

Desenvolvimento de um algoritmo computacional baseado em grafos para síntese de conversores cc-cc

André Luiz Silva Crivellari* André Salume Lima Ferreira Leão**
Marcia Helena Moreira Paiva* Walbermark Marques dos Santos*

* Programa de pós graduação em engenharia elétrica, Universidade
Federal do Espírito Santo, ES

** Engenheiro Eletricista, Empresa Aruna Energia Solar, Espírito
Santo, ES.

Abstract: In this paper, a computational algorithm made in the MATLAB MathWorks (2016) program for the synthesis of cc-cc converters is proposed. From the union of converter derivation works and determination of the voltage and current values of the elements involved, a set of converters is obtained that correspond to the designer's specifications, such as static gain, number of switches, galvanic isolation and if the input current and output is not pulsed.

Resumo: Neste artigo é proposto um algoritmo computacional feito no programa MATLAB MathWorks (2016) para síntese de conversores cc-cc. A partir da união de trabalhos de derivação de conversores e de determinação dos valores de tensão e corrente dos elementos envolvidos, é obtido um conjunto de conversores que atendem as especificações do projetista, tais como ganho estático, quantidade de chaves, isolamento e se a corrente de entrada e saída não é pulsada.

Keywords: dc-dc converters, synthesis, power electronics, graph theory, state space approach

Palavras-chaves: conversores cc-cc; síntese; eletrônica de potência; teoria de grafos, modelagem de conversores cc-cc PWM

LIST OF SYMBOLS

V_s	Matriz dos valores da tensão média dos semicondutores.	nl_{ca}	Quantidade de cordas do circuito de alta frequência.
V_{Ct}	Valor da tensão nos capacitores que estão como galhos no grafo.	Nl_{cc}	Matriz de incidência dos indutores do circuito de um conversor cc-cc completo.
q	Quantidade de fontes de tensão ou corrente linearmente independentes nos circuitos de alta frequência.	nl_{cc}	Quantidade de cordas do circuito de baixa frequência.
A	Matriz de incidência.	Ns_{cc}	Matriz de incidência dos semicondutores em um circuito de um conversor cc-cc completo.
A_f	Matriz de incidência escalonada	nt_{ca}	Quantidade de galhos das árvores do circuito de alta frequência.
A_l	Bloco das cordas da matriz de incidência.	nt_{cc}	Quantidade de galhos das árvores do circuito de baixa frequência.
A_t	Bloco dos galhos da matriz de incidência.	Q_f	Matriz de corte.
B_a	Matriz de volt-segundo dos semicondutores em um circuito de alta frequência.	R_l	Resistências que estão como cordas no grafo.
B_f	Matriz de circuitos fechados.	R_t	Resistências que estão como galhos no grafo.
d_i	Razão cíclica no intervalo i .	S_i	Semicondutor i .
G_l	Condutância das resistências que estão como cordas no grafo.	T_s	Período de chaveamento.
G_t	Condutância das resistências que estão como cordas no grafo.	V_{Ll}	Valor da tensão nos indutores no grafo.
I_{Ct}	Valor da corrente nos capacitores no grafo.	V_{Rl}	Valor da tensão nas resistências que estão como cordas no grafo.
I_g	Valor da corrente da fonte de tensão V_g .	V_{Rt}	Valor da tensão nas resistências que estão como galhos no grafo.
I_{Lt}	Valor da corrente nos indutores no grafo.	VS_i	Valor médio da tensão na chave i .
M_{ca}	Matrizes de incidência dos grafos de alta frequência.	vs_i	Valor instantâneo da chave i em um intervalo.
M_{cc}	Matrizes de incidência dos grafos de baixa frequência.	C	Conjunto composto por capacitores.
Nl_{cc}	Matriz de incidência do grafo em completo.	G	Conjunto composto por condutâncias.
n_g	Quantidade de fontes de tensão.	L	Conjunto composto por indutores.
n_s	Quantidade de elementos de chaveamento.	R	Conjunto composto por resistores.
		V_g	Conjunto composto por fontes de tensão.

V_s Matriz dos valores da tensão média dos semicondutores.

1. INTRODUÇÃO

Os conversores cc-cc são usados em diversas aplicações, existindo diferentes topologias para um mesmo ganho estático requisitado. Vários trabalhos na literatura desenvolvem procedimentos de síntese de diversas de topologias.

Zhou (2005) e Maksimovic (2007) propõem a aplicação da teoria de grafos para síntese geral de conversores cc-cc PWM, sendo base para outros trabalhos, como o de Lu et al. (2016), que criou um sistema de geração de conversores com tolerância de falta aplicando os princípios do trabalho de Zhou (2005). Outro trabalho nessa vertente, Gaubert and Chanedeau (2009), foca em achar topologias elevadoras quadráticas para aplicação em sistemas fotovoltaicos usando o procedimento de Maksimovic (2007). Além destes, recentemente Mo et al. (2022) criou um sistema computadorizado para síntese de conversores multi-porta com só um indutor. Além destes, Panigrahi et al. (2020) e Panigrahi et al. (2021) desenvolveu processos de síntese desses mesmos conversores, mas com a utilização do princípio de volt-segundo presente nos indutores dos conversores para assim sintetizar somente os que atendem uma certa especificação de ganho estático. Em outro trabalho, Li et al. (2022) utiliza de um método de construção junto de regras para gerações de novos conversores quadráticos. Mesmo cumprindo os objetivos propostos pelos autores, esses dois últimos pecam pela complexidade.

Assim, este artigo visa desenvolver um escopo mais amplo: um algoritmo elaborado no programa MATLAB MathWorks (2016) capaz de sintetizar todos os conversores cc-cc de uma porta que forneça tanto os circuitos sintetizados, como também os valores de operação em condução contínua de todos os componentes unindo a metodologia de Zhou (2005) e Maksimovic (2007) com a formulação em espaço de estados de Balabanian (1969) e Middlebrook and Čuk (1977).

2. SÍNTESE DE CONVERSORES

2.1 Teoria de grafos aplicada em conversores cc-cc

O artigo seguirá as definições de Balabanian (1969) para os termos: ramo, nó, árvore, galho, corda e corte, além estabelecer a matriz de incidência, de circuitos fechados e de corte. Um ramo é um componente que conecta dois nós de um grafo. Um nó é a posição em que um ramo se conecta. Cada nó tem uma quantidade de ramos ligados a ele e a essa quantidade é dado o nome de grau. Uma árvore é um conjunto de ramos em um grafo que se conecta a todos os nós, msem formar circuito fechado. Os ramos que constituem uma árvore são chamados galhos. Uma corda é todo ramo de um grafo que não faz parte da árvore. Um corte, também chamado *cutset*, é um conjunto de ramos de um grafo cuja remoção causa o surgimento de dois grafos conectados. Uma matriz de incidência, A , é uma matriz retangular em que as colunas representam os ramos e os nós, as linhas. Para indicação da conectividade do ramo em cada nó, são atribuídos os valores 0,1 e -1 indicando,

respectivamente, situações em que o ramo não se conecta, se conecta saindo do nó ou se conecta chegando ao nó.

