TO:	WECC MVS
FROM:	James Weber, PowerWorld Corporation
SUBJECT:	Branch Stability Model SCMOV (Series Capacitor Metal Oxide Varistor)
DATE:	June 3, 2024

This memo describes a model of the over-current and over-voltage protection for a series capacitor which uses a Metal Oxide Varistor (MOV) and bypass switch. The model is based on the following paper from 1987.

D. L. Goldsworthy, "A linearized model for MOV-protected series capacitors", IEEE Transactions on Power Systems, Vol. PWRS-2, No. 4, November 1987

See the Goldsworthy paper for a detailed explanation of this technology. Below is a description of how the model impacts the software simulation.

The Input Parameters for SCMOV are as follows

Icrated	Capacitor Rated current in Amps
Icappro	Capacitor protective level current, pu on Icrated base
Ithresh	Threshold value for MOV activation. Any value smaller than 0.94879 will be ignored and treated as equal to 0.94879.
Enerlim	MOV energy limit in Mjoules
Enerdly	Bypass delay associated with Enerlim in seconds
Imovlim	MOV current limit in pu of Icrated
Imovdly	Bypass delay associated with Imovlim in seconds
Icaplim	Capacitor current limit in pu of Icrated
Icapdly	Bypass delay associated with Icaplim in seconds
Iinsert	Insertion current in pu of Icrated
Tinsert	Insertion time in seconds
ImovTup	Pickup time for the Imovlim in seconds
IcapTup	Pickup time for the Icaplim in seconds

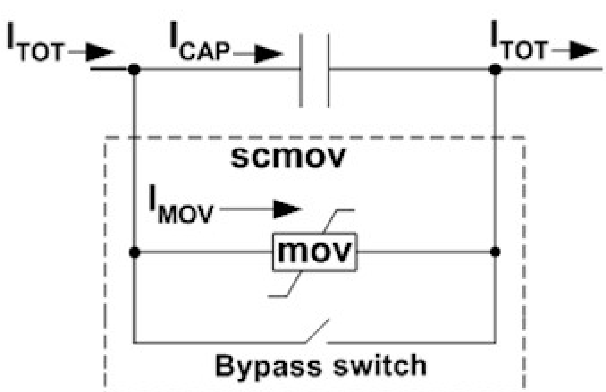
Other Output values for SCMOV are as follows

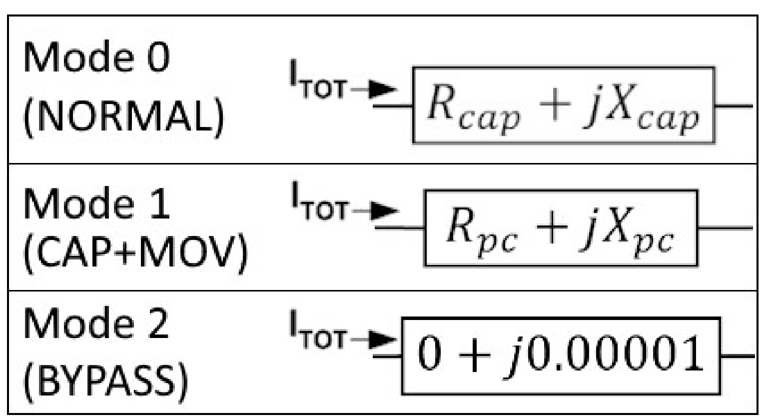
Mode	Mode of operation 0 = NORMAL (model impedance as Rcap + jXcap) 1 = CAP+MOV (model impedance as Rpc + jXpc calculated as described below) 2 = BYPASS (model impedance as 0.0000001 + j0.00001
ItotAmp	Total current magnitude seen by the network solution in Amps
IcapAmp	Current magnitude across the capacitor in Amps
ImovAmp	Current magnitude across the MOV in Amps
EnergyMOV	Accumulated energy absorbed by the MOV in MegaJoules
Itotpu	Total current magnitude seen by the network solution in per unit on the system base
Icappu	Current magnitude across the capacitor in per unit on the system base
Imovpu	Current magnitude across the MOV in per unit on the system base
Rused	Rpu (on system base) used in the network algebraic solution (varies depending on the mode)
Xused	Xpu (on system base) used in the network algebraic solution (varies depending on the mode)

SCMOV Three Modes of Operation

The complex currents used when modeling this device are shown in the figure below.
The model operates in three different modes with the treatment in the algebraic network boundary equation solution as follows.

