StorageController Element

Celso Rocha, Jouni Peppanen, Paulo Radatz, Matthew Rylander, Roger Dugan

1 Purpose

This technical note describes the modelling and operation of StorageController element, including several demonstration examples.

2 Why?

OpenDSS StorageController element has passed through a major update in 2018/2019. This technical note has been elaborated to detail some of the new features, but also to expand the previous StorageController documentation [1] with a comprehensive description of the model of the controller. The new features are:

- Addition of *I-PeakShave*, *I-PeakShaveLow* and *PeakShaveLow* dispatch modes;
- Possibility to operate simultaneously with InvControl control element controlling the same Storage elements;
- Addition of $DispFactor$ property to aid in convergence;
- Possibility to select the regulated measure (power or current) from a specific phase or a combination between different phases of the monitored element through the property $MonPhase$;
- Possibility to specify the deadband as an absolute value. Useful when the desired target value is low. See properties $kWBand$ and $kWBandLow$;
- Improved control actions description in Event Log report for better interpretation of operation within each control loop iteration;
- Full compatibility with recently updated Storage element;

3 Brief Introduction

The StorageController element is designed to control a fleet of Storage elements, which can be as small as composed by a single element, and perform such tasks as dispatching the Storage elements to follow load. For example, it can be difficult to perform load following based on local intelligence only at each Storage element location because the local load may not reflect the load, e.g. at the substation, for example, that needs capacity relief. The StorageController was created to perform such tasks as load following and peakshaving.

In this document, the general model is first presented, and then its operation in different dispatch modes is detailed. After that, examples for each dispatch mode are given. Finally, a complete table with all available properties of this element is given at the end of this document.

4 Modelling

The model concept is depicted in Figure 1. The “fleet” may consist of one or more Storage devices.

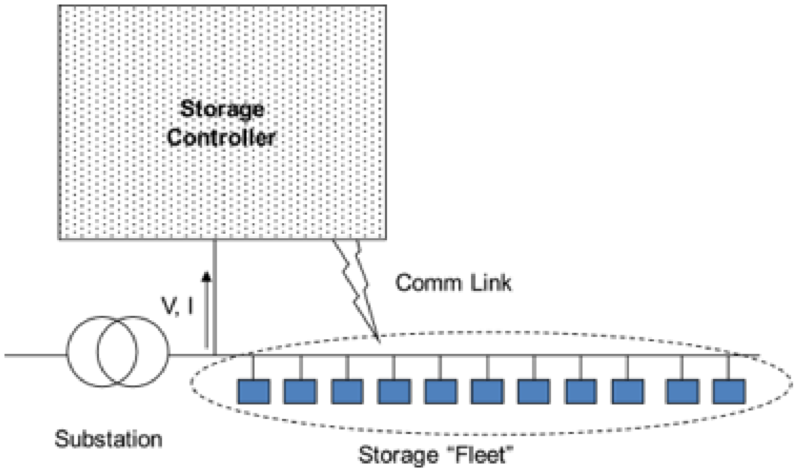


Figure 1: StorageController concept

Like other controllers in OpenDSS, the user defines a StorageController as monitoring a terminal of one of the current-carrying devices in the circuit. This is usually a Line or Transformer object, but could be any circuit element including a Generator object representing, for example, a solar PV installation with varying power output.

Another key property is *ElementList*. This is a list of Storage-class elements controlled by this controller. If the list is not defined, all Storage elements in the circuit are assumed to be controlled by this controller. You may list the elements as an array of names or you may use the “file=” syntax within the array delimiters:

ElementList = [MyElement1, MyElement2, MyElement3,...] ElementList = [File=*listfile.txt*]

Where *listfile.txt* is a file containing the names of the Storage elements one to a line. (Only the name of the element should be given; the class name “Storage” is assumed.)

You will get an error message if more than one StorageController attempts to control the same Storage device.

5 Dispatch Modes

There are currently 7 discharge modes and 4 charging modes available, from which two modes, Loadshape and Time, can be used to manage both discharging and charging of the fleet. The controller can operate simultaneously with any combination of a discharge only and a charge only mode. Each mode can be further divided as “powerflow-driven” or “time-driven”. Table 1 summarizes all dispatch modes available and their classification.

A “power flow-driven” mode has its operation based on an electrical measure resulting from a power flow solution, like in Figure 1. The measured quantity (power or current) is compared to a target, and the controller requests each element of the fleet to dispatch based on the difference between the measurement and the target values.

All "power flow-driven" modes employ a deadband controller concept. A target value is set ($kWTarget$, $kWTargetLow$) and a band around the target is specified. When the kW in the monitored branch exceeds the $kWTarget$ value plus half the bandwidth, the fleet is dispatched to bring the power back to the center of the band. The dispatch follows the regulated measure until the storage dispatch requested to a specific storage element wants to go negative, at which point it is set to idling state. The bandwidth is set either as a percentage of the respective target values through properties $\%kWBand$ and $\%kWBandLow$ (defaults to 2%) or as absolute values, through $kWBand$ and $kWBandLow$ properties.

Each Storage element may have a different weight, if you want the StorageController to dispatch some elements at a higher rate than others. The weights default to 1.0, but may be defined by an array.

A "time-driven" mode has its operation directly or indirectly based on time. The deadband concept does not apply to these modes. The dispatch rate is pre-defined based on either a dispatch curve or a fixed rate ($\%RatekW$ and $\%RateCharge$ properties).

Table 1: Summary of dispatch modes for StorageController element

Requested State	**Driver**	**Mode**
Discharging Only	Power Flow	PeakShave
		I-PeakShave
		Follow
		Support
	Time	Schedule
Charging Only	Power Flow	PeakShaveLow
		I-PeakShaveLow
Both	Time	Time
		LoadShape

5.1 Discharging Only

5.1.1 PeakShave

PeakShave is the default discharge mode. The control attempts to discharge the storage fleet to keep power in the monitored element within a bandwidth (specified by either $\%kWBand$ or $kWBand$ properties) of, or below, the kW value specified by the $kWTarget$ property. The controller will request as much power as necessary, limited by the inverter rated kW ($kWrated$ property) of each element of the fleet, to accomplish this. The fleet turns itself off when it runs out of stored energy. The basic concept is to follow the load by keeping the load in the monitored terminal at or below the value. Figure 2 illustrates the operation of the controller in this mode assuming that the fleet has enough energy and power capacity to completely shave the peak. The monitored branch would typically be a line or transformer that might be considered overloaded at the target value. Note that it is possible to change $kWtarget$ on the fly for real time control simulation.

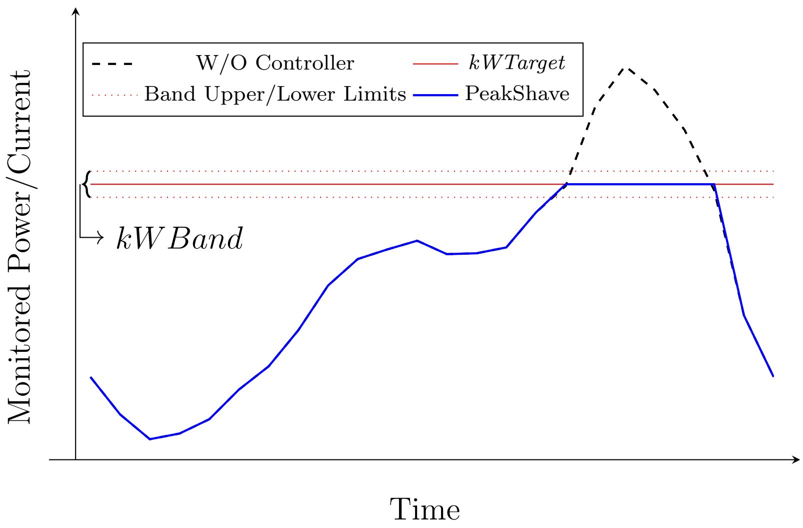


Figure 2: Operation in PeakShave/I-PeakShave Mode

5.1.2 I-PeakShave

The operation principle in I-PeakShave mode is similar to the PeakShave mode, except that the control attempts to discharge storage to keep **current** in the monitored element below the target given in k-amps (thousands of amps). When this control mode is active, the property *kWTarget* will be expressed in k-amps and all the other parameters will be adjusted to match the amps (current) control criteria.

5.1.3 Follow

The control is triggered by time (see *TimeDischargeTrigger* property) and resets the *kWTarget* property value to the present monitored element power. In this mode, the fleet tends to be dispatched more often than in PeakShave mode by being dispatched near the peak, even if the daily peak itself significantly varies throughout the simulation period, for instance, throughout the year (in simple PeakShave mode, the fleet may only be dispatched a few days per year). Even though it is triggered by time, it has been classified as a “power flow-driven” mode since, once triggered, and the target value updated, it operates exactly the same as the PeakShaveMode, i.e, it attempts to discharge the fleet to keep power in the monitored element no greater than the new target. *kWThreshold* property can be used in this mode to prevent the fleet to discharge on days when the load is less than the specified value. Figure 3 illustrates the operation of the controller in this mode.

5.1.4 Support

This is essentially the opposite of PeakShave mode. The fleet is dispatched to keep the power in the monitored terminal at or above*kWTarget*. The typical application would be to support renewable generation when power output falls off to events like cloud transients. The power monitored is assumed normally negative (see Example 7.3). Figure 4 illustrates the operation of the controller in this mode.

5.1.5 Schedule

In Schedule mode, a trapezoidal-shaped discharge schedule is specified through *Tup* (up ramp duration), *TFlat* (flat duration), and *Tdn* (down ramp duration) properties, as shown in Figure 5. The

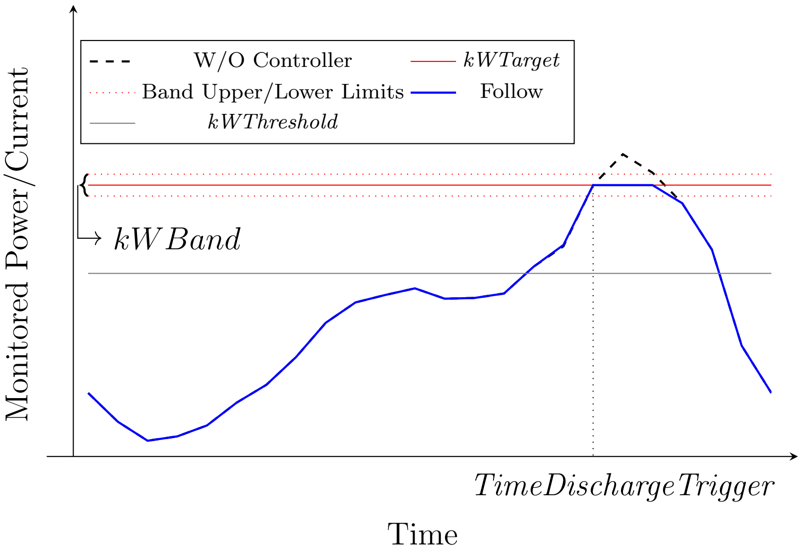


Figure 3: Operation in Follow Mode

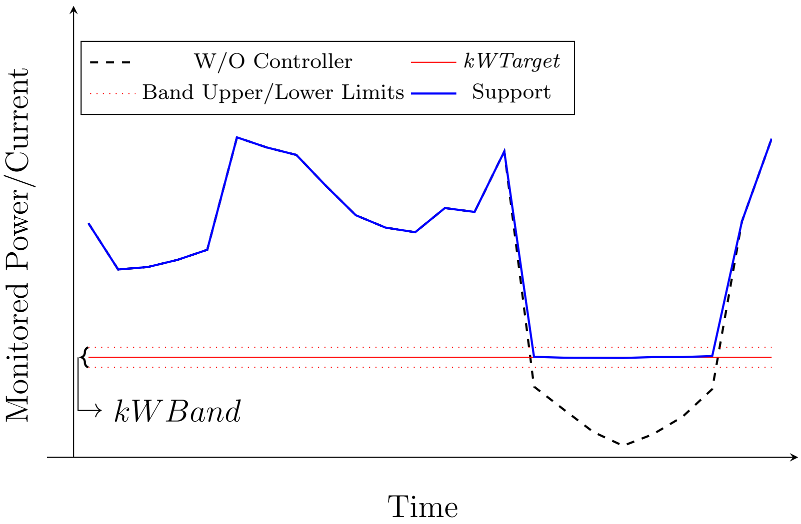


Figure 4: Operation in Support Mode

schedule start time is set by $TimeDischargeTrigger$ and the rate of discharge for the flat part is determined by $\%RatekW$.

In this mode, whenever the simulation time advances and it is within the trapezoidal-shaped discharged schedule, a new request is sent by the controller to each storage element in the fleet.

