

Hemerson Farias da Silva

**CONVERSOR CC-CC DO TIPO BOOST SÍNCRONO
APLICADO À NANOSSATÉLITES**

Trabalho de Conclusão de Curso
submetido ao Departamento de
Engenharia Elétrica e Eletrônica da
Universidade Federal de Santa
Catarina para a obtenção do título de
Bacharel em Engenharia Eletrônica
Orientador: Dr. Leonardo Kessler
Slongo

Florianópolis
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silva, Hemerson
Conversor CC-CC tipo boost síncrono aplicado à
nanossatélites / Hemerson Silva ; orientador,
Leonardo Slongo, 2019.
74 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro
Tecnológico, Graduação em Engenharia Eletrônica,
Florianópolis, 2019.

Inclui referências.

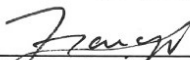
1. Engenharia Eletrônica. 2. Conversor CC-CC. 3.
Conversor boost síncrono. 4. nanossatélites. 5.
Electrical Power System. I. Slongo, Leonardo. II.
Universidade Federal de Santa Catarina. Graduação em
Engenharia Eletrônica. III. Título.

Hemerson Farias da Silva

Conversor CC-CC tipo boost síncrono aplicado à nanossatélites

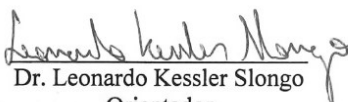
Esta Trabalho foi julgada adequada para obtenção do Título de Bacharel em Engenharia Eletrônica e aprovada em sua forma final pela Banca Examinadora

Florianópolis, 1 de Agosto de 2019.



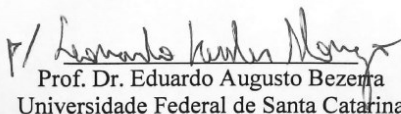
Prof. ^{PhD.} Fernando Rangel de Sousa
Coordenador do Curso

Banca Examinadora:

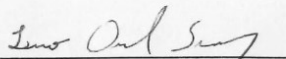


Dr. Leonardo Kessler Slongo
Orientador
Universidade Federal de Santa Catarina

Prof. Dr. Eduardo Augusto Bezena



Universidade Federal de Santa Catarina



Dr. Laio Oriel Semán
Universidade Federal de Santa Catarina

Este trabalho é dedicado a Deus, aos meus pais, a minha noiva e a minha irmã

AGRADECIMENTOS

Agradeço primeiramente a Deus que me deu força e sabedoria para realizar esse trabalho e me fortalecendo nos dias difíceis.

A minha família, que durante os meus anos na graduação me deu todo o suporte necessário, tendo paciência comigo nos meus dias de estresse e principalmente por me apoiarem tanto pra eu conseguir terminar com êxito.

Agradeço a minha noiva Thuanne Vargas, por todo o apoio e incentivo durante esse período que me deu, acreditando mais do que ninguém em minha capacidade de concluir esse projeto.

Por fim, e não menos importante, agradeço o meu orientador, Leonardo Slongo, por ter me orientado com tanta eficácia e me dar suporte nos momentos importantes, e também aos integrantes do FloripaSat que contribuíram em muito no meu desenvolvimento pessoal e técnico, além de trazer tantos desafios, agregando em muito em minha carreira profissional.

As coisas que são impossíveis aos homens, são possíveis a Deus (Jesus Cristo, 32 d.C.)

RESUMO

Conversores de tensão são amplamente utilizados atualmente em sistemas embarcados, e principalmente tratando-se de nanossatélites, eles possuem requisitos muito específicos, como baixa potência, alta eficiência, pequenas dimensões mecânicas e baixa complexidade de controle. Esse trabalho visa aumentar a eficiência do conversor CC-CC tipo boost com a topologia de conversor síncrono utilizando um MOSFET do tipo P para comutar, em vez de um diodo, com a finalidade de diminuir as perdas em condução. Esta topologia foi simulada e apresentou uma eficiência de no máximo 55,23% em comparação com a arquitetura anterior que possui eficiência de 75%.

Palavras-chave: Conversor CC-CC, Boost síncrono, Electrical Power System (EPS), Nanossatélites.

ABSTRACT

Voltage converters are widely used today in embedded systems. In space applications, and especially for nanosatellites, they have very specific requirements such as low power, high efficiency, small mechanical dimensions and low control complexity. This work aims to increase the efficiency of the DC-DC Boost converter with the synchronous topology by using a P-channel MOSFET as switch instead of a diode to avoid negative current. This topology has been simulated and presented a maximum efficiency of 55.23% compared to the previous architecture, which has an efficiency of 75%.

Keywords: DC-DC Converter, Synchronous Boost, Electrical Power System (EPS), Nanosatellites.

LISTA DE FIGURAS

Figura 1 - Diagrama do EPS	22
Figura 2 - Diagrama do conversor boost atual	24
Figura 3 - Circuito Equivalente de Painel Solar: (a) ideal e (b) com resistência série R_s e resistência shumpt paralela	25
Figura 4 - Curvas de corrente x tensão e potência x tensão	27
Figura 5 - Circuito base boost	28
Figura 6 - Boost com PWM em ON.....	28
Figura 7 - Boost com PWM em OFF	29
Figura 8 - Comportamento da corrente IL.....	30
Figura 9 - Vou/Vin x Duty Cycle.....	31
Figura 10 - Circuito do conversor boost intercalado.....	33
Figura 11- Circuito do conversor Boost síncrono	34
Figura 12 - Circuito do coversor boost no modelo PFM.....	34
Figura 13 – Conversor boost LC intercalado	35
Figura 14 - Conversor Buck-Boost Síncrono	35
Figura 15 - Proposta desse trabalho utilizando sincronismo	36
Figura 16 - Fluxograma do P&O.....	39
Figura 17- Conversor Síncrono	40
Figura 18 - Estágios 3 do conversor.....	40
Figura 19 – Blocos de simulação de atrasos.....	41
Figura 20 - Sistema Perturba e Observa.....	42
Figura 21 - Simulação do P&O	43
Figura 22 - Simulação detalhada P&O.....	43
Figura 23 - Simulação mais detalhada do P&O	44
Figura 24- Circuito Boost Proposto.....	45
Figura 25 - Placa Proposta	46
Figura 26 - Modo Up-Down.....	47
Figura 27 – Controle de PWM em modo Up-Down	47
Figura 28 - Análise do controle sem atraso	48
Figura 29 - Circuito do Driver.....	48
Figura 30 - Corrente no Indutor vs tempo	50
Figura 31 - PWMs gerados e corrente no indutor vs tempo.....	50
Figura 32 – Tensão da saída na do circuito vs tempo.....	51
Figura 33 - Potência na saída e na entrada para 250khz vs tempo	51
Figura 34 - Potência para 500 khz.....	51
Figura 35 - Resposta do Driver	52
Figura 36 - Tensão no Gate de PMOS e do NMOS	52
Figura 37 - Saída do Driver em 500 khz	53
Figura 38 - Tensão nos Gates, na saída e no Painel Solar	54

LISTA DE TABELAS

Tabela 1 - Parâmetros do módulo fotovoltaico de células solares.....	27
Tabela 2 - Comparação entre as propostas	36
Tabela 3- Resultados do circuito sem o painel solar	53

LISTA DE ABREVIATURAS E SIGLAS

MPPT – Maximum Power Point Tracking

EPS – Electrical Power System

1U – CubeSat com dimensões de no máximo 10x10x10 cm

RDS-ON - Resistência dreno source - RDS em estado ligado ON

SUMÁRIO

1	INTRODUÇÃO.....	21
1.1	MOTIVAÇÃO	22
1.2	DEFINIÇÃO DO PROBLEMA.....	23
1.3	ORGANIZAÇÃO DO TRABALHO	24
2	FUNCIONAMENTO DE CONVERSORES BOOST.....	25
2.1	PAINÉIS SOLARES.....	25
2.2	CONVERSORES BOOST.....	28
3	REVISÃO DO ESTADO DA ARTE	33
4	METODOLOGIA	37
4.1	MÉTODO PERTURBA E OBSERVA.....	37
4.2	CONVERSOR SÍNCRONO	39
5	PROPOSTA.....	41
5.1	PERTURBA E OBSERVA.....	41
5.2	CONVERSOR SÍNCRONO	44
6	RESULTADOS.....	50
6.1	SIMULAÇÃO	50
6.2	BANCADA	52
7	CONCLUSÃO E TRABALHOS FUTUROS	55
	REFERÊNCIAS.....	57
	APÊNDICE A – Descrição	61
	APÊNDICE B – Código Fonte do MPPT para Simulink	62
	APÊNDICE C – Firmware.....	63
	APÊNDICE D – Simulink do conversor boost	73
	APÊNDICE E – Simulink do MPPT com boost.....	74

1 INTRODUÇÃO

Nanossatélites possibilitaram médias empresas, institutos de pesquisa e principalmente universidades a participarem do desenvolvimento de tecnologias aeroespaciais, visto que seu custo é muito menor se comparado a satélites tradicionais, na casa de dezenas de milhares de dólares para o primeiro e centenas de milhões de dólares para o segundo, como também possuem um tempo de desenvolvimento menor (WOELLERT et al., 2011). São satélites que têm uma massa total variando de 1 a 10 kg, com dimensões reduzidas, e com seus devidos subsistemas para operar de forma eficiente em uma missão espacial. Atualmente, possuem diversas aplicações como: comunicação, sensoriamento remoto, monitoramento de radiação, campo magnético, climático, agricultura, desmatamento, astronômico, entre outros (VARGAS-CUENTAS; ROMAN-GONZALEZ, 2016).

Em 1999 mediante um trabalho entre Jordi Puig-Suari, um professor da *California Polytechnic State University (Cal Poly)* e Bob Twiggs, professor do *Space Systems Design Lab (SSDL)* foi desenvolvido o padrão chamado CubeSat (WOELLERT et al., 2011). Padrão este que possui como base a unidade 1U, um satélite em formato de cubo cuja dimensão e peso máximo são de 10cm x 10cm x 11.35cm (1U) e 1,33 kg, respectivamente (MEHRPARVA, 2014).

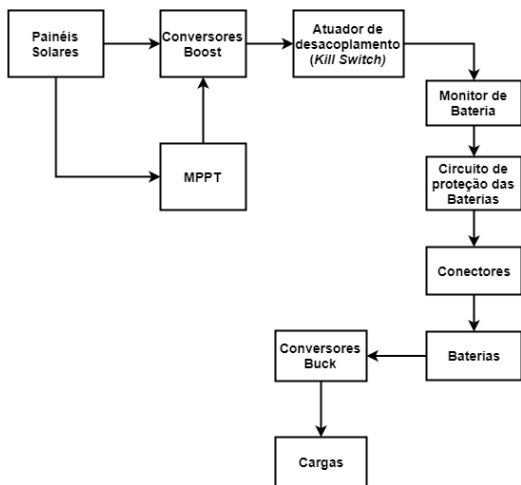
Os subsistemas de serviço de um nanossatélite são geralmente constituídos por: módulo de energia, responsável por armazenar a energia adquirida pelos painéis solares e distribuí-la para os outros sistemas; módulo de comunicação, que envia e recebe dados e comandos da Terra; módulo de processamento de dados, cuja finalidade é de manipular os dados a bordo do satélite e sincronizar ações e o fluxo de dados entre os módulos e a Terra, através do módulo de comunicação, ou armazenar os dados em um memória não volátil; módulo de controle de altitude (ADCS), cuja função é de medir a orientação do satélite e ajustá-la conforme a necessidade da missão. Além disso, em geral, os nanossatélites contêm um ou mais *payloads* (cargas úteis), que são os instrumentos/subsistemas para realizar os experimentos e testes em órbita (WOELLERT et al., 2011).

Os CubeSats usualmente são lançados em Órbitas Terrestres Baixas, cuja altitude está entre 160 e 2000 km acima da superfície terrestre. A faixa de órbita comumente utilizada por CubeSats é entre 450 e 650 km (WOELLERT et al., 2011).