Mode	Treatment in Algebraic Network Boundary Equations	Currents Reported for Output
NORMAL Mode=0	The original Rpu and Xpu of the series capacitor is used to model the device in the algebraic network boundary equations. Note: during a algebraic network boundary equation solution, the device will instantaneously change to Mode 1 (CAP+MOV) if the current threshold Ithresh is exceed as described in the transitions below. This special transition during the algebraic solution is needed to prevent very large current spikes that occur at fault inception on or near the terminals of these series capacitors.	Itot = magnitude of current through branch Icap = Itot Imov = 0.0
CAP+MOV Mode=1	An $\mathrm{R_{pc}} + \mathrm{jX_{pc}}$ is calculated as described in the image below based on function of the current Ipux described in the Goldsworthy reference paper. While operating in this mode the energy absorbed by the MOV is calculated by determining the resistance of the equivalent impedance $R_{pc} + jX_{pc}$ and the branch impedance $\mathrm{R_{cap}} + \mathrm{jX_{cap}}$. $\mathrm{R_{mov}} + \mathrm{jX_{mov}} = \frac{1}{\frac{1}{\mathrm{R_{pc}+jX_{pc}}} - \frac{1}{\mathrm{R_{cap}+jX_{cap}}}}$ (so $\mathrm{R_{mov}}$ is real part of this) The power absorbed over a time step as resistive loss is calculated as follows MegaWattMOV = (Rmov*Imovpu*Imovpu)*SystemMVABase Integration is approximated by multiplying the TimeStep length in seconds by MegaWattMOV and keeping the accumulated sum as the value EnergyMOV (with units of MegaJoules). Also note that the accumulated EnergyMOV is not reset during the simulation even when transferring out of this mode. Note: The Goldsworthy paper suggested using the following equation for the power loss MegaWattMOV = (Rpc*Itotpu*Itopu)*SystemMVABase When Rcap = 0, this gives the same answer.	Itot = magnitude of current through branch Icap = current calculated from terminal voltage and original Rcap and Xcap Imov = magnitude of the *complex* current calculation (Itot - Icap)
BYPASS Mode=2	When operating in Mode 2 (BYPASS) the minimum internal impedance is used of R + jX = 0.0000001 + j0.00001	Itot = magnitude of current through branch Icap = 0 Imov = 0



