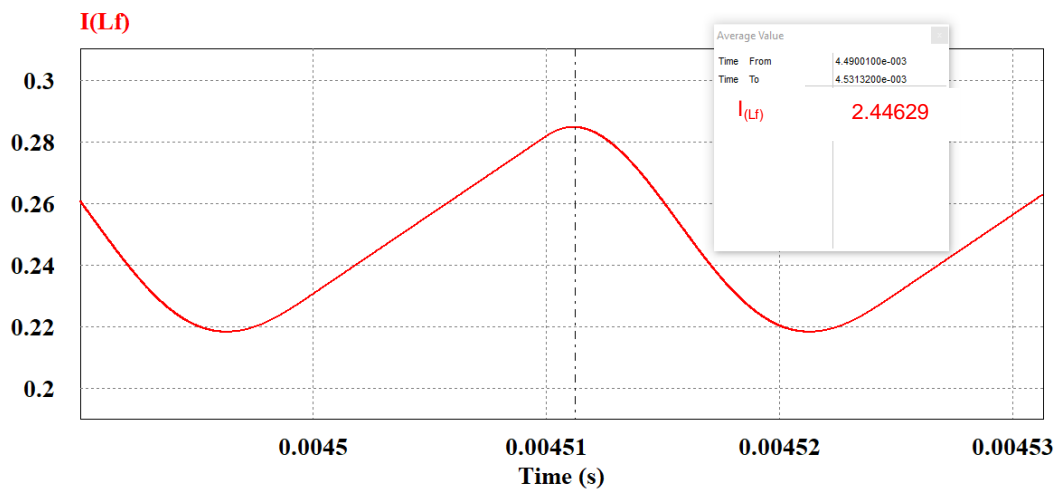
Fonte: (Autor, 2021)

Nota-se também que o resultado para a tensão sobre o capacitor C_s foi conforme esperado, obtendo um valor aproximado de 48 V.

Figura 21 – Corrente no indutor L_s (saída) simulada

Fonte: (Autor, 2021)

Figura 22 – Corrente no indutor L_f (entrada) simulada



Fonte: (Autor, 2021)

Destaque para a corrente média neste indutor, que foi de 0,2446 A. Portanto, os cálculos teóricos foram coerentes e satisfatórios comparados aos resultados de simulação.

A corrente eficaz de saída pode ser calculada a partir desses resultados. Neste caso:

$$P_o = R_s I_{s_{RMS}}^2 = R_s \frac{1}{2\pi} \int_0^{2\pi} iL^2_{f(\omega t)} d\omega t = 12,953 \times 0,4812^2 = 2,9993W \quad (74)$$

Logo, o resultado da equação (74) satisfaz a diretriz de projeto, que era dissipar 3 W de potência sobre a resistência R_s .

4.3 PROJETO DA ETAPA RETIFICADORA

De acordo com Kazimierczuk (2011), o retificador Classe E é o homólogo do inversor Classe E. Além disso, ainda de acordo com o autor, o retificador apresenta algumas outras vantagens, como por exemplo, alta eficiência, possuindo apenas uma chave passiva e um capacitor para o circuito ressonante (além do filtro LC de saída).

4.3.1 Obtenção dos parâmetros e simulação da etapa retificadora

A etapa retificadora do circuito retificador foi modelada como uma impedância equivalente mostrada na Figura 5, em que uma capacitância C_i em série com a resistência R_s representa a carga formada pelo circuito da Figura 6.

O capacitor C_s é, então, uma capacitância resultante da associação de C_{s2} com C_i . Deste modo, a etapa inversora é vista como uma fonte de corrente CA. Considerando esta aproximação, as formas de onda do retificador Classe E é conforme apresentado na Figura 23.

Algumas proposições são assumidas para a análise do retificador:

- O diodo retificador d é ideal, a tensão reversa e a resistência são nulas para polarização direta. Na polarização reversa a resistência é infinita. A capacitância da junção é linear e é incluída em C_d ;
- O controle do retificador é feito pela corrente senoidal iL_s ; e
- O ripple na corrente de saída do retificador - I_o através do indutor L_{filter} é considerado insignificante.

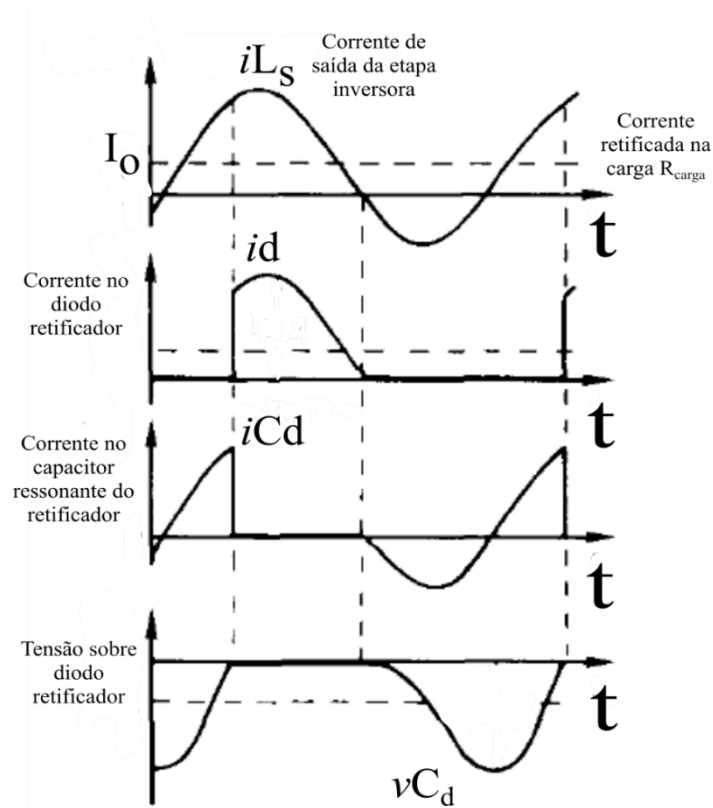
A corrente iL_s é considerada uma senoide ideal e pode ser definida por uma função como demonstrado em (75). Assim como a parte inversora, a parte retificadora também vai ter duas etapas de operação. Analisando a Figura 23, a corrente I_o é igual à corrente iL_s na fase φ . Desta dedução, a corrente I_o é igual a equação (76).

Pela lei de Kirchhoff das correntes, é possível descrever a corrente que irá circular pelo capacitor C_d (paralelo ao diodo retificador), ficando na forma da equação (77).

Agora, definindo que a corrente de saída I_o é conforme a equação (78), é possível determinar a tensão sobre o capacitor C_d . Essa definição é feita por Kazimierczuk (1989, p.573), e aplicada na equação (77), obtendo como resultado a equação (79).

Ainda de acordo com o autor (Kazimierczuk, 1989), é possível determinar uma relação de ganho entre a tensão de entrada da etapa inversora e retificadora diretamente. Entretanto, pode ser necessário a inserção de um transformador para razões de ganho muito alta.

Figura 23 – Formas de onda do Retificador Classe E



Fonte: (Kazimierczuk, 2021. Adaptado)

A tensão sobre o diodo da etapa retificadora também deve operar de modo ressonante. Essa condição implica em uma imposição a ser feita para a solução da equação (79). Essa condição, de acordo com Kazimierczuk (1989), deve ser de acordo com a igualdade na equação (80).

Com isso, uma relação do ângulo φ em função de D_d é estabelecida conforme mostrado na equação (81).

$$iL_s(\omega t) = I_m \sin(\omega t + \varphi) \quad (75)$$

$$I_o = I_m \sin(\varphi) \quad (76)$$

$$iC_d(\omega t) = I_o \left(1 - \frac{\sin(\omega t + \varphi)}{\sin(\varphi)} \right) \quad (77)$$

$$I_o = \frac{V_o}{R_{carga}} \quad (78)$$

$$\begin{aligned} v_{d(\omega t)} &= \frac{1}{\omega C_d} \int_0^{\omega t} i_{C_3} d\omega t \\ &= \frac{V_o}{\omega C_3 R_{carga}} \left(\omega t + \frac{\cos(\omega t + \varphi) - \cos(\varphi)}{\sin(\varphi)} \right) \end{aligned} \quad (79)$$

$$v_{d(2\pi(1-D_d))} = 0 \quad (80)$$

$$\tan(\varphi) = \frac{1 - \cos(2\pi D_d)}{2\pi(1 - D_d) + \sin(2\pi D_d)} \quad (81)$$

Ainda de acordo com Kazimierczuk, existe uma relação direta entre a resistência R_s (resistência da etapa inversora) com a resistência da carga R_{carga} (resistência da etapa retificadora). Esta razão é conforme expressada na equação (82), em que depende diretamente do ângulo φ .

Além desta relação, o autor também demonstra a relação entre a capacitância que deve ser deduzida da capacitância C_s (isto é, C_i), obtida do modelo da etapa inversora do Classe E. O valor que deve ser reduzido de C_s se deve ao fato de que a impedância que modela a etapa retificadora adiciona uma capacitância em série (conforme Figura 5), e associações dessa forma de capacitância resulta em valores menores. Esta relação está mostrada na equação (83).

Por fim, a relação para determinação do valor do capacitor paralelo ao diodo também é definida pelo autor, na forma mostrada na equação (84).

$$\frac{R_s}{R_{carga}} = 2 \sin^2(\varphi) \quad (82)$$

$$\left(\frac{\frac{C_i}{C_d} = \pi}{\pi(1-D_d) + \sin(2\pi D_d) - \frac{1}{4}\sin(4\pi D_d) \cos(2\varphi) - \frac{1}{2}\sin(2\varphi)\sin^2(2\pi D_d) - 2\pi(1-D_d)\sin(\varphi)\sin(2\pi D_d - \varphi)} \right) \quad (83)$$

$$\omega C_d R_s = \frac{\sin^2(\varphi)}{\pi} \left\{ 1 - \cos(2\pi D_d - 2\pi^2) - 2\pi^2(1 - D_d)^2 + \frac{(2\pi(1 - D_d) + \sin(2\pi D_d))^2}{1 - \cos(2\pi D_d)} \right\} \quad (84)$$

Com isso, é possível determinar os valores dos parâmetros de projeto da etapa retificadora. Para fins de simulação e posterior validação por implementação, a Tabela 5 apresenta os valores definidos para esta etapa e na Tabela 6, os valores calculados a partir das equações (75)-(84).

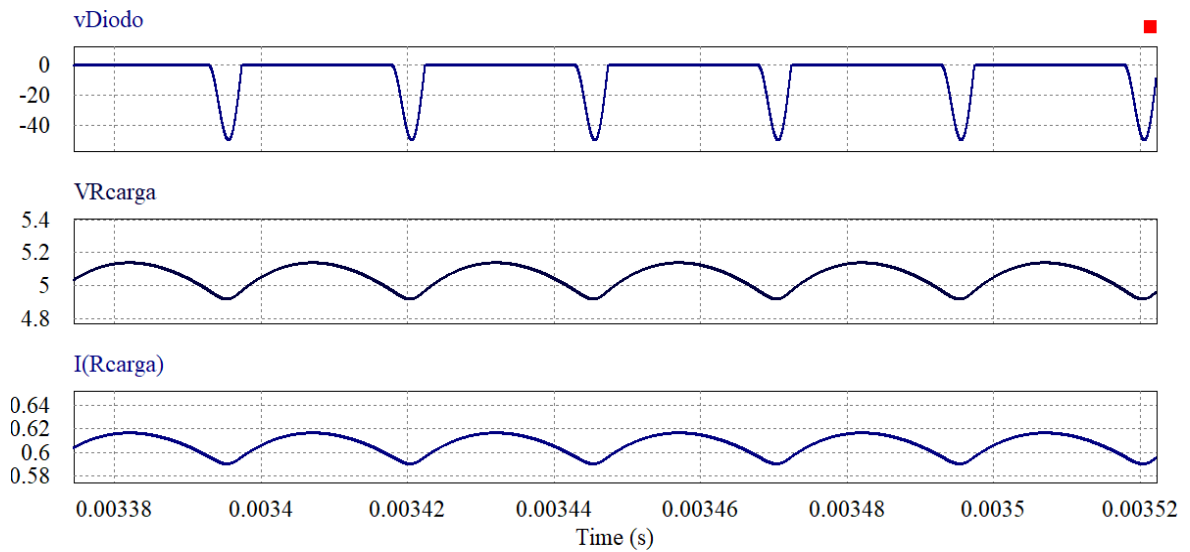
A potência teórica entregue na carga está considerando rendimento ideal, sem perdas. O cálculo do filtro LC de saída do retificador foi obtido a partir das equações mostradas nos fragmentos do Anexo I.

Com isso, o circuito da Figura 5 foi implementado no Psim® e simulado, o qual forneceu os resultados mostrados na Figura 24. A tensão sobre o diodo retificador apresentou a operação ressonante.

Tabela 5 – Especificações e parâmetros calculados para o retificador Classe E

Parâmetro	Descrição	Valor
I_o	Corrente retificada na carga - R_{carga}	0,6 A
R_{carga}	Resistência da carga etapa retificada	8,3334 Ω
C_d	Capacitor paralelo ao diodo	5,5732 nF
C_i	Capacitância da impedância que modela a etapa retificadora	728,974 nF
C_{s2}	Capacitância resultante para a etapa inversora do Classe E	81,646 nF
D_d	Período de condução do diodo retificador	0,7623
φ	Ângulo de defasagem da corrente iL_s	61,8340°
L_{filter}	Indutância do filtro LC da etapa retificadora	493,125 μH
C_{filter}	Capacitância do filtro LC da etapa retificadora	3,2104 μF

Figura 24 – Resultados do circuito Classe E – Etapa retificadora



Fonte: (Autor, 2021)

4.4 IMPLEMENTAÇÃO E RESULTADOS EXPERIMENTAIS

Para validação da metodologia proposta de projeto do conversor classe E com aplicação do PSO, o circuito da Figura 5 foi implementado para coleta de dados. Na Tabela 7 estão os valores dos componentes reais e os calculados.

