Proposta de Execução de um Método de Gerenciamento de Demanda através de Controle Externo de Inversor Fotovoltaico com Bateria

João Felipe A. Santiago*. Lucas V. Belinaso*. Mauricio Sperandio*. Rafael C. Ney**

**Universidade Federal de Santa Maria, Santa Maria, RS 97105-900 Brasil (e-mail: joaofelipe.santiago@ecomp.ufsm.br, lucas@gepoc.ufsm.br, mauricio.sperandio@ufsm.br).*

***CEEE-D, Grupo Equatorial (e-mail: rafael.ney@ceee.equatorialenergia.com.br)*

Abstract: The energy management method called Peak Shaving is a valuable approach, which increases the efficiency of the electricity grid and reduces costs for consumer agents, especially in contexts of high demand. Furthermore, this technique proves to be interesting for the new Brazilian energy challenges. This work proposes to build a computational infrastructure in order to enable the execution of this energy management method through a photovoltaic generation unit. In this sense, the construction of a supervisory will be presented, which will provide a view of the behaviour of the multiple electrical variables, which involve the generation unit, enabling the realisation of the energy optimization technique. This system consisted of a Modbus serial communication, MQTT message protocol, Python programming language, InfluxDB database and the Grafana interactive viewer. The infrastructure development was satisfactory with regard to data monitoring, but the control steps did not meet the design requirements.

Resumo: O método de gerenciamento de energia chamado Peak Shaving é uma abordagem valiosa, a qual eleva a eficiência da rede elétrica e reduz custos para os agentes consumidores, principalmente em contextos de ultrapassagem de demanda contratada. Além do mais, tal técnica se mostra interessante para os novos desafios energéticos brasileiros. Este trabalho se propoẽ a construir uma infraestrutura computacional de modo a viabilizar a execução deste método de gerenciamento energético em uma unidade de geração fotovoltaica com baterias. Nesse sentido, será apresentado a construção de um supervisório, o qual fornecerá uma visão do comportamento das múltiplas variáveis elétricas, que envolvem a unidade de geração. Tal sistema foi constituído por uma comunicação serial Modbus, protocolo de mensagens MQTT, linguagem de programação Python, banco de dados InfluxDB e o visualizador interativo Grafana. O desenvolvimento da infraestrutura foi satisfatória no que diz respeito ao monitoramento de dados, porém as etapas de controle ainda não puderam ser finalizadas.

Keywords: Energy; Modbus; Raspberry; SCADA; Inverter

Palavras-chaves: Energia; Modbus; Raspberry; SCADA; Inversor.

1. INTRODUÇÃO

Projeta-se que no Brasil, até 2024, haverão mais de 880 mil agentes (residências, comércios, indústrias, instituições) com uma unidade de geração de energia solar fotovoltaica, segundo dados da ANEEL (2017). Isso significa que o país teria mais de 3,2 GW de potência instalada proveniente deste tipo de recurso energético. Apesar de tal ampliação gerar impactos positivos em termos de qualidade energética, inovação e meio ambiente, existem consequências problemáticas associadas a esse contexto que se apresenta no horizonte. Para KIMAIYO (2019), o aumento das conexões de gerações fotovoltaicas somada ao aumento de cargas não lineares desafiam a qualidade de energia da rede elétrica. Isso significa que o sistema de energia pode apresentar variações indesejadas de tensão, frequência e baixa confiabilidade.

Portanto, sistemas de armazenamento de energia (SAE) tornam-se interessantes em termos de otimização energética. No que se refere ao uso de estratégias de otimização utilizando SAE, uma das técnicas mais comuns é o alívio de carga denominado *peak shaving*. Tal estratégia consiste em definir um limite de potência que um fornecedor irá consumir da rede (P_{Max}), que pode ser definida como a demanda contratada. Durante a operação do sistema é preciso medir a potência instantânea de todos os elementos do sistema a fim de definir a potência necessária para a bateria compensar a ultrapassagem de demanda. Nos momentos em que a

demanda da(s) carga(s) é menor, a bateria é carregada pela fonte energética alternativa.