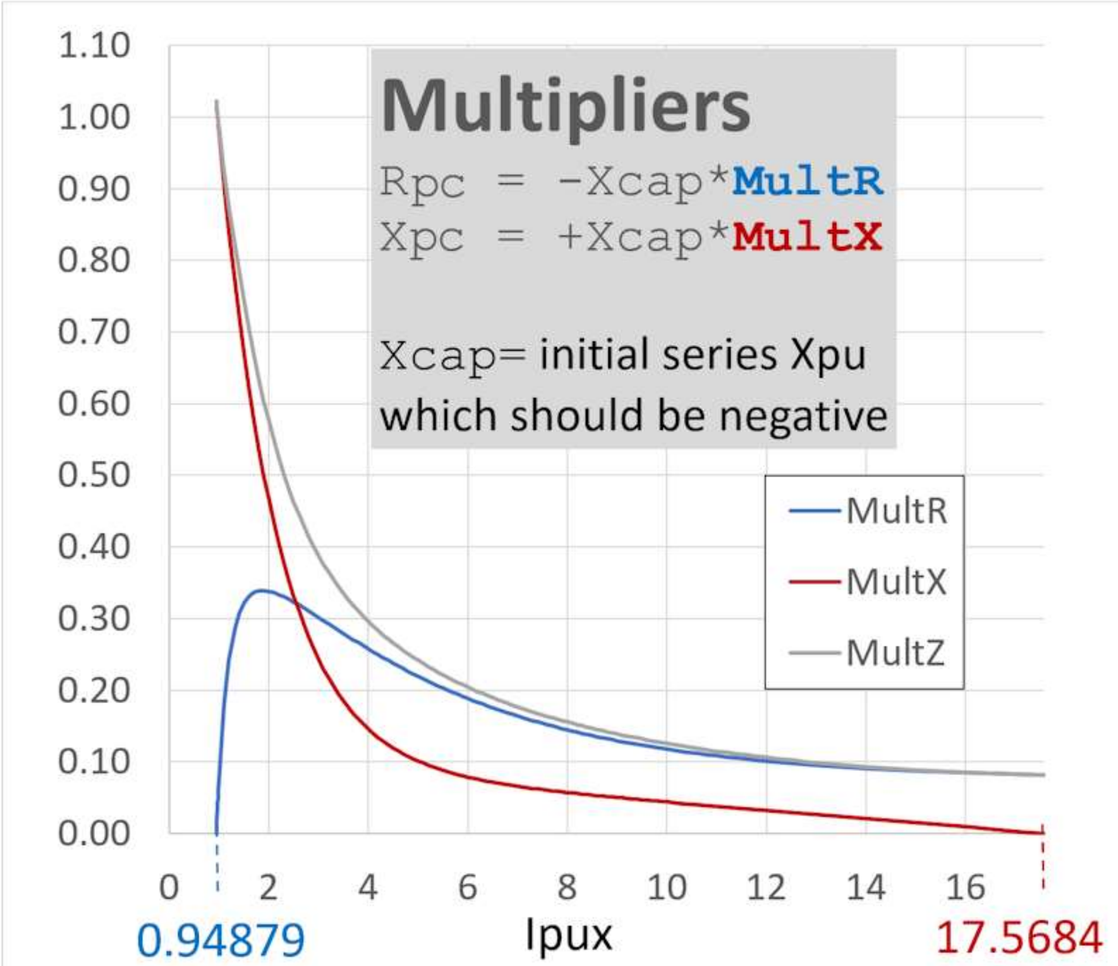


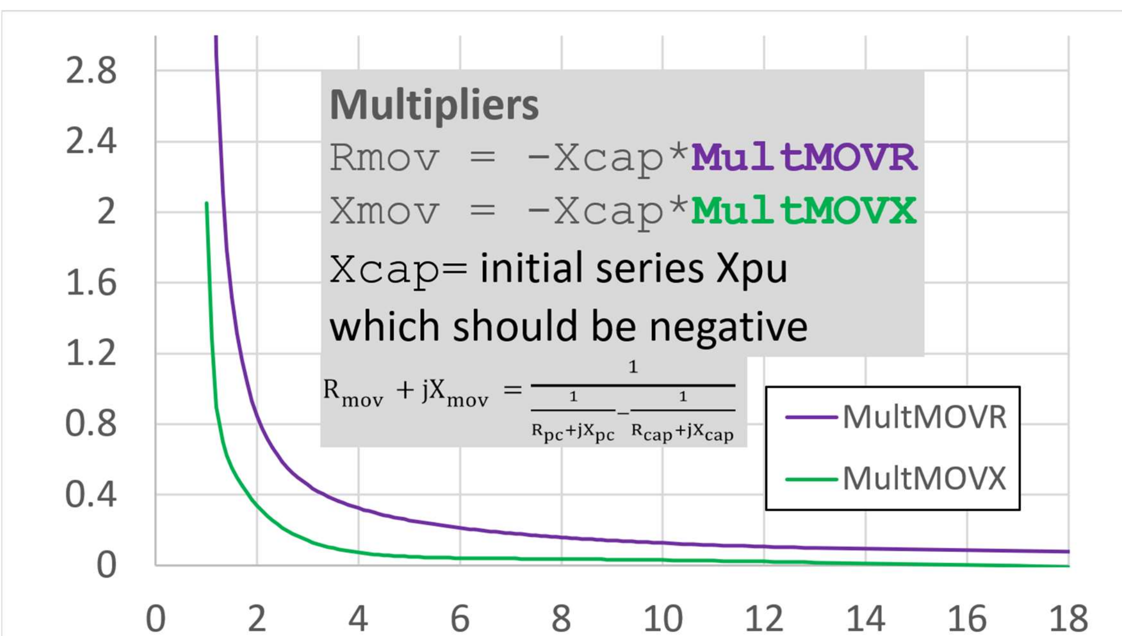
$$I_{pux} = \frac{I_{totAmps}}{I_{cappro} I_{crated}}$$

Restrict Ipux to between 0.94879 and 17.5684
to prevent MultR and MultX from being negative values

$$MultR(I_{pux}) = 0.0745 + 0.49e^{-0.243*I_{pux}} - 35.0e^{-5.0*I_{pux}} - 0.60e^{-1.4*I_{pux}}$$

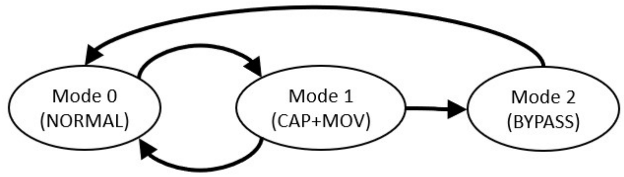
$$MultX(I_{pux}) = 0.1010 - 0.005749 * I_{pux} + 2.088e^{-0.8566*I_{pux}}$$





Mode Transitions

Transitions between the 3 operating modes can occur as follows with 4 possible transitions as depicted in the next image. The conditions under which these mode transitions occur are described in the following table.