Este trabalho trata especificamente do módulo de energia de um CubeSat, que em geral é constituído por: painéis solares dispostos em cada face do cubo, as baterias e um módulo de controle de carga e distribuição de energia.

Neste trabalho em particular, o módulo de energia, chamado EPS (Electrical Power System) possui os blocos conforme Figura 1. A energia adquirida pelos painéis solares é convertida pelo boost que tem como finalidade converter a tensão do painel solar na tensão de carga/operação da bateria, sendo que a tensão da bateria é maior que a dos painéis solares. Esse conversor é controlado mediante um algoritmo *Maximum Power Point Tracking* (MPPT) para obter a melhor eficiência possível na conversão, forçando a tensão da entrada a atingir seu ponto máximo de potência. Em seguida, a tensão das baterias é convertida novamente, dessa vez tem sua tensão diminuída para por fim, alimentar os outros módulos (EITERER, 2017).

Figura 1 - Diagrama do EPS



1.1 MOTIVAÇÃO

A energia adquirida pelos painéis solares é imprescindível para um CubeSat e impacta diretamente em todos os módulos. Por conta disso a eficiência dos conversores do EPS atinge diretamente na energia recebida pelas baterias. Perceber que o primeiro conversor boost projetado pela equipe FloripaSat possuía baixa eficiência se comparada

ao ideal, foi a maior motivação para realizar esse trabalho, visando fornecer a maior potência possível para as baterias e consequentemente para todo o satélite.

A segunda motivação é contribuir com a equipe FloripaSat, por se tratar de uma iniciativa de estudantes da UFSC que desenvolvem nanossatélites científicos. Os resultados deste trabalho poderão beneficiar a equipe no desenvolvimento dos próximos satélites

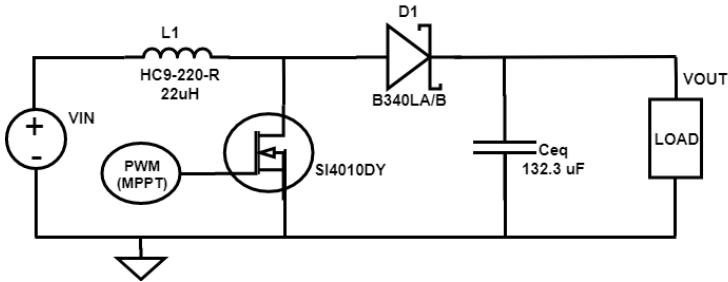
1.2 DEFINIÇÃO DO PROBLEMA

Atualmente, o FloripaSat, possui um conversor boost para cada par de painéis solares. Como o satélite dispõe de 6 painéis, ocorre que há, portanto, 3 conversores boost, sendo que cada um apresenta uma eficiência de no máximo 75%, isso por conta das resistências dos materiais, bem como pela queda de tensão presente no diodo quando está em condução. Quando o satélite está em órbita, são iluminados no máximo 3 painéis solares simultaneamente, que entregam no máximo 3W (FILHO et al, 2018). Contabilizando somente a perda nos conversores boost, nesse ponto onde há maior radiação solar, existe 750mW de perda de energia. Diante dessa baixa eficiência, percebe-se que os conversores boost em si podem ser melhorados. Via software, já foi implementado em outros trabalhos o algoritmo MPPT, porém, se tratando de hardware, o circuito é bem robusto, como pode ser visto na Figura 2.

Ao se analisar esse tipo de circuito, identifica-se que o diodo Schottky apresenta uma queda de tensão de 0,31 V. Ao ser submetido a uma corrente de 500 mA, a perda sobre ele será de 155 mW. Multiplicando-se isso por 3, que é o número de conversores, a perda total será de 465 mW.

É diante disso que se vê a necessidade de se aperfeiçoar o conversor boost de maneira a oferecer mais energia para os outros subsistemas e assim beneficiar o CubeSat como um todo.

Figura 2 - Diagrama do conversor boost atual



1.3 ORGANIZAÇÃO DO TRABALHO

No próximo capítulo são abordados os objetivos principais e os específicos deste trabalho.

No Capítulo 3, o funcionamento do boost é explicado em detalhes, analisando artigos e trabalhos que trataram do problema da eficiência do boost de variadas formas, via melhoramento do hardware.

Já no Capítulo 4 são abordadas as metodologias deste trabalho.

No quinto capítulo é abordada a proposta em si, com todos os detalhes de componentes e o que foi desenvolvido como solução neste trabalho.

No Capítulo 6 são tratados os resultados obtidos e comparação com o trabalho anterior da equipe FloripaSat.

Por fim, no Capítulo 7, são apresentadas as conclusões e os futuros trabalhos.

2 FUNCIONAMENTO DE CONVERSORES BOOST

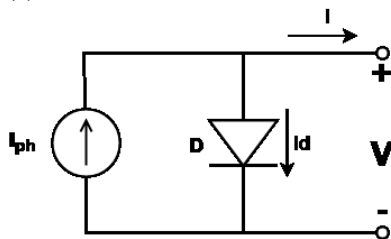
No CubeSat estudado, a tensão que os painéis solares fornece é menor do que a tensão necessária para carregar as baterias. Para que seja elevada a tensão são utilizados conversores do tipo boost. Este processo de conversão permite transformar a energia captada pelos painéis solares em energia armazenada na bateria. Antes de detalhar o funcionamento deste tipo de circuito, é importante entender como funcionam os painéis solares.

2.1 PAINÉIS SOLARES

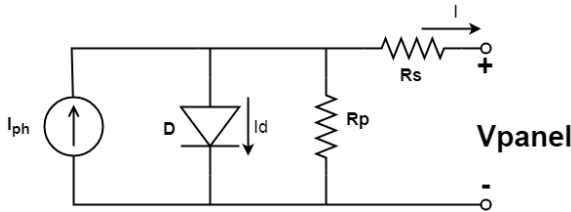
Os painéis solares têm seu comportamento baseado no efeito fotovoltaico, que ocorre quando há diferença de potencial na junção de dois materiais distintos no momento em que recebe radiação eletromagnética (luz). Quando o semiconductor recebe um fóton através dessa radiação, o elétron é excitado de um nível de energia inicial (E_i) até o nível de energia final (E_f). Mas isso ocorrerá somente se a energia for superior a energia de *bandgap*, caso contrário não há influência do fóton na célula solar. Após essa transição do elétron com energia maior que a energia de *bandgap*, o elétron passa da camada de valência para a camada de condução, gerando corrente elétrica. (JÄGER et al., 2014)

O comportamento do painel solar pode ser representado como uma fonte de corrente e um diodo em paralelo (para representar a junção p-n), como é visto na Figura 3 a. Mas, por conta das perdas geradas pela impureza do material e pela junção p-n, para representar corretamente o painel solar, é necessário acrescentar uma resistência em série R_s e outra em paralelo R_p como é apresentado na Figura 3 b (JÄGER et al., 2014).

Figura 3 - Circuito Equivalente de Painel Solar: (a) ideal e (b) com resistência série R_s e resistência shunt paralela (a)



(b)



A corrente do painel é proporcional à radiação aplicada, pois quanto maior a radiação, mais elétrons atravessam da banda de valência para a banda de condução e gera mais corrente elétrica.

Segundo Erdem e Erdem (2013), as equações que caracterizam o painel solar são as apresentadas abaixo. A corrente no diodo, depende da tensão do painel bem como a tensão no resistor R_s e é representada pela Equação (2.1):

$$I_D = I_o \cdot \left[e^{\left(\frac{V+R_s I_s}{V_t} \right)} - 1 \right] \quad (2.1)$$

Onde, I_o é a corrente de saturação do diodo, V_{panel} a tensão da saída, R_s a resistência em série, R_p a resistência em paralelo e V_t a tensão térmica do diodo.

A corrente em R_p (I_{R_p}) é dada pela equação (2.2), através da Lei de Ohm:

$$I_{R_p} = \frac{V+I.R_s}{R_p} \quad (2.2)$$

Fazendo (2.1) – (2.2) e aplicando a Lei de Kirchhoff, a corrente fornecida pelo painel solar é conforme a equação (2.3):

$$I = I_{ph} - I_o \cdot \left[e^{\left(\frac{V+R_s I_s}{V_t} \right)} - 1 \right] - \frac{V+I.R_s}{R_p} \quad (2.3)$$

Onde I_{ph} é a corrente no fotodiodo. Como esse comportamento é para uma célula solar, para conhecer o painel solar em si, temos que averiguar para a quantidade de células que constituem o painel solar, definido segundo a Tabela 1.

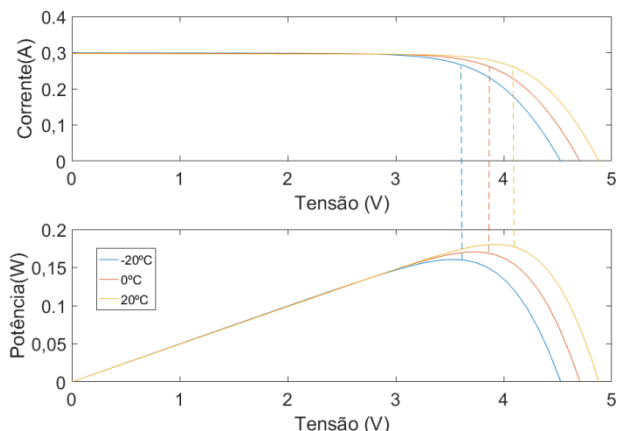
Tabela 1 - Parâmetros do módulo fotovoltaico de células solares

Parâmetros	Célula Fotovoltaica	Módulo Fotovoltaico
Número de células	I	n
Resistência série	R_s	$R_{s\text{módulo}} = n.R_s$
Resistência paralela	R_p	$R_{p\text{módulo}} = n.R_p$
Corrente fotogerado	I_{ph}	$I_{ph\text{módulo}} = n.I_{ph}$
Tensão de saída	V	$V_{\text{módulo}} = n.V$
Corrente de saída	I	$I_{\text{módulo}} = n.I$

Fonte: COELHO (2008)

Definido quantas células solares serão usadas no painel solar, o painel solar possui um comportamento distinto para cada temperatura. Assim sendo, há um gráfico de corrente por tensão para cada temperatura e consequentemente seu melhor ponto de máxima potência, cuja tensão específica para esse ponto é chamada de Tensão de Máxima Potência conforme é demonstrado na Figura 4. O Circuito responsável por atingir esse ponto de máxima potência é chamado de MPPT.