O estudo de gerenciamento de energia com baterias exige intrinsecamente a existência de um sistema capaz de supervisionar e atuar sobre os dispositivos participantes de tal infraestrutura. Neste sentido, este trabalho tem como finalidade operacionalizar um sistema SCADA (*Supervisory Control and Data Aquisition*) para atuar em uma unidade de geração de energia fotovoltaica e que implementa a técnica de gerenciamento de energia *peak shaving*. Logo, para a concretização desta proposta utilizou-se o computador externo *Raspberry Pi* modelo *3 b+*, tanto para se comunicar com o inversor quanto para enviar os parâmetros lidos até um servidor. A comunicação entre o inversor e o computador externo se deu através do protocolo serial ModBus. Já para o envio de mensagens da máquina computacional até o broker usou-se o protocolo MQTT. Foi empregado o banco de dados InfluxDB e o aplicativo web para visualização interativo Grafana. A unidade de geração de energia solar conta com uma bateria da fabricante Dyness modelo PowerBox F-5.0 e um inversor híbrido monofásico do fabricante Solis modelo RHI-5K-48ES. A Figura 1 apresenta um diagrama que resume as variáveis e equipamentos presentes no sistema. Já a Figura 2 mostra o inversor e a bateria que foram utilizados.

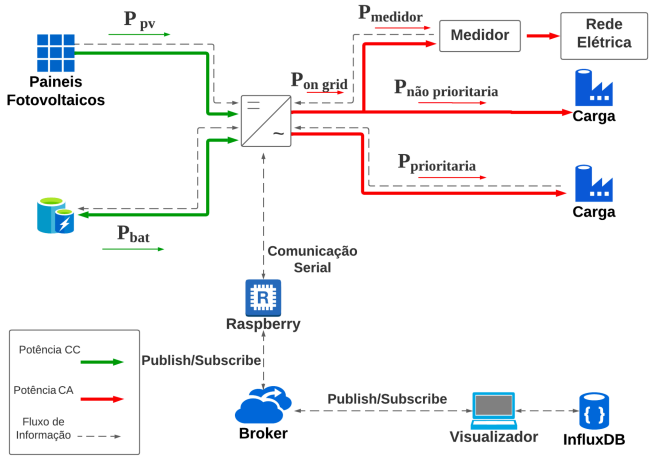

Fig 1. Diagrama do sistema.

Fig 2. Foto dos dispositivos instalados.

2. O MÉTODO PEAK SHAVING

Dentro do contexto de *smart grids,* KARMIRIS (2013) discorre que a abordagem *peak shaving* tem o potencial de se tornar uma das aplicações mais importantes dessa área. Isso porque, baseado em momentos específicos de demanda é possível amortecer o custo da energia. Seu princípio de funcionamento é bastante simples porém traz fortes consequências do ponto de vista financeiro. Para esse tipo de gerenciamento de energia, LEVRON (2012) explica que para ser implementado, basta identificar os momentos em que uma carga elétrica qualquer exige da rede elétrica, uma potência acima de um determinado limiar. Quando isso acontece, uma fonte energética secundária atende a demanda que está acima deste limite.

Existem duas grandes vantagens, segundo KARMIRIS (2013), que essa aplicação traz. A primeira envolve considerar que toda a rede elétrica é dimensionada para atender os picos de demanda. Porém, tal situação ocorre em períodos relativamente curtos do dia, o que faz com que toda a infraestrutura da rede de transmissão e distribuição fique subutilizada na grande maioria do tempo. Isso significa que o custo para o consumidor final é fortemente baseado nesse pequeno intervalo de tempo. Outro ponto trazido por KARMIRIS (2013) é o de levar em conta que, quase sempre, os picos de demanda coincidem com o momento em que a taxa de energia atinge seu maior preço. Portanto, para os consumidores que pagam taxas variáveis e multas por exceder determinados limites de potência, o uso da técnica de *peak shaving* tem o potencial de reduzir o valor que será pago para as administradoras da rede elétrica.

