

Implementação de uma carga fictícia utilizando *Processor-in-the-Loop*

José H. M. Luna
DIMEL/SEGEL – Seção de Grandezas Elétricas
INMETRO
Rio de Janeiro, Brasil

Luís G. B. Rolim, Robson F. S. Dias
COPPE/UFRJ – Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil

Resumo — O uso eficiente da energia elétrica tem ganhado notoriedade. Neste contexto, é indispensável a manutenção da exatidão dos instrumentos de medição em uso, assegurada através de ensaios. Objetivando agilidade, o uso da carga fictícia é a forma preferencial de energizar o medidor, pela sua versatilidade e peso reduzido, no entanto pouco utilizada em virtude do custo. Visando o desenvolvimento de uma carga fictícia de baixo custo, este trabalho apresenta o resultado da simulação de um protótipo para ensaios de exatidão em medidores de energia elétrica monofásicos. O PSIM foi utilizado para simular o circuito com o controle dos conversores sendo realizado no ambiente de *Processor-in-the-Loop* (PIL) em um microcontrolador C2000™ Delfino™ da Texas Instruments.

Palavras chave — Controle preditivo, *processor-in-the-loop*, Inversor, medição de energia elétrica, Simulação de conversores de potência.

I. INTRODUÇÃO

O mercado de energia elétrica brasileiro, de acordo com [1], conta em 2015 com 77 milhões de unidades consumidoras, onde 85% deste total é de consumidores residenciais, cada qual com sua medição individualizada. Sendo o comércio de energia elétrica precificado através do resultado desta medição individualizada, é necessário garantir sua exatidão, proporcionando assim um comércio justo e uma relação de confiança entre as partes.

A forma de assegurar que a exatidão do medidor atende às especificações é através de uma verificação, onde sua medição é comparada com outro instrumento de medição com exatidão superior e rastreabilidade estabelecida, e para tanto, faz-se necessário que uma quantidade de energia atravesse o medidor [2].

Os meios usuais de excitar o medidor de modo a checar sua exatidão são solicitar à unidade consumidora que utilize seus aparelhos elétricos ou conectar à saída do medidor uma carga resistiva variável. Ambos os métodos trazem transtornos à operação, a primeira por depender da presença do usuário e a segunda por depender geralmente de equipamento pouco ergonômico em virtude de seu peso considerável em razão das potências envolvidas. Para ambos, pode-se citar ainda o desperdício de energia elétrica. De forma alternativa, pode-se utilizar um equipamento conhecido como carga fictícia ou carga fantasma, que é capaz de gerar tensão e/ou corrente ajustáveis, de modo a excitar o medidor com potência desejada.

Neste trabalho, a planta da carga fictícia será modelada no PSIM com o algoritmo de controle dos conversores de potência sendo realizado por um microcontrolador (MCU) C2000™ Delfino™ da Texas Instruments. A comunicação entre ambos, MCU e PSIM, é realizada via porta serial. Nas próximas seções será apresentada a descrição da estrutura do ambiente de desenvolvimento utilizando *Processor-in-the-Loop* (PIL), a descrição da planta projetada para modelar a carga fictícia e, por fim, os resultados da simulação.

II. DESCRIÇÃO DA CARGA FICTÍCIA

Além dos procedimentos de distribuição estabelecidos pela agência reguladora do setor, a ANEEL [3], os medidores de energia elétrica (MEE) em uso devem atender às exigências definidas pelo Inmetro em [4], isso considerando apenas MEE eletrônicos em razão da tendência nacional de substituir o atual parque de MEE eletromecânicos. Dentre os ensaios realizados para observar sua adequação ao regulamento, focar-se-á no ensaio de exatidão, cuja finalidade é averiguar se os erros apresentados pelo medidor estão dentro dos limites estabelecidos em [4] e replicados nas Tabelas 1 e 2.

TABELA I
LIMITES DE ERRO PERCENTUAIS PARA MEDIÇÃO ATIVA

Limites de erro percentuais para medidores com índice de classe			
D	C	B	A
± 0,4	± 1,2	± 2,2	± 3,2

TABELA II
LIMITES DE ERRO PERCENTUAIS PARA MEDIÇÃO REATIVA

Limites de erro percentuais para medidores com índice de classe			
D	C	B	A
± 0,8	± 2,4	± 4,4	± 6,4

Assim, monta-se o esquema ilustrado na Fig. 1 para um MEE monofásico, onde a carga fictícia é representada pelas fontes de tensão e de corrente delimitadas pelo traçado vermelho.

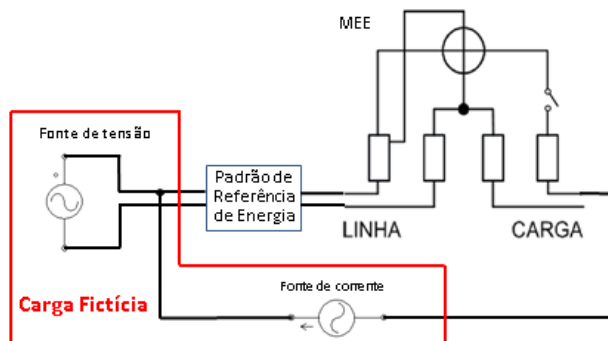


Fig. 1. Diagrama de ligação do MEE e carga fictícia

A caixa identificada como Padrão de Referência de Energia na Fig. 1 é outro MEE, no entanto com exatidão melhor do que o MEE sob ensaio.

Conforme estabelecido em [5], o procedimento para ensaiar um MEE consiste em comparar a leitura do MEE com o padrão de referência, levantando assim o seu erro de leitura.

