



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

PROJETO DE CONTROLADORES APLICADO A CONVERSORES ESTÁTICOS DE CUBESAT

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA (PIBIC/INPE/CNPq)

Robert Uíliam Marin Viaro (UFSM, Bolsista PIBIC/CNPq)
E-mail: robertviarometal1@hotmail.com

Everson Mattos (CRS/CRS, Orientador)
E-mail: everson.mattos@inpe.br

Dr. Vinicius Foletto Montagner (CRS/CRS, Coorientador)
E-mail: vfmontagner@gmail.com

COLABORADORES

Dr. Gustavo Guilherme Koch (GEPOC/UFSM)

Outubro de 2021



SUMÁRIO

RESUMO	5
1. INTRODUÇÃO	6
2. OBJETIVO DO TRABALHO.....	8
3. MATERIAIS E MÉTODOS.....	9
a. Modelagem dos Conversores:.....	9
b. Análise da Primeira Etapa de Operação do conversor <i>buck</i>	10
c. Análise da Segunda Etapa de Operação do conversor <i>buck</i>	11
d. Análise da Primeira Etapa de Operação do conversor <i>boost</i>	13
e. Análise da Segunda Etapa de Operação do conversor <i>boost</i>	14
f. Primeira Etapa de Operação do conversor <i>buck-boost</i>	15
g. Segunda Etapa de Operação do conversor <i>buck-boost</i>	16
4. RESULTADOS:.....	18
5. Comprovação experimental do projeto	21
6. CONCLUSÃO.....	26
7. APÊNDICE	27
8. REFERÊNCIAS BIBLIOGRÁFICAS	32

LISTA DE ILUSTRAÇÕES

- Figura 1 – Algumas configurações de CubeSats
- Figura 2 – Algumas categorias de satélites de acordo com sua massa.
- Figura 3 – Diagrama dos principais subsistemas de um CubeSat e suas interações.
- Figura 4 – Algumas categorias de satélites de acordo com sua massa.
- Figura 5 – Conversores *buck*, *boost* e *buck-boost* estudados neste relatório.
- Figura 6 – Conversor *buck* e esquema de controle.
- Figura 7 – Etapa 1, para o conversor *buck*.
- Figura 8 – Etapa 2, para o conversor *buck*.
- Figura 9 – Etapa 1, para o conversor *boost*.
- Figura 10 – Etapa 2, para o conversor *boost*.
- Figura 11 – Etapa 1, para o conversor *buck-boost*.
- Figura 12 – Etapa 2, para o conversor *buck-boost*.
- Figura 13– Ambiente de desenvolvimento do App Designer do Matlab.
- Figura 14 – Ambiente de desenvolvimento do App Designer do Matlab, com código.
- Figura 15 – Interface do aplicativo em linguagem Matlab®.
- Figura 16 – Aba de solução “Funções de Transferência”.
- Figura 17 – Exemplo para o conversor *buck*, figuras de solução de projeto.
- Figura 18– Janela automática que permite exportar os resultados para o PSIM®.
- Figura 19– Aba que apresenta o projeto de controle analógico para o conversor selecionado.
- Figura 20– Exportação do arquivo de resultados de projeto e simulação em PSIM®.
- Figura 21– Resposta do controlador implementado em função de transferência e analógico, simulação no software PSIM®
- Figura 22– Resposta dinâmica dos controladores em função de transferência e analógico no PSIM®.
- Figura 23– Conversor e controle implementado.
- Figura 24– Variação de -50% de carga
- Figura 25– Variação de +50% de carga



LISTA DE SÍMBOLOS E ABREVIATURAS

- EPS – Electrical Power Subsystem (Subsistema Elétrico de Potência).
- COTS – Commercial off-the-Shell (Componentes Comerciais de Prateleira);
- PL – Payloads Subsystem (Cargas Úteis);
- OBCs – On Board Computer Subsystem (Subsistema do Computador de Bordo);
- Cs – Communication Subsystem (Subsistema de Comunicação);
- AsC – Attitude and Control Subsystem (Subsistema do Controle de Atitude);
- TsC – Command and Telemetry Subsystem (Subsistema de Comando e Telemetria);
- CsDH – Command and Data Handling Subsystem (Subsistema de Comanda e Manipulação de dados);
- EPS – Electrical Power Subsystem (Subsistema Elétrico de Potência);
- PI – Ação de Controle Proporcional Integral;
- PD – Ação de Controle Proporcional Derivativa;
- PID – Ação de Controle Proporcional Integral e Derivativa.



RESUMO

Os conversores CC-CC conectados em painéis fotovoltaicos são de suma importância em aplicações aeroespaciais. Pois, são os responsáveis em última instância pelo sucesso da missão. A falha em conversores ou do controle desses conversores, dependendo do tipo de falha, pode tornar a missão impraticável. Os conversores estáticos são sistemas dinâmicos não lineares que apresentam variações paramétricas e não paramétricas, as quais devem ser suportadas pelo controlador. Com isso, estratégias de projeto de controladores robustos podem ser implementadas obtendo-se bom desempenho.

As técnicas de controle podem ser classificadas em 2 grandes grupos. O primeiro grupo é o de controle clássico e utiliza os seguintes controladores: Controle On/Off; Controle auto-operado; Controle proporcional; Controle proporcional derivativo; Controle proporcional integral; Controle proporcional integral derivativo; Avanço de fase; Atraso de fase; Avanço e atraso de fase. O segundo grupo é composto pelo Controle moderno, o qual engloba os seguintes ramos: Controle multivariável; Controle adaptativo; Controle ótimo; Controle não linear; Controle preditivo; Controle robusto; Controle inteligente. Este trabalho tem seu foco em controle robusto, ótimo e no uso de técnicas metaheurísticas de projetos subótimos de controlares e observadores de estado.



1. INTRODUÇÃO

CubeSats são nanossatélites de grande interesse para uso em pesquisas espaciais em universidades e centros de pesquisa devido ao seu pequeno tamanho e custos mais baixos em comparação com os satélites convencionais. O termo CubeSat é um acrônimo formado pela palavra cube (cubo, em inglês) acrescida das três letras iniciais da palavra satélite. Esse termo é utilizado devido a sua forma geométrica de um cubo e cada unidade CubeSat (1 U) apresenta arestas de 10 cm obedecendo uma especificação de domínio público. A unidade Cubesat possui volume de 1 L e sua carga útil pode ter massa de até 1,33 kg. Combinações de unidades (1 U) são utilizadas para formar satélites maiores (2U, 3U, etc.) [1].

Pequenos satélites são cada vez mais usados em missões científicas e exploratórias, que incluem aquisição de dados para pesquisas em astrofísica, astrobiologia, física espacial, ciências lunares ou como plataforma para demonstração de tecnologia nas áreas de propulsão, comunicação e outras. Além disso, os CubeSats têm um papel muito importante no treinamento de recursos humanos para a área espacial com atenção especial aos alunos universitários. Missões espaciais desde sua concepção até a execução demandam altos custos e levam geralmente longos tempos de desenvolvimento que praticamente impede um aprendizado completo dentro do período típico de uma formação em nível superior.

O primeiro conceito de CubeSat surgiu em 1999 com Jordi Puig-Suari e Bob Twiggs da *California Polytech State University* (Cal Poly) e da Universidade de Stanford, respectivamente. A ideia inicial dos autores era fornecer aos alunos a oportunidade de participar de um projeto espacial completo, incluindo a construção, os testes e a operação de um artefato com características similares aos primeiros satélites lançados.

Outra característica importante dos CubeSats está associada a utilização de componentes comerciais comuns (ou COTS, do inglês comercial off-the-shell), ou seja, sem qualificação para uso no espaço. Isso resulta numa redução significativa em termos de custo e tempo de desenvolvimento. Em função de todo seu conjunto de características, sua utilização migrou rapidamente da universidade para outros setores, incluindo as indústrias aeroespaciais [2].

Assim, esses nanossatélites representam uma inovação interessante na área aeroespacial e, também, no modelo de negócios a eles associados, uma vez que maiores iniciativas estão no âmbito dos governos. Em 2017, o número de CubeSats lançados por ano superou o número de satélites convencionais.

As principais características dos CubeSats são:

- Satélites compostos por unidades padronizadas cúbicas de 1U (10x10x10 cm), formando composições de 2U, 3U, 6U etc.;
- Uso de sistemas de ejeção em órbita padronizados, denominados, por exemplo, P-POD (do inglês, *Poly Picosatellite Orbital Deployer*) ou SSPL (do inglês, *Space Shuttle Picosatellite Launcher*), capazes de liberar diversos satélites pela mesma interface. Existem sistemas comerciais destinados a satélites 1U, 2U, 3U e 6U;
- Uso de componentes COTS nos subsistemas de bordo.

Na Figura 1, são mostradas algumas configurações usuais de CubeSats.

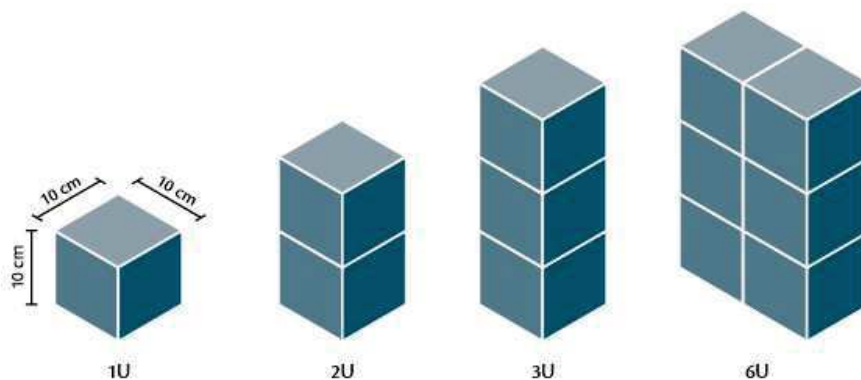


Figura 1 – Algumas configurações de CubeSats.