From Mode	Transition to Mode	Description
NORMAL Mode=0	CAP+MOV Mode=1	When in Mode 0 (NORMAL), at the end of each time step of the simulation if the following condition is met then the device will switch to Mode 1 (CAP+MOV) Ipux > Ithresh , where Ipux = ItotAmp/(Icappro*Icrated) . Note that the current base for getting Ipux in per unit is different than the per unit for comparisons to Icaplim and Imovlim described later because the multiplier Icappro is used with Ipux. During an algebraic network boundary equation solution, the device will instantaneously switch to Mode 1 (CAP+MOV) if the current threshold Ithresh is exceed as described in Mode 1 description. If this occurs the mode is switched and the algebraic solution is immediately redone.
NORMAL Mode=0	BYPASS Mode=2	This direct transition will not happen
CAP+MOV Mode=1	NORMAL Mode=0	When in Mode 1 (CAP+MOV), at the end of each time step, if Ipux <= Ithresh then the device transitions to Mode 0 (NORMAL) at next time step.
CAP+MOV Mode=1	BYPASS Mode=2	When in Mode 1 (CAP+MOV), at the end of each time step, three different checks are done to determine if a transition to Mode 2 (BYPASS) is made. 1. Enerlim If the accumulated energy absorbed by the MOV stored in EnergyMOV exceeds the MOV energy limit Enerlim for more than Enerdly seconds, then the device will bypass. If Enerlim = 0 this bypassing option is ignored. 2. Imovlim: If the current passing through the MOV exceeds Imovlim for ImovTup seconds, the capacitor and MOV are bypassed after a delay of Imovdly seconds (even if the MOV current falls below the Imovlim during this delay time). If Imovlim = 0 this bypassing option is ignored. Imovlim is in per unit on Icrated base so proper conversions are required when comparing with it. 3. Icaplim: If the current passing through the capacitor exceeds Icaplim for IcapTup seconds, the capacitor and MOV are bypassed after a delay of Icapdly seconds (even if the capacitor current falls below the Icaplim during this delay time). If Icaplim = 0 this bypassing option is ignored. Icaplim is in per unit on Icrated base so proper conversions are required when comparing with it.
BYPASS Mode=2	NORMAL Mode=0	When in Mode 2 (BYPASS), the model allows for reinsertion of the device if the bypass decision was made due to the Imovlim or the Icaplim. Reinsertion will occur if all other bypass conditions are not met and the current is below the Iinsert for Tinsert seconds. When this transition occurs the Enerdly timer for Enerlim is reset and will start counting again If the device was set to BYPASS due to the EnerLim, then the device will not reinsert the capacitor and MOV. One exception to this is if another stability model or a user event has intentionally removed the Bypass, then this transition will reset the EnergyMOV = 0 and reset the Enerdly timer.
BYPASS Mode=2	CAP+MOV Mode=1	This direct transition will not happen

Pseudo-Code forVariables maintained across time steps

```
Branch Bypass status is something independent of this model
Mode       : integer
EnergyMOV : float
Itotpu     : float
Imovpu     : float
Icappu     : float
CACHE_BypassSentICAP   : boolean
CACHE_BypassSentIMOV   : boolean
CACHE_BypassSentEnergy : boolean
TimeOfAboveEnerLim     : float
TimeOfAboveEnerLimSet  : boolean
TimeOfBelowIinsert     : float
TimeOfBelowIinsertSet  : boolean
TimeOfAboveImovLim     : float
TimeOfAboveImovLimSet  : boolean
TimeOfAboveIcapLim     : float
TimeOfAboveIcapLimSet  : boolean
```

Pseudo-Code for Model Initialization

```
MOVIsConducting = False
EnergyMOV = 0
Itotpu = calculate current from terminal voltage using Rcap and Xcap value
Icappu = Itotpu
Imovpu = 0
CACHE_BypassSentICAP   = false
CACHE_BypassSentIMOV   = false
CACHE_BypassSentEnergy = Branch initially bypassed // treat initial as permanent bypass
TimeOfAboveEnerLim     = 0
TimeOfAboveEnerLimSet  = false
TimeOfBelowIinsert     = 0
TimeOfBelowIinsertSet  = false
TimeOfAboveImovLim     = 0
TimeOfAboveImovLimSet  = false
TimeOfAboveIcapLim     = 0
TimeOfAboveIcapLimSet  = false
```

Pseudo-Code for function to Recalculate Rpc, Xpc, and Rmov based on a new value of Itotpu

```
// recalculate Rpc and Xpc using Multiplier functions based on present total current
procedure RecalculateRpcAndXpc(Itotpu)
   Ipux = Itotpu*Ibase/Icrated/Icappro
   If     Ipux < 0.94879 then Ipux = 0.94879   // don't let MultiplierR become negative
   ElseIf Ipux > 17.5684 then Ipux = 17.5684   // don't let MultiplierX become negative
   EndIf
   MultR = 0.0745 + 0.49*exp(-0.243*Ipux) - 35.0*exp(-5.0*Ipux) - 0.60*exp(-1.40*Ipux)
   MultX = 0.1010 - 0.005749*Ipux + 2.088*exp(-0.8566*Ipux)
   if MultR < 0 then MultR = 0 // shouldn't happen but just make sure
   if MultX < 0 then MultX = 0 // shouldn't happen but just make sure
   Rpc = -MultR*Xcap // Xcap = original X
   Xpc = +MultX*Xcap

   Gpc =  Rpc/(sqr(Rpc) + sqr(Xpc))
   Bpc = -Xpc/(sqr(Rpc) + sqr(Xpc))
   Gcap =  Rcap/(sqr(Rcap) + sqr(Xcap))
   Bcap = -Xcap/(sqr(Rcap) + sqr(Xcap))
   Gmov = Gpc - Gcap
   Bmov = Bpc - Bcap;
   Rmov = Gmov/(sqr(Gmov) + sqr(Bmov))
```

