OpenDSS PVSystem and InvControl Element Models

1 OpenDSS PVSystem Element Model

Figure 1 shows a schematic diagram of the PVSystem element model recently implemented into OpenDSS version 8.A.A at Build BB. The model combines the photovoltaic (PV) array and the PV inverter. It assumes that the inverter is capable of tracking the maximum power point (mpp) of the panel quickly, allowing adequate use in QSTS simulations with a time step of at least one second. This considerably facilitates PV modeling and is an adequate simplification for most interconnection impact studies.

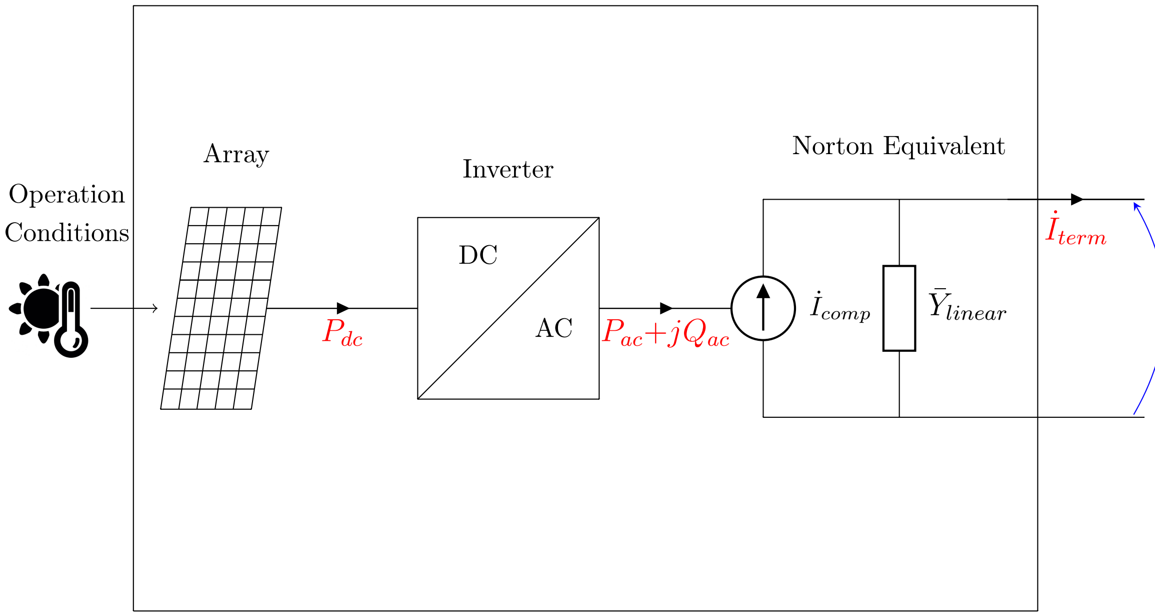

Figure 1: Block diagram of the PVSystem element model

1.1 Definition of PVSystem properties

The properties needed to define a PVSystem element in OpenDSS can be separated into 3 groups:

1. PV array properties;
2. PV inverter properties;
3. Operation conditions properties.

These groups are described separately below.

1.1.1 PV array properties

The PV array properties that need to be defined in the model are:

- *Pmpp*: The rated max power of the PV array, in kW, for $1\,kW/m^2$ irradiance and a user-defined temperature. The *P-TCurve* property should be defined relative to the user-defined temperature.
- *P-TCurve*: Correction factor curve of the PV array per unit of *Pmpp* as a function of the PV array temperature. As can be seen in Figure 2, the correction factor should be 1.0 for the temperature at which *Pmpp* is defined, in this case 25 °C. The PV array generates its rated max power, *Pmpp*, operating at that temperature, 25 °C, combined with $1\,kW/m^2$ irradiance. To define this curve in OpenDSS, the user must use the XYCurve object, as shown below.

```
New XYCurve.FatorPvsT npts=4 xarray=[0 25 75 100] yarray=[1.2 1.0 .8 .6]
```

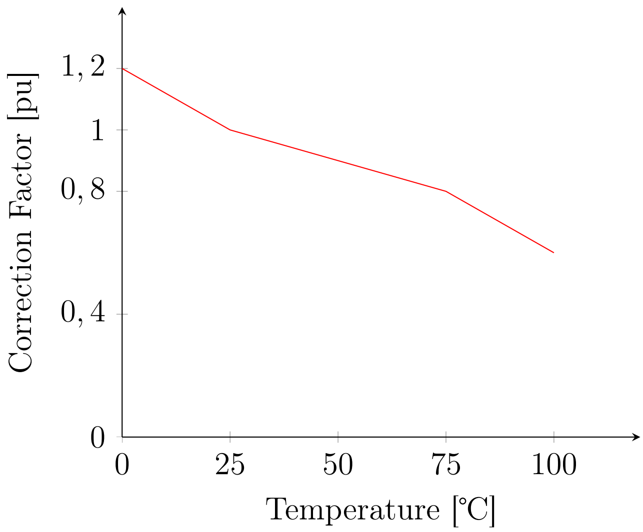

Figure 2: Correction factor vs Temperature

1.1.2 PV inverter properties

The inverter properties that may be defined in the model are:

- *kVA*: kVA rating of inverter;
- *kV*: Nominal rated (1.0 per unit) voltage, kV, for PVSystem element.

  If 3-phase PVSystem elements, specify phase-phase kV.

  If 1-phase wye (star or LN), specify phase-neutral kV.

  If 1-phase delta or phase-phase connected, specify phase-phase kV;
- *Phases*: Number of Phases of PVSystem element;
- *bus1*: Bus to which the PVSystem element is connected. May include specific node definition;

- *conn*: Connection of PVSystem element;
- *PF*: Power factor for the output power. Setting this property will cause the inverter to operate in CONSTANT POWER FACTOR MODE. By definition, when this value is negative, the inverter absorbs reactive power (inductive characteristic) and, when positive, it provides reactive power (capacitive characteristic);
- *kvar*: Output reactive power. Setting this property forces the inverter to operate in CONSTANT kvar MODE. When this value is negative, the inverter absorbs reactive power (inductive characteristic) and, when positive, provides reactive power (capacitive characteristic);
- *pctPmpp*: Upper limit on output active power as a percentage of *Pmpp*;
- *%cutin*: Percentage of inverter's kVA rating, see *kVA* property. When the inverter is OFF, the power from the PV array must be greater than this value for the inverter to turn ON;
- *%cutout*: Percentage of inverter's kVA rating, see *kVA* property. When the inverter is ON, it turns OFF when the power from the array drops below this value;
- *kvarLimit*: Maximum value of reactive power in *kvar* that the inverter can provide to the grid;
- *kvarLimitneg*: Maximum value of reactive power in *kvar* that the inverter can absorb from the grid (as an un-signed value);
- *VarFollowInverter*: Boolean variable which indicates that the reactive power does not respect the inverter status.

  When set to True, the PVSystem reactive power will cease when the inverter status is OFF, due to the power from PV array dropping below *%cutout*. The reactive power will begin again when the power from PV array is above *%cutin*.

  When set to False, the PVSystem will provide/absorb reactive power regardless of the status of the inverter;
- *WattPriority*: Boolean variable which determines whether the inverter should prioritize the active or reactive power when its capacity, kVA, is exceeded.

  When set to True, the active power is prioritized.

  When set to False, the reactive power is prioritized;
- *PFPriority*: Boolean variable which, when set to True, forces the power factor value to its rated value, PF, when the inverter capacity, kVA, is exceeded. This property, if enabled, takes precedence over *WattPriority* property;
- *pctPminNoVars*: Percentage of the *Pmpp*. The inverter is not allowed to provide/absorb reactive power when its active output power is less than this value;
- *pctPminkvarLimit*: Percentage of the *Pmpp*. The inverter can provide/absorb reactive power up to its maximum allowed value, *kvarlimit* or *kvarLimitneg*, respectively, when its active output power is greater than this value;
- *EffCurve*: Inverter efficiency curve. This curve characterizes the variation of the inverter efficiency as a function of the power from PV array, P_{dc}, in per unit of inverter's kVA rating, as can be seen in the Figure 3. For each DC bus voltage, there is an efficiency curve of the inverter,

however, the present version of the model only defines one curve that generally corresponds to the rated voltage of the DC bus. To define this curve in OpenDSS, the user must use the *XYCurve* object, as shown below.

```
New XYCurve.Eff npts=4 xarray=[.1 .2 .4 1.0] yarray=[.86 .9 .93 .97]
```

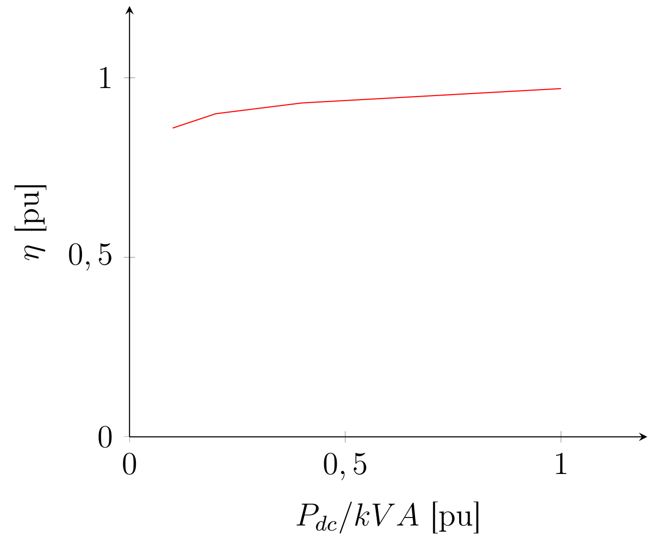

Figure 3: Inverter efficiency curve

1.1.3 Operation conditions properties

The irradiance and temperature on the PV array are the data that define the operating condition.