Fonte: adaptado de Radius Space.

Como mencionado anteriormente, Na Figura 2, é mostrada a classificação de alguns satélites conforme sua massa.

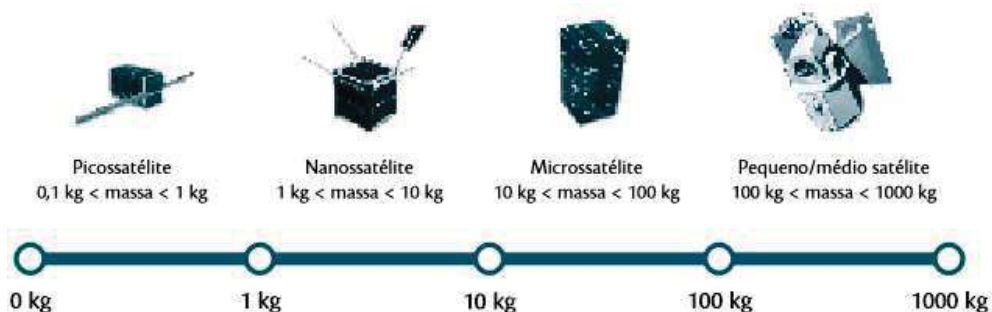


Figura 2 – Algumas categorias de satélites de acordo com sua massa.

Fonte: adaptada de OTE/CGEE.

No Brasil, pode-se citar alguns CubeSats que foram marcantes no contexto como o NanoSatC-Br1 desenvolvido no INPE-CRS.

Por traz da aparente simplicidade de um CubeSat, para torna-lo viável é necessário o trabalho conjunto de uma equipe multidisciplinar de engenheiros e técnicos que se dedicarão ao projeto de cada um dos subsistemas do nanosatélite. A Figura 3 mostra os principais subsistemas do satélite.

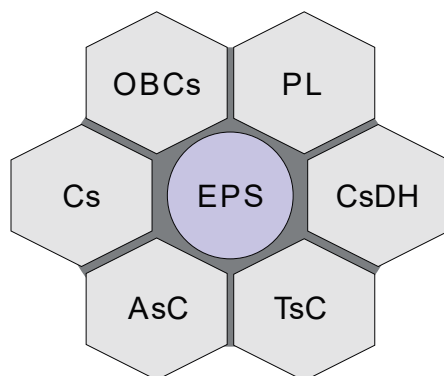


Figura 3 – Diagrama dos principais subsistemas de um CubeSat e suas interações.

Fonte: autoria própria.

onde: PL (Payloads Subsystem); OBCs (On Board Computer Subsystem); Cs (Communication Subsystem); AsC (Attitude and Control Subsystem); TsC (Command and Telemetry Subsystem); CsDH (Command and Data Handling Subsystem) e EPS (Electrical Power Subsystem).

O subsistema elétrico de potência (EPS) é um dos subsistemas mais importantes do nanosatélite, isso porque nas missões espaciais cerca de 29% dos defeitos em EPS para satélites de órbita LEO (CubeSats) são no submódulo de regulação e controle de energia [1],[2]. É a maior ocorrência de falhas em satélites se forem desconsiderados os eventos de falhas externos imprevisíveis que ocorrem em painéis fotovoltaicos, como por exemplo micrometeoros. Esses dados corroboram com a importância do tema EPS e controle nas pesquisas de engenharia aeroespacial. A falha nesse subsistema acarreta o comprometimento total da missão do satélite. O EPS pode ser subdividido em: fonte primária é tipicamente um arranjo de painéis solares; fonte secundária é tipicamente baterias recarregáveis; Distribuição de energia é o conjunto de cabos e conectores; e Regulação e controle é o conversor(es) e o controle das tensões e correntes desse(s) conversor(es) conforme mostrado na Figura 4.

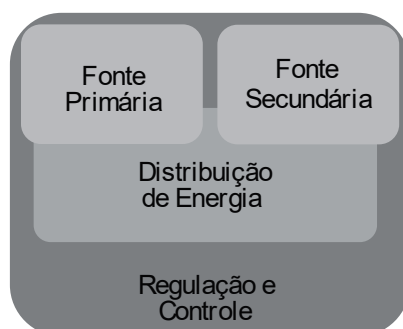


Figura 4 – Algumas categorias de satélites de acordo com sua massa.

Fonte: autoria própria.

2. OBJETIVO DO TRABALHO



Desenvolver um App em matlab para auxiliar no projeto de controle dos subsistemas elétricos de potência de pequenos satélites.

Objetivos Específicos:

Implementar contronrole PI, PID, PD para conversores CC-CC de topologia padrão (*boost*, *buck*, *buck-boost*) utilizados em sistemas de energia aeroespaciais;

Implementar controladores analógicos para os conversores CC-CC de topologia padrão (*boost*, *buck*, *buck-boost*) utilizados em sistemas de energia aeroespaciais.

3. MATERIAIS E MÉTODOS

a. Modelagem dos Conversores:

Os conversores analisados neste relatório são dos mostrados na Figura 5, ou seja, os conversores *buck*, *boost* e *buck-boost*, por serem os mais simples e frequentemente utilizados em sistemas aeroespaciais de pequeno porte [1].

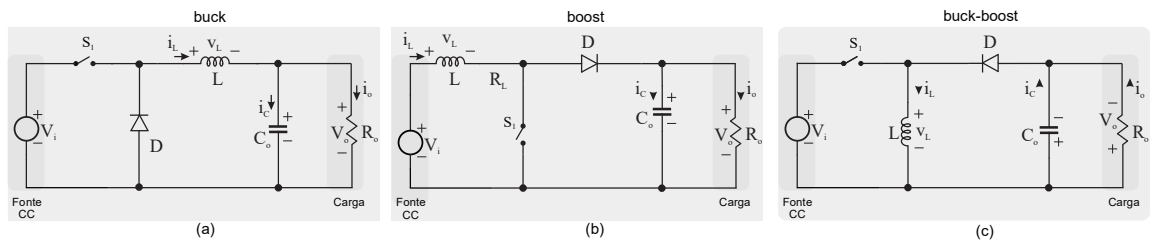


Figura 5 – Conversores *buck*, *boost* e *buck-boost* estudados este relatório.

Fonte: autoria própria.

Usando modelos médios de pequenos sinais em espaço de estados conforme [3] pode-se encontrar uma representação do modelo matemático do conversor na forma de funções de transferência utilizando para isso a transformada de Laplace [3-9]. A seguir, para o conversor *buck*, são mostrados os principais passo para modelagem do conversor em espaço de estados.

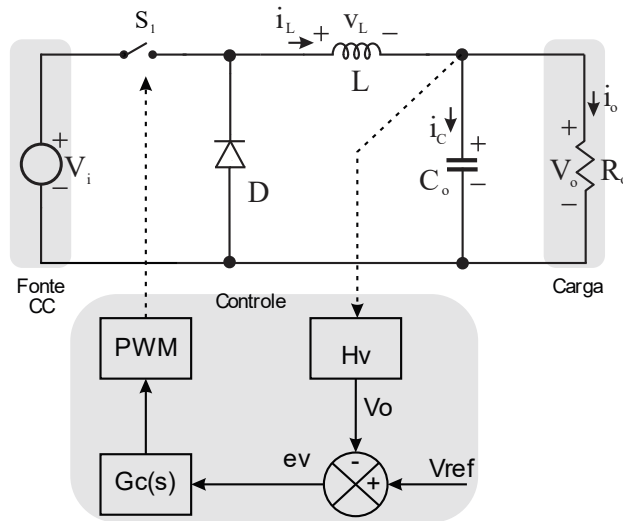


Figura 6 – Conversor *buck* e esquema de controle.

Fonte: autoria própria.

As equações dinâmicas do indutor e do capacitor são dadas por:

$$\begin{cases} v_L(t) = L \frac{d i_L(t)}{dt} \\ i_C(t) = C \frac{d v_C(t)}{dt} \end{cases} \quad (1)$$

Considera-se com variáveis de estado a tensão no capacitor e a corrente no indutor, passível de ser utilizado no projeto de controladores. Para encontrar um modelo de pequenos sinais, é necessário analisar o circuito do conversor em cada etapa de operação. Uma etapa de operação é o nome dado a cada circuito formado pela variação do conversor, ou seja, a chave aberta forma uma etapa e a chave fechada outra etapa de operação. Analisando o circuito da Figura 5 para cada etapa de operação tem-se:

b. Análise da Primeira Etapa de Operação do conversor *buck*

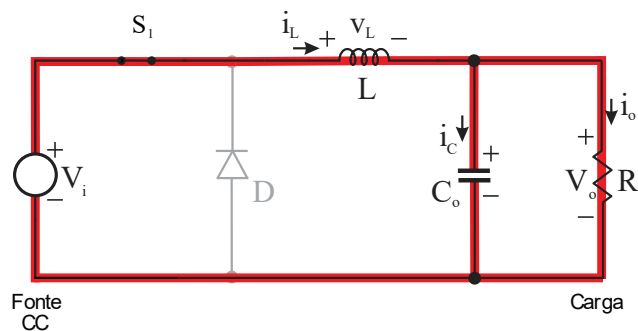


Figura 7 – Etapa 1, para o conversor *buck*.

Fonte: autoria própria.