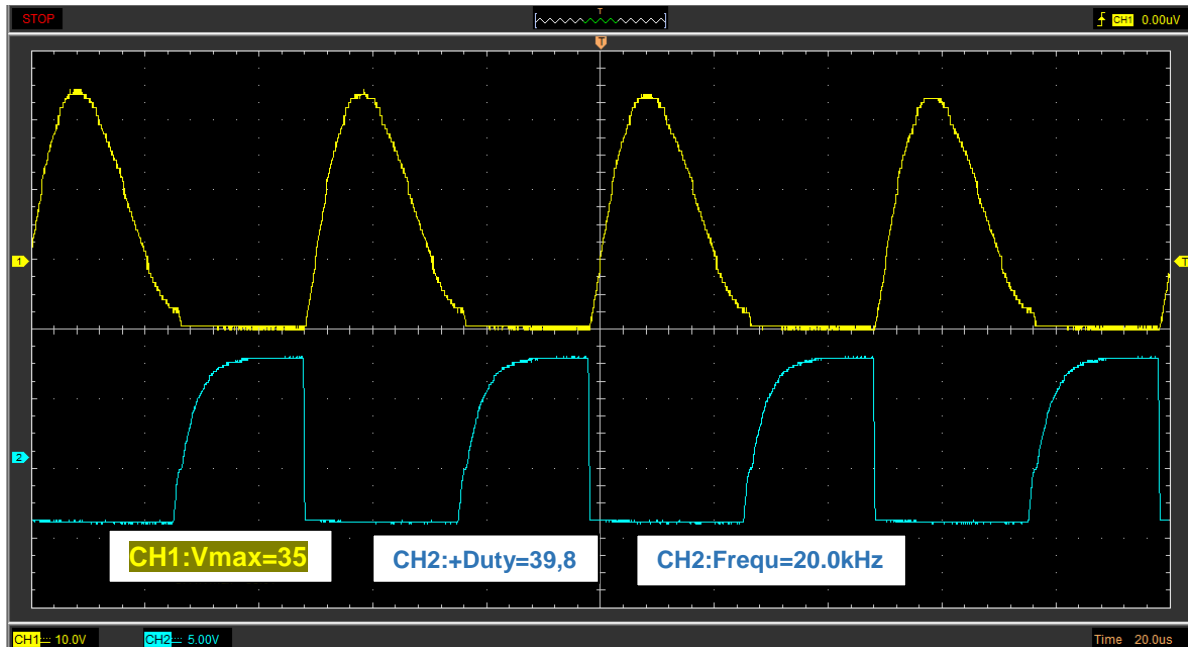
Tabela 6 - Especificações calculadas para Classe E – Etapa retificadora e inversora

Componente	Valor calculado	Valor medido
L_f	2,348 mH	2,2 mH
L_s	308,594 μ H	318 μ H
C_{s2}	73,423 nF	74,7 nF
C_p	81,554 nF	83,2 nF
R_{carga}	12,953 Ω	10,1 Ω
L_{filter}	493,125 μ H	517 mH
C_{filter}	3,2104 nF	3 μ F

Fonte: (Autor, 2021)

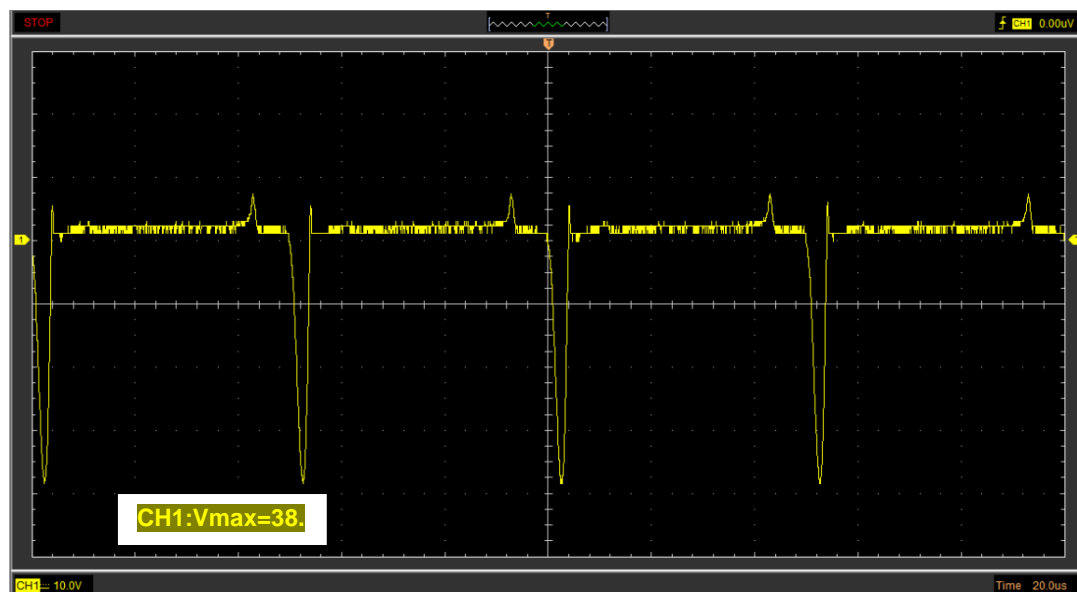
Com isso, foram obtidos os resultados mostrados nas figuras subsequentes.

Figura 25 – Tensão sobre capacitor C_p paralelo a chave ativa.



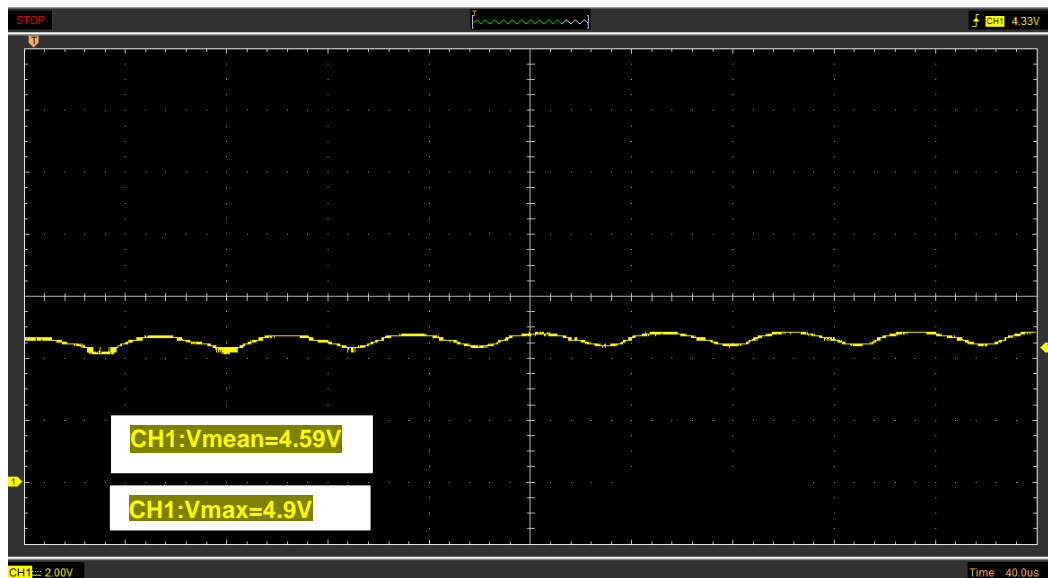
Fonte: (Autor, 2021) .

Figura 26 – Tensão sobre capacitor C_d , ou diodo.



Fonte: (Autor, 2021)

Figura 27 – Tensão de saída retificada



Fonte: (Autor, 2021)

Nota-se que as curvas ficaram semelhantes às obtidas tanto no modelo teórico quanto as da obtidas na simulação. Isso pode ser observado comparando-se as curvas das Figuras 15, 16 e 22. Considerando que a tensão de entrada é 12 V e que o resultado normalizado mostrou-se ser 3 vezes este valor, logo, o resultado na simulação foi 36,29 V, o que é coerente. Na implementação, foi obtido 35,5 V, entretanto, as perdas não foram consideradas e alguns componentes reais utilizados tinham seus valores próximos aos valores calculados, o que pode justificar a diferença entre estes valores.

A mesma análise pode ser feita comparando-se as Figuras 24, e 26 para a tensão sobre o diodo e a Figura 24 para a tensão na carga. Pela simulação, o valor de pico da tensão sobre o diodo era para ser superior, em módulo, a 40 V e na implementação foi medido 38,4 V. O mesmo ocorreu com a tensão retificada na carga, em que a média simulada foi de 5 V e a medida foi de 4,59 V.

Outro ponto de destaque é em relação ao tempo de projeto, que ficou reduzido ao tempo de execução do algoritmo, ou seja, 45 segundos.

5 CONSIDERAÇÕES FINAIS

O PSO aplicado diretamente na etapa inversora mostrou-se válido diante dos resultados obtidos, demonstrando a sua eficiência para obtenção de parâmetros que em trabalhos anteriores eram arbitrados ou escolhidos a partir de figuras de méritos para auxílio no projeto.

De fato, a aplicação de métodos de otimização demonstra um significativo ganho para desenvolvimento de projetos que vão se tornando cada vez mais complexos, e é nesse pilar se sustenta a principal contribuição dessa dissertação, tendo-se em vista que o esforço maior está na fase de estruturar a descrição matemática do conversor, neste caso. Uma vez que esta fase é superada, o atendimento as especificações de projeto passam a ser, ou, podem ser atendidas com uma modificação relativamente simples da equação (72).

Além disso, outras possibilidades para o incremento do algoritmo podem ser determinadas pela imposição de outras condições de projeto desejadas, lembrando que neste caso foi definido somente que os valores de pico fossem minimizados.

Quanto ao conversor, houve também uma revisão bibliográfica para a implementação conjunta das etapas retificadora e inversora, possibilitando a contextualização de uma aplicação entre níveis usuais de tensão. Essa revisão demonstra uma outra contribuição deste trabalho, para ser um material de apoio a introdução deste assunto.

Ressalta-se que a implementação foi feita em um “*lab-home*” devido as restrições de acesso e de deslocamento nesse período de pandemia, e que no Apêndice C apresenta-se a relação dos equipamentos e componentes utilizados.

Para trabalhos futuros, sugere-se um estudo para a ampliação da aplicação do PSO a outras topologias de conversores e ainda, para a etapa de controle, que não fora abordada neste trabalho. Pode-se, ainda, dentro da topologia Classe E, fazer o estudo com a aplicação de outros algoritmos de otimização. Sugere-se, também, a extensão do PSO considerando a etapa retificadora para seleção dos melhores valores.

E ainda, a continuidade da extensão deste trabalho para simulação em tempo real, com a integração deste algoritmo elaborado.

REFERÊNCIAS

BISOGNO, F. E. **Energy-Related System Normalization and Decomposition Targeting Sensitivity Consideration**, PhD Dissertation. 2006.

CIELO, L. B. Procedimento de Projeto de Controladores Robustos para Conversores de Potência por meio de Otimização por Enxame de partículas. Dissertação de Mestrado. Universidade Federal de Santa Maria, 2020.

CHARLES, K.A.; SADIKU, M.N.O. **Fundamentals of Electric Circuits**, 5th Edition. The McGraw-Hill Companies, Inc., New York, New York, 2013

CHEN, C.-T. **Linear System Theory and Design**. 3rd ed., The Oxford series in electrical and computer engineering. 1999.

EBERHART, R. KENNEDY, J. A New Optimizer Using particle Swarm Theory. In: Sixth International Symposium on Micro Machine and Human Science, IEEE.

EBERHART, R.; KENNEDY, J. A new method optimizer using particle theory. In: **IEEE. In proceedings of the sixth international symposium on micro machine and human Science**. [S.l.], 1995. IECI-17, p.39-43.

ENGLEITNER, R. Análise estática normalizada e modelagem de pequenos sinais do conversor classe – E utilizando transformadores peizoelétricos. Dissertação de Mestrado. Universidade Federal de Santa Maria, 2011.

KAZIMIERCZUK, M. K. Resonant dc/dc Converter with Classe-E Inverter and Class-E Rectifier. In: **IEEE Transactions on Industrial Electronics and Control Instrumentation.**, vol. 36, no 4, november 1989.

KAZIMIERCZUK, M. K. Resonant dc/dc Converter with Classe-E Inverter and Class-E Rectifier. In: **IEEE Transactions on Industrial Electronics and Control Instrumentation.**, vol. 36, no 4, november 1989.

KAZIMIERCZUK, M. K. KRZYSZTOF, P. Exact Analysis of Class E Tuned Power Amplifier at any Q and Switch Duty Cycle . In: **IEEE Transactions on Circuits and Systems.**, vol. Cas-34, no 2, february 1987

KAZIMIERCZUK, M. **RESONANT POWER CONVERTERS**, 2th Edition. A JOHN WILEY & SON, INC., PUBLICATION, 2011

MILLER, E.J. Resonant Switching Power Conversion. In: **IEEE Power Electronics Specialists Conference.**, Brasov, ROMANIA. 8-10 June 1976

RAO, S. S. **Engineering Optmizations – Theory and Praticce**. 4th Edition. A JOHN WILEY & SON, INC., PUBLICATION, 2009

TALUKDER, S. Mathematical Modelling and Applications of Particle Swarm Optimization. Master's Thesis. Blekinge Institute of Technology, 2011.

SCHWARZ, F. C. A method of resonant current pulse modulation for ower converters. In: **IEEE Transactions on Industrial Electronics and Control Instrumentation**. [S.l.: s.n.], 1970. IECI-17, p. no. 3.

SANGOI, L. M. Abordagem normalizada para conversores ressonantes. Dissertação de Mestrado. Universidade Federal de Santa Maria, 2017.

Tomoharu, etal,. Steady-State Analysis of Isolated Class-E² Converter Outside Nominal Operation. In: IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 64, NO. 4, APRIL 2017

XINQUIN, W. Analysis of Class-E Rectifier with Low Output-Filter Inductance. In: **8th International Conference on Renewable Energy Research and Aplications.**, Brasov, ROMANIA, nov. 3-6 2019.