Para realizar o ensaio de exatidão, é necessário que uma tensão seja aplicada e uma corrente circule em ambos os instrumentos, sendo este o papel da carga fictícia, conforme já exposto na introdução. De modo geral, circuito se baseia em dois conversores, um para estabelecer tensão e outro para corrente, conectados conforme diagrama apresentado na Fig. 1. O diferencial deste procedimento reside no fato de se tratar de dois conversores isolados, onde a fonte de tensão possui uma baixa corrente de saída enquanto que a fonte de corrente trabalha com baixa tensão, resultando em uma baixa potência demandada pelo equipamento, reduzindo assim dissipação de calor e o espaço necessário para acomodar o circuito.

Uma realização possível para a carga fictícia está esquematizada na Fig. 2 e será implementada neste trabalho utilizando a proposta de efetuar o controle dos conversores VSI e Buck utilizando o MCU através da técnica de *Processor-in-the-Loop* (PIL) [6].

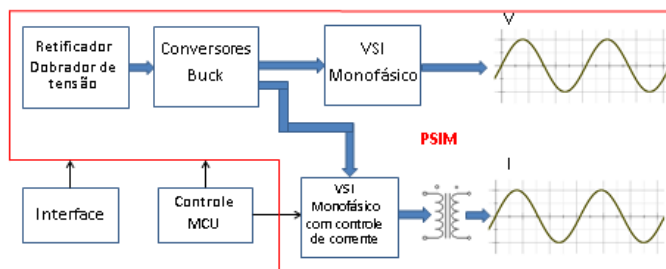


Fig. 2. Esquema da carga fictícia

Pode-se dividir a planta apresentada na Fig. 2 em três blocos:

- 1) Bloco ac/dc composto pelo retificador dobrador e por dois conversores buck assíncronos.
- 2) Bloco dc/ac composto pelos dois inversores.
- 3) Bloco computacional composto por uma interface gráfica e pelo controle efetuado pelo MCU.

Os blocos 1 e 2 são simulados no PSIM. Os componentes do bloco 3 são processos externos que interagem com a simulação através de comunicação serial conforme será visto na próxima seção. Uma breve descrição dos blocos 1 e 2 será dada nesta seção, já o bloco 3 será tratado na Seção III.

A. Retificador dobrador

Este é o primeiro circuito da carga fictícia, responsável por fornecer uma tensão dc retificada com duas vezes o valor da tensão de pico ac da entrada. Este retificador é necessário por se tratar de uma carga fictícia monofásica, onde se supõe ter disponível apenas tensão de fase. A Fig. 3 apresenta o esquema do retificador dobrador.

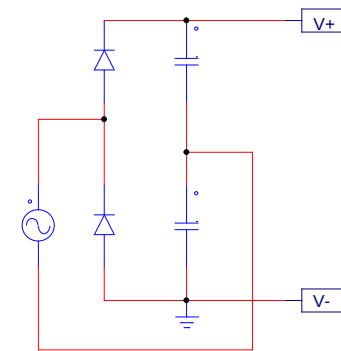


Fig. 3. Retificador dobrador

B. Conversores Buck

A tensão retificada é disponibilizada na entrada de dois conversores buck assíncronos, cada qual responsável por fornecer o nível dc necessário para o inversor correspondente. A Fig. 4 mostra o conversor buck referente ao inversor da fonte de tensão enquanto que a Fig. 5 mostra o conversor buck referente ao inversor da fonte de corrente.

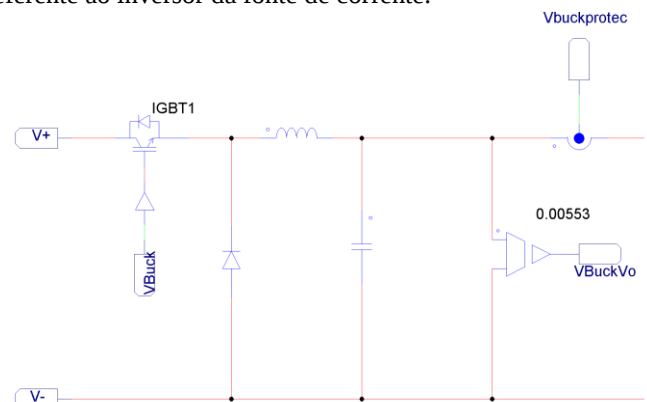


Fig. 4. Conversor Buck para fonte de tensão

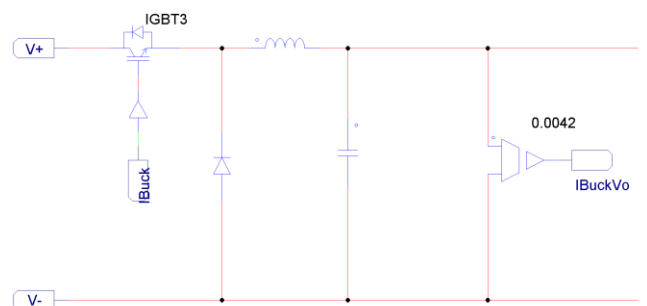


Fig. 5. Conversor Buck para fonte de corrente

A estrutura dos dois conversores buck é semelhante, diferindo apenas na constante de ganho do sensor utilizado para realimentar a tensão de saída para o controle, já que níveis dc distintos são necessários para os valores nominais de tensão e corrente de saída admitidos no projeto da carga fictícia: 127 V e 15 A.