Utilizando as Leis de Kirchhoff tem-se o sistema de equações mostrado em (2)

$$\begin{cases} -v_i(t) + v_o(t) + v_L(t) = 0 \\ i_c(t) = i_L(t) - i_o(t) \end{cases} \quad (2)$$

Isolando as variáveis de estado em cada uma das equações de (2) tem-se:

$$-v_i(t) + v_L(t) = 0$$

$$L \frac{d i_L(t)}{dt} = v_i(t)$$

$$i_{C_o}(t) = -i_{R_o}(t)$$

$$C_o \frac{d v_o(t)}{dt} = -\frac{v_o}{R_o}$$

Escrevendo as equações dinâmicas em forma matricial obtêm-se (3)

$$\dot{Kx} = Ax + Bu$$

$$\begin{bmatrix} L & 0 \\ 0 & C_o \end{bmatrix} \begin{bmatrix} \dot{i}_L \\ \dot{v}_o \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_o} \end{bmatrix} \begin{bmatrix} i_L \\ v_o \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} V_i \\ u \end{bmatrix} \quad (3)$$

c. Análise da Segunda Etapa de Operação do conversor *buck*

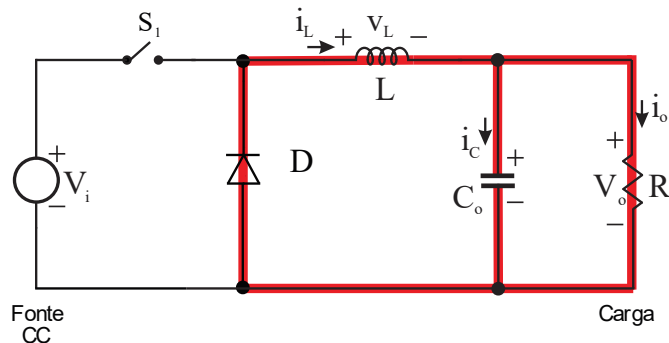


Figura 8 – Etapa 2, para o conversor *buck*.

Fonte: autoria própria.

Utilizando as Leis de Kirchhoff tem-se o sistema de equações mostrado em (2)

$$\begin{cases} v_L(t) + v_o(t) = 0 \\ i_L(t) = i_c(t) + i_{R_o}(t) \end{cases} \quad (4)$$

Isolando as variáveis de estado em cada uma das equações de (4) tem-se:



$$v_L(t) + v_o(t) = 0$$

$$\boxed{L \frac{d i_L(t)}{dt} = -v_o(t)}$$

$$i_c(t) = i_L(t) - i_{R_o}(t)$$

$$\boxed{C \frac{d v_c(t)}{dt} = i_L(t) - \frac{v_o(t)}{R_o}}$$

Escrevendo as equações dinâmicas em forma matricial obtêm-se (5)

$$K \dot{x} = Ax + Bu$$

$$\begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \dot{i}_L \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & -\frac{1}{R_o} \end{bmatrix} \begin{bmatrix} i_L \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} V_s \\ u \end{bmatrix} \quad (5)$$

Como o sistema é não linear, para aplicar as técnicas clássicas de controle deve-se linearizar em torno de um ponto de operação. Após perturbar as variáveis de controle, a entrada e a saída obtemos o modelo dado em (6).

O modelo médio não linear é dado por:

$$\begin{cases} K \frac{d\bar{x}}{dt} = [DA_1 + (1-D)A_2] \bar{x} + [DB_1 + (1-D)B_2] \bar{u} \\ \bar{y} = [DC_1 + (1-D)C_2] \bar{x} + [DE_1 + (1-D)E_2] \bar{u} \end{cases} \quad (6)$$

Perturbando e linearizando em torno do ponto de operação, assumindo as perturbações são muito menor do que o valor das variáveis no ponto quiescente, tem-se:

$$K \dot{\hat{x}} = \underbrace{A\hat{x} + Bu}_{DC} + [(A_1 - A_2)X + (B_1 - B_2)U] \hat{d}$$

$$\boxed{\dot{\hat{x}} = A_p \hat{x} + B_p \hat{u}}$$

$$\hat{y} = \underbrace{CX + EU}_{DC} + C\hat{x} + E\hat{u} + [(C_1 - C_2)X + (E_1 - E_2)U] \hat{d}$$

$$\boxed{\hat{y} = C_p \hat{x} + E_p \hat{u}}$$

Onde

$$A_p = [K^{-1}(DA_1 + (1-D)A_2)]$$

$$B_p = [K^{-1}[B] \quad K^{-1}[(A_1 - A_2)X + (B_1 - B_2)U]]$$

$$C_p = [(DC_1 + (1-D)C_2)]$$

$$E_p = [[B] \quad [(C_1 - C_2)X + (E_1 - E_2)U]]$$

Aplicando a transformada de Laplace tem-se:

$$\boxed{\frac{\hat{y}}{\hat{u}}(s) = \left[C_p (sI - A_p)^{-1} B_p + E_p \right]}$$

onde

$$A_p = \left[K^{-1}(DA_1 + (1-D)A_2) \right]$$

$$B_p = \left[K^{-1}[B] \quad K^{-1}[(A_1 - A_2)X + (B_1 - B_2)U] \right]$$

$$C_p = \left[(DC_1 + (1-D)C_2) \right]$$

$$E_p = \left[[B] \quad [(C_1 - C_2)X + (E_1 - E_2)U] \right]$$

d. Análise da Primeira Etapa de Operação do conversor *boost*

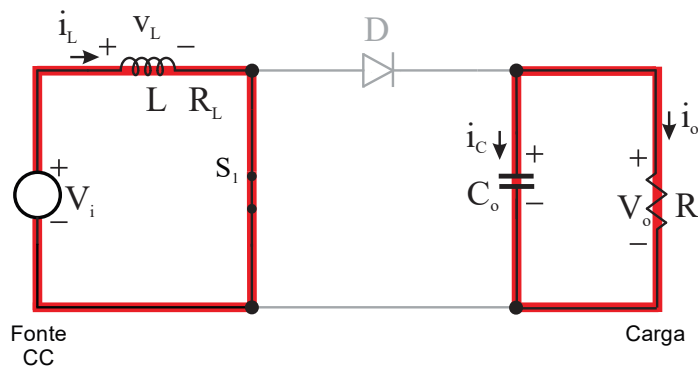


Figura 9 – Etapa 1, para o conversor *boost*.

Fonte: autoria própria.

Utilizando Kichhoff pode-se escrever as seguintes equações para o circuito da Figura 9. O sistema de equação dado em (7) é para a primeira etapa de operação do conversor *boost*.

$$\begin{cases} -v_i(t) + v_L(t) = 0 \\ i_c(t) = -i_o(t) \end{cases} \quad (7)$$

Resolvendo o sistema de equação de (7) e usando as equações (1), obtêm-se:

$$-v_i(t) + v_L(t) = 0$$

$$\boxed{L \frac{d i_L(t)}{dt} = v_i(t)}$$

$$i_c(t) = -i_o(t)$$

$$\boxed{C \frac{d_{v_o}(t)}{dt} = -\frac{v_o}{R_o}}$$

Escrevendo de forma matricial resulta em (8)

$$\dot{Kx} = Ax + Bu$$

$$\underbrace{\begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix}}_K \underbrace{\begin{bmatrix} \dot{i}_L \\ \dot{v}_o \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_o} \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} i \\ v_o \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{B_1} \underbrace{[V_i]}_u \quad (8)$$

e. Análise da Segunda Etapa de Operação do conversor *boost*

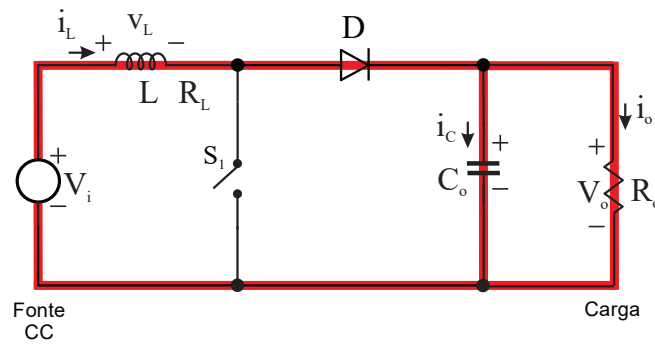


Figura 10 – Etapa 2, para o conversor *boost*.

Fonte: autoria própria.

Para essa etapa o sistema de equações resultantes da aplicação de Kichhoff é dado em (9). Proveniente da análise do circuito mostrado na Figura 10.

$$\begin{cases} -v_i(t) + v_L(t) + v_o(t) = 0 \\ i_L(t) = i_c(t) + i_o(t) \end{cases} \quad (9)$$

Resolvendo o sistema de equação de (9) e usando as equações (1), obtêm-se:

$$-v_i(t) + v_L(t) + v_o(t) = 0$$

$$\boxed{L \frac{d_{i_L}(t)}{dt} = v_i(t) - v_o(t)}$$

$$i_c(t) = i_L(t) - i_o(t)$$

$$\boxed{C \frac{d_{v_o}(t)}{dt} = i_L(t) - \frac{v_o(t)}{R_o}}$$

Escrevendo de forma matricial resulta em (10)

$$K\dot{x} = Ax + Bu$$

$$\underbrace{\begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix}}_K \underbrace{\begin{bmatrix} \dot{i}_L \\ \dot{v}_o \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & -1 \\ 1 & -\frac{1}{R_o} \end{bmatrix}}_{A_2} \underbrace{\begin{bmatrix} i_L \\ v_o \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{B_2} \underbrace{[V_i]}_u \quad (10)$$

Aplicando a mesma metodologia utilizada para o conversor *buck* pode-se encontrar a função de transferência que relaciona a tensão de saída V_o com a razão cíclica, dada por

$$G_{(V_o, D)}(s) = \frac{V_i}{(D-1)^2} \frac{R_o(1-D)^2 - Ls}{CLR_o s^2 + Ls + R_o(1-D)^2} \quad (11)$$

f. Primeira Etapa de Operação do conversor *buck-boost*

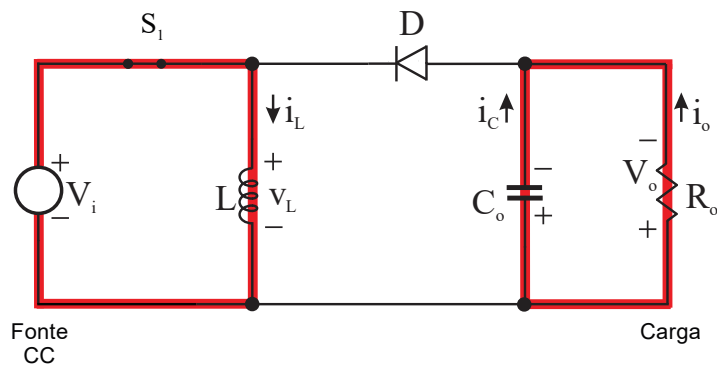


Figura 11 – Etapa 1, para o conversor *buck-boost*.