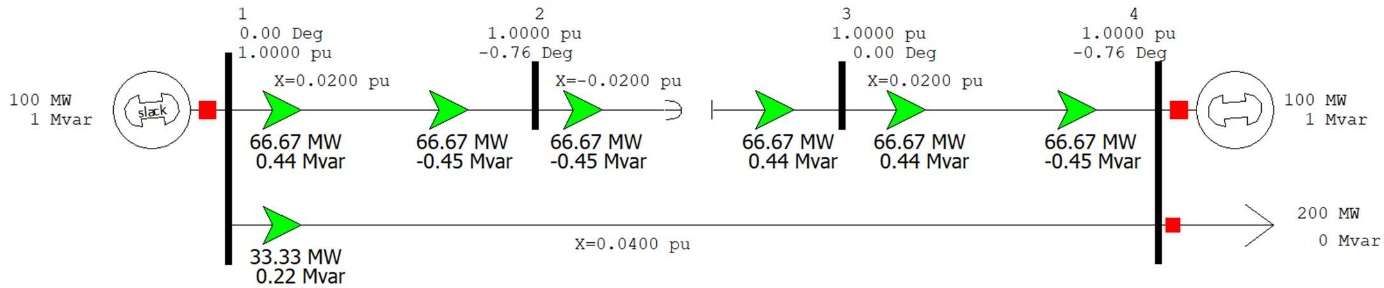
Pseudo-Code for Mode Transitions run at the end of each time-step

```
// Determine the present per unit currents on the system base
If (Branch Is Bypass) Then // This is Mode 2
  Itotpu = calculate currents from terminal voltage small impedance used for bypass
  Imovpu = 0
  Icappu = 0
  CACHE_BypassSentICAP = false
  CACHE_BypassSentIMOV = false
  If (not CACHE_BypassSentEnergy) Then
    if (Itotpu*CACHE_IBase/Icrated <= Iinsert)
       and ( (Imovpu*IBase/Icrated <= Imovlim) or (Imovlim = 0) ) // make sure bypass conditions are not met
       and ( (Icappu*IBase/Icrated <= Icaplim) or (Icaplim = 0) )
    then
      MOVIsConducting = False
      if not TimeOfBelowIinsertSet then begin
        TimeOfBelowIinsert = PresentTime
        TimeOfBelowIinsertSet = true
      EndIf
      if (PresentTime - TimeOfBelowIinsert >= Tinsert) then begin
        Write Message indicating a transition from BYPASS to NORMAL has been scheduled immediately
        Schedule an event to remove the BYPASS on the next timestep (this event will transition us to Mode 0)
        MOVIsConducting = False
        TimeOfAboveEnerLimSet = False // reset EnerLim timer
      EndIf
    Else
      TimeOfBelowIinsertSet = false // reset Insert timer
    EndIf
  EndIf
Else
  // The SCMOV model itself will never reinsert after it causes a bypass due to the EnergyMOV > Enerlim.
  // We know that EnergyMOV > Enerlim has happened because CACHE_BypassSentEnergy = TRUE, so the only
  // way we get here  is if another stability model or a user event has intentionally removed the Bypass.
  // In that situation I think we just assume the EnergyMOV has dissipated.
  If CACHE_BypassSentEnergy then
    Write message indicating that we are reseting the EnergyMOV and its timer
    CACHE_BypassSentEnergy = False  // reset to allow EnergyMOV to cause it to bypass again
    EnergyMOV = 0                   // clear energy to make 0.0 again
    TimeOfAboveEnerLimSet = False   // reset EnerDly timer
  EndIf

  ICapComplex = calculate currents from terminal voltage using original R and X value (per unit system base)
  If MOVIsConducting Then // This is Mode 1 (CAP+MOV)
    ItotComplex = calculate current from terminal voltage and Rpc and Xpc (per unit system base)
    ImovComplex = Complex Difference of (ItotComplex - IcapComplex)
    Itotpu = ItotComplex.Magnitude
    Icappu = IcapComplex.Magnitude
    Imovpu = ImovComplex.Magnitude
    if TimeStep > 0 then EnergyMOV = EnergyMOV + TimeStep*sqr(Imovpu)*Rmov*SystemMVABase
  Else // This is Mode 0 (NORMAL)
    Itotpu = ICapComplex.Magnitude
    Icappu = Itotpu
    Imovpu = 0
  EndIf
```

```
  // Check various Timers and Threshold that determine Bypass commands
  If (not CACHE_BypassSentEnergy) AND (Enerlim > 0) AND (EnergyMOV > fEnerlim) Then
