# SANDIA REPORT

SAND2018-11692
Unlimited Release
Printed October, 2018

# Implementation of a Grid Connected Battery-Inverter Fleet Model

David Rosewater, Sigifredo Gonzalez

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology and Engineering Solutions of Sandia, LLC.



Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

| | |
|---|---|
| Telephone: | (865) 576-8401 |
| Facsimile: | (865) 576-5728 |
| E-Mail: | reports@adonis.osti.gov |
| Online ordering: | http://www.osti.gov/bridge |

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

| | |
|---|---|
| Telephone: | (800) 553-6847 |
| Facsimile: | (703) 605-6900 |
| E-Mail: | orders@ntis.fedworld.gov |
| Online ordering: | http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online |

SAND2018-11692
Unlimited Release
Printed October, 2018

# Implementation of a Grid Connected Battery-Inverter Fleet Model

David Rosewater, Sigifredo Gonzalez

**Abstract**

Batteries are designed to store electrical energy. The increasing variation in time value of energy has driven the use of batteries as controllable distributed energy resources (DER). This is enabled though low-cost power electronic inverters that are able to precisely control charge and discharge. This paper describes the software implementation of an open-source battery inverter fleet models in python. The Sandia BatteryInverterFleet class model can be used by scientists, researchers, and engineers to perform simulations of one or more fleets of similar battery-inverter systems connected to the grid. The program tracks the state-of-charge of the simulated batteries and ensures that they stay within their limits while responding to separately generated service requests to charge or discharge. This can be used to analyze control and coordination, placement and sizing, and many other problems associated with the integration of batteries on the power grid. The development of these models along with their python implementation was funded by the Grid Modernization Laboratory Consortium (GMLC) project 1.4.2. Definitions, Standards and Test Procedures for Grid Services from Devices.

# Contents

**Nomenclature** **6**

**1 BatteryInverterFleet Class Implementation** **7**

1.1 Introduction . . . . . . . . . . 7

1.2 Scope . . . . . . . . . . 7

1.3 Class Methods . . . . . . . . . . 8

1.4 State-of-Charge Models . . . . . . . . . . 10

1.4.1 Fleet Response Implementation . . . . . . . . . . 12

1.4.2 Model Validation . . . . . . . . . . 12

1.5 Autonomous Operation . . . . . . . . . . 13

1.5.1 Autonomous Operation Model Validation . . . . . . . . . . 15

1.6 Fleet Impact Metrics . . . . . . . . . . 16

1.7 Parameters . . . . . . . . . . 16

**References** **21**

# Nomenclature

**Ah** amp-hours

**BESS** battery energy storage system

**CRM** charge reservoir model

**DER** distributed energy resources

**EoL** end-of-life

**EPS** electric power system

**ERM** energy reservoir model

**GMLC** grid modernization laboratory consortium

**kW** kilowatt

**kWh** kilowatt-hours

**p.f.** power factor

**p.u.** per-unit

**SoC** state-of-charge

**SoH** state-of-health

# Chapter 1

# BatteryInverterFleet Class Implementation

This report introduces the python-based BatteryInverterFleet Class and describes its use and interface with other classes. This open-source software is implemented in the larger context of a transformation of the U.S. electric power system (EPS) which we will briefly discuss before the details of the model.

## 1.1 Introduction

The EPS is an interconnected marvel that goes far beyond just generation, load, and associated infrastructure. The traditional use of centralized power generation-to-load is something becoming more outdated and our aging infrastructure is being pushed beyond its original design. The proliferation of non-traditional loads and renewable power sources challenges grid operators to maintain the reliability and resiliency of the power grid. To help meet the increasing demands on the EPS, controllable distributed energy resources (DER) are being deployed to help create a smart, efficient, and reliable utility grid and the DER classes include responsive loads and appliances in buildings, several types energy storage devices, electric vehicle storage/chargers, smart photovoltaic inverters, and other DER that can enhance the reliability and stability of the bulk power system.

## 1.2 Scope

To address the challenges facing the EPS, the Grid Modernization Laboratory Consortium (GMLC) project 1.4.2., Definitions, Standards and Test Procedures for Grid Services from Devices, was scoped to enable and broaden the deployment of various types of DER classes that are able to ease the burden on the grid. The GMLC project accomplishes this goal through the development and validation of simulation models that enable a deeper understanding of the implementation of DER that provide more reliable and cost-effective means of producing, conveying, and storing power. To assess the capabilities for each of the DER to support the EPS, a performance metric must be utilized as a means of comparing the effectiveness of each type of DER to meet the needs of the grid. The development of a means to accurately and efficiently assess the capabilities of the devices is essential for deployment. By showcasing the performance of DER to supply grid services, utilities and consumers will gain confidence in the technologies.

A team comprised of national laboratories, each specializing in DER device classes and/or grid services, have developed modular simulation source code for evaluating the performance of DER devices to provide a variety of grid services, like ancillary services, peak load management, voltage regulation, frequency regulation, and wholesale market price response. The development to quantify the performance for the different device classes was accomplished by implementing a battery equivalent model for each of the device classes. The battery equivalent model provides a uniform means of representing the properties of any device class (and fleet of devices) as a virtual battery and allows a quantifiable means of assessing the performance of the technology. This provides a level-playing field for the various classes to be assessed for performance and how they are compared for dispatch purposes and provides a means to implement model-based performance metrics to assess grid services. The battery equivalent model implementation for a battery-based DER device class is presented in this paper. The device class is utilized as a fleet of devices to provide the impact on the associated grid service.