Fonte: autoria própria.

Utilizando Kichhoff pode-se escrever as seguintes equações para o circuito da Figura 9. O sistema de equação dado em (12) é para a primeira etapa de operação do conversor *boost*.

$$\begin{cases} -v_i(t) + v_L(t) = 0 \\ i_c(t) = -i_o(t) \end{cases} \quad (12)$$

Resolvendo o sistema de equação de (12) e usando as equações (1), obtêm-se:

$$-v_i(t) + v_L(t) = 0$$

$$\boxed{L \frac{d i_L(t)}{dt} = v_i(t)}$$

$$i_c(t) = -i_o(t)$$

$$\boxed{C \frac{d v_o(t)}{dt} = -\frac{v_o}{R_o}}$$

Escrevendo de forma matricial resulta em (13)

$$\dot{K}x = Ax + Bu$$

$$\underbrace{\begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix}}_K \underbrace{\begin{bmatrix} \dot{i}_L \\ \dot{v}_o \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_o} \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} i \\ v_o \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{B_1} \underbrace{[V_i]}_u \quad (13)$$

g. Segunda Etapa de Operação do conversor *buck-boost*

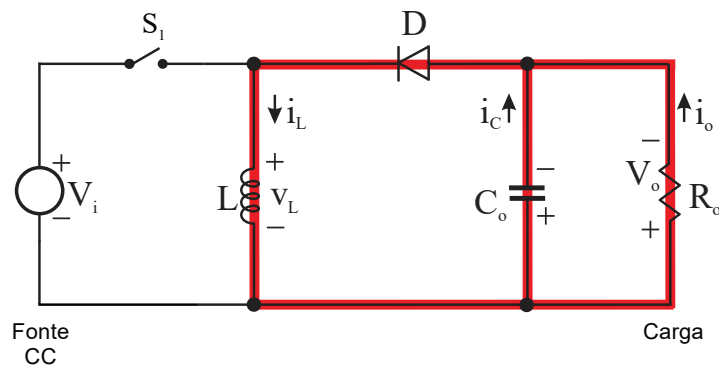


Figura 12 – Etapa 2, para o conversor *buck-boost*.

Fonte: autoria própria.

Para essa etapa o sistema de equações resultantes da aplicação de Kichhoff é dado em (9). Proveniente da análise do circuito mostrado na Figura 10.

$$\begin{cases} v_L(t) - v_o(t) = 0 \\ i_L(t) + i_C(t) + i_o(t) = 0 \end{cases} \quad (14)$$

Resolvendo o sistema de equação de (9) e usando as equações (1), obtêm-se:

$$v_L(t) + v_o(t) = 0$$

$$\boxed{L \frac{d i_L(t)}{dt} = -v_o(t)}$$

$$i_C(t) = i_L(t) - i_o(t)$$

$$\boxed{C \frac{d v_o(t)}{dt} = -i_L(t) - \frac{v_o(t)}{R_o}}$$

Escrevendo de forma matricial resulta em (10)

$$\dot{K}x = Ax + Bu$$



$$\underbrace{\begin{bmatrix} L & 0 \\ 0 & C \end{bmatrix}}_K \underbrace{\begin{bmatrix} \dot{i}_L \\ v_o \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & -\frac{1}{R_o} \end{bmatrix}}_{A_2} \underbrace{\begin{bmatrix} i_L \\ v_o \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{B_2} \underbrace{[V_i]}_u \quad (15)$$

Aplicando a mesma metodologia utilizada para o conversor *buck-boost* pode-se encontrar a função de transferência que relaciona a tensão de saída V_o com a razão cíclica, dada por

$$G_{(V_o, D)}(s) = \frac{V_i}{(D-1)^2} \frac{(R_o + D^2 R_o - 2DR_o - DLs)}{(R_o D^2 - 2R_o D + CLR_o s^2 + Ls + R_o)} \quad (16)$$

O App Designer do Matlab[®] foi utilizado para projeto do aplicativo. O ambiente de desenvolvimento do aplicativo pode ser visto nas Figuras 8 e 9. A linguagem de programação utilizada no aplicativo é uma mescla da linguagem de programação usada em *scripts* do Matlab[®] e a linguagem de programação orientada a objeto C++.

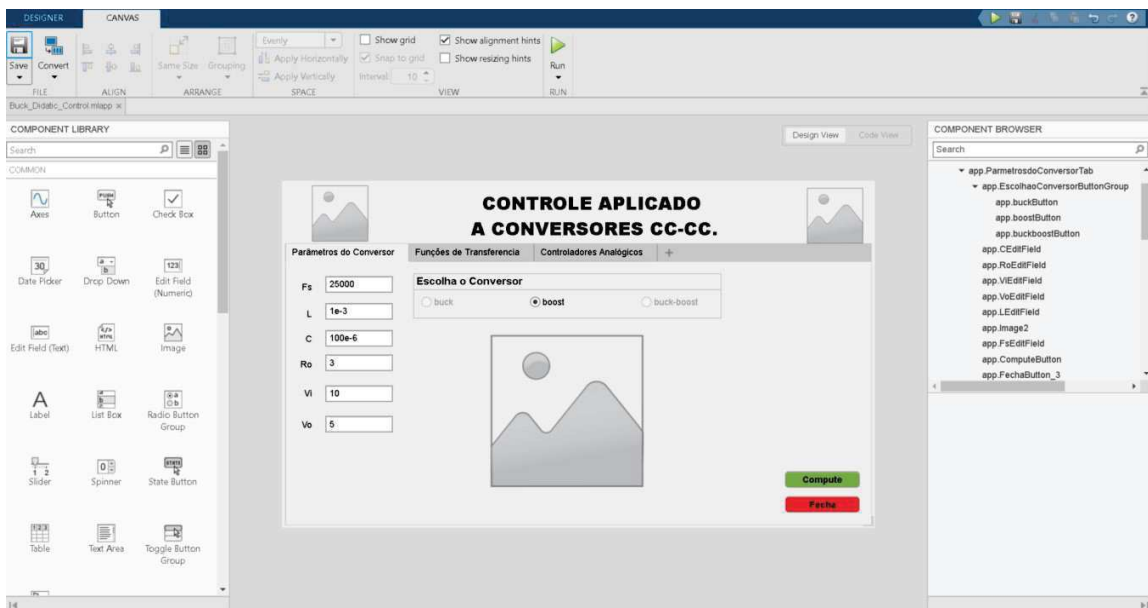


Figura 13– Ambiente de desenvolvimento do App Designer do Matlab.

Fonte: autoria própria.

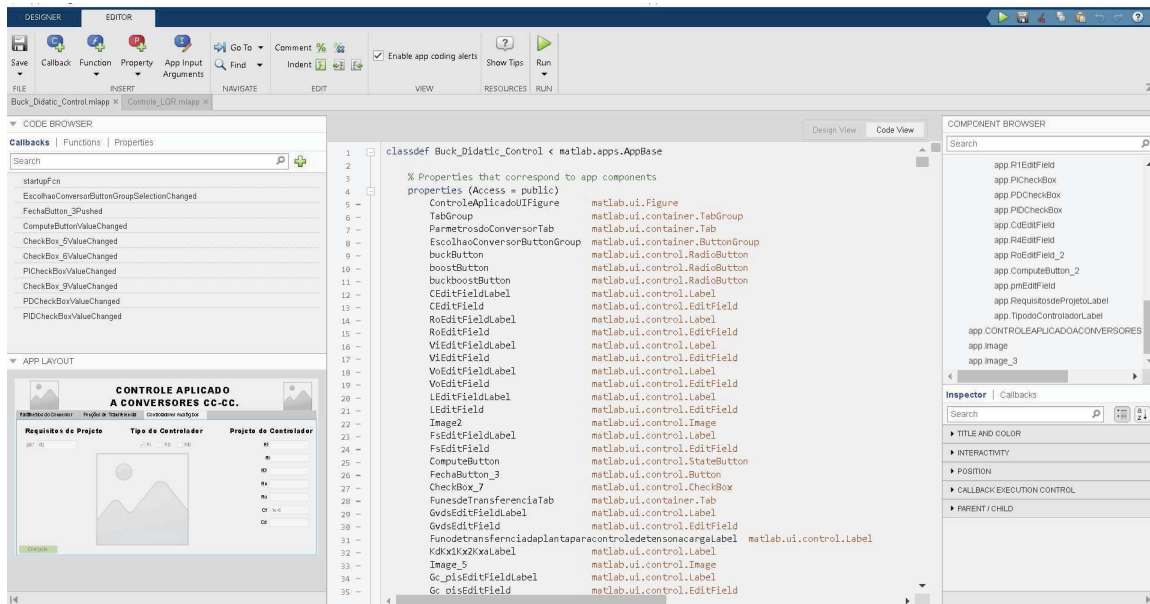


Figura 14 – Ambiente de desenvolvimento do App Designer do Matlab, com código.

