TO: WECC REMTF

FROM: POUYAN POURBEIK, PEACE-PLLC; PPOURBEIK@PEACE-PLLC.COM

SUBJECT: PROPOSAL FOR DER_A MODEL

DATE: 10/11/16 (REVISED 11/16/16; 3/6/17, 3/15/17; 3/28/17; 3/29/17; 3/31/17; 4/17/17; 10/5/17; 11/9/17; 2/9/18; 2/15/18; 3/9/18; 7/17/18; 8/29/18; 9/11/18; 6/19/19[1])

CC: A. GAIKWAD, E. FARANTATOS AND J. BOEMER, EPRI

Recent efforts of the WECC LMTF and REMTF have resulted in pursuing modularization of the WECC composite load model structure. Thus, one of the ideas has been, as also recently proposed by NERC [1], to have various distributed generation (DG) models that can be plugged into the composite load model at the end of the distribution feeder model, as shown in Figure 1. Here we are interested primarily in what is defined in [1] as Retail-Scale Distributed Energy Resources (R-DER): distributed energy resources that offset customer load. Although, conceptually this model could also be used for the U-DER [1]. These types of DG are primarily residential and commercial distributed energy resources, and the majority of these types of DG is photovoltaic (PV) generation. To date the best model available for distributed PV generation has been the *pvd1* model that was developed in WECC and exists in many of the commercial software tools used in WECC. However, as discussed at the last several WECC REMTF meetings the *pvd1* model is quite limited as it is not able to represent some of the various functionalities that are at present under discussion in the IEEE 1547 standard development process and also discussed in the California Rule 21 document(s). To that end an initial proposal for a so-called *pvd2* model was developed and shared at the WECC REMTF and MVWG meeting in November 2016 – the first revision of this memo.

There have been significant and valuable input provided in the development of this document by the following individuals, for which we are truly grateful:

Kara Clark, NREL
Jens Boemer, EPRI
Roberto Favela, El Paso Electric
Deepak Ramasubramanian, EPRI
Anish Gaikwad, EPRI
Juan Sanchez-Gasca, GE
Jay Senthil, Siemens PTI
Spencer Tacke, MID
Song Wang, PacifiCorp
Jamie Weber, PowerWorld

---

[1] A very minor editorial changes has been made to the parameter list to be consistent with the final implementation of the model in several commercial software platforms. The parameter *typeflag* has been changed such that 1 implies a generator and 0 implies a storage device.