The scope of Sandia's work was to develop and validate the source code for modeling a fleet of battery-inverter devices. Also referred to as battery energy storage systems (BESS), these devices can act as loads, power sources, and even as transmission or distribution assets depending on how they are operated. Battery's store electrical energy as chemical and, when connected to the grid by power electronic based inverters, are extremely flexible in responding to power commands issued by utilities or markets. However, they are physically limited in how much energy they can supply or absorb, meaning that an operator must carefully decide when and how much to charge/discharge to maximize the benefits being supplied. These energy limits are the primary focus of our modeling efforts. Section 1.3 introduces the battery-inverter fleet model and its methods, Section 1.4 then covers the framework by which the model implements the energy limits of battery in simulation, Section 1.5 then explains how autonomous operation functions are implemented and validated, Section 1.6 outlines the battery degradation models used to calculate the fleet impact metrics, and lastly Section 1.7 lists all of the model parameters and explains their various effects on fleet simulations.

## 1.3 Class Methods

In this section, we introduce the BatteryInverterFleet class and explain its methods. The illustration in Fig. 1.1 shows the full list of available methods and roughly how they connect to each other. The GridInfo Class can be constructed by passing it the name of a csv file that contains time, frequency and voltage data in the correct format. The BatteryInverterFleet then takes two inputs to construct: a GridInfo object, and a string designating which state-of-charge (SoC) model to use (either ERM, or CRM). The '__init__' method opens the local config.ini file to extract the parameters for the appropriate SoC model and all other fleet parameters. The SoC models are described in Section 1.4. A full list of fleet model parameters can be found in Section 1.7.

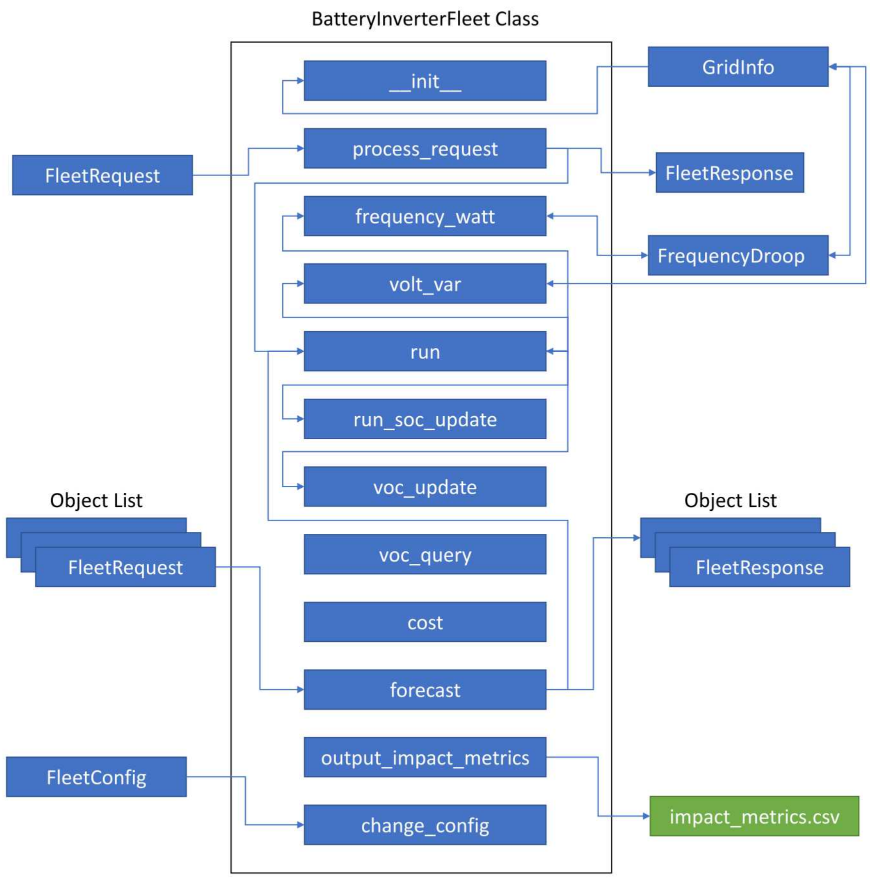



**Figure 1.1.** BatteryInverterFleet Class Method Diagram

Once a BatteryInverterFleet is constructed, its primary interface is through the 'process _request' method. This method accepts a 'fleet_request' object, and returns a 'fleet_response' object by calling the run method internally. Using the 'process_request' method in this way also updates the internal fleet states according to the model used. A secondary interface in through the 'forecast' method that accepts a list of 'fleet_request' objects that make up a proposed schedule. This method then returns a list of 'fleet_response' objects, corresponding to the forecast fleet response, without changing the internal states of the fleet.

The configuration file contains parameters to enable or disable autonomous operation. When autonomous operation is disabled, the BatteryInverterFleet will only respond to fleet requests. When enabled, the BatteryInverterFleet will still respond to fleet requests but these requests may be modified based on grid conditions. The BatteryInverterFleet has the option for two kinds of autonomous operation functions: frequency/watt and volt/var. These functions are described in Section 1.5. Note that the 'change_config' method will accept a FleetConfig object to programmatically change the parameters for autonomous operation.

The BatteryInverterFleet will keep track of how many cycles each device has undergone during operation and the corresponding state-of-health (SoH). The SoH, along with the incremental cost associated with degradation in SoH are saved to a csv file when the 'output_impact_metrics' method is called. These calculations are described in Section 1.6.