Fonte: autoria própria.

4. RESULTADOS:

Na Figura 9 mostra-se a interface de abertura do aplicativo em desenvolvimento. Esse aplicativo permite calcular de forma rápida e simples as funções de transferência, tanto contínuas como discretas para os conversores clássicos, não isolados, *buck*, *boost* e *buck-boost*, o quais são conversores muito utilizados em eletrônica de potência e em sistema aeroespaciais.

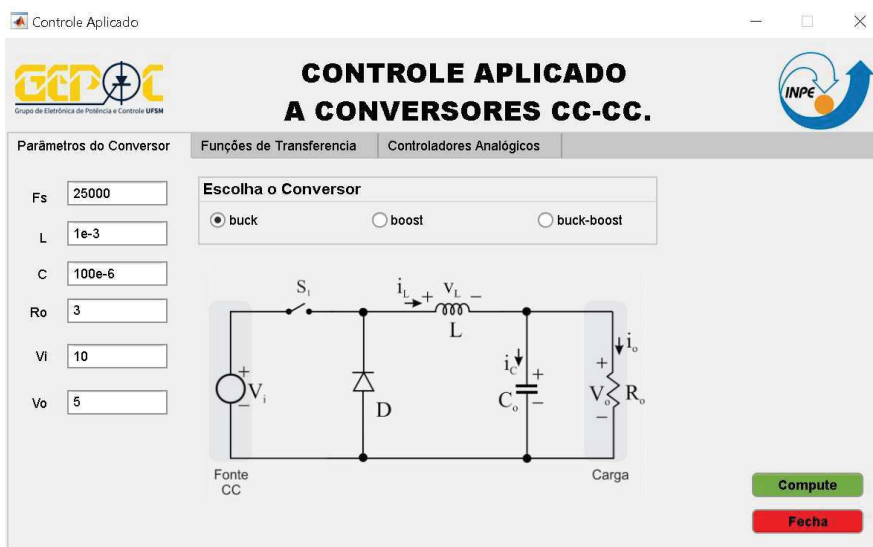


Figura 15 – Interface do aplicativo em linguagem Matlab®.

Fonte: autoria própria.



Ao se pressionar o botão “compute” o aplicativo efetua o cálculo do modelo do conversor selecionado e preenche a aba de “Funções de Transferência” mostrada na Figura 10.

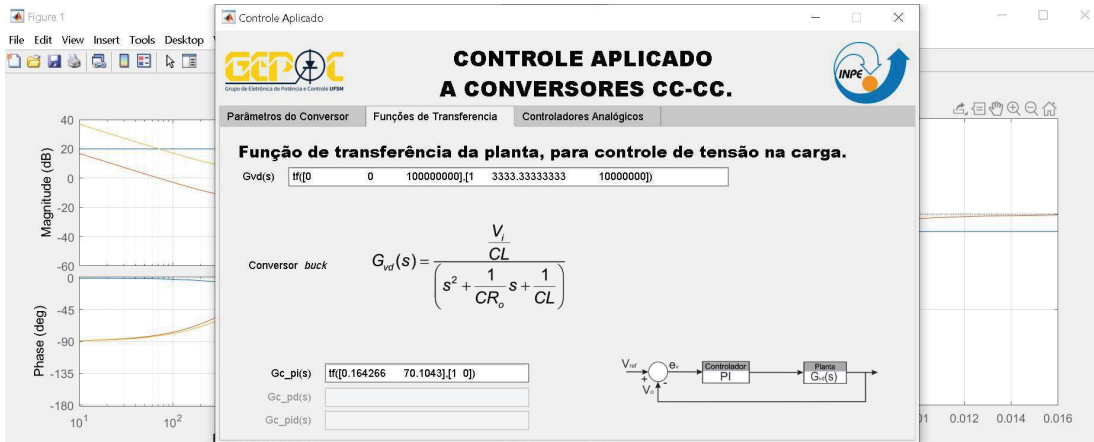


Figura 16 – Aba de solução “Funções de Transferência”.

Fonte: autoria própria.

Além disso, será plotado de forma automática a solução de projeto para o controlador selecionado de (PI, PD ou PID), conforme mostrado na Figura 11.

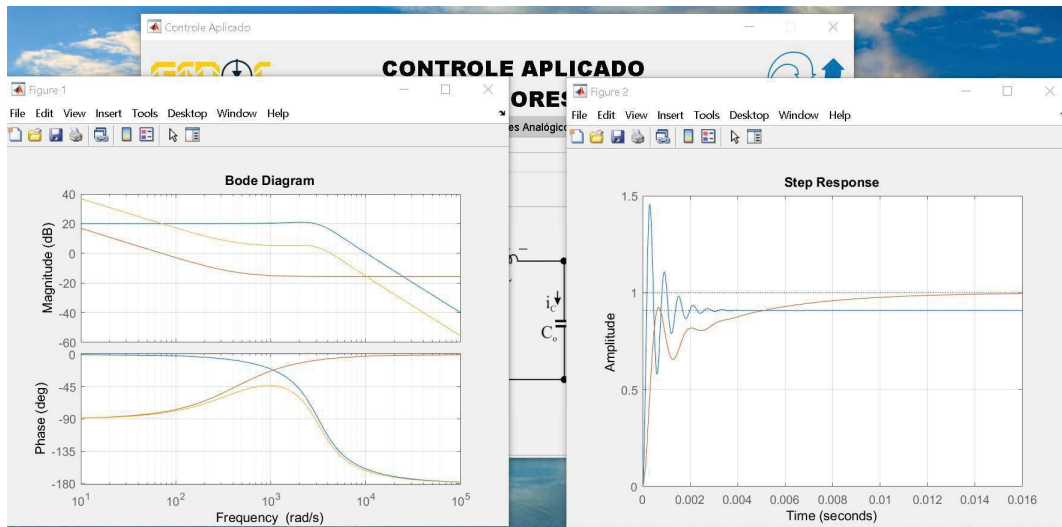


Figura 17 –Exemplo para o conversor *buck* , figuras de solução de projeto.

Fonte: autoria própria.

E ainda abrirá uma janela para permitir que sejam exportados os resultados para uso no software de simulação de controle utilizado em eletrônica de potência chamado de PSIM[®] da Powersim Inc., a janela de exportação dos dados pode ser visualizada na Figura 11.

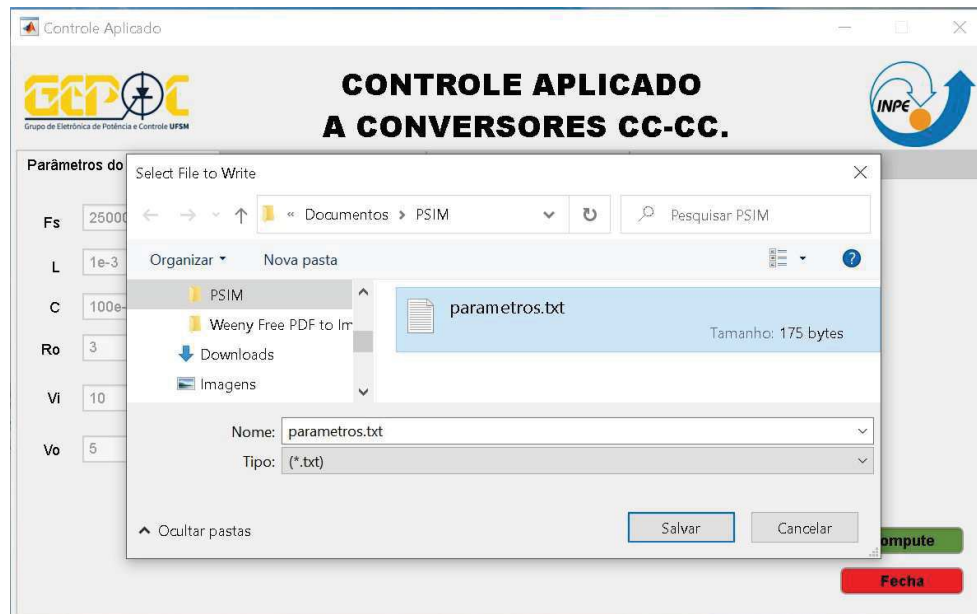


Figura 18– Janela automática que permite exportar os resultados para o PSIM[®].

Fonte: autoria própria.

A última aba do aplicativo, ainda não concluída, mostra o projeto analógico do controlador para o conversor escolhido como se mostra na Figura 12. Nessa aba pode-se escolher os valores dos componentes para o projeto de controlares analógico (PI,PD ou PID).