5.2 Charging Only

5.2.1 PeakShaveLow

In PeakShaveLow mode, the fleet is set to charge when the power at a monitored element is below a specified target ($kWTargetLow$). The controller will request as much power as necessary, limited by the inverter rated kW ($kWrated$ property) of each element of the fleet, to keep the power within the bandwith ($\%kWBandLow$ or $kWBandLow$) around $kWTargetLow$. Figure 6 illustrates the operation of the controller in this mode assuming that the fleet has enough energy and power capacity to completely fill the valley below $kWTargetLow$.

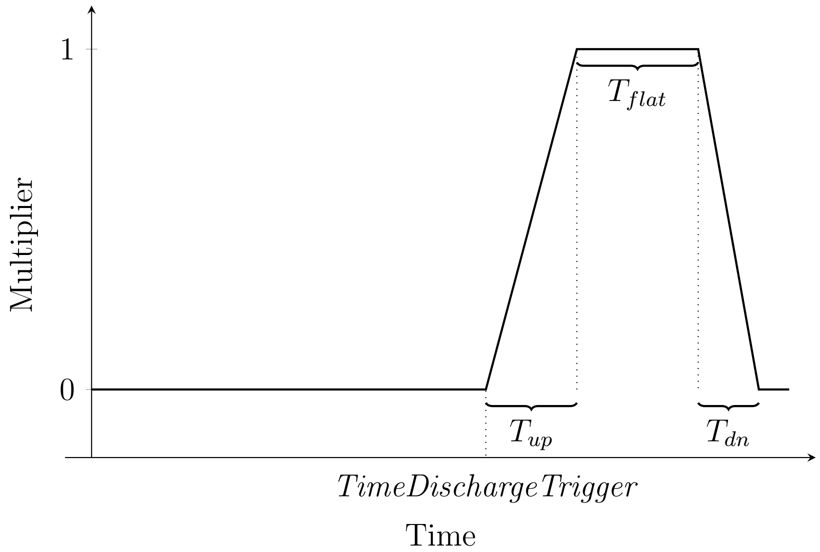


Figure 5: Dispatch shape as defined by Tup, $Tflat$, Tdn and $timeDischargeTrigger$ properties in Schedule Mode

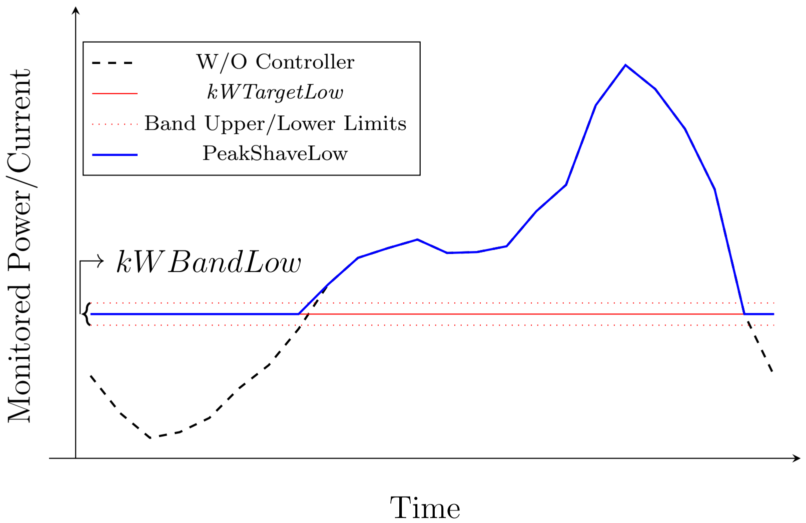


Figure 6: Operation in PeakShaveLow/I-PeakShaveLow Mode

5.2.2 I-PeakShaveLow

The operation principle in I-PeakShaveLow mode is similar to the PeakShaveLow mode, except that the control sets the fleet to charge when the **current** (Amps) at a monitored element is below a specified amps target ($kWTargetLow$). The storage will charge as much power as necessary, limited by the inverter rated kW, to keep the amps within the deadband around $kWTargetLow$. When this control mode is active, the property $kWTargetLow$ will be expressed in k- amps and all the other parameters will be adjusted to match the amps (current) control criteria.

5.3 Discharging and Charging

5.3.1 LoadShape

In this mode both discharging and charging precisely follow a per-unit LoadShape curve. The controller sends a new request to each storage element of the fleet whenever there has been a change in the curve value. When the value is positive all units are discharged at the rate of the per-unit value of the curve. When the value is zero, all units are set to idling. When the value is negative, all units are set to charging at the rate of the per unit value of the curve. This mode has the same operation principle as the "Follow" self-dispatch mode for storage elements [2].

5.3.2 Time

In Time mode all storage elements are set to discharge when in the course of simulation the time of day passes the specified hour of day by the *TimeDisChargeTrigger* property (hour is a decimal value, e.g., 10.5 = 1030). The storage elements go into idling mode when their storage declines to the reserve value. The discharging rate is set by the *%RatekW* property.

For charging, it operates similarly, but triggered by *TimeChargeTrigger*. The storage elements switch to the idling mode when the storage is completely charged or the mode is changed by the controller. This is the default mode for charging. The charging rate is set by the *%RateCharge* property.

The algorithm for this mode is quite simple. Whenever the simulation time instant t passes either $TimeChargeTrigger$ or $TimeDischargeTrigger$, each element of the fleet is set to charge or discharge, respectively, by the specified rate if it has enough energy capacity left. It is important to note that the controller sends a request to the fleet at the specific time instant t only and, if nothing else changes the state of a particular element of the fleet (for instance, a manual state request or rate of dispatch change through *state* and *kW* storage element properties), it will keep following the request sent at t, until its fully charged or depleted down to $\%reserve$.

6 Operation in Power-Flow Driven Modes

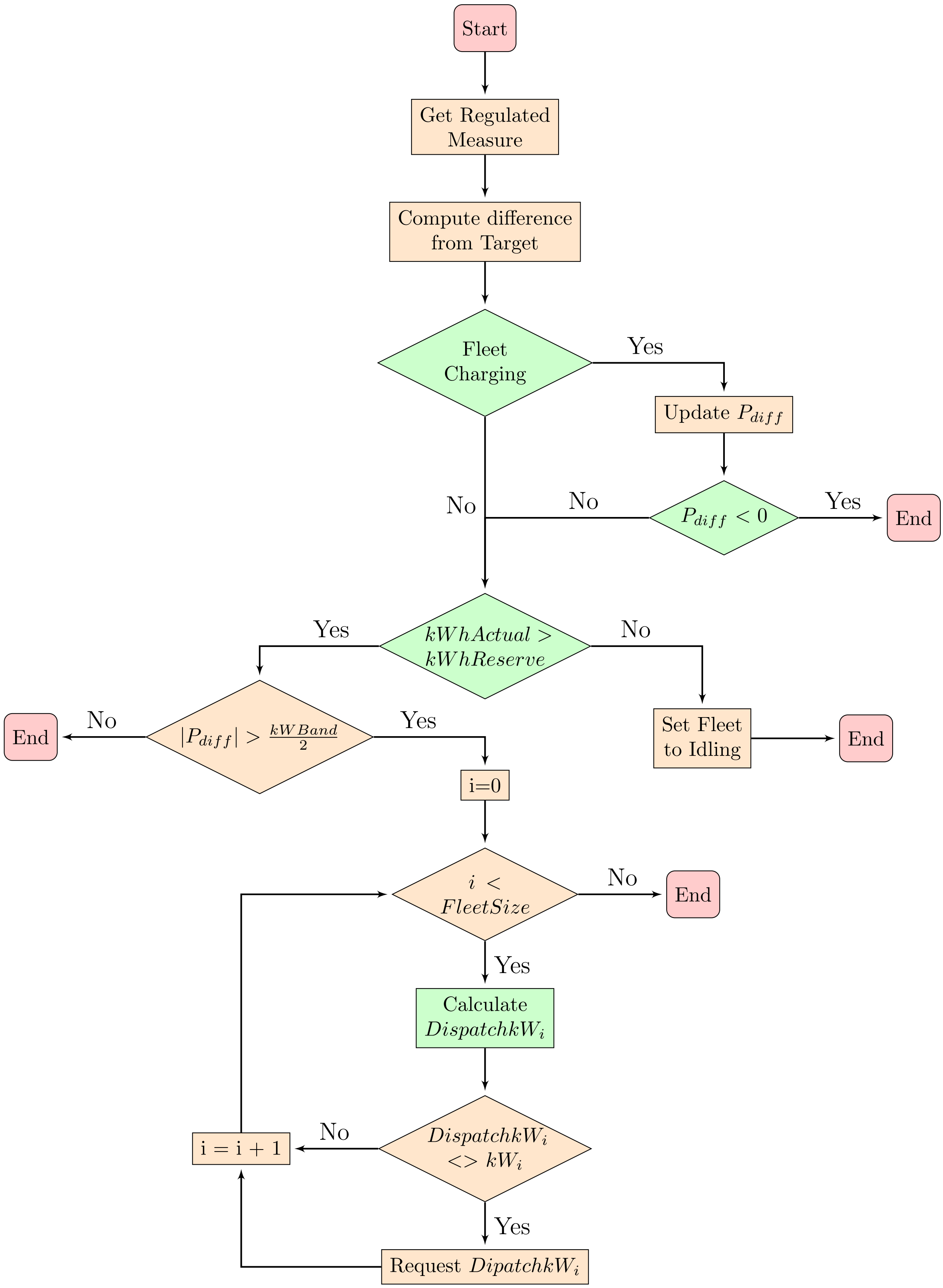
The controller operation is essentially the same in all the different power flow-driven modes. The operation is illustrated in Figure 7, where the blocks in green work differently for charging and discharging modes. This algorithm is executed every control iteration (see [3], [4] and [5] for an overview of how OpenDSS handles control elements). It is important to note that there is no time delay involved in the operation of this control element, which could represent for instance, communication delays [1]. This means that the actions pushed to the control queue are executed at the same simulation time in all control modes (static, time, etc), see the OpenDSS manual [4] for more information on different control modes. Its convergence criteria is satisfied when no other action is pushed to the control queue, which is determined by not setting either power or state changes to any storage element of the fleet.

Each block is detailed below:

1. Get Regulated Measure: The regulated measure, P_{reg}, for power, or I_{reg}, for current, is calculated based on the power/current flowing into the terminal $Terminal$ of the monitored element

[1]In particular, StorageController is assumed to operate as fast as InvControl element, i.e., when operating together, their control actions are executed simultaneously.

Figure 7: Algorithm for "power flow-driven" dispatch modes run at every control iteration and assuming discharging modes (in charging modes, the implementation differs for the green blocks).



Element and the monitored phase or combination of phases specified in *MonPhase* from the most recent successful power flow solution;

2. Compute difference from Target: The power/current difference between the regulated measure and the target is the error that the controller tries to reduce to a value within *kWBand*. Its calculation depends on the selected dispatch mode and it is direction dependent[2] as explained below.

   (a) PeakShave (PeakShaveLow) and Follow: $P_{diff} = P_{reg} - kWTarget(kWTargetLow)$

   Note that the regulated power is assumed to be normally positive, i.e, a P_{diff} greater than $kWBand/2$ leads the controller to request the fleet to discharge an amount of P_{diff} in order to limit P_{reg} to $kWTarget$, for discharging modes. For charging modes, a negative P_{diff} leads the controller to request the fleet to charge the same amount.

   (b) I-PeakShave (I-PeakShaveLow): $P_{diff} = I_{reg} - kWTarget(kWTargetLow)$

   Same idea as for PeakShave and Follow modes, except that all variables represent current magnitude rather than active power.

   (c) Support: $P_{diff} = P_{reg} + kWTarget$

   In support mode, as commented in 5.1.4, the regulated power is assumed to be normally negative. Thus, P_{diff} is calculated based on the sum of P_{reg} and the target, such that a regulated power less than $kWTarget$ leads to a positive P_{diff}, which leads the controller to discharge the fleet.

   After calculating P_{diff}, its value is stored in StorageController's *kWneed* read-only property, which is accessible to the user. See Table 3 for a list of all properties available.

3. Fleet Charging (for Discharging Modes): This block is necessary to make sure that the controller will not inadvertently try to dispatch the fleet because of the increase in P_{reg}/I_{reg} due to the power it consumes while in charging state. Thus, the power being consumed by the fleet is discounted from P_{diff} and it is updated as shown below:

$$P_{diff} = P_{diff} + FleetkW$$

   where *FleetkW* is the total power currently being charged/discharged by the fleet, following the generator convention. If the updated P_{diff} is negative, the controller does not send any dispatch request to the fleet. In other words, any overloading due to charging is ignored by the controller, which may lead to a situation in which P_{reg} is greater than $kWTarget$ by a value greater than $kWBand$ and the controller will still not try to dispatch the fleet. The purpose of this mechanism is to make sure the controller will not affect the charging of the fleet, otherwise there might not be enough energy stored to discharge the fleet when needed. Nonetheless, the total power consumed by the fleet during charging can be controlled by the available charging dispatch modes and configured such that *kWtarget* is not surpassed during charging instants.