A matriz de circuito fechado e de corte são aplicações das leis de Kirchhoff das malhas e das correntes, respectivamente. Organizando a matriz de incidência com as primeiras colunas sendo os galhos e as últimas colunas serem as cordas e escalonando-a, é obtido uma matriz $A_f = [A_t A_l]$. A relação entre as matrizes de incidência, de circuito fechado e de corte está descrita por (1).

$$\begin{aligned} A_f &= [A_t \ A_l] \\ B_f &= [-(A_t^{-1} * A_l)' \ I] \\ Q_f &= [I \ A_t^{-1} * A_l] \end{aligned} \quad (1)$$

Um conversor PWM cc-cc é definido como um circuito elétrico composto por um conjunto de fontes de tensão V_g , resistores \mathbf{R} , indutores \mathbf{L} e capacitores \mathbf{C} e um conjunto de semicondutores maior ou igual a 2. Tal conversor respeita as seguintes regras definidas por Maksimovic (2007): cada resistor presente deve estar em um circuito fechado formado por $\mathbf{C} \cup V_g$ e não pode haver no grafo resultante do circuito elétrico um corte contendo somente elementos do conjunto de \mathbf{L} .

O método de síntese de conversores desenvolvido por Zhou (2005) baseia-se nos circuitos de baixa e alta frequência. O circuito equivalente de baixa frequência contém os capacitores e fontes de tensão em circuito aberto, os indutores e as fontes de corrente em curto. Já os circuitos de alta frequência são o oposto: capacitores e fonte de tensão de entrada em curto-circuito; indutores e fontes de corrente em circuito aberto. A Figura 1a ilustra o conversor buck e os circuitos de baixa e alta frequência e a representação em forma de grafos do conversor.

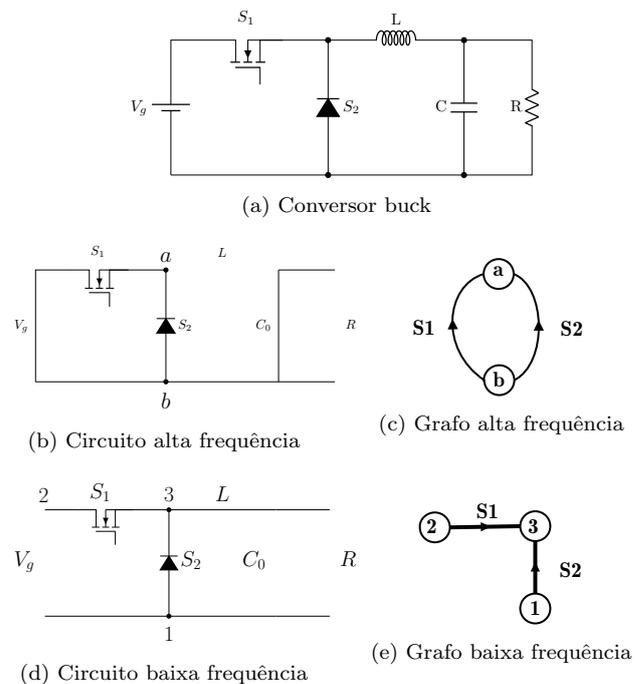


Figura 1. Conversor Buck e os circuitos de baixa e alta frequência

Para que um grafo seja uma representação de um circuito de alta frequência de um conversor cc-cc PWM, é preciso que três regras sejam respeitadas: não deve existir um nó com grau igual a um; o grafo deve ser conectado; as tensões e correntes dos semicondutores devem ser determinadas no grafo para cada etapa de operação. Essa última regra citada é referente ao comportamento dos semicondutores. Como descrito por Zhou (2005), um semicondutor em um circuito de alta frequência comporta-se como uma fonte de tensão quando está acionado e uma fonte de corrente quando está desligado. Nos dois casos, a amplitude é igual ao negativo do seu valor médio de tensão ou corrente, respectivamente. Considerando o conversor buck, considere o seguinte comportamento dos elementos de chaveamento: no instante entre $[0, d_1.T_s]$ a chave S_1 está acionada e a chave S_2 está desligada. No instante entre $[d_1.T_s, T_s]$ é o oposto. A Figura 2a e 2b ilustram essa situação. Nesse caso, conforme modelagem de Zhou (2005) para semicondutores, as tensões das chaves para o circuito de alta frequência são determinadas e a representação de grafo de um circuito de alta frequência da Figura 2 é válida.

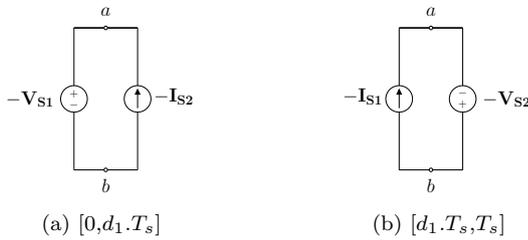


Figura 2. Circuito de alta frequência nas etapas de chaveamento conforme grafo da Figura 1c.

Além disso, na modelagem proposta em Zhou (2005), as chaves devem sempre obedecer à regra de *volt-segundo* igual a zero, como exposto por (2). A partir desta formulação, a matriz $B_a * V_s$ é representada, como indicado por (3), composto por (2) após o processo de escalonamento.

$$\frac{1}{T_s} * \int_0^{T_s} v_{s1} * dt = d_1 * (-V_{S1}) + (1 - d_1) * (-V_{S2}) = 0$$

$$\frac{1}{T_s} * \int_0^{T_s} v_{s2} * dt = d_1 * (-V_{S1}) + (1 - d_1) * (-V_{S2}) = 0$$

$$B_a * V_s = [d_1 \quad (1 - d_1)] * \begin{bmatrix} V_{s1} \\ V_{s2} \end{bmatrix} \quad (3)$$

O circuito de baixa frequência é determinado pelas quantidades de nós e ramos do circuito de alta frequência, determinada conforme (4). A variável q , em suma, é a quantidade de linhas da matriz B_a escalonada.

$$nt_{cc} = q + n_g$$

$$nl_{cc} = nt_{ac} + nl_{ac} - q - n_g \quad (4)$$

2.2 Inserção de indutores e capacitores e representação final de um conversor válido.

M_{ca} e M_{cc} , ilustrados pelas Figuras 1c e 1e, são as representações matriciais dos grafos do conversor buck. Com