The 'cost' method is intended for use in schedule optimization. It accepts an initial SoC, a final SoC, and a time difference between them. It returns the power required to go from initial to final SoC in the allotted time and any associated costs. The cost is calculated the same as in the 'output_impact_metrics' method and is associated with the cycle-life and end-of-life costs. The 'cost' method also returns a Able variable that is 1 if the fleet is able to achieve the change in SoC in the allotted time and 0 if is is unable due to encountering one of its limits. This method can be used with dynamic programming to optimize a SoC/power schedule over a finite time horizon.

## 1.4 State-of-Charge Models

The battery-inverter model is implemented in two ways: energy reservoir model (ERM) and the charge reservoir model (CRM). Energy reservoir model (ERM) is a term for the class of state-of-charge (SoC) models that define capacity in units of energy (kWh). These models are highly efficient for simulation but can be inaccurate over large changes in SoC. ERM are widely used in research on the integration of energy storage with the grid. This includes studies using ERM that ignore efficiency losses [5, 6, 18], includes both charge and discharge efficiencies [8, 9, 12, 13, 21, 22, 25, 28], and those that include some form of self-discharge power [2, 8, 14, 15, 26]. The ERM implementation used in the battery-inverter model is shown in (1.1).

$$Q_{cap}\dot{\varsigma} = \eta_e \max(p_e, 0) + \min(p_e, 0) + p_{sd} \tag{1.1a}$$

$$\varsigma_{min} \leq \varsigma \leq \varsigma_{max} \tag{1.1b}$$

$$p_{min} \leq p_e \leq p_{max} \tag{1.1c}$$

where $\varsigma$ is the SoC, $p_e$ is the ac power (+ charge, - discharge), $Q_{cap}$ is the energy capacity, $\eta_e$ is the energy efficiency, $p_{sd}$ is the self-discharge power, and $\varsigma_{min}$, $\varsigma_{max}$, $p_{min}$, and $p_{max}$ are the SoC and power limits respectively.

Charge reservoir model (CRM) is a term for the class of BESS models that define capacity in units of charge (Ah). These models are less computationally efficient than ERM but have a higher potential for accuracy based on the increase number of represented battery dynamics. CRM are also widely used in scientific literature. This includes studies using CRM that ignore coulombic efficiency [4, 10, 16, 27], and those that include some form of self-discharge current [11, 23, 24]. The CRM implementation used in the battery-inverter model is shown in (1.2).

$$
\begin{align}
p_{dc} &= \phi_2 p_e^2 + \phi_1 p_e + \phi_0 \tag{1.2a}\\
p_{dc} &= i_{bat} v_{bat} \tag{1.2b}\\
v_{oc} &= f(\varsigma) \tag{1.2c}\\
\dot{v}_1 &= \frac{-1}{R_1 C_1} v_1 + \frac{1}{C_1} i_{bat} \tag{1.2d}\\
\dot{v}_2 &= \frac{-1}{R_2 C_2} v_2 + \frac{1}{C_2} i_{bat} \tag{1.2e}\\
v_{bat} &= v_{oc} + R_0 i_{bat} + v_1 + v_2 \tag{1.2f}\\
C_{cap} \dot{\varsigma} &= \eta_c \max(i_{bat}, 0) + \min(i_{bat}, 0) + i_{sd} \tag{1.2g}\\
\varsigma_{min} &\leq \varsigma \leq \varsigma_{max} \tag{1.2h}\\
p_{min} &\leq p_e \leq p_{max} \tag{1.2i}\\
i_{min} &\leq i_{bat} \leq i_{max} \tag{1.2j}\\
v_{min} &\leq v_{bat} \leq v_{max} \tag{1.2k}\\
& \tag{1.2l}
\end{align}
$$

where $\varsigma$ is the SoC, $p_e$ is the ac power, $p_{dc}$ is the dc power, $i_{bat}$ is the current provided to battery system (+ charge, - discharge), $v_{bat}$ is the battery terminal voltage, $v_{oc}$ is the battery open-circuit-voltage, $v_1$ is the first battery dynamic voltage, $v_2$ is the second battery dynamic voltage, and $\varsigma$ is the battery SoC. The parameters in (1.2) are as follows: $\phi_0$, $\phi_1$, and $\phi_2$ are the coefficients of a quadratic ac/dc conversion efficiency curve fit, $f(\varsigma)$ can be defined to be a 'Linear', 'Quadratic', 'Cubic', or 'CubicSpline' function (along with the associated coefficients), $R_0$, $R_1$, $C_1$, $R_2$, and $C_2$ are the equivalent circuit parameters, $C_{cap}$ is the charge capacity, $\eta_c$ is the coulombic efficiency, and $i_{sd}$ is the self-discharge current. The constraints, $\varsigma_{min}$, $\varsigma_{max}$, $p_{min}$, $p_{max}$, $i_{min}$, $i_{max}$, $v_{min}$, and $v_{max}$ are the SoC, power, current, and voltage limits respectively.

Both models have additional constraints to handle limits on power factor (p.f.), apparent power, and ramp rate. In the order, ramp-rate, absolute P or Q limits, apparent power limit, then p.f. limits, the algorithm first checks if any of these limits are violated and the, if they are, it changes the power requested of a given device to the closest value that satisfies the constraint. For the apparent power limit, the fleet model can be configured to either favor real power (P priority) or reactive power (not P priority, or Q priority). For P priority, if the apparent power limit is exceeded, the device will reduce its reactive power until the apparent power limit is satisfied. For Q priority, if the apparent power limit is exceeded, the device will reduce its real power until the apparent power limit is satisfied.