   For charging modes, the condition is changed to check whether the fleet state is Discharging. The basic idea is the same. In this case, to avoid P_{diff} to be negative due to the decrease in

[2]The appropriate terminal, *terminal*, of the monitored element, *element*, must be selected according to the dispatch mode and based on the expected power flow direction for the monitored element. Power flowing into the monitored terminal defines the sign of P_{reg}.

P_{reg} while discharging, because if the fleet discharges ($FleetkW > 0$), P_{reg} tends to decrease and may inadvertently trigger the controller to charge the fleet. Thus, for charging modes, if the updated P_{diff} is positive, the controller does not send any dispatch request to the fleet.

4. $kWhActual > kWhReserve$ (for Discharging Modes): For discharging modes, if the fleet is already in discharging state or in case it is not but P_{diff} is positive, the controller might send new dispatch values to each element of the fleet if some conditions are met. First, the total energy stored by the fleet, $kWhActual$ must be greater than the total reserve energy, $kWhReserve$, otherwise the fleet is just set to idling state. In case the current fleet state is not idling and this condition is not satified, an action is pushed to the control queue, forcing a new iteration of the controller.

   For charging modes, if the fleet is already in charging state or in case it is not but P_{diff} is negative, the first condition modifies to check whether there is enough capacity left to charge the fleet, i.e., $kWhActual$ must be less than the total energy capacity, $kWhTotal$, otherwise the fleet is just set to idling state.

5. $|P_{diff}| > \frac{kWBand}{2}$: The second condition verifies if the absolute value of P_{diff} is out of the deadband or not. This is where the controller algorithm usually ends in its last control iteration if the available power and energy of the fleet are enough to perform the requested action (fully shave the peak, fully fill the valley, fully support a generator, etc).

6. Calculate $DispatchkW_i$: If P_{diff} is not within the band, the controller iterates over all storage elements of the fleet and calculates an incremental power, ΔP_i to be requested to each element of the fleet, i, according to

$$\Delta P_i = kWNeed * \frac{Weight_i}{\sum_{i=0}^{FleetSize} Weight_i} * DispFactor \tag{1}$$

   where $FleetSize$ corresponds to the total number of storage under control of this control element, $Weight_i$ corresponds to the weight given to a particular storage through $Weights$ property and $DispFactor$ is a property added to aid in eventual convergence issues that might happen to the controller. It reduces the incremental power requested to each storage element of the fleet in order to slowly move P_{reg}/I_{reg} towards $kWTarget$. It is suggested to not play with this property unless control loop convergence issues have been identified as being caused by the StorageController element.

   For I-PeakShave and I-PeakShaveLow modes, $kWNeed$ is converted from current to power based on the number of phases and the rated voltage of each element, i, of the fleet.

   Then, the incremental power is added to the nominal power currently being dispatched by the storage element i, $PresentkW_i$, which already satisfies any constraints in the nominal active output power due to the inverter capability curve. See [6]. However, the final requested dispatch is limited to the rated active power of the inverter, $kWRating$, which can be written as

$$DispatchkW_i = Min\{kWRating_i,\ PresentkW_i + \Delta P_i\} \tag{2}$$

   for discharging modes, and

$$DispatchkW_i = Max\{-kWRating_i,\ PresentkW_i + \Delta P_i\} \tag{3}$$

for charging modes, if $PresentkW_i + \Delta P_i$ is negative. Note that it assumes that the inverter active power rating is the same for charging and discharging.

If in equations (2) and (3) the resulting power is negative and positive, respectively, the storage element i is just set to idling state instead.

7. $DispatchkW_i <> kW_i$: The request of the recently calculated $DispatchkW_i$ is only sent to the storage element i if it is actually different from the previous request sent, kW_i. This condition is necessary so that the controller will not end up in an infinite loop when there is no power enough in the fleet to regulate the power or current to the desired target. If that is the case, the algorithm does not request any change to the element i.

8. Send $DispatchkW_i$: Finally, if there has been a change, the controller requests a new power to be dispatched by the storage element. In this case it also pushes a new action to OpenDSS control queue (see [5]) to force a new control iteration.

   Note that the requested power may not be the actual power delivered in the next power flow solution, since the final P and Q combination must satisfy the constraints imposed by the inverter capability curve (note that the element might also be operating with a reactive power dispatch mode that also requests some vars). The requested kW and the actual delivered kW in each control iteration can be checked in the event log report, as shown in example 7.1.

7 Examples

All examples utilize the IEEE 8500-Node Test Feeder. This circuit is available in OpenDSS installation folder and should be located in "$C : \backslash ProgramFiles \backslash OpenDSS \backslash IEEETestCases \backslash 8500 - Node$" if the standard installation procedure has been used. Furthermore, all modifications and scripts run to generate each example are also available in OpenDSS "$Examples \backslash StorageController - TechNote$" folder.

The base script for all cases is shown below. The unbalanced version of the circuit is considered, which is compiled through "Master-unbal.dss" file. The regulators and capacitor controllers originally present have been disabled to avoid any undesired discrepancies between the operation of the StorageController under different settings. A power monitor named $m1\_PQ$ has been placed at the feeder head. In most cases, the controller will also be monitoring powers and currents at this same location. Additionally, a loadshape has been specified and assigned to all loads of the circuit through *batchedit* command. Seven 3-phase storage elements with the same efficiency curve have been randomly allocated at the MV level of the feeder, Figure 8. Note that not all storage elements have the same power and energy capacities. The initial state of charge of all elements has been set to 70% such that there is some energy capacity available for charging the fleet at the beginning of the day. The operation of each storage element is recorded through a monitor in mode 3.

Any other code snippets used to run each example will be shown in the respective subsection.

```
!Compile Circuit and add loadshapes to all loads
Compile Master-unbal.dss ! From IEEE 8500 Buses TestCase Folder
BatchEdit CapControl..* enabled=False
BatchEdit RegControl..* enabled=False
New Monitor.m1_PQ Line.ln5815900-1 terminal=1 mode=1 ppolar=No
New LoadShape.loads_loadshape interval=0 npts=24 csvfile=[LoadShape1.csv]
BatchEdit Load..* daily=loads_loadshape

! Inverter Efficiency Curve
New XYCurve.Eff npts = 4 xarray = [.1  .2  .4  1.0] yarray = [.86  .9  .93  .97]

!Storage fleet
New Storage.A phases=3 bus1=l3235258 kv=12.47 %idlingkW=1
~ kWhrated=500.0 %stored=70 kWrated=100.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2
New Storage.B phases=3 bus1=m1069483 kv=12.47 %idlingkW=1
~ kWhrated=1000.0 %stored=70 kWrated=200.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2
New Storage.C  phases=3 bus1=p862322  kv=12.47 %idlingkW=1
~ kWhrated=1650.0 %stored=70 kWrated=350.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2
New Storage.D phases=3 bus1=m1047615 kv=12.47 %idlingkW=1
~ kWhrated=1250.0 %stored=70 kWrated=300.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2
New Storage.E phases=3 bus1=m1069556 kv=12.47 %idlingkW=1
~ kWhrated=500.0 %stored=70 kWrated=150.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2
New Storage.F phases=3 bus1=l2688692 kv=12.47 %idlingkW=1
~ kWhrated=1200.0 %stored=70 kWrated=200.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2
New Storage.G phases=3 bus1=m1089131 kv=12.47 %idlingkW=1
~ kWhrated=1250.0 %stored=70 kWrated=250.0 EffCurve=Eff vminpu=0.8 vmaxpu=1.2

New Monitor.storage_A_states element=Storage.A mode=3
New Monitor.storage_B_states element=Storage.B mode=3
New Monitor.storage_C_states element=Storage.C  mode=3
```

```
New Monitor.storage_D_states element=Storage.D mode=3
New Monitor.storage_E_states element=Storage.E mode=3
New Monitor.storage_F_states element=Storage.F mode=3
New Monitor.storage_G_states element=Storage.G mode=3
```

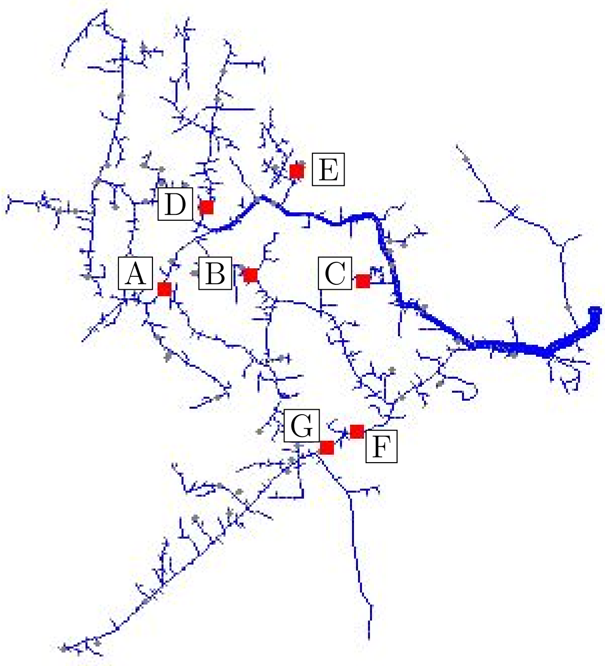


Figure 8: IEEE 8500 buses test case with location of storage fleet

7.1 PeakShave (Discharging) and Time (Charging)

7.1.1 PeakShaving Cases

In this example, the storage controller is set to operate in peakshave mode for discharging and time mode for charging. The peakshaving target is 10MW with the default band of +/-1%. The charging is set to start at 2am, with a rate of 50%. The power monitored by the controller in peakshave mode is at the first terminal of the same line where a monitor has been placed in the base script and the fleet reserve has been set to 20%.

```
Set casename=Complete
New StorageController.SC element=Line.ln5815900-1 terminal=1 modedis=peakShave
~ MonPhase=AVG kwtarget=10200 modecharge=Time timeChargeTrigger=2 %rateCharge=50
~ %reserve=20 eventlog=yes

/*
set casename=NoPower
Edit Storage.A kWrated=50
Edit Storage.B kWrated=100
Edit Storage.C kWrated=175
Edit Storage.D kWrated=150
Edit Storage.E kWrated=75
```

```
Edit Storage.F kWrated=100
Edit Storage.G kWrated=125
*/

/*
set casename=NoEnergy
Edit Storage.A kWhrated=250  %stored=70
Edit Storage.B kWhrated=500  %stored=70
Edit Storage.C kWhrated=825  %stored=70
Edit Storage.D kWhrated=625  %stored=70
Edit Storage.E kWhrated=250  %stored=70
Edit Storage.F kWhrated=600  %stored=70
Edit Storage.G kWhrated=625  %stored=70
*/

!Solution Settings
Set maxcontroliter=50
Set mode=Daily
Set stepsize=1h
Set number=24

Solve

Export monitors m1_pq
Export Eventlog
Export monitors storage_A_states
Export monitors storage_B_states
Export monitors storage_C_states
Export monitors storage_D_states
Export monitors storage_E_states
Export monitors storage_F_states
Export monitors storage_G_states
```

Three different cases are run. In the first one, named "Complete", no changes are applied to the fleet ratings, which is enough to completely shave the peak, as shown in Figure 9. In the second one, "NoPower", the rated power of each element of the fleet is reduced in 50%. In this case, it can be seen that the fleet is not able to totally shave the peak from 6pm to 8pm. In the last case, the total energy capacity of the fleet has been reduced to 50% of its original value. Because of that, the fleet reaches its reserve energy at 9pm, and cannot continue to shave the peak after that.

Note that the total necessary time to charge the fleet at the beginning of the day is different for each case due to different power and energy rated capacities. Also note the little increase in the power measured when the fleet is idling, due to the existence of idling losses.

Figure 10 shows the power dispatched by each element of the fleet for the complete shaving case. At the beginning of the day, indeed, each element charges with 50% of their respective rated power. In time mode, as explained in section 5.3.2, the fleet remain in the state set by the controller until there is no longer enough capacity left in the fleet, which can occur at different time instants for each element depending on each element's ratings. For instance, element E takes only three hours to be fully charged, while element F takes 5 hours. This can also be seen in Figure 11.