APÊNDICE A – Código utilizado para o PSO

12/06/2021

Untitled1

In []:

```

# Class-E Icc
%reset -f
from matplotlib import pyplot as plt
import numpy as np
from math import pi

## BLOCO 1 - Bibliotecas ###

import sympy as sp # symbolic lib
import scipy as sci # lib for integration
import scipy.signal as sgn # lib for state space modelling
from scipy.optimize import fsolve # numerical solve function
from IPython.display import display
import math

from numpy import random

## BLOCO 2 - Bibliotecas ###

#Definição dos parâmetros para o circuito
#O PSO vai procurar os melhores Q1, A3, Dc
#As entradas para o PSO são Vcc, Potência de saída po, e as não idealidades dos compone
ntes (características intrínsecas)

Vcc = 12
pos = 3
fs = 40e3
ILsoArb = 31.68213405
VCsoArb = 53.0465706
A1Arb = 0.99453099
A2Arb = 0.22164779
VCpo= 0
rLf = 0
rLs = 0
rCs = 0
rCp = 0
rCh = 0

## BLOCO 3 ###

clear (Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)

## BLOCO 4 - Para Limpar resíduos das variáveis###

# Função Classe E
def classeE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo):

    IC=[ILfo,ILso,VCso,VCpo]

    AI=np.array([[0,0,0,0],[0,-A1/Q1,-A1,0],[0,A1,0,0],[0,0,0,0]])
    BI=np.array([[1],[0],[0],[0]])
    CI=np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1],[0,0,0,1]])
    DI=np.array([[0],[0],[0],[0],[0]])
    systemI = sgn.StateSpace(AI,BI,CI,DI)

    t=[0,Dc*2*pi]
    u=[1,1]

```

localhost:8888/nbconvert/html/Desktop/Material disserta Área de Trabalho/Codigos_jupyter/Untitled1.ipynb?download=false

2/9

12/06/2021

Untitled1

```

    toutI, yI, xI=systemI.output(u,t,IC)

    AII = np.array ([[0,0,0,-A3],[0,-A1/Q1,-A1,A2],[0,A1,0,0],[A3,-A2,0,0]])
    BII = np.array([[1],[0],[0],[0]])
    CII = np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1],[0,0,0,1]])
    DII = np.array ([[0],[0],[0],[0],[0]])

    systemII=sgn.StateSpace(AII,BII,CII,DII)

    IC2=xI[1]

    t=[0,(1-Dc)*2*pi]
    u=[1,1]
    toutII, yII, xII=systemII.output(u,t,IC2)

    return xI[1], xII[1], systemI, systemII

## BLOCO 5 - Função Classe E para solução dos Estados###

#Função Classe E1
def classE1(X):

    Dc = vetDcUp[n]
    Q1 = vetQ1Up[n]
    A3 = vetA3Up[n]
    ILso=X[0]
    VCso=X[1]
    A1=X[2]
    A2=X[3]
    ILfo=(A2/A3)*ILso
    IC = [ILfo,ILso,VCso,VCpo]
    IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[1]
    eq1=IC2-IC

    return eq1

## BLOCO 6 - Função Classe E1 para Inserção de valores para condições iniciais###

# Função classE_new com novas matrizes CI e CII - Recalculo considerando os valores iniciais
def classE_new(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a):

    IC=[ILfo,ILso,VCso,VCpo]

    AI=np.array([[0,0,0,0],[0,-A1/Q1,-A1,0],[0,A1,0,0],[0,0,0,0]])
    BI=np.array([[1],[0],[0],[0]])
    CInew=np.array([[ (a*A1*A3**2)/(Q1*A2**2),0,0,0],[0,(a*A1*A3)/(Q1*A2),0,0],[0,0,(A1*A3)/A2,0],[0,0,0,0]])
    DI=np.array([[0],[0],[0],[0]])
    systemI = sgn.StateSpace(AI,BI,CInew,DI)

    t=[0,Dc*2*pi]
    u=[1,1]
    toutI, yI, xI=systemI.output(u,t,IC)

    AII = np.array ([[0,0,0,-A3],[0,-A1/Q1,-A1,A2],[0,A1,0,0],[A3,-A2,0,0]])

```


12/06/2021

Untitled1

```

    BII = np.array([[1],[0],[0],[0]])
    CIInew = np.array([[ (a*A1*A3**2)/(Q1*A2**2),0,0,0],[0,(a*A1*A3)/(Q1*A2),0,0],[0,0,(
A1*A3)/A2,0],[0,0,0,A3]])
    DII = np.array ([[0],[0],[0],[0]])

    systemII=sgn.StateSpace(AII,BII,CIInew,DII)

    IC2=xI[1]

    t=[0,(1-Dc)*2*pi]
    u=[1,1]
    toutII, yII, xII=systemII.output(u,t,IC2)

    return xI[1], xII[1], systemI, systemII

## BLOCO 7 ###

#Função classE1_new com novas matrizes CI e CII e "a"

def classE1_new(X):

    ILso=X[0]
    VCso=X[1]
    A1=X[2]
    A2=X[3]
    A3=vetA3Up[n]
    Dc=vetDcUp[n]
    Q1=vetQ1Up[n]
    ILfo=(A2/A3)*ILso
    IC = [ILfo,ILso,VCso,VCpo]
    IC2=classE_new(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[1]
    eq1=IC2-IC

    return eq1

## BLOCO 8 ###

#Inicialização do PSO
#- vetQ : Vetor para armazenar valores para Q1
#- vetDc : Vetor para armazenar valores para Dc
#- fit : Vetor para armazenar resultados para cada "n" valor obitido na rotina. Este va
lor deve ser o maior possível
#- veloc : Vetor velocidade
#- aval - avaliação do "rendimento" para o enxame. DeLa se determina o PBeste e o GBest

##*Bloco com inicialização das partículas para Q1 e DC##
Nit = 30 # Número de iterações (P-1)
P = 31 # Número de partículas > Nit
c1 = 0.5          ## Peso do coeficiente pessoal
c2 = 0.5          ## Peso do coeficiente global
r1 = 1           ## Coeficiente aleatório pessoal
r2 = 1           ## Coeficiente aleatório global
wmax = 0.9       ## Coeficiente de inércia máx
wmin = 0.4       ## Coeficiente de inércia min
Q1min = 5        ## Menor Valor para Fator de qualidade
Q1max = 100      ## Maior valor para Fator de qualidade
Dcmin = 40       ## Menor Valor para Razão cíclica
Dcmax = 80       ## Maior valor para Razão cíclica
A3min = 1        ## Menor valor para A3

```

12/06/2021

Untitled1

```

A3max = 17          ## Maior valor para A3

vetQ1 = random.randint(Q1min,Q1max, size=(P))

vetDc = random.randint(Dcmin,Dcmax,size=(P))
vetDc = vetDc/100
Dcmin = Dcmin/100
Dcmax = Dcmax/100

vetA3 = random.randint(A3min,A3max,size=(P))
vetA3 = vetA3/10
A3min = A3min/10
A3max = A3max/10

fit = np.zeros(P)
aval = np.zeros(P)

PBestQ1 = np.zeros(P)
PBestDc = np.zeros(P)
PBestA3 = np.zeros(P)

GBestQ1 = np.zeros(P)
GBestDc = np.zeros(P)
GBestA3 = np.zeros(P)

velocQ1 = np.zeros(P) ## Primeiro vetor coluna, é relativo a Q1
velocDc = np.zeros(P) ## Segundo vetor coluna, é relativo a Dc
velocA3 = np.zeros(P) ## Terceiro vetor coluna, é relativo a A3

vetQ1Up = vetQ1
vetDcUp = vetDc
vetA3Up = vetA3

print (vetQ1)
print (vetDc)
print (vetA3)

## BLOCO 9 ###

#LAÇOS PRINCIPAIS
n=0
it=0
teste = 5
melhortt=1000e10000
while it<=Nit: ## while 1
    n=0
    w = wmax - ((wmax-wmin)/Nit)*it
    avalpast = aval
    aval = np.zeros(P)
    while n < P : ## while 2

        X = fsolve(classE1,[ILsoArb,VCsoArb,A1Arb,A2Arb])
        ILso=X[0]
        VCso=X[1]
        A1=X[2]

```

12/06/2021

Untitled1

```

A2=X[3]
A3=vetA3Up[n]
Dc=vetDcUp[n]
Q1=vetQ1Up[n]
ILfo=(A2/A3)*ILso
ICo=[ILfo,ILso,VCso,VCpo]
##-----##
systemI=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[2]
systemII=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[3]
IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[1]
##-----##
t1=np.linspace(0,Dc*2*pi,1000)
u1 = np.ones_like(t1)
tout1, y1, x=systemI.output(u1,t1,ICo)
t2=np.linspace(0,(1-Dc)*2*pi,1000)
u2 = np.ones_like(t2)*1
IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[0]
tout2, y2, x=systemII.output(u2,t2,IC2)
##-----##
po_R_Vcc2=(A3**2*A1**2)/(A2**2*Q1**2)*1/(2*pi)*(np.trapz((y1[:,1])**2,tout1)+np
.trapz((y2[:,1])**2,tout2))
a=1/po_R_Vcc2
###----Recalculo-----##
X = fsolve(classE1,[ILsoArb,VCsoArb,A1Arb,A2Arb])
##-----##
ILso=X[0]
VCso=X[1]
A1=X[2]
A2=X[3]
ILfo=(A2/A3)*ILso

ICo=[ILfo,ILso,VCso,VCpo]
##-----##
systemI=classE_new(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[2]
systemII=classE_new(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[3]
IC2=classE_new(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[1]
##-----##
t1=np.linspace(0,Dc*2*pi,1000)
u1 = np.ones_like(t1)
tout1, y1, x=systemI.output(u1,t1,ICo)
t2=np.linspace(0,(1-Dc)*2*pi,1000)
u2 = np.ones_like(t2)*1
IC2=classE_new(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[0]
tout2, y2, x=systemII.output(u2,t2,IC2)
##-----##

po = pos
Rs = (Vcc**2)/(a*po)

ilf1 = max( abs((Vcc/(Rs*a))*y1[:,0]))
ilf2 = max( abs((Vcc/(Rs*a))*y2[:,0]))
ilf = max(ilf1,ilf2)

ilfo1 = max( abs((Vcc/(Rs*a))*y1[:,1]))
ilfo2 = max( abs((Vcc/(Rs*a))*y2[:,1]))
ilfo = max(ilfo1,ilfo2)

vco1 = max( abs(Vcc*y1[:,2]))
vco2 = max( abs(Vcc*y2[:,2]))
vco = max(vco1,vco2)

```

12/06/2021

Untitled1

```

vcp1 = max( abs(Vcc*y1[:,3]))
vcp2 = max( abs(Vcc*y2[:,3]))
vcp = max(vcp1,vcp2)

stress = ilf + ilfo + vco + vcp
aval[n] = stress
#print ('aval =', aval[n])
#print ('aval:',aval)
#print ('')
n = n+1 ## Fim do while 2 - Laço de avaliações

####---MATRIZ PBestQ1 e PBesteDc---##
melhor = min(aval)      ## Identificar a melhor resposta do primeiro co
njunto
                                ## de partículas
                                ##
posic=0                    ##
n=0                        ##
while n < P:                ##
    if aval[n] < avalpast [n]: ##
        PBestQ1[n] = vetQ1Up[n] ##
        PBestDc[n] = vetDcUp[n] ##
        PBestA3[n] = vetA3[n]   ##
    n = n+1                  ##
n=0                          ##
while n < P:                 ##
    if melhor == aval [n]:   ##
        posic = n           ##
    n = n+1                  ##
n=0                            ##
while n < P:                 ##
    if melhor <= melhortt:   ##
        GBestQ1[n]=vetQ1Up[posic] ##
        GBestDc[n]=vetDcUp[posic] ##
        GBestA3[n]=vetA3Up[posic] ##
        melhortt = melhor    ##
    n = n+1                  ##
#####
#####---BLOCO DE ATUALIZAÇÕES DE VARIÁVEI ---###
#####
velocUp1 = (w*velocQ1)+(c1*random.rand()*(PBestQ1-vetQ1-1-vetQ1)) ##
velocUp2 = (w*velocDc)+(c1*random.rand()*(PBestDc-vetDc-1-vetDc)) ##
velocUp3 = (w*velocA3)+(c1*random.rand()*(PBestA3-vetA3-1-vetA3)) ##

##
vetQ1 = vetQ1Up
##
vetDc = vetDcUp
##
vetA3 = vetA3Up
##
vetQ1Up = vetQ1+velocUp1
##
vetDcUp = vetDc+velocUp2

```

12/06/2021

Untitled1

```

##
vetA3Up = vetA3+velocUp3
##
#####
#####CORRIGIR A CONVERSÃO DAS PARTÍCULAS#####
#####
n= 0 ##
while n<P: ##
    if vetQ1Up[n] < Q1min : ##
        vetQ1Up[n] = vetQ1[n] ##
        vetDcUp[n] = vetDc[n] ##
        vetA3Up[n] = vetA3[n] ##
    if vetQ1Up[n] > Q1max : ## Contorno de Q1
        vetQ1Up[n] = vetQ1[n] ##
        vetDcUp[n] = vetDc[n] ##
        vetA3Up[n] = vetA3[n] ##
    n=n+1 ##
#####

#####
n= 0 ##
while n<P: ##
    if vetDcUp[n] < Dcmin : ##
        vetDcUp[n] = vetDc[n] ##
        vetQ1Up[n] = vetQ1[n] ##
        vetA3Up[n] = vetA3[n] ##
    if vetDcUp[n] > Dcmax : ## Contorno de Dc
        vetDcUp[n] = vetDc[n] ##
        vetQ1Up[n] = vetQ1[n] ##
        vetA3Up[n] = vetA3[n] ##
    n=n+1 ##
#####

#####
n= 0 ##
while n<P: ##
    if vetA3Up[n] < A3min : ##
        vetDcUp[n] = vetDc[n] ##
        vetQ1Up[n] = vetQ1[n] ##
        vetA3Up[n] = vetA3[n] ##
    if vetA3Up[n] > A3max : ## Contorno de A3
        vetDcUp[n] = vetDc[n] ##
        vetQ1Up[n] = vetQ1[n] ##
        vetA3Up[n] = vetA3[n] ##
    n=n+1 ##
#####

print('Iteração:',it)
fit[it] = melhortt
print('fit',fit[it])
#print ('fit:',fit)
#print ('melhor=',melhor)
#print ('melhortt=',melhortt)

it=it+1 ## fim do while 1 - Laço Principal

plt.plot(fit)

```

12/06/2021

Untitled1

```

print ('')
print ('Q1:',GBestQ1[it-1])
print ('')
print ('Dc:',GBestDc[it-1])
print ('')
print ('A3:',GBestA3[it-1])
print ('')
print ('Q1n=',GBestQ1[0])
print ('Dcn=',GBestDc[0])
print ('A3n=',GBestA3[0])

#print ('GBestQ1:',GBestQ1)
#print ('GBestDc:',GBestDc)

print ('')

#print ('PBestQ1:',PBestQ1)
#print ('')
#print ('PBestDc:',PBestDc)
#print ('')
print (min (fit))

## BLOCO 10 ###

#Saída dos Dados

print ('Q1n=',GBestQ1[0])
print ('Dcn=',GBestDc[0])
print ('A3n=',GBestA3[0])
print ('')
print ('Vcc = ',Vcc)
print ('pos = ',pos)
print ('')
print ('fs = ',fs)
print (vetQ1)
print (vetDc)
print (vetA3)
print (GBestDc)
print (GBestA3)
print (GBestQ1)
GBestA3
fig,ax = plt.subplots()
ax.plot(fit,'bo--')
ax.set(xlabel='Iteração',ylabel='fitness')
ax.grid()
fig.savefig("teste.png")

## BLOCO 11 ###