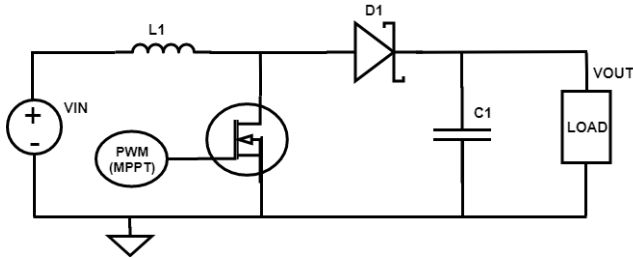
Figura 4 - Curvas de corrente x tensão e potência x tensão



2.2 CONVERSORES BOOST

Conversores do tipo boost são essenciais em nanossatélites, pois com eles é possível utilizar a energia adquirida pelos painéis solares e armazená-la nas baterias. A forma básica desse conversor é exposta na Figura 5:

Figura 5 - Circuito base boost

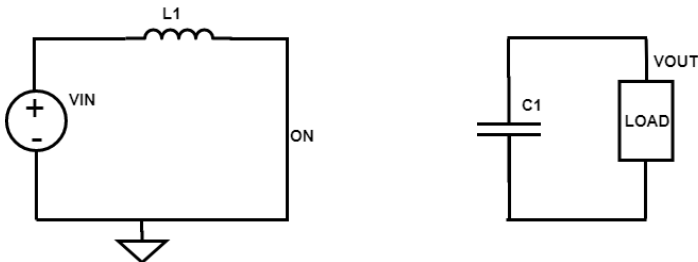


Quando o PWM está em nível alto (Figura 6), o transistor é fechado e o indutor é magnetizado, armazenando a energia da entrada. Enquanto isso, o capacitor descarrega a tensão nele armazenada. Logo,

$$V_{L1} = V_{in} \quad (2.4)$$

$$i_{c1} = -\frac{V_{out}}{R_{load}} \quad (2.5)$$

Figura 6 - Boost com PWM em ON

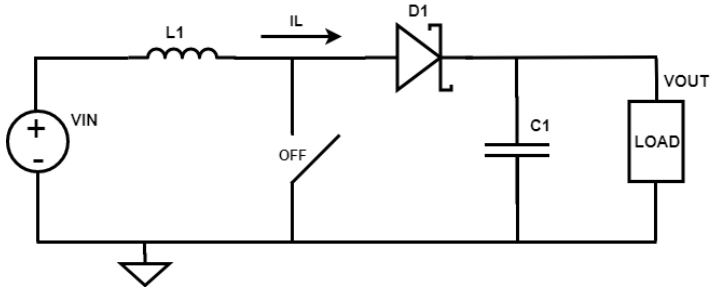


Quando o PWM fica em estado baixo, o transistor é aberto (Figura 7) e toda a corrente armazenada no indutor é descarregada, como é apresentado na Figura 8. Nesse caso, a tensão no indutor a corrente no capacitor são (ERICKSON; MAKSIMOVIC, 2001):

$$V_{L1} = V_{in} - V_{out} \quad (2.6)$$

$$i_{c1} = i_L - \frac{V_{out}}{R_{load}} \quad (2.7)$$

Figura 7 - Boost com PWM em OFF



O diodo em um período T , a potência total no indutor deve ser nula, portanto (ERICKSON; MAKSIMOVIC, 2001):

$$\int_0^T V_{L1} dt = V_{in} \cdot D \cdot T + (V_{in} - V_{out})(1 - D)T \quad (2.8)$$

$$V_{in}(D + 1 - D) - V_{out}(1 - D) = 0 \quad (2.9)$$

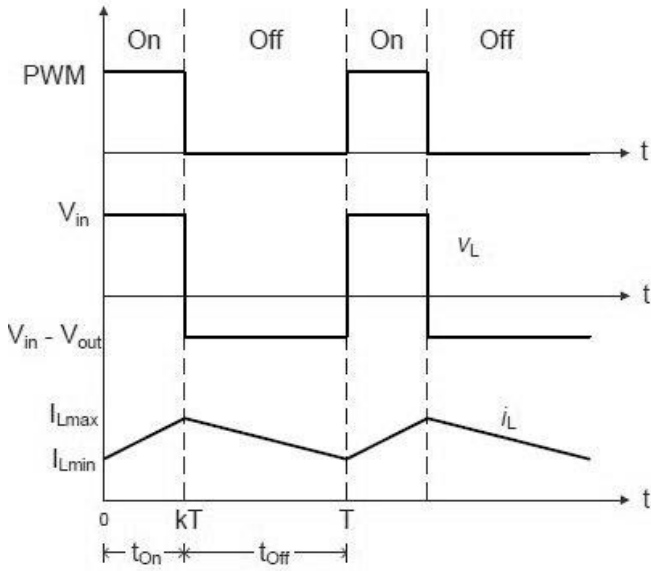
$$V_{in} = V_{out}(1 - D) \quad (2.10)$$

E, portanto, a saída pela entrada será inversamente proporcional ao tempo que o PWM fica em estado LOW:

$$\frac{V_{out}}{V_{in}} = \frac{1}{(1-D)} \quad (2.11)$$

A Figura 8 mostra o comportamento da corrente no indutor, quando é carregada (transistor fechado) e quando descarregada (transistor aberto).

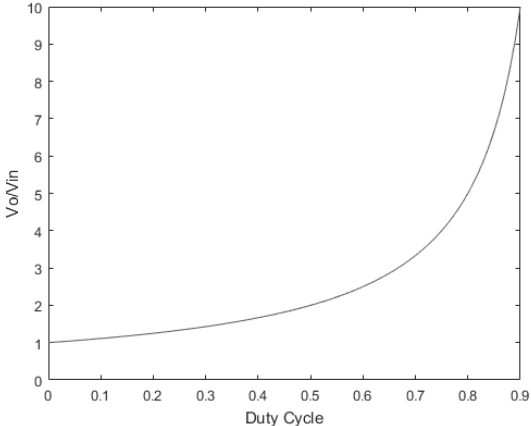
Figura 8 - Comportamento da corrente IL



Fonte: Pradhan e Panda (2018)

Conforme a equação 2.11, se fixada à entrada e a saída não tiver uma tensão, ela dependerá do valor do duty cycle, indicado na Figura 9.

Figura 9 - $V_{ou}/V_{in} \times \text{Duty Cycle}$

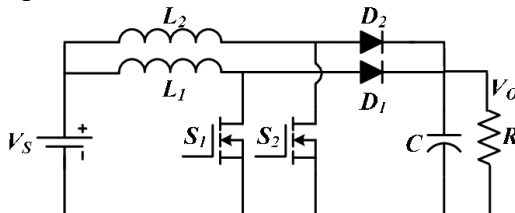


3 REVISÃO DO ESTADO DA ARTE

O conversor boost atual presente no EPS do FloripaSat utiliza o algoritmo Perturba e Observa. Possui a técnica de controle direto da tensão dos terminais dos painéis solares visando controlar sua tensão e captar a máxima potência que ele pode oferecer. No entanto, esse tipo de circuito tem perdas de potência consideráveis no diodo e na comutação e por conta disso, apresenta eficiência inferior a 75%, o que é uma desvantagem para esse tipo de circuito (Figura 2).

Banerjee, Ghosh e Rana (2016) propuseram a implementação de duas fases intercaladas onde são colocados dois indutores e dois diodos em paralelo com a finalidade de diminuir o ripple tanto das correntes quanto das tensões e melhorar a eficiência do conversor. E embora haja uma melhora significativa desta, ainda há perda de potência pelo diodo. Além disso, esse artigo não propõe nenhuma técnica de controle do boost, mas somente com uma tensão fixa na entrada e na saída (Figura 10).

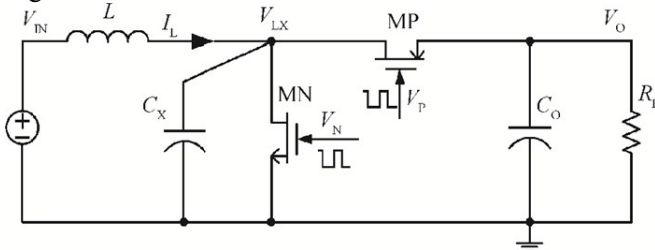
Figura 10 - Circuito do conversor boost intercalado



Fonte: Banerjee (2016)

Já Yajun, Xinquan e Qiang (2014) trazem como proposta ao circuito tradicional a substituição do diodo por um MOSFET, visando diminuir a perda de potência do diodo. Os *Gates* dos dois MOSFETs são controlados de forma inversa: enquanto um estiver em estado On o outro fique em estado Off e vice-versa. Se tratando de controle, é utilizado o dead time control (DTC) cuja finalidade é detectar quando a corrente no indutor é nula para controlar os transistores. Contudo, nesta proposta, tanto a tensão de entrada quanto a tensão de saída são fixas em 3.3 V.

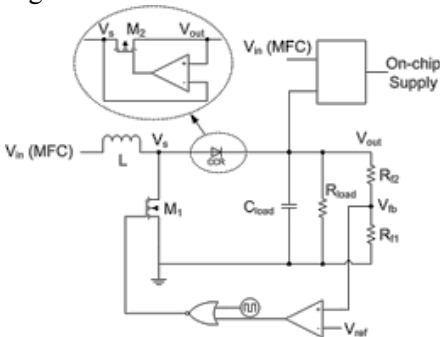
Figura 11- Circuito do conversor Boost síncrono



Fonte: Yajun(2014)

Zhang et al. (2015) apresentam um estudo para baixas tensões para uma célula combustível microbiana miniaturizada. Nela há uma troca de um diodo para um Mosfet retificador de controle cujo intuito é bloquear correntes nulas ou negativas que passam pelo indutor utilizando a técnica de controle semelhante a anterior: *adaptive dead-time control*(ADTC). Neste trabalho o controle da realimentação é implementada analógicamente, diferindo da proposta atual do boost (Figura 12).

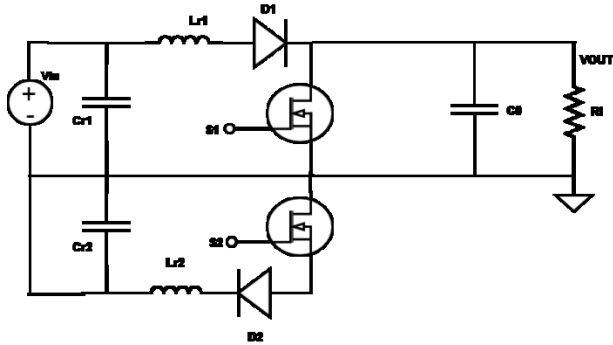
Figura 12 - Circuito do conversor boost no modelo PFM



Fonte: Zhang (2015)

Valipour, Ordonez e Mahdavi (2019) Apresentam boost do tipo LC intercalado onde o indutor, o capacitor e o MOSFET são duplicados e intercalados, visando melhorar a eficiência do circuito de fases intercaladas diminuindo o ripple. É também utilizado o método de controle de variação da frequência para controlar os *Gates* dos MOSFETs. Esse circuito, porém, possui perda de potência devido a presença dos diodos.

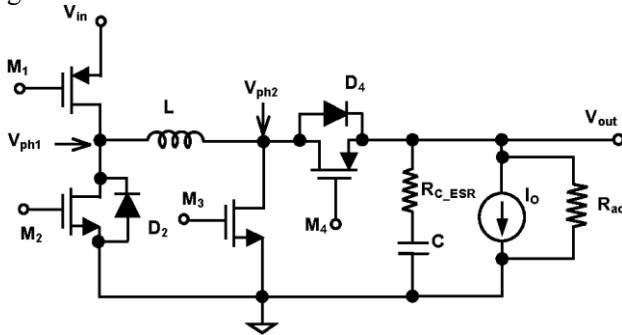
Figura 13 – Conversor boost LC intercalado



Fonte: Valipour, Ordonez e Mahdavi (2019)

Sahu e Rincon-mora (2004) propõe um circuito Buck-Boost síncrono com 4 FETs de potência, onde a cada par, é controlado em conjunto. Esse circuito possui um controle de compensação de frequência e utiliza do *Dead-Time control* (Figura 14).

Figura 14 - Conversor Buck-Boost Síncrono

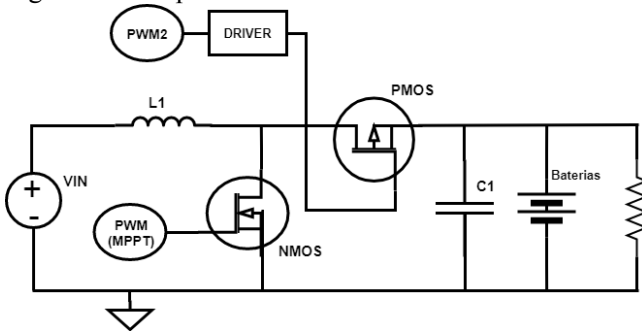


Fonte: Sahu e Rincon-mora (2004)

Vale lembrar que todos esses trabalhos citados acima utilizam como controle um circuito com realimentação, que exige compensação.

A estratégia desse trabalho é utilizar um conversor boost do tipo síncrono, onde O MOSFET de canal N é controlado via algoritmo MPPT e o MOSFET do canal P é controlado com base no NMOS, conforme demonstra na Figura 15.