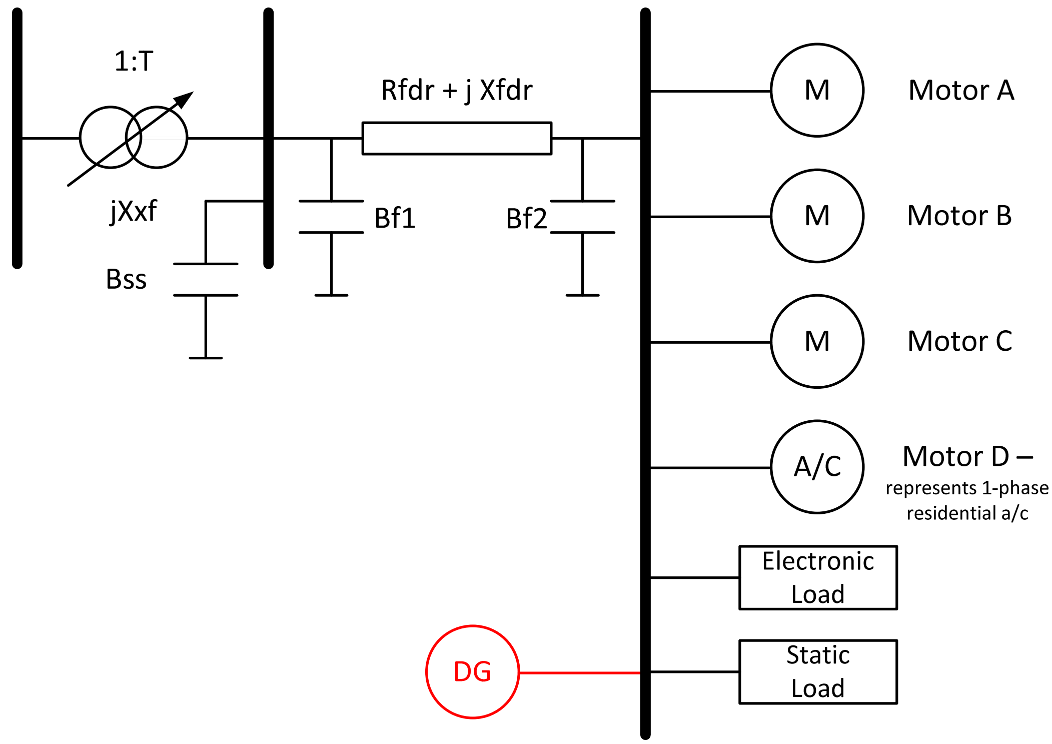


Figure 1: Location of DG plugging into the composite load model.

The *DER_A* Model Proposal

The recently developed 2nd generation renewable energy system (RES) generic models include also a PV model [2], and energy storage [3]. Those models are, however, intended primarily for large scale PV plants connected at the transmission level (or perhaps U-DER as defined in [1]). A complete PV plant model using the generic RES models would incorporate the *regc_a*, *reec_b* and *repc_a* model, as well as the *lhvrt* and *lhfrt* models. This constitutes a total of five (5) modules, one-hundred and twenty-one (121) parameters and sixteen (16) states. Thus, the PV model is as complex as the composite load model itself. It is thus clear that the large-scale PV generic model is perhaps too complex for use with the composite load model. Thus, we need a simpler model but one that has more features than the existing *pvd1* model. With this in mind, the model shown in Figure 2 is proposed as the new *der_a* model.

The complete parameter list for the propose *der_a* model is given in Table 1. The salient features of the model are as follows:

1. The model is actually based on taking the large-scale generic PV model (*regc_a*, *reec_b*, *repc_a*) and reducing the models significantly and lumping the reduced models into a single model.

2. The final model has 48 parameters and 10 states, which is roughly 1/3 of the number of parameters of the full large-scale PV generic model. None-the-less, it preserves a significant number of its features, namely:

    a. Frequency and voltage control emulation, with asymmetric deadband. The voltage control only allows for proportional control, which is consistent with most of the discussions within IEEE 1547.

b. It also allows for constant power factor and constant Q-control.

c. This model is intended primarily to be used as an aggregated model of a large number of distributed generators. Thus, the parameters *Vrfrac*, *vl0*, *vl1*, *vh0*, *vh1, tvl0*, *tvl1*, *tvh0* and *tvh1* collectively allow for emulation of partial tripping of the aggregated DG model. This is explained in detail, per the pseudo code for this function in the appendix, and associated Figure 3. In this case the linear drop-off (shown in Figure 3) is intended to emulate the gradient of voltage along the feeder.

d. The frequency tripping is modeled in simple terms. If frequency goes below *fl* for more than *tfl* seconds, then the entire model will trip. If frequency goes above *fh* for more than *tfh* seconds, then the entire model will trip. This block is disabled, if voltage is below *Vpr*, to avoid tripping on frequency spikes (as calculated in simulation) due to sudden voltage drops. This is all depicted in Figure 2 and in an expanded view in Figure 4.

e. It allows for modeling ramp-rate limits on the real-power recovery following a fault or during primary-frequency response and also models a basic current limit with P/Q priority options.

f. To allow for the model to absorb, as well as generate, real power (i.e. having an additional flag, *typeflag*, parameter that can be 0 or 1. If *typeflag* is1 (or any number other than 0) this means it is a generator, and if it is 0 it is an energy storage device. This will allow users to use this model to emulate distributed energy storage. **<u>WARNING:</u> it should be understood that the model does not incorporate the emulation of charging and discharging (as is done in the *reec_c* model). Thus, if this model is used to simulated distributed energy storage it must be understood that this model is not appropriate for simulations that would span over a time period that is a significant portion of the charging/discharging capacity of the battery. For example, if a distributed energy storage device is capable of providing many minutes to hours of power output (or charging) then in a 10 to 30 second simulation (typical of planning studies) there would be little impact on the state-of-charge of the battery and so this model is quite adequate. However, if the battery energy storage device has only a few seconds worth of capacity, then this model is not appropriate.**