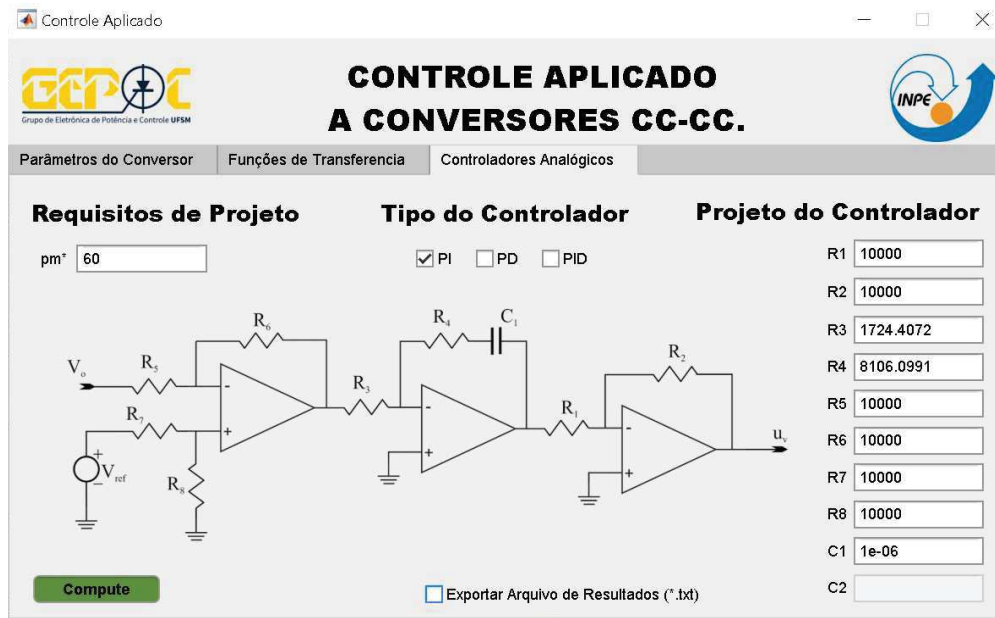


Figura 19– Aba que apresenta o projeto de controle analógico para o conversor selecionado.

Fonte: autoria própria.

Após a realização do projeto verifica-se o seu desempenho, para isso utiliza-se o software PSIM[®] que é mais bem aceito na área de estudo de eletrônica de potência. A Figura 12 mostra uma simulação com o projeto realizado para o conversor *buck*, tanto para o sistema

simulado por função de transferência como para o projeto com amplificadores operacionais (projeto analógico).

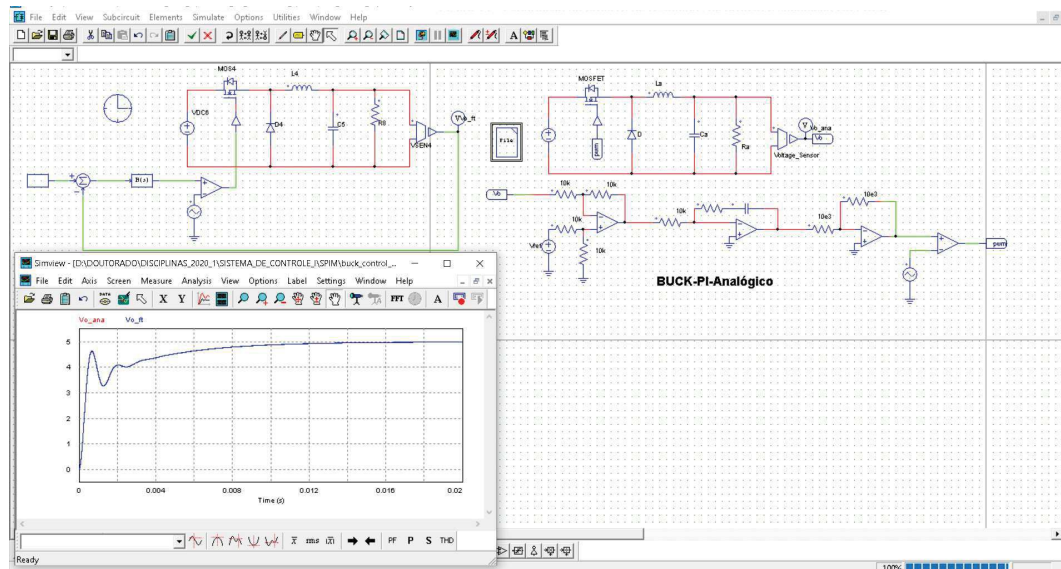


Figura 20– Exportação do arquivo de resultados de projeto e simulação em PSIM®.

Fonte: autoria própria.

5. Comprovação experimental do projeto

Dando continuidade ao trabalho de pesquisa iniciado em 2019, no qual foi desenvolvido um aplicativo em MATLAB® para projeto de controladores analógicos para conversores CC-CC. Nesta nova etapa da pesquisa, foi incluída a implementação prática do controle em um protótipo e a apresentação dos resultados de simulação e experimentais. O trabalho mostra o problema de regulação da tensão de saída de um conversor CC-CC, abrangendo: i) a modelagem do conversor em função de transferência, incluindo incerteza na carga e na tensão de entrada, ii) as especificações de projeto em termos de margem de fase e frequência de cruzamento, iii) o projeto dos ganhos do controlador utilizando a função pidtune do MATLAB®, iv) a implementação do controle com amplificadores operacionais, com a escolha dos resistores e capacitores associados a estes amplificadores, v) a apresentação dos resultados experimentais do sistema de controle em malha fechada aplicado a um conversor buck de potência nominal de .10 W e operando com frequência de comutação de 20 kHz. A escolha deste conversor deu-se por apresentar uma topologia simples e que pode servir de base para desenvolvimento de controladores para conversores mais complexos, embarcados em subsistemas elétricos de potência de pequenos satélites. O conversor buck foi modelado matematicamente, utilizando modelo médio em espaço de estados, para se obter uma função de transferência. A escolha da função de transferência da planta, conversor buck, usada para o projeto do controlador PID, levou em conta as variações paramétricas da tensão de entrada e da carga. Foi considerada adequada para o projeto do controlador a planta com maior variação de fase. Após a escolha da planta, foi executada a função pidtune do MATLAB®. Para sua execução, foram usados a margem de



fase do sistema compensado maior ou igual a 60° e a frequência de cruzamento por zero dB, do sistema compensado, uma década abaixo da frequência de comutação do conversor buck. Executou-se o aplicativo e foram obtidos os valores dos componentes adequados para a implementação de um PID analógico. O conversor buck foi implementado juntamente controlador PID, utilizando amplificadores operacionais, resistores e capacitores. Os circuitos foram testados, e os sinais de saída foram comparados com resultados de simulação, comprovando-se a viabilidade técnica do projeto pelo aplicativo proposto.

A Tabela 01, mostra os valores dos componentes utilizados para implementação prática do conversor buck.

Tabela 01 - Definição dos componentes do conversor buck

Componente	Valor
Inductor (L)	216 uH
Capacitor (C)	47 uF
Tensão de saída (V_o)	6 V
Tensão de entrada (V_i)	30 V
Frequencia de chaveamento (fsw)	20 kHz

Fonte: Autor

O modelo foi validado conforme mostra a Figura 21

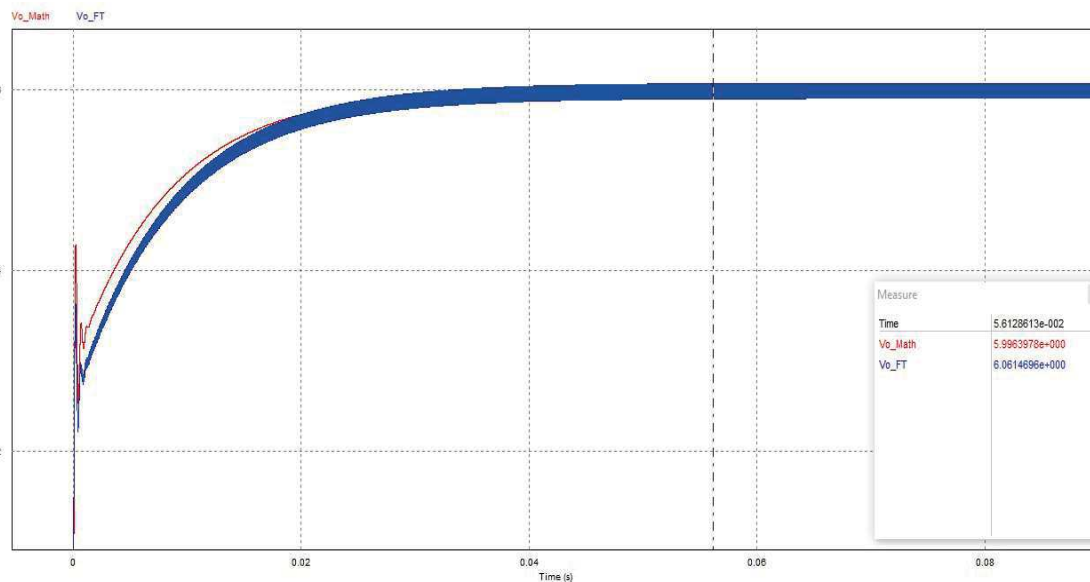


Figura 21– Resposta do controlador implementado em função de transferência e analógico, simulação no software PSIM[®].

Fonte: autoria própria.



O projeto do controlador conforme descrito anteriormente resultou nos ganhos dados por

$$C_{pi}(s) = \frac{0,03162s + 7,779}{s} \quad (17)$$

Os valores do controlador analógico estão dados na Tabela 02.

Tabela 02 – Valores dos componentes do controlador analógico

Componente	Valor
C ₂	1 uH
R ₃ =R ₄	10 kΩ
R ₂	4,065 kΩ
R ₁	128,551 kΩ

Fonte: Autor

Comparando o resultado de simulação do controlador analógico como o controlador em função de transferência, para isso, foi realizada uma variação de +50% de carga está indicada na variação da tensão de saída, verifica-se que o controlador com implementação analógica responde com uma dinâmica muito próxima a do controlador por função de transferência, conforme mostrado na Figura 22.