Irradiance The properties to define the irradiance for a duration of time are:

- *irradiance*: Base value of irradiance in kW/m^2 for QSTS simulations and present irradiance value for static simulations;
- *yearly* or *daily* or *duty*: yearly, daily or duty irradiance curves, respectively. These curves are defined in *pu* of the value defined in the *irradiance* property. To define this curve in OpenDSS, Figure 4, you must use the *LoadShape* object, as shown below.

```
New Loadshape.Irrad npts=24 interval=1
~ mult=[0 0 0 0 0 0 .1 .2 .3 .5 .8 .9 1.0 1.0 .99 .9 .7 .4 .1 0 0 0 0 0]
```

Temperature

- *temperature*: Present temperature on the PV array. This property is used only for static simulations, while *Tshape* is used for QSTS simulations;
- *Tshape*: Temperature curve for a duration of time, in ℃, on the PV array. To define this curve in OpenDSS, the user must use the *Tshape* object, as shown below.

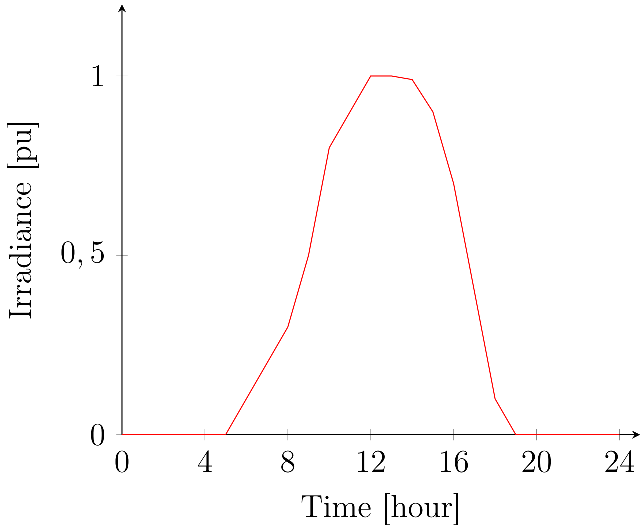

Figure 4: Daily irradiance curve

```
New Tshape.Temp npts=24 interval=1
~ temp=[25 25 25 25 25 25 25 25 35 40 45 50 60 60 55 40 35 30 25 25 25 25 25 25]
```

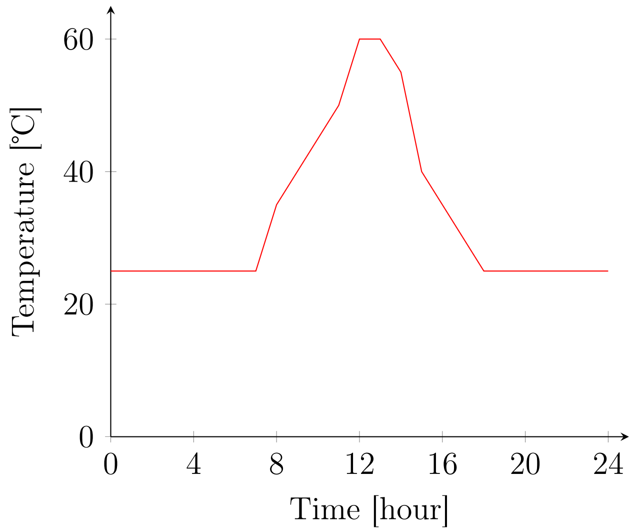

Figure 5: Daily temperature curve

1.2 Interface to circuit model

PVSystems are considered to be PC elements that are modeled by a Norton equivalent in which the constant and linear Norton admittance is included in $\bar{Y}_{system}$ and the nonlinear characteristics are represented by the compensation current sources. The Norton equivalent is calculated based on the present output power delivered to the circuit, as presented in the subsubsection 1.2.1, and few PVSystem properties.

1.2.1 Power delivered to the circuit

The steps taken to obtain the power delivered to the circuit in the time step, t, are presented in this section. This process can be repeated multiple times for the same time step, t, within the control loop if there is an InvControl element controlling this PVSystem.

Step 1: Calculation of the output power from PV array The irradiance at a time step, t, is converted into DC output power from PV array, $P_{dc}[t]$, according to Equation 1.

$$P_{dc}[t] = Pmpp \times irradiance \times irradiance[t] \times PTCurve(Temperature[t]) \tag{1}$$

Where:

- $irradiance$: Base value of irradiance in kW/m^2. For static simulations, this value represents the present irradiance on PV array and therefore there is no need to define $irradiance[t]$;
- $irradiance[t]$: Value of the yearly, daily or duty irradiance curve in the time step, t;
- $PTCurve(Temperature[t])$: Value of the correction factor in the time step t due to the temperature $Temperature[t]$;
- $Temperature[t]$: Value of the temperature curve in time step t. For static simulations, however, the user must set the present temperature $Temperature$ instead of this property.

Besides the inverter efficiency curve, another simplification of the model considers that the ratio of the output power from PV array, P_{dc}, to irradiance is linear, as can be seen in the Simplified curve (red curve) of Figure 6, however, the relationship between them, Real curve, is not constant for the entire range of irradiance values. It should be noted that the Real curve (blue curve) of Figure 6 does not correspond to a measured curve, it is just illustrative.

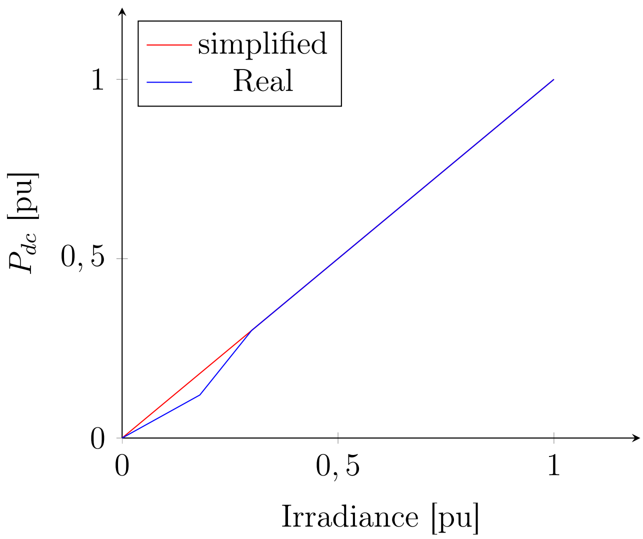

Figure 6: PV array output power vs Irradiance

Step 2: Checking the PV inverter status Checking the inverter status in time step t depends on its state in the previous time step, $t-1$, as shown below.

When the inverter status is OFF in $t-1$, it means that $P_{ac}[t-1]=0$. It is turned ON in t according to Equation 2.

$$Status[t] = \text{ON}, \text{ if } P_{dc}[t] \geq \frac{\%cutin \times kVA}{100} \tag{2}$$

When the inverter status is ON in $t-1$, it means that $P_{ac}[t-1]>0$. It is turned OFF in t according to Equation 3.

$$Status[t] = \text{OFF}, \text{ if } P_{dc}[t] < \frac{\%cutout \times kVA}{100} \tag{3}$$

One model constraint is the need to have *%cutout* less than or equal to *%cutin*.