g. In the portion of the model associated with modeling primary-frequency response, the feedback signal (*Pgen*) is taken from the power-order (*Pord*) and not the terminal of the model. The reason for this is as follows. In steady-state, with frequency at its nominal value, the error into the proportional-integral controller (*Kpg*, *Kig)* is zero. The power reference *Pref* will initialize to *Pgen,* and the frequency error is zero. Now if *Freq_flag* = 1 and a fault occurs nearby, which results through the action of the *Vrfrac* logic of partial tripping of the "aggregated" DER, then the terminal electrical power of the *DER_A* will go down. Now if this value were feedback to *Pgen*, then the error into the proportional-integral controller (*Kpg*, *Kig)* would now become positive and *Pord* will increase until it hits *Pmax*, or until the electrical power output of the model is again equal to *Pref*, whichever comes first. This is not appropriate, since there has been no system frequency deviation and thus we do not want an increase in the power output of the model, and we also do not want the model to restore the power lost due to partial tripping effected by *Vrfrac*. Therefore, by taking the power feedback from the power-order (*Pord*) prior to the *Vrfrac* block, this problem is avoided. This ensures that *Pord* is always equal to *Pref*, which is actually what we wish to achieve, and *Vrfrac* acts independently on *Ipcmd*. Furthermore, the user should be allowed to set *Tpord* and *Tp* to zero (0). By doing this and setting *Kpg* = 0 and using a non-zero value of *Kig*, a simple proportional only droop-control can be effected, since the closed loop around *Pord* in this case creates a simple time-constant equal to *1/Kig*. **Important Note:** in this case, i.e. when Tpord = Tp = 0, Kig cannot be set to zero, it must be a positive number.

h. For similar reasons as above, the feedback to the power factor controller is also from *Pord* and not the terminals of the model.

i. The ramp-rate of recovery after a fault on active-current (*rrpwr*) should also be imposed (in the opposite direction) when the model is being used to "emulate" charging of an energy storage device. That is, when *Pgen* is negative, then *rrpwr* should have its sign changed and it becomes the ramp-rate at which charging power (power being absorbed by the model) increases after a fault[2]. As stated above under item *f.* this is a simple aggregated model, and it does not represent any of the dynamics associated with the energy storage mechanism (e.g. battery dynamics). Therefore, it may not be appropriate to use this model to emulate events where in mid-simulation (during a normal 10 – 30 second stability simulation) the device changes from charging to discharging, etc.

The following additional comments are also pertinent to fully define the model:

1. The decision at the last REMTF meeting was to introduce a basic voltage source representation at the model interface to the network (a reduced version of the interface presented in [5]) to avoid potential numerical problems. This requires only one additional parameter Xe. Since this is an aggregated model, the resistance of the source impedance is neglected, and Xe should be set to a relatively large value, e.g. 0.3 to 0.5 pu. Once the beta version of the model is implemented the REMTF will test the model and experiment with values of Xe to make a decision on a default value.
2. *Vt* in the model refers to the voltage at the terminal of the *der_a* model, which is for example (Figure 1) the load bus at the end of the feeder when modeling R-DER. Similarly, *Pgen* is the electrical power being generated at the terminals of the *der_a* model.
3. Upon initialization, *Pref* and *Qref* will be determined in the software platform to properly initialize the models, based on the initial P/Q output of the *der_a* model.
4. In this case, it is quite possible that one may have both constant Q control and voltage control in-effect simultaneously or constant pf control and voltage control. To illustrate, let us assume that the PV inverter is in constant Q control, holding a small lagging power factor (or in constant pf control at a small lagging pf), such that it is generating 0.1 MVAr on a 2 MVA unit. Then let us say *Kqv* = 10 (voltage proportional control) and *dbd1*=*dbd2*=0.05 with *Vref0* = 1.0. Now so long as the voltage remains within 0.95 to 1.05 pu, the Q output of the unit remains at 0.1 MVAr and it does nothing to try to regulate voltage. If a major event happens to depress the voltage (e.g. a fault) or raise the voltage (e.g. load drop) outside of the deadband, then the proportional controls will act to increases/decrease Q until the voltage comes back inside the deadband, at which point Q drops back to its initial value. This is generally in keeping with the main proposed control strategies under discussion in IEEE 1547. There are of course other possible control strategies, but this model being an implementation of aggregated behavior, the group consensus was to keep it in the simplest format. There are a few caveats, however:

[2] A simple way to model this is:

```
If dIp/dt (derivative of state 9) > +rrpwr then
  If Ip (state 9) >= 0 then
   dIp/dt = +rrpwr
Elseif dIp/dt < -rrpwr then
  If Ip <= 0 then
   dIp/dt = -rrpwr
EndIf
```

This is checked at every-time step.

a. If *Kqv* is non-zero, then upon initialization *dbd1* < *Vt* - *Vref0* < *dbd2*, where *Vref0* is a user defined value as well as *dbd1* and *dbd2*. If this condition is not met, then the software tool should force the *Vref0* = *Vt* and indicate this to the user in a warning message.

b. If *dbd1=dbd2=0* (which should typically not be done, since these DG models are not intended to tightly control voltage) and *Kqv* is non-zero, then the program should give a warning/error message to the user and indicate that *Vref0* has been set to equal to *Vt* (to force the error to zero and thus the output of the voltage leg to zero); the initial Q from power flow is then initialized off of the constant Q/pf leg. This is the simplest solution in this case.

c. Lastly, there is one possible control problem here. If this model were used to model a single large inverter-based device connected to a very weak grid point where the voltage is highly affected by this device, then there could be a possibility for limit-cycling (i.e. voltage goes outside deadband, device brings voltage inside deadband by changing Q, Q drops to constant initial value once voltage is within the deadband, voltage goes outside deadband, etc.).

5. In general, it is not recommended that this model be used to model large plants.

6. To model the ramp-rate on *P* (*dPmax*/*dPmin*) this can be done with a first-order lag-block (with limits on the derivative of the state) which has a time constant equal to the integration time-step of the simulation.

7. The current limit is modeled as follows:

a. Q-priority: *Iqmx* = *Imax*; *Iqmin* = -*Imax*; *Ipmax* = $\sqrt{Imax^2 - Iqcmd^2}$; if *typeflag* = 0 then *Ipmin* = - *Ipmax,* else *Ipmin* = 0

b. P-priority: *Ipmx* = *Imax*; *Iqmax* = $\sqrt{Imax^2 - Ipcmd^2}$; *Iqmin* = -*Iqmax*; if *typeflag* = 0 then *Ipmin* = - *Ipmax,* else *Ipmin* = 0

8. The power factor angle reference (*pfaref*) is not a user-defined parameter. If power factor control is selected (*Pflag* = 1), then *pfaref* is calculated internally by the software program as arctan($Qgen_o/Pgen_o$), where $Qgen_o$ and $Pgen_o$ are the initial real and reactive power output of the model, respectively, as determined by the initial power flow solution.

9. Finally, during initialization, the software program should check to ensure that the terminal voltage (Vt) of the model initializes to a value that is greater than *vl1*. Also, *vl1* must be greater than or equal to *vl0*. If either of these conditions are not met, the program should present an error message to the user indicating that the value of *vl1* and *vl0* are inappropriate, and thus the model is completely ignoring the *Vrfrac* block (Figure 3, and associated logic). A similar check should be made on *vh1* and *vh0*. Also, a check should be made to ensure that *tvl1*, *tvl0*, *tvh1*and *tvh0* are all greater than or equal to zero. There is no limitation on which of these timer values should be greater or smaller. **Note:** the purpose of these timers is to allow for the emulation of inverters disconnecting under low-voltage scenarios, for example, legacy technology may disconnect quickly for a small voltage dip (i.e. one may set *vl1* = 0.9 and *tvl1* = 0.1 s) while part of the aggregate model may be representing modern inverters that comply with newer standards where it will not disconnect unless the voltage drops significantly for a longer duration (e.g. one may set *vl0* = 0.5 and *tvl0* = 1 s). Thus, this is to allow for testing various aspects of standards such as IEEE Std 1547 requirements, California Rule 21, etc.