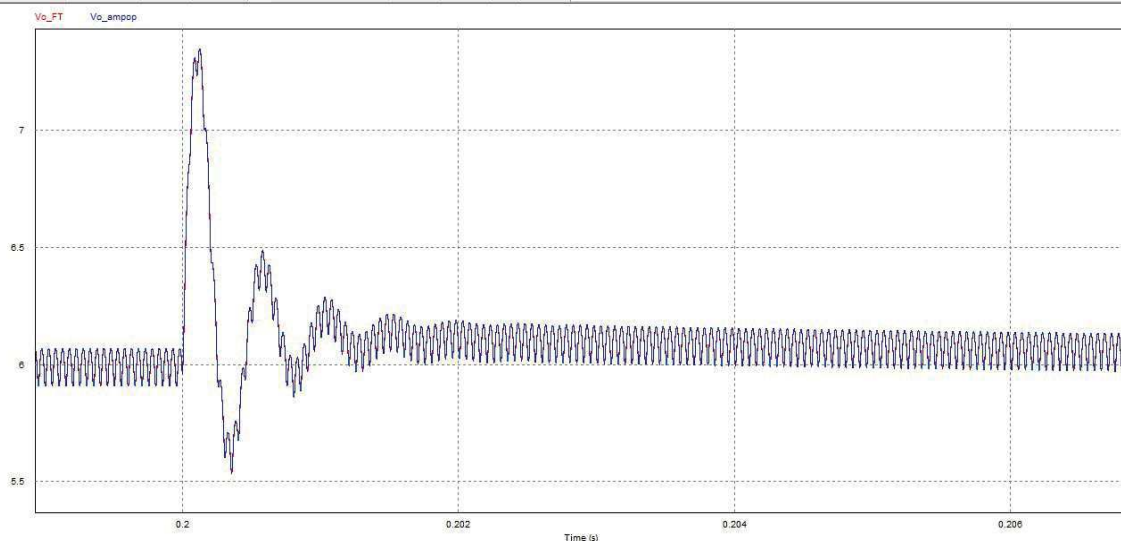


Figura 22– Resposta dinâmica dos controladores em função de transferência e analógico no PSIM.

Fonte: autoria própria.

O conversor foi implementado em placa padrão, incluindo o controlador analógico projetado. A Figura 23 mostra o protótipo e o controle analógico.

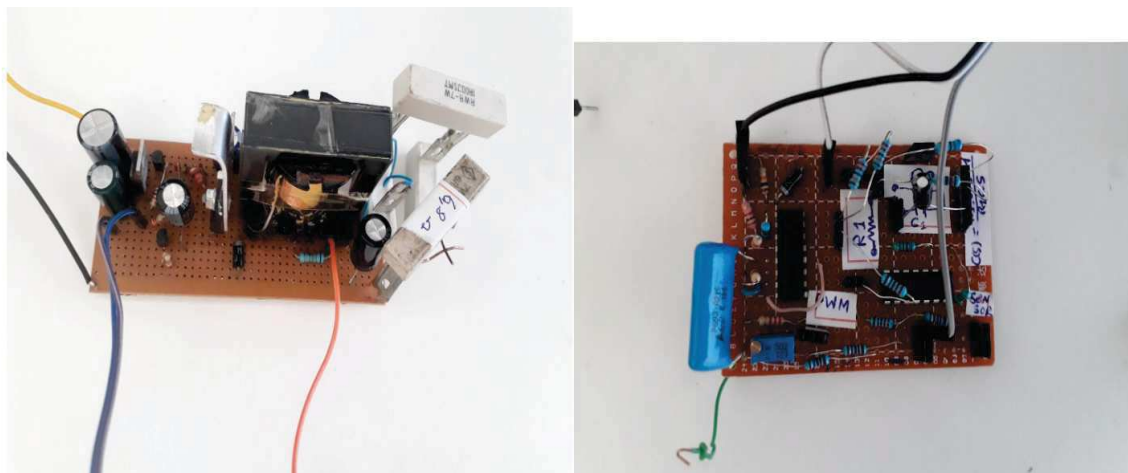


Figura 23– Conversor e controle implementado.

Fonte: autoria própria.

Aplicando-se uma variação de carga de -50% é possível verificar que a resposta transitória

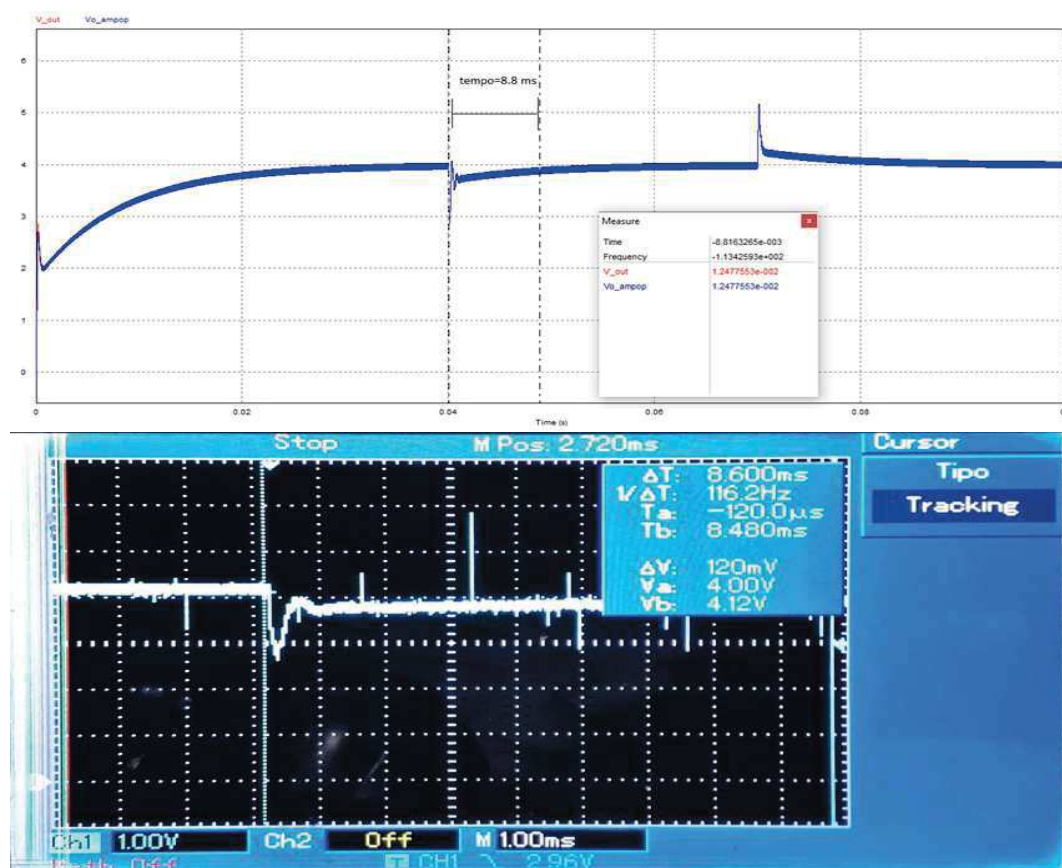


Figura 24– Variação de -50% de carga

Fonte: autoria própria.



Tanto da resposta prática quando da resposta de simulação são similares e próximas de 9ms. Novamente, realizando variação de carga no sentido contrário de +50% confirma-se a boa correspondência das respostas.

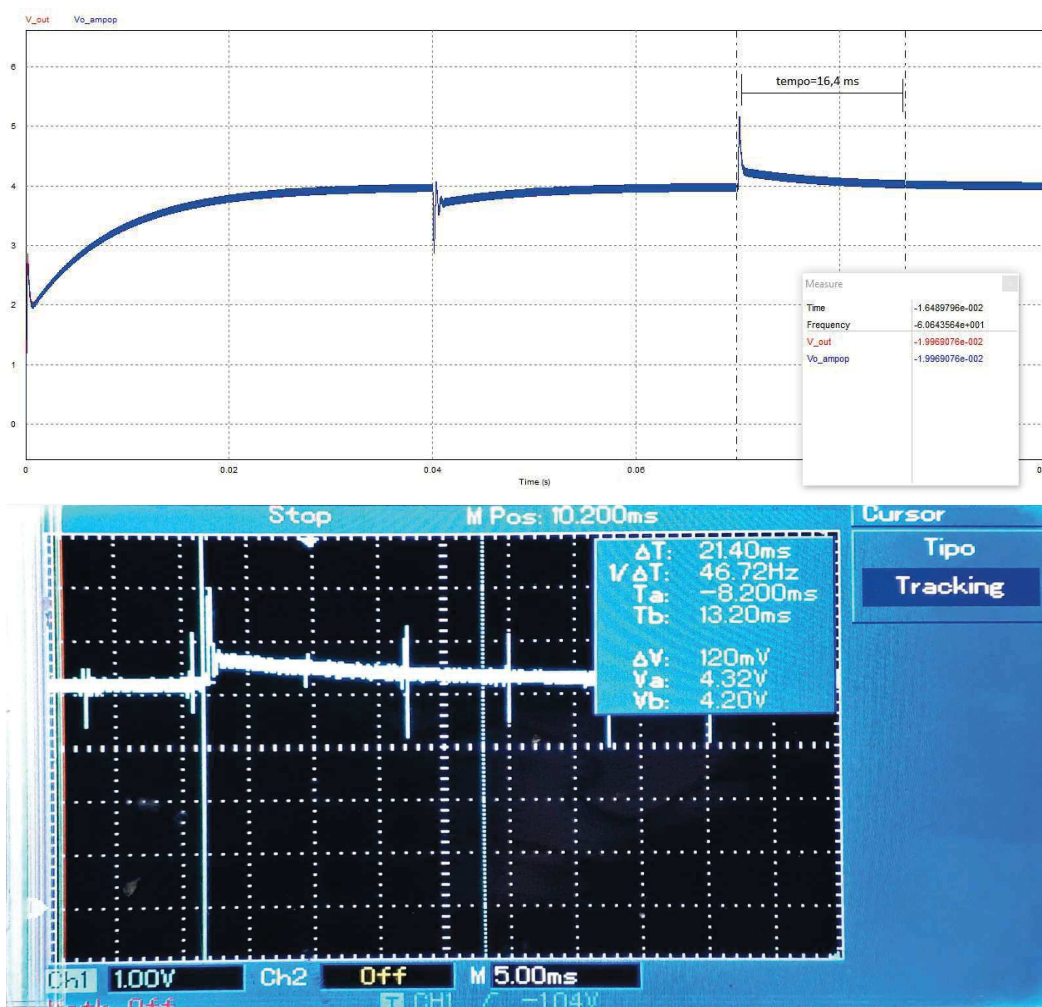


Figura 25– Variação de +50% de carga.