### 1.4.1 Fleet Response Implementation

When real and reactive power requests are sent to a fleet of two or more devices, the power request must be split between the devices such that the request is met without exceeding any of the device's limits. To accomplish this task, the battery-inverter fleet model divides the total request by the number of devices in the fleet and sends the scaled request to all available devices. Under certain conditions, some of the devices in a fleet may not be able to supply the full amount of power requested of them. When this happens, the algorithm designates these devices as 'not available,' and then divides the total power shortfall equally among the remaining 'available' devices. This process continues until either or the power request is met to within a threshold or all of the devices are 'not available.' This process works the same for both active and reactive power. Fig. 1.2 shows a flow chart for this algorithm.

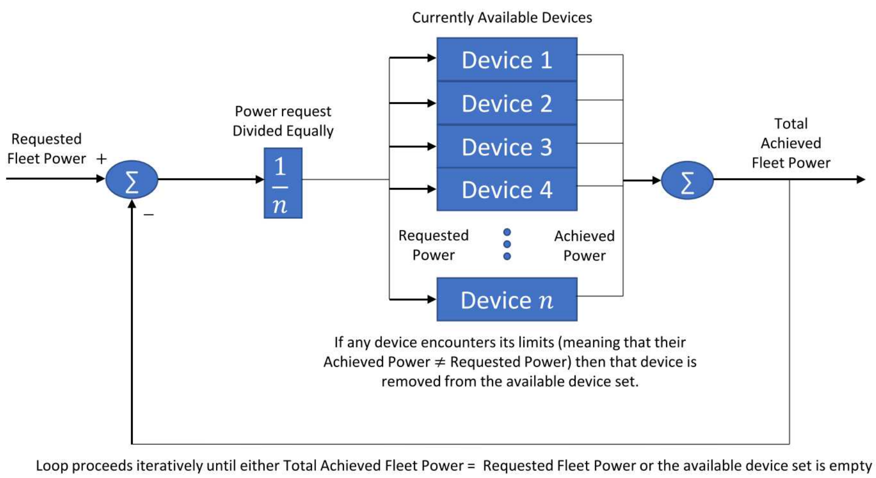
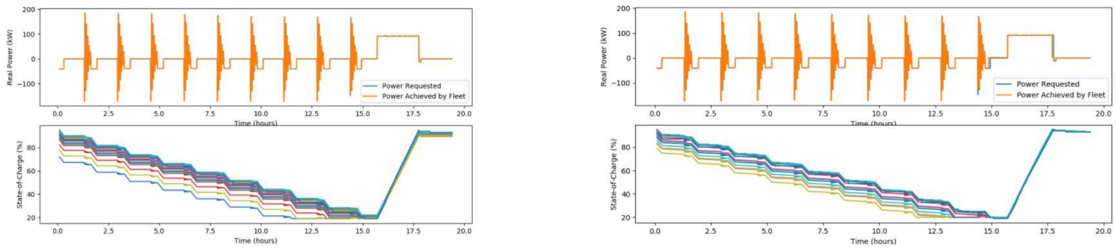



**Figure 1.2.** Fleet Response Recursive Implementation Diagram

### 1.4.2 Model Validation

The ERM and CRM discussed above were validated by simulating 30 devices with real test data, shown in Fig. 1.3(a) for ERM and Fig. 1.3(b) for CRM, and comparing their SoC trajectories to the SoC trajectory reported by the battery management system of the device under test. a standard normal distribution is used in this example to randomly initialize the SoC of each device in the simulated fleets. This simulation test performs three tasks. First, it demonstrates that both models produce results that closely follow the actual achieved performance of a real battery-inverter system. Second, it demonstrates what happens when some of the devices in the fleet expounder their low SoC limits. In this case the burden of supplying the requested power falls on the remaining available units as described above. Third it demonstrates the difference between the models in that the CRM fleet ends up curtailing power near the end of the test whereas the ERM based fleet does not. While potentially more accurate, the CRM takes considerably longer to simulate the test and hence may not be computationally efficient enough for very large power system simulations with hundreds or thousands of batteries.




(a) 30 ERM based battery-inverter devices

(b) 30 CRM based battery-inverter devices

**Figure 1.3.** Battery-Inverter Fleet Model Validation Simulation Test

## 1.5 Autonomous Operation

Autonomous responses to local grid conditions is configured to match the 2018 IEEE 1547 Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces [3]. The frequency/power function is implemented using the following two equations.

Operation for low-frequency conditions

$$p = \min_{f<60-db_{\mathrm{UF}}} \left\{ p_{\mathrm{pre}} + \frac{(60-db_{\mathrm{UF}})-f}{60\,k_{\mathrm{UF}}}; p_{\mathrm{avl}} \right\} \tag{1.3}$$

Operation for high-frequency conditions

$$p = \min_{f>60+db_{\mathrm{OF}}} \left\{ p_{\mathrm{pre}} + f - \frac{(60-db_{\mathrm{OF}})}{60\,k_{\mathrm{OF}}}; p_{\mathrm{min}} \right\} \tag{1.4}$$

where:

- $p$ is the active power output,104 in p.u. of the DER nameplate active power rating
- $f$ is the disturbed system frequency in Hz