M_{ca} e M_{cc} , o próximo passo é a inserção de indutores, capacitores e o resistor que representará a saída do conversor. As colunas dessas matrizes representam os elementos de chaveamento, devendo ter a mesma sequência de representação das chaves S_1, S_2, \dots, S_n em ambas. Uma consideração importante para M_{cc} é a escolha da ordem para as linhas da matriz. No caso deste artigo, as linhas seguirão esta sequência: a primeira, segunda e terceira linha representam, respectivamente, os nós do terminal comum do conversor, o terminal positivo da tensão de entrada e o positivo da tensão de saída. Tal ordenação é importante para saber as conexões em que estarão a fonte de tensão de entrada e a resistência que representa a saída do conversor, nesse caso, os ramos [1 2] e [1 3], respectivamente. No caso de M_{ca} , não há necessidade de uma regra no ordenamento das linhas, cada ordem escolhida gerará um novo conversor, já que matriz de incidência dos indutores depende do ordenamento de linhas de M_{ca} .

$$M_{ca} = \begin{matrix} & S1 & S2 \\ a & \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \end{matrix}, \quad M_{cc} = \begin{matrix} & S1 & S2 \\ 1 & \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & -1 \end{bmatrix} \end{matrix} \quad (5)$$

Para a inserção dos indutores, usa-se o procedimento descrito por Zhou (2005). O algoritmo de alocação de indutores, algoritmo 1, compara os números da matriz M_{cc} com uma linha M_{ca} considerando a quantidade de elementos de chaveamento n_s .

Algoritmo 1 Algoritmo de alocação de indutores

```

1: procedimento COMPROW( $M_{ac}, M_{dc}, n_s$ )
2:    $n1_{dc} \leftarrow [\dots]$ 
3:    $n2_{dc} \leftarrow [\dots]$ 
4:    $ncap_{dc} \leftarrow [\dots]$ 
5:    $l_{indice} \leftarrow 0$ 
6:    $conj_{indice} \leftarrow [\dots]$ 
7:    $nConjunto \leftarrow 0$ 
8:   para  $i \leftarrow 1$  to row( $M_{ac}$ ) faça
9:      $ncap_{auxdc} \leftarrow [\dots]$ 
10:    se  $i == 1$  então
11:       $d_{dc} \leftarrow M_{dc}$ 
12:    caso contrário
13:       $d_{dc} \leftarrow n2_{dc}$ 
14:    fim se
15:     $n2_{dc} \leftarrow [\dots]$ 
16:     $conj_{indice}[k] \leftarrow [\dots]$ 
17:    para  $j \leftarrow 1$  to row( $d_{dc}$ ) faça
18:      para  $k \leftarrow 1$  to  $n_s$  faça
19:         $c_{dc}[j, k] \leftarrow 0$ 
20:        se  $d_{dc}[i, k] == 0$  então
21:           $c_{dc}[j, k] \leftarrow 0$ 
22:        caso contrário se  $d_{dc}[i, k] == M_{ac}[j, k]$  então
23:           $c_{dc}[j, k] \leftarrow 0$ 
24:        caso contrário
25:           $c_{dc}[j, k] \leftarrow M_{ac}[i, k]$ 
26:        fim se
27:      fim para
28:     $nConjunto \leftarrow nConjunto + 1$ 
29:     $conj_{indice}[k] \leftarrow nConjunto$ 
30:     $t, row(n2_{dc})$ 
31:     $w, row(n1_{dc})$ 
32:    se  $c_{dc}[j, 1 : n_s] == zeros(1, n_s)$  então
33:       $n1_{dc}[w + 1, 1 : n_s] \leftarrow d_{dc}[j, 1 : n_s]$ 
34:      se isempty( $(d_{dc}[j, : n_s + 1])$ ) então
35:         $n1_{dc}[w + 1, n_s + 1] \leftarrow 0$ 

```

```

36:      caso contrário
37:           $n1_{dc}[w + 1, n_s + 1] \leftarrow d_{dc}[w + 1, n_s + 1]$ 
38:      fim se
39:      caso contrário se  $C_{dc}[j, 1 : n_s] == D_{dc}[j, 1 : n_s]$  entao
40:          se  $j < 3$  entao
41:               $n1_{dc}[w + 1, 1 : n_s] \leftarrow [0, 0, \dots, 0]$ 
42:               $n2_{dc}[t, 1 : n_s] \leftarrow d_{dc}[j, 1 : n_s]$ 
43:               $l_{indice} = l_{indice} + 1;$ 
44:               $n1_{dc}[w + 1, n_s + 1] \leftarrow l_{indice}$ 
45:               $n2_{dc}[t, n_s + 1] \leftarrow l_{indice}$ 
46:          caso contrário
47:               $n1_{dc}[w + 1, 1 : n_s] \leftarrow d_{dc}[j, 1 : n_s]$ 
48:              se isempty( $(d_{dc}[j, : n_s + 1])$ ) entao
49:                   $n1_{dc}[w + 1, n_s + 1] \leftarrow 0$ 
50:              caso contrário
51:                   $n1_{dc}[w + 1, n_s + 1] \leftarrow d_{dc}[w + 1, n_s + 1]$ 
52:              fim se
53:          fim se
54:      caso contrário
55:           $n1_{dc}[w + 1, 1 : n_s] \leftarrow n1_{dc}[j, :] - c_{dc}[j, 1 : n_s]$ 
56:           $n2_{dc}[t, 1 : n_s] \leftarrow c_{dc}[j, :]$ 
57:           $l_{indice} = l_{indice} + 1;$ 
58:           $n1_{dc}[w + 1, n_s + 1] \leftarrow l_{indice}$ 
59:           $n2_{dc}[t, n_s + 1] \leftarrow l_{indice}$ 
60:      fim se
61:  fim para
62: fim para
63: fim procedimento

```

Considerando o exemplo do conversor buck e as suas matrizes, (5), a saída do algoritmo de alocação de indutores está representado na Tabela 1.

Tabela 1. Resposta final do algoritmo de alocação dos indutores

	D_{cc}	M_{ca}	C_{ac}	$N1_{ac}$	$N2_{ac}$
Iteração 1	0 1 1 1 0 0 0 1 0 x x x	1 0 1 1 0 0 1 0 0 x x x	-1 -1 1 1 -1 -1 0 0 1 -1 -1 1		
Iteração 2	-1 -1 -1 -1 0 0 -1 -1 1 x x x				

A matriz $N1_{cc}$ contém a matriz de incidência do grafo por completo, com as colunas de $1 : n_s$ indicando a orientação dos elementos de chaveamento e com a última coluna indicando o posicionamento do indutor. Estas novas colunas geradas são alocadas em uma nova matriz, $N1_{cc}$. A Tabela 2 demonstra o processo relatado.