Step 3: Calculation of the inverter desired active output power The inverter desired active output power is calculated by Equation 4. In this step, the maximum active power limit is also verified by means of the *pctPmpp* property of the PVSystem and the active power limit, $P_{Limit}[t]$, resulting from an eventual operation of the volt-watt function of an InvControl element that would control the PVSystem.

$$P'_{ac}[t] = \begin{cases} 0, \text{ if the inverter status is OFF} \\ P_{LimitMin}[t], \text{ if } P_{dc}[t].EffCurve \geq P_{LimiteMin}[t] \\ P_{dc}[t].EffCurve, \text{ otherwise} \end{cases} \tag{4}$$

Where:

$$P_{LimitMin}[t] = \begin{cases} \min\left(\dfrac{pctPmpp \times Pmpp}{100}, P_{Limit}[t]\right), P_{Limit}[t] \text{ from volt-watt} \\ \dfrac{pctPmpp \times Pmpp}{100}, \text{ otherwise} \end{cases} \tag{5}$$

Step 4: Specification of inverter desired reactive power The inverter desired reactive power, Q'_{ac}, is defined separately from the active power and can be specified as either a fixed kvar value, using the *kvar* property, or as a function of a constant power factor using *PF* property. In the first case, the inverter must try to keep the value of the reactive power constant and equal to *kvar*, regardless of the present value of the output active power. In the second case, which corresponds to the smart inverter function named as constant power factor, the inverter changes the reactive power to maintain the nominal power factor. Another way of setting the desired value of the reactive power, Q'_{ac}, is done by the operation of few functions of the InvControl element, as shown in the section 2.

Step 5: Checking the inverter reactive power limits The desired reactive power after the verification of its limits is presented in equations 6 and 7. Equation 6 is used when the desired reactive power, $Q'_{ac}[t]$, is positive and, Equation 7, when it is negative.

$$Q''_{ac}[t] = \begin{cases} Q'_{ac}[t], \text{ if } Q'_{ac}[t] < |Q_{Limit}[t]| \\ |Q_{Limit}[t]|, \text{ if } Q'_{ac}[t] \geq |Q_{Limit}[t]| \end{cases} \tag{6}$$

$$Q''_{ac}[t] = \begin{cases} -|Q_{Limit_{neg}}[t]|, \text{ if } Q'_{ac}[t] < -|Q_{Limit_{neg}}[t]| \\ Q'_{ac}[t], \text{ if } -|Q_{Limit_{neg}}[t]| \leq Q'_{ac}[t] \end{cases} \tag{7}$$

Equation 8 and Equation 9 present the limits of the reactive power provided and absorbed, respectively, as a function of the active power delivered to the grid.

$$|Q_{Limit}[t]| = \begin{cases} 0, \text{ if } P_{ac}[t] < P_{min} \\ \dfrac{kvarLimit \times P_{ac}[t]}{pctPminNoVars}, \text{ if } P_{min} \leq P_{ac}[t] < P_{max} \\ kvarLimit, \text{ if } P_{ac}[t] \geq P_{max} \end{cases} \tag{8}$$

$$|Q_{Limit_{neg}}[t]| = \begin{cases} 0, \text{ if } P_{ac}[t] < P_{min} \\ \dfrac{kvarLimitNeg \times P_{ac}[t]}{pctPminNoVars}, \text{ if } P_{min} \leq P_{ac}[t] < P_{max} \\ kvarLimitNeg, \text{ if } P_{ac}[t] \geq P_{max} \end{cases} \tag{9}$$

Where:

- $P_{min} = \dfrac{pctPminNoVars \times Pmpp}{100}$
- $P_{max} = \dfrac{pctPminkvarLimit \times Pmpp}{100}$

If the inverter status is OFF, $Q''_{ac}[t]$ can only be non-zero when the *VarFollowInverter* property is set to NO or False.

Step 6: Checking inverter capability The maximum inverter capacity corresponds to its kVA rating. Active and reactive power that are delivered to the grid can be defined according to Equations 10 and 11 when the inverter capacity is not exceeded.

$$P_{ac}[t] = P'_{ac}[t] \tag{10}$$

$$Q_{ac}[t] = Q''_{ac}[t] \tag{11}$$

If its capacity is exceeded, the PV inverter limits the value of $|\bar{S}[t]|$ to its kVA rating, kVA property. As a result, $P'_{ac}[t]$ and/or $Q''_{ac}[t]$ should be reduced to meet one of the following 3 priorities:

- **Power factor priority** Equations 12 and 13 present the values of the active and reactive power that are delivered to the grid when the inverter capacity is exceeded under power factor priority.

$$P_{ac}[t] = kVA * FP \tag{12}$$

$$Q_{ac}[t] = kVA * \sqrt{1 - FP^2} \tag{13}$$

- **Active power priority** Equation 14 presents the new reduced value of the reactive power that is delivered to the circuit when the inverter capacity is exceeded under active power priority. If the active power is greater than or equal to the capacity of the inverter, the new value of the active power that is delivered to the grid becomes the kVA rating and therefore, there is no room for reactive power, according to Equation 15.

$$Q_{ac}[t] = \sqrt{kVA^2 - P_{ac}[t]^2} \tag{14}$$

$$P_{ac}[t] = \begin{cases} kVA, \text{ if } P'_{ac}[t] \geq kVA \\ P'_{ac}[t], \text{ otherwise} \end{cases} \tag{15}$$

- **Reactive power priority** Equation 16 presents the new reduced value of the active power that is delivered to the circuit when the inverter capacity is exceeded under reactive power priority. If the reactive power is greater than or equal to the capacity of the inverter, the new value of the reactive power becomes the capacity of the inverter itself, according to Equation 17, but this can only happen if the inverter is capable of providing or absorbing reactive power without any active power, in other words, when the *VarFollowInverter* property is set to NO or False.

$$P_{ac}[t] = \sqrt{kVA^2 - Q_{ac}[t]^2} \tag{16}$$

$$Q_{ac}[t] = \begin{cases} kVA \times signal(Q''_{ac}[t]), \text{ if } |Q''_{ac}[t]| \geq kVA \\ Q''_{ac}[t], \text{ otherwise} \end{cases} \tag{17}$$

Finally, the apparent power provided by the PVSystem in a time step t can be written according to Equation 18.

$$\bar{S}[t] = P_{ac}[t] + jQ_{ac}[t] \tag{18}$$

1.2.2 Norton equivalent

This section presents the Norton equivalent definition for a three-phase PVSystem connected to the grid through the bus named as Connection Bus, as presented in Figure 7.

By observing Figure 7, the following electric quantities can be defined:

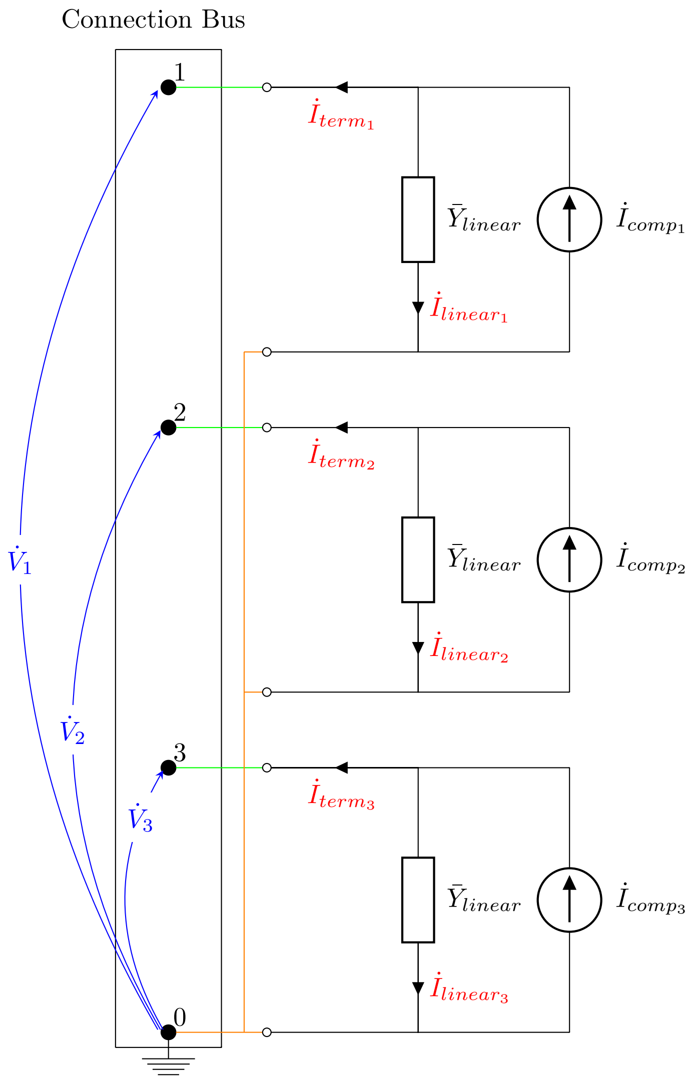

Figure 7: A three-phase PVSystem modeled as a Norton Equivalent

- Nodal voltages:

$$\dot{\mathbf{V}} = \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_3 \end{bmatrix} \tag{19}$$