2.1 Concepção da Execução do Método

Em termos de conceito de implementação, foi preciso providenciar a existência de uma forma ou uma estrutura apta a configurar determinados parâmetros operacionais da bateria, através de uma comunicação remota. Para isso, levou-se em consideração duas características do inversor. A primeira, diz respeito a sua capacidade de parametrizar o funcionamento da bateria via comunicação serial CAN. Já a segunda, leva em conta a disponibilidade dele receber, de máquinas externas, comandos que irão alterar as características operacionais do equipamento de armazenamento, via protocolo Modbus. Com esses materiais e recursos, a estratégia, em nível conceitual, utilizada para pôr em prática o *peak shaving* consiste na realização de verificações periódicas da potência da(s) carga(s) conectada(s) ao inversor ($P_{prioritária}$ e $P_{não\ prioritária}$) sendo a potência prioritária aquela que é alimentada exclusivamente de forma *off-grid* e a não prioritária é aquela cuja fonte é híbrida; o nível de estado de carga (*SOC*); a potência vinda dos painéis solares (P_{pv}) e a potência do medidor ($P_{medidor}$). Em linhas gerais, dentro de um estágio de averiguação, o sistema SCADA compara a demanda energética suprida pela rede, com o limite de fornecimento (P_{max}) . Caso haja carga suficiente na bateria e o limite tiver sido superado, a bateria é então descarregada com uma potência apta a compensar a extrapolação da linha de tolerância. Foi estipulado também duas parametrizações para carregar o dispositivo de armazenamento. Uma em que a potência de carregamento é constante ($P_{carga,\ const}$), e outra que a carrega com a máxima potência disponível.

Nos momentos em que a demanda está abaixo do limite, ocorre a verificação da viabilidade de se realizar o carregamento da bateria. Na condição de não ser possível nem realizar o atendimento de demanda da carga, nem o carregamento do dispositivo de armazenamento, a bateria é então desabilitada.

Considerando que o inversor possua elevada eficiência de conversão de energia (> 97%), a equação de equilíbrio de potências da Figura 1 é dada por:

$$P_{pv} + P_{bat} - P_{medidor} - P_{não\ prioritaria} - P_{prioritaria} = 0 \quad (1)$$

onde

$$P_{não\ prioritaria} = P_{on\ grid} - P_{medidor} \quad (2)$$

A potência da bateria é dada por:

$$P_{bat} = P_{medidor} + P_{não\ prioritaria} + P_{prioritaria} - P_{pv} = P_{on\ grid} + P_{prioritaria} - P_{pv} \quad (3)$$

A potência do medidor deve ser limitada de acordo com a demanda. Com isso, quando $P_{medidor}$ é positivo significa que a rede elétrica está recebendo energia do inversor. Na situação contrária ($P_{medidor} < 0$) a demanda de $P_{não\ prioritaria}$ é maior do que P_{pv} consegue entregar. Portanto, neste contexto, a rede elétrica está fornecendo energia ao sistema.

Neste sentido, quando $P_{medidor}$ ultrapassa o limite de demanda ($P_{medidor} < P_{max}$) e a bateria possui SOC suficiente, ela é então descarregada, respeitando e considerando a equação de equilíbrio de potências do sistema e P_{max}. Por outro lado, na circunstância em que $P_{medidor}$ indica que o limite contratado não está sendo ultrapassado, a bateria é carregada com a máxima potência disponível.

A tabela 1 resume a determinação da potência da bateria necessária para implementar o *peak shaving*. Portanto, o carregamento e descarregamento do dispositivo de armazenamento ocorre apenas em condições adequadas. Importante mencionar que os valores de SOC_{max} e SOC_{min} precisam ser tais que evitem *overcharging* e *undercharging* da bateria, contudo seus valores dependem das especificações de seu fabricante. No escopo deste trabalho, a bateria utilizada exigiu um SOC_{min} de 10% e SOC_{max} arbitrário de 80% . Percebe-se na equação 1, a existência do parâmetro $P_{carga,\ const}$, o qual consiste em uma potência de carregamento cujo valor é dimensionado com a intenção de otimizar a utilização da bateria.

Importante mencionar que o inversor precisa estar configurado de tal forma que apenas ele esteja habilitado para carregar a bateria.