    If not TimeOfAboveEnerLimSet Then
      TimeOfAboveEnerLim = PresentTime
      TimeOfAboveEnerLimSet = true
      If Enerdly > 0 Then Write out a messsage indicating that the Enerdlg timer has started
    EndIf
    If (PresentTime - TimeOfAboveEnerLim >= Enerdly) Then
      CACHE_BypassSentEnergy = true
      TimeOfAboveEnerLimSet = false
      Schedule an event to apply a BYPASS on the next Time Step (this event transitions us to Mode 2)
    EndIf
  EndIf

  If (ImovLim <= 0) OR (Imovpu*IBase/Icrated >= Imovlim) Then
    If TimeOfAboveImovLimSet then
      Write out a messsage indicating that the ImovTup timer has stopped
      TimeOfAboveImovLimSet = False
    EndIf
  ElseIf (not CACHE_BypassSentIMOV) Then
    If not TimeOfAboveImovLimSet Then
      TimeOfAboveImovLim = PresentTime
      TimeOfAboveImovLimSet = true
      If ImovTup > 0 Then Write out a messsage indicating that the ImovTup timer has started
    EndIf
    If (PresentTime - TimeOfAboveImovLim >= ImovTup) Then
      CACHE_BypassSentIMOV = true
      TimeOfAboveImovLimSet = False
      Write message indicating a transition from CAP_MOV to BYPASS has been scheduled to occur in Imovdly
      Schedule an event to apply a BYPASS in Imovdly seconds (this event will transition us to Mode 2)
    EndIf
  EndIf

  If (IcapLim <= 0) OR (Icappu*IBase/Icrated >= Icaplim) Then
    If TimeOfAboveIcapLimSet then
      Write out a messsage indicating that the IcapTup timer has stopped
      TimeOfAboveIcapLimSet = False
    EndIf
  ElseIf (not CACHE_BypassSentICAP) Then
    If not TimeOfAboveIcapLimSet Then
      TimeOfAboveIcapLim = PresentTime
      TimeOfAboveIcapLimSet = true
      If IcapTup > 0 Then Write out a messsage indicating that the IcapTup timer has started
    EndIf
    If (PresentTime - TimeOfAboveIcapLim >= IcapTup) Then
      CACHE_BypassSentICAP = true
      TimeOfAboveIcapLimSet = False
      Write message indicating a transition from CAP_MOV to BYPASS has been scheduled to occur in Icapdly
      Schedule an event to apply a BYPASS in Icapdly seconds (this event will transition us to Mode 2)
    EndIf
  EndIf


  // Manage the Mode Transitions between Mode 0 and 1
  If MOVIsConducting Then // This is Mode 1 (CAP+MOV)
    if (Itotpu*CACHE_IBase/Icrated/Icappro <= Ithresh) Then
      MOVIsConducting = False
      Write message indicating a transition from CAP+MOV to NORMAL
    Else
      RecalculateRpcAndXpc(Itotpu) // recalculate Rpc and Xpc and Rmov (see other procedure above)
    EndIf
  Else // This is Mode 0 (NORMAL)
    If (Itotpu*IBase/Icrated/Icappro > fIthresh) Then
      MOVIsConducting = True // transition to MOV mode
      RecalculateRpcAndXpc(Itotpu) // recalculate Rpc and Xpc and Rmov (see other procedure above)
      Write message that we have transitioned from NORMAL to CAP+MOV Mode
    EndIf
  EndIf
EndIf
```