10. In all cases of the controls, timers, etc. the filtered value of voltage (*Vt_filt*) and frequency (*Frq_filt*) should be used.

Table 1: Model parameter list

Parameter	Description
Trv	transducer time constant (s) for voltage measurement
Trf	transducer time constant (s) for frequency measurment (must be ≥ 0.02 s)
dbd1	lower voltage deadband ≤ 0 (pu)
dbd2	upper voltage deadband ≥ 0 (pu)
Kqv	proportional voltage control gain (pu/pu)
Vref0	voltage reference set-point > 0 (pu)
Tp	transducer time constant (s)
Tiq	Q control time constant (s)
Ddn	frequency control droop gain ≥ 0 (down-side)
Dup	frequency control droop gain ≥ 0 (up-side)
fdbd1	lower frequency control deadband ≤ 0 (pu)
fdbd2	upper frequency control deadband ≥ 0 (pu)
femax	frequency control maximum error ≥ 0 (pu)
femin	frequency control minimum error ≤ 0 (pu)
Pmax	Maximum power (pu)
Pmin	Minimum power (pu)
dPmax	Power ramp rate up > 0 (pu/s)
dPmin	Power ramp rate down < 0 (pu/s)
Tpord	Power order time constant (s)
Kpg	active power control proportional gain
Kig	active power control integral gain
Imax	Maximum converter current (pu)
vl0	voltage break-point for low voltage cut-out of inverters
vl1	voltage break-point for low voltage cut-out of inverters
vh0	voltage break-point for high voltage cut-out of inverters
vh1	voltage break-point for high voltage cut-out of inverters
tvl0	timer for vl0 point
tvl1	timer for vl1 point
tvh0	timer for vh0 point
tvh1	timer for vh1 point
Vrfrac	fraction of device that recovers after voltage comes back to within vl1 < V < vh1
fl	frequency break-point for low frequency cut-out of inverters
fh	frequency break-point for high frequency cut-out of inverters
tfl	timer for fl (Tfl > Trf)
tfh	timer for fh
Tg	Current control time constant
rrpwr	Power rise ramp rate following a fault > 0 (pu/s)
Tv	time constant on the output of the voltage/frequency cut-out
Vpr	voltage below which frequency tripping is disabled
Pflag	0 - for constant Q control, and 1 - constant power factor control
Pqflag	0 - Q priority, 1 - P priority for current limit
Freq_flag	0 - frequency control disabled, and 1 - frequency control enabled
Ftripflag	0 - frequency tripping disabled; 1 - frequency tripping enabled
Vtripflag	0 - voltage tripping disabled; 1 - voltage tripping enabled
typeflag	0 - the unit is a storage device and Ipmin = - Ipmax; 1 (or any value other than 0) - the unit is a generator Ipmin = 0;
Xe	Source impedance reactive > 0 (pu)
Iqh1	Maximum limit of reactive current injection, p.u.
Iql1	Minimum limit of reactive current injection, p.u.

References:

[1] NERC DRAFT Reliability Guideline: Modeling Distributed Energy Resources in Dynamic Load Models, September 2016

(available on-line at: http://www.nerc.com/pa/RAPA/rg/ReliabilityGuidelines/Reliability_Guideline-Modeling_DER_in_Dynamic_Load_Models-DRAFT.pdf)

[2] R.T. Elliott, A. Ellis, P. Pourbeik, J.J. Sanchez-Gasca, J. Senthil and J. Weber, "Generic Photovoltaic System Models for WECC – A Status Report", Proceedings of the IEEE PES GM 2015, Denver, CO, July 2015.

[3] P. Pourbeik, "Simple Model Specification for Battery Energy Storage System", EPRI memo, date: 1/15/15 (revised 2/25/15; 3/6/15; 3/18/15 rev3)

https://www.wecc.biz/Reliability/REEC_C_031815_rev3%20Model%20Spec.pdf

[4] WECC Generic Solar Photovoltaic System Dynamic Simulation Model Specification, 2012

https://www.wecc.biz/Reliability/WECC-Solar-PV-Dynamic-Model-Specification-September-2012.pdf

[5] D. Ramasubramanian, Z. Yu, R. Ayyanar, V. Vittal and J. M. Undrill, "Converter Model for Representing Converter Interfaced Generation in Large Scale Grid Simulations", IEEE Trans. PWRS, April 2016.