Table 2 shows the event log with all control actions executed by the controller. Note that the controller sets the fleet to discharge the first time at 6pm, when the regulated power is greater

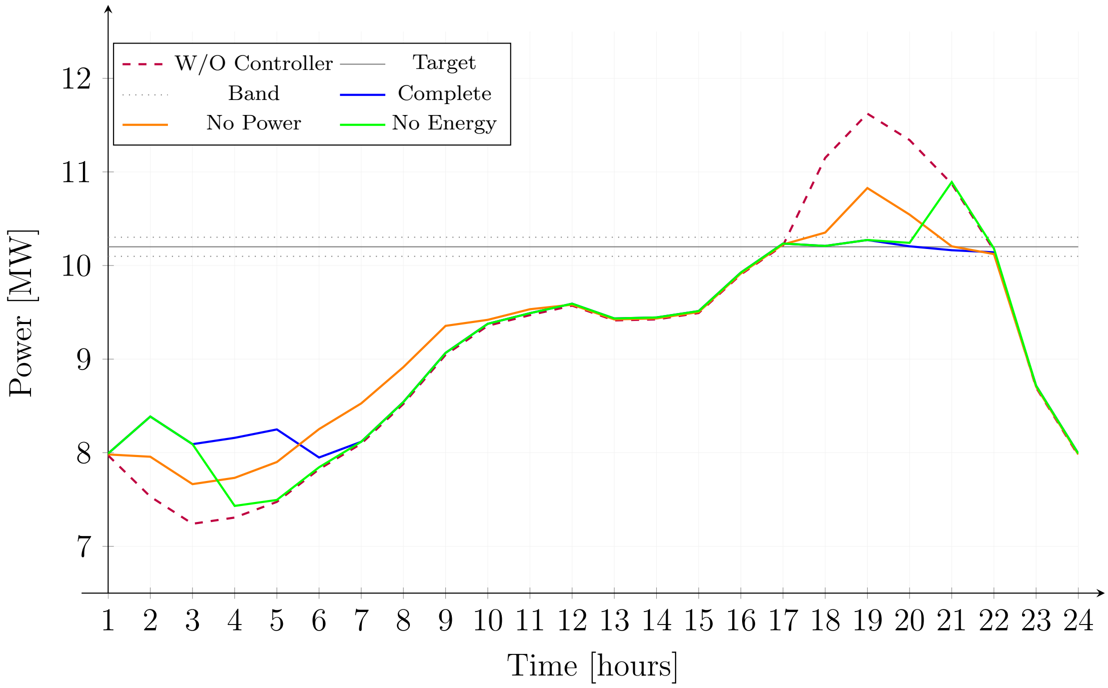


Figure 9: Powers at the monitored line element for each PeakShaving case

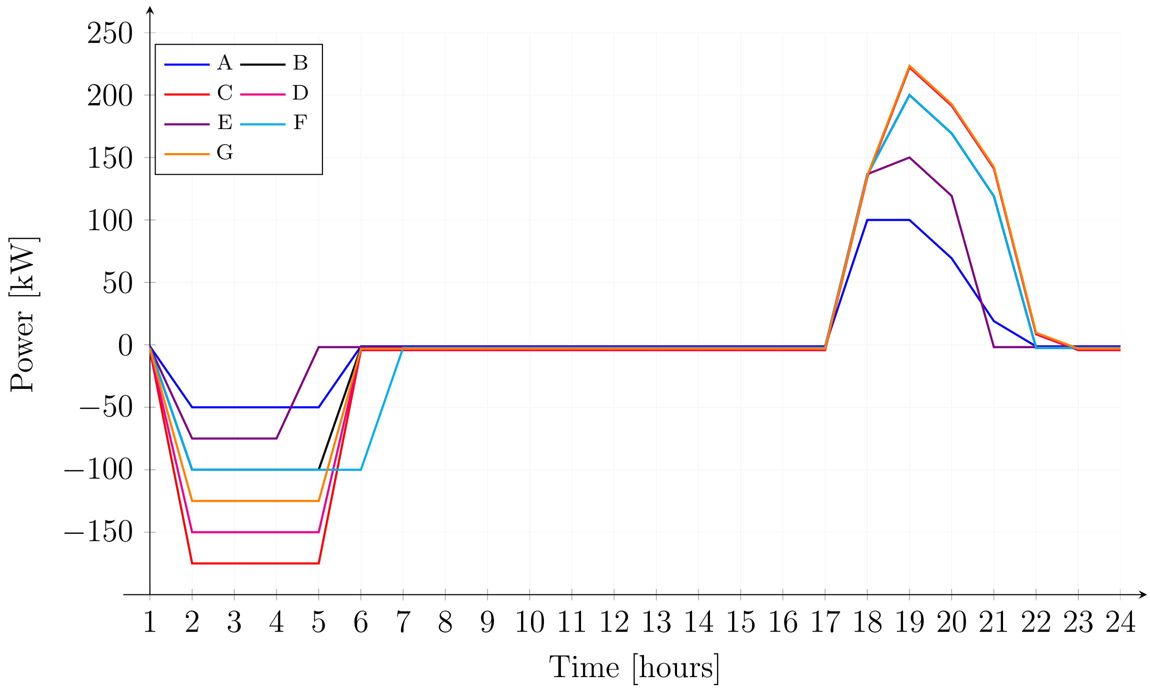


Figure 10: Power output at each element of the fleet for the case with complete shaving

than the specified target band for the first time (see the power monitored in the case without any storage elements and controller in the circuit). The power requested by the controller to element A is limited to its rated power (Equation 2). All other elements are requested the same power increment. Take elements B and F, for instance. The total power needed to drive the regulated power to the target is $969.34kW$. As the fleet is constituted by 7 elements and they are all given the same weight (see $Weights$ property), the incremental change of power requested is $138.48kW$. As their power consumption during idling state is $2kW$ (1% of $kWRated$) due to idling losses plus 0.43kW due to the corresponding inverter losses (taken from the data exported from monitors *storage_B_states* and *storage_F_states*), the power requested to these two elements is $138.48 - 2.43 = 136.05kW$.

As the system's loading increases to the peak, at 7pm, the controller requests more power from elements B to G. The amount of incremental power requested is enough to reach storage's E, F and B rated power, $150kW$, $200kW$ and $200kW$, respectively. The other three remaining elements' rated power is enough to fulfill the requested power necessary to fully shave the peak. See in Table 2 that it requires an extra control iteration to accomplish this.

At 8pm, the system's loading starts to decrease. Thus, the controller requests the fleet to reduce its dispatch. Note that because the controller operates by requesting incremental changes in the power dispatched, the entire fleet gets its dispatched power reduced by the same amount: $\frac{215.108}{7} = 30.73kW$.

At 9pm, storage E is completely depleted, as shown in Figure 11. Thus, the controller does not send any dispatch request to this element. However, the decrement in the power requested to the other six elements is enough to drive the regulated power to the band around the 10.2 MW target. Then, the controller does not push any other actions to the control queue in the second control iteration.

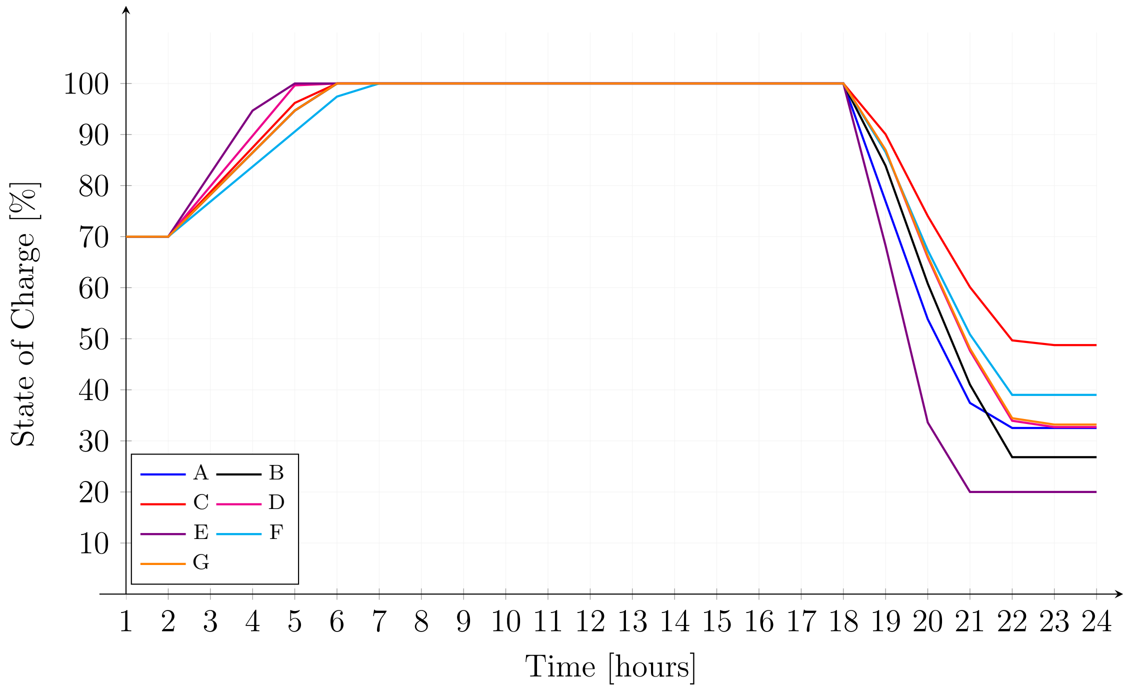


Figure 11: State of charge of each element of the fleet for the case with complete shaving

At 10pm, the total loading reduction leads the controller to decrement the total requested power in $778.753kW$, which means that each element is requested to reduce its dispatch by more than $110kW$ in the first control iteration. However, as storage E is already idling, only the other 6 elements receive the request. As storage A is dispatching less than 50kW at 9pm, the resulting power would be negative, so the element goes into idling state.

The resulting reduction (lower than the required) in the regulated power from the power flow of the next control iteration is not enough to drive it to the target value. Thus, the controller sends new actions (dispatch requests) in the second control iteration. In this iteration, the necessary power dispatch reduction is only 149.9kW. The power change requested to the fleet also leads storages B and F to go into idling state.

In the third control iteration, the power flow is solved again and the regulated power is finally within the target band. Thus, no more control action are required. Then, at 10pm only storages C, D and G remain in discharge mode and are responsible for shaving the peak.

Finally, at 11pm the system's loading reduces significantly and the controller sets these three element to idling state as well.

Table 2: Event log of PeakShaving in the simulation case with complete shaving

Hour	**Control Iter.**	**Action**
2	1	FLEET SET TO CHARGING BY TIME TRIGGER
18	1	ATTEMPTING TO DISPATCH 969.34 KW WITH 7350 KWH REMAINING AND 1470 KWH RESERVE.
18	1	REQUESTING STORAGE.A TO DISPATCH 100 KW. FINAL KWOUT IS 100 KW
18	1	REQUESTING STORAGE.B TO DISPATCH 136.05 KW. FINAL KWOUT IS 136.05 KW
18	1	REQUESTING STORAGE.C TO DISPATCH 134.23 KW. FINAL KWOUT IS 134.23 KW
18	1	REQUESTING STORAGE.D TO DISPATCH 134.836 KW. FINAL KWOUT IS 134.836 KW
18	1	REQUESTING STORAGE.E TO DISPATCH 136.657 KW. FINAL KWOUT IS 136.657 KW
18	1	REQUESTING STORAGE.F TO DISPATCH 136.05 KW. FINAL KWOUT IS 136.05 KW
18	1	REQUESTING STORAGE.G TO DISPATCH 135.443 KW. FINAL KWOUT IS 135.443 KW
19	1	ATTEMPTING TO DISPATCH 486.586 KW WITH 6263.18 KWH REMAINING AND 1470 KWH RESERVE.
19	1	REQUESTING STORAGE.B TO DISPATCH 200 KW. FINAL KWOUT IS 200 KW
19	1	REQUESTING STORAGE.C TO DISPATCH 203.742 KW. FINAL KWOUT IS 203.742 KW
19	1	REQUESTING STORAGE.D TO DISPATCH 204.349 KW. FINAL KWOUT IS 204.349 KW
19	1	REQUESTING STORAGE.E TO DISPATCH 150 KW. FINAL KWOUT IS 150 KW
19	1	REQUESTING STORAGE.F TO DISPATCH 200 KW. FINAL KWOUT IS 200 KW
19	1	REQUESTING STORAGE.G TO DISPATCH 204.955 KW. FINAL KWOUT IS 204.955 KW
19	2	ATTEMPTING TO DISPATCH 128.888 KW WITH 6263.18 KWH REMAINING AND 1470 KWH RESERVE.
19	2	REQUESTING STORAGE.C TO DISPATCH 222.154 KW. FINAL KWOUT IS 222.154 KW
19	2	REQUESTING STORAGE.D TO DISPATCH 222.761 KW. FINAL KWOUT IS 222.761 KW
19	2	REQUESTING STORAGE.G TO DISPATCH 223.368 KW. FINAL KWOUT IS 223.368 KW
20	1	ATTEMPTING TO DISPATCH -215.108 KW WITH 4726.16 KWH REMAINING AND 1470 KWH RESERVE.
20	1	REQUESTING STORAGE.A TO DISPATCH 69.2703 KW. FINAL KWOUT IS 69.2703 KW
20	1	REQUESTING STORAGE.B TO DISPATCH 169.27 KW. FINAL KWOUT IS 169.27 KW
20	1	REQUESTING STORAGE.C TO DISPATCH 191.425 KW. FINAL KWOUT IS 191.425 KW
20	1	REQUESTING STORAGE.D TO DISPATCH 192.032 KW. FINAL KWOUT IS 192.032 KW
20	1	REQUESTING STORAGE.E TO DISPATCH 119.27 KW. FINAL KWOUT IS 119.27 KW