O controle de ambos os conversores é executado por compensadores PI discretizados pelo método trapezoidal que rodam no MCU. As medições VBuckVo e IBuckVo na saída dos conversores referentes aos inversores de tensão e corrente, respectivamente, são transmitidas ao MCU, que retorna com os sinais de controle das chaves correspondentes, VBuck e

IBuck, modulando um PWM com portadora triangular de 20 kHz. Um fragmento do controle PI discretizado com controle *anti-windup* do conversor *buck* da fonte de tensão está descrito na Fig. 6, possuindo a mesma estrutura para o conversor *buck* da fonte de corrente.

```

v_erro_ant=v_erro;
v_erro = Vbuckref - Vbuckmedido;
v_mp = kp * v_erro;
v_mi = v_mi + kiTs*(v_erro+v_erro_ant);
vm_presat = v_mp + v_mi;
if (vm_presat > buckMax)
    VBuck = buckMax;
else if (vm_presat < buckMin)
    VBuck = buckMin;
else
    VBuck = vm_presat;

```

Fig. 6. Fragmento do algoritmo do controle PI dos conversores *buck*

C. Inversores

A topologia de inversor utilizada neste trabalho foi a ponte H completa com MOSFETs acionados por um PWM com esquema de modulação unipolar e portadora triangular de 50 kHz. Para o inversor da fonte de tensão, uma tabela de seno foi emulada no PSIM por uma onda dente de serra passando por um bloco *Zero-Order-Hold* em 15,36 kHz, que é equivalente a uma tabela de seno com 256 valores. Um filtro LC sintonizado em 840 Hz foi colocado na saída do conversor, enquanto que uma resistência alta modela a carga inserida pelo circuito de medição de tensão do MEE. A Fig. 7 exibe o modelo do inversor de tensão enquanto a Fig. 8 exibe o esquema de modulação em conjunto com o sistema de proteção.

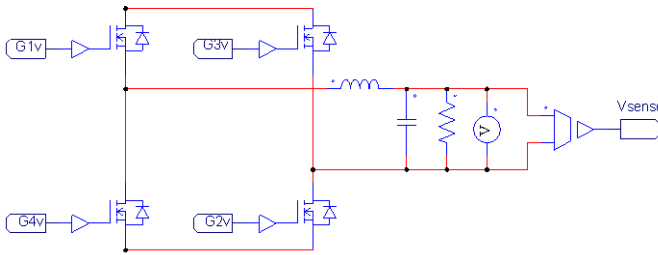


Fig. 7. Inversor da fonte de tensão

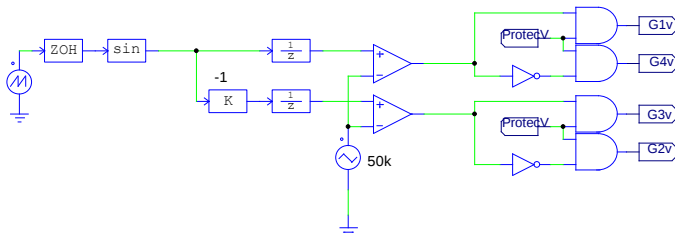


Fig. 8. Esquema de modulação do inversor da fonte de tensão

O controle da amplitude do inversor da fonte de tensão foi realizado em malha aberta, com índice fixo de modulação PWM e amplitude dc de sua entrada previamente ajustada para obter uma senoide com 127 V rms na saída da carga fictícia. Tal estratégia foi adotada levando em consideração que os sensores de tensão do MEE são dispositivos de alta impedância que causam pouca interferência no gerador.

O inversor da fonte de corrente tem em sua saída um transformador abaixador de tensão, uma indutância atuando como filtro no secundário do trafo e uma resistência baixa

modelando o *shunt* tipicamente utilizado como sensor de corrente em MEE monofásicos. A Fig. 9 apresenta o esquema deste inversor. A Fig. 10 exibe o esquema de modulação do inversor de corrente, semelhante ao já apresentado na Fig. 8.

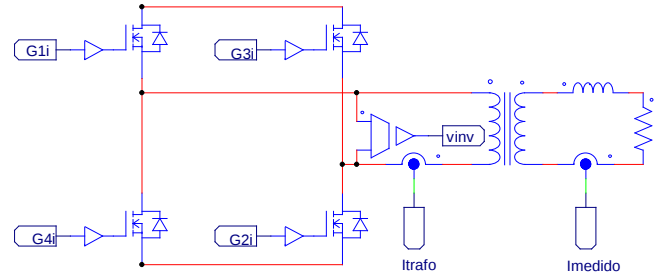


Fig. 9. Inversor da fonte de corrente

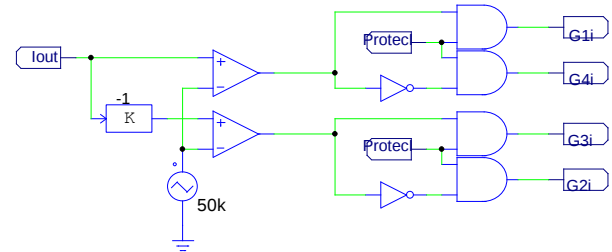


Fig. 10. Esquema de modulação do inversor da fonte de corrente

O controle do inversor de corrente foi realizado através do controle preditivo descrito em [7], que consegue rastrear o sinal de referência com precisão e baixa distorção. A estratégia por trás do controle preditivo é simples e pode ser resumida da seguinte forma: para toda iteração do algoritmo de controle, se busca o valor médio da tensão de saída do inversor que pode fazer o valor médio da corrente no indutor atingir sua referência até ao final do período de modulação seguinte àquele em que todos os cálculos são realizados [8].

A equação diferencial que modela o inversor e sua carga é discretizada e transformada numa equação à diferenças. De [7] tem-se o algoritmo que calcula durante o intervalo $\Delta T[k-1]$, o valor médio da tensão necessário para, no intervalo $\Delta T[k]$, corrigir a média dos dois valores anteriores de erro da corrente. Este algoritmo está apresentado em (1) e é implementado diretamente no MCU, onde os valores de tensão e corrente medidos são obtidos nos respectivos sensores nomeados como **vinv** e **Imedido** na Fig. 9.

$$v_l^{av}[k] = -\frac{1}{2}v_l^{av}[k-1] + v_l^{av}[k-2] + \frac{1}{2}v_l^{av}[k-3] + \frac{L_e}{\Delta T} \left\{ \frac{3}{2}i_e[k-1] - \frac{1}{2}i_e[k-2] - \frac{1}{2}i_e[k-3] \right\} \quad (1)$$

onde

- $v_l^{av}[k-n]$ Medição da tensão média do inversor no intervalo $\Delta T[k-n]$, $n = 0, 1, 2, 3$;
- $i_e[k-n]$ Erro da corrente no instante de amostragem $[k]$
- L_e Indutância equivalente vista pelo inversor;
- ΔT Intervalo de tempo entre duas amostragens.