```

APÊNDICE B – Código utilizado para a solução dos estado e obtenção dos parâmetros de projeto

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSSES_RECOMECO-Copy1

```
In [2]: # Class-E Icc
%reset -f
```

```
In [3]: from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
import numpy as np
from math import pi

import sympy as sp # symbolic lib
import scipy as sci # lib for integration
import scipy.signal as sgn # lib for state space modelling
from scipy.optimize import fsolve # numerical solve function
from IPython.display import display
import math
```

Definição dos parâmetros normalizados e variáveis

L_s Indutor Saída

C_s Capacitor Saída

R_s Resistor de Saída

C_p Capacitor Chave

L_f Indutor Filtro

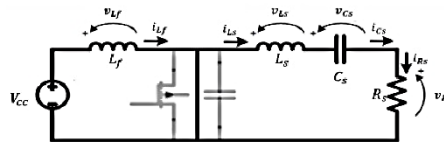
$$\omega_1 = \frac{1}{\sqrt{L_s C_s}}, \omega_2 = \frac{1}{\sqrt{L_s - C_p}}, \omega_3 = \frac{1}{\sqrt{L_f - C_p}}$$

$$A_1 = \frac{\omega_1}{\omega}, A_2 = \frac{\omega_2}{\omega}, A_3 = \frac{\omega_3}{\omega}, \frac{C_s}{C_p} = \frac{\omega_2^2}{\omega_1^2} = \frac{A_2^2}{A_1^2} \text{ and } \frac{L_s}{L_f} = \frac{\omega_3^2}{\omega_1^2} = \frac{A_3^2}{A_1^2}$$

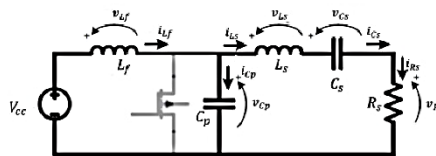
Series quality factor is defined as

$$Q_1 = \frac{\omega_1 L_s}{R_s} = \frac{1}{\omega_1 C_s R_s}$$

Stage I



Stage II



Stage I

Switch on ($0 \leq \omega t < 2\pi Dc$)

$$\begin{aligned} L_f \frac{d i_{L_f}(t)}{dt} &= V_{CC} \\ L_s \frac{d i_{L_s}(t)}{dt} &= -R_s i_{L_s}(t) - v_{C_s}(t) \\ C_s \frac{d v_{C_s}(t)}{dt} &= i_{L_s}(t) \\ C_p \frac{d v_{C_p}(t)}{dt} &= 0 \end{aligned}$$

Stage II

Switch off ($2\pi Dc \leq \omega t < 2\pi$)

$$\begin{aligned} L_f \frac{d i_{L_f}(t)}{dt} &= -v_{C_p} + V_{CC} \\ L_s \frac{d i_{L_s}(t)}{dt} &= -R_s i_{L_s}(t) - v_{C_s}(t) + v_{C_p}(t) \\ C_s \frac{d v_{C_s}(t)}{dt} &= i_{L_s}(t) \\ C_p \frac{d v_{C_p}(t)}{dt} &= i_{L_f} - i_{L_s}(t) \end{aligned}$$

State Space Equations

$$\begin{pmatrix} \dot{i}_{L_f}(t) \\ \dot{i}_{L_s}(t) \\ \dot{v}_{C_s}(t) \\ \dot{v}_{C_p}(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & -\frac{1}{L_s} & 0 \\ 0 & \frac{1}{C_s} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_{L_f}(t) \\ i_{L_s}(t) \\ v_{C_s}(t) \\ v_{C_p}(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{L_f} \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot V_{CC}$$

$$\begin{pmatrix} \dot{i}_{L_f}(t) \\ \dot{i}_{L_s}(t) \\ \dot{v}_{C_s}(t) \\ \dot{v}_{C_p}(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -\frac{1}{L_f} \\ 0 & -\frac{R_s}{L_s} & -\frac{1}{L_s} & \frac{1}{L_s} \\ 0 & \frac{1}{C_s} & 0 & 0 \\ \frac{1}{C_p} & -\frac{1}{C_p} & 0 & 0 \end{pmatrix} \begin{pmatrix} i_{L_f}(t) \\ i_{L_s}(t) \\ v_{C_s}(t) \\ v_{C_p}(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{L_f} \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot V_{CC}$$

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSES_RECOMEKO-Copy1

The state vector

$$\mathbf{X}(t) = \begin{pmatrix} i_{L_f}(t) & 0 & 0 & 0 \\ 0 & i_{L_s}(t) & 0 & 0 \\ 0 & 0 & v_{C_s}(t) & 0 \\ 0 & 0 & 0 & v_{C_p}(t) \end{pmatrix}, \mathbf{X}_e(t) = \begin{pmatrix} \frac{\sqrt{2}\sqrt{L_f}}{2} i_{L_f}(t) & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}\sqrt{L_s}}{2} i_{L_s}(t) & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}\sqrt{C_s}}{2} v_{C_s}(t) & 0 \\ 0 & 0 & 0 & \frac{\sqrt{2}\sqrt{C_p}}{2} v_{C_p}(t) \end{pmatrix}$$

The output vector

$$\mathbf{y}(t) = \begin{pmatrix} i_{L_f}(t) \\ i_{L_s}(t) \\ v_{C_s}(t) \\ v_{R_s}(t) \end{pmatrix}$$

Considering $\mathbf{X}(t)$ and $\mathbf{X}_e(t)$ time invariant, the linear transformation can be done calculating the similarity transform matrix \mathbf{P} as

$$\mathbf{P} = \mathbf{X}_e \mathbf{X}^{-1}$$

The new matrices are thus defined

$$\mathbf{A}_I = \frac{1}{w} \mathbf{P} \mathbf{A} \mathbf{P}^{-1}$$

$$\mathbf{B}_I = \frac{1}{w} \mathbf{P} \mathbf{B}$$

$$\mathbf{C}_I = \mathbf{C} \mathbf{P}^{-1}$$

The new matrices \mathbf{P} considerando o fator \mathbf{K}

$$\mathbf{P}_K = \mathbf{K} \mathbf{P}$$

```
In [4]: i_Lf,i_Ls,v_Cs,v_Cp,L_f,L_s,C_s,C_p,R_s,w,V_cc=sp.symbols('i_Lf,i_Ls,v_Cs,v_Cp,L_f,L_s,C_s,C_p,R_s,w,V_cc')
X=sp.Function('X')
X_e=sp.Function('X_e')
P=sp.Function('P')
P_K=sp.Function('P_K')
X=sp.Matrix([[i_Lf,0,0,0],[0,i_Ls,0,0],[0,0,v_Cs,0],[0,0,0,v_Cp]])
X_e=sp.Matrix([[sp.sqrt(L_f/2)*i_Lf,0,0,0],[0,sp.sqrt(L_s/2)*i_Ls,0,0],[0,0,sp.sqrt(C_s/2)*v_Cs,0],[0,0,0,sp.sqrt(C_p/2)*v_Cp]])
P = X_e*X.inv()
```

```
In [5]: #Matrizes da representação em espaço de estado das equações diferenciais

A_I = sp.Matrix([[0,0,0,0],[0,-R_s/L_s,-1/L_s,0],[0,1/C_s,0,0],[0,0,0,0]])
B_I=sp.Matrix([[1/L_f],[0],[0],[0]])
C_I = sp.Matrix ([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,0],[0,R_s,0,0]])

A_II = sp.Matrix([[0,0,0,-1/L_f],[0,-R_s/L_s,-1/L_s,1/L_s],[0,1/C_s,0,0],[1/C_p,-1/C_p,0,0]])
B_II=sp.Matrix([[1/L_f],[0],[0],[0]])
C_II = sp.Matrix ([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1],[0,R_s,0,0]])
#display(C_II);
```

```
In [9]: ## Primeira transformação das matrizes de estado, pela matriz de transformação em termos da energia dos elementos
# energy state space matrices
A_Ie=1/(w)*P*A_I*P.inv();
B_Ie=1/(w)*P*B_I;
C_Ie=C_I*P.inv();
# energy state space matrices
A_IIe=1/w*P*A_II*P.inv()
B_IIe=1/w*P*B_II
C_IIe=C_II*P.inv();
#display (C_IIe);
#display (C_IIe)
```

Correction normalizing factor regarding input source can be defined as inverse of the non-zero energy based matrix B element, as following

```
In [11]: K = 1/(V_cc*B_Ie[0])
#display(K)

#Redefinindo a matriz de P
P_K = K*P
#display(P_K)
```


12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSES_RECOMEKO-Copy1

```
In [12]: ##Segunda transformação das matrizes de estado, agora considerando o Fator K na matriz P
# energy state space matrices
A_Iek=1/(w)*P_K*A_I*P_K.inv()
B_Iek=1/(w)*P_K*B_I
C_Iek=C_I*P_K.inv()

# energy state space matrices
A_IIek=1/w*P_K*A_II*P_K.inv()
B_IIek=1/w*P_K*B_II
C_IIek=C_II*P_K.inv()

print('C_IIek:')
#display(C_IIek)

C_IIek:
```

Reescrevendo as definições acima, apenas para facilitar a visualização

$$\omega_1 = \frac{1}{\sqrt{L_s C_s}}, \omega_2 = \frac{1}{\sqrt{L_s C_p}}, \omega_3 = \frac{1}{\sqrt{L_f C_p}}$$

$$A_1 = \frac{\omega_1}{\omega}, \quad A_2 = \frac{\omega_2}{\omega}, \quad A_3 = \frac{\omega_3}{\omega}, \quad \frac{C_p}{C_s} = \frac{\omega_1^2}{\omega_2^2} = \frac{A_1^2}{A_2^2} \quad \text{and} \quad \frac{L_s}{L_f} = \frac{\omega_3^2}{\omega_2^2} = \frac{A_3^2}{A_2^2}$$

Series quality factor is defined as

$$Q_1 = \frac{\omega_1 L_s}{R_s} = \frac{1}{\omega_1 C_s R_s}$$

Inverso da taxa de transferência de potência "a"

$$a = \frac{Rdc}{R_s} = \frac{Vcc}{Icc R_s} = \frac{Vcc^2}{P_o R_s}$$

- ### Transcrição dos termos das matrizes

$$\begin{aligned} -\frac{R_s}{L_s \omega_1} &= -\frac{R_s \omega_1}{L_s \omega} = -\frac{A_1}{Q_1} \\ -\frac{1}{\sqrt{C_s} \sqrt{L_s} \omega} &= -\frac{\omega_1}{\omega} = -A_1 \\ \frac{1}{\sqrt{C_s} \sqrt{L_s} \omega} &= \frac{\omega_1}{\omega} = A_1 \\ -\frac{1}{\sqrt{C_p} \sqrt{L_f} \omega} &= -\frac{\omega_3}{\omega} = -A_3 \\ -\frac{1}{\sqrt{C_p} \sqrt{L_s} \omega} &= -\frac{\omega_2}{\omega} = -A_2 \\ \frac{1}{\sqrt{C_p} \sqrt{L_f} \omega} &= \frac{\omega_3}{\omega} = A_3 \end{aligned}$$

Obtenção condição ótima

$$\frac{L_f \omega i L_f}{Vcc} = \frac{\sqrt{L_f} \sqrt{L_s} i L_s}{Vcc} \rightarrow \sqrt{L_f} i L_f = \sqrt{L_s} i L_s \rightarrow i L_f = \frac{\sqrt{L_s} i L_s}{\sqrt{L_f}} \rightarrow i L_f = \frac{A_2 i L_s}{A_3}$$

```
In [14]: ##Matrizes reescritas por parâmetros ressonantes
A_1,A_2,A_3,Q_1,a =sp.symbols('A_1,A_2,A_3,Q_1,a')

AIn = sp.Matrix([[0,0,0,0],[0,-A_1/Q_1,-A_1,0],[0,A_1,0,0],[0,0,0,0]])
BIn = sp.Matrix([[1],[0],[0],[0]])
CIn = sp.Matrix([[ (a*A_1*A_3**2)/(Q_1*A_2**2),0,0,0],[0,(a*A_1*A_3)/(Q_1*A_2),0,0],[0,0,(A_1*A_3)/A_2,0],[0,0,0,0]])

AIIIn = sp.Matrix([[0,0,0,-A_3],[0,-A_1/Q_1,-A_1,A_2],[0,A_1,0,0],[A_3,-A_2,0,0]])
BIIIn = sp.Matrix([[1],[0],[0],[0]])
CIIIn = sp.Matrix([[ (a*A_1*A_3**2)/(Q_1*A_2**2),0,0,0],[0,(a*A_1*A_3)/(Q_1*A_2),0,0],[0,0,(A_1*A_3)/A_2,0],[0,0,0,A_3]])

#print('CIIIn:')
#display(CIIIn)
#display(BIn)
```

```
In [15]: clear (Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)
```