- $p_{\rm avl}$ is the available active power, in p.u. of the DER rating
- $p_{\rm pre}$ is the pre-disturbance active power output, defined by the active power output at the point of time the frequency exceeds the deadband, in p.u. of the DER rating
- $p_{\rm min}$ is the minimum active power output due to DER prime mover constraints, in p.u. of the
- $db_{\rm OF}$ is a single-sided deadband value for high-frequency and low-frequency, respectively, in Hz
- $db_{\rm UF}$ is a single-sided deadband value for high-frequency and low-frequency, respectively, in Hz
- $k_{\rm OF}$ is the per-unit frequency change corresponding to 1 per-unit power output change (frequency droop), unitless
- $k_{\rm UF}$ is the per-unit frequency change corresponding to 1 per-unit power output change (frequency droop), unitless

The volt/var function is implemented through a linear interpolation between ($V_{\rm set}$,$Q_{\rm set}$) points supplied in the fleet configuration. While there are many ways to specify a volt/var function curve, (V,Q) points are compatible and consistent with he methods used in IEC61850-90-7 [1] and IEEE 1547 [3]. The plot in Fig. 1.4 shows an example of how to use (V,Q) points to define a volt/var curve.

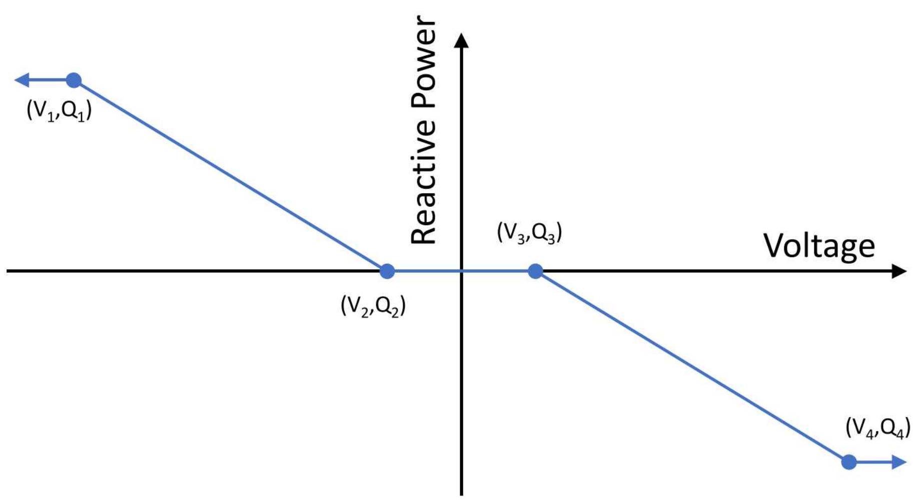
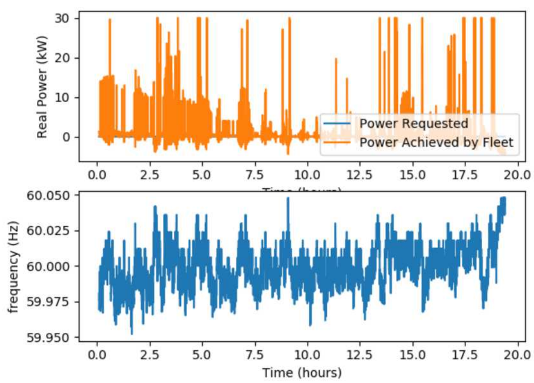



**Figure 1.4.** Autonomous volt/var function example curve

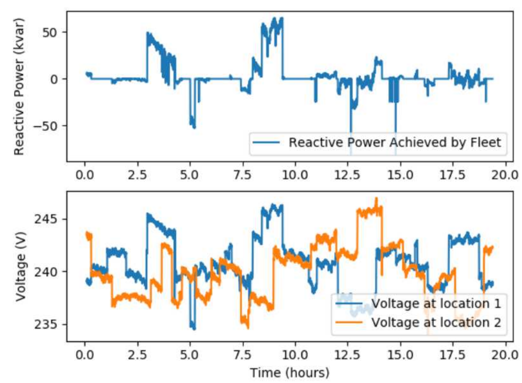
Whenever the a fleet is configured for autonomous operation, each device in a fleet will use the GridInfo class to look up the frequency and voltage of the location assigned to it and use these functions to modify their real and reactive power.

### 1.5.1 Autonomous Operation Model Validation

The Autonomous Operation Model discussed above was validated by running it on a frequency and voltage profile. The same frequency was used for all devices, whereas two locations with different voltage profiles were used. The default parameters had to be changed to be more sensitive as the voltage and frequency were otherwise close enough to nominal to stay within the deadbands for each respective function. The frequency and fleet real power response are shown in Fig. 1.5(a), and the voltages and fleet reactive power response are shown in Fig. 1.5(b).