Continues on next page

Table 2 – *Continuation from previous page*

Hour	Control Iter.	Action
20	1	REQUESTING STORAGE.F TO DISPATCH 169.27 KW. FINAL KWOUT IS 169.27 KW
20	1	REQUESTING STORAGE.G TO DISPATCH 192.638 KW. FINAL KWOUT IS 192.638 KW
21	1	ATTEMPTING TO DISPATCH -351.972 KW WITH 3496.27 KWH REMAINING AND 1470 KWH RESERVE.
21	1	REQUESTING STORAGE.A TO DISPATCH 18.9886 KW. FINAL KWOUT IS 18.9886 KW
21	1	REQUESTING STORAGE.B TO DISPATCH 118.989 KW. FINAL KWOUT IS 118.989 KW
21	1	REQUESTING STORAGE.C TO DISPATCH 141.143 KW. FINAL KWOUT IS 141.143 KW
21	1	REQUESTING STORAGE.D TO DISPATCH 141.75 KW. FINAL KWOUT IS 141.75 KW
21	1	REQUESTING STORAGE.F TO DISPATCH 118.989 KW. FINAL KWOUT IS 118.989 KW
21	1	REQUESTING STORAGE.G TO DISPATCH 142.357 KW. FINAL KWOUT IS 142.357 KW
22	1	ATTEMPTING TO DISPATCH -778.753 KW WITH 2673.63 KWH REMAINING AND 1470 KWH RESERVE.
22	1	REQUESTING STORAGE.A TO DISPATCH -92.2619 KW. SETTING STORAGE.A TO IDLING STATE. FINAL KWOUT IS -1.21359 KW
22	1	REQUESTING STORAGE.B TO DISPATCH 7.73813 KW. FINAL KWOUT IS 7.73813 KW
22	1	REQUESTING STORAGE.C TO DISPATCH 29.8925 KW. FINAL KWOUT IS 29.8925 KW
22	1	REQUESTING STORAGE.D TO DISPATCH 30.4993 KW. FINAL KWOUT IS 30.4993 KW
22	1	REQUESTING STORAGE.F TO DISPATCH 7.73813 KW. FINAL KWOUT IS 7.73813 KW
22	1	REQUESTING STORAGE.G TO DISPATCH 31.1061 KW. FINAL KWOUT IS 31.1061 KW
22	2	ATTEMPTING TO DISPATCH -149.93 KW WITH 2673.63 KWH REMAINING AND 1470 KWH RESERVE.
22	2	REQUESTING STORAGE.B TO DISPATCH -13.6804 KW. SETTING STORAGE.B TO IDLING STATE. FINAL KWOUT IS -2.42718 KW
22	2	REQUESTING STORAGE.C TO DISPATCH 8.474 KW. FINAL KWOUT IS 8.474 KW
22	2	REQUESTING STORAGE.D TO DISPATCH 9.0808 KW. FINAL KWOUT IS 9.0808 KW
22	2	REQUESTING STORAGE.F TO DISPATCH -13.6804 KW. SETTING STORAGE.F TO IDLING STATE. FINAL KWOUT IS -2.42718 KW
22	2	REQUESTING STORAGE.G TO DISPATCH 9.6876 KW. FINAL KWOUT IS 9.6876 KW
23	1	ATTEMPTING TO DISPATCH -1526.14 KW WITH 2627.38 KWH REMAINING AND 1470 KWH RESERVE.
23	1	REQUESTING STORAGE.C TO DISPATCH -209.546 KW. SETTING STORAGE.C TO IDLING STATE. FINAL KWOUT IS -4.24757 KW
23	1	REQUESTING STORAGE.D TO DISPATCH -208.939 KW. SETTING STORAGE.D TO IDLING STATE. FINAL KWOUT IS -3.64078 KW
23	1	REQUESTING STORAGE.G TO DISPATCH -208.332 KW. SETTING STORAGE.G TO IDLING STATE. FINAL KWOUT IS -3.03398 KW

7.1.2 Effect of $MonPhase$ property

In the previous example, as the property $MonPhase$ has been set to AVG. This means that the regulated measure used by the controller is the total three-phase power of the monitored line at the specified terminal, which is, in fact, the power shown in Figure 9. However, as shown in Figure 12,

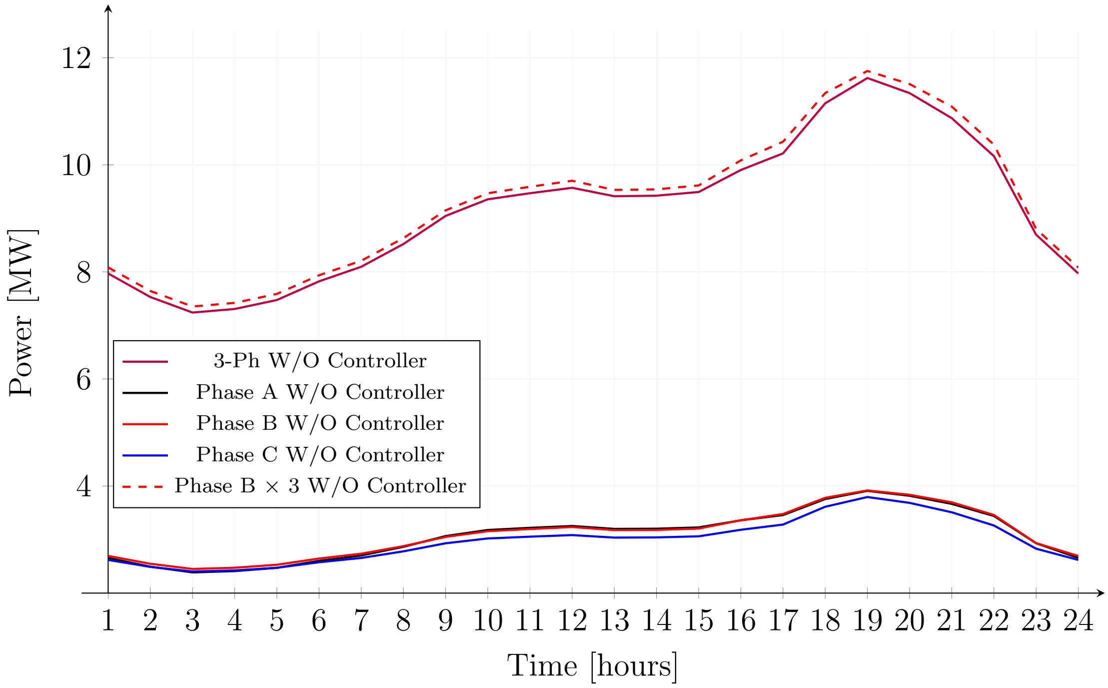


Figure 12: Powers at the monitored element without controller

the simulated circuit is unbalanced. In this case, $MonPhase$ may be useful. It allows the user to select a measure in a specific phase to be the controller input value. Consider that we would like to shave the peak adopting the most loaded phase, which is phase B (nodes "2" in the MV level, $MonPhase = 2$). As the monitored element is a 3-phase line, the controller will assume the actual power at the specific location is 3 times the measured quantity, shown in Figures 12 and 13.

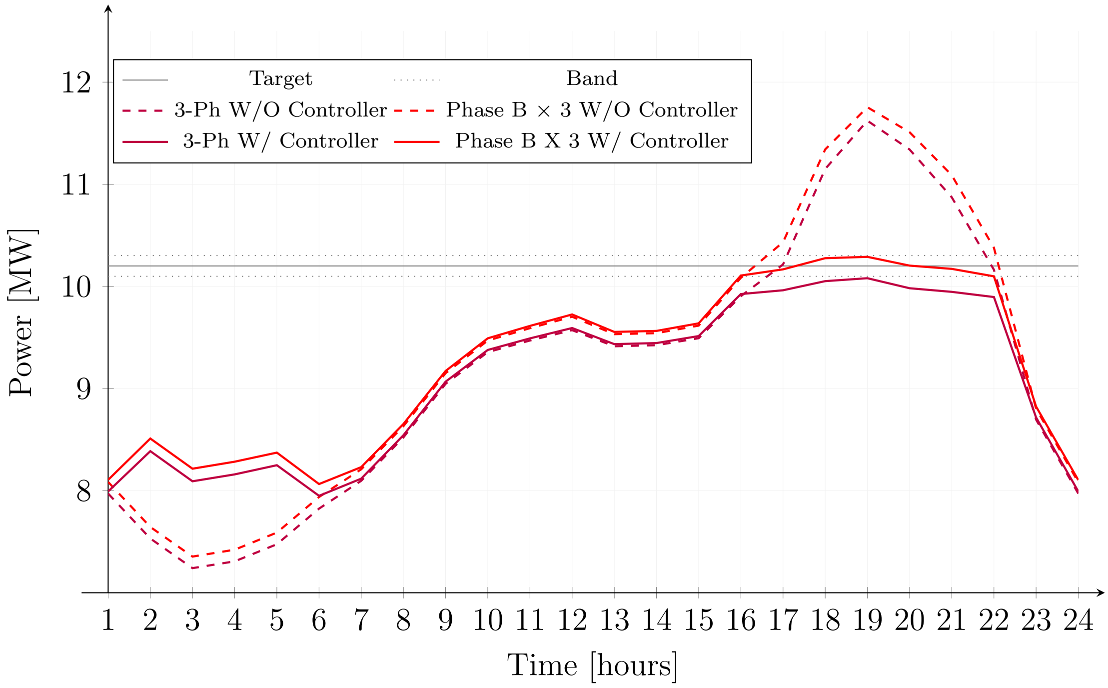


Figure 13: Powers at the monitored element for PeakShaving with $MonPhase = 2$

Figure 13 shows that, indeed, the controller regulates the monitored value to the target band. The practical effect is that the actual three-phase power measured stays below the specified target. This would be useful if there is a meter monitoring only a single phase. Supposing that the loading of each phase varies throughout the day such that one phase is the most loaded one during a specific period but not for others, the user could also use $MonPhase = MAX$ instead (which is the default value), and the controller would look for the most loaded phase at every single time step.

The intent of this property is to guarantee that the controller would not let the power in any individual phase to be overloaded. Note that if $MonPhase = AVG$, the regulation of the total 3-phase active power does not mean that the single-phase power in each phase is within the acceptable range if there is significant unbalance.

7.2 Follow (Discharging) and Time (Charging)

As explained in 5.1.3, the controller operation in Follow mode is similar to the PeakShave mode. The difference is that there is no fixed target value. The target varies according to the power measured at the time specified in *TimeDischargeTrigger* property.

The discharge trigger has been set to 6pm. Figure 14 shows that the controller regulates the power measured at 6pm indeed.

```
New StorageController.SC element=Line.ln5815900-1 terminal=1 modedis=follow
~ MonPhase=AVG kwtarget=10200 modecharge=Time timeChargeTrigger=2 %rateCharge=50
~ timeDischargeTrigger=18 %reserve=20 eventlog=yes
```

Note that the corresponding band (in kW) also varies depending on the target set at *TimeDischargeTrigger*. A fixed band can also be set with property *kWBand*. In this situation, the automatic update of the band as percentage of the varying target is disabled.