O sinal de referência utilizado para o inversor de corrente é uma amostra do sinal de tensão, não sendo deste modo,

possível ajustar a fase da corrente, apenas a amplitude multiplicando este sinal de referência por uma constante. A medição do sinal de tensão que serve como referência vem da medição **Vsense**, que pode ser observada na Fig. 7. Ainda de acordo com [7], este sinal de referência é substituído por uma sua estimativa em avanço e é realizada com o propósito de reduzir o atraso de fase da corrente gerada. Um fragmento do algoritmo de controle implementado no MCU está descrito na Fig. 11.

```
//Estimativa da referência
iref_k2 = iref_k1;
iref_k1 = Iref;
iref = 3.5*iref_k1-3*iref_k2;

// Algoritmo de controle
ie_now = iref-Imedido; // calcula erro da corrente
vi_k3 = vi_k2; // Atribui os valores anteriores da tensão
//medida na saída do conversor
vi_k2 = vi_k1;
vi_k1 = Vin;
ie_k3 = ie_k2; // Atribui os valores anteriores de erro de corrente.
ie_k2 = ie_k1;
ie_k1 = ie_now;
vi_now = -0.5*vi_k1+vi_k2+0.5*vi_k3+LeDt*(1.5*ie_k1-0.5*ie_k2-0.5*ie_k3);
```

Fig. 11. Fragmento do algoritmo do controle preditivo da corrente

Também foi previsto proteção para os conversores contra sobre corrente, supondo falhas externas ao equipamento, como um curto na fonte de tensão ou a operação da fonte de corrente em aberto. Para o inversor de tensão, a proteção é comandada pela leitura da corrente na saída do conversor *buck* através do sensor **Vbuckprotec** conforme pode ser visto na Fig. 4. A Fig. 12 apresenta o algoritmo utilizado para proteção da fonte de tensão.

```
/* Código para cálculo do valor RMS sobre um oitavo de ciclo para
desarmar a saída do conversor no caso de ultrapassar um valor
específico, que neste caso é 2. */

if (tempo >= nsumV*Tperiod*0.125)
{
    nsumV = nsumV + 1.;
    rmsV = sqrt(sumV*delta/Tperiod);
    sumV = 0.;
}
sumV = sumV + Vbuckprotec*Vbuckprotec;
if ((rmsV > 2) || overcur_flagV == 1)
{
    ProtecV = 0;
    overcur_flagV = 1;
}
else
{
    ProtecV = 1;
}
```

Fig. 12. Fragmento do algoritmo de proteção do conversor de tensão

É calculado o valor rms a cada 1/8 de ciclo da forma de onda e uma comparação é realizada. Caso este valor supere o estabelecido, o sinal **ProtecV** é levado a zero, o que desliga as portas AND que servem de interface entre a saída do PWM e o acionamento das chaves conforme pode ser observado na Fig. 8.

Para o inversor da fonte de corrente, como se trata de uma fonte de corrente, naturalmente tem-se controle sobre a amplitude da corrente em sua saída. O raciocínio é semelhante, no entanto, foi utilizado o valor médio da corrente

medida no primário do trafo pelo sensor **Itrafo** para fins de proteção.

A próxima seção irá descrever o ambiente da simulação, assim como a conexão entre seus diversos componentes.

III. DESCRIÇÃO DO AMBIENTE DE SIMULAÇÃO

O microcontrolador (MCU) TMS320F28377S Delfino da Texas Instruments foi utilizado para o controle da carga fictícia, sendo este MCU baseado em uma CPU de 32-bits com suporte a ponto flutuante e capaz de alcançar até 200 MHz. Foi utilizada uma das portas de comunicação serial (SCI/UART) para a troca de dados entre o MCU e a simulação do circuito rodando em um PC. Esta configuração aumenta o realismo da simulação, sendo um primeiro passo para testar a execução do algoritmo de controle no MCU [9].

A simulação foi hospedada no PSIM - Powersim Inc, que é um *software* de simulação com solução em passo-fixo desenvolvido para sistemas baseados em eletrônica de potência. Este *software* possui um bloco chamado DLL Block, que permite ao usuário escrever um programa, compilá-lo e vinculá-lo ao PSIM. Este bloco foi utilizado para criar uma interface de troca de dados entre a simulação e o MCU, onde um programa escrito em C++ faz uso de funções padrões do Windows API para gerenciar a comunicação via porta serial utilizando o padrão RS-232.

Ainda neste trabalho, uma interface gráfica foi criada para interagir com a simulação via porta serial utilizando o padrão RS-232, com um controle enviando um comando para definir a amplitude da fonte de corrente e um *display* para exibir o valor rms da corrente calculado por um bloco dentro da simulação. A Fig. 13. ilustra um diagrama da conexão entre os componentes da simulação, onde é possível observar que a interface e o MCU se conectam à simulação através de uma porta serial específica.