(a) Frequency/watt functionality




(b) Volt/var functionality

## 1.6 Fleet Impact Metrics

This section describes how battery aging metrics are calculated. The simplest approach to calculating the rate of degradation in batteries ($\dot{\varrho}$) is to assume that it is a linear function of cycle-throughput. Under this assumption, degradation can be modeled as proportional to the absolute value of the battery power as shown in (1.5) when using the ERM [7,13,22] or to the absolute value of the battery current as shown in (1.6) when using the CRM [17,19,20].

$$\dot{\varrho} = \frac{|p_e|}{(1+\frac{1}{\eta_e})L_{\text{cyc}}Q_{cap}} \tag{1.5}$$

$$\dot{\varrho} = \frac{|i_{bat}|}{(1+\frac{1}{\eta_c})L_{\text{cyc}}C_{cap}} \tag{1.6}$$

where $p_e$ is ac real power, $i_{bat}$ is the battery current, $\eta_c$ is the coulombic efficiency, $L_{\text{cyc}}$ is the rated cycle-life to end-of-life EoL, $Q_{cap}$ is the energy capacity, and $C_{cap}$ is the charge capacity. Note that, when adding this up over a discrete simulation time, this form of degradation is equivalent to calculating the $\ell_1$ norm power as shown in (1.7) or the $\ell_1$ norm current as shown in (1.8). The regularization weight $\Pi_{\text{cyc}}$ has units of $\$/kW$ or \$/A depending on which equation it is in because of the units of the relevant decision variable.

$$f_b(\boldsymbol{p}_e) = \Pi_{\text{cyc}}\,||\boldsymbol{p}_e||_1 \tag{1.7a}$$

$$\text{where} \quad \Pi_{\text{cyc}} = \frac{\Delta t\,\mathrm{C}_{\text{EoL}}}{(1+\frac{1}{\eta_e})L_{\text{cyc}}Q_{cap}} \tag{1.7b}$$

$$f_b(\boldsymbol{i}_{bat}) = \Pi_{\text{cyc}}\,||\boldsymbol{i}_{bat}||_1 \tag{1.8a}$$

$$\text{where} \quad \Pi_{\text{cyc}} = \frac{\Delta t\,\mathrm{C}_{\text{EoL}}}{(1+\frac{1}{\eta_c})L_{\text{cyc}}C_{cap}} \tag{1.8b}$$

where $f_b$ is the fleet impact metric based on battery degradation, $\Delta t$ is the length of the simulation time step, and $\mathrm{C}_{\text{EoL}}$ is the anticipated EoL cost (although the capital cost of the BESS can also be used).

## 1.7 Parameters

Model parameters are implemented in the 'config.ini' file. Changing parameter values in this file will implement the changes when the BatteryInverterFleet is constructed. The general fleet parameters are listed in Table 1.1, the fleet configuration parameters including those for autonomous operation are listed in Table 1.2, the ERM specific parameters are listed in Table 1.3, the CRM specific parameters are listed in Tables 1.4 and 1.5, and the fleet impact metric parameters are listed in Table 1.6.

**Table 1.1.** General Parameters

| **Name** | **Default Value** | **Description** |
|---|---|---|
| Name | TestBat | Fleet Name |
| ModelType | ERM | Can be specified as ERM or CRM, defines which model is used to simulate the fleet. |
| MaxPowerCharge | 7 | Maximum charge power limit (kW) |
| MaxPowerDischarge | -7 | Maximum discharge power limit (kW) |
| MaxApparentPower | 7 | Maximum apparent power limit (kVA) |
| MinPF | 1 | Minimum power factor [0,1] |
| MaxSoC | 95 | Maximum state-of-charge limit |
| MinSoC | 19 | Minimum state-of-charge limit |
| MaxRampUp | 7.0 | Maximum ramp-rate for decreasing discharge or increasing charge power (kW/time-step) |
| MaxRampDown | -7.0 | Maximum ramp-rate for increasing discharge or decreasing charge power (kW/time-step) |
| NumberOfDevices | 30 | The number of battery-inverter devices in the fleet |
| Locations | 0,0,1 | location designations for each device (positive integers), 0 if not specified |
| FleetModelType | Standard Normal SoC Distribution | Defines the starting SoC for each device in the fleet, Options: Uniform, Standard Normal SoC Distribution. If Uniform is specified, all devices will start with the same SoC equal to the initial state 'soc' defined below. If Standard Normal SoC Distribution is specified then each starting device's SoC will be randomly assigned from a standard normal probability density function defined by a mean, defined by the initial state 'soc' defined below, and a standard deviation, specified by 'SOC_STD' defined below. |
| SOC_STD | 10 | Optional parameter for standard deviation in starting state of charge. |
| t | 0.0 | Initial time |
| soc | 95 | Initial state-of-charge, or the mean value of the distribution for initial states-of-charge for devices |

**Table 1.2.** Fleet Configuration Parameters

| Name | Default Value | Description |
|---|---|---|
| is_P_priority | True | When the apparent power limit is exceeded, this Boolean variable indicates if the real power is prioritized (True) or if reactive power is prioritized (False) |
| is_autonomous | True | This Boolean variable either enables (True) or disables (False) all autonomous operation functions |
| FW21_Enabled | True | Frequency/watt operation function enable (True) or disable (False) |
| db_UF | 0.0036 | single-sided deadband value for low-frequency, in Hz |
| db_OF | 0.0036 | single-sided deadband value for high-frequency, in Hz |
| k_UF | 0.005 | per-unit frequency change corresponding to 1 per-unit power output change (frequency droop), unitless |
| k_OF | 0.005 | per-unit frequency change corresponding to 1 per-unit power output change (frequency droop), unitless |
| P_avl | 1.0 | available active power, in p.u. of the DER rating |
| P_min | -1.0 | minimum active power output due to DER prime mover constraints, in p.u. of the DER rating |
| P_pre | 0.0 | |
| VV11_Enabled | True | Volt/var operation function enable (True) or disable (False) |
| Vset | 232.8, 237.6, 242.4, 247.2 | This parameter is a list of voltages corresponding to the list of vars in Qset. When enabled these points define the volt/var curve that the devices will follow when not supplied a reactive power request by a service (q=none). |
| Qset | 3.5, 0, 0, -3.5 | This parameter is a list of vars corresponding to the list of voltages in Vset. |