Tabela 1. Determinação da potência da bateria

P_{bat}	Condição
$P_{max} + P_{não\ prioritaria} + P_{prioritaria} - P_{pv}$	$P_{medidor} \geq P_{max}$ e $SOC < SOC_{max}$
$-P_{max} + P_{não\ prioritaria} + P_{prioritaria} - P_{pv}$	$P_{medidor} \leq -P_{max}$ e $SOC > SOC_{min}$
$-\lvert P_{carga,\ const}\rvert$	$\lvert P_{medidor}\rvert \geq \lvert P_{max}\rvert - \lvert P_{carga,\ const}\rvert$ e $SOC < SOC_{max}$

0	outros casos

3. ESTRUTURA COMPUTACIONAL

Os equipamentos empregados para realização do sistema SCADA foram um *Raspberry Pi 3 b+* e um conversor USB/RS485, modelo CH340. O estabelecimento de conexão entre o inversor e o Raspberry via padrão RS485 fez-se via o emprego do conversor USB/RS485. Este equipamento, por sua vez, possui uma interface de conexão bastante simples, pois exige-se apenas a ligação entre o conversor de interface e o inversor, nos barramentos A e B do padrão RS485. Tal configuração possibilitou a operacionalização da comunicação serial Modbus.

Com o uso de um cabo Ethernet para conectar o Raspberry à Internet somado à exploração do protocolo MQTT, o computador pode se conectar com um *broker*, o qual tem como propósito receber todas as publicações realizadas por essa máquina. Em uma outra ponta, há o *Visualizador* (Figura 1), o qual é um segundo computador conectado ao mesmo *broker*. Portanto, toda a vez que o Raspberry realizar uma publicação (*publish*) esta segunda máquina, ao receber a mensagem, a armazena no banco de dados InfluxDB, o qual por sua vez estará conectado ao visualizador de dados Grafana.

Em resumo o *Visualizador* apenas acompanha a alteração do comportamento das variáveis elétricas, ao passo que o Raspberry possui todos os parâmetros e instruções para aplicar o método de gerenciamento proposto. Isso significa dizer que, em tempo de compilação, o Raspberry recebe valores fixos de P_{max}, SOC_{max}, SOC_{min} e $P_{carga,const}$; por outro lado, em tempo de execução, as variáveis $P_{não\ prioritaria}$, $P_{prioritaria}$, P_{pv}, $P_{medidor}$ e SOC são lidas periodicamente do inversor. Por fim, com esses dados, aplicam-se as condições descritas na Tabela 1 para definir o valor de P_{bat}.

3.1 Recursos de Software

Todo o código necessário para implementar o sistema SCADA foi desenvolvido, utilizando a linguagem de programação Python. Isso devido a facilidade de uso dos pacotes necessários tanto para a comunicação serial, quanto para a troca de mensagens via *broker*. Em relação ao protocolo Modbus, este trabalho utilizou o pacote *pymodbus*, o qual possui o módulo *client.sync.* Dentro deste módulo há a definição da classe *ModbusSerialClient*, nela é possível acessar todos os métodos necessários para estabelecer uma comunicação entre o Raspberry e o inversor. Para a comunicação indireta foi utilizado o pacote *paho*, que é quem possui uma variedade de módulos que tangem o protocolo MQTT. No escopo desta implementação, operou-se com a classe *client*, presente no subpacote *mqtt*. Por último, foi necessário o emprego da biblioteca Influxdb, para assim realizar as inserções das amostras coletadas no banco de dados.

3.2 Especificidades do Protocolo Modbus

Ao todo, 61 registradores Modbus do inversor são monitorados ou controlados. Todos respeitam a política de organização definida pelo fabricante do inversor. Em suma, a organização dos parâmetros presentes nos registradores baseia-se em endereço, tamanho em bits, escala de conversão e nome. Cada uma dessas características cria exigências distintas em termos de conversão. Isso porque, o dado bruto que se interage com cada registrador não representa o verdadeiro valor da variável que deseja-se conhecer ou relacionar-se. Para este trabalho, utilizou-se apenas dois tipos de registradores Modbus, os *Input Registers* e os *Holding Registers*. O primeiro é composto de 16 bits e restringe-se a fornecer apenas leitura de informações. Já o segundo tipo, também é formado por 16 bits, porém permite a realização de leitura e escrita de dados.