Figura 15 - Proposta desse trabalho utilizando sincronismo



Fonte: Próprio Autor (2004)

Tabela 2 - Comparação entre as propostas

	MPPT	Técnica de circuito em paralelo	Dead Time Control	MOS de energia paralela	Remoção do diodo	Varição da Tensão de Entrada
Arquitetura EPS do FloripaSat	X					X
Duas Fases Intercaladas		X				
DTC			X		X	
Zhang			X	X	X	
Chen, Chang e Lin		X		X	X	
Valipour, Ordóñez e Mahdavi		X				
Nymand e Andersen		X		X		
Proposta	X				X	X

4 METODOLOGIA

4.1 MÉTODO PERTURBA E OBSERVA

Conhecendo o ponto de máxima potência, como é possível alcançá-lo? Existem diversas técnicas conhecidas.

Há muitos algoritmos MPPT desenvolvidos e implementados por pesquisadores. Em geral, as técnicas de algoritmos MPPT podem ser divididas em duas categorias, chamadas de método direto e método indireto. O método direto de algoritmos MPPT é independente do conhecimento das características dos módulos fotovoltaicos. Os algoritmos MPPT que estão inclusos nessa categoria são Perturba e Observa (P&O), método de condutância incremental (INCond.), feedback de tensão ou de corrente, método de lógica difusa e o método de rede neural. O método indireto requer conhecimento prévio do fotovoltaico gerador; é baseado numa relação matemática obtida de dados empíricos. Métodos como consulta de tabela, circuito aberto de tensão de painéis solares, curto-circuito de corrente fotovoltaica e outros algoritmos MPPT estão incluídos no método indireto. (NGAN; TAN, 2011, p. 22).

Como em outros trabalhos já foram discutidas essas técnicas, optou-se por utilizar o algoritmo que havia sido implementado no satélite anteriormente, a técnica Perturba e Observa. É mais frequentemente usada porque além de ser simples, necessita de poucos parâmetros (NGAN; TAN, 2011). O funcionamento desse algoritmo é mostrado na Figura 16.

Para rastrear a potência máxima, antes de entrar no loop é necessário estabelecer um valor de duty cycle inicial. No entanto, é muito difícil prever qual será a radiação exata na hora que o satélite for lançado, por isso, é definido um duty cycle médio de 0,4. Após isso é lida a corrente e a tensão, e calculada a potência no ponto, sendo

realizada a primeira iteração. No fim desta interação são atualizados os valores de corrente, tensão, potência e duty cycle em (n). Assim sendo, na segunda iteração já é possível decidir, através das comparações se o duty cycle será incrementado:

$$D(n) = D(n + 1) + \Delta D \quad (4.1)$$

Ou decrementado:

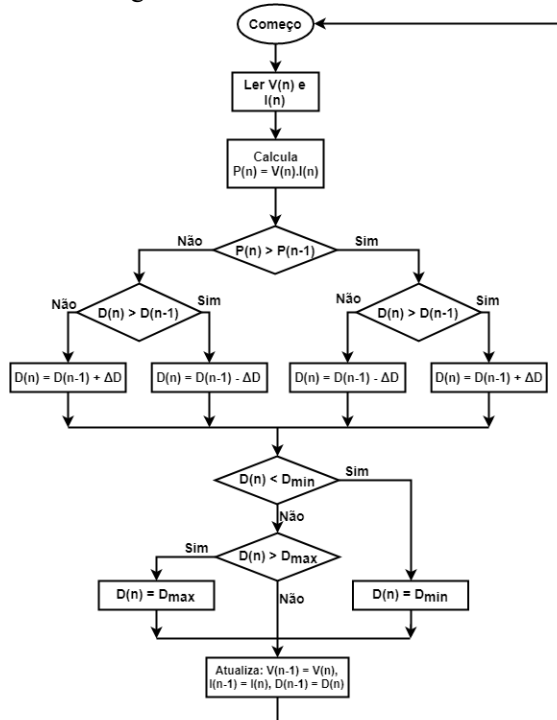
$$D(n) = D(n - 1) + \Delta D \quad (4.1)$$

Se a potência atual for maior que a potência do instante anterior ($P(n) > P(n-1)$) então quer dizer que o algoritmo está indo em direção a máxima potência e, portanto, deve manter seu sentido de perturbação do duty cycle, ou seja, se o duty cycle atual é maior que o anterior, então o sistema incrementa o duty cycle, caso contrário, diminui.

Mas se a potência atual for menor que a do instante anterior ($P(n) < P(n-1)$), o sistema deve inverter o sentido da perturbação, ou seja, se se o duty cycle atual for maior que o anterior, o sistema decrementa o duty cycle, se for menor, incrementa, conforme é mostrado na Figura 16.

Um problema desse algoritmo é a oscilação em torno da máxima potência a cada iteração. Este comportamento causa perda de energia. Para minimizar isso se deve considerar o trade-off entre o tamanho da perturbação (ΔD) e o tempo de resposta do circuito, visto que quanto maior a perturbação, menor o tempo de resposta, porém, maior é a oscilação em torno da máxima potência. Por outro lado, quanto menor a perturbação, mais tempo o sistema leva para chegar na máxima potência, mas menor será a flutuação em torno da potência máxima (FEMIA et al., 2005).

Figura 16 - Fluxograma do P&O

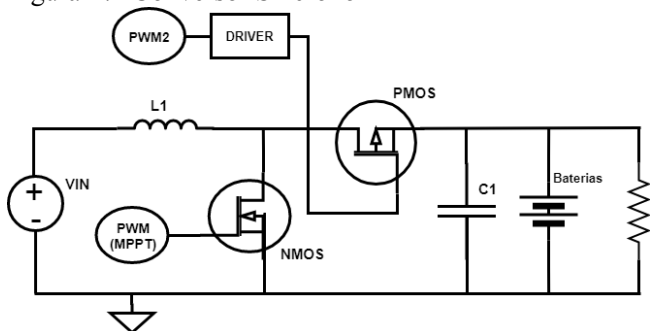


Fonte: Adaptado de COELHO (2008)

4.2 CONVERSOR SÍNCRONO

O conversor síncrono utiliza um MOSFET de canal P no lugar do Diodo (YAJUN; XINQUAN; QIANG, 2014) como demonstra na Figura 17, visando diminuir a perda de componente. O mosfet tipo P possui uma junção PNP do dreno para a fonte, com uma queda de tensão muito menor que do diodo utilizado.

Figura 17- Conversor Síncrono

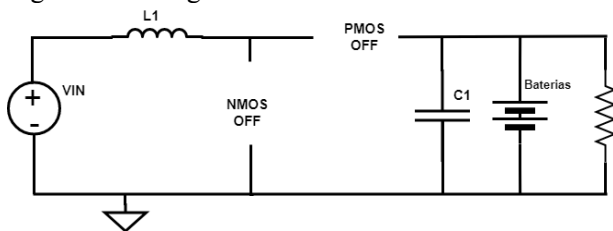


Os sinais de PWM e PWM2 são configurados de forma que haja três estágios de controle: NMOS fechado e PMOS aberto; NMOS aberto e PMOS fechado, e NMOS e PMOS aberto (YAJUN; XINQUAN; QIANG, 2014).

Como já foi mencionado, no primeiro estágio o indutor é carregado enquanto que a tensão no capacitor é descarregada na bateria. O segundo estágio, assim como no boost original, é quando a corrente que está armazenada no indutor é descarregada no capacitor.

Já o ultimo estágio, quando tanto o PMOS quanto o NMOS estão abertos (Figura 18), possui indutância e capacitâncias parasitas que podem descarregar o indutor (se ele ainda não estiver sido totalmente descarregado) por conta da capacitância entre o dreno e a fonte do MOSFET tipo N. Então, é interessante o MOSFET tipo P ligar logo depois que o NMOS abrir, tão logo quanto puder, para que a corrente seja descarregada na saída. Mas, deve ser analisado esse tempo mínimo para que não ocorra dos dois transistores ligarem ao mesmo tempo e a bateria ser descarregada.

Figura 18 - Estágios 3 do conversor



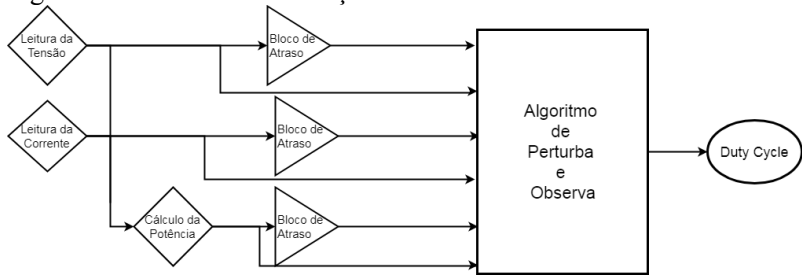
5 PROPOSTA

5.1 PERTURBA E OBSERVA

Como mencionado anteriormente, esta técnica será a utilizada neste projeto, tendo em vista a sua simplicidade bem como a sua já utilização no satélite. Para validação do algoritmo, foram realizadas simulações na plataforma Simulink, em conjunto com o Simscape, para simular de forma mais aproximada o painel solar.

É necessário gerar os sinais atrasados ($n-1$) para fazer as comparações. Para isso utilizou-se o bloco de memória para cada um (Figura 19).

Figura 19 – Blocos de simulação de atrasos



O código fonte do Simulink para a implementação do algoritmo proposto é o seguinte:

```
function duty_cycle = fcn(power, power_delay , voltage, voltage_delay, duty_cycle_delay)
```

```
if power > power_delay P(n) > P(n-1)  
if voltage > voltage_delay V(n) > V(n-1)  
    duty_cycle = duty_cycle_delay - 0.01; %Decr. Duty Cycle  
else  
    duty_cycle = duty_cycle_delay + 0.01; %Inc. Duty Cycle  
end  
else  
if voltage > voltage_delay  
    duty_cycle = duty_cycle_delay + 0.01; %Incr. Duty Cycle  
else  
    duty_cycle = duty_cycle_delay - 0.01; %Decr. Duty Cycle
```

```

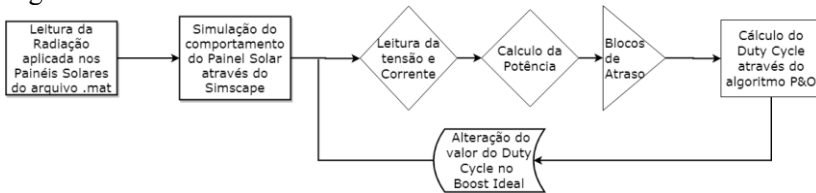
end
end

if duty_cycle_delay < 0.04 %se for menor que Dmin
    duty_cycle = 0.04; %Duty cycle mínimo
else
    if duty_cycle_delay > 0.95 %se for maior que Dmax
        duty_cycle = 0.95; %Duty cycle máximo
    else
        end
    end
end
end

```

Sendo assim, implementou também um boost ideal, visando analisar somente o MPPT. A Figura 20 retrata o comportamento do sistema.

Figura 20 - Sistema Perturba e Observa



A simulação desse algoritmo, mostrados na Figura 21 e na Figura 22, mostra que a potência vai em direção à potência máxima. Assim também acontece com o duty cycle, que fica ajustando a cada interação do laço.