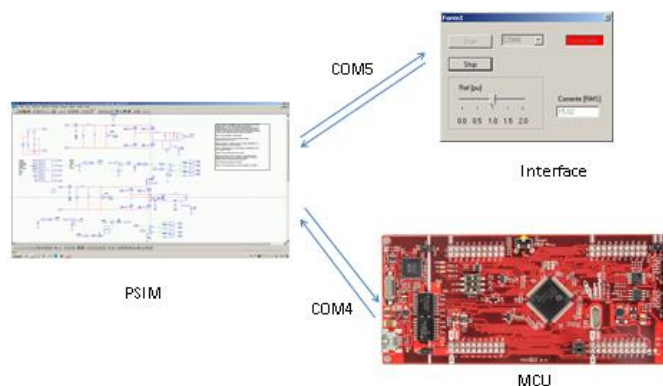


Fig. 13. Diagrama da conexão entre os módulos da simulação

A Fig. 14 apresenta o bloco de DLL utilizado para comunicação entre a simulação e o MCU, onde cada uma das medições são variáveis do tipo *double*. À esquerda do bloco estão os sinais de entrada que são enviados ao MCU via porta serial. À direita estão os sinais de controle transmitidos pelo MCU à simulação.

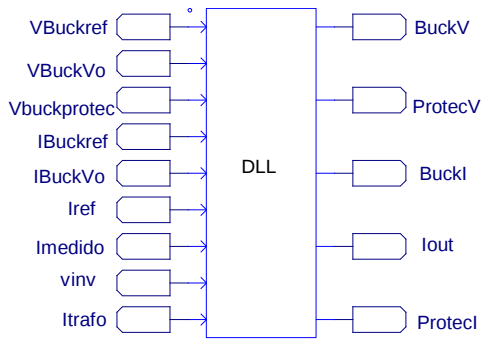


Fig. 14. DLL para comunicação entre simulação e MCU

Na Fig. 14, os sinais que entram no bloco DLL são as medições da planta já apresentada de forma desmembrada na Fig. 4, Fig. 5, Fig. 7, Fig. 8 e Fig. 9 e podem ser definidos como:

VBuckref	Valor de referência normalizado para conversor <i>buck</i> da fonte de tensão;
VBuckVo	Medição da tensão de saída do conversor <i>buck</i> da fonte de tensão;
Vbuckprotec	Medição da corrente de saída do conversor <i>buck</i> da fonte de tensão para fins de proteção;
IBuckref	Valor de referência normalizado para conversor <i>buck</i> da fonte de corrente;
IBuckVo	Medição da tensão de saída do conversor <i>buck</i> da fonte de corrente;
Iref	Sinal de referência normalizado para inversor da fonte de corrente;
Imedido	Medição da corrente no secundário do Trafo do inversor da fonte de corrente utilizado no controle preditivo da corrente;
vinv	Medição da tensão na saída do inversor da fonte de corrente utilizado no controle preditivo da corrente;
Itrafo	Medição da corrente de saída do inversor da fonte de corrente para fins de proteção.

Os sinais de controle à direita do bloco na Fig. 14 retornados pelo MCU irão atuar sobre as chaves e circuitos de proteção e podem ser definidos como:

BuckV	Controle da chave do conversor <i>buck</i> da fonte de tensão;
ProtecV	Controle da proteção da fonte de tensão;
BuckI	Controle da chave do conversor <i>buck</i> da fonte de corrente;
Iout	Controle das chaves da ponte H do inversor da fonte de corrente;
ProtecI	Controle da proteção da fonte de tensão.

A Fig. 15 mostra o esquema do controle da amplitude da fonte de corrente.

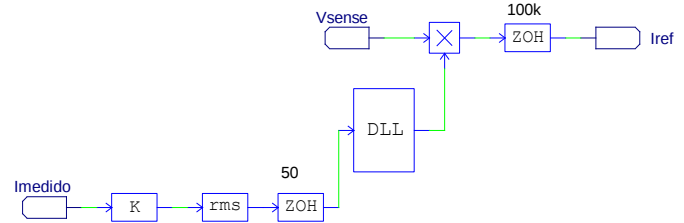


Fig. 15. Controle da amplitude da fonte de corrente

A amostra do sinal de tensão obtida pelo sensor **Vsense** passa por um bloco multiplicador. Este bloco recebe a constante de multiplicação de um segundo bloco DLL, que realiza a conexão via porta serial entre a simulação e a interface de controle, que é exibida na Fig. 16.

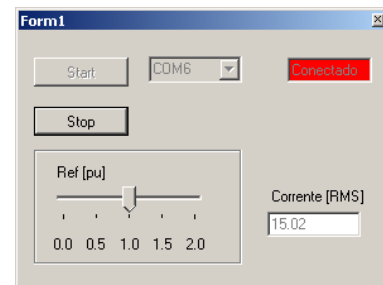


Fig. 16. Interface de controle

O resultado da multiplicação é o sinal **Iref**, que é discretizado e passa para entrada do bloco DLL de mesmo nome na Fig. 14. Este sinal serve de referência para o controle preditivo do inversor da fonte de corrente.

A interface de controle tem disponível um *trackbar* para controle da amplitude da corrente. Este controle varia entre 0 e 2, sendo referenciado a um valor nominal de corrente de 15 A. Os dois botões *Start* e *Stop* foram criados para conectar e desconectar a interface à porta serial. Além de enviar o comando para controlar a amplitude da corrente, a interface de controle também exibe o valor da corrente em rms que é calculado a partir da corrente medida na saída da fonte de corrente da carga fictícia. Este valor é enviado via porta serial através de um bloco DLL no PSIM, conforme pode ser observado na Fig. 15. A comunicação entre a interface e a simulação foi encapsulada em um protocolo simples, contendo 2 *bytes* de cabeçalho, 1 *byte* para os dados, 1 *byte* de fechamento e 1 *byte* de *checksum*.

Os sinais de referência VBuckref e IBuckref para os conversores *buck* são valores constantes que passam por um integrador e um limitador a fim de criar *softstart* através de uma rampa inicial. O esquema da geração da referência está ilustrado na Fig. 17.