O endereço nada mais é do que a referência que precisará ser utilizada tanto para realizar a leitura de uma variável, quanto para seu eventual controle. O tamanho de um parâmetro varia entre 16 e 32 bits, ou seja, é definido por um ou dois registradores. Portanto, métodos específicos precisam ser utilizados para a correta conversão de cada caso.

Para os casos de uma variável de 32 bits sem sinal, é necessário realizar o deslocamento apropriado dos 16 bits mais significativos, concatená-los com os menos significativos e atribuir esse número a uma variável. A constante utilizada para realizar a concatenação é tal que desloca todos os bits do registrador mais significativos 16 bits à esquerda. Portanto, basta multiplicar este registrador por 2^{16} (65.536) e somar com o registrador dos primeiros 16 bits da variável amostrada.

Porém, para as variáveis de 32 bits com sinal exige-se levar em consideração as regras do complemento de dois, para a representação de números negativos. Para esta circunstância, primeiro o código verifica se o valor acessado é negativo ou não. As regras deste tipo de representação consideram que o bit mais significativo indica o sinal do número (0 é positivo, 1 negativo). Ou seja, é necessário verificar qual é o valor do registrador que representa a parte alta do número. Assim, caso ele seja maior do que 2^{15} (32.768), tem-se a certeza de que trata-se de um número negativo. Nesta essa circunstância, além da multiplicação por 2^{16} para realizar o deslocamento é preciso inverter todos os bits e somar mais um. Caso o número seja positivo, apenas o deslocamento se faz necessário. Nas circunstâncias em que a variável amostrada é de 16 bits com sinal, apenas a verificação do sinal e sua

eventual modificação é exigida. O valor de verificação é o mesmo que para o caso de 32 bits.

O último fator necessário para realizar a devida conversão das amostras monitoradas é a escala. De acordo com o protocolo de comunicação do fabricante, cada registrador precisa ser multiplicado por uma constante adequada. Por exemplo, o registrador 33134, indica o valor da corrente fluindo sobre a bateria e seu fator de escala é 10, portanto para ter o valor adequado é necessário multiplicar a magnitude do registrador por 0,1.

3.3 Especificidades do Protocolo MQTT

Além de interagir com o inversor, o Raspberry se comunica com o *broker* público da instituição *Eclipse Foundation,* chamado *mqtt.eclipseprojects.io*. A conexão com esse broker exige, previamente, a instanciação de um objeto cliente MQTT e dois callbacks. Um dos métodos (*on_connect()*) é utilizado para quando a conexão é estabelecida com o *broker,* o outro (*on_message()*) para os casos em que o computador recebe uma mensagem de uma publicação (não se aplica ao Raspberry, pois ele não realiza inscrição em nenhum tópico).

Após realizar as configurações prévias e a conexão com o broker, um *loop* é executado. Este *loop é definido* pelo método *loop_start()* e está presente no módulo *paho*. Tal método tem duas utilidades, a primeira é liberar a *main* de possíveis bloqueios de processo que podem acontecer devido aos procedimentos internos de uma comunicação de rede, e o outro visa garantir uma conexão com o broker de maneira ininterrupta.

Importante mencionar que esses procedimentos são executados tanto no Raspberry (agente publicador de mensagens), quanto no *Visualizador* (agente que realiza *inscrições*).

Após finalizar as devidas conversões relacionadas ao acesso dos dados internos do inversor e o estabelecimento de conexão com o broker, o Raspberry deve realizar a publicação dos valores. Para isto, basta incitar o método *mqtt_publish()*, tal função recebe três parâmetros; a referência do objeto cliente, o nome do tópico em que se deseja realizar a publicação e o seu respectivo valor.

No *Visualizador*, a realização das inscrições é realizada no momento em que o callback *on_connect()* é executado. Isso significa que, após essa etapa, toda vez que o Raspberry, ou qualquer outra máquina conectada ao mesmo broker realizar a publicação de um valor em algum dos tópicos inscritos, o computador de visualização irá receber. Mais especificamente, o método *on_message()* será executado para esse tipo de situação.