Tabela 2. Matriz com o posicionamento das chaves e a alocação dos indutores

	Nós	$N1_{cc}[1 : n_s]$	$N1_{cc}[n_s + 1]$	$N1_{cc}$
Iteração 1	1	0 1	0	0
	2	1 0	0	0
	3	0 0	1	1
Iteração 2	4	-1 -1	1	-1

Conforme mostrado na 2, as linhas de 1 a 3 de $N1_{cc}[1 : n_s]$ somadas são exatamente a linha 1 da Equação 1 da matriz M_{ca} . Cada conjunto de iteração das linhas da matriz $N1_{cc}[1 : n_s]$ representa uma das linhas da matriz M_{ca} .

2.3 Inserção de Capacitores

Para que as alocações dos semicondutores e indutores constituam um conversor de mesmo M_{ca} considerado no algoritmo 1, é preciso que nós de cada iteração sejam interligados por capacitores. Tal procedimento é descrito pelo Algoritmo 2.

Algoritmo 2 Algoritmo alocação de capacitores

```

procedimento ALOCACAP( $conj_{indice}, n_{cap_{dc}}$ )
2:   $l_1 \leftarrow \text{row}(\text{Conj}_{indice})$ 
   para  $i == 1 : l_1$  faça
4:       $n_{no} \leftarrow \text{column}(\text{Conj}_{indice}[i])$ 
        $n_{cap} \leftarrow n_{no} - 1$ 
6:       $ramos_{cap} = \text{nchoosek}([1 : n_{no}], 2)$ 
        $n_{q,amo} \leftarrow \text{row}(ramos_{cap}, 1)$ 
8:       $escolha_{caps} = \text{nchoosek}([1 : n_{q,amo}], n_{cap})$ 
        $arvore_{comb} = \text{GeraArvore}g(escolha_{caps}, i)$ 
10:      $\text{Conjunto}_{caps}(i).vetor_{caps} = arvore_{comb}$ 
12: fim para
fim procedimento

```

Algoritmo 3 Algoritmo GeraArvore

```

procedimento GERAARVORE( $escolha_{caps}, i$ )
    $l_1 \leftarrow$  quantidade de combinações de  $escolha_{caps}$ 
3:  para  $j == 1 : l_1$  faça
   Af ← matriz incidência de  $escolha_{caps}(j)$ 
   Bf ←  $-(A_t^{-1} * A_l)' I$ 
6:  se Bf → nulo & i==1 &  $escolha_{caps}(j)$  contém [1 2]
   entao
        $escolha_{caps}(j)$  é uma árvore
        $arvore_{comb} = escolha_{caps}(j)$ 
9:  caso contrário
   se Bf → nulo & i=1 entao
        $escolha_{caps}(j)$  é uma árvore
        $arvore_{comb} = escolha_{caps}(j)$ 
12: caso contrário
    $escolha_{caps}(j)$  não é uma árvore
15: fim se
   fim se
   fim para
18: fim procedimento

```

O algoritmo procede da seguinte maneira: $Conj_{indice}$ é uma matriz em que cada linha contém os nós de cada iteração da primeira coluna da Tabela 2. Esses nós que precisam ser conectados por capacitores. Assim, para cada linha de $Conj_{indice}$ é extraído a quantidade de nós a serem conectados e a quantidade de capacitores necessários. Essas duas variáveis são representados por n_{no} e n_{cap} . Em seqüência, determinam-se as combinações válidas de ramos por meio da função $nchoosek(n, k)$, uma função disponível na biblioteca do programa MATLAB MathWorks (2016) que retorna as combinações de n itens para k escolhas. Todas as combinações serão armazenadas em $escolha_{caps}$. O próximo passo é determinar quais as combinações de $escolha_{caps}$ formam árvores. Isso será feito a partir da função $GeraArvore$, descrito no Algoritmo 3. $GeraArvore$ analisa cada combinação de $escolha_{caps}$, primeiramente gerando a matriz incidência conforme descrito em 1 e verificando se a matriz B_f resultante não contém nenhum circuito fechado. Caso não tenha, o conjunto forma uma árvore. Caso seja na primeira iteração, contendo os nós [1 2 3], a função $GeraArvore$ deve garantir que o conjunto possua um ramo [1 2], o ramo da fonte de entrada. Com as árvores da linha i de $Conj_{indice}[i]$ determinadas, armazenam-se essas combinações em $conjunto_{caps}(i).vetor_{caps}$. O processo continua até todas as linhas de $Conj_{indice}(i)$ tiverem as árvores armazenadas no objeto $conjunto_{caps}$.

Exemplificando, considere a Tabela 2. $Conj_{indice}(1)$ terá [1 2 3] na primeira linha e [4] na segunda. Para a primeira

linha, os ramos possíveis são: [1,2], [3,2] e [1,3]. Desses três ramos, como se necessita de dois capacitores, há 3 combinações válidas: [1 2] e [1 3], [1 2] e [3 2], [1 3] e [3 2]. Destas três combinações, todas formam árvores, pois a matriz B_f resultante de cada combinação de ramos é nula, entretanto, como é a primeira linha de $Conj_{indice}(1)$, [1 3] e [2 3], será descartado por $GeraArvoreVg$. Salvando essas duas combinações, parte para a segunda linha, $Conj_{indice}(2)$. Como só há um nó, não há árvore e o $conjunto_{caps}(2).vetor_{caps}$ é nulo. Desta forma, os conjuntos de capacitores possíveis a serem alocados estão determinados por $conjunto_{caps}(1).vetor_{caps}$ tendo duas combinações possíveis, gerando 2 conversores.

Para o exemplo considerado anteriormente, escolhe-se a primeira combinação, [1 2] e [1 3]. Essa escolha está presente na Tabela 3, sendo a representação final do conversor com todos os elementos devidamente alocados.

Tabela 3. Resposta final do algoritmo - Alocação dos Capacitores

	Nós	Ns_{cc}	Nl_{cc}	V_g	R	$NCap_{cc}$
Iteração 1	1	0	1	0	-1	-1
	2	1	0	0	1	0
	3	0	0	1	0	1
Iteração 2	4	-1	-1	-1	0	0

2.4 Derivação de topologias

É possível obter uma topologia derivada com entrada e saída com corrente não pulsada. Segundo Zhou (2005), adicionando indutores e capacitores de forma que uma das linhas 1,2 e 3 de $Nl_{cc}[1 : ns]$ contenham só zeros, fará com que esses terminais sejam conectados por um indutor.