Figura 21 - Simulação do P&O

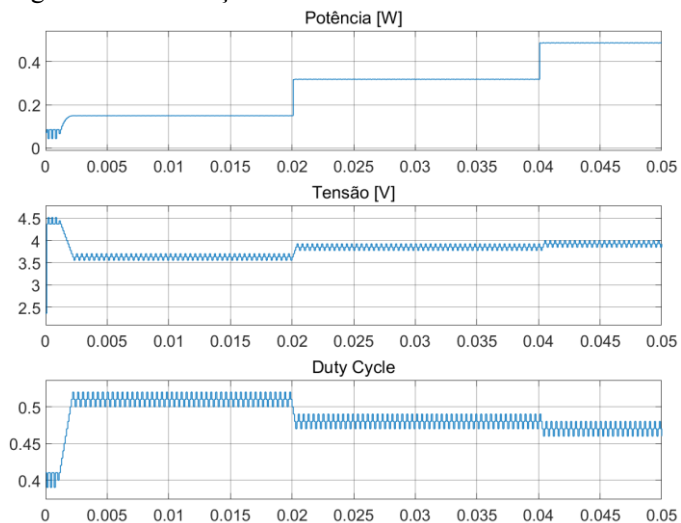
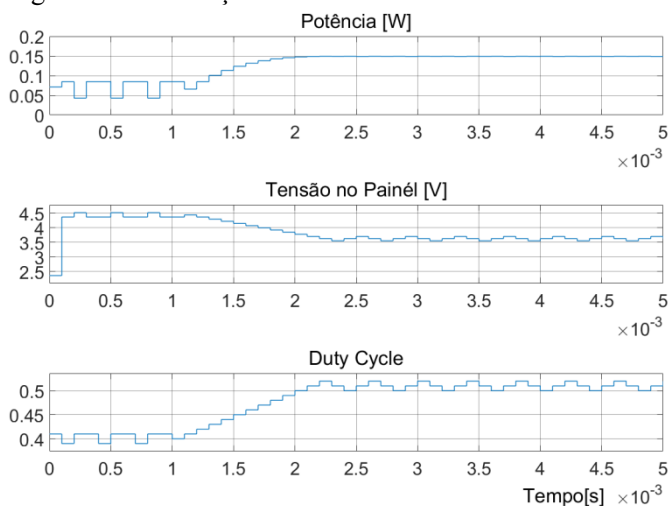
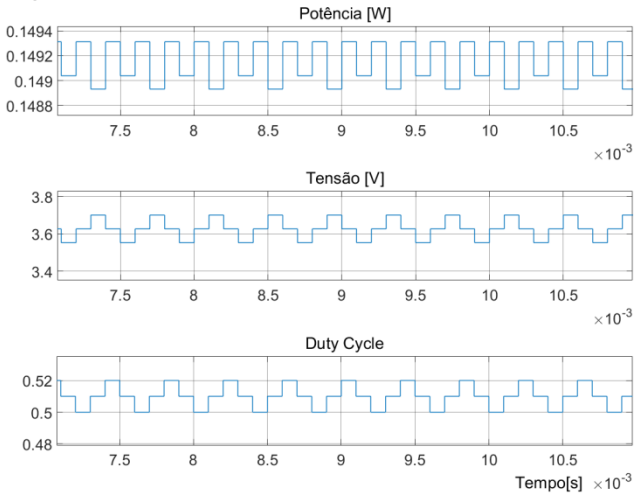


Figura 22 - Simulação detalhada P&O



Aproximando ainda mais, é mostrado na Figura 23 que a potência fica de fato oscilando por conta do algoritmo do perturba e observa implementado, que ajusta o duty cycle para sempre atingir esse ponto.

Figura 23 - Simulação mais detalhada do P&O



Se tratando dessa técnica aplicada no boost, utilizou-se o programa contido no Apêndice E, que realiza os mesmo ajustes que o das simulações.

5.2 CONVERSOR SÍNCRONO

Para definir os componentes do conversor, optou-se por utilizar os componentes que já continham anteriormente, demonstrados anteriormente na Figura 2, visto que já foram calculados para terem a melhor eficiência e, além disso, seria mais adequado para comparar os resultados. E para escolher o MOSFET canal P, foi pensado nas seguintes características: RDS-ON pequenos ($m\Omega$); resposta rápida (na ordem de poucos ns) e tensão de threshold não muito próximo de 0. Definido isso, optou-se por utilizar o MDD3754 que atende a esses requisitos. O circuito a seguir apresenta os componentes utilizados (Figura 24). A placa impressa, com seus devidos componentes está apresentada na Figura 25.

Figura 24- Circuito Boost Proposto

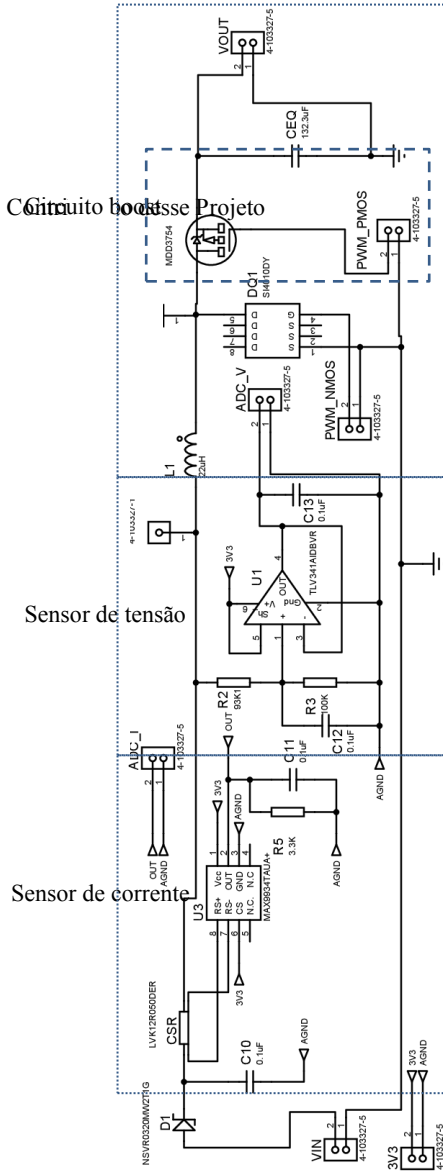
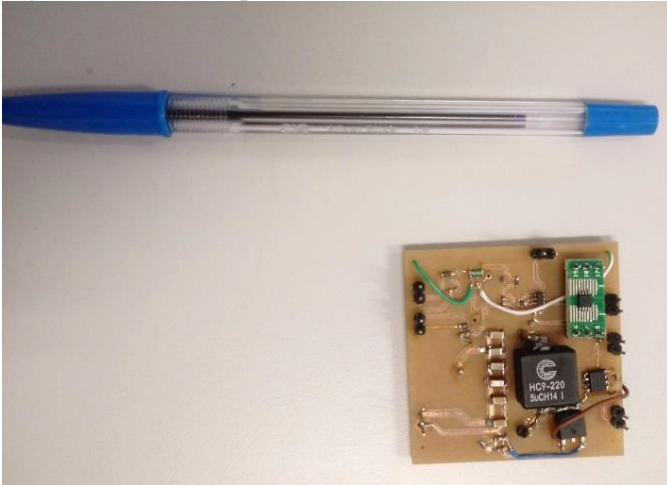


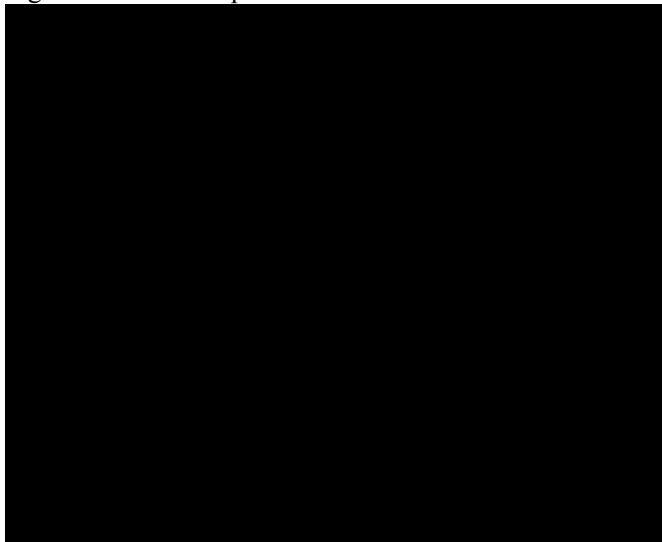
Figura 25 - Placa Proposta



Como já apresentado no capítulo anterior, os *Gates* dos MOSFETs precisam ser comutados de forma que o circuito tenha os três estágio de operação. Utilizou-se para isso a placa Directly Coupled do FloripaSat por conter um microcontrolador que já é utilizado em outras placas, por ter um cristal com frequência alta de 32 Mhz, necessário para ter muitos pontos para fazer as operações de ajuste do duty cycle. Optou-se por alterar a frequência de operação dos gates de 500 khz para 250 khz, porque, ao realizar em 500khz, percebeu-se que a onda PWM ao ser conectada ao NMOS, tinha seu tempo de subida e o de descida muito alto, comprometendo o controle do PMOS. Além disso, manteve os 64 pontos, já que o modo *Timer in Up/Down* tem o número de pontos dividido por 2.

Para definir o controle do PWM foi alterado o modo de contagem do contador de *Timer in Up Mode* para *Timer in Up/Down Mode*. No primeiro modo, ao atingir o nível de $TAxCCR1$, o pino de saída ficava em nível alto até o nível máximo $TAxCCR0$. No modo *Timer in Up/Down*, a contagem vai de 0 até $TAxCCR0$ e depois desce até 0 e o pino fica em nível alto quando essa contagem está maior que $TAxCCR2$, configurado para o modo de saída em 2: *Toggle/Set*, e 6: *Toggle/Reset* conforme demonstra a Figura 26:

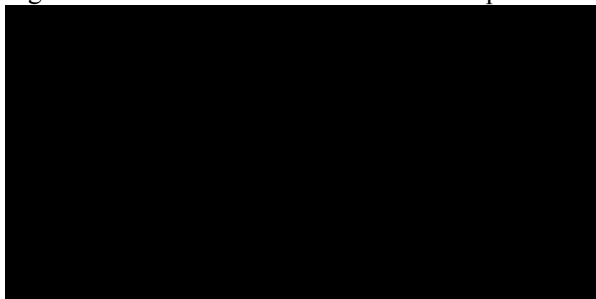
Figura 26 - Modo Up-Down



Fonte: User Guide MSP430x5xx and MSP430x6xx Family

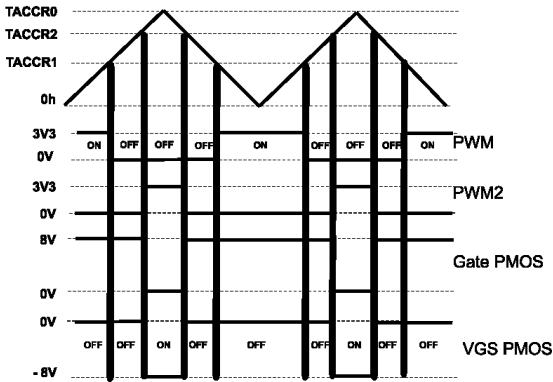
Por esse modo é possível ajustar tanto o PWM quanto PWM2 da seguinte forma: utilizar um valor menor de TACC2 para PWM2 e assim ter um duty cycle maior, conforme mostra a Figura 27.

Figura 27 – Controle de PWM em modo Up-Down



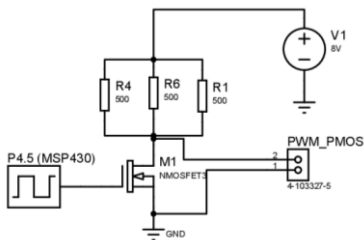
Esse sinal de PWM2 passa pelo driver, que eleva a tensão de 3,3V para 8V e com isso, habilita o controle do MOSFET tipo P. Como a saída está fixa na tensão da bateria (de 6,4 a 8,0 V) VGS será a tensão do driver – Tensão da bateria, que nesse caso está em 8V, como mostra a Figura 28.

Figura 28 - Análise do controle sem atraso



Para amplificar o sinal de 3,3V a 8V foi utilizado um PMOS com canal N com a configuração Fonte Comum, no qual opera em regime de saturação. Para definição dos resistores, é necessário fazer realizar um trade-off entre a velocidade de subida e de descida com a corrente no Driver. Foi escolhido então, o resistor de $166,67\ \Omega$ (Figura 29).