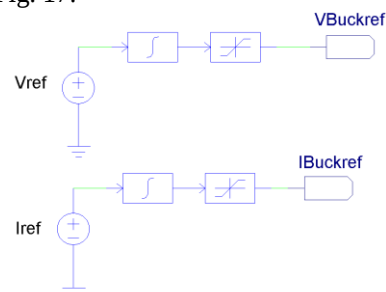


Fig. 17. Geração da referência dos conversores *buck*

IV. SIMULAÇÃO E RESULTADOS

Para apresentar o funcionamento da carga fictícia, resultados da simulação são apresentados a seguir. Resultados para as partes críticas do projeto serão apresentados, como THD e amplitude das formas de onda e fase da corrente. Também serão apresentados os resultados da análise de desempenho do MCU.

Na Fig. 18 tem-se, na parte superior, a forma de onda da corrente e, na parte inferior, a forma de onda da tensão.

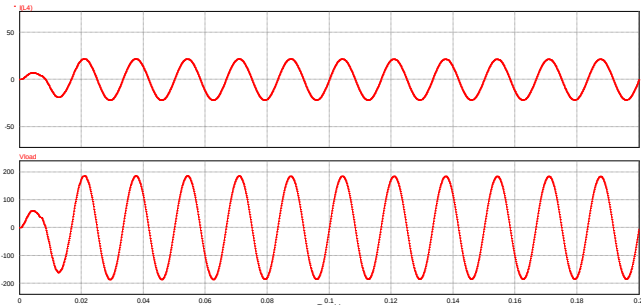


Fig. 18. Formas de onda da tensão e corrente

Vê-se claramente se tratar de formas de onda senoidais, com distorção abaixo de 1 %, conforme determina [4]. Os valores de THD da corrente e da tensão foram, respectivamente, de 0,8 % e 0,5 % conforme Fig. 19. Ainda de acordo com a Fig. 19, a amplitude da tensão de da corrente ficou dentro da faixa de 1% estabelecida em [4].

RMS Value		THD	
Time From	1.0000000e-007	Fundamental Frequency	6.0000000e+001 HZ
Time To	2.0000000e-001	(L4)	8.6448631e-003
I(L4)	1.5035834e+001	Vload	5.0884035e-003
Vload	1.2814590e+002		

Fig. 19. THD e amplitude da corrente e tensão

A fase entre a tensão e corrente ficou abaixo de 1° , também especificação definida em [4]. A Fig. 20 apresenta as duas formas de onda em fase, com o valor do fator de potência calculado pelo PSIM, sendo muito próximo de 1.

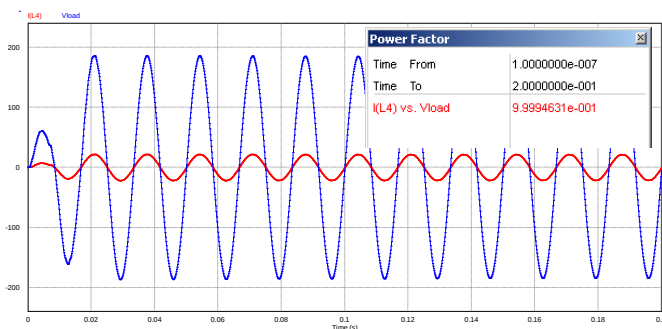


Fig. 20. Fase entre corrente e tensão

A Fig. 21 apresenta o detalhe do cruzamento do zero das duas formas de onda, onde se vê que, embora dentro da especificação, a corrente está adiantada de aproximadamente $0,56^\circ$.

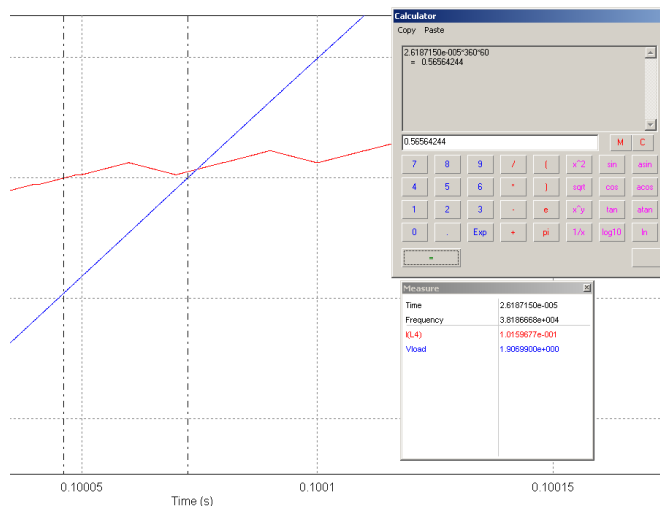


Fig. 21. Detalhe da fase entre corrente e tensão

A Fig. 22 mostra a evolução do consumo de potência medida na saída do retificador dobrador, que alimenta toda a fonte. No início da simulação há um pico de consumo, chegando a 400 W, justificado pelo carregamento dos elementos armazenadores de energia. Após 100 ms, o consumo se estabiliza em aproximadamente 10 W.

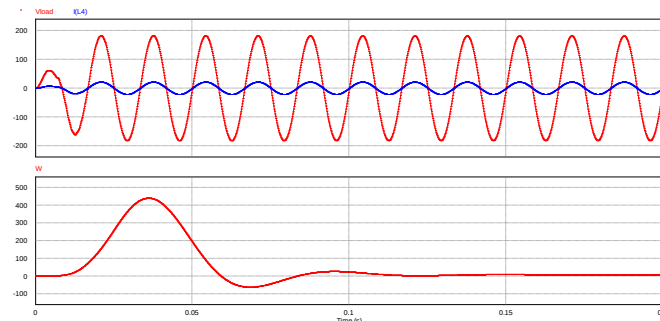


Fig. 22. Consumo de potência ao longo da simulação

Apesar do baixo valor de consumo total, a potência equivalente vista pelo medidor na saída da carga fictícia fica em torno de 1800 W conforme valores apresentados na Fig. 23, sendo esta a principal vantagem da carga fictícia.