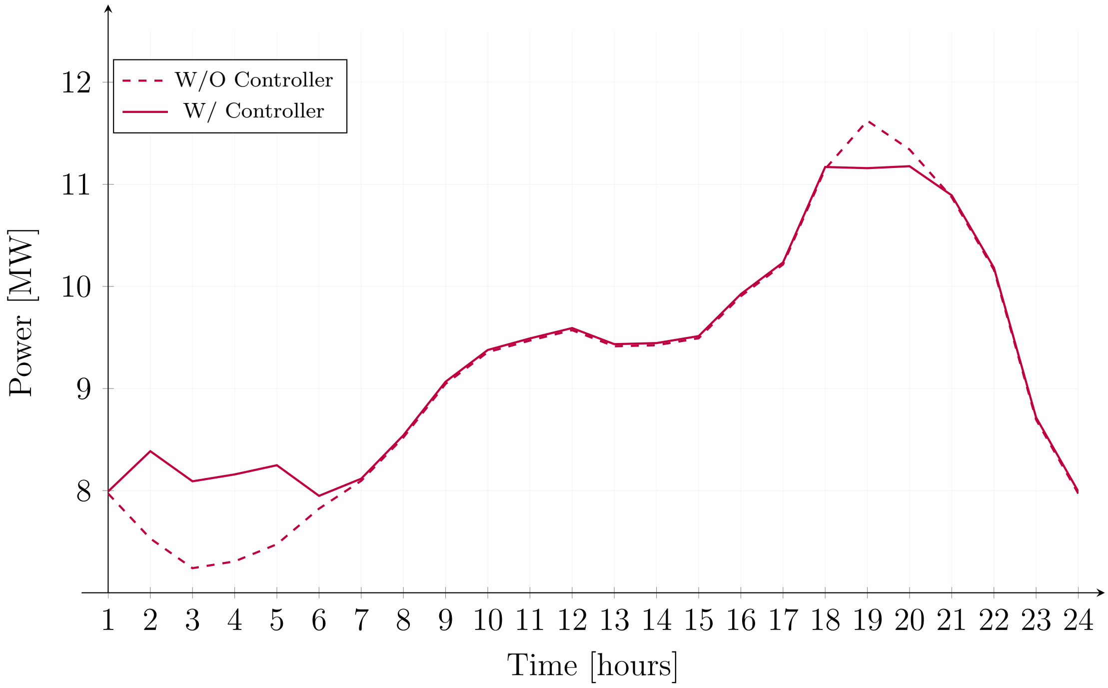


Figure 14: Powers at the monitored element in Follow mode with *DischargeTrigger* set to 6pm

7.3 Support (Discharging) and Time (Charging)

To demonstrate the StorageController operation in Support mode, a generator has been added to the circuit as shown in Figure 15.

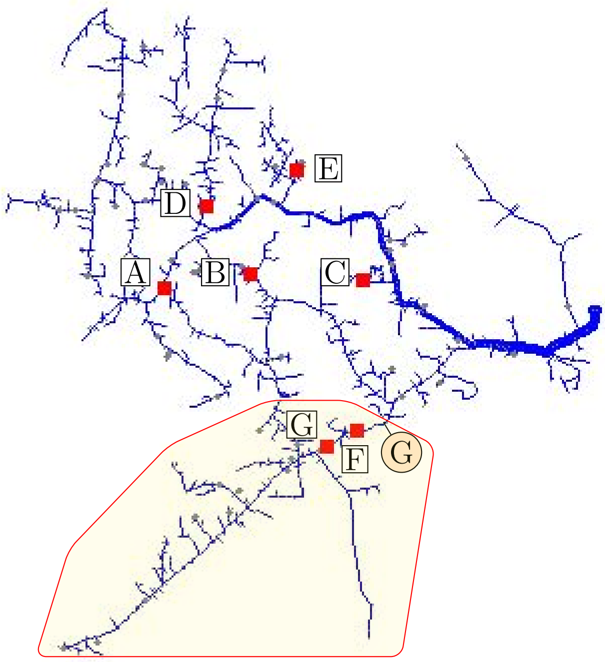


Figure 15: IEEE 8500 buses test case with location of storage fleet and the generator added for example 7.3

Only the two storage elements located downstream of the generator will participate in the regulation, i.e., storages F and G.

Thus, these two elements are explicitly specified in controller's *ElementList* property. Their ratings have also been intentionally increased so they can provide full support to the generator. The power monitored by the controller has been set to the first line segment upstream the generator such that both the generator and the storage fleet will directly affect it. The controller target has been set to 2000kW. This means that whenever the power exported from the zone highlighted in Figure 15 is less than the target, the controller will dispatch the fleet to provide the necessary support.

```
New LoadShape.DGdispatch npts=24 interval=1 mult=[0.78 0.65 0.65 0.67 0.70 1
    0.98 0.97 0.9 0.83 0.8 0.79 0.85 0.84 1 0.4 0.35 0.32 0.3 0.32 0.35 0.4 0.8
    1]
New Generator.G model=1 bus1=m1108295 kW=8000 daily=DGdispatch kV=12.47
~ vmaxpu=1.2 vminpu=0.8
New Monitor.Pmon element=Line.LN6351524-1 terminal=2 mode=1 ppolar=no

New StorageController.SC element=Line.ln6351524-1 terminal=1 modedis=support
~ kwtarget=2000 modecharge=Time timeChargeTrigger=2 %rateCharge=50 MonPhase=AVG
~ elementList=[F G] %reserve=20 eventlog=yes

Edit Storage.F kWrated=1500 kVA=1500 kWhrated=8500 %stored=70
Edit Storage.G kWrated=500 kWrated=500 kWhrated=3500 %stored=70
```

A simplified single-line diagram is shown in Figure 16 with all relevant elements and measures. P_{mon}^{SC} corresponds to the power monitored by the controller. It is negative if the lower portion of the circuit exports power to the upstream section, satisfying the sign convention adopted for this mode.

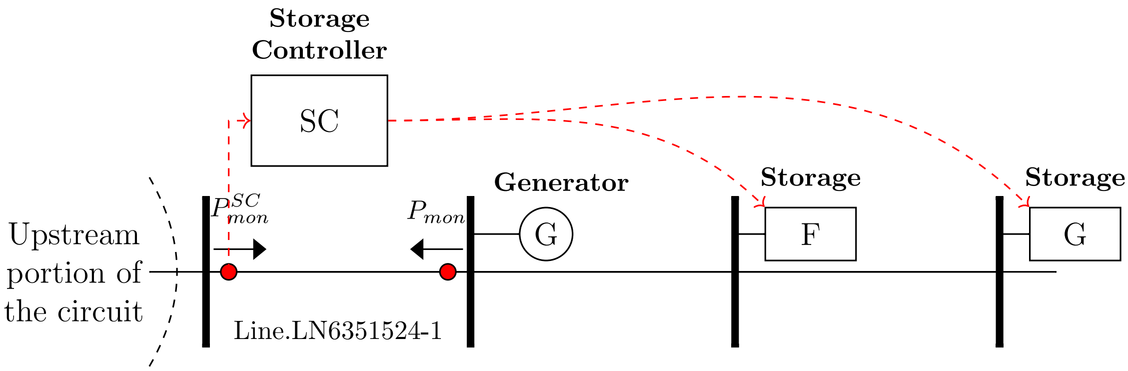


Figure 16: Simplified single-line diagram of lower portion of the circuit

Figure 17 shows that the fleet composed by storages F and G is able to provide support to the generator when the exported power drops below 2MW, which happens between 4pm and 10pm. Note that the charging of the fleet has the same effect of additional loads in the lower portion of the circuit, reducing the amount of power exported.

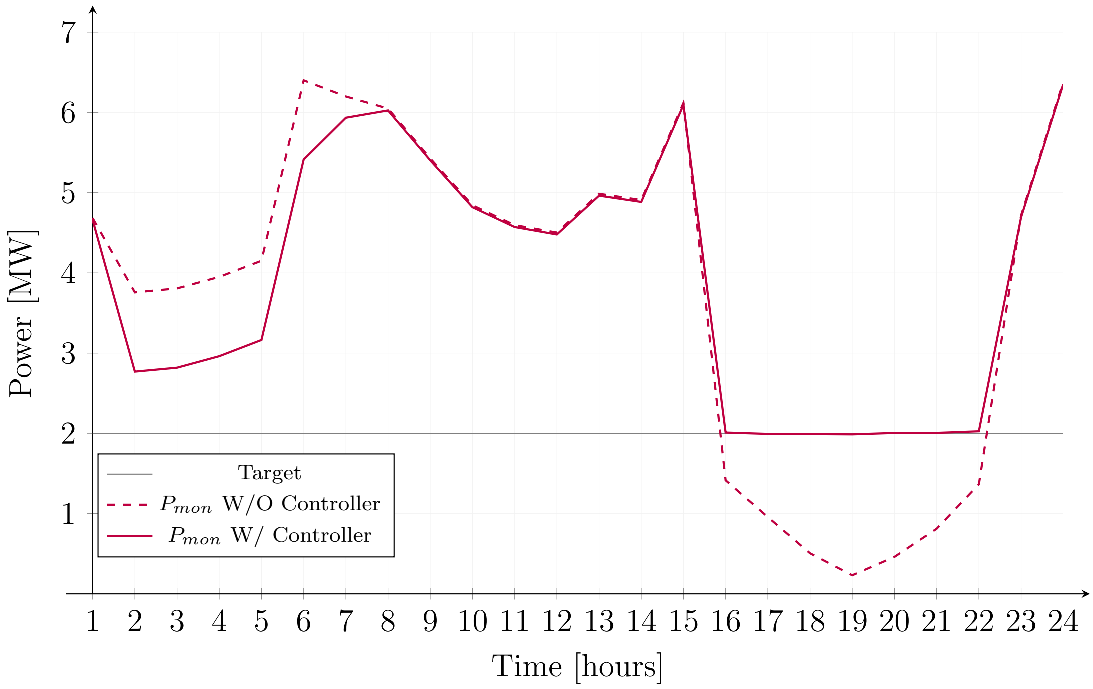


Figure 17: Powers entering the second terminal of the monitored line element in example 7.3

7.4 Schedule (Discharging) and Time (Charging)

In Schedule mode, the power requested by the controller to the fleet does not directly depend on a previous power flow solution. However, previous power flow solutions may *indirectly* affect the available energy at the storage, which in turns may impede the fleet to follow the specified schedule, as shown in this example. The storage controller has been specified as follows:

```
New StorageController.SC element=Line.ln5815900-1 terminal=1 modedis=schedule
~ tup=3 tflat=4 tdn=2 timeDischargeTrigger=14 modecharge=Time
~ timeChargeTrigger=2  %ratekW=100 %rateCharge=50 %reserve=20 eventlog=yes
```

Note that even though the controller does not utilize any power/current reference in this mode, it requires an element to be specified.

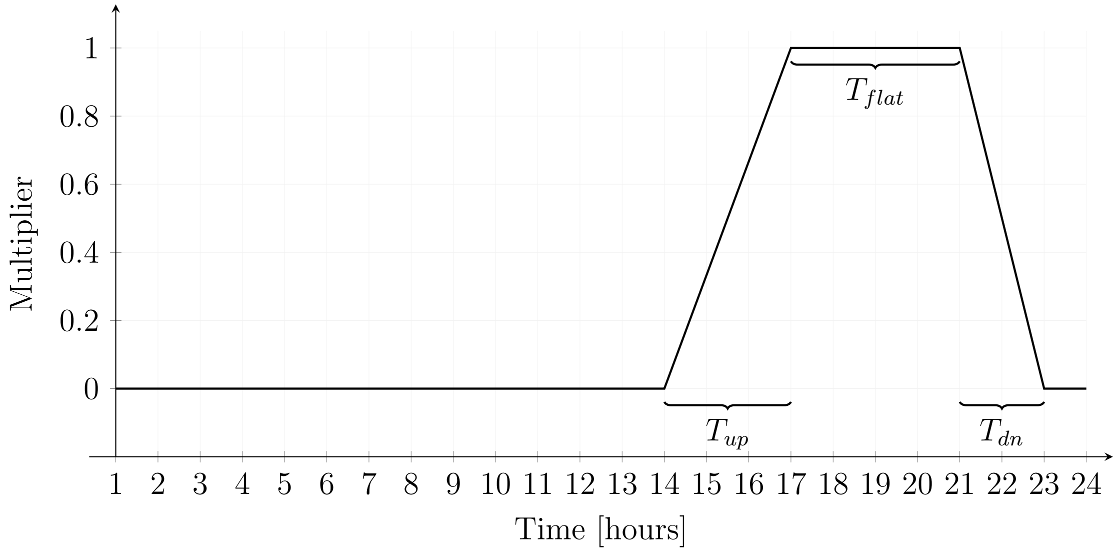


Figure 18: Dispatch shape as defined by Tup, $Tflat$, Tdn and $timeDischargeTrigger$ properties in example 7.4

Figure 19 shows the power measured at the feeder head along with the total power dispatched by the fleet. Note that during discharging, it precisely follows the schedule until 6pm. However, at 7pm some elements of the fleet are fully depleted, as shown in Figure 20. The elements that still have energy stored continue following the schedule, until 9pm, when the entire fleet has reached the specified $\%reserve$.