Example Simulation

Following sample system has all 230 kV buses. All resistances are 0.0. Other information needed for network model is in picture.



Generator MVABase for both generators is 500 MVA and the following GENROU Machine, ESAC1A Exciter and TGOV1 Governor models and parameters. The Series Cap from bus 2-3 has the SCMOV parameters below. A constant impedance model is used for the load at bus 4.

GENROU

H	3
D	0
Ra	0
Xd	2.1
Xq	0.5
Xdp	0.2
Xqp	0.5
Xdpp	0.18
Xl	0.15
Tdop	7
Tqop	0.75
Tdopp	0.04
Tqopp	0.05
S1	0
S12	0
RComp	0
XComp	0

ESAC1A

Tr	0
Tb	0
Tc	0
Ka	200
Ta	0.02
VaMax	7.5
VaMin	-7.5
Te	0.8
Kf	0.03
Tf	1
Kc	0.2
Kd	0.38
Ke	1
E1	3
SE1	0.03
E2	4
SE2	0.1
Vrmax	14
Vrmin	-14
Spdmlt	0

TGOV1

Trate	0
R	0.05
T1	0.5
Vmax	1
Vmin	0
T2	2.5
T3	7.5
Dt	0

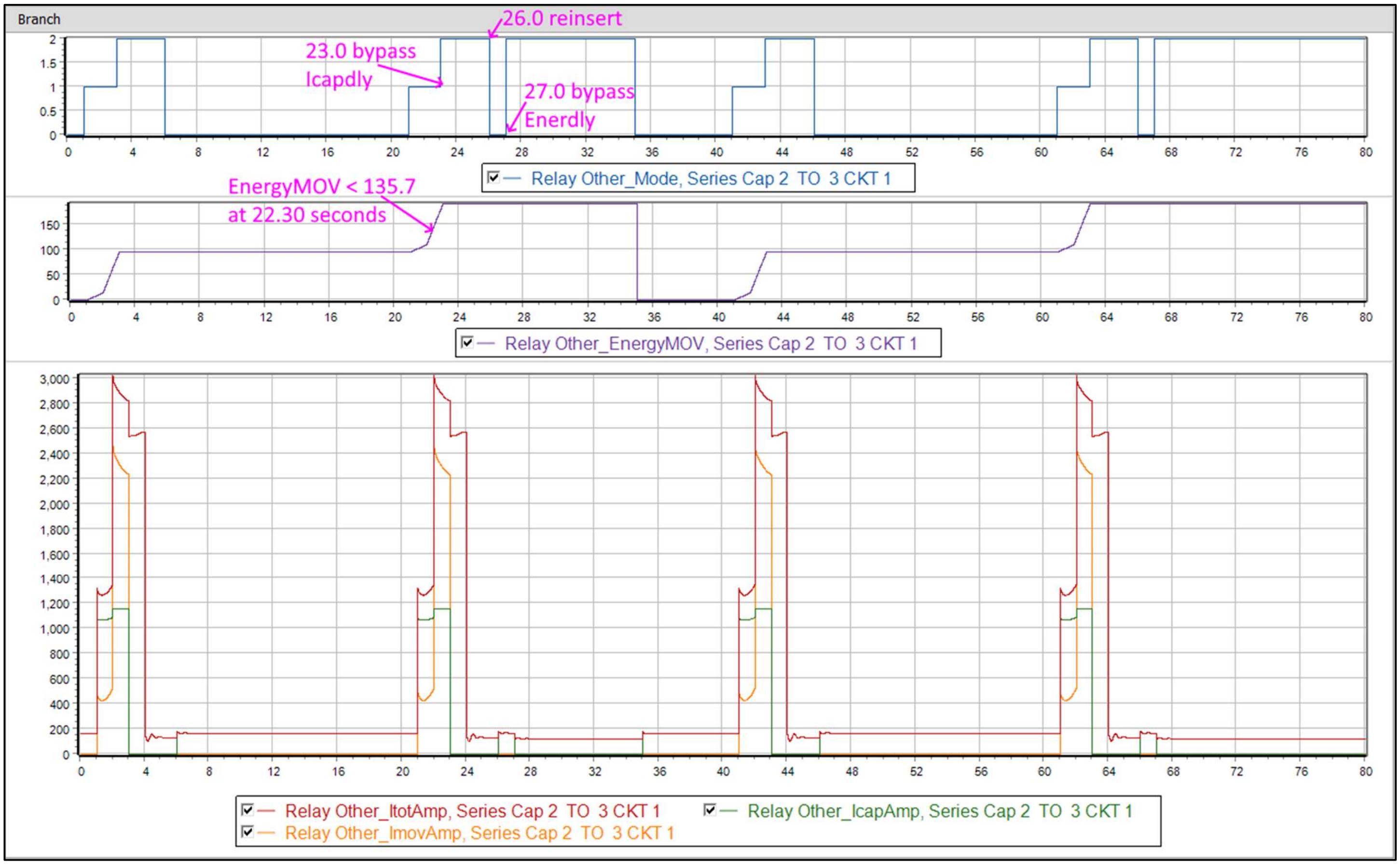
SCMOV

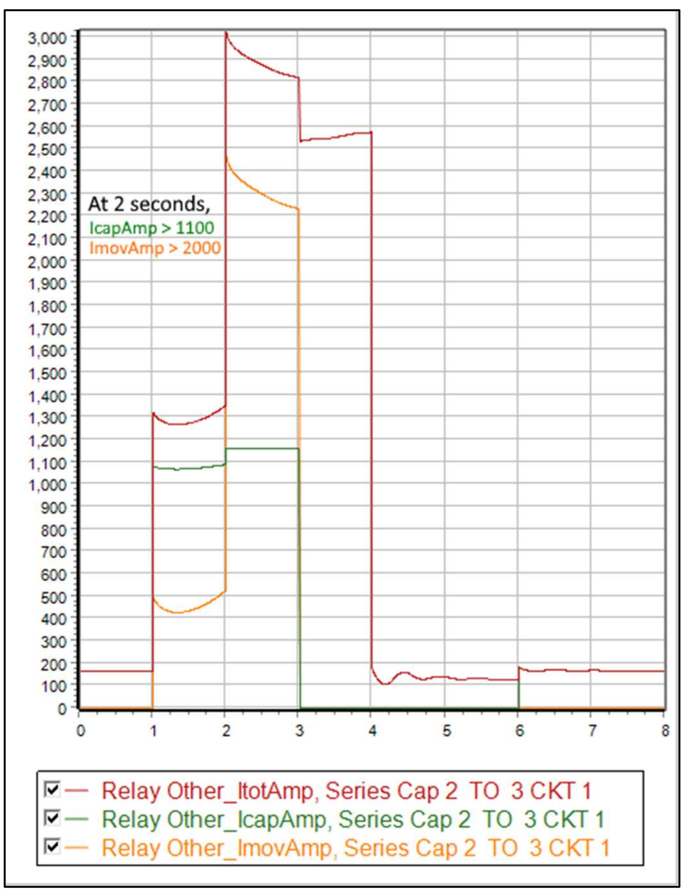
Icrated	1000
Icappro	1
Ithresh	0.98
Deaccel	0.1
Enerlim	135.7
Enerdly	1
Imovlim	2
Imovdly	1
Icaplim	1.1
Icapdly	0.8
Operdly	0
Iinsert	0.3
Tinsert	2
ImovTup	0.1
IcapTup	0.2

A Simulation is run that applies a fault at Bus 3 with fault impedance 0.1 per unit at 1.00 second. Fault impedance is changed to 0.02 at 2.00 seconds. Fault is cleared at 4.00 seconds. This sequence 3 events is then repeated at 20, 40, and 60 seconds. Finally, a user entered removal of the bypass is added at 35 seconds.

Time	Object	Action
1.0	Bus 3	FAULT 3PB IMP 0 0.1
2.0	Bus 3	FAULT 3PB IMP 0 0.02
4.0	Bus 3	CLEARFAULT
21.0	Bus 3	FAULT 3PB IMP 0 0.1
22.0	Bus 3	FAULT 3PB IMP 0 0.02
24.0	Bus 3	CLEARFAULT
35.0	Branch 2 3 1	NOTBYPASS
41.0	Bus 3	FAULT 3PB IMP 0 0.1
42.0	Bus 3	FAULT 3PB IMP 0 0.02
44.0	Bus 3	CLEARFAULT
61.0	Bus 3	FAULT 3PB IMP 0 0.1
62.0	Bus 3	FAULT 3PB IMP 0 0.02
64.0	Bus 3	CLEARFAULT