- Terminal currents:

$$\dot{\mathbf{I}}_{\mathbf{term}} = \begin{bmatrix} \dot{I}_{term_1} \\ \dot{I}_{term_2} \\ \dot{I}_{term_3} \end{bmatrix} \tag{20}$$

- Norton or compensation currents:

$$\dot{\mathbf{I}}_{\mathbf{comp}} = \begin{bmatrix} \dot{I}_{comp_1} \\ \dot{I}_{comp_2} \\ \dot{I}_{comp_3} \end{bmatrix} \tag{21}$$

- Linear currents:

$$\mathbf{\dot{I}_{linear}} = \begin{bmatrix} \dot{I}_{linear_1} \\ \dot{I}_{linear_2} \\ \dot{I}_{linear_3} \end{bmatrix} \tag{22}$$

The normal power flow solution technique in OpenDSS is based on the Equation 23.

$$\mathbf{\dot{I}} = \mathbf{\bar{Y}_{system}} \times \mathbf{\dot{V}} \tag{23}$$

Where:

- $\mathbf{\dot{I}}$: Injected current array;
- $\mathbf{\dot{V}}$: System voltage array.

OpenDSS computes the compensation currents from each PC element to populate the injected current array. Therefore, the main purpose of this section is to understand how OpenDSS calculates the compensation currents from the PVSystem such as, $\dot{I}_{comp_1}$, $\dot{I}_{comp_2}$ and $\dot{I}_{comp_3}$

Firstly, it is possible by means of Kirchhoff's first law to express the compensation currents in function of the linear and terminal currents, as presented in Equation 24.

$$\begin{bmatrix} \dot{I}_{comp_1} \\ \dot{I}_{comp_2} \\ \dot{I}_{comp_3} \end{bmatrix} = \begin{bmatrix} \dot{I}_{term_1} \\ \dot{I}_{term_2} \\ \dot{I}_{term_3} \end{bmatrix} + \begin{bmatrix} \dot{I}_{linear_1} \\ \dot{I}_{linear_2} \\ \dot{I}_{linear_3} \end{bmatrix} \tag{24}$$

The linear currents can be written as a function of the nodal voltages, Equation 25, and the terminal currents as a function of both the three-phase apparent power provided by the PVsystem , $\bar{S}_{3\phi}$, and the nodal voltages, Equation 26. (It is assumed that PVSystem operates with constant power).

$$\begin{bmatrix} \dot{I}_{linear_1} \\ \dot{I}_{linear_2} \\ \dot{I}_{linear_3} \end{bmatrix} = \begin{bmatrix} \bar{Y}_{linear} & 0 & 0 \\ 0 & \bar{Y}_{linear} & 0 \\ 0 & 0 & \bar{Y}_{linear} \end{bmatrix} \cdot \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_3 \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} \dot{I}_{term_1} \\ \dot{I}_{term_2} \\ \dot{I}_{term_3} \end{bmatrix} = \begin{bmatrix} \frac{\bar{S}^*_{3\phi}}{3} & 0 & 0 \\ 0 & \frac{\bar{S}^*_{3\phi}}{3} & 0 \\ 0 & 0 & \frac{\bar{S}^*_{3\phi}}{3} \end{bmatrix} \cdot \begin{bmatrix} 1/\dot{V}^*_1 \\ 1/\dot{V}^*_2 \\ 1/\dot{V}^*_3 \end{bmatrix} \tag{26}$$

The Norton or linear admittance, $\bar{Y}_{linear}$, is calculated using the apparent power provided by the element, according to Equation 27.

$$\bar{Y}_{linear} = \frac{\bar{S}^*_{3\phi}}{V_n^2} \tag{27}$$

Where:

- V_n corresponds the rated voltage of the PVSystem.

The final expression for the calculation of the compensation currents, therefore, is presented in Equation 28.

$$\begin{bmatrix} \dot{I}_{comp_1} \\ \dot{I}_{comp_2} \\ \dot{I}_{comp_3} \end{bmatrix} = \begin{bmatrix} \frac{\bar{S}^*_{3\phi}}{3} & 0 & 0 \\ 0 & \frac{\bar{S}^*_{3\phi}}{3} & 0 \\ 0 & 0 & \frac{\bar{S}^*_{3\phi}}{3} \end{bmatrix} \cdot \begin{bmatrix} 1/\dot{V}^*_1 \\ 1/\dot{V}^*_2 \\ 1/\dot{V}^*_3 \end{bmatrix} + \begin{bmatrix} \frac{\bar{S}^*_{3\phi}}{V_n^2} & 0 & 0 \\ 0 & \frac{\bar{S}^*_{3\phi}}{V_n^2} & 0 \\ 0 & 0 & \frac{\bar{S}^*_{3\phi}}{V_n^2} \end{bmatrix} \cdot \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_3 \end{bmatrix} \tag{28}$$

The compensation currents are calculated as shown above if two conditions are satisfyed. First, the *model* property of the PVSystem is set equal to 1, the default OpenDSS value for constant power operation. And second, each nodal voltage has to be in the range defined by the V_{minpu} and V_{maxpu} properties, which by default are 0.9 and 1.1, respectively. Outside this range, the corresponding phase of the element is modeled as constant impedance.

2 OpenDSS InvControl Element Model

The functionality of inverters modeled in OpenDSS can be separated into two groups. The first consists of the functionalities modeled on the PC element itself, PVSystem and Storage. And the second has the features that are modeled on the OpenDSS InvControl element, the ones described in this section.

This separation is made due to the nature of the functionalities from the point of view of the OpenDSS simulation process. The functions of the first group are able to change the output power value of the PC element without the need for any power flow result and therefore they need just one control iteration of the control loop. On the other hand, the functions of the second group need a power flow result to then check whether or not it is necessary to perform a control action (change the output power of the PC element) and this is done through the control loop.

For InvControl, voltages are sampled to check if any control action is required. These voltages may be the terminal or nodal voltages of the PC element or, alternatively, the line-to-line or line-to-ground voltages of a monitored bus or buses. In the description of the following properties, the term monitored voltage refers to the voltage chosen by means of one of the 2 options above. The subsection 2.4 presents a more detailed explanation of how the monitored voltage is calculated.

To use the InvControl element it is necessary to define two sets of properties, the first one corresponds to the properties that is common to all the smart functions, and the second one corresponds to the properties specific to each smart function.

2.1 Common properties

The common properties that need to be defined in InvControl are listed below:

- *DERList*: Array list of PVSystem and/or Storage elements to be controlled. If not specified, all PVSystems and Storages in the circuit are assumed to be controlled by this control;
- *mode*: Smart inverter function with which the InvControl will control the PC elements specified in *DERList*, according to the options below:
  - *VOLTVAR*: Smart inverter volt-var function. This mode attempts to CONTROL the vars according to one or two volt-var curves, depending on the monitored voltages, present active power output, and the capabilities of the PC element;
  - *VOLTWATT*: Smart inverter volt-watt function. This mode attempts to LIMIT the watts according to one defined volt-watt curve, depending on the monitored voltages and the capabilities of the PC element;
  - *DYNAMICREACCURR*: Smart inverter Dynamic Reactive Current (DRC) function. This mode attempts to increasingly counter deviations by CONTROLLING vars, depending on the monitored voltages, present active power output, and the capabilities of the PC element;
- *Combimode*: Combination of smart inverter functions with which the InvControl will control the PC elements in *DERList*, according to the options below:
  - *VV_VW*: Smart inverter volt-var and volt-watt function;
  - *VV_DRC*: Smart inverter volt-var and DRC function.
- *voltage_curvex_ref*: Base voltage that is considered to calculate the monitored voltage in *pu*, the options are listed below:
  - *rated*: It uses as base voltage the rated voltage of the controlled element;
  - *avg*: It uses an average value as the base voltage. This average value is calculated using the monitored voltage values of previous time steps that are stored in a moving window. This window has a length in units of time, defined in the *avgwindowlen*;
  - *ravg*: It uses as base voltage the rated voltage of the controlled. However, the value that is used to calculate the monitored voltage in *pu* is the average value of the moving window described in option *avg*.
- *avgwindowlen*: The time length of the averaging window;
- *VoltageChangeTolerance*: Tolerance in *pu* of the control loop convergence associated to the monitored voltage in *pu*. This value is compared with the difference of the monitored voltage in *pu* of the current and previous control iterations of the control loop;
- *RateofChangeMode*: Auxiliary option that aims to limit the changes of the desired reactive power and the active power limit between time steps, the alternatives are listed below:
  - *INACTIVE*: It indicates there is no limit on rate of change imposed for either active or reactive power output;
  - *LPF*: A low-pass RC filter is applied to the desired reactive power and/or the active power limit to determine the output power as a function of a time constant defined in the *LPFTau* property;

- *RISEFALL*: A rise and fall limit in the change of active and/or reactive power expressed in terms of *pu* power per second, defined in the *RiseFallLimit*, is applied to the desired reactive power and/or the active power limit.