Figura 29 - Circuito do Driver



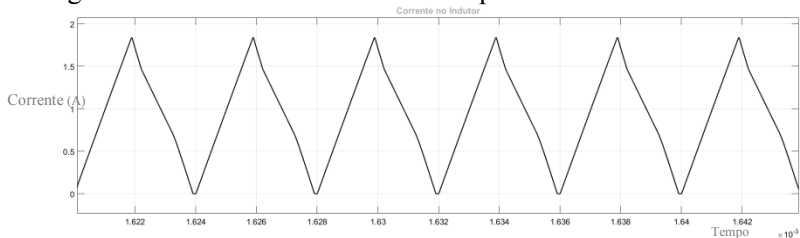
6 RESULTADOS

Primeiramente, o conversor boost foi simulado utilizando a plataforma Simulink para averiguar se o sistema se comportaria como o esperado e analisar se a solução de fato é viável.

6.1 SIMULAÇÃO

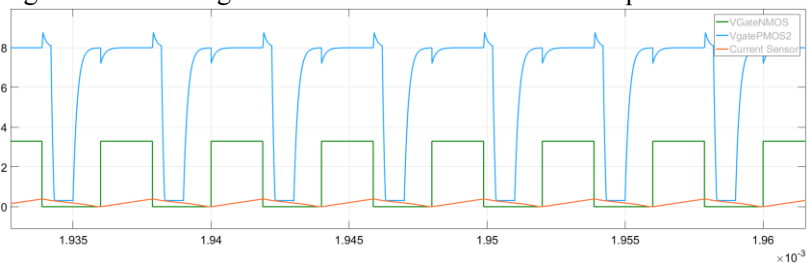
O valor do indutor foi escolhido tal que a condução seja crítica, ou seja, a corrente no indutor sempre seja positiva. Isso é percebido na simulação, conforme a Figura 30 abaixo.

Figura 30 - Corrente no Indutor vs tempo



Na Figura 31 abaixo, consta os atrasos contidos nos MOSFETs, captados na simulação. Percebe-se que o indutor de fato é carregado quando o NMOS está conduzindo, e descarregado quando não está. Além disso, há um atraso para comutar, mas esse atraso é pequeno, se comparado com o período da onda.

Figura 31 - PWMs gerados e corrente no indutor vs tempo



Foi colocada uma resistência de 50 na saída com a finalidade de simular uma carga, para averiguar o comportamento da variação da tensão e após a resposta transitória, o sistema se estabiliza e a tensão variou de acordo com a variação no duty cycle, como mostra a Figura 32 abaixo.

Figura 32 – Tensão da saída na do circuito vs tempo

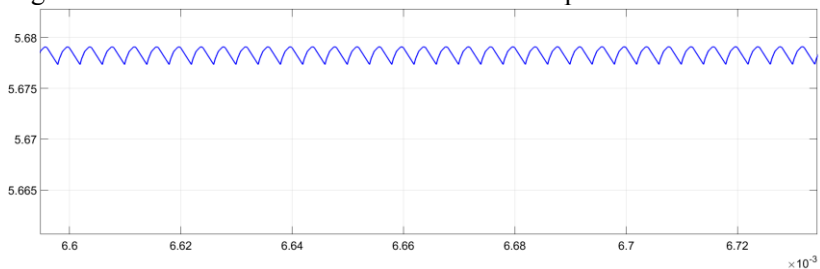
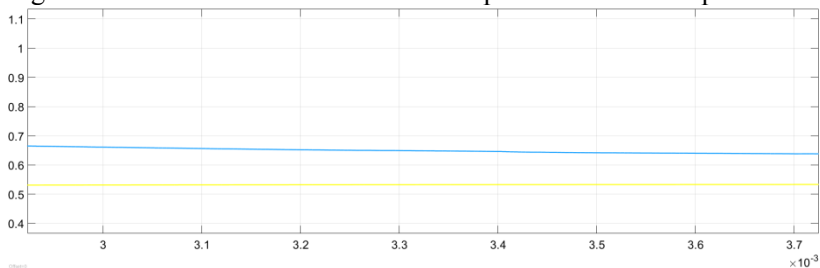


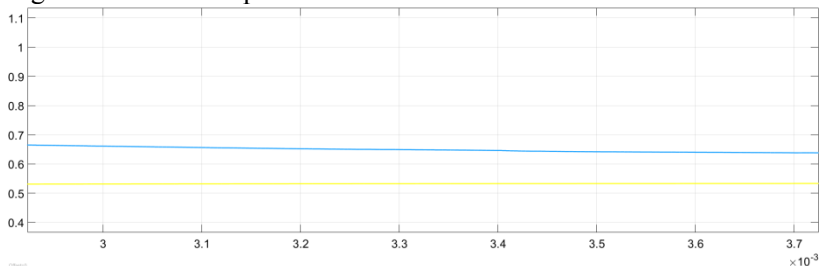
Figura 33 - Potência na saída e na entrada para 250khz vs tempo



Após decidir trocar para 250 khz, essa eficiência ficou comprometida e atingiu 61 % de pico (Figura 33)

Mas, quando foi simulado em 500 khz, o circuito apresentou uma eficiência de 84% (Figura 34).

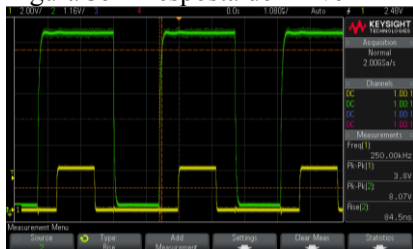
Figura 34 - Potência para 500 khz



6.2 BANCADA

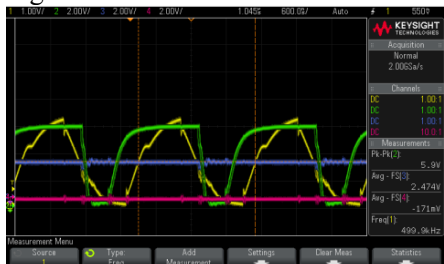
Em relação ao Driver, seu comportamento foi com pouco atraso, mas conforme o esperado (Figura 35). O Canal 1 é a tensão que irá ao Gate do NMOS. O Canal 2 é a tensão após a passagem do Driver.

Figura 35 - Resposta do Driver



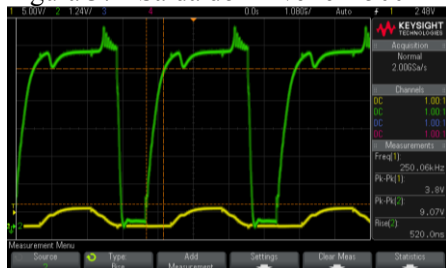
Mas, quando colocado na placa, esse atraso é bem maior, pois as capacitâncias de entrada tanto do NMOS quanto do PMOS são respectivamente, 3595 pF e 868 pF e a capacitância de saída são 1040 pF para o NMOS e 72 pF para o PMOS. Isso, fez com que fosse comprometido o tempo de resposta desses MOSFETs e por consequência, a forma de onda do controle. A Figura 36 mostra esse atraso obtido, sendo que o canal 1 é a tensão no Gate do NMOS, o canal 2 é a tensão no Gate do PMOS e o canal 3 é a tensão nos painéis solares. O tempo de subida do NMOS é de 500 ns, acarretando uma condução tanto do PMOS quanto do NMOS. Além disso, o problema de comutação se deve também ao material que é composto o MOSFET, pois para esse tipo de aplicação o ideal seria nitreto de gálio ou carbeto de Silício, por terem uma capacitância e tempo de resposta menor que os de silício.

Figura 36 - Tensão no Gate de PMOS e do NMOS



Para tratar esse problema, foi escolhido alterar a frequência de operação para 250 KHz, com a finalidade de analisar o comportamento do controle proposto (Figura 37). Conforme apresentado na figura, embora seja possível identificar um atraso, foi ajustado para que o duty cycle do PWM2 fosse 0,390625 maior que o do NMOS (Figura 36) para compensar esse atraso.

Figura 37 - Saída do Driver em 500 khz



Para averiguar a eficiência do circuito, aplicou-se uma tensão de entrada, com o objetivo de simular algumas placas solares, principalmente para diminuir a influência da potência que o driver precisa e obteve-se a Tabela 3. Um analisador de potência foi utilizado como carga eletrônica na saída do conversor boost, possibilitando configurar com precisão a tensão e a corrente de saída.

Tabela 3- Resultados do circuito sem o painel solar

Vin	Iin(mA)	Vout	Iout(mA)	Pin	Pout	Eficiência	V(Driver)	I(Driver)(mA)
2,496	50,4	8,00	-6,9	0,1914	-0,055	28,84%	8,0	8,20
3,299	76,1	8,00	-17,7	0,3166	0,1416	44,73%	8,0	8,20
3,400	82,3	8,00	-19,59	0,34468	0,1567	45,48%	8,0	8,10
3,5	85,6	8,00	-21,93	0,3644	0,1754	48,16%	8,0	8,10
3,600	91,4	8,00	-24,48	0,3978	0,1958	49,23%	8,0	8,60
3,699	95,8	8,00	-26,01	0,420792	0,2081	49,46%	8,0	8,30
3,8	98,1	8,00	-28,34	0,43998	0,2267	51,54%	8,0	8,40
3,899	101,1	8,00	-30,5	0,45744	0,2440	53,35%	8,0	7,90
3,870	98,9	8,00	-29,9	0,450	0,2392	53,17%	8,0	8,40
4,07	113,0	8,00	-36,45	0,528	-0,291	55,23%	8,0	8,40

Vemos, portanto que a eficiência do sistema é de no máximo 55,23%, o que é muito baixa, se comparado aos boosts tradicionais.

Montando todo o circuito, temos como resultado a Figura 38, que mostra a tensão nos Gates (Canais 1 e 2), a tensão na saída (Canal3) e a tensão nos painéis solar (Canal 4).

Figura 38 - Tensão nos Gates, na saída e no Painel Solar



E com esses dados, a eficiência ficou em torno de 41,41 %. Com todo o circuito montado as perdas por comutação se intensificam, e a eficiência é reduzida.

7 CONCLUSÃO E TRABALHOS FUTUROS

A topologia em si, teoricamente é melhor, conforme foi mostrado nas simulações. Contudo, por conta do tempo de subida e descida dos gates do NMOS serem elevados, houve um comprometimento do controle do PMOS e isso implicou em reduzir a frequência de operação. Mas, como não foi alterado o projeto de indutância e capacitância para operarem em 250 khz, a eficiência obtida não pode ser considerada real.

Além disso, quando é utilizado frequências tão altas como 500 kHz, os problemas de radio frequência aparecem, pois qualquer comprimento de fio passa a ser uma antena capaz de acrescentar ruído ao sistema e prejudicar na eficiência.

A técnica do MPPT rastreou o ponto de máxima potência, tanto nas simulações quanto nos testes. Além disso, a mudança do Up Mode para o Up/Down no controle dos *Gates* possibilitou o controle do PMOS ser de acordo com o do NMOS. Outra possibilidade seria implementar um algoritmo que compare quando a tensão do *Gate* do NMOS estiver em nível baixo e utilizar essa informação para ativar ou desativar o *Gate* do PMOS.

Reprojetar o hardware para 250 khz não é uma solução viável, uma vez que o tamanho dos indutores aumentaria e por consequência, ocuparia maior espaço na placa e esse estudo já foi realizado e optou-se por isso, operar em 500 khz.

Como sugestão de trabalhos futuros, pode trabalhar em melhorar a comutação do MOSFET, escolhendo outros materiais de fabricação como nitreto de gálio ou carbeto de Silício, para que o tempo de subida e de descida seja menor e assim seja possível utilizar a técnica do boost síncrono.

REFERÊNCIAS

BANERJEE, Subrata; GHOSH, Arnab; RANA, Niraj. Design and fabrication of closed loop Two-Phase Interleaved Boost Converter with Type-III controller. Iecon 2016 - 42nd Annual Conference Of The Ieee Industrial Electronics Society, [s.l.], p.3331-3336, out. 2016. IEEE.