3.4 Inserção das Amostras no Banco de Dados

Os procedimentos realizados, após o recebimento de uma mensagem, consistem em realizar a instanciação de um objeto para interagir com o banco de dados InfluxDB e inserir as medidas no sistema de armazenamento de informação. A instanciação envia 4 parâmetros, o endereço IP do banco de dados, sua porta (o influxDB utiliza a 8086 por padrão), o nome do banco de dados, seu nome de usuário e senha.

Após a criação deste objeto, o método *send_data_to_influx_db()* é chamado para, enfim, anexar o dado no banco de dados. Este método recebe como parâmetros a referência do objeto cliente, o nome do tópico e o valor a ser inserido. Internamente, sua lógica consiste em utilizar estes argumentos para elaborar um formato de dados JSON e posteriormente chamar o método, da biblioteca influxdb, *write_points()*.

3.5 Visualizador de Dados

A última etapa do sistema SCADA diz respeito à visualização do monitoramento e controle do inversor através do software Grafana. Portanto, todas essas informações estão presentes no painel de controle Solis_PickShaving_inv1 da aplicação. Importante ressaltar que tais elementos estão presentes apenas no computador *Visualizador*, o que significa que apenas máquinas computacionais conectadas na mesma LAN estão aptas a acessar esses dados.

4. RESULTADOS

4.1 Monitoramento

Todas as funcionalidades que tangem o monitoramento dos parâmetros elétricos do inversor atingiram níveis de operacionalidade completa. O painel de controle foi dividido em 4 filas: Status, *Battery, Meter e Backup*. A primeira fileira apresenta informações a respeito do inversor em si, tais como modelo do dispositivo, estado atual, modo de operação e dados de erros.

Já na fila *Battery* há uma série de dados referentes às variáveis operacionais da bateria de lítio-ion. Neste campo de informações é possível ter conhecimento da corrente que está fluindo sobre o dispositivo, a tensão aplicada, o valor da corrente máxima para carga e descarga, habilitação da bateria, valor de proteção contra overvoltage, undervoltage, modo de operação, entre outros tipos de dados.

As informações presentes em *Meter* concentram-se em fornecer uma compreensão a respeito da potência recebida do barramento CC dos painéis; a potência ativa gerada pelo inversor e a entregue aos medidores da rede elétrica; corrente e tensão dos medidores; valor dos limitadores de potência ativa gerada pelo inversor; corrente e tensão do barramento CA do inversor e estado atual do equipamento.

O campo de dados *Backup* apresenta dados relativos à carga conectada ao inversor. Isso significa que é possível conhecer a potência, corrente e tensão dela; além da visualização de

parâmetros de controle relacionados à tensão, frequência e habilitação dos dados de controle.

A Figura 3 apresenta o resultado do monitoramento, presente no software Grafana, do comportamento da tensão eficaz da energia que passa pelo medidor, ao longo do tempo.

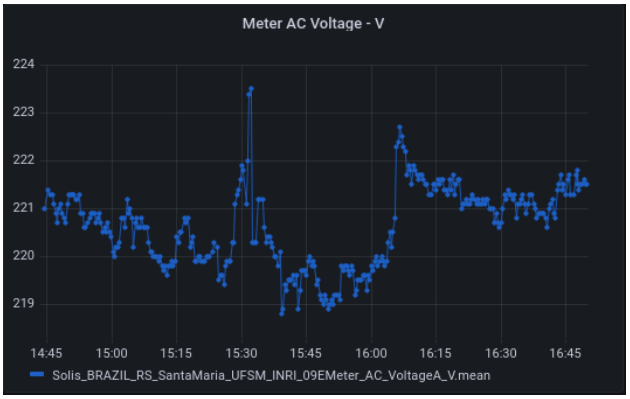

Fig 3. Tensão eficaz ao longo do tempo.