- *RiseFallLimit*: Limit in *pu* power per second used by the *RISEFALL* option of the *RateofChangeMode* property. The base value for this ramp is defined in the *RefReactivePower* property and/or in *VoltwattYAxis*;
- *LPFTau*: Filter time constant of the *LPF* option of the *RateofChangeMode* property. The time constant will cause the low-pass filter to achieve 95% of the target value in 3 time constants.
- *monBus*: Array list of monitored buses with their nodes;
- *monBusVbase*: Array list of rated voltages of the buses and their nodes presented in the *monBus* property. This list may have different line-to-line and/or line-to-ground voltages;
- *monVoltageCalc*: Options for calculation of monitored voltage in V. These options can be applied to the nodal voltages of the controlled element or the list of voltages of the buses defined in the property *monBus*:
  - *AVG*: Calculates the mean of the voltages;
  - *MAX*: Stores the maximum voltage;
  - *MIN*: Stores the minimum voltage.

  Considering that the list of voltages of the buses defined in *monBus* may present values with different magnitudes, the list that is actually used in this step corresponds to the list that presents its values in V based on the same base voltage which is the rated voltage of the controlled element. Step 3 of subsection 2.4 presents this process in more detail.

2.2 Properties of functions that CONTROL reactive power

The functions that control reactive power of the controlled element have in common a property that defines the base reactive power and two that are associated with the control loop convergence. These functions are volt-var, DRC, volt-watt combined with volt-var and volt-var combined with DRC.

- *RefReactivePower*: Defines the base reactive power for either the provided and absorbed reactive power, according to one of the following options:
  - *VARAVAL*: The base values for the provided and absorbed reactive power are equal to the available reactive power;
  - *VARMAX*: The base values of the provided and absorbed reactive power are equal to the value defined in the *kvarLimit* and *kvarLimitNeg* properties, respectively.
- *VarChangeTolerance*: Tolerance in *pu* of the convergence of the control loop associated with reactive power. For the same control iteration, this value is compared to the difference between the desired reactive power value in *pu* with the output reactive power in *pu* of the controlled element;

- *deltaQ_factor*: Maximum change in *pu* from the prior reactive output power to the desired reactive output power during each control iteration. If it is not set or set equal -1, OpenDSS will use its automatic convergence algorithm.

2.2.1 Properties of smart inverter volt-var function

The following list shows the properties that should be set when the volt-var function is enabled through one of the ensuing three modes: *VOLTVAR*, *VV_VW* OR *VV_DRC*.

- *vvc_curve1*: Name of the *XYCurve* object containing the volt-var curve. The positive values of the y axis of the volt-var curve represent values in *pu* of the provided base reactive power. The negative values of the axis y, are values in *pu* of the absorbed base reactive power. Those defined in the *RefReactivePower* property;

2.2.2 Properties of smart inverter DRC function

The following list shows the properties that should be set when the DRC function is enabled through one of the ensuing two modes: *DYNAMICREACCURR* or *VV_DRC*.

- *DbvMin*: The per-unit voltage lower limit of the deadband of the DRC function. When the monitored voltage is within this deadband, no reactive power can be provided or absorbed;
- *DbvMax*: The per-unit voltage upper limit of the deadband of the DRC function. When the monitored voltage is within this deadband, no reactive power can be provided or;
- *ArGraLowV*: Ratio between the reactive power provided or absorbed and the difference between the monitored voltage and the average voltage of the averaging window of the DRC function. This rate is applied when the monitored voltage is less than *DbvMin*;
- *ArGraHiV*: Ratio between the reactive power provided or absorbed and the difference between the monitored voltage and the average voltage of the averaging window of the DRC function. This rate is applied when the monitored voltage is greater than *DbvMax*;
- *DynReacavgwindowlen*: The time length of the averaging window of the DRC function.

2.3 Properties of functions that LIMIT active power

The functions that limit the active power of the controlled element have two properties in common that are associated with the convergence of the control loop. The functions that limit active power are volt-watt and volt-watt combined with volt-var.

- *ActivePChangeTolerance*: Tolerance in *pu* of the convergence of the control loop associated with active power. For the same control iteration, this value is compared to the difference between the active power limit in *pu* resulted from the convergence process and the one resulted from the volt-watt function;

- *deltaP_factor*: Maximum change in *pu* from the prior active power limit to the one resulted from the volt-watt curve during each control iteration. If it is not set or set equal -1, OpenDSS will use its automatic convergence algorithm.

2.3.1 Properties of smart inverter volt-watt function

The following list shows the properties that should be set when the volt-watt function is enabled through one of the ensuing two modes: *VOLTWATT* or *VV_VW*.

- *voltwatt_curve*: Name of the *XYCurve* object containing the volt-watt curve;
- *VoltwattYAxis*: Defines the y axis in *pu* of the volt-watt curve. The options for this property are listed below:
  - *PMPPPU*: The y axis corresponds to the value in *pu* of *Pmpp* property of the PVSystem;
  - *PAVAILABLEPU*: The y axis corresponds to the value in *pu* of the available active power of the PVSystem;
  - *PCTPMPPPU*: The y axis corresponds to the value in *pu* of the power *Pmpp* multiplied by $\frac{1}{100}$ of the *pctPmpp* property of the PVSystem;
  - *KVARATINGPU*: The y axis corresponds to the value in *pu* of the *kVA* property of the PVSystem.

2.4 InvControl operation in the control loop

This section shows the operation of the control loop when only one InvControl is the control element in the circuit.

Figure 8 shows the block diagram of the control loop and as can be seen there are no delayed control actions when only InvControl elements are considered.

The operation of InvControl is described for the time step t and control iteration j. The following are the main steps in the control loop that are described:

1. Step 1: Power Flow;
2. Step 2: Sample the Control Elements;
3. Step 3: Control Actions for t Done?;
4. Step 4: Execute Control Actions for t.

Step 1: Power Flow To perform the power flow, OpenDSS calculates the PC compensation currents of the PC elements using their powers. Therefore, the active power, $P_{ac}[t]_j$, and reactive power, $Q_{ac}[t]_j$, of the PVSystem element are calculated according to the six steps described in 1.2.1.

As a results of the power flow, all system voltages are calculated.