Continuando com o exemplo da Seção 3, para gerar uma variação do conversor buck com uma entrada de corrente não pulsada, basta que a linha 2 da matriz Ns_{cc} seja igual a zero. Essa inserção de um indutor aumenta o conjunto da Iteração 1, aumentando a quantidade de nós em $Conj_{indice}(1)$, sendo preciso executar o procedimento da Seção 2.3 para uma nova alocação de capacitores. A Tabela 4 mostra o exemplo do conversor buck após a inserção da nova árvore de capacitores.

Tabela 4. Matriz de incidência das chaves e indutores considerando entrada contínua.

	Nós	$Nl_{cc}[1 : ns]$	$Nl_{cc}[ns+1]$	Nl_{cc}
Iteração 1	1	0	1	0
	2	0	0	2
	3	0	0	1
	2'	1	0	2
Iteração 2	4	-1	-1	1

2.5 Valores de tensão e corrente dos elementos presentes nos conversores.

Para obter os valores de tensão e corrente dos elementos a partir do uso da formulação do espaço de estados é preciso, primeiro, obter os circuitos resultantes das etapas de chaveamento. Para isso é usada a *netlist* completa, descrita na Tabela 5. Essa *netlist* contém a informação sobre quais nós os semicondutores estão conectados. Assim, para obter os circuitos representantes de cada instante

de chaveamento, basta considerar que o semicondutor acionado em cada etapa de operação gera um curto entre os dois nós onde se conecta, substituindo um dos números de sua indicação de conexão por outro.

Tabela 5. Netlist do conversor buck

S_1	S_2	L_1	V_g	R	C_1
2	1	3	2	3	3
4	-4	-4	-1	-1	-1

Exemplificando, com o acionamento da chave S_1 no primeiro intervalo de chaveamento, o nó 2 e o nó 4 se tornam um só. Assim, o nó 4 é substituído pelo nó 2, eliminam-se os semicondutores e a Netlist 1 da Tabela 6 é gerada. O mesmo procedimento é realizado na segunda etapa de chaveamento. Após esse processo, é preciso identificar se há algum circuito fechado nas etapas de operação contendo $C \cup V_g$. Caso exista, uma resistência em série ser inserida neste circuito fechado, incluído-a na *netlist*, e assim tornando possível a formulação do espaço de estado na forma $A * \bar{X} + B * \mathbf{V}_g$. Essa resistência seria a resistência parasita presente em capacitores. Deste modo, considerando o exemplo do conversor buck, as duas *netlists* de cada etapa de operação estão descritas na 6. Com essas *netlists*, a matriz de incidência para cada etapa de chaveamento é obtida.

Tabela 6. Netlist do conversor buck para as etapas de chaveamento

	V_g	C_1	R	L_1
Netlist 1 $[0, d_1.T_s]$	2	3	3	3
	-1	-1	-1	-2
Netlist 2 $[d_1.T_s, T_s]$	2	3	3	3
	-1	-1	-1	-1

Para o equacionamento do circuito, segue-se a formulação de Balabanian (1969). A Equação 6 é o exemplo da primeira etapa de operação do conversor buck, representado pela *netlist* 1 da 6. Na matriz de incidência resultante de cada *netlists*, os elementos devem estar organizados com os galhos primeiro depois as cordas. Indutores sempre serão cordas e capacitores e fontes de tensão sempre serão árvores. Dessa forma a ordem adotada deve ser: $\mathbf{V}_g, \mathbf{C}, \mathbf{R}_t; \mathbf{R}_1, \text{ e } \mathbf{L}$.

$$A_f = [A_t \ A_l] = \begin{matrix} \text{Árvores} & \text{Links} \\ \text{Nós} & V_g & C_1 & R_1 & L_1 \\ 1 & \begin{bmatrix} -1 & -1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (6)$$

Particionando a matriz entre galhos e cordas, é obtido a matriz de corte e de circuito fechado, conforme Equação (6). Com a matriz de cordas, obtém-se 7. Com seus blocos, é utilizada a notação presente em Balabanian (1969), descrita na Equação (10), para obter os valores das tensões de capacitores e indutores. Com as matrizes de espaço de estados obtidas em cada etapa de operação, é utilizado o processo descrito por Middlebrook and Čuk (1977) para determinação dos respectivos valores médios de $[V_{Ct} \ I_{Lt}]'$ conforme (11). O vetor representa os intervalos de chaveamento, $d = [d_1, d_2, \dots, d_n]$ para $d_n = 1 - \sum_{i=1}^{n-1} d_i$.

$$Q_f = [I \ Q_l] \quad Q_l = \begin{matrix} \mathbf{V}_g \\ \mathbf{C} \\ \mathbf{R}_t \end{matrix} \begin{matrix} \mathbf{R}_l & \mathbf{L} \\ \begin{bmatrix} Q_{er} & Q_{el} \\ Q_{cr} & Q_{cl} \\ Q_{rr} & Q_{rl} \end{bmatrix} \end{matrix} \quad (7)$$

$$\begin{aligned} \mathbf{G}_l &= \mathbf{R}_l^{-1} \\ \mathbf{G}_t &= \mathbf{R}_t^{-1} \\ \mathbf{R} &= \mathbf{R}_l + Q'_{rr} * \mathbf{R}_t * Q_{rr} \\ \mathbf{G} &= \mathbf{G}_t + Q_{rr} * \mathbf{G}_l * Q'_{rr} \end{aligned} \quad (8)$$

$$\begin{aligned} y_a &= Q_{cr} * \mathbf{R}^{-1} * Q'_{cr} \\ h_a &= -Q_{cl} + Q_{cr} * \mathbf{R}^{-1} * Q'_{rr} * \mathbf{R}_t * Q_{rl} \\ z_a &= Q'_{rl} * \mathbf{G}^{-1} * Q_{rl} \\ g_a &= Q'_{cl} - Q'_{rl} * \mathbf{G}^{-1} * Q_{rr} * \mathbf{G}_l * Q_{cr} \\ y_b &= Q_{cr} * \mathbf{R}^{-1} * Q'_{cr} \\ g_b &= Q'_{el} - Q'_{rl} * \mathbf{G}^{-1} * Q_{rr} * \mathbf{G}_l * Q'_{er} \end{aligned} \quad (9)$$

$$\left[\frac{dV_{Ct}}{dt} \right] = \begin{bmatrix} \mathbf{C} & 0 \\ 0 & \mathbf{L} \end{bmatrix} * \left\{ \begin{bmatrix} -y_a & h_a \\ g_a & z_a \end{bmatrix} \begin{bmatrix} V_{Ct} \\ I_{Lt} \end{bmatrix} + \begin{bmatrix} -y_b \\ g_b \end{bmatrix} [v_g] \right\} \quad (10)$$

$$\begin{aligned} \bar{X} &= [V_{Ct} \ I_{Lt}] = \bar{A}^{-1} * \bar{B} * v_g \\ \bar{A} &= \sum_{i=1}^n (d_i * A_i) \quad \bar{B} = \sum_{i=1}^n (d_i * B_i) \end{aligned} \quad (11)$$