Real Power		Measure	
Time From	1.0000000e-007	Time	1.4900000e-001
Time To	2.0000000e-001	Vload	-7.2947579e+001
Vload vs. I(L4)	1.8720317e+003	I(L4)	-8.3784768e+000
Only 2 curves must be i...		vV	1.0405364e+001

Fig. 23. Potência resultante na saída e consumida na carga fictícia

O ambiente de desenvolvimento da Texas Instruments dispõe de uma funcionalidade de contagem de ciclos de *clock* entre dois pontos de *breakpoint*. Esta funcionalidade foi explorada com a finalidade de obter o tempo de execução do algoritmo de controle, sendo convertida em tempo tomando como base a frequência de operação do MCU, de 200 MHz. Também é possível, no modo de depuração do *software*, determinar onde o programa irá rodar no MCU, se na memória FLASH ou na RAM. No momento da compilação, pode-se ainda configurar o uso ou não do suporte a operações de ponto

flutuante, o que melhora consideravelmente o desempenho da CPU. Estas três funcionalidades foram avaliadas e, para análise de desempenho do MCU, três condições foram consideradas:

- 1) Programa rodando na FLASH e com suporte do hardware de ponto flutuante;
- 2) Programa rodando na RAM e com suporte do hardware de ponto flutuante;
- 3) Programa rodando na RAM e sem suporte do hardware de ponto flutuante.

A primeira condição avaliada gastou 351 ciclos de clock para executar todo o algoritmo de controle, incluindo os dois controladores PI dos conversores *buck*, o controle preditivo da corrente e a proteção dos conversores. A Fig. 24 exibe um detalhe da ferramenta de medição do período apontado, que equivale a aproximadamente $1,8 \mu s$. Considerando que o intervalo de iteração do algoritmo de controle foi fixado em $10 \mu s$, esta condição de operação atende ao projeto.

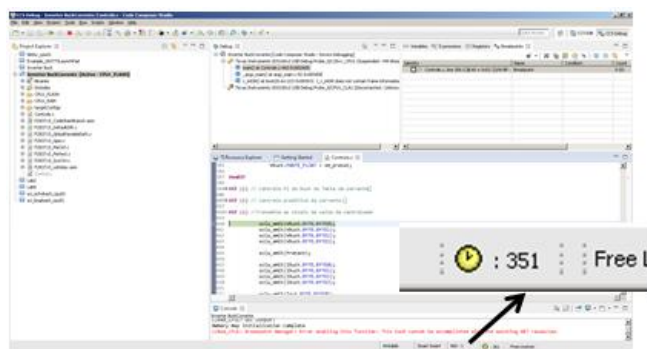


Fig. 24. FLASH com suporte de hardware de ponto flutuante

O resultado da segunda avaliação teve como resultado o apresentado na Fig. 25, onde o MCU gastou 283 ciclos de clock para executar o mesmo programa, o que equivale a aproximadamente $1,4 \mu s$. Tal resultado era esperado, já que o desempenho da memória RAM em termos de velocidade é ligeiramente superior à memória FLASH.

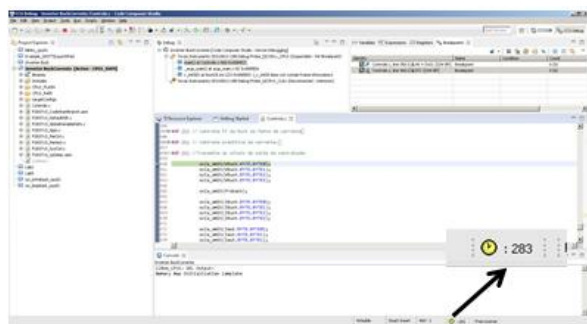


Fig. 25. RAM com suporte de hardware de ponto flutuante

Por último, a Fig. 26 apresenta o resultado do algoritmo de controle rodando em RAM, mas agora sem o suporte do hardware de ponto flutuante. Nesta condição, o MCU gastou 5066 ciclos de clock, o que equivale a aproximadamente $26 \mu s$, o que torna o projeto inviável para esta condição.

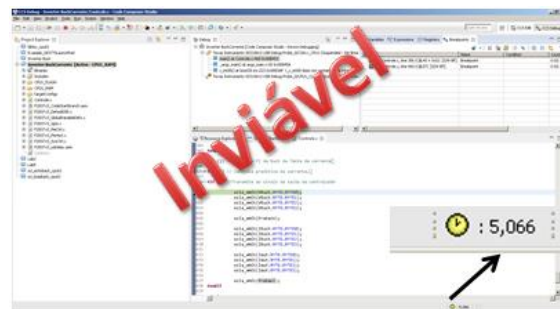


Fig. 26. RAM sem suporte de hardware de ponto flutuante

Uma segunda análise foi realizada quanto à proteção do sistema. Para simular uma falta na fonte de tensão, um curto em sua saída foi provocado por uma resistência baixa conforme pode ser observado na Fig. 27.

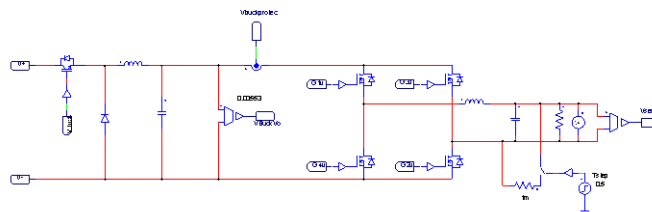


Fig. 27. Simulação de falha na fonte de tensão

Na Fig. 28 tem-se, na parte superior, o sinal de proteção e na parte inferior, a medição da corrente na saída do conversor *buck* obtida pelo sensor *Vbuckprotec*.