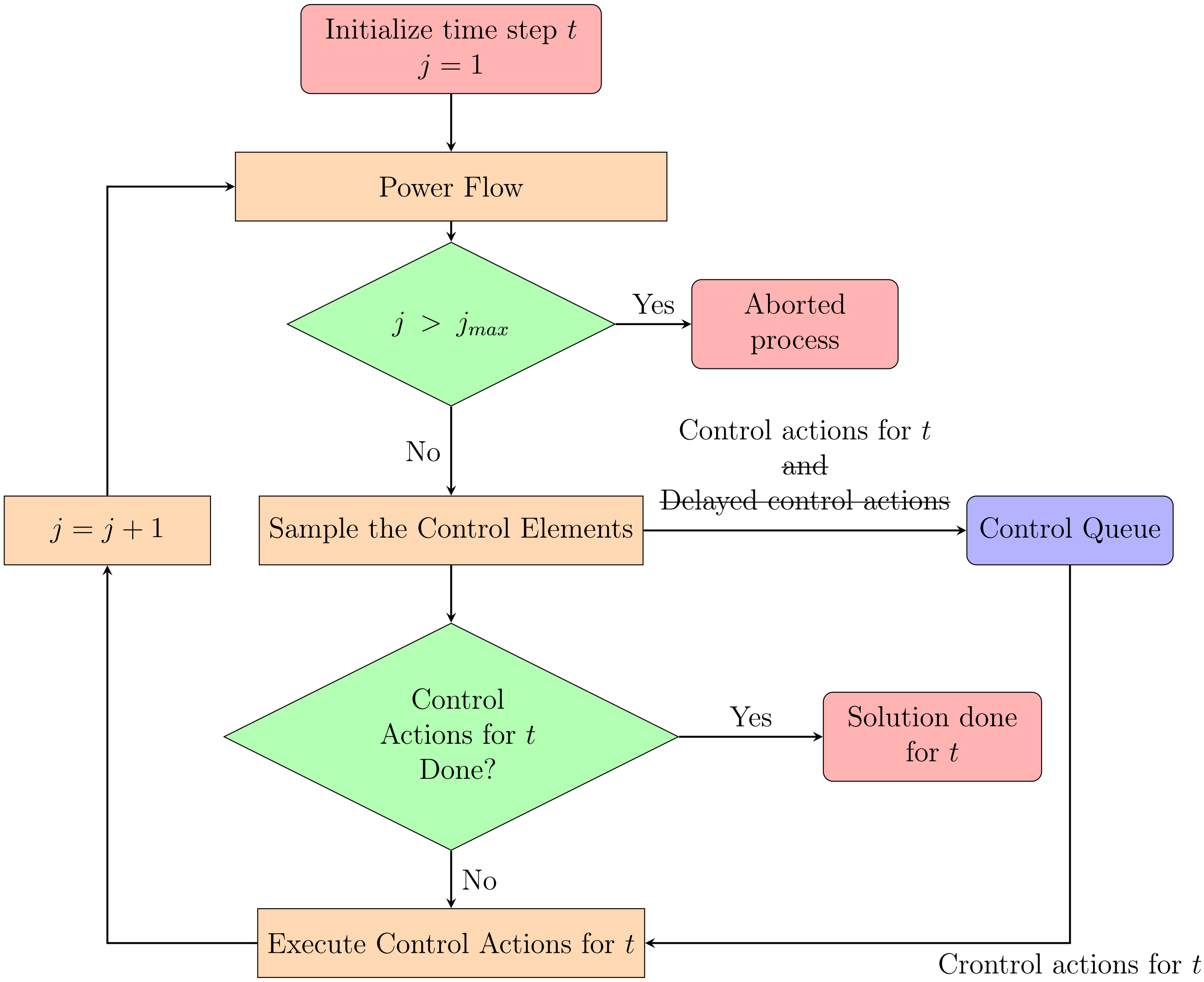

Figure 8: Block diagram of the control loop when only *InvControl* elements are considered

Step 2: Sample the Control Elements The InvControl samples voltages that are used to verify the need for a control action. If this is the case, InvControl includes this action in the control list. The processes performed in this step are presented in the following items:

- **Calculation of the monitored voltage in volts**

  The voltages sampled by InvControl can be the nodal voltages of the controlled element or line-to-line and/or line-to-ground voltages of monitored bus or buses.

  By default the InvControl samples the nodal voltages of the controlled element. For example, for a three-phase element, the voltages monitored in V can be written according to Equation 29.

$$[V_{mon_1}[t]_j \ , \ V_{mon_2}[t]_j \ , \ V_{mon_3}[t]_j] \tag{29}$$

  In order to sample voltages of monitored bus or buses, the *monBus* and *monBusVBase* properties of InvControl must be used. The *monBus* property stores the monitored buses and their nodes defined in a list and the *monBusVBase* property stores the list with the corresponding base voltages for each of these monitored buses. For instance, Equation 30 shows the list with two monitored voltages and Equation 31 the list with their respective base voltages.

$$[V_{monBusA_1}[t]_j \ , \ V_{monBusB_{12}}[t]_j] \tag{30}$$

$$[V_{base_1} \ , \ V_{base_2}] \tag{31}$$

Where:

- $V_{monBusA_1}[t]_j$ corresponds to the line-to-ground voltage of the node 1 of the bus A;
- $V_{monBusB_{12}}[t]_j$ corresponds to the line-to-line voltage between the nodes 1 and 2 of the bus B.

The values of the voltages of Equation 30 can present base voltages with distinct values and therefore, this list must have its values corrected for a same base voltage which correspond to the rated voltage of the controlled element, as presented in Equation 32. This is necessary so that the calculation of the monitored voltage in V is made considering voltages of the same order of magnitude.

$$[V_{monBusA_1}[t]_j \times \frac{V_{element_{base}}}{V_{base_1}} \ , \ V_{monBusB_{12}}[t]_j \times \frac{V_{element_{base}}}{V_{base_2}}] \tag{32}$$

Where:

- $V_{element_{base}}$ corresponds to the rated voltage of the controlled element which is the value of kV property of the *PVSystem*.

A unique monitored voltage in V is calculated from one of the two lists present in Equations 29 or 32. The options for this selection are present in the *monVoltageCalc* property and are applied in the list to calculate the maximum, minimum or average value, according to the options described below:

- For *AVG*, the average monitored voltage is calculated by applying the mean in the list;
- For *MAX*, the monitored voltage is the maximum value present in the list;
- For *MIN*, the monitored voltage is the minimum value present in the list.

Thus, as a result, the monitored voltage in V is defined as $V_{mon}[t]_j$.

- **Calculation of the per-unit monitored voltage**

The monitored voltage in *pu*, $v_{mon}[t]_j$, applied to the smart inverter volt-var and volt-watt functions, is calculated using the monitored voltage in V, $V_{mon}[t]_j$, through one of the options in the *voltage_curvex_ref* property shown below:

- For *rated*, the monitored voltage in *pu* is calculated according to Equation 33.

$$v_{mon}[t]_j = \frac{V_{mon}[t]_j}{V_{element_{base}}} \tag{33}$$

- For *avg*, the monitored voltage in *pu* is calculated according to Equation 34.

$$v_{mon}[t]_j = \frac{V_{mon}[t]_j}{V_{mon_{avg}}[t]} \tag{34}$$

- For *ravg*, the monitored voltage in *pu* is calculated according to Equation 35.

$$v_{mon}[t]_j = \frac{V_{mon_{avg}}[t]}{V_{elemento_{base}}} \tag{35}$$

Where:

- $V_{mon_{avg}}[t]$ is the mean value of the monitored voltages in *pu* of the previous m time steps that can be stored in the moving window with length defined in the *avgwindowlen* property, as can be seen in Equation 36.

$$V_{mon_{avg}}[t] = \sum_{k=1}^{m} v_{mon}[t-k]_j \tag{36}$$

For instance, an averaging window with three time step length, $V_{mon_{avg}}[t]$ is calculated according to Equation 37.

$$V_{mon_{avg}}[t] = \frac{1}{3} \times (V_{mon_{avg}}[t-1] + V_{mon_{avg}}[t-2] + V_{mon_{avg}}[t-3]) \tag{37}$$

For smart inverter DRC functions, the monitored voltage in *pu* is always calculated according to Equation 38.

$$v_{mon_{drc}}[t]_j = \frac{V_{mon}[t]_j}{V_{elemento_{base}}} \tag{38}$$

- **Checking the need for control action**

If the convergence criteria of the InvControl are satisfied, no control action is sent to the control list in the current iteration, j. However, for the first control iteration, $j = 1$, always the action is sent, so the criteria given below are only valid for $j > 1$.

Voltage convergence criterion The voltage criterion is performed by comparing the monitored voltage in *pu* between the control iterations j and $j-1$, according to Equation 39.

$$v_{mon}[t]_j - v_{mon}[t]_{j-1} < \xi_v \tag{39}$$

If the DRC function is selected, the Equation 40 presents the convergence criterion that needs to be satisfied.

$$v_{mon_{drc}}[t]_j - v_{mon_{drc}}[t]_{j-1} < \xi_v \tag{40}$$

Where ξ_v corresponds to the tolerance defined in the *VoltageChangeTolerance* property.

If the combined volt-var and DRC function is enabled, both the criterion of Equation 39 and Equation 40 must be satisfied.

Reactive power convergence criterion The convergence criterion for reactive power is performed by comparing the desired reactive power in *pu*, $q_{Dend}[t]_j$, and the reactive power in *pu* of the controlled element, both from the control iteration j, as Equation 41.

$$\left| q_{Dend}[t]_j - \frac{Q_{ac}[t]_j}{Q_{base}[t]_{j-1}} \right| < \xi_q \text{ if } Q_{base}[t]_{j-1} \geq 0 \tag{41}$$

$$\left| q_{Dend}[t]_j - \frac{Q_{ac}[t]_j}{Q_{base_{neg}}[t]_{j-1}} \right| < \xi_q \text{ if } Q_{base}[t]_{j-1} < 0 \tag{42}$$

Where:

- ξ_q corresponds to the tolerance defined in the *VarChangeTolerance* property;
- $q_{Dend}[t]_j$ is discussed in the Equation 69 of Step 4;
- $Q_{base}[t]_{j-1}$ and $Q_{base_{neg}}[t]_{j-1}$ are the base reactive power values for the provided and absorbed reactive powers from the previous control iteration, $j-1$, respectively. The calculation of these values is presented in Step 4.