4.1 Controle

Criou-se a classe *Parametros_IO*, com o intuito de facilitar a programação e a compreensão da lógica do sistema. Essa classe é composta exclusivamente de atributos, os quais ao todo são 11. A grande vantagem de sua utilização é a facilidade que ela traz para retirar ou acrescentar novas variáveis para serem monitoradas e/ou controladas. No que tange a gestão dos parâmetros do sistema, via protocolo Modbus, três atributos da classe são utilizados. São eles: o endereço do registrador, o valor que deseja-se atribuir a ele e por fim, o valor de retorno que o inversor fornece ao Raspberry a cada solicitação de escrita.

A instanciação dos objetos desta classe é baseada nos valores existentes em uma lista chamada *INV_REGS*. Isso significa que a definição das medidas a serem efetuadas pelo sistema e os valores a serem escritos nos registradores acontecem em tempo de compilação. A consequência desta forma de funcionamento cria um caráter estático no modo como o controle é executado.

Como descrito na subseção 2.1, a operação do método *peak shaving* depende fortemente do monitoramento da potência que a carga exige e o SOC da bateria. Além de exigir, em linhas gerais, a alteração de 4 valores. Isso significa ter que gerenciar a habilitação do dispositivo de armazenamento, definir a realização de carregamento ou descarregamento, determinar a potência de saída ou entrada da bateria e impedir que a rede carregue o dispositivo de armazenamento.

Infelizmente, apenas o registrador relacionado com a imposição de limitação entre rede e bateria pode ser integralmente controlado. Isso pois, apesar do monitoramento presente no software Grafana indicar alteração bem sucedida nos valores dos registradores relacionados à bateria, tais modificações não apresentaram alteração no comportamento final no equipamento de armazenamento.

Um dos poucos comportamentos que o sistema construído foi capaz de controlar foi a potência nominal que o inversor entrega ao sistema, que em tese pode ser para a bateria, cargas ou rede elétrica. Neste caso, variando o valor do registrador chamado de *Power Limiting Seting* pela documentação do fabricante foi possível atingir tal feito. Como a potência nominal do inversor é de 3 kW esta variável então define, em porcentagem, o quanto ela será gerada pelo inversor.

As Figuras 4 e 5 permitem observar o fenômeno descrito, pois a alteração *Power Limiting Seting* impactou diretamente na potência CC, vinda dos painéis solares uma vez que, sob condições de incidência solar normal, a potência que chega no inversor entre os horários de 15 h e 30 min até pouco mais de 16 h são menores do que a potência após este horário. Portanto, nesta circunstância, tal fenômeno ocorre exclusivamente pela atuação da variável *Power Limiting Seting.*

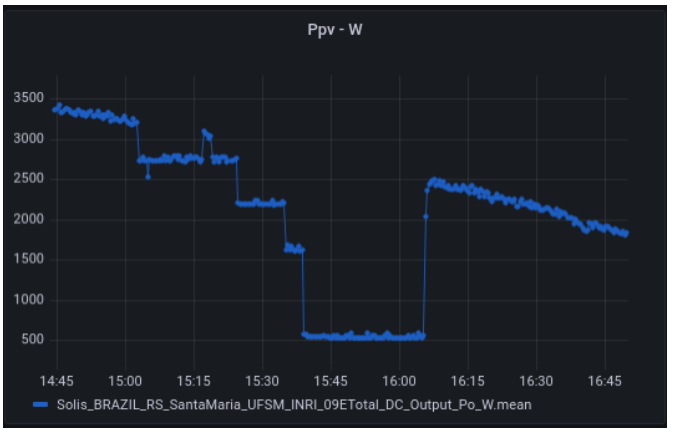

Fig 4. Potência gerada pelo inversor (P_{pv})

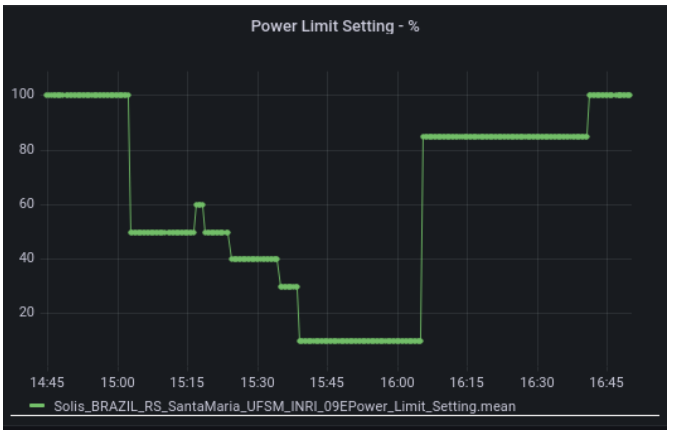

Fig 5. Controle da geração de energia do inversor.