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSSES_RECOMECCO-Copy1

```

In [16]: # ---BLOCO QUE VEM DO OUTRO SCRIPT-----
Q1n= 5.004965317077119
Dcn= 0.40040095030252665
A3n= 0.28748726629497784

Vcc = 12
pos = 3

fs = 40000.0
#####

rLf = 0
rLs = 0
rCs = 0
rCp = 0
rCh = 0

ILsoArb = 31.68213405
VCsoArb = 53.0465706
A1Arb = 0.99453099
A2Arb = 0.22164779
Dc= Dcn
VCpo=0
Q1= Q1n
A3=A3n

In [17]: def classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo):

    IC=[ILfo,ILso,VCso,VCpo]

    AI=np.array([[0,0,0,0],[0,-A1/Q1,-A1,0],[0,A1,0,0],[0,0,0,0]])
    BI=np.array([[1],[0],[0],[0]])
    CI=np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]],[0,0,0,1],[0,0,0,1]])
    DI=np.array([[0],[0],[0],[0],[0]])
    systemI = sgn.StateSpace(AI,BI,CI,DI)

    t=[0,Dc*2*pi]
    u=[1,1]
    toutI, yI, xI=systemI.output(u,t,IC)

    AII = np.array ([[0,0,0,-A3],[0,-A1/Q1,-A1,A2],[0,A1,0,0],[A3,-A2,0,0]])
    BII = np.array([[1],[0],[0],[0]])
    CII = np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1],[0,0,0,1]])
    DII = np.array ([[0],[0],[0],[0],[0]])

    systemII=sgn.StateSpace(AII,BII,CII,DII)

    IC2=xI[1]

    t=[0,(1-Dc)*2*pi]
    u=[1,1]
    toutII, yII, xII=systemII.output(u,t,IC2)

    return xI[1], xII[1], systemI, systemII

def classE1(X):

    ILso=X[0]
    VCso=X[1]
    A1=X[2]
    A2=X[3]
    ILfo=(A2/A3)*ILso
    IC = [ILfo,ILso,VCso,VCpo]
    IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[1]
    eq1=IC2-IC

    return eq1

X = fsolve(classE1,[ILsoArb,VCsoArb,A1Arb,A2Arb]) #ILso,VCso,A1,A2 arbitrados
#print (X) ## Devolve ILso,VCso,A1 e A2 calculados

In [18]: ILso=X[0]
VCso=X[1]
A1=X[2]
A2=X[3]
ILfo=(A2/A3)*ILso

ICo=[ILfo,ILso,VCso,VCpo]

```

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSSES_RECOMECCO-Copy1

```

In [19]: systemI=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[2]

systemII=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[3]

IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[1]

In [24]: t1=np.linspace(0,Dc*2*pi,1000)
u1 = np.ones_like(t1)
tout1, y1, x=systemI.output(u1,t1,ICo)
#plt.plot(tout1, y1[:,0])
t2=np.linspace(0,(1-Dc)*2*pi,1000)
u2 = np.ones_like(t2)*1
IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo)[0]
tout2, y2, x=systemII.output(u2,t2,IC2)
#plt.plot(tout2+Dc*2*pi, y2[:,0])
#plt.axhline(ICo[0],color='red',lw=1) #ICo[0] é o valor da corrente ILfo inicial da primeira etapa
#plt.axhline(IC2[0],color='red',lw=1) #IC2[0] é o valor da corrente ILfo inicial da segunda etapa
#plt.axvline(Dc*2*pi,color='red',lw=1)
#plt.axvline(1*2*pi,color='red',lw=1)
#plt.axvline(0,color='red',lw=2)
#plt.axvline(Dc*2*pi,color='red',lw=1)

In [601]: #Corrente na carga, saída
plt.plot(tout1, y1[:,1])
plt.plot(tout2+Dc*2*pi, y2[:,1])
plt.axhline(ICo[1],color='red',lw=1) #Corrente ILso inicial da primeira etapa
plt.axhline(IC2[1],color='red',lw=1) #Corrente ILso inicial da segunda etapa
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)

In [602]: Tensão sobre capacitor de saída
plt.figure()
plt.plot(tout1, y1[:,2])
plt.plot(tout2+Dc*2*pi, y2[:,2])
plt.axhline(ICo[2],color='red',lw=1)
plt.axhline(IC2[2],color='red',lw=1)
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)

In [596]: Tensão sobre Chave/Capacitor
plt.figure()
plt.plot(tout1, y1[:,3])
plt.plot(tout2+Dc*2*pi, y2[:,3])
plt.axhline(ICo[3],color='red',lw=1)
plt.axhline(IC2[3],color='red',lw=1)
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
plt.show()

In [25]: po_R_Vcc2=(A3**2*A1**2)/(A2**2*Q1**2)*1/(2*pi)*(np.trapz((y1[:,1])**2,tout1)+np.trapz((y2[:,1])**2,tout2))
a=1/po_R_Vcc2
#print(po_R_Vcc2)
#print(a)

```

$$\omega_1 = \frac{1}{\sqrt{L_s C_s}} \quad \omega_2 = \frac{1}{\sqrt{L_s - C_p}} \quad \omega_3 = \frac{1}{\sqrt{L_f - C_p}}$$

$$A_1 = \frac{\omega_1}{\omega} \quad , \quad A_2 = \frac{\omega_2}{\omega} \quad , \quad A_3 = \frac{\omega_3}{\omega} \quad , \quad \frac{C_p}{C_s} = \frac{\omega_1^2}{\omega_2^2} = \frac{A_1^2}{A_2^2} \quad \text{and} \quad \frac{L_s}{L_f} = \frac{\omega_3^2}{\omega_2^2} = \frac{A_3^2}{A_2^2}$$

Series quality factor is defined as

$$Q_1 = \frac{\omega_1 L_s}{R_s} = \frac{1}{\omega_1 C_s R_s}$$

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSES_RECOMEKO-Copy1

```

In [27]: #####----- REPETIÇÃO DO CÁLCULO COMA AS NOVAS MATRIZES CI E CII----- #####
def classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a):

    IC=[ILfo,ILso,VCso,VCpo]

    AI=np.array([[0,0,0,0],[0,-A1/Q1,-A1,0],[0,A1,0,0],[0,0,0,0]])
    BI=np.array([[1],[0],[0],[0]])
    CInew=np.array([[ (a*A1*A3**2)/(Q1*A2**2),0,0,0],[0,(a*A1*A3)/(Q1*A2),0,0],[0,0,(A1*A3)/A2,0],[0,0,0,0]])
    ##([[1,0,0,0],[0,1,0,0],[0,0,(A1*A3)/A2,0],[0,0,0,0]])
    DI=np.array([[0],[0],[0],[0]])
    systemI = sgn.StateSpace(AI,BI,CInew,DI)

    t=[0,Dc*2*pi]
    u=[1,1]
    toutI, yI, xI=systemI.output(u,t,IC)

    AII = np.array ([[0,0,0,-A3],[0,-A1/Q1,-A1,A2],[0,A1,0,0],[A3,-A2,0,0]])
    BII = np.array([[1],[0],[0],[0]])
    CIInew = np.array([[ (a*A1*A3**2)/(Q1*A2**2),0,0,0],[0,(a*A1*A3)/(Q1*A2),0,0],[0,0,(A1*A3)/A2,0],[0,0,0,A3]])
    ##([[1,0,0,0],[0,1,0,0],[0,0,(A1*A3)/A2,0],[0,0,0,A3]])
    DII = np.array ([[0],[0],[0],[0]])

    systemII=sgn.StateSpace(AII,BII,CIInew,DII)

    IC2=xI[1]

    t=[0,(1-Dc)*2*pi]
    u=[1,1]
    toutII, yII, xII=systemII.output(u,t,IC2)

    return xI[1], xII[1], systemI, systemII

def classE1(X):

    ILso=X[0]
    VCso=X[1]
    A1=X[2]
    A2=X[3]
    ILfo=(A2/A3)*ILso
    IC = [ILfo,ILso,VCso,VCpo]
    IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[1]
    eq1=IC2-IC

    return eq1

#X = fsolve(classE1,[ILsoArb,VCsoArb,A1Arb,A2Arb]) #ILso,VCso,A1,A2 arbitrados
#print (X) ## Devolve ILso,VCso,A1 e A2 calculados

```

```

In [523]: ILso=X[0]
VCso=X[1]
A1=X[2]
A2=X[3]
ILfo=(A2/A3)*ILso

ICo=[ILfo,ILso,VCso,VCpo]

```

```

In [524]: systemI=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[2]

systemII=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[3]

IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[1]

```

```

In [595]: t1=np.linspace(0,Dc*2*pi,1000)
u1 = np.ones_like(t1)
tout1, y1, x=systemI.output(u1,t1,ICo)
plt.plot(tout1, y1[:,0])
t2=np.linspace(0,(1-Dc)*2*pi,1000)
u2 = np.ones_like(t2)*1
IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[0]
tout2, y2, x=systemII.output(u2,t2,IC2)
plt.plot(tout2+Dc*2*pi, y2[:,0])
plt.axhline(ICo[0],color='red',lw=1) #ICo[0] é o valor da corrente Ilfo inicial da primeira etapa
plt.axhline(IC2[0],color='red',lw=1) #IC2[0] é o valor da corrente Ilfo inicial da segunda etapa
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(1*2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.figure()

```

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSSES_RECOMECO-Copy1

```
In [593]: Corrente na carga, saída
plt.plot(tout1, y1[:,1])
plt.plot(tout2+Dc*2*pi, y2[:,1])
plt.axhline(ICo[1],color='red',lw=1) #Corrente ILso inicial da primeira etapa
plt.axhline(IC2[1],color='red',lw=1) #Corrente ILso inicial da segunda etapa
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
```

```
In [592]: Tensão sobre capacitor de saída
plt.figure()
plt.plot(tout1, (Vcc/Vcc)*y1[:,2])
plt.plot(tout2+Dc*2*pi, (Vcc/Vcc)*y2[:,2])
plt.axhline(ICo[2],color='red',lw=1)
plt.axhline(IC2[2],color='red',lw=1)
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
```

```
In [591]: Tensão sobre Chave/Capacitor
plt.figure()
plt.plot(tout1, Vcc*y1[:,3])
plt.plot(tout2+Dc*2*pi, Vcc*y2[:,3])
plt.axhline(ICo[3],color='red',lw=1)
plt.axhline(IC2[3],color='red',lw=1)
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
plt.show()
```

```
In [30]: f = fs
po = pos
ws = 2*pi*f
w1 = ws*A1
w2 = ws*A2
w3 = ws*A3
Rs = (Vcc**2)/(a*po)
#Rs = Rsn
Ls = Q1/A1*Rs/ws
Cs = 1/(Q1*A1*ws*Rs)
Cp = Cs*((A1**2)/(A2**2)) #A1/(Q1*A1*ws*Rs) #Cs*((A1**2)/(A2**2))
Lf = Ls*(A2**2/A3**2) #A2**2/A3**2*Q1/A1*Rs/ws #Ls*(A2**2/A3**2)
#print('Vcc =',Vcc)
#print('Dc =',Dc)
#print('f =',f)
#print('Ls =',Ls)
#print('Cs =',Cs)
#print('Rs =',Rs)
#print('Cp =',Cp)
#print('Lf =',Lf)
#print('a =',a)
#print('Q1=',Q1)
```

```
In [590]: t1=np.linspace(0,Dc*2*pi,1000)
u1 = np.ones_like(t1)
tout1, y1, x=systemI.output(u1,t1,ICo)
plt.plot(tout1, (Vcc/(Rs*a))*y1[:,0])
t2=np.linspace(0,(1-Dc)*2*pi,1000)
u2 = np.ones_like(t2)*1
IC2=classE(Dc,Q1,A1,A2,A3,ILfo,ILso,VCso,VCpo,a)[0]
tout2, y2, x=systemII.output(u2,t2,IC2)
plt.plot(tout2+Dc*2*pi, (Vcc/(Rs*a))*y2[:,0])
plt.axhline(ICo[0],color='red',lw=1) #ICo[0] é o valor da corrente ILfo inicial da primeira etapa
plt.axhline(IC2[0],color='red',lw=1) #IC2[0] é o valor da corrente ILfo inicial da segunda etapa
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(1*2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.figure()
```

```
In [589]: Corrente na carga, saída
plt.plot(tout1, ((Vcc/(Rs*a))/(Vcc/(Rs*a)))*y1[:,1])
plt.plot(tout2+Dc*2*pi, ((Vcc/(Rs*a))/(Vcc/(Rs*a)))*y2[:,1])
plt.axhline(ICo[1],color='red',lw=1) #Corrente ILso inicial da primeira etapa
plt.axhline(IC2[1],color='red',lw=1) #Corrente ILso inicial da segunda etapa
plt.axvline(Dc*2*pi,color='red',lw=1)
plt.axvline(2*pi,color='red',lw=1)
plt.axvline(0,color='red',lw=1)
```

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSES_RECOMEKO-Copy1

```
In [33]: Icc = Vcc/(Rs*a)
Pcc = Vcc*Icc
ilfrms = np.sqrt((1/(2*pi))*(np.trapz(((Vcc/(Rs*a))*y1[:,0])**2,tout1)+np.trapz(((Vcc/(Rs*a))*y2[:,0])**2,tout2)))# Valor RMS de ilf
#rLf = 0 #Resistência ohmica do indutor de entrada
PrLf = (ilfrms**2)*rLf # Potência dissipada sobre resistência do indutor
ilsrms = np.sqrt((1/(2*pi))*(np.trapz(((Vcc/(Rs*a))*y1[:,1])**2,tout1)+np.trapz(((Vcc/(Rs*a))*y2[:,1])**2,tout2)))# Valor RMS de ils
#rLs = 0 #Resistência ohmica do indutor de saída
PrLs = (ilsrms**2)*rLs # Potência dissipada sobre indutor de saída
#vCsrms = np.sqrt((1/(2*pi))*(np.trapz((Vcc*y1[:,2])**2,tout1)+np.trapz((Vcc*y2[:,2])**2,tout2)))# Valor RMS de vCs
iCsrms = ilsrms
#rCs = 0#Resistência ohmica do capacitor de saída
PrCs = (iCsrms**2)*rCs # Potência dissipada sobre capacitor de saída
#vCprms = np.sqrt((1/(2*pi))*(np.trapz((Vcc*y1[:,3])**2,tout1)+np.trapz((Vcc*y2[:,3])**2,tout2)))# Valor RMS de vCp
iCprms = np.sqrt((1/(2*pi))*(np.trapz(((Vcc/(Rs*a))*y2[:,0]-y2[:,1])**2,tout2)))
#rCp = 0 #Resistência ohmica da chave
PrCp = (iCprms**2)*rCp # Potência dissipada sobre chave
iChrms = np.sqrt((1/(2*pi))*(np.trapz(((Vcc/(Rs*a))*y1[:,0]-y1[:,1])**2,tout1))) ##Corrente RMS na chave
#rCh = 0 # RDS on da chave
PrCh = (iChrms**2)*rCh
ptt = PrLf+PrLs+PrCs+PrCp+PrCh
rendimento = po/(po+ptt)
#print('Po:',po,'W')
#print('')
#print('PrLf:',PrLf)
#print('PrLs:',PrLs)
#print('PrCs:',PrCs)
#print('PrCp:',PrCp)
#print('PrCh:',PrCh)
#print('')
#print('ptt:',ptt)
#print('')
#print('Rendimento:',rendimento)
```

```
In [34]: Ilfmed = (1/(2*pi))*(np.trapz(((Vcc/(Rs*a))*y1[:,0]),tout1)+np.trapz(((Vcc/(Rs*a))*y2[:,0]),tout2))# Valor Médio de ilf
ILsmed = (1/(2*pi))*(np.trapz(((Vcc/(Rs*a))*y1[:,1]),tout1)+np.trapz(((Vcc/(Rs*a))*y2[:,1]),tout2))# Valor Médio de ils
vCsmed = (1/(2*pi))*(np.trapz((Vcc*y1[:,2]),tout1)+np.trapz((Vcc*y2[:,2]),tout2))# Valor Médio de vCs
vCpmed = (1/(2*pi))*(np.trapz((Vcc*y1[:,3]),tout1)+np.trapz((Vcc*y2[:,3]),tout2))# Valor Médio de vCp
```

```
In [36]: #plt.plot(tout1, (Vcc/(Rs*a))*(y1[:,0]-y1[:,1]))
#plt.plot(tout2+Dc*2*pi, (Vcc/(Rs*a))*(y2[:,0]-y2[:,1]))
ilf1 = max( abs((Vcc/(Rs*a))*y1[:,0]))
ilf2 = max( abs((Vcc/(Rs*a))*y2[:,0]))
ilf = max(ilf1,ilf2)
ilfo1 = max( abs((Vcc/(Rs*a))*y1[:,1]))
ilfo2 = max( abs((Vcc/(Rs*a))*y2[:,1]))
ilfo = max(ilfo1,ilfo2)
vco1 = max( abs(Vcc*y1[:,2]))
vco2 = max( abs(Vcc*y2[:,2]))
vco = max(vco1,vco2)
vcp1 = max( abs(Vcc*y1[:,3]))
vcp2 = max( abs(Vcc*y2[:,3]))
vcp = max(vcp1,vcp2)

ich = abs((Vcc/(Rs*a))*(y1[:,0]-y1[:,1]))
#print (max(ich))
#Corrente conjunta na chave e capacitor
```