COELHO, Roberto Francisco. ESTUDO DOS CONVERSORES BUCK E BOOST APLICADOS AO RASTREAMENTO DE MÁXIMA POTÊNCIA DE SISTEMAS SOLARES FOTOVOLTAICOS. 2008. 198 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, 2008.

EITERER, Bruno Vale Barbosa. Modelagem, Simulação e Testes de um Sistema de Energia Aplicado a Nanosatélites. 2017. 63 f. TCC (Graduação) - Curso de Engenharia Elétrica, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2017.

ERDEM, Zekiye; ERDEM, M. Bilgehan. A Proposed Model of Photovoltaic Module in Matlab/Simulink™ for Distance Education. Procedia - Social And Behavioral Sciences, [s.l.], v. 103, p.55-62, nov. 2013. Elsevier BV.

ERICKSON, Robert W.; MAKSIMOVIC, Dragan. **Fundamentals of Power Electronics**. 2. ed. Boulder: Springer, 2001. 883 p.

FEMIA, N. et al. Optimization of Perturb and Observe Maximum Power Point Tracking Method. **Ieee Transactions On Power Electronics**, [s.l.], v. 20, n. 4, p.963-973, jul. 2005. Institute of Electrical and Electronics Engineers (IEEE)

FILHO, Edegar Morsch et al. PRELIMINARY RESULTS FOR ENERGY HARVESTING MAXIMIZATION WITH FOCUS ON HEATER CONTROL OF BATTERIES IN NANOSATELLITES. In: III IAA LATIN AMERICAN CUBESAT WORKSHOP, 2018, Ubatuba. Ubatuba: Iaa, 2018. p. 1 - 11.

GANTA, Mounica; NIRUPA, Pallam Reddy; AKSHITHA, Thimmadi. Em: "Simple And Efficient Implementation Of Two-Phase Interleaved Boost Converter For Renewable Energy Source". International Journal Of Emerging Technology And Advanced Engineering. Chennai, p. 2250-2459. 4 abr. 2012

Jäger, Klaus et al. Solar Energy. Fundamentals, Technology and Systems.

MEHRPARVA, Arash. CubeSat Design Specification. 13. ed. California: The Cubesat Program, 2014. 16 p.

NGAN, Mei Shan; TAN, Chee Wei. A study of maximum power point tracking algorithms for stand-alone Photovoltaic Systems. **2011 Ieee Applied Power Electronics Colloquium (iapec)**, Skudai, p.22-27, abr. 2011.

NYMAND, M.; ANDERSEN, M.a.e.. High-Efficiency Isolated Boost DC–DC Converter for High-Power Low-Voltage Fuel-Cell Applications. *IEEE Transactions On Industrial Electronics*, [s.l.], v. 57, n. 2, p.505-514, fev. 2010

PRADHAN, Arjyadhara; PANDA, Bhagabat. A Simplified Design and Modeling of Boost Converter for Photovoltaic Sytem. **International Journal Of Electrical And Computer Engineering (ijece)**, Bhubneshwar, v. 8, n. 1, p.141-149, 4 jan. 2018.

SAHU, B.; RINCON-MORA, G.a.. A Low Voltage, Dynamic, Noninverting, Synchronous Buck-Boost Converter for Portable Applications. **Ieee Transactions On Power Electronics**, [s.l.], v. 19, n. 2, p.443-452, mar. 2004. Institute of Electrical and Electronics Engineers (IEEE)

VALIPOUR, Hamed; ORDONEZ, Martin; MAHDAVI, Mohammad. High Efficiency Interleaved LC Resonant Boost Topology: Analysis and Design. *IEEE Transactions On Power Electronics*, [s.l.], p.1-1, 2019.

WOELLERT, Kirk et al. Cubesats: Cost-effective science and technology platforms for emerging and developing nations. *Advances In Space Research*, [s.l.], v. 47, n. 4, p.663-684, fev. 2011. Elsevier BV.

XUE, Jing; LEE, Hoi. A 2-MHz 60-W Zero-Voltage-Switching Synchronous Noninverting Buck–Boost Converter With Reduced Component Values. **Ieee Transactions On Circuits And Systems II: Express Briefs**, [s.l.], v. 62, n. 7, p.716-720, jul. 2015. Institute of Electrical and Electronics Engineers (IEEE).

YAJUN, Li; XINQUAN, Lai; QIANG, Ye. Em: "High efficiency and low electromagnetic interference boost DC–DC converter". Journal Of Semiconductors. Xiam, p. 1-8. abr. 2014.

ZHANG, Xu et al. A High-Efficiency DC–DC Boost Converter for a Miniaturized Microbial Fuel Cell. IEEE Transactions On Power Electronics, [s.l.], v. 30, n. 4, p.2041-2049, abr. 2015.

APÊNDICE B – Código Fonte do MPPT para Simulink

```

1  function duty_cycle = fcn(power, power_delay , voltage,
   voltage_delay, duty_cycle_delay)
2
3  %%% Vi = Vo(1-D) %%%
4  if power > power_delay %%%Aumento D -> DIMINUI Vin
5      if voltage > voltage_delay
6          duty_cycle = duty_cycle_delay -0.01;%%INCREMENTA VIN
7      else
8          duty_cycle = duty_cycle_delay +0.01;%%DECREMENTA VIN
9      end
10 else
11     if voltage > voltage_delay
12         duty_cycle = duty_cycle_delay+0.01;%%DECREMENTA VIN
13     else
14         duty_cycle = duty_cycle_delay-0.01;%%INCREMENTA VIN
15     end
16 end
17 end
18 if duty_cycle_delay < 0.04
19     duty_cycle = 0.04;
20 else
21     if duty_cycle_delay > 0.95
22         duty_cycle = 0.95;
23     else
24         end
25 end

```

APÊNDICE C – Firmware

```
1 #include <msp430.h>
2 #include "PWM.h"
3 #include "eps_ADC.h"
4
5
6 /*
7  * main.c
8  */
9
10 #define Rsense 0.05
11 #define Gain 0.025
12 #define Rout 8000
13 #define increase 1
14 #define decrease 0
15
16 volatile unsigned int cont_20m = 0;
17 volatile unsigned int cont_100m = 0;
18 volatile float voltage = 0;
19 volatile float current = 0;
20 volatile unsigned int action = 0;
21 volatile float power = 0;
22 volatile float previous_power = 0;
23 volatile float previous_DUTY_CYCLE = 0.125;
24 volatile float PWM_PERIOD = 64;
25 volatile float PWM_DUTY_CYCLE = 0.25;
26
27 void config_MSP430(void);
28 void mppt(void);
29
30 void main(void) {
31
32
33     WDCTL = WDTPW | WDTHOLD; // Stop watchdog timer
34     config_MSP430();
35     while(1);
36 }
37
38
39 void config_MSP430(void)
40 {
41     clock_config();
42
43     TBCTL |= TBSSEL_2 + MC_3 ;
44     TBCTL |= TACLK;
45
46
47     P4SEL |= BIT1 | BIT5; // P4.1 and P4.5 option select
48     P4DIR |= BIT1 | BIT5; // P4.1 and P4.5 outputs
49
50
51     TBCCR0 = 64; // PWM Period
52     TBCTL1 = OUTMOD_2; // CCR1 PWM toggle/set
```

```

53     TBCCTL5 = OUTMOD_6;           // CCR5 PWM toggle/reset
54     TBCCR1 = 25;                  // CCR1 PWM first duty cycle
55     if (TBCCR1 <20){
56         TBCCR5 = 64-TBCCR1;      //
CCR5 PWM duty cycle
57     }
58         else if (TBCCR1 <32){
59             TBCCR5 = TBCCR1*2;
60         }
61
62         else if (TBCCR1 <48){
63             TBCCR5 = 55;
64         }
65         else if (TBCCR1 <62){
66             TBCCR5 = 60;
67         }
68
69         else{
70             TBCCR5 = 61;
71         }
72
73
74     P3DIR |= BIT4 + BIT5;        // used to debug
timer
75     TBCCTL0 = CCIE;              // TBCCR0 interrupt
enabled
76     TBCCR0 = 64;                 // timer B capture
and compare register
77     TBCTL = TBSSEL_2 + MC_3;    // SMCLK, contmode
78
79
80     P7SEL |= BIT5;              // ADCI-X A5 select
p7.4 function as ADC
81     P7SEL |= BIT7;              // ADCI-X A5 select
p7.4 function as ADC
82
83
84
85     REFCTL0 &= ~REFMSTR;        //
allows reference control in the ADC12 registers
86
87     ADC12CTL0 = ADC12ON | ADC12MSC | ADC12SHT0_15 |
ADC12SHT1_15 | ADC12REFON;      // turn on ADC12, enable
multiple sample and conversion, extend sampling time to avoid
overflow of results and turns on the internal reference
88     ADC12CTL1 = ADC12SHP + ADC12CONSEQ_1;
// use sampling timer, sequence of channels
89     ADC12MCTL0 = ADC12INCH_0 + ADC12SREF_2;
// ref+=VeRef, channel = A0
90     ADC12MCTL1 = ADC12INCH_1 + ADC12SREF_2;
// ref+=VeRef, channel = A1
91     ADC12MCTL2 = ADC12INCH_2 + ADC12SREF_2;
// ref+=VeRef, channel = A2
92     ADC12MCTL3 = ADC12INCH_3 + ADC12SREF_2;

```



```

// ref+=VeRef, channel = A3
93     ADC12MCTL4 = ADC12INCH_4 + ADC12SREF_2;
// ref+=VeRef, channel = A4
94     ADC12MCTL5 = ADC12INCH_5 + ADC12SREF_2;
// ref+=VeRef, channel = A5
95     ADC12MCTL6 = ADC12INCH_6 + ADC12SREF_2;
// ref+=VeRef, channel = A6
96     ADC12MCTL7 = ADC12INCH_7 + ADC12SREF_2;
// ref+=VeRef, channel = A7
97     ADC12MCTL8 = ADC12INCH_8 + ADC12SREF_2;
// ref+=VeRef, channel = A8
98     ADC12MCTL9 = ADC12INCH_9 + ADC12SREF_2;
// ref+=VeRef, channel = A9
99     ADC12MCTL10 = ADC12INCH_10 + ADC12SREF_2;
// ref+=VeRef, channel = A10
100    ADC12MCTL11 = ADC12INCH_11 + ADC12SREF_2;
// ref+=VeRef, channel = A11
101    ADC12MCTL12 = ADC12INCH_12 + ADC12SREF_2;
// ref+=VeRef, channel = A12
102    ADC12MCTL13 = ADC12INCH_13 + ADC12SREF_2;
// ref+=VeRef, channel = A13
103    ADC12MCTL14 = ADC12INCH_14 + ADC12SREF_2;
// ref+=VeRef, channel = A14
104    ADC12MCTL15 = ADC12INCH_15 + ADC12EOS + ADC12SREF_2;
// ref+=VeRef, channel = A15
105    ADC12CTL0 |= ADC12ENC;
// Enable conversions
106                                     // Enable
conversions
107
108    __bis_SR_register(GIE);
// enables interrupts
109 }
110
111 #pragma vector=TIMERB0_VECTOR
112 __interrupt void Timer_B (void)
113 {
114     if(cont_20m == 1) // period = CCR0 *
2 * cont_20m / clock => 20m = 40000*2*cont_20m/(8*10^6) => cont
= 1 (starts at 0)
115     {
116         P3OUT ^= BIT4; // toggle P3.4 to
determine period of timer (20ms)
117
118         measurement_ADC(); // read solar
panels values
119
120         voltage += ((ADC12MEM13*2.5/4095)*((300e3 +
180e3)/(300e3))); // add voltage to calculate mean value
121         current +=
(ADC12MEM15*2.5/4095)*(1/(0.05*0.025*8000)); //add
current to calculate mean value
122         cont_20m = 0;
// reset cont_20m