**Table 1.3.** ERM Specific Parameters

| Name | Default Value | Description |
|---|---|---|
| EnergyCapacity | 5.9441 | Energy capacity (kWh) |
| EnergyEfficiency | 0.6788 | Round trip energy efficiency (%) |
| SelfDischargePower | 0 | Self-discharge power (kW) |

**Table 1.4.** CRM Specific Parameters

| **Name** | **Default Value** | **Description** |
|---|---|---|
| InvName | TestInv | Inverter name |
| InvType | TestType | Inverter type (descriptive) |
| Coeff0 | -0.0721 | Quadratic coefficient for dc power to ac power conversion function |
| Coeff1 | 0.99107 | Linear coefficient for dc power to ac power conversion function |
| Coeff2 | -0.0151 | Offset coefficient for dc power to ac power conversion function |
| BatName | TestBat | Battery name |
| BatType | LiIon | Battery type (descriptive) |
| NCells | 14 | Number of cells in the battery string (for multiple parallel strings, multiply charge capacity 'ChargeCapacity' and divide ohmic resistance 'R0' by number of strings) |
| VOCModelType | Cubic | Open-circuit-voltage function type. Options: 'Linear', 'Quadratic', 'Cubic', or 'CubicSpline'. |

If 'Linear' is specified, will use VOC_Model_M and VOC_Model_b parameters to specify slope and intercept.

If 'Quadratic' is specified, will use VOC_Model_A, VOC_Model_B and VOC_Model_C parameters for the quadratic, linear, and offset coefficients respectively.

If 'Cubic' is specified, will use VOC_Model_A, VOC_Model_B, VOC_Model_C, and VOC_Model_D parameters for the cubic, quadratic, linear, and offset coefficients respectively.

If 'CubicSpline' is specified, will read in comma separated parameter lists in VOC_Model_A, VOC_Model_B, VOC_Model_C, and VOC_Model_D to specify the cubic, quadratic, linear, and offset coefficients within each SoC range specified in the comma separated list in VOC_Model_SOC_LIST. The 'CubicSpline' function is configured to work with the MATLAB 'spline' function meaning that the beginning of each SoC range is subtracted from the SoC value before the cubic function is applied.

| | | |
|---|---|---|
| VOC_Model_A | 0.962857 | Cubic coefficient for open circuit voltage |
| VOC_Model_B | -0.717143 | Quadratic coefficient for open circuit voltage |
| VOC_Model_C | 0.41 | Linear coefficient for open circuit voltage |
| VOC_Model_D | 3.445 | Offset coefficient for open circuit voltage |

**Table 1.5.** CRM Specific Parameters (continued)

| **Name** | **Default Value** | **Description** |
|---|---|---|
| MaxCurrentCharge | 150.0 | Maximum charge current limit (A) |
| MaxCurrentDischarge | -150.0 | Maximum discharge current limit (A) |
| MaxVoltage | 4.2 | Maximum battery cell voltage limit (V) |
| MinVoltage | 3.3 | Minimum battery cell voltage limit (V) |
| ChargeCapacity | 135.2366 | Charge capacity (Ah) |
| CoulombicEfficiency | 0.9462 | Coulombic efficiency (%) |
| SelfDischargeCurrent | 0 | Self-discharge current (A) |
| R0 | 0.001096 | Per-cell ohmic (dc) resistance ($\Omega$) |
| R1* | 999999999.0 | Per-cell resistance from diffusion time-constant ($\Omega$) |
| R2* | 999999999.0 | Per-cell capacitance from diffusion time-constant ($F$) |
| C1* | 999999999.0 | Per-cell resistance from cell capacitance time-constant ($\Omega$) |
| C2* | 999999999.0 | Per-cell capacitance from cell capacitance time-constant ($F$) |

* setting these values to be very large effectively ignores the dynamic components in the simulation model.

**Table 1.6.** Fleet Impact Metric Parameters

| **Name** | **Default Value** | **Description** |
|---|---|---|
| eol_cost | 6000 | Anticipated cost to be incurred at end-of-life. Should include both battery replaceent/refurbishment costs as well as any revenue from second-life use resale ($) |
| cycle_life | 10000 | Cycle-life, number of cycles from SoH = 100% to SoH = 0% (#) |
| soh | 100 | Initial state-of-health (SoH) (%) |

# References

[1] Communication networks and systems for power utility automation - part 90-7: Object models for power converters in distributed energy resources (der) systems. *IEC TR 61850-90-7*, 2013.

[2] Gridlab-d wiki: Battery model. Online, 2018.

[3] Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces. *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pages 1–138, April 2018.

[4] H. H. Abdeltawab and Y. A. R. I. Mohamed. Market-oriented energy management of a hybrid wind-battery energy storage system via model predictive control with constraint optimizer. *IEEE Transactions on Industrial Electronics*, 62(11):6658–6670, Nov 2015.

[5] Jonathan Donadee and Marija D. Ilic. Stochastic optimization of grid to vehicle frequency regulation capacity bids. *IEEE Transactions on Smart Grid*, 5(2):1061–1069, 2014.

[6] Jonathan Donadee and Jianhui Wang. AGC signal modeling for energy storage operations. *IEEE Transactions on Power Systems*, 29(5):2567–2568, 2014.