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSES_RECOMECO-Copy1

```

In [ ]: print('Vcc =',Vcc)
print('Dc =',Dcn)
print('f =',f)
print('Ls =',Ls)
print('Cs =',Cs)
print('Rs =',Rs)
print('Cp =',Cp)
print('Lf =',Lf)
print('rLf=',rLf)
print('rLs=',rLs)
print('rCs=',rCs)
print('rCp=',rCp)
print('rCh=',rCh)
print('')
print('')

print('a =',a)
print('T =',1/a)
print('Q1=',Q1n)
print('A1=',A1)
print('A2=',A2)
print('A3=',A3n)

print('')
print('')
print('iLfrms=',iLfrms)
print('iLsrms=',iLsrms)
print('iCsrms=',iCsrms)
print('iCprms=',iCprms)
print('iChrms=',iChrms)
print('Perdas totais=',ptt)
print('')
print('')

print('Pot Lf =',PrLf)
print('Pot Ls =',PrLs)
print('Pot Cs =',PrCs)
print('Pot Cp =',PrCp)
print('Pot Ch =',PrCh)
print('')
print('')
print('Rendimento=',rendimento)
print('')
print('')
print('Valores de Pico:')

print('ilf =',ilf)
print('ilfo =',ilfo)
print('vco =',vco)
print('vcp/vch =',vcp)
print('ich =',max(ich))
print('')
print('PotOut =',(iLsrms**2)*Rs))
print('Icc',Icc)
print('Pcc',Pcc)
print('')
print('soma =',(ilf + ilfo + vco +vcp ))

```

```

In [ ]: #BLOCO RETIFICADOR###
Vo = 5
Rcarga = (Vo**2)/Pcc
Io = Vo/Rcarga

phirad = np.arcsin(np.sqrt(Rs/(2*Rcarga)))

phi = (phirad*180)/pi

print('Rs =',Rs)
print('Rcarga =',Rcarga)
print('Rs/Rcarga =',Rs/Rcarga)
print('phi =',phi)
print('Io =',Io)

```

```

In [ ]: print('Rcarga =',Rcarga)
print('phirad =',phirad)
print('phi =',phi)
#print('Mr =',Mr)
#print('Rdc =',Rdc)
#print('M =',M)

```

```

In [575]: Dd = 0.7633
phiteste = (np.arctan((1 - np.cos(2*pi*Dd)) / (2*pi*(1-Dd) + np.sin(2*pi*Dd))) *180 )/pi
print('phi =',phiteste)

phi = 61.83496741117498

```

12/06/2021

ESTUDO_SEMANA_17_08-ESTRESSES_RECOMEKO-Copy1

```
In [ ]: #Dd = 0.5
#phi = 32.48
#phirad = (phi*pi)/180

P1 = np.sin(phirad)**2/pi #0 argumento deve ser em radianos, por isso uso "form" e não "phi"
P2 = np.cos(2*pi*Dd)
P3 = 2*(pi**2)*((1-Dd)**2)
P4 = (2*pi*(1-Dd)+np.sin(2*pi*Dd))**2
P5 = 1-np.cos(2*pi*Dd)
wC3Ri = P1*(1-P2-P3+(P4/P5))
wC3Ri
```

```
In [ ]: P6 = 1-np.cos(2*pi*Dd)
P7 = 2*(pi**2)*((1-Dd)**2)
P8 = (2*pi*(1-Dd) + np.sin(2*pi*Dd))**2
P9 = 1-np.cos(2*pi*Dd)
wC3RL = (1/(2*pi))*(P6-P7 + (P8/P9))
wC3RL
```

```
In [ ]: P10 = pi*(1-Dd)+np.sin(2*pi*Dd)
P11 = (1/4)*(np.sin(4*pi*Dd)*np.cos(2*phirad))
P12 = (1/2)*(np.sin(2*phirad)*(np.sin(2*pi*Dd)**2))
P13 = 2*pi*(1-Dd)*np.sin(phirad)*np.sin((2*pi*Dd)-phirad)
CipC3 = pi*(1/(P10-P11-P12-P13))
CipC3
```

```
In [ ]: C3a = wC3Ri/(ws*Rs)
C3b = wC3RL/(ws*Rcarga)
Ci = CipC3 * C3a
Cs2 = (Cs*Ci)/(Ci-Cs)
Dil = 0.1

Lfilter = ((1-Dd)*Vo)/(Dil*f*Io)

Cfilter = 25/((pi**2)*(f**2)*Lfilter)

print (C3a)
print (C3b)
```

```
In [ ]: ##Relatório Retificador##
print ('Resistencia Rdc = ',Rdc)
print ('Tensão RMS do inversor = ',Vrms)
print ('Ganho do inversor = ',Ma)
print ('Ganho do Restificador = ',Mr)
print ('Ganho Geral 1 = ',M1)
print ('Ganho Geral 2 = ',M2)
print ('Ganho Geral 3 = ',M3)
print ('Angulo phi = ',phi,'°')
print ('Rs (Ri) / Rcarga(RL) = ',RelRes)
print ('D do retificador = ',Dd)
print ('wC3Rs (Ri) = ',wC3Ri)
print ('wC3RL (Rcarga) = ',wC3RL)
print ('Angulo phi em radianos = ',phirad)
print ('Ci/C3 = ',CipC3)
print ('Ci = ',Ci)
print ('C3 = ',C3b)
print ('Cs2 = ',Cs2)
print ('Io = ',Io)
print ('Lfilter = ',Lfilter)
print ('Cfilter = ',Cfilter)
print ('Rcarga = ',Rcarga)
print ('PoCalc = ',PoCalc)
print ('pos = ',pos)
```

```
In [ ]: #Para transferir para o PSIM#
print('Vcc = ',Vcc)
print('Dc = ',Dcn)
print('f = ',f)
print('Ls = ',Ls)
print('Cs = ',Cs)
print('Rs = ',Rs)
print('Cp = ',Cp)
print('Lf = ',Lf)
print('C3 = ',C3a)
print('Cs2 = ',Cs2)
print('Lfilter = ',Lfilter)
print('Cfilter = ',Cfilter)
print('Rcarga = ',Rcarga)
print('Ci = ',Ci)
print('Im = ',ilfo)
print('')
print('Q1n=', Q1n)
print('Dcn=', Dcn)
print('A3n=', A3n)
```


12/06/2021

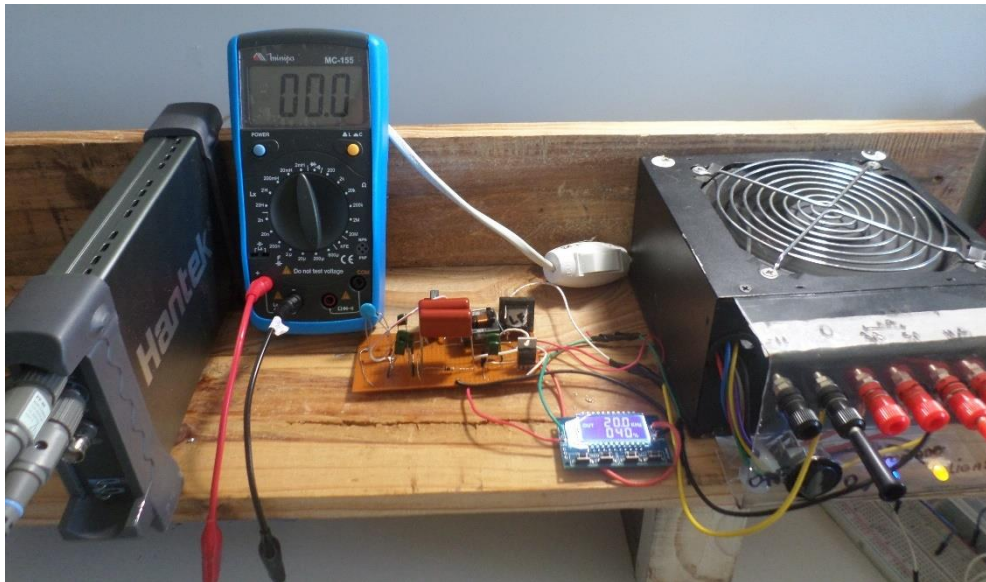
ESTUDO_SEMANA_17_08-ESTRESSES_RECOMECO-Copy1

```
In [ ]: ##Método de cálculo 2 do retificador ##  
## phi, phirad, Vo, Io, RL (Rcarga) são os mesmos acima calculados##  
Vrms = iLsrms*Rs  
MA = Vrms/Vcc  
Rdc = a*Rs  
MR = Vo/Vrms  
M = MR*MA  
Vrms
```

```
In [ ]: print ('Ganho do inversor:',MA)  
print ('Ganho do Retificador:',MR)  
print ('Ganho Total:',M)
```

APENDICE C – Principais materiais utilizados e foto da implementação

Material	Identificação
Chave	Mosfet IRFZ44N
Diodo	MUR460 - Diodo Retificador Ultra-Rápido 4A 600V
Medidor LCR	Minipa MX-155
Osciloscópio	Hantek 6022BE
Gerador PWM	Módulo gerador de sinal PWM 1Hz-150kHz – M2D1L01-A
Fonte ATX	400W
Multímetro	Minipa ET-3200B
Placa de fenolite	Placa de fenolite perfurada trilha 5x10
Protoboard	-

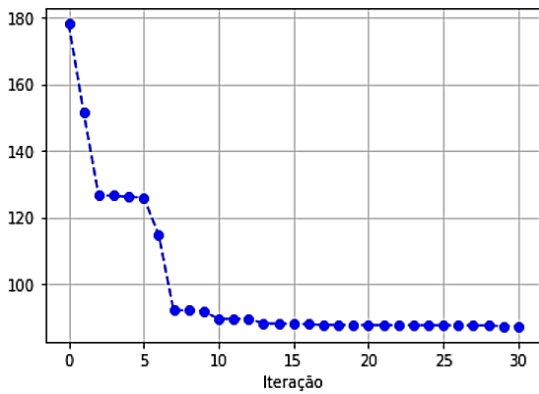
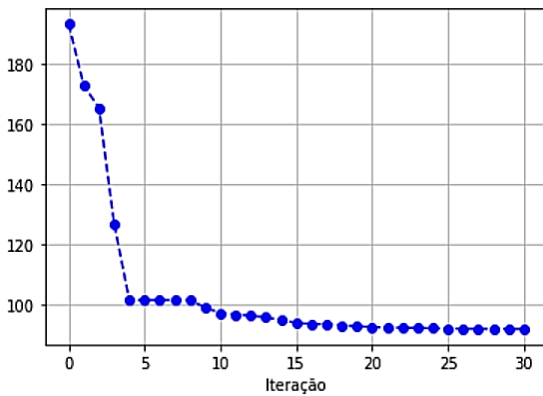
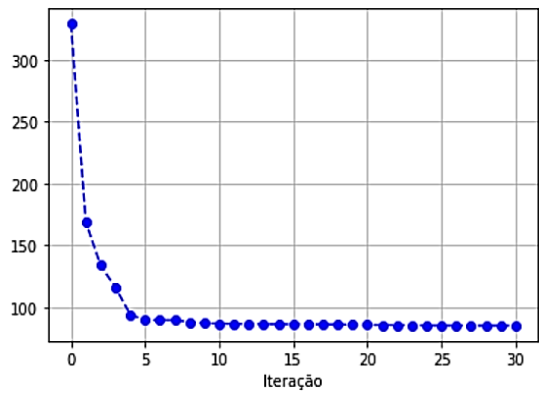
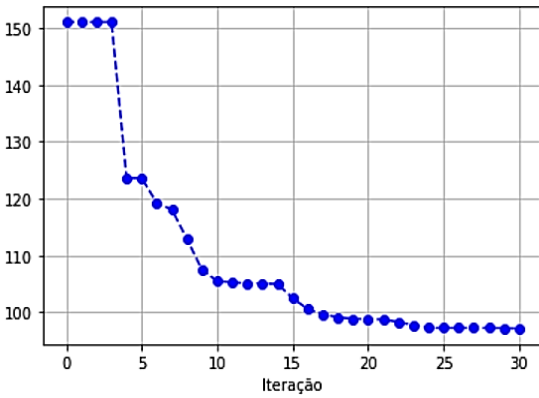
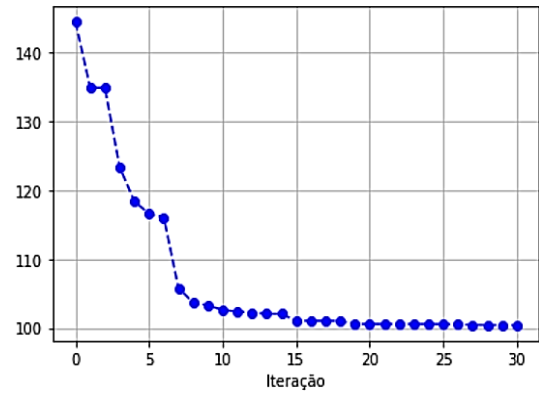
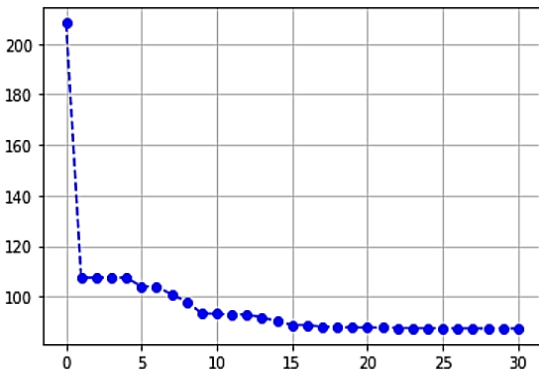
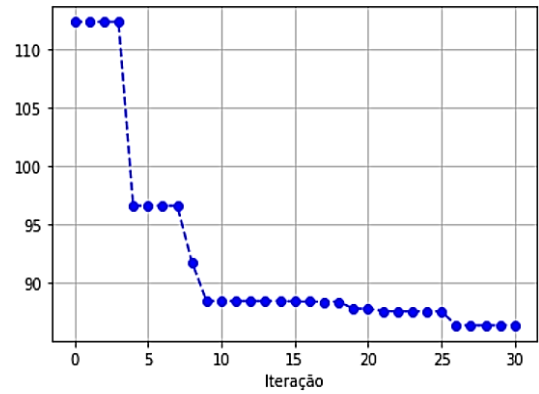
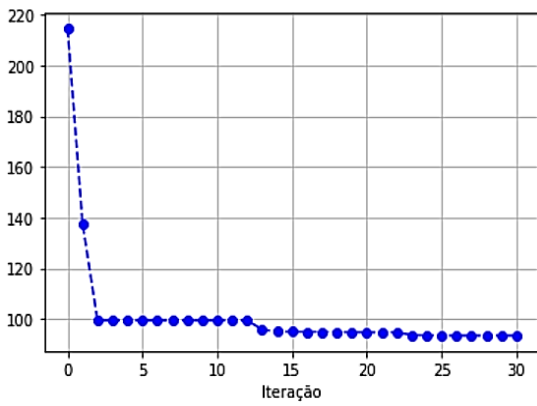