```

```

123     }
124     else
125     {
126         cont_20m++; //
increments cont_20m to achieve desired output period
127     }
128     if(cont_100m == 19) // period
= CCR0 * 2 * cont_100m / clock => 200m =
40000*2*cont_100m/(8*10^6) => cont = 19 (starts at 0)
129     {
130         P2OUT ^= BIT5; // toggle
P2.5 to determine period of timer (200ms)
131         voltage = voltage/10; // take
mean of 10 values of voltage
132         current = current/10; // take
mean of 10 values of current
133
134         mppt(); //Use MPPT
135
136         voltage = 0;
137         current = 0;
138
139         cont_100m = 0; // reset
cont_100m
140     }
141     else
142     {
143         cont_100m++; //
increments cont_100m to achieve desired output period
144     }
145 }
146
147 void mppt(void)
148 {
149     power = voltage*current; // take
input power
150     PWM_DUTY_CYCLE = TBCCR1 / PWM_PERIOD; //take
input duty cycle
151
152
153
154
155     if(power > previous_power) //compare
P(n)>P(n-1)
156     {
157         if (PWM_DUTY_CYCLE > previous_DUTY_CYCLE)
// Compare D(n)>D(n-1)
158         {
159             TBCCR1 = (PWM_DUTY_CYCLE+0.015625)*PWM_PERIOD;
// D(n) = D(n-1) + Delta_D
160         }
161         else
162         {
163             TBCCR1 = (PWM_DUTY_CYCLE-0.015625)*PWM_PERIOD;

```

```

// D(n) = D(n-1) - Delta_D
164     }
165     }
166     else
167     {
168         if (PWM_DUTY_CYCLE > previous_DUTY_CYCLE) //
Compare D(n)>D(n-1)
169         {
170             action = decrease;
// set test bit
171             TBCCR1 = (PWM_DUTY_CYCLE-0.015625)*PWM_PERIOD;
// D(n) = D(n-1) - Delta_D
172         }
173         else
174         {
175             action = increase;
// set test bit for next iteration
176             TBCCR1 = (PWM_DUTY_CYCLE+0.015625)*PWM_PERIOD;
// D(n) = D(n-1) + Delta_D
177         }
178     }
179     if ((PWM_DUTY_CYCLE*PWM_PERIOD) < PWM_PERIOD_MIN) {
//Compare D(n) < D(min)
180         TBCCR1 = PWM_PERIOD_MIN;
//Set D(n) = D_min
181     }
182     else if ((PWM_DUTY_CYCLE*PWM_PERIOD) > PWM_PERIOD_MAX) {
//Compare D(n) > D(max)
183         TBCCR1 = PWM_PERIOD_MAX; //Set
D(n) = D_max
184     }
185     else
186     {
187         TBCCR1 = TBCCR1; //if D_min
< D(N) < D(max), then D(n) = D(n)
188     }
189
190     if (TBCCR1 < 35) { //Compare D(NMOS) <
0.546875 //then Duty cycle of
191         TBCCR5 = TBCCR1+25; //then Duty cycle of
PMOS = D(NMOS) + 0.390625
192     }
193     else if (TBCCR1 < 45) { //Compare D(NMOS) <
0.703125 //then Duty cycle of
194         TBCCR5 = 60; //then Duty cycle of
PMOS = 0.9375
195     }
196
197     else {
198         TBCCR5 = 61; //else D(PMOS) =
0.953125
199     }
200
201     previous_power = power;

```

```
// save power value for next iteration
202     previous_DUTY_CYCLE = PWM_DUTY_CYCLE;
//save duty cycle for the next iteration
203
204 }
```

```
1  /*
2  * Test_PWM.h
3  *
4  *   Created on: 05/07/2016
5  *   Author: Hemerson Farias
6  */
7
8  #ifndef FSAT_EPS_MPPT_TEST_TEST_PWM_H_
9  #define FSAT_EPS_MPPT_TEST_TEST_PWM_H_
10
11
12 #define PWM_PERIOD_MIN  5           // pwm min
13
14 #define PWM_PERIOD_MAX  30         // pwm min
15
16
17 #endif /* FSAT_EPS_MPPT_TEST_TEST_PWM_H_ */
```

```

1  #include <msp430.h>
2  #include "eps_ADC.h"
3
4  /*
5   * main.c
6   */
7
8  volatile int adc0_msb=0x00;
9  volatile int adc0_lsb=0x00;
10 volatile int adc1_msb=0x00;
11 volatile int adc1_lsb=0x00;
12 volatile int adc2_msb=0x00;
13 volatile int adc2_lsb=0x00;
14 volatile int adc3_msb=0x00;
15 volatile int adc3_lsb=0x00;
16 volatile int adc4_msb=0x00;
17 volatile int adc4_lsb=0x00;
18 volatile int adc5_msb=0x00;
19 volatile int adc5_lsb=0x00;
20 volatile int adc6_msb=0x00;
21 volatile int adc6_lsb=0x00;
22 volatile int adc7_msb=0x00;
23 volatile int adc7_lsb=0x00;
24 volatile int msp_ts_msb=0x00;
25 volatile int msp_ts_lsb=0x00;
26
27 void measurement_ADC(void)
28 {
29     ADC12CTL0 |= ADC12SC;
30     // Start convn - software trigger
31     adc0_lsb= ADC12MEM0 & 0xff;
32     // bitwise and operation with '0000000011111111b' to extract
33     LSB
34     adc0_msb= ADC12MEM0>>8;
35     // 8 bit shift to extract MSB
36     adc1_lsb= ADC12MEM1 & 0xff;
37     // bitwise and operation with '0000000011111111b' to extract
38     LSB
39     adc1_msb= ADC12MEM1>>8;
40     // 8 bit shift to extract MSB
41     adc2_lsb= ADC12MEM2 & 0xff;
42     // bitwise and operation with '0000000011111111b' to extract
43     LSB
44     adc2_msb= ADC12MEM2>>8;
45     // 8 bit shift to extract MSB
46     adc3_lsb= ADC12MEM3 & 0xff;
47     // bitwise and operation with '0000000011111111b' to extract
48     LSB
49     adc3_msb= ADC12MEM3>>8;
50     // 8 bit shift to extract MSB
51     adc4_lsb= ADC12MEM4 & 0xff;
52     // bitwise and operation with '0000000011111111b' to extract
53     LSB
54     adc4_msb= ADC12MEM4>>8;

```

```
// 8 bit shift to extract MSB
40     adc5_lsb= ADC12MEM5 & 0xff;
// bitwise and operation with '000000001111111b' to extract
LSB
41     adc5_msb= ADC12MEM5>>8;
// 8 bit shift to extract MSB
42     msp_ts_lsb= ADC12MEM6 & 0xff;
// bitwise and operation with '000000001111111b' to extract
LSB
43     msp_ts_msb= ADC12MEM6>>8;
// 8 bit shift to extract MSB
44
45
46 }
```

```
1  /*
2  * eps_ADC.h
3  *
4  * Created on: 16/11/2016
5  * Author: Bruno Eiterer
6  */
7
8  #ifndef FSAT_EPS_MPPT_TEST_EPS_ADC_H_
9  #define FSAT_EPS_MPPT_TEST_EPS_ADC_H_
10
11
12  void measurement_ADC(void);
13
14
15  #endif /* FSAT_EPS_MPPT_TEST_EPS_ADC_H_ */
```

```

1  /**
2  * \file
3  *
4  * \brief Clock configuration source
5  *
6  * \author Bruno Vale Barbosa Eiterer
7  <brunoeiterer@gmail.com>
8  */
9  #include <msp430.h>
10 #include <stdint.h>
11 #include "clock.h"
12 #include "avoid_infinif_loops.h"
13
14 /**
15 * \brief Configures the MCU clocks
16 *
17 * Configures all of the MCU clocks frequency and source,
18 based on the following
19 * table:
20 *
21 * | Clock | Source | Divider | Frequency |
22 * |-----|-----|-----|-----|
23 * | MCLK  | XT2    | 4       | 8 MHz     |
24 * | SMCLK | XT2    | 1       | 32 MHz    |
25 * | ACLK  | XT2    | 2       | 16 MHz    |
26 *
27 * To do so, it configures P7.2 and P7.3 as XT2, enables
28 XT2 in UCSCTL6, enables
29 the internal load capacitor in UCSCTL6, chooses the
30 respective dividers in UCSCTL5 and
31 * finay chooses XT2 as the source in UCSCTL4. Then it
32 loops until all the fault
33 * flags are cleared, indicating that the external crystal
34 is working properly.
35 *
36 * \param -
37 * \return -
38 */
39
40 void clock_config(void){
41
42     config_avoid_infinif_loops(1000);           // Maximum
43     time on the loop: (TA2CCR0/clock): 1000/250000: 4ms
44     while((BAKCTL & LOCKBAK) && !avoid_infinif_loops())
45     // Unlock XT1 pins for operation
46         BAKCTL &= ~(LOCKBAK);
47
48     P7SEL |= BIT2 + BIT3; //XT2
49     UCSCTL6 &= ~XT2OFF;           // Enable XT1 & XT2
50     UCSCTL6 |= XCAP_3;           // Internal
51     load cap
52 }

```

```

46     UCCTL5 |= DIVM__4 + DIVS__1 + DIVA__2;
47     UCCTL4 |= SELA__XT2CLK + SELS__XT2CLK + SELM__XT2CLK;
// SMCLK = MCLK = ACLK = XT2
48
49     config_avoid_infinif_loops(250);           // Maximum time
on the loop: (TA2CCR0/clock): 250/250000: lms
50     do {
51         UCCTL7 &= ~(XT2OFFG | XT1LFOFFG | XT1HFOFFG |
DCOFFG); // Clear XT2,XT1,DCO fault flags
52         SFRIFG1 &= ~OFIFG;                   // Clear
fault flags
53     } while ((SFRIFG1 & OFIFG) && !avoid_infinif_loops());
54
55 }

```

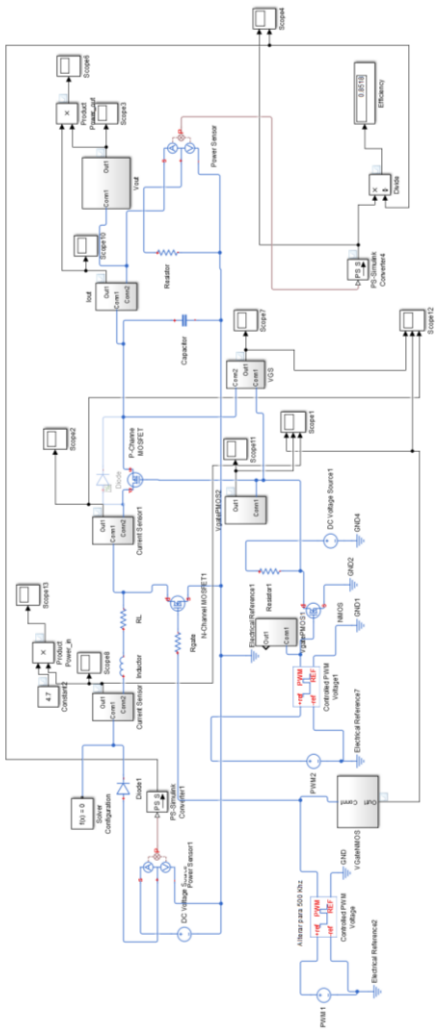
```

1  /**
2  * \file
3  *
4  * \brief Clock configuration header
5  *
6  * \author Bruno Vale Barbosa Eiterer
<brunoeiterer@gmail.com>
7  */
8
9  #ifndef CLOCK_H_
10 #define CLOCK_H_
11
12 /**
13 * \brief Configures the MCU clocks
14 */
15
16 void clock_config(void);
17
18
19
20 #endif /* CLOCK_H_ */

1  #ifndef AVOID_H_
2  #define AVOID_H_
3
4  uint8_t avoid_infinif_loops(void);
5  void config_avoid_infinif_loops(uint16_t i);
6
7  #endif

```


APÊNDICE D – Simulink do conversor boost



APÊNDICE E – Simulink do MPPT com boost