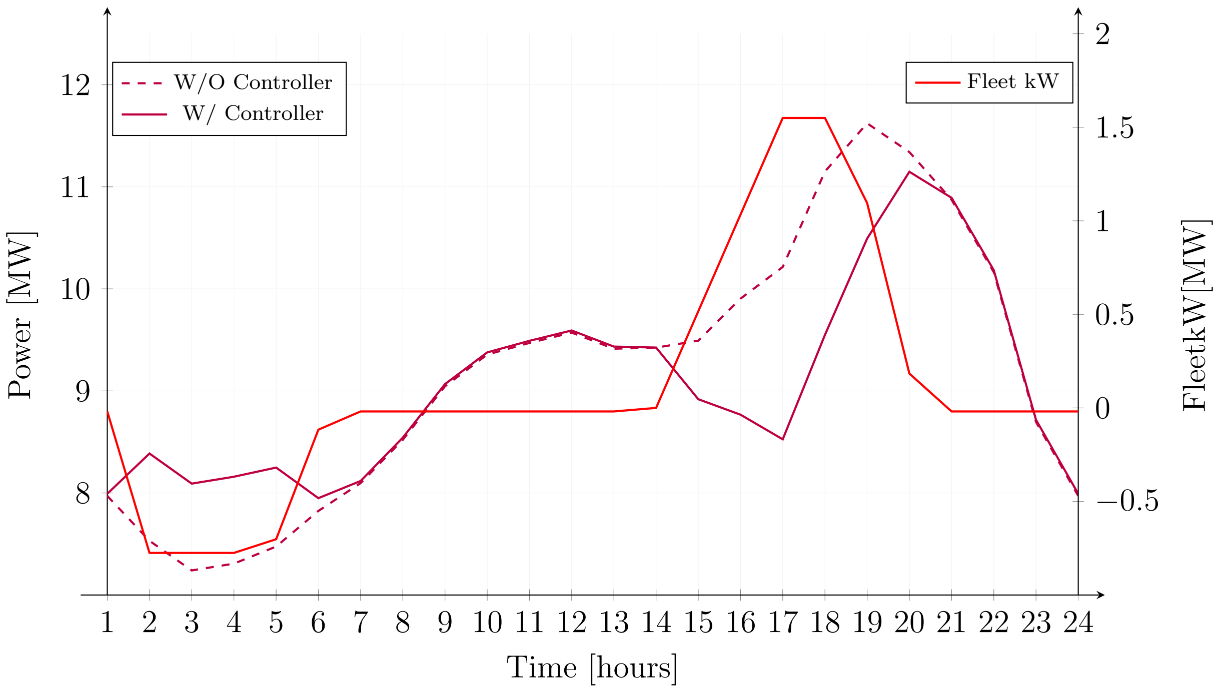


Figure 19: Powers at the monitored element and the total power dispatched by the fleet in example 7.4

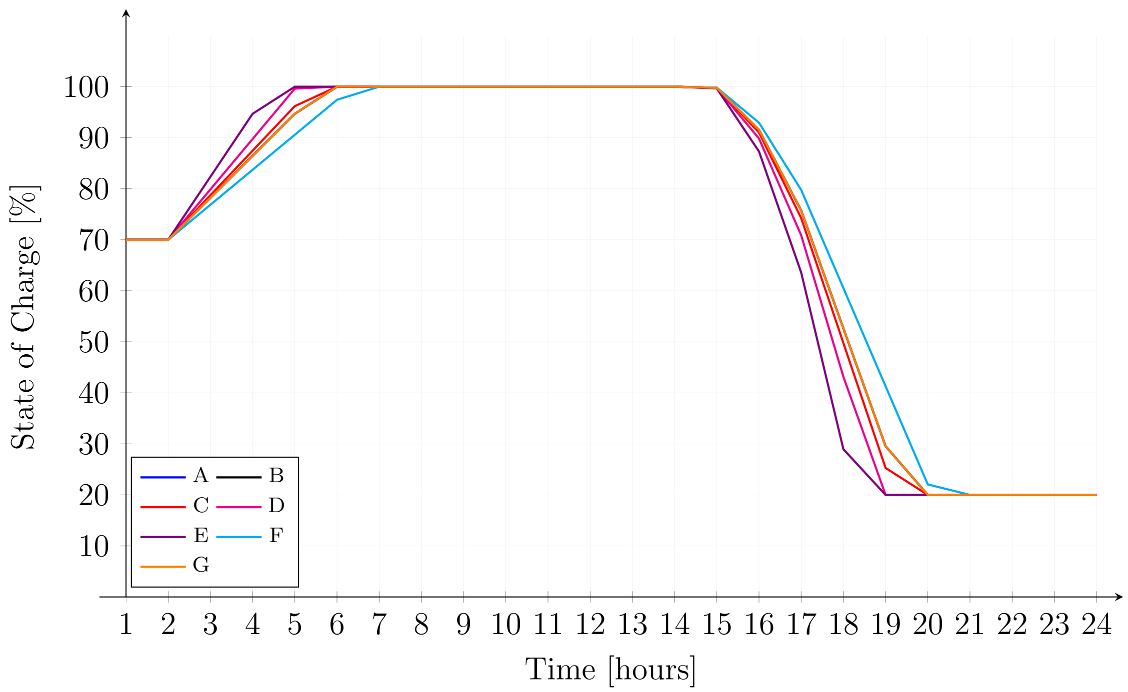


Figure 20: State of charge of each element of the fleet in example 7.4

7.5 PeakShave (Discharging) and PeakShaveLow (Charging)

In this example, PeakShave and PeakShaveLow modes are applied simultaneously. For the PeakShave mode, the target has been set to 10.2MW and for the PeakShaveLow, it has been set to 8.5MW. The initial stored energy of each element of the fleet has been reduced to 20% of the respective rated capacity, $kWhrated$, such that the fleet operates in charging mode from 1 to 8am.

```
BatchEdit Storage..* %stored=20
New StorageController.SC element=Line.ln5815900-1 terminal=1 modedis=peakShave
~ MonPhase=AVG kwtarget=10200 modecharge=peakShaveLow kwtargetLow=8500
~ eventlog=yes %reserve=20
```

The fleet is able to shave both peaks, as shown in Figure 21. Note that the charging of the fleet starts at midnight, as the system's loading drops below $kWTargetLow$ at this time step.

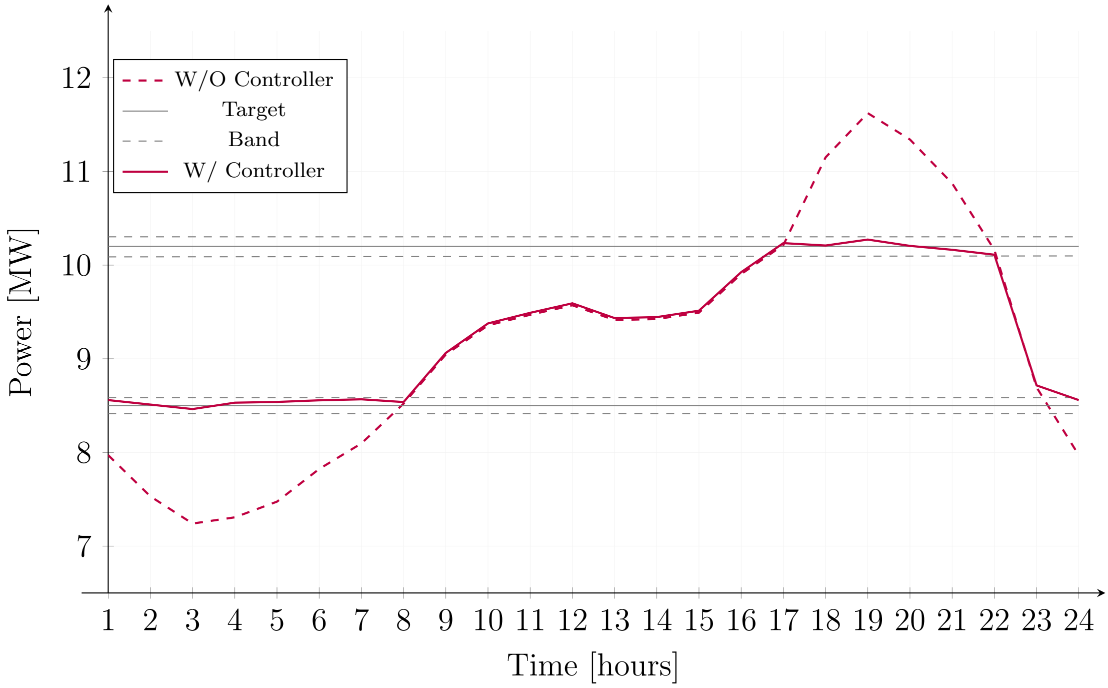


Figure 21: Powers at the monitored element in example 7.5

7.6 I-PeakShave (Discharging) and I-PeakShaveLow (Charging)

An example illustrating the utilization of the peakshave modes based on current, I-PeakShave (for Discharging) and I-PeakShaveLow (for Charging), is provided with the program in the folder "*Examples\StoCtrl_Current_PeakShave*".

7.7 Loadshape (Charging and Discharging)

In loadshape mode, both charging and discharging of the fleet follows a dispatch curve defined in a loadshape object, as follows.

```
New LoadShape.SC_loadshape interval=1 npts=24 mult = [0,
    0,-0.3,-0.45,-0.5,-0.45,-0.3,0,0,0,0,0,0,0,0,0.3,0.5,0.8,0.9,0.8,0.5,0.3,0,0]

New StorageController.SC element=Line.ln5815900-1 terminal=1 modedis=loadshape
~ daily=SC_loadshape %rateCharge=50 %reserve=20  eventlog=yes
```

Note that it is unnecessary to set *ModeCharge* to Loadshape. This happens automatically when the discharge mode is set to Loadshape. Figure 22 shows the dispatch curve and the power measured at the beginning of the feeder with and without the controller.

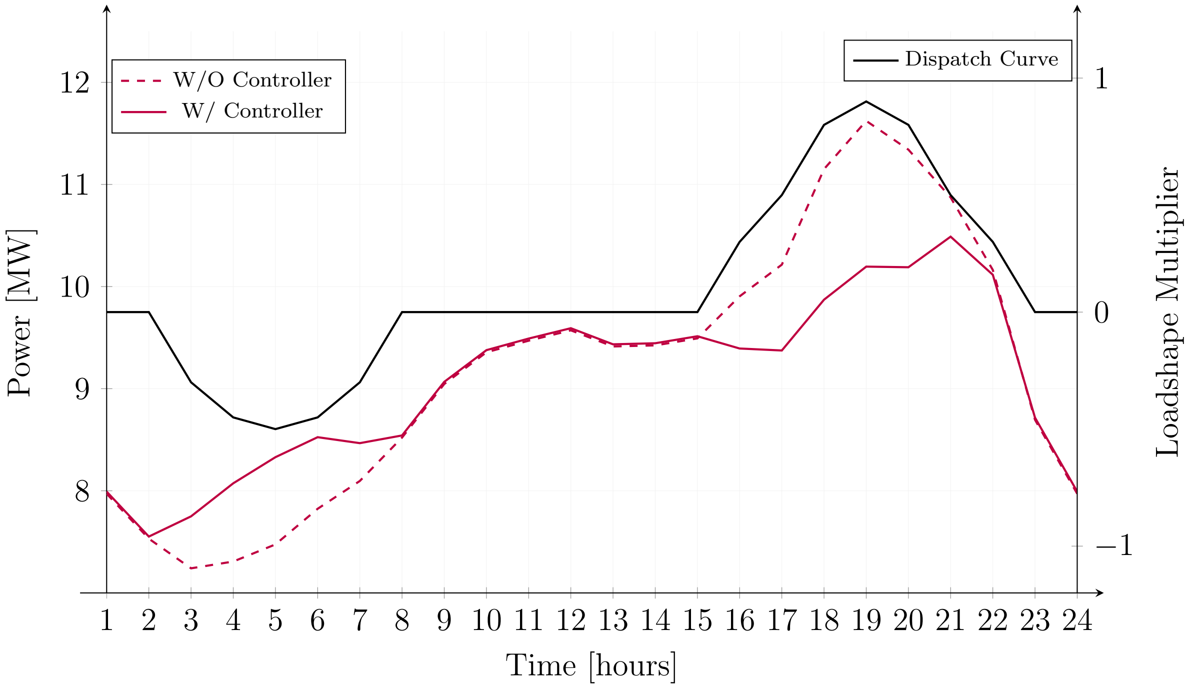


Figure 22: Powers at the monitored element and dispatch curve in example 7.7

7.8 Time (Charging) and (Discharging)

Time discharging mode works the same way as charging, i.e., the fleet is set to charge at the time instant set by the respective time trigger and the fleet remains in the desired dispatch state until fully discharged.

```
New StorageController.SC element=Line.ln5815900-1 terminal=1 modedis=time
~ timeDischargeTrigger=17 %ratekW=100 modecharge=time timeChargeTrigger=2
~ %rateCharge=50 %reserve=20 eventlog=yes
```