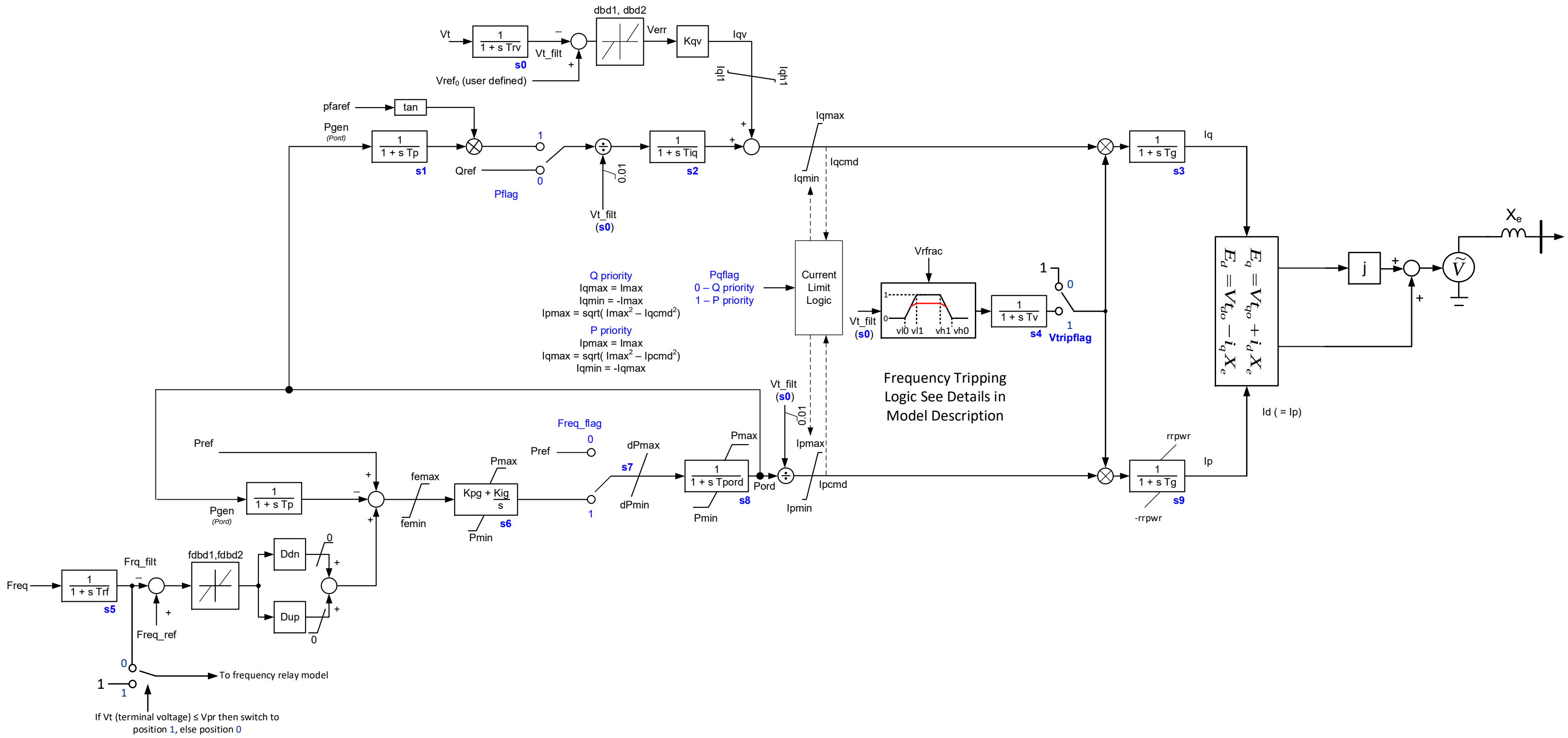
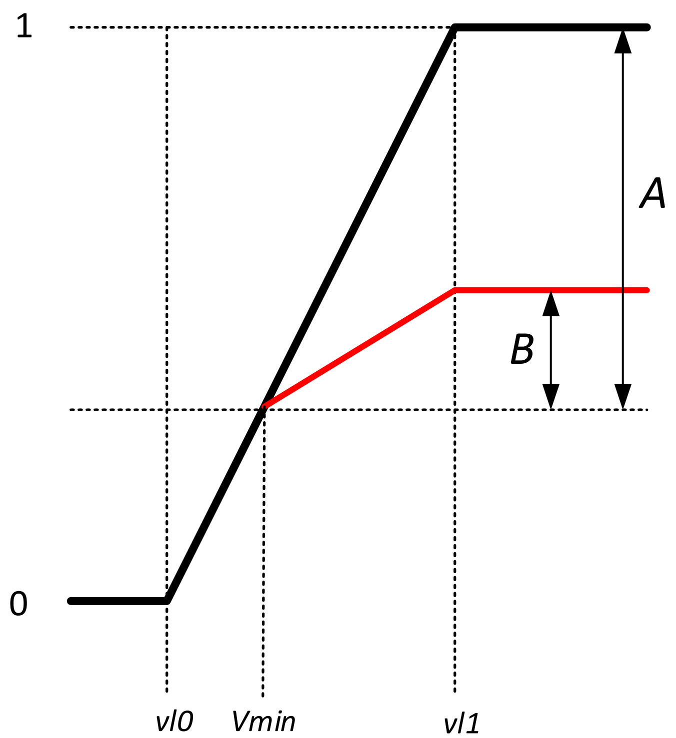