Os valores da tensão nos indutores e nas resistências existentes e corrente nos capacitores e da fonte de tensão de entrada em cada etapa de operação são calculados conforme (14) com os valores de $[V_{Ct} \ I_{Lt}]'$ de (11). Como a resistência que representa a saída sempre será uma corda, \mathbf{V}_{Rl} armazenará o valor da tensão de saída. O valor da corrente da fonte de entrada pode ser calculado a partir da (13), obtida de uma simplificação da modelagem feita por Fowler (1969).

$$\begin{aligned} M_{Rl} &= Q'_{rr} * \mathbf{R}_t * Q_{rr} * \mathbf{G}_l + \mathbf{I} \\ M_{Rt} &= Q_{rr} * \mathbf{G}_l * Q'_{rr} * \mathbf{R}_t + \mathbf{I} \\ y_{rb} &= M_{Rl}^{-1} * Q'_{er} \\ g_{rb} &= \mathbf{R}_t * M_{Rt}^{-1} * -Q_{rr} * \mathbf{G}_l * Q'_{er} \\ y_{rl} &= M_{Rl}^{-1} * Q'_{cr} \\ h_{rl} &= M_{Rl}^{-1} * -Q'_{rr} * \mathbf{R}_t * Q_{rl} \\ z_{rt} &= \mathbf{R}_t * M_{Rt}^{-1} * -Q_{rr} * \mathbf{G}_l * Q_{cr} \\ g_{rt} &= \mathbf{R}_t * M_{Rt}^{-1} * -Q_{rl} \end{aligned} \quad (12)$$

$$\begin{aligned} y_i g_a &= -Q_{er} * \mathbf{G}_l * [I + Q'_{rr} * \mathbf{R}_t * Q_{rr} * \mathbf{G}_l]^{-1} * Q'_{cr} \\ h_i g_{a1} &= -Q_{er} * \mathbf{G}_l * [I + Q'_{rr} * \mathbf{R}_t * Q_{rr} * \mathbf{G}_l]^{-1} * Q'_{rr} \\ h_i g_a &= Q_{el} + h_i g_{a1} * \mathbf{R}_t * Q_{rl} \end{aligned} \quad (13)$$

$$\begin{bmatrix} I_{Ct} \\ V_{Ll} \\ V_{Rl} \\ V_{Rt} \\ I_g \end{bmatrix} = \begin{bmatrix} -y_a & h_a \\ g_a & z_a \\ y_{ra} & h_{ra} \\ g_{ra} & z_{ra} \\ y_i g_a & h_i g_a \end{bmatrix} * \begin{bmatrix} V_{Ct} \\ I_{Lt} \end{bmatrix} + \begin{bmatrix} -y_b \\ g_b \\ y_{rb} \\ g_{rb} \\ y_i g_b \end{bmatrix} * [v_g] \quad (14)$$

Por último, é preciso obter os valores das correntes e tensões dos semicondutores em cada etapa de operação. Utilizando o *netlist* completo do conversor, obtém-se a matriz de incidência contendo também os semicondutores. Para cálculo da tensão nos elementos de chaveamento,

organiza-se a matriz de incidência completa de forma que as colunas dos elementos de chaveamento estejam por último, ou seja, chaves e diodos como cordas. Dessa forma, a partir da matriz de circuito fechado, conforme (1), será obtida a equação geral de tensão de cada elemento de chaveamento, sendo uma soma de tensão dos componentes calculados nas Equações de (7) a (14) para cada instante de chaveamento. Para o cálculo das correntes é o mesmo processo, com a diferença que matriz de incidência completa será alterada de forma as chaves serem consideradas árvores. Deste modo, a partir da matriz de corte, a corrente das chaves será determinada pela soma das correntes dos outros elementos.

3. RESULTADOS

Os resultados considerarão conversores de 2 e 3 chaves. Primeiramente deve-se determinar todas as combinações disponíveis de grafos ca que obedecem às regras descritas em 2.1. Assim, aplica-se a notação de Tiernan (1970) para obter todas as combinações de grafos para um valor de n_s . Esse formato será usado, pois, permite um jeito simples de alocar ramos paralelos. (15) demonstra os espaços de alocação de chaves para diferentes dimensões de matrizes.

$$G_2 = \frac{1}{2} \begin{bmatrix} 0 & a_{12} \\ a_{21} & 0 \end{bmatrix}, \quad G_3 = \frac{1}{3} \begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \quad (15)$$

O processo de alocação das chaves para geração do grafo ca parte da geração de vetores da partição do número de chaves. Para $n_s = 3$ as partições numéricas são: $3*1$, $1*2 + 2*1$, $1*1 + 1*2 + 1*3$. Transformando isso em combinações: $[3 \ 0 \ 0]$, $[2 \ 1 \ 0]$, $[1 \ 1 \ 1]$. $[3 \ 0 \ 0]$ e $[2 \ 1 \ 0]$ podem ser alocados em G_2 . Assim, as combinações $[a_{12} \ a_{21}]$ são: $[3 \ 0]$, $[0 \ 3]$, $[1 \ 2]$ e $[2 \ 1]$. Já combinação $[1 \ 1 \ 1]$ precisa de 3 espaços. Assim, a combinação $[1 \ 1 \ 1]$ alocado em $[a_{12} \ a_{21} \ a_{13} \ a_{23} \ a_{31} \ a_{32}]$ de G_3 , gera o vetor $[1 \ 1 \ 1 \ 0 \ 0 \ 0]$ e mais outras 19 possibilidades criadas por permutação. Considerando conversores de duas chaves, $[1 \ 1]$ e $[2 \ 0]$ podem ser alocados na matriz G_2 como $[1 \ 1]$, $[2 \ 0]$ e $[0 \ 2]$. Em resumo, para duas chaves, há 3 possibilidades de grafos ca. Para três chaves, 24. Entretanto, nem todas essas combinações são úteis. É preciso eliminar combinações que levam a casos com nó de grau igual a 1 ou casos redundantes. Em casos redundantes, ambos grafos chegarão ao mesmo valor da matriz B_a caso tenham uma mesma matriz de chaveamento e consequentemente, chegarão sempre nos mesmos conversores no processo de síntese. Eliminado estes casos, quantidade de grafos ca válidos para 3 chaves passam de 24 para 4. Já para 2 chaves passa de 3 para 2. As Tabelas 7 e 8 descrevem os casos válidos restantes.