[7] S. Ebbesen, P. Elbert, and L. Guzzella. Battery state-of-health perceptive energy management for hybrid electric vehicles. *IEEE Transactions on Vehicular Technology*, 61(7):2893–2900, Sept 2012.

[8] EPRI. *OpenDSS STORAGE Element and STORAGECONTROLLER Element*, March 2011.

[9] T. Erseghe, A. Zanella, and C. G. Codemo. Optimal and compact control policies for energy storage units with single and multiple batteries. *IEEE Transactions on Smart Grid*, 5(3):1308–1317, May 2014.

[10] Xue Feng, H. B. Gooi, and S. X. Chen. Hybrid energy storage with multimode fuzzy power allocator for pv systems. *IEEE Transactions on Sustainable Energy*, 5(2):389–397, 2014.

[11] Anthony M. Gee, Francis V. P. Robinson, and Roderick W. Dunn. Analysis of battery lifetime extension in a small-scale wind-energy system using supercapacitors. *IEEE Transactions on Energy Conversion*, 28(1):24–33, mar 2013.

[12] Y. Huang, S. Mao, and R. M. Nelms. Adaptive electricity scheduling in microgrids. *IEEE Transactions on Smart Grid*, 5(1):270–281, Jan 2014.

[13] N. Jayasekara, M. A. S. Masoum, and P. J. Wolfs. Optimal operation of distributed energy storage systems to improve distribution network load and generation hosting capability. *IEEE Transactions on Sustainable Energy*, 7(1):250–261, Jan 2016.

[14] Leonardo H. Macedo, John F. Franco, Marcos J. Rider, and Ruben Romero. Optimal operation of distribution networks considering energy storage devices. *IEEE Transactions on Smart Grid*, 6(6):2825–2836, 2015.

[15] P. Malysz, S. Sirouspour, and A. Emadi. An optimal energy storage control strategy for grid-connected microgrids. *IEEE Transactions on Smart Grid*, 5(4):1785–1796, July 2014.

[16] Modeling and Validation Work Group. Wecc battery storage dynamic modeling guideline. Technical report, Western Electricity Coordinating Council, 2016.

[17] S. J. Moura, J. L. Stein, and H. K. Fathy. Battery-health conscious power management in plug-in hybrid electric vehicles via electrochemical modeling and stochastic control. *IEEE Transactions on Control Systems Technology*, 21(3):679–694, May 2013.

[18] E. Perez, H. Beltran, N. Aparicio, and P. Rodriguez. Predictive power control for pv plants with energy storage. *IEEE Transactions on Sustainable Energy*, 4(2):482–490, April 2013.

[19] H. E. Perez, X. Hu, S. Dey, and S. J. Moura. Optimal charging of li-ion batteries with coupled electro-thermal-aging dynamics. *IEEE Transactions on Vehicular Technology*, 66(9):7761–7770, Sept 2017.

[20] Y Riffonneau, S Bacha, F Barruel, and S Ploix. Optimal power flow management for grid connected PV systems with batteries. *IEEE Transactions on Sustainable Energy*, 2(3):309–320, jul 2011.

[21] Carlos Suazo-Martinez, Eduardo Pereira-Bonvallet, Rodrigo Palma-Behnke, and Xiao-Ping Zhang. Impacts of energy storage on short term operation planning under centralized spot markets. *IEEE Transactions on Smart Grid*, 5(2):1110–1118, 2014.

[22] J. Tant, F. Geth, D. Six, P. Tant, and J. Driesen. Multiobjective battery storage to improve pv integration in residential distribution grids. *IEEE Transactions on Sustainable Energy*, 4(1):182–191, Jan 2013.

[23] Sercan Teleke, Mesut E. Baran, Subhashish Bhattacharya, and Alex Q. Huang. Optimal control of battery energy storage for wind farm dispatching. *IEEE Transactions on Energy Conversion*, 25(3):787–794, 2010.

[24] G. Wang, M. Ciobotaru, and V. G. Agelidis. Power management for improved dispatch of utility-scale pv plants. *IEEE Transactions on Power Systems*, 31(3):2297–2306, May 2016.

[25] Trudie Wang, Haresh Kamath, and Steve Willard. Control and optimization of grid-tied photovoltaic storage systems using model predictive control. *IEEE Transactions On Smart Grid*, 5(2):1010–1017, 2014.

[26] Yunfeng Wen, Chuangxin Guo, Hrvoje Pandzic, and Daniel S. Kirschen. Enhanced security-constrained unit commitment with emerging utility-scale energy storage. *IEEE Transactions on Power Systems*, 31(1):652–662, 2016.

[27] Liu Xiong, Wang Peng, and Loh Poh Chiang. A Hybrid AC/DC Microgrid and Its Coordination Control. *IEEE Transactions on Smart Grid*, 2(2):278–286, 2011.

[28] Pingliang Zeng, Zhi Wu, Xiao ping Zhang, Caihao Liang, and Yantao Zhang. Model predictive control for energy storage systems in a network with high penetration of renewable energy and limited export capacity. Power Systems Computation Conference, Wroclaw, Poland, 2014.

# DISTRIBUTION:

| | | | |
|---|---|---|---|
| 1 | Robert G. Pratt, Robert.Pratt@pnnl.gov (electronic copy) | | |
| 1 | MS 1108 | Babu Chalamala, 8811 | |
| 1 | MS 1033 | Abraham Ellis, 8812 | |
| | | , | |
| 1 | MS 0899 | Technical Library, 9536 (electronic copy) | |

v1.40

Sandia National Laboratories