Figure 2: Proposed distributed energy resource model version A (*DER_A*).



$$B = Vrfrac \times A = Vrfrac \left(\frac{vl1 - Vmin}{vl1 - vl0}\right)$$

Figure 3: Effect of the *Vrfrac* (the code for this logic is provide in the Appendix)

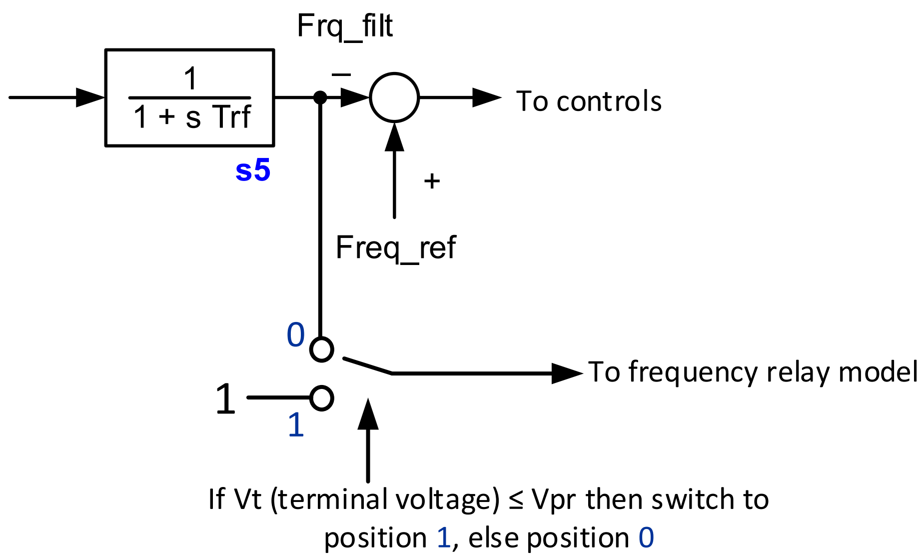


Figure 4: Input to frequency tripping logic.

Appendix: **Pseudo code for the Vrfrac block.**

The block shown in Figure 3 is implemented consistent with the existing *pvd1* model, as described in the PVD1 model specification [4]. However, the pseudo code and logic here is quite different to that in [4], as it is presently implemented in the *pvd1* model, since we have added two time parameters *tvl0* and *tvl1*, which determine when the limits are imposed once the assigned time has lapsed. That is, the output of the block will always track the path of the black line in Figure 3, unless certain conditions are met. If the voltage stays below *vl1* for a duration greater than *tvl1*, then it will now always follow the path of the red line when the voltage recovers. If the voltage stays below *vl0* for greater than *tvl0*, then the output will always remain at zero.