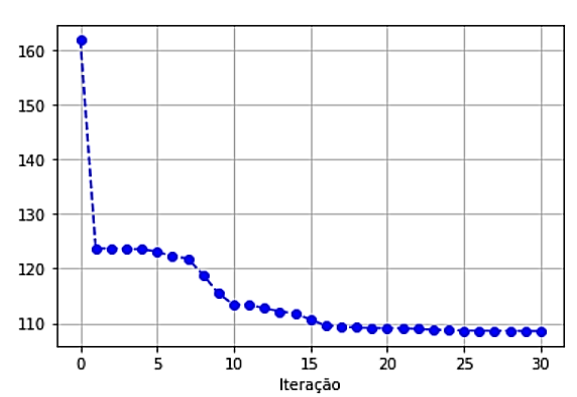
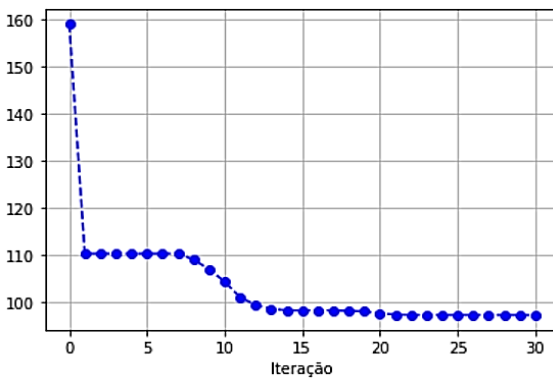
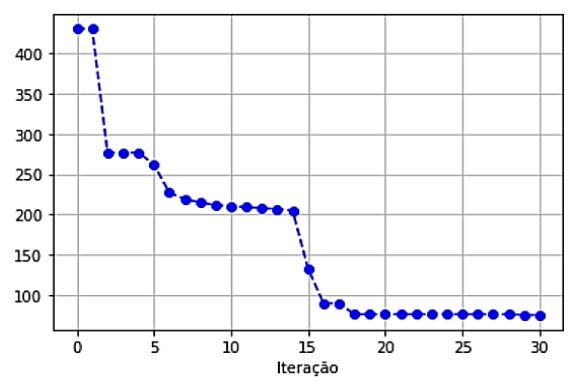
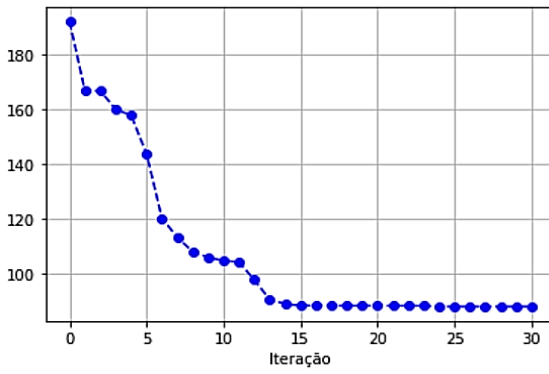
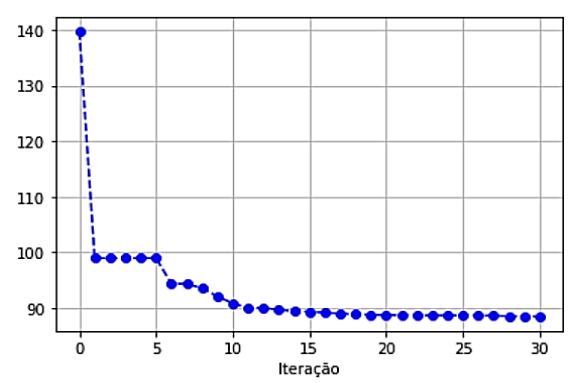
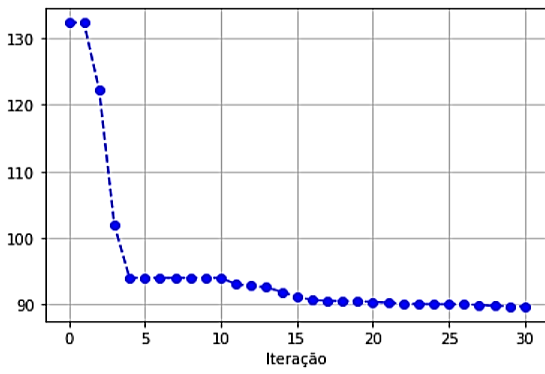
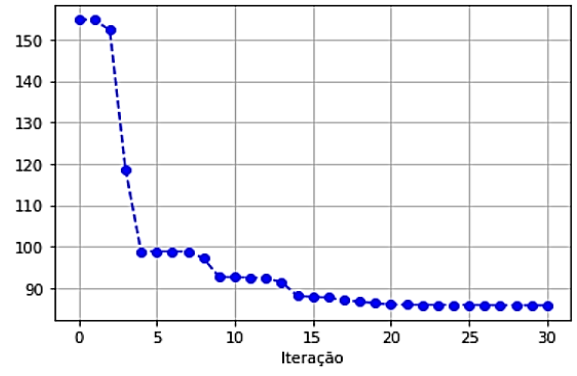
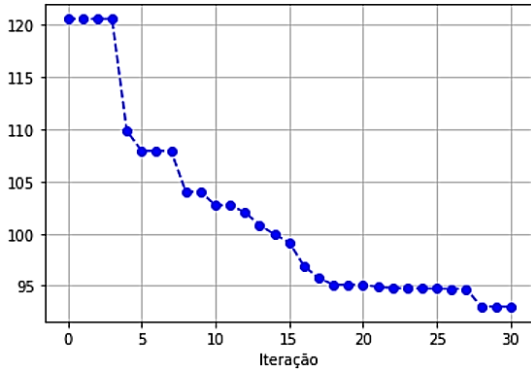
APENDICE D – Valores individuais dos parâmetros e da função objetivo para cada execução.

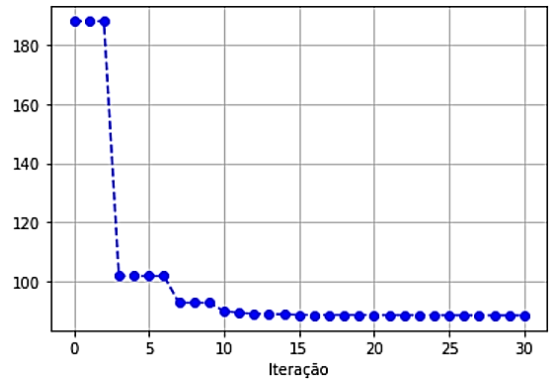
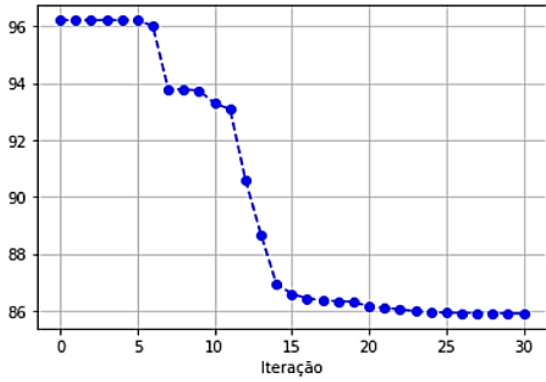
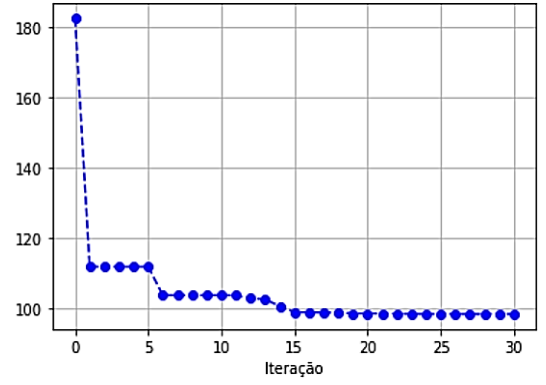
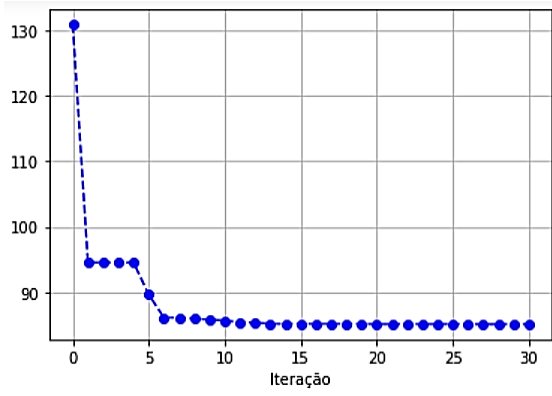
Execução	Valores da função objetivo	Parâmetros
1	93,4314	$Q_1 = 5,001$
		$D_c = 0,4150$
		$A_3 = 0,5390$
2	86,3072	$Q_1 = 5,0200$
		$D_c = 0,4040$
		$A_3 = 0,2830$
3	87,4253	$Q_1 = 5,000$
		$D_c = 0,4030$
		$A_3 = 0,4019$
4	100,37	$Q_1 = 5,010$
		$D_c = 0,4585$
		$A_3 = 0,3588$
5	97,1129	$Q_1 = 5,002$
		$D_c = 0,400$
		$A_3 = 0,7355$
6	85,244	$Q_1 = 5,001$
		$D_c = 0,4026$
		$A_3 = 0,2525$
7	87,187	$Q_1 = 5,002$
		$D_c = 0,4030$
		$A_3 = 0,3798$
8	85,8949	$Q_1 = 5,078$
		$D_c = 0,4000$
		$A_3 = 0,2985$
9	92,029	$Q_1 = 5,095$
		$D_c = 0,4168$
		$A_3 = 0,4257$
10	108,543	$Q_1 = 7,4548$

		$D_c = 0,4024$
		$A_3 = 0,2904$
11	98,4811	$Q_1 = 98,4811$
		$D_c = 0,4473$
		$A_3 = 0,4223$
12	97,19	$Q_1 = 5,000$
		$D_c = 0,400$
		$A_3 = 0,7375$
13	92,984	$Q_1 = 5,6853$
		$D_c = 0,4052$
		$A_3 = 0,3121$
14	85,839	$Q_1 = 5,019$
		$D_c = 0,4005$
		$A_3 = 0,3283$
15	88,616	$Q_1 = 5,006$
		$D_c = 0,4020$
		$A_3 = 0,4727$
16	89,667	$Q_1 = 5,011$
		$D_c = 0,4212$
		$A_3 = 0,2448$
17	88,425	$Q_1 = 5,000$
		$D_c = 0,4152$
		$A_3 = 0,2740$
18	74,546	$Q_1 = 5,1618$
		$D_c = 0,4102$
		$A_3 = 0,1615$
19	87,954	$Q_1 = 5,3340$
		$D_c = 0,4007$
		$A_3 = 0,2613$
20	85,15	$Q_1 = 5,0046$
		$D_c = 0,4004$
		$A_3 = 0,2874$

APENDICE E – Gráficos das execuções de teste/resultados estatísticos.







ANEXO I – Fragmento do artigo *Resonant dc/dc Converter with Class-E Inverter and Class-E Rectifier*

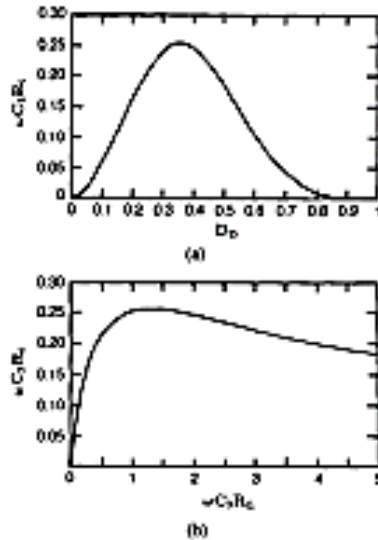


Fig. 7. Input resistance R_i normalized with respect to $X_{C1} = 1/\omega C_1$: (a) $\omega C_1 R_i$ versus D_D ; (b) $\omega C_1 R_i$ versus $\omega C_1 R_L$.