5. CONCLUSÕES

O que explica o porquê das modificações nos registradores que, em tese são de controle da bateria, não causarem impacto no seu comportamento é a incapacidade dos recursos de hardware e software do inversor em associar esses elementos de memória, com comandos que devam ser enviados até o dispositivo acumulador de energia. Portanto, faz-se necessário providenciar a substituição do modelo de inversor utilizado.

Outro aspecto que merece reflexão diz respeito à forma como os dados de monitoramento são acessados. Como dito na subseção 3.5, da forma como o SCADA foi construído, apenas máquinas computacionais presentes na mesma rede local tem a capacidade de acessar os dados presentes no software visualizador de dados Grafana. Uma possível proposta que tem o potencial de permitir acesso remoto à plataforma é a utilização de serviços desta natureza oferecidos pelo Centro de Processamento de Dados (CPD) da Universidade Federal de Santa Maria (UFSM).

Além do acesso remoto, uma funcionalidade que merece aperfeiçoamento é a forma como ocorre a execução das definições e requisições de controle. A otimização deve ser tal que permita a parametrização das variáveis aptas ao gerenciamento em tempo de execução, criando assim manipulação de dados de forma dinâmica. Uma possibilidade é a investigação de se criar *threads* as quais são executadas apenas quando informações ou sinais são recebidas de usuários ou do sistema operacional. Para isso seria necessário descobrir bibliotecas, ou pacotes python os quais permitam manipulação, escalonamento e envio de sinais para processos. Outra alternativa seria a criação de threads para receber pacotes MQTT relacionados ao encaminhamento de *setpoints* de controle ao Raspberry. Porém, por se tratar de threads há a grande possibilidade de também ser necessário a utilização de métodos capazes de realizarem chamadas de sistema ao SO do Raspberry.

Para tornar o SCADA humanamente amigável e largamente utilizável é imprescindível a existência de uma interface interativa, gráfica e intuitiva. Uma vez que é preciso levar em consideração que um eventual operador que possa fazer uso desta infraestrutura, a qual implementa gerenciamento de energia, muito provavelmente não terá disposição e intimidade para configurar o sistema via código fonte. Idealiza-se, portanto, a necessidade de construir uma interface gráfica a qual permite determinar, em tempo de execução das variáveis de controle e monitoramento, que o Raspberry irá interagir com o inversor.

Todos os componentes de hardware, protocolos, bibliotecas, pacotes e aplicações empregadas na implementação convergiram na direção de proporcionar o sucesso da operacionalização do supervisionamento.

Importante destacar que, apesar da inviabilidade de execução do método *peak shaving*, este trabalho foi capaz de construir toda a amostragem de dados de uma unidade de geração de energia fotovoltaica, além de sua respectiva visualização, empregando integralmente softwares e protocolos de comunicação *open-sources*.

AGRADECIMENTOS

Os autores agradecem ao Programa de P&D ANEEL, e a CEEE-D | Grupo Equatorial, pelo financiamento deste trabalho via Contrato Nº 5000003925.

REFERÊNCIAS

ANEEL - AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (2017). Nota Técnica n° 0056/2017-SRD/ANEEL, *ANEEL*, 28 páginas.

KIMAIYO, B. K e col. (2019). Effect of voltage unbalance on the power quality of three-phase grid-connected PV inverters, (2019), *International Engineer Electrical Congress.*

KARMIRIS. G.; TENGNÉR. T. (2013). Peak Shaving Control Method for Storage. *ABB AB*.

LEVRON, Y e col. (2012). Power systems' optimal peak-shaving applying secondary storage. *Electric Power Systems Research* Vol. 89.páginas 80-84.