Tabela 7. Tabela de grafos ca para duas chaves com dois nós

Vetores	a_{12}	a_{21}
1	0	2
2	1	1

Agora, considerando as etapas de chaveamento, em todos os casos contendo duas chaves, o único intervalo de cha-

Tabela 8. Grafos ca válidos sem nó pendente para três chaves com três nós

Vetores	Dimensão da Matriz	Intervalos					
		a_{12}	a_{21}	a_{13}	a_{23}	a_{31}	a_{32}
1	G_2	0	3	X	X	X	X
2	G_2	1	2	X	X	X	X
3	G_3	0	1	1	0	0	1
4	G_3	1	0	1	1	0	0

veamento válido está descrito pela Tabela 9. Já para três chaves, há 4 intervalos válidos. Em todos os casos, são respeitadas as regras de Zhou (2005) para grafos ca.

Tabela 9. Combinações de dois intervalos de chaveamento para conversores de 2 chaves

2*Combinações	Intervalos	
	$[0, d_1.T_s]$	$[d_1.T_s, T_s]$
1	S_1	S_2

Tabela 10. Combinações de dois intervalos de chaveamento para conversores de 3 chaves

2*Combinações	Intervalos	
	$[0, d_1.T_s]$	$[d_1.T_s, T_s]$
1	S_1	$S_2 ; S_3$
2	$S_1 ; S_2$	S_3
3	$S_1 ; S_3$	S_2

O passo seguinte é a determinação de grafos cc válidos. A Tabela 11 ilustra as árvores disponíveis para os dois grafos resultantes presentes na Tabela 7. Para 2 chaves, a quantidade de combinações disponíveis é de 2 vetores da Tabela 7 vezes as três árvores da Tabela 11.

Tabela 11. Combinações de possíveis árvores e cordas disponíveis para conversores de duas chaves.

2*Combinações	Árvores	
	S_1	S_2
1	$[1,2]$	$[1,3]$
2	$[1,2]$	$[2,3]$
3	$[1,3]$	$[2,3]$

Em cada combinação, pode haver mais de um conversor válido, graças as várias possibilidades de alocação de capacitores. Considerando os conversores abaixadores de duas chaves com até dois indutores e dois capacitores, a Tabela 12 mostra as 14 diferentes topologias possíveis de ser obtidos com o mesmo M_{ca} e M_{cc} de (16), mas com ordem de linhas árvores de capacitores diferentes. Neste exemplo, M_{ca} é obtido pela linha 1 da Tabela 7 e M_{cc} pela linha 3 da Tabela 11. Dos 14 conversores abaixadores obtidos na Tabela 12, além das conhecidas, buck e buck com filtro, há as topologias C1, D4, F1, e F3, descritas em Tymerski and Vorperian (1986). Além das versões tradicionais, há as mesmas topologias, mas com o capacitor de saída deslocado do ramo $[3 \ 1]$ para o ramo $[3 \ 2]$.

$$M_{ca} = \frac{1}{2} \begin{bmatrix} 1 & 2 \\ -1 & -1 \\ 1 & 1 \end{bmatrix}, \quad M_{cc} = \frac{1}{3} \begin{bmatrix} 1 & 2 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \quad (16)$$

A Tabela 13 apresenta seis exemplos de síntese de conversores de três chaves. Estes usam a etapa de chaveamento da linha 1 da Tabela 10 e gerados a partir do grafo ca 3 da Tabela 8. Este grafo ca apresenta um galho e duas cordas. Segundo (4), terá 4 nós e 3 árvores. Combinando os seis ramos possíveis: $[1 \ 2]$, $[1 \ 3]$, $[1 \ 4]$, $[2 \ 3]$, $[2 \ 4]$ e $[3 \ 4]$, há 96 grafos cc disponíveis para a síntese dos conversores. Com 96 combinações e duas linhas ca, há 196 combinações que geram conversores. Destes 96, 6 das topologias que são abaixadoras estão descritos na Tabela 13.

4. CONCLUSÃO

Este artigo descreveu o processo de síntese de conversores cc-cc PWM em condução contínua a partir de trabalhos de Zhou (2005) e a teoria de grafos para circuitos elétricos. Os algoritmos descritos em pseudocódigo e em texto nesse artigo foram implementados no *software* MATLAB MathWorks (2016). A metodologia descrita gera um gama de novos conversores, considerando quantidade de chaves, a partir de dois grafos, representando um circuito de alta e baixa frequência. Com isso, é possível descrever famílias de conversores com diferentes quantidades de chaves. A desvantagem está em somente descrever os modelos de conversor cc-cc do tipo PWM, preso às regras determinadas de comportamento de indutores e capacitores e a sua alocação. Para ampliar o espaço de *design*, permitindo síntese de conversores ressonantes, isolados ou cc-ca, é preciso identificar as regras dos componentes presentes nessas classes e usar a teoria de grafos para obter notações matemáticas para então realizar a aplicação em um algoritmo.

AGRADECIMENTOS

Os autores agradecem à CAPES pela bolsa de estudo e à empresa Aruna Energia Solar pelo suporte.

REFERÊNCIAS

- Balabanian, Norman e Bickart, T.A. (1969). *Electrical network theory*. Wiley, New York.
- Fowler, E.R. (1969). *A unified state-space approach to RLCT two-port transfer function synthesis*. Ph.D. thesis. URL <https://shareok.org/handle/11244/32774?show=full>.
- Gaubert, J.P. and Chandedeau, G. (2009). Evaluation of dc-to-dc converters topologies with quadratic conversion ratios for photovoltaic power systems. In *2009 13th European Conference on Power Electronics and Applications*, 1–10. IEEE. URL <https://ieeexplore.ieee.org/document/5279114>.
- Li, G., Amirabadi, M., Chen, X., and Lehman, B. (2022). The methodology of constructing the quadratic converters. *IEEE Journal of Emerging and Selected Topics in Power Electronics*.
- Lu, D.D.C., Soon, J.L., and Verstraete, D. (2016). Derivation of dual-switch step-down dc/dc converters with