Neglecting power losses in the rectifier, $P_O = P_{oA}$, i.e., $V_O^2/R_L = V_{rms}^2/R_i$. Hence, using (3), the ac/dc voltage transfer function of the rectifier is

$$M_R = \frac{V_O}{V_{rms}} = \sqrt{\frac{R_L}{R_i}} = \frac{1}{\sqrt{2} \sin \phi}. \quad (7)$$

The values of M_R are tabulated in Table I as a function of D_D . For example, for $D_D = 0.4$, $R_i/R_L = 0.294$, and $M_R = 1/\sqrt{0.294} = 1.8443$.

Taking into account the overall converter efficiency η , the dc/dc voltage transfer function of the converter can be expressed as

$$M = \frac{V_O}{V_I} = \sqrt{\frac{R_1 R_L}{R_{dc} R_i}} = \sqrt{\eta} M_A M_R. \quad (8)$$

For instance, at $D = 0.5$, $Q_1 = 5$, and $D_D = 0.4$, $M = 0.7245 \cdot 1.8443 \cdot \sqrt{\eta} = 1.3353 \cdot \sqrt{\eta}$. A transformer may be added to obtain a desired value of M .

V. DESIGN PROCEDURE

Design a dc/dc converter with a minimum operating frequency of 1 MHz, which will boost a dc input voltage from 20 to 25 V for load resistances ranging from 70 Ω to ∞ .

It is sufficient to design the converter for the full-load resistance $R_{Lmin} = 70 \Omega$ because peak values of all currents and voltages decrease with R_L . Assume the overall converter efficiency $\eta = 85$ percent. From power relationships, we have

$$\begin{aligned} P_O &= V_O^2/R_L = 25^2/70 = 8.93 \text{ W}; \\ P_I &= P_O/\eta = 8.93/0.85 = 10.51 \text{ W}; \\ I_I &= P_I/V_I = 10.51/20 = 0.526 \text{ A}; \text{ and} \\ I_O &= V_O/R_L = 25/70 = 0.347 \text{ A}. \end{aligned}$$

Assume the optimum class-E operation for the full load at $f = 1$ MHz, $D = 0.5$ (transistor on-switch duty cycle), and

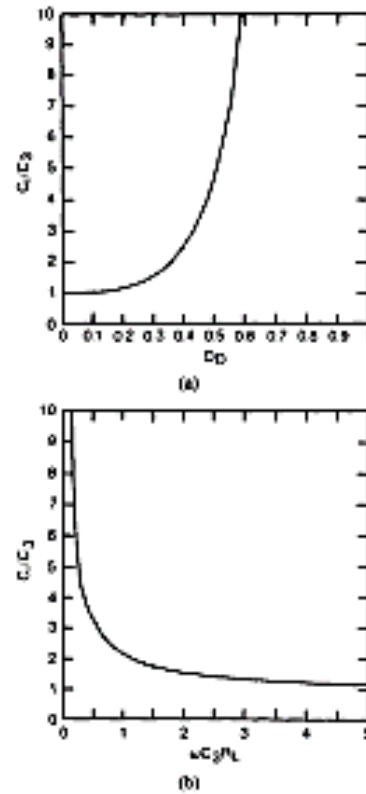


Fig. 8. Normalized input capacitance C_1/C_2 of the rectifier: (a) C_1/C_2 versus D_D ; (b) C_1/C_2 versus $\omega C_1 R_L$.

TABLE I
PARAMETERS OF CLASS-E RECTIFIER

D_D	ϕ	$\omega C_1 R_L$	I_{om}/I_o	V_{rms}/V_o	R_i/R_L	$\omega C_1 R_i$	C_1/C_2	M_R
0	0	∞	∞	0	0	0	1	∞
0.1	1.75	29.902	19.40	2.019	0.8019	0.0980	1.0253	23.124
0.2	0.33	8.2295	8.318	2.374	0.8264	0.1669	1.1905	6.1876
0.25	0.93	5.8625	6.712	2.401	0.8595	0.2132	1.3167	4.1087
0.3	13.75	3.1481	5.297	2.566	0.8138	0.2437	1.5463	2.9149
0.4	22.54	1.6376	3.688	2.976	0.7918	0.2433	2.4413	1.8443
0.5	32.48	0.9185	2.862	3.563	0.7578	0.1636	4.7259	1.3167
0.6	43.22	0.5114	2.490	4.447	0.6577	0.1045	11.851	1.0253
0.7	54.43	0.3016	2.228	5.827	1.2254	0.0419	42.844	0.8686
0.75	66.28	0.1847	2.151	7.112	1.8065	0.0221	190.38	0.8142
0.8	80.14	0.0958	2.093	8.869	1.6728	0.0097	393.88	0.7720
0.9	116.02	0.0435	2.022	11.75	1.5138	0.0064	8827.6	0.7229
1	90	0	2	∞	2	0	∞	0.7071

$Q_1 = 5$. From [3], $P_I/R_i/V_I^2 = R_i/R_{dc} = 0.5249$, $\omega R_i C_1 = 0.2067$, $\omega R_i C = 0.2269$, $\omega L/R_i = 6.673$, $I_{SM}/I_I = 2.783$, and $V_{SM}/V_I = 3.610$. Thus

$$\begin{aligned} R_i &= 0.5249 V_I^2 / P_I = 0.5249 \cdot 25^2 / 10.51 = 19.98 \Omega, \text{ let} \\ R_i &= 20 \Omega; \\ C_1 &= 0.2067 / \omega R_i = 0.2067 / (2\pi \cdot 10^6 \cdot 20) = 1.645 \text{ nF}; \\ C &= 0.2269 / \omega R_i = 0.2269 / (2\pi \cdot 10^6 \cdot 20) = 1.806 \text{ nF}; \\ L_r &= 5.673 R_i / \omega = 5.673 \cdot 20 / (2\pi \cdot 10^6) = 18.06 \mu\text{H}; \\ I_{SM} &= 2.783 I_I = 2.783 \cdot 0.526 = 1.462 \text{ A}; \\ V_{SM} &= 3.610 V_I = 3.61 \cdot 20 = 72.2 \text{ V}; \text{ and} \\ I_m &= \sqrt{2 P_O / R_i} = \sqrt{2 \cdot 10.51 / 20} = 1.025 \text{ A}. \end{aligned}$$

The minimum value of the input filter inductance at which

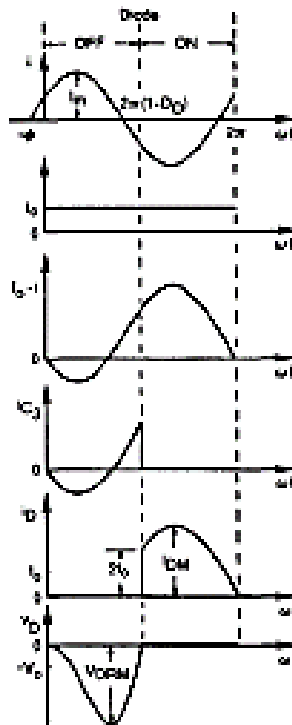


Fig. 15. Idealized current and voltage waveforms in the class-E rectifier.

Using the relation $I_O = V_O/R_L$ and (A4), one obtains the voltage across the capacitor C_3 and the rectifier diode

$$\begin{aligned} v_D &= \frac{1}{\omega C_3} \int_0^{\omega t} i_{C3} d(\omega t) \\ &= \frac{V_O}{\omega C_3 R_L} \left[\omega t + \frac{\cos(\omega t + \phi) - \cos \phi}{\sin \phi} \right]. \quad (\text{A5}) \end{aligned}$$

The rectifier diode turns off (ideally) when its voltage reaches zero, i.e., $v_D[2\pi(1-D_D)] = 0$. Substitution of this condition to (A5) yields the relationship between D_D and ϕ given by (1). Since the dc component of the voltage across L_f is zero, the average value of the diode voltage is

$$\begin{aligned} V_{D,AV} &= -V_O = \frac{1}{2\pi} \int_0^{2\pi(1-D_D)} v_D d(\omega t) \\ &= \frac{V_O}{2\pi\omega C_3 R_L} \left[2\pi^2(1-D_D)^2 - 1 + \cos 2\pi D_D \right. \\ &\quad \left. - \frac{2\pi(1-D_D) + \sin 2\pi D_D}{\tan \phi} \right]. \quad (\text{A6}) \end{aligned}$$

Simplifying this expression and using (1), one obtains the relationship between D_D and $\omega C_3 R_L$ given by (2).

Fig. 14(d) shows the model of the rectifier for the time interval during which the diode is on, i.e., for $2\pi(1-D_D) < \omega t < 2\pi$. From (A1) and (A3), the current through the diode is given by

$$i_D = I_O - i = I_O - I_m \sin(\omega t + \phi) = I_O \left[1 - \frac{\sin(\omega t + \phi)}{\sin \phi} \right]. \quad (\text{A7})$$

The input power of the rectifier contains only the fundamental component because the input current i is a sine wave. Therefore, it suffices to determine the input impedance of the rectifier at the fundamental frequency f . This impedance consists of an input resistance R_i and an input capacitance C_i connected in series, as shown in Fig. 1(b). The input voltage of the rectifier is $v_{DR} = -v_D$, where v_D is given by (A5). The input voltage v_{DR} can be expanded into the Fourier series in such a way that the sinusoidal component represents the voltage v_{R1} across R_i , and the cosinusoidal component represents the voltage v_{X1} across C_i at the frequency f . Thus, the fundamental component of the input voltage of the rectifier is

$$v_1 = v_{R1} + v_{X1} = V_{Rm1} \sin(\omega t + \phi) - V_{Xm1} \cos(\omega t + \phi) \quad (\text{A8})$$

where V_{Rm1} and V_{Xm1} are the amplitudes of v_{R1} and v_{X1} , respectively. Keeping in mind that $v_{DR} = -v_D$ and using (A5)

$$\begin{aligned} V_{Rm1} &= \frac{1}{\pi} \int_0^\pi v_{DR} \sin(\omega t + \phi) d(\omega t) \\ &= \frac{V_O}{2\pi\omega C_3 R_L} \left[\frac{1}{2} \cos \phi \sin 4\pi D_D + \sin \phi \right. \\ &\quad \cdot \left(1 - \cos 2\pi D_D + \frac{1}{2} \sin^2 2\pi D_D \right) \\ &\quad + 2\pi(1-D_D) \cos(2\pi D_D - \phi) \\ &\quad \left. + \frac{\cos \phi(1 - \cos 2\pi D_D + \frac{1}{2} \sin^2 2\pi D_D)}{\tan \phi} \right]. \quad (\text{A9}) \end{aligned}$$

For $D_D = 0.5$, $V_{Rm1} = (4/\sqrt{\pi^2 + 4})V_O = 1.074V_O$. The input resistance of the rectifier at the fundamental frequency f is defined as

$$R_i = \frac{V_{Rm1}}{I_m}. \quad (\text{A10})$$

Substitution of (A3) and (A9) into (A10) yields (3). Using (A5)

$$\begin{aligned} V_{Xm1} &= -\frac{1}{\pi} \int_0^{2\pi} v_{DR} \cos(\omega t + \phi) d(\omega t) = \frac{V_O}{\pi\omega C_3 R_L} \left\{ \frac{\cos \phi[\pi(1-D_D) + \sin 2\pi D_D - \frac{1}{2} \sin 4\pi D_D]}{\tan \phi} \right. \\ &\quad \left. + \sin \phi \left[\pi(1-D_D) + \frac{1}{4} \sin 4\pi D_D + \sin 2\pi D_D \right] - \cos \phi \sin^2 2\pi D_D + 2\pi(1-D_D) \sin(2\pi D_D - \phi) \right\}. \quad (\text{A11}) \end{aligned}$$

ANEXO II – Fragmento do livro RESONANT POWER CONVERTERS

If it is required that the peak-to-peak value of the ripple current cannot exceed $I_{r(max)}$, the minimum value of the filter inductance can be found as

$$L_{f(min)} = \frac{(1 - D_{max})V_O}{fI_{r(max)}}. \quad (4.26)$$

88 CLASS E LOW dv/dt RECTIFIERSTABLE 4.1 Parameters of Class E Low dv/dt Rectifier with a Parallel Capacitor

D	$\phi(^{\circ})$	ωCR_L	I_{DM}/I_O	V_{DM}/V_O	R_i/R_L	ωCR_i	C_i/C	M_{VR}
0	0	∞	∞	2	0	0	1	∞
0.05	0.45	125.34	39.69	2.022	0.0001	0.0152	1.0031	90.704
0.1	1.75	29.962	19.40	2.079	0.0019	0.0560	1.0233	23.124
0.15	3.83	12.398	12.48	2.164	0.0089	0.1109	1.0738	10.572
0.2	6.59	6.3295	8.918	2.274	0.0264	0.1669	1.1665	6.1578
0.25	9.93	3.5855	6.712	2.407	0.0595	0.2132	1.3167	4.1007
0.3	13.75	2.1481	5.207	2.566	0.1130	0.2427	1.5483	2.9749
0.4	22.54	0.8276	3.608	2.976	0.2940	0.2433	2.4443	1.8443
0.5	32.48	0.3183	2.862	3.562	0.5768	0.1836	4.7259	1.3167
0.6	43.21	0.1114	2.460	4.447	0.9377	0.1045	11.831	1.0327
0.7	54.49	0.0316	2.228	5.927	1.3254	0.0419	42.944	0.8686
0.75	60.28	0.0147	2.151	7.112	1.5085	0.0221	100.88	0.8142
0.8	66.14	0.0058	2.093	8.889	1.6728	0.0097	293.83	0.7732
0.9	78.02	0.0035	2.022	17.78	1.9138	0.0004	8837.6	0.7229
1	90	0	2	∞	2	0	∞	0.7071

Placing, as a rule of thumb, the corner frequency of the output filter $f_o = 1/(2\pi\sqrt{L_f C_f})$ at least a decade below the operating frequency of the rectifier f , the minimum value of the filter capacitor is given by

$$C_{f(min)} = \frac{25}{\pi^2 f^2 L_{f(min)}}. \quad (4.27)$$