Active power convergence criterion The convergence criterion for active power is performed by comparing the active power limit in *pu*, $p_{Lend}[t]_j$, with the active power limit in *pu* which is the result of the convergence process, according to Equation 43.

$$\left| \frac{P_{Limit}[t]_j}{P_{base}[t]_{j-1}} - p_{Lend}[t]_j \right| < \xi_p \tag{43}$$

Where:

- ξ_p corresponds to the tolerance defined in the *ActivePChangeTolerance* property;
- $p_{Lend}[t]_j$ is discussed in the Equation 68 of Step 4;
- $P_{base}[t]_{j-1}$ is the base active power value from the provius control iteration, $j-1$. The calculation of these values is presented in Step 4.

It is important to notice that this comparison does not use the output active value of the controlled element. The reason is that the nature of the volt-watt function is to set a active power limit instead of a desired value. Therefore, the convergence criterion for active power could be satisfied with an output active power of the controlled element less than the active power limit set by the volt-watt function.

Step 3: Control Actions for t Done? The control loop converges for a time step t when there is no control action in the control list for this time step. Otherwise, the process proceeds to Step 4 described below.

Step 4: Execute Control Actions for t The control action for InvControl is performed by changing the reactive power and/or limiting the active power of the PVSystem. The items performed in this step are described below:

- **Calculation of base values of power**

  Two base values of reactive power are defined, one used when the reactive power is positive or provided. And another, when the reactive power is negative or absorbed. These values depend on the options of the *RefReactivePower* property as shown below:

  - For *VARMAX*, the base values of reactive power are calculated according to Equation 44 and Equation 45 to be used when the reactive power is positive (capacitive) and negative (inductive), respectively.

$$Q_{base}[t]_j = kvarLimit \tag{44}$$

$$Q_{base_{neg}}[t]_j = kvarLimitNeg \tag{45}$$

  - For *VARAVAL*, both base values are calculated according to Equation 46.

$$Q_{base}[t]_j = Q_{base_{neg}}[t]_j = \sqrt{kVA^2 - P_{ac}^2[t]_j} \tag{46}$$

    If $Q_{base}[t]_j$ calculated above results in 0, then $Q_{base}[t]_j$ is set to be equal to *kvarLimit*.

  The base value of active power depends on the options of the *VoltwattYAxis* property as shown below:

  - For *PMPPPU*, the base value is equal to the *Pmpp* property of the PVSystem, as can be seen in the Equation 47.

$$P_{base}[t]_j = Pmpp \tag{47}$$

  - For *PAVAILABLEPU*, the base value corresponds to the available active power of the PVSystem in the time step t, which is calculated according to Equation 48.

$$P_{base}[t]_j = P_{dc}[t].EffCurve \tag{48}$$

Where $P_{dc}[t] = Pmpp \times irradiance \times irradiance[t] \times PTCurve(Temperature[t])$, as shown in Equation 1.

- For *PCTPMPPPU*, the base value is calculated according to Equation 49.

$$P_{base}[t]_j = \frac{pctPmpp \times Pmpp}{100} \tag{49}$$

- For *KVARATING*, the base value corresponds to the kVA rating of the inverter, as can be seen in the Equation 50.

$$P_{base}[t]_j = kVA \tag{50}$$

- **Calculation of the smart inverter function**

The goal of this item is to calculate the desired reactive power, $q_{Dfun}[t]_j$, and/or the active power limit, $p_{Lfun}[t]_j$, according to the modes below:

Smart Inverter volt-var Function For the smart inverter volt-var function, the monitored voltage in pu is applied , $v_{mon}[t]_j$, in the volt-var curve defined in the *vvc_curve1* property to obtain the desired reactive power value in pu, $q_{Dfun}[t]_j$. The Figure 9 below shows an example of a user-defined volt-var curve.

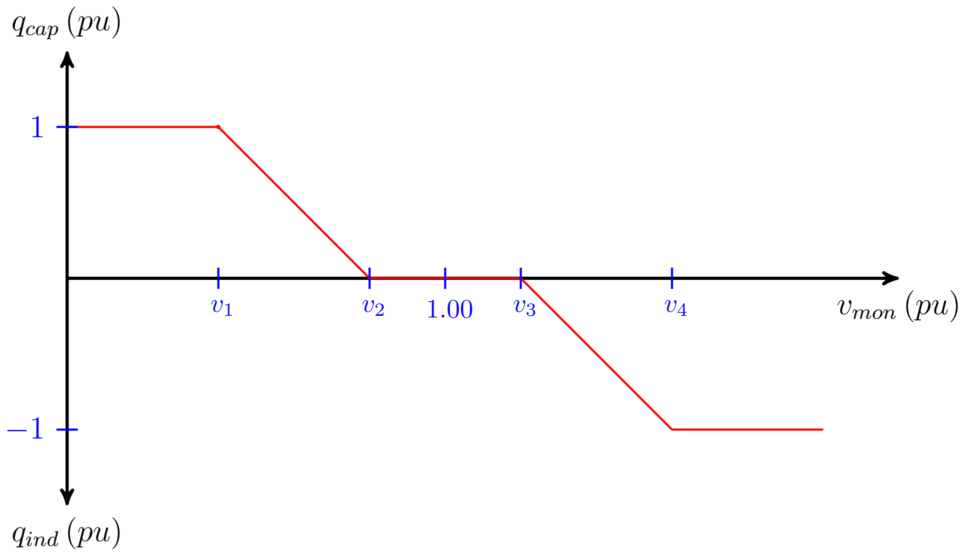

Figure 9: volt-var curve

The desired reactive power in pu can be written according to Equation 51.

$$q_{Dfun}[t]_j = f_{vv}(v_{mon}[t]_j) \tag{51}$$

Smart inverter DRC function For the smart inverter DRC function, the desired reactive power value is set according to Equation 52.

$$q_{Dfun}[t]_j = \begin{cases} -\Delta v_{mon_{drc}}[t]_j \times ArGraLowV \text{ if } v_{mon_{drc}}[t]_j < DbvMin \\ -\Delta v_{mon_{drc}}[t]_j \times ArGraHiV \text{ if } v_{mon_{drc}}[t]_j > DbvMax \\ 0, \text{ otherwise} \end{cases} \tag{52}$$

Where the voltage difference is calculated according to Equation 53.

$$\Delta v_{mon_{drc}}[t]_j = v_{mon_{drc}}[t]_j - v_{window_{drc}}[t]_j \tag{53}$$

And $v_{window_{drc}}[t]_j$ corresponds to the mean of the voltages of the averaging window of the DRC function. This average voltage is calculated according to Equation 54 considering that the averaging window is capable of storing the voltages of m previous time steps. The time-scale length of this window is set in the *DynReacavgwindowlen* property.

$$v_{window_{drc}}[t]_j = \sum_{k=1}^{m} v_{mon_{drc}}[t-k]_j \tag{54}$$

Combined Smart inverter volt-var and DRC function For the combined mode of the smart volt-var and DRC functions, the desired reactive power value is the sum of the result of Equation 51 and Equation 52, according to Equation 55.

$$q_{Dfun}[t]_j = f_{vv}(v_{mon}[t]_j) + \begin{cases} -\Delta v_{mon_{drc}}[t]_j \times ArGraLowV \text{ if } v_{mon_{drc}}[t]_j < DbvMin \\ -\Delta v_{mon_{drc}}[t]_j \times ArGraHiV \text{ if } v_{mon_{drc}}[t]_j > DbvMax \\ 0, \text{ otherwise} \end{cases} \tag{55}$$

Smart inverter volt-watt function For the smart inverter volt-watt function, the monitored voltage in *pu* is applied, $v_{mon}[t]_j$, in the volt-watt curve defined in the *voltwatt_curve* property to obtain the active power limit value in *pu*, $p_{Lfun}[t]_j$. The Figure 10 shows an example of a user-defined volt-watt curve.

The active power limit value in *pu* is calculated as shown in the Equation 56.

$$p_{Lfun}[t]_j = f_{vw}(v_{mon}[t]_j) \tag{56}$$

- **Checking auxiliary options**

  The auxiliary options that can be selected are intended to limit the variations of the active power limit and/or the desired reactive power during time steps. As a result of this verification, $p_{Lopc}[t]_j$ and $q_{Dopc}[t]_j$ are defined.