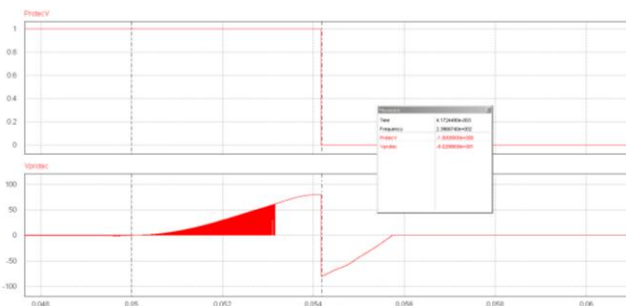


Fig. 28. Proteção da fonte de tensão

O curto ocorre em 50 ms, e o sinal de proteção atua 4 ms depois indo a zero e consequentemente desligando as chaves de potência através das portas AND que servem de interface entre a saída do PWM e o acionamento das chaves conforme pode ser observado na Fig. 8.

Para a fonte de corrente, a falha inserida foi uma chave que se abre num instante determinado, conforme pode ser visto na Fig. 29.

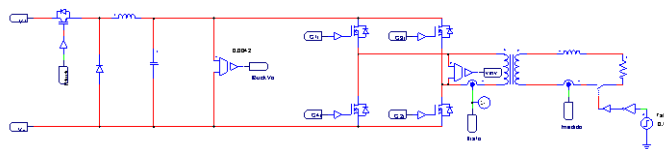


Fig. 29. Simulação de falha na fonte de corrente

Após a abertura da chave, que ocorre em 100 ms, cerca de um semi ciclo depois a proteção atua, desligando as chaves através das portas AND.

A Fig. 30 mostra, na parte superior a corrente no secundário do Trafo.

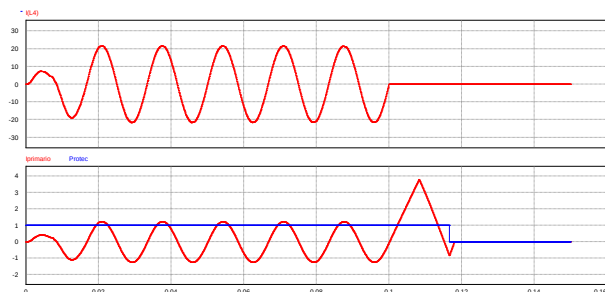


Fig. 30. Proteção da fonte de corrente

Na parte inferior da Fig. 30 estão a corrente no primário em vermelho e o sinal de controle da proteção em azul. Vê-se que no instante da falha, a corrente no secundário vai a zero. Para tentar compensar o erro, o controle atua aumentando o ciclo de trabalho, o que acaba por gerar uma rampa com um valor médio. A detecção deste valor médio é utilizada para atuar a proteção.

V. CONCLUSÃO

Um ambiente de simulação com *processor-in-the-loop* utilizando o PSIM e comunicação serial via RS-232 foi apresentado. Os resultados da simulação mostraram que os objetivos do projeto, como o atendimento às especificações, desempenho do microcontrolador e sistema de proteção foram alcançados.

Estudos posteriores serão outras topologias de filtro na saída dos conversores, visando reduzir a dimensão física dos componentes, sincronizar o tempo da simulação com o tempo do processador a fim de simular no ambiente de *hardware-in-the-loop*, podendo utilizar assim mais recursos internos como seu PWM e realizar a montagem do protótipo. Por fim, incrementar o protocolo de comunicação, com detecção de erros mais robusta.

REFERÊNCIAS

- [1] ABRADÉE, A distribuição de energia. [Online, acessado em dez/2015]. Disponível: <http://www.abradee.com.br/setor-de-distribuicao/a-distribuicao-de-energia>
- [2] INMETRO, Vocabulário Internacional de Metrologia: Conceitos fundamentais e gerais e termos associados (VIM 2012). [Online, Acessado em abr/2016]. Disponível: http://www.inmetro.gov.br/inovacao/publicacoes/vim_2012.pdf
- [3] ANEEL, Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST, Revisão 2: Módulo 5 – Sistemas de Medição. [Online, acessado em jan/2016]. Disponível: http://www.aneel.gov.br/arquivos/PDF/Modulo5_Revisao_2.pdf
- [4] Inmetro, Portaria Inmetro nº 587, de 05 de novembro de 2012. [Online, acessado em jan/2016]. Disponível: <http://www.inmetro.gov.br/legislacao/rtac/pdf/RTAC001929.pdf>
- [5] INMETRO, NIT-DIGEL-004 - Procedimento de Verificação em Campo de Medidores de Energia Elétrica por Solicitação do Usuário/Proprietário. . [Online, acessado em jan/2016]. Disponível: <http://www.inmetro.gov.br/metlegal/docDisponiveis.asp>
- [6] L.G.B. Rolim, U.A. Miranda, B.W. França, R.M. Fernandes, M. Aredes, "Hardware-in-the-loop evaluation of dsp-controlled converters", em Power Electronics Conference, 2009, COBEP '09, pp.805 – 809.
- [7] G. H. Bode, P. C. Loh, M. J. Newman, and D. G. Holmes, "An improved robust predictive current regulation algorithm," em Proc. 5th Int. Conf. Power Electron. Drive Syst. (PEDS'03), 2003, vol. 2, pp. 1058-1063.

- [8] Simone Buso, Paolo Mattavelli, "Digital Current Mode Control" em *Digital Control in Power Electronics*, EUA.
- [9] Zhenhua Jiang, Rodrigo Leonard, Roger Dougal, Heman Figueroa, Antonello Monti, "Processor-in-the-Loop Simulation, Real-Time Hardware-in-the-Loop Testing, and Hardware Validation of a Digitally-Controlled, Fuel-Cell Powered Battery-Charging Station", em Power Electronics Specialists Conference (PESC 04), 2004, vol. 3, pp. 2251 – 2257.