Tabela 12. Topologias abaixadoras com duas chaves sintetizadas pelo script

Grafo AC	Grafo DC	Netlist	Conversor	Grafo AC	Grafo DC	Netlist	Conversor
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & R & L_1 & S_1 & S_2 \\ 2 & 3 & 3 & 4 & 1 & 2 \\ -1 & -1 & -1 & -3 & -4 & -4 \end{bmatrix}$	Buck	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 3 & 4 & 1 & 2 \\ -1 & -2 & -1 & -3 & -4 & -4 \end{bmatrix}$	Buck capacitor de saída deslocado
$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 5 & 3 & 4 & 5 & 4 & 5 \\ -1 & -1 & -4 & -1 & -1 & -2 & -3 & -3 \end{bmatrix}$	Conversor C1 Tymerski and Vorperian (1986)	$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 5 & 3 & 4 & 5 & 4 & 5 \\ -1 & -2 & -4 & -1 & -1 & -2 & -3 & -3 \end{bmatrix}$	Conversor C1 capacitor de saída deslocado
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 4 & 2 \\ -1 & -1 & -2 & -1 & -3 & -1 & -5 & -5 \end{bmatrix}$	Conversor D4 Tymerski and Vorperian (1986)	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 4 & 2 \\ -1 & -2 & -2 & -1 & -3 & -1 & -5 & -5 \end{bmatrix}$	Conversor D4 capacitor de saída deslocado
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 4 & 2 \\ -1 & -1 & -3 & -1 & -3 & -1 & -5 & -5 \end{bmatrix}$	Conversor F1 Tymerski and Vorperian (1986)	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 4 & 2 \\ -1 & -2 & -3 & -1 & -3 & -1 & -5 & -5 \end{bmatrix}$	Conversor F1 capacitor de saída deslocado
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 1 & 4 \\ -1 & -1 & -1 & -1 & -3 & -2 & -5 & -5 \end{bmatrix}$	Conversor Buck com filtro	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 1 & 4 \\ -1 & -1 & -2 & -1 & -3 & -1 & -5 & -5 \end{bmatrix}$	Conversor Buck com filtro capacitor de saída deslocado
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 1 & 2 \\ -1 & -1 & -3 & -1 & -3 & -2 & -5 & -5 \end{bmatrix}$	Conversor F3 Tymerski and Vorperian (1986)	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 3 & 4 & 3 & 5 & 4 & 1 & 2 \\ -1 & -2 & -3 & -1 & -3 & -2 & -5 & -5 \end{bmatrix}$	Conversor F3 capacitor de saída deslocado
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 4 & 4 & 3 & 5 & 4 & 1 & 4 \\ -1 & -1 & -3 & -1 & -3 & -1 & -5 & -5 \end{bmatrix}$	Conversor com 2 chaves 2 capacitores 2 indutores	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & L_2 & S_1 & S_2 \\ 2 & 4 & 4 & 3 & 5 & 4 & 1 & 4 \\ -1 & -1 & -3 & -1 & -3 & -2 & -5 & -5 \end{bmatrix}$	Conversor Buck capacitor de saída deslocado com filtro de entrada

Tabela 13. Topologias abaixadoras com três chaves sintetizadas pelo script

Grafo AC	Grafo DC	Netlist	Conversor	Grafo AC	Grafo DC	Netlist	Conversor
$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & S_1 & S_2 & S_3 \\ 2 & 3 & 5 & 3 & 4 & 1 & 4 & 2 \\ -1 & -1 & -4 & -1 & -1 & -5 & -3 & -5 \end{bmatrix}$	Conversor SD3 Zhou (2005)	$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & S_1 & S_2 & S_3 \\ 2 & 3 & 5 & 3 & 4 & 3 & 4 & 1 \\ -1 & -1 & -4 & -1 & -2 & -5 & -3 & -5 \end{bmatrix}$	Conversor SD7 Zhou (2005)
$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & S_1 & S_2 & S_3 \\ 2 & 3 & 5 & 3 & 4 & 1 & 1 & 2 \\ -1 & -1 & -4 & -1 & -3 & -5 & -4 & -5 \end{bmatrix}$	Conversor SD1 Zhou (2005)	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & R & L_1 & S_1 & S_2 & S_3 \\ 2 & 3 & 5 & 3 & 4 & 1 & 1 & 2 \\ -1 & -1 & -4 & -1 & -3 & -5 & -4 & -5 \end{bmatrix}$	Conversor SD6 Zhou (2005)
$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & C_3 & R & L_1 & L_2 & S_1 & S_2 & S_3 \\ 2 & 3 & 4 & 6 & 3 & 5 & 6 & 1 & 5 & 2 \\ -1 & -1 & -1 & -5 & -1 & -3 & -4 & -5 & -4 & -6 \end{bmatrix}$	Conversor com 3 chaves 3 capacitores 2 indutores	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} V_g & C_1 & C_2 & C_3 & R & L_1 & L_2 & S_1 & S_2 & S_3 \\ 2 & 3 & 4 & 6 & 3 & 5 & 6 & 6 & 5 & 5 \\ -1 & -1 & -3 & -5 & -1 & -1 & -2 & -4 & -4 & -3 \end{bmatrix}$	Conversor com 3 chaves 3 capacitores 2 indutores

fault-tolerant capability. *IEEE Transactions on Power Electronics*, 31(9), 6064–6068.

Maksimovic, D. (2007). *Synthesis of PWM and quasi-resonant DC-to-DC power converters*. Ph.D. thesis, California Institute of Technology. doi:10.7907/B8XA-2R90. URL <https://resolver.caltech.edu/CaltechETD:etd-02132007-105916>.

MathWorks (2016). Matlab. URL <https://www.mathworks.com/products/matlab.html>.

Middlebrook, R.D. and Čuk, S. (1977). A general unified approach to modelling switching-converter power stages. *International Journal of Electronics*, 42(6), 521–550. doi:10.1080/00207217708900678. URL <http://www.tandfonline.com/doi/abs/10.1080/00207217708900678>.

Mo, L., Huang, J., Chen, G., Qing, X., and Hu, Y. (2022). Computer-aided systematic topology derivation of single-inductor multi-input multi-output converters from working principle. *IEEE Transactions on Circuits and Systems I: Regular Papers*.

Panigrahi, R., Mishra, S.K., and Joshi, A. (2021). Synthesis of an optimum converter topology for a specified voltage conversion ratio. *IEEE Transactions on Industry Applications*, 57(4), 3923–3934.

Panigrahi, R., Mishra, S.K., Joshi, A., and Ngo, K.D. (2020). Dc-dc converter synthesis: An inverse problem. *IEEE Transactions on Power Electronics*, 35(12), 12633–12638.

Tiernan, J.C. (1970). An efficient search algorithm to find the elementary circuits of a graph. *Communications of the ACM*, 13(12), 722–726.

Tymerski, R. and Vorperian, V. (1986). Generation, classification and analysis of switched-mode dc-to-dc converters by the use of converter cells. In *INEC '86 - International Telecommunications Energy Conference*, 181–195. doi:10.1109/INTLEC.1986.4794425.

Zhou, D. (2005). *Synthesis of pwm dc-to-dc power converters*. Ph.D. thesis, California Institute of Technology. doi:10.7907/JEK6-1P40. URL <https://resolver.caltech.edu/CaltechETD:etd-01312005-163350>.