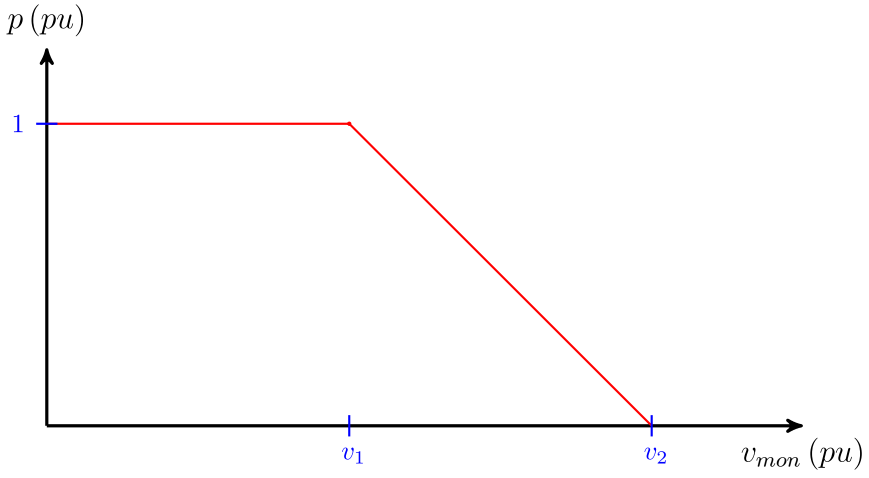

Figure 10: volt-watt curve

Standard The standard option (default) does not limit the variation of the active power limit and the deserid reactive power, as can be seen in the Equation 57 and Equation 58.

$$p_{Lopc}[t]_j = p_{Lfun}[t]_j \tag{57}$$

$$q_{Dopc}[t]_j = q_{Dfun}[t]_j \tag{58}$$

LPF This option consists of applying the active power limit, $p_{Lfun}[t]_j$, and/or the desired reactive power, $q_{Dfun}[t]_j$, as input of a first-order low pass filter with time constant equal to τ. As a result, the values of the options are set according to Equations 59 and 60. These equations present as initial conditions the results of these equations in the previous time step, $p_{Lopc}[t-1]$ and $q_{Dopc}[t-1]$

$$p_{Lopc}[t]_j = p_{Lfun}[t]_j \times (1 - e^{-\frac{t}{\tau}}) + p_{Lopc}[t-1] \times e^{-\frac{t}{\tau}} \tag{59}$$

$$q_{Dopc}[t]_j = q_{Dfun}[t]_j \times (1 - e^{-\frac{t}{\tau}}) + q_{Dopc}[t-1] \times e^{-\frac{t}{\tau}} \tag{60}$$

Where τ is the value set in the *LPFTau* property.

RF This option limits the change of the active power limit and the desired reactive power between time steps to a maximum value. Equations 61 and 62 present the application of this option.

$$p_{Lopc}[t]_j = \begin{cases} p_{Lopc}[t-1] + \frac{\Delta p}{\Delta t}_{RF} \times \Delta t, \text{ if } (p_{Lfun}[t]_j - p_{Lopc}[t-1]) > \frac{\Delta p}{\Delta t}_{RF} \times \Delta t \\ p_{Lopc}[t-1] - \frac{\Delta p}{\Delta t}_{RF} \times \Delta t, \text{ if } (p_{Lfun}[t]_j - p_{Lopc}[t-1]) < -\frac{\Delta p}{\Delta t}_{RF} \times \Delta t \\ p_{Lfun}[t]_j, \text{ otherwise} \end{cases} \tag{61}$$

$$q_{Dopc}[t]_j = \begin{cases} q_{Dopc}[t-1] + \frac{\Delta q}{\Delta t}_{RF} \times \Delta t, \text{ if } q_{Dfun}[t]_j - q_{Dopc}[t-1] > \frac{\Delta q}{\Delta t}_{RF} \times \Delta t \\ q_{Dopc}[t-1] - \frac{\Delta q}{\Delta t}_{RF} \times \Delta t, \text{ if } q_{Dfun}[t]_j - q_{Dopc}[t-1] < -\frac{\Delta q}{\Delta t}_{RF} \times \Delta t \\ q_{Dfun}[t]_j, \text{ otherwise} \end{cases} \tag{62}$$

Where:

- $\dfrac{\Delta p}{\Delta t}_{RF}$ and $\dfrac{\Delta q}{\Delta t}_{RF}$ are the value set in the *RiseFallLimit* property;
- Δt is the stepsize of the QSTS simulation. On other words, it corresponds the difference between two consecutive time steps, t and $t-1$, in a time scale.

- **Checking inverter limits**

  The convergence process must use values that respect the limits presented in the modeling of the PC element, in this sense, the values of the active power limit and the desired reactive power are calculated considering the restrictions of the inverter. As a result of this check, $p_{Llim}[t]_j$ and $q_{Dlim}[t]_j$ are defined.

  Equation 63 presents the calculation for the mode that have the smart inverter volt-watt function.

$$p_{Llim}[t]_j = \min\left(\frac{\sqrt{kVA^2 - Q_{ac}^2[t]_j}}{P_{base}[t]_j}, \frac{pctPmpp \times Pmpp}{100 \times P_{base}[t]_j}\right) \tag{63}$$

  For modes that have the smart inverter volt-var and/or DRC functions, the checking the inverter limits depends of the options of the *RefReactivePower* property, as can be seen below:

  - For *VARMAX*, the calculation of the reactive power limit of the inverter is performed by means of Equation 64 when the inverter is configured to give priority to active power.

$$q_{Dlim}[t]_j = \begin{cases} \min\left(\dfrac{\sqrt{kVA^2 - P_{ac}^2[t]_j}}{Q_{base}[t]_j}, \dfrac{|Q_{Limit}[t]_j|}{Q_{base}[t]_j}\right) \text{ if } q_{Dopc}[t]_j > 0 \\ \min\left(\dfrac{\sqrt{kVA^2 - P_{ac}^2[t]_j}}{Q_{base_{neg}}[t]_j}, \dfrac{|Q_{Limit_{neg}}[t]_j|}{Q_{base_{neg}}[t]_j}\right) \text{ if } q_{Dopc}[t]_j < 0 \end{cases} \tag{64}$$

    On the other hand, with priority for reactive power, the calculation follows Equation 65.

$$q_{Dlim}[t]_j = \begin{cases} \dfrac{|Q_{Limit}[t]_j|}{Q_{base}[t]_j} \text{ if } q_{Dopc}[t]_j > 0 \\ \dfrac{|Q_{Limit_{neg}}[t]_j|}{Q_{base_{neg}}[t]_j} \text{ if } q_{Dopc}[t]_j < 0 \end{cases} \tag{65}$$

  - For *VARAVAL*, the calculation of the reactive power limit is also performed using Equation 65.

Where $|Q_{Limit}[t]_j|$ and $|Q_{Limit_{neg}}[t]_j|$ are calculated in Step 5 of subsubsection 1.2.1.

- **Calculation of final/end values for the convergence process**

  In order to define the most restrictive values to be used in the convergence process. The final values of the active power limit and the desired reactive power are calculated according to Equation 66 and Equation 67, respectively.

$$p_{Lend}[t]_j = \min\left(q_{Dopc}[t]_j, q_{Dlim}[t]_j\right) \tag{66}$$

$$q_{Dend}[t]_j = \min\left(q_{Dopc}[t]_j, q_{Dlim}[t]_j\right) \times signal(q_{Dopc}[t]_j) \tag{67}$$

- **Convergence process**

  The values calculated in the Equations 66 and 67 are not directly applied to set the new active power limit and/or desired reactive power of the PVSystem. The reason is trying to avoid instability in the convergence process. Therefore, Equations 68 and 69 present the values in kW and $kvar$ of the active power limit and the desired reactive power that are used to update the values of the controlled element.

$$P_{Limit}[t]_j = P_{ac}[t]_j + (p_{Lend}[t]_j \times P_{base}[t]_j - P_{ac}[t]_j) \times \Delta P \tag{68}$$

$$Q_{Desired}[t]_j = \begin{cases} Q_{ac}[t]_j + (q_{Dend}[t]_j \times Q_{base}[t]_j - Q_{ac}[t]_j) \times \Delta Q \text{ if } q_{Dend}[t]_j \geq 0 \\ Q_{ac}[t]_j + (q_{Dend}[t]_j \times Q_{base_{neg}}[t]_j - Q_{ac}[t]_j) \times \Delta Q \text{ if } q_{Dend}[t]_j < 0 \end{cases} \tag{69}$$

  Where:

  - ΔP corresponds to the value set in the *deltaP_factor* property;
  - ΔQ corresponds to the value set in the *deltaQ_factor* property.

  However, if the user wants that OpenDSS takes care of ΔP and/or ΔQ values itself, the user can define them as -1 or just not define any value for them.

- **Updates the powers of the PVSystem element**

  The six steps described in subsubsection 1.2.1 are performed again, however, in Step 3 it is considered the active power limit value, $P_{Limit}[t]_j$, and, in Step 4, the value of the desired reactive power, Q'_{ac}, is the value of $Q_{Desired}[t]_j$.

  Then, the control iteration $j+1$ starts in the first step of the control loop, which is the power flow.