Note that *Vmin* in Figure 3 is not an input parameter by the user. It is an internal software variable which is keeping track of the minimum voltage that the terminal of the model reaches during a simulation immediately after the timer *tvl1* times out. That is, *Vmin* is the lowest point of Vt during a simulation, but at the moment that timer tvl1 times out it is set to (and kept at) the value of Vt at that instant. This is done to avoid jumps in the response due to movement (oscillations) in voltage. For example, consider the following scenario. During an event *Vt* goes down to *Vmin_a*, then comes back up to *Vt_2*, and then goes again down to *Vmin_b*, at which time the timer for tvl1 times out. Thus, it is the value of *Vmin_b* which we would like *Vmin* to be set to. This is depicted below in Figure 5.

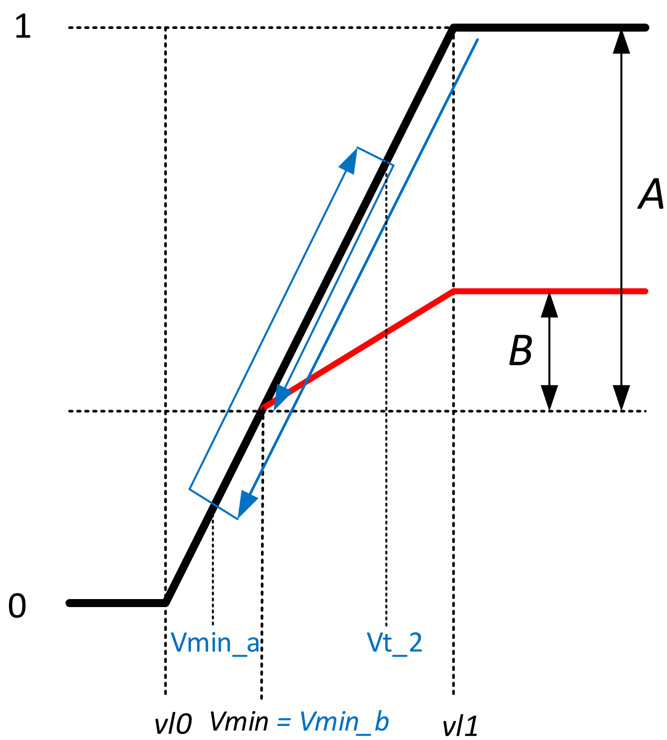


Figure 5: Understanding how Vmin is determined.

Vmin should initialize to the initial value of *Vt* or a default value (e.g. 1.0).

Timer 1 = 0

Timer 2 = 0

Counter 1 = 0

Counter 2 = 0

```
If Vt < vl1 and Timer 1 = 0
        Start Timer 1
elseif Vt > vl1 and Timer 1 started
        Reset Timer 1
end
If Vt < vl0 and Timer 2 = 0
        Start Timer 2
elseif Vt > vl0 and Timer 2 started
        Reset Timer 2
end
if Vmin <= vl0
        Vmin = vl0
end
if Vt <= vl0 or Counter 2 = 1
        Multiplier = 0.0
elseif Vt <= vl1 and Counter 1 = 0
        Multiplier = (Vt – vl0) / (vl1 – vl0)
elseif Vt <= vl1 and Counter 1 = 1
        Multiplier = ((Vmin – vl0) + Vrfrac * (Vt - Vmin)) / (vl1 – vl0)
elseif Vt >= vl1 and Counter 1 = 0
        Multiplier = 1
else
        Multiplier = Vrfrac * ((vl1 - Vmin) / (vl1 – vl0)) + ((Vmin – vl0)/(vl1 – vl0))
end
if Counter1 = 0
        if Timer1 > tvl1
                Counter1=1
                Vmin = Vt
        end
end
if Counter2 = 0
        if Timer2 > tvl0
                Counter2=1
        end
```

end

The key here is that Counter 1 (2) get set only if the condition of being below *vl1* (*vlo*) is met for the given time duration and once that condition is met the block remains in that state indefinitely. Also, *Vmin* is set to the value of Vt at the point when timer 1 (*tvl1*) times out.

The same logic is then implemented for Vt exceeding *vh1* while keeping track of the maximum voltage reached during the simulation (*Vmax*).