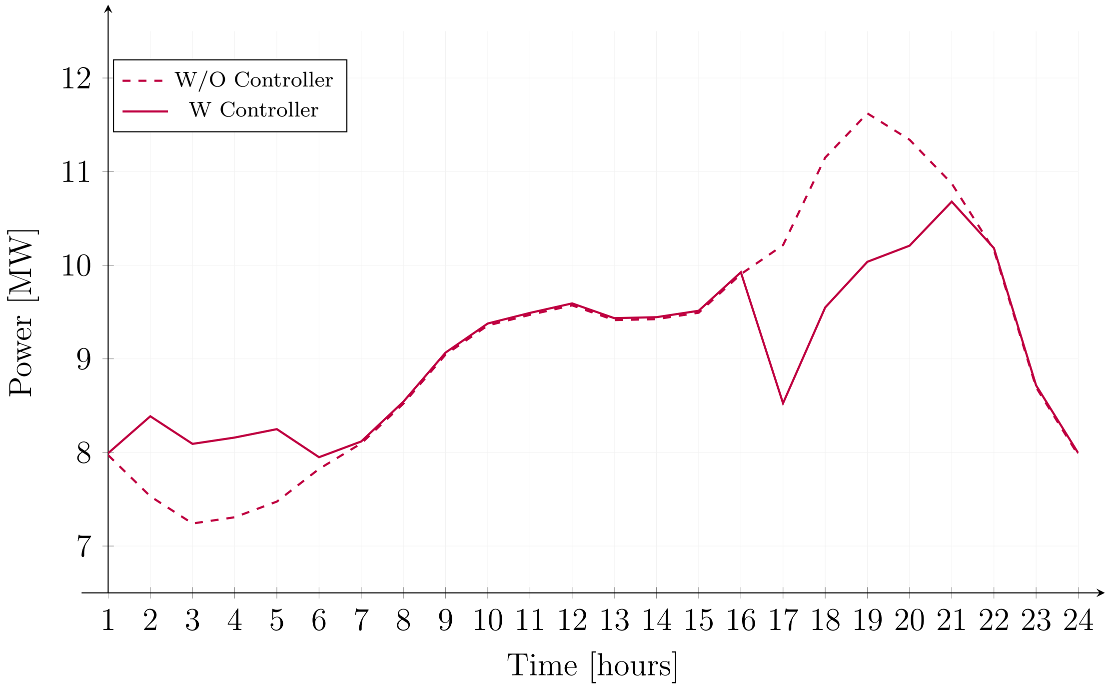


Figure 23: Powers at the monitored element in example 7.8

Note that the fleet takes 4 hours to charge 30% of its energy capacity (the initial state of charge of each element of the fleet is 70%), whereas it takes only 5 hours to discharge 80% down to %reserve. This is due to the fact that the charging rate is 50% whereas the discharging rate has been set to 100%.

8 List of properties

Table 3 lists all available properties of the StorageController element.

Table 3: Properties of the StorageController element

Property	Description
(1) Element	Full object name of the circuit element, typically a line or transformer, which the control is monitoring. There is no default; Must be specified.
(2) Terminal	Number of the terminal of the circuit element to which the StorageController control is connected. 1 or 2, typically. Default is 1. Make sure to select the proper direction on the power for the respective dispatch mode.
(3) MonPhase	Number of the phase being monitored or one of {AVG|MAX|MIN} for all phases. Default=MAX. Must be less than the number of phases. Used in PeakShave, Follow, Support and I-PeakShave discharging modes and in PeakShaveLow, I-PeakShaveLow charging modes. For modes based on active power measurements, the value used by the control is the monitored one multiplied by the number of phases of the monitored element.
(4) kWTarget	kW/kamps target for Discharging. The storage element fleet is dispatched to try to hold the power/current in band at least until the storage is depleted. The selection of power or current depends on the Discharge mode (PeakShave → kW, I-PeakShave → kamps).
(5) kWTargetLow	kW/kamps target for Charging. The storage element fleet is dispatched to try to hold the power/current in band at least until the storage is fully charged. The selection of power or current depends on the charge mode (PeakShavelow → kW, I-PeakShavelow→ kamps).
(6) %kWBand	Bandwidth (% of Target kW/kamps) of the dead band around the kW/kamps target value. Default is 2% (+/-1%). No dispatch changes are attempted if the power in the monitored terminal stays within this band.
(7) kWBand	Alternative way of specifying the bandwidth. (kW/kamps) of the dead band around the kW/kamps target value. Default is 2% of kWTarget (+/-1%). No dispatch changes are attempted if the power in the monitored terminal stays within this band.
(8) %kWBandLow	Bandwidth (% of kWTargetLow) of the dead band around the kW/kamps low target value. Default is 2% (+/-1%). No charging is attempted if the power in the monitored terminal stays within this band.
(9) kWBandLow	Alternative way of specifying the bandwidth. (kW/kamps) of the dead band around the kW/kamps low target value. Default is 2% of kW-TargetLow (+/-1%). No charging is attempted if the power in the monitored terminal stays within this band.

Continues on next page

Table 3 – *Continuation from previous page*

Property	Description
(10) ElementList	Array list of Storage elements to be controlled. If not specified, all storage elements in the circuit not presently dispatched by another controller are assumed dispatched by this controller.
(11) Weights	Array of proportional weights corresponding to each storage element in the ElementList. The needed kW or kvar to get back to center band is dispatched to each storage element according to these weights. Default is to set all weights to 1.0.
(12) ModeDischarge	{PeakShave*|Follow|Support|Loadshape|Time|Schedule|I-PeakShave} Mode of operation for the DISCHARGE FUNCTION of this controller. In PeakShave mode (Default), the control attempts to discharge storage to keep power in the monitored element below the kWTarget. In Follow mode, the control is triggered by time and resets the kWTarget value to the present monitored element power. It then attempts to discharge storage to keep power in the monitored element below the new kWTarget. See TimeDischargeTrigger. In Support mode, the control operates oppositely of PeakShave mode: storage is discharged to keep kW power output up near the target. In Loadshape mode, both charging and discharging precisely follows the per unit loadshape. Storage is discharged when the loadshape value is positive. In Time mode, the storage discharge is turned on at the specified %RatekW at the specified discharge trigger time in fractional hours. In Schedule mode, the Tup, TFlat, and Tdn properties specify the up ramp duration, flat duration, and down ramp duration for the schedule. The schedule start time is set by TimeDischargeTrigger and the rate of discharge for the flat part is determined by %RatekW. In I-PeakShave mode, the control attempts to discharge storage to keep current in the monitored element below the target given in k-amps (thousands of amps), when this control mode is active, the property kWTarget will be expressed in k-amps.

Continues on next page

Table 3 – *Continuation from previous page*

Property	Description
(13) ModeCharge	{Loadshape|Time*|PeakShaveLow|I-PeakShaveLow} Mode of operation for the CHARGE FUNCTION of this controller. In Loadshape mode, both charging and discharging precisely follows the per unit loadshape. Storage is charged when the loadshape value is negative. In Time mode, the storage charging FUNCTION is triggered at the specified %RateCharge at the specified charge trigger time in fractional hours. In PeakShaveLow mode, the charging operation will charge the storage fleet when the power at a monitored element is below a specified KW target (kWTarget_low). The storage will charge as much power as necessary to keep the power within the deadband around kWTarget_low. In I-PeakShaveLow mode, the charging operation will charge the storage fleet when the current (Amps) at a monitored element is below a specified amps target (kWTarget_low). The storage will charge as much power as necessary to keep the amps within the deadband around kWTarget_low. When this control mode is active, the property kWTarget_low will be expressed in k-amps and all the other parameters will be adjusted to match the amps (current) control criteria.
(14) TimeDischargeTrigger	Default time of day (hr) for initiating Discharging of the fleet. During Follow or Time mode discharging is triggered at a fixed time each day at this hour. If Follow mode, storage will be discharged to attempt to hold the load at or below the power level at the time of triggering. In Time mode, the discharge is based on the %RatekW property value. Set this to a negative value to ignore. Default is 12.0 for Follow mode; otherwise it is -1 (ignored).
(15) TimeChargeTrigger	Default time of day (hr) for initiating charging in Time control mode. Set this to a negative value to ignore. Default is 2.0. (0200). When this value is >0 the storage fleet is set to charging at this time regardless of other control criteria to make sure storage is topped off for the next discharge cycle.
(16) %RatekW	Sets the kW discharge rate in % of rated capacity for each element of the fleet. Applies to TIME control mode, SCHEDULE mode, or anytime discharging is triggered by time.
(17) %RateCharge	Sets the kW charging rate in % of rated capacity for each element of the fleet. Applies to TIME control mode and anytime charging mode is entered due to a time trigger.
(18) %Reserve	Use this property to change the % reserve for each storage element under control of this controller. This might be used, for example, to allow deeper discharges of storage or in case of emergency operation to use the remainder of the storage element.

Continues on next page

Table 3 – *Continuation from previous page*

Property	Description
(19) kWhTotal	(Read only). Total rated kWh energy storage capacity of storage elements controlled by this controller.
(20) kWTotal	(Read only). Total rated kW power capacity of storage elements controlled by this controller.
(21) kWhActual	(Read only). Actual kWh stored of all controlled storage elements.
(22) kWActual	(Read only). Actual kW output of all controlled storage elements.
(23) kWneed	(Read only). KW needed to meet target.
(24) Yearly	Dispatch loadshape object, If any, for Yearly solution Mode.
(25) Daily	Dispatch loadshape object, If any, for Daily solution mode.
(26) Duty	Dispatch loadshape object, If any, for Dutycycle solution mode.
(27) EventLog	{Yes/True|No/False} Default is No. Log control actions to Eventlog.
(28) InhibitTime	Hours (integer) to inhibit Discharging after going into Charge mode. Default is 5.
(29) Tup	Duration, hrs, of upramp part for SCHEDULE mode. Default is 0.25.
(30) TFlat	Duration, hrs, of flat part for SCHEDULE mode. Default is 2.0.
(31) Tdn	Duration, hrs, of downramp part for SCHEDULE mode. Default is 0.25.
(32) kWThreshold	Threshold, kW, for Follow mode. kW has to be above this value for the Storage element to be dispatched on. Defaults to 75% of the kWTarget value. Must reset this property after setting kWTarget if you want a different value.
(33) DispFactor	Defaults to 1 (disabled). Set to any value between 0 and 1 to enable this parameter. Use this parameter to reduce the amount of power requested by the controller in each control iteration. It can be useful when maximum control iterations are exceeded due to numerical instability such as fleet being set to charging and idling in subsequent control iterations (check the Eventlog).
(34) ResetLevel	The level of charge required for allowing the storage to discharge again after reaching the reserve storage level. After reaching this level, the storage control will not allow the storage device to discharge, forcing the storage to charge. Once the storage reaches thislevel, the storage will be able to discharge again. This value is a number between 0.2 and 1.
(35) Seasons	With this property the user can specify the number of targets to be used by the controller using the list given at "SeasonTargets"/"SeasonTargetsLow", which can be used to dynamically adjust the storage controller during a QSTS simulation. The default value is 1. This property needs to be defined before defining SeasonTargets/SeasonTargetsLow.

Continues on next page

Table 3 – *Continuation from previous page*

Property	Description
(36) SeasonTargets	An array of doubles specifying the targets to be used during a QSTS simulation. These targets will take effect only if SeasonRating=true. The number of targets cannot exceed the number of seasons defined at the SeasonSignal.The difference between the targets defined at SeasonTargets and SeasonTargetsLow is that SeasonTargets applies to discharging modes, while SeasonTargetsLow applies to charging modes.
(37) SeasonTargetsLow	An array of doubles specifying the targets to be used during a QSTS simulation. These targets will take effect only if SeasonRating=true. The number of targets cannot exceed the number of seasons defined at the SeasonSignal.The difference between the targets defined at SeasonTargets and SeasonTargetsLow is that SeasonTargets applies to discharging modes, while SeasonTargetsLow applies to charging modes.

9 References

[1] R. Dugan and D. Montenegro, “Opendss storage element and storagecontroller element,” EPRI, OpenDSS Tech. Note, October 2019.

[2] C. Rocha, J. Peppanen, P. Radatz, M. Rylander, and R. Dugan, “Storage element,” EPRI, OpenDSS Tech. Note, November 2019.

[3] P. Radatz, W. Sunderman, and C. Rocha, “Opendss pvsystem and invcontrol element models,” EPRI, OpenDSS Tech. Note, November 2019.

[4] R. Dugan and D. Montenegro, “The open distribution system simulator (opendss),” EPRI, OpenDSS Tech. Note, April 2018.

[5] A. Birchfield, “Python to opendss interface for mopdeling control systems,” EPRI, OpenDSS Tech. Note, July 2015.

[6] C. Rocha, P. Radatz, J. Peppanen, M. Rylander, and R. Dugan, “Inverter modelling,” EPRI, OpenDSS Tech. Note, 2019.