Fonte: autoria própria.



6. CONCLUSÃO

Ao longo desse projeto foram revisados os conceitos sobre CubeSats e seus subsistemas elétricos de potência (EPS), desenvolvido um software em linguagem Matlab com a função de avaliar o controle e a estabilidade de conversores CC-CC clássicos (*buck*, *boost* e *buck-boost*) que são utilizados comumente em subsistemas elétricos de potência (EPS). As principais dificuldades sanadas no decorrer dos trabalhos realizados foram sanadas com busca em artigo, livros e com o auxílio do orientador. Uma publicação em formato de resumo deve ser submetida a Jornada Acadêmica Integrada da UFSM, assim que as inscrições forem abertas, como parte dos requisitos de renovação de bolsa.



7. APÊNDICE

A seguir está o Código Parcial das principais rotinas:

Modelagem do conversor *buck* (já adequada para incluir as não idealidades dos componentes):

```
function [Gi,Gv,Gvi,Ap,Bp,Cp,Ep,Y,X] = buck(app)
    clc;
    format long G

    Vo=str2num(app.VoEditField.Value);
    Vi=str2num(app.ViEditField.Value);
    D=Vo/Vi;           % Ponto de operação

    L=str2num(app.LEditField.Value);
    Co=str2num(app.CEditField.Value);
    Vd=0;
    Rd=0;
    RL=0;
    Rc=0;
    Rs=0;
    Ro=str2num(app.RoEditField.Value);

    % Modelo médio em espaço de estados
    % Etapa 1
    %diL/dt=
        a11=(0 - 1*(RL + Rs + (Rc*Ro)/(Rc + Ro)) + 0*(Rc/(Rc
+ Ro) - 1))/L;
        a12=(0 - 0*(RL + Rs + (Rc*Ro)/(Rc + Ro)) + 1*(Rc/(Rc
+ Ro) - 1))/L;
    %dvC/dt=
        a21=-(0- Ro*1)/(Co*(Rc + Ro));
        a22=-(1 - Ro*0)/(Co*(Rc + Ro));

    A1=[a11    a12
        a21    a22];

    %diL/dt=
    b11=(1 - 0*(RL + Rs + (Rc*Ro)/(Rc + Ro)) + 0*(Rc/(Rc + Ro)
- 1))/L;
    b12=-(0 - Ro*0)/(Co*(Rc + Ro));
    %dvC/dt=
    b21=0;
    b22=0;

    B1=[b11    b12
        b21    b22];
```



```
% saída
% y=iL
c11=1;
c12=0;
%y=Vo;
c21=Rc*Ro/(Rc+Ro);
c22=Ro/(Rc+Ro);

C1=[c11  c12
     c21  c22];

% y=iL
e11=0;
e12=0;
%y=Vo;
e21=0;
e22=0;

E1=[e11  e12
     e21  e22];

clear a11 a12 a21 a22 b11 b12 b21 b22 c11 c12 c21 c22 e11 e12 e21 e22;

% Etapa 2
%diL/dt=
a11=-(0 + 1*(RL + Rd + (Rc*Ro)/(Rc + Ro)) -
0*(Rc/(Rc + Ro) - 1))/L;
a12=-(0 + 0*(RL + Rd + (Rc*Ro)/(Rc + Ro)) -
1*(Rc/(Rc + Ro) - 1))/L;
%dvC/dt=
a21=-(0 - Ro*1)/(Co*(Rc + Ro));
a22=-(1 - Ro*0)/(Co*(Rc + Ro));

A2=[a11  a12
     a21  a22];

%diL/dt=
b11=0;
b12=-1/L;
%dvC/dt=
b21=0;
b22=0;

B2=[b11  b12
     b21  b22];

% y=iL
c11=1;
c12=0;
e11=0;
```



```
e12=0;
```

```
%y=Vo;  
c21=(Ro*Rc)/(Rc + Ro);  
c22=Ro/(Rc + Ro);  
e21=0;  
e22=0;
```

```
C2=[c11  c12  
     c21  c22];  
E2=[e11  e12  
     e21  e22];
```

```
%Matrizes médias  
A=D*A1+(1-D)*A2;  
B=D*B1+(1-D)*B2;  
C=D*C1+(1-D)*C2;  
E=D*E1+(1-D)*E2;
```

```
%Ponto de operação em regime permanente  
U = [Vi  
      Vd];  
% X = simplify(-inv(A)*B*U);  
% %Y = ((-C*inv(A)*B+E)*U);  
% Y = simplify(C*X+E*U);
```

```
X = (-inv(A)*B*U);  
%Y = ((-C*inv(A)*B+E)*U);  
Y = (C*X+E*U);
```

```
%Definição do modelo padrão (pequenos sinais)  
Ap=A;  
% Bp=[B (A1-A2)*X+(B1-B2)*U];  
Bp=(A1-A2)*X+(B1-B2)*U;  
Cp=C;  
% Ep=[E (C1-C2)*X+(E1-E2)*U];  
Ep=(C1-C2)*X+(E1-E2)*U;
```

```
%% FT  
[n,d]=ss2tf(Ap,Bp,Cp,Ep,1);  
Gi=tf(n(1,:),d);  
Gv=tf(n(2,:),d);  
Gvi=minreal(Gv/Gi);
```

```
end
```



Sintonia automática do controlador PI

```
function [R1,Rf,Cf,Rp] = PI_Controller(app)
    app.PICheckBox.Value=0;
    s=tf('s');
    [~,Gv,~,~,~,~,~,~] = buck(app);
    [~,~,~,Wm]=margin(Gv)
        %% PI usando tecnicas de atraso de fase

    % requisitos
    ep=0;
    mp=str2num(app.pmEditField.Value)-180;

    if (mp>0)|| (mp<-180)
        mydlg = warndlg('Derfina um valor estável de margem de
fase (ex.: 60 Graus)', 'ATENÇÃO');
        waitfor(mydlg);
    end

    % busca margem de fase desejada
    w=logspace(-1,6,1e6);
    [Mag,Phase,Wrad]=bode(Gv,w);
    mag=squeeze(Mag);
    phase=squeeze(Phase);
    M=20*log10(mag);
    k=find(phase<=mp);
    Wm=Wrad(k(1));           % Encontrou Wm

    Mg=M(k(1))
    Ph=phase(k(1))

    % garantir nova Wm
    Kc=10^(-Mg/20);

    % Melhorar estabilidade
    z=Wm/100;

    % Projeto completo
    Gpi=Kc*(s+z)/s;

    Vo=str2num(app.VoEditField.Value);
    Vi=str2num(app.ViEditField.Value);
    D=Vo/Vi;           % Ponto de operação

    L=str2num(app.LEditField.Value);
    Co=str2num(app.CEditField.Value);
    Ro=str2num(app.RoEditField.Value);
    %varNames = {'R1','Rf','Rp','Cf'};
```



```
T2 =  
table({'Vi1='; 'L1='; 'C1='; 'R1='; 'Fsw1='; 'Vref1='; 'Ri1='; 'Rf1='; 'Rp1='; 'Cf1='  
'}, [Vi;L;Co;Ro;Fsw;Vo;R1;Rf;Rp;Cf], 'VariableNames', { '%Arquivo', 'de dados  
PI'});  
  
[f,p,ix] = uinputfile('*.txt');  
if (f==0)  
    return  
end  
cd (p);  
outputfile=strcat(p,f);  
writetable(T2,outputfile,'Delimiter',' ');  
app.PICheckBox.Value=0;  
  
end
```



8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Robinson, P. A., Spacecraft Environmental Anomalies Handbook, GL-TR-89-0222, Hanscom Air Force Base, MA: Air Force Geophysics Laboratory, 1989.
- [2] Bedingfield, K. L., Leach, R. D. and Alexander, M. B., “Spacecraft System Failures and Anomalies Attributed to the Natural Space Environment,” NASA Reference Publication 1390.
- [3] R. W. Erickson, Fundamentals of Power Electronics. Chapman & Hall, New York, USA, 1997.
- [4] D. C. Martins & I. Barbi, Eletrônica de Potência: Introdução ao Estudo dos Conversores CC-CA. Edição dos Autores, 2ª Edição, Florianópolis, SC, Brasil, 2008.
- [5] M. L. Heldwein, EMC Filtering of Three-Phase PWM Converters. Published by Südwestdeutscher Verlag für Hochschulschriften & Co. KG, Germany, 2009.
- [6] S. Chowdhury, S. P. Chowdhury and P. Crossley, Microgrids and Active Distribution Networks. Published by The Institution of Engineering and Technology, London, United Kingdom, 2009.
- [7] M. H. Rashid, Power Electronics Handbook. Academic Press, San Diego, California, USA, 2001.
- [8] D. C. Martins & I. Barbi, Eletrônica de Potência: Conversores CC-CC Básicos Não Isolados. Edição dos Autores, 3ª Edição, Florianópolis, SC, Brasil, 2008.
- [9] A. S. Kislovski, R. Redl and N. O. Sokal, Dynamic Analysis of Switching-Mode DC/DC Converters. VanNostrand Reinhold, New York, USA, 1991.