The resulting simulation using a 0.004 second timestep is as follows. Second image is the first 8 seconds closeup.





The first 30 seconds of simulation occur with the follow events.

- At 1.0 seconds the fault is applied and the current immediately exceed the Ithresh value and we transition to CAP+MOV mode, but we do NOT exceed any of the values that would cause bypassing.
- At 2.0 second the fault impedance decreases causing currents to increase. Immediately the Imovlim and the Icaplim thresholds and timers start for these limits.
- At 2.1 seconds, the Imovlim timer of 0.1 seconds expires so a bypass event is scheduled with a delay of 1.0 seconds.
- At 2.2 seconds, the Icaplim timer of 0.2 seconds expires so a bypass event is scheduled with a delay of 0.8 seconds.
- At 3.0 seconds, the series cap is bypassed due to the 0.8 second time delay for Icapdly.
- At 3.1 seconds, the 2nd bypass event from Imovdly is ignored because bypassing has already occurred.
- At 4.0 seconds the fault clears and the total EnergyMOV accumulated by that point is 95.957 MJ which is not enough to cause a permanent bypass
- At 4.0 seconds, the ItotAmp drops below Iinsert and the timer for Tinsert starts
- At 6.0 seconds the Tinsert timer expires and the series cap is reinserted
- This sequence of events is then repeated at time 21 seconds, however this time the EnergyMOV has continued to accumulate and will change the response
- At 22.30 seconds the EnergyMOV > Enerlim value and if it stayed there until 23.30 seconds it would have done a permanent bypass, however the model bypasses before this at 23.00 seconds due to the Icapdly again.
- At 24.0 seconds the Tinsert timer starts again and the device is inserted at 26.0 second
- At 26.0 seconds the EnergyMOV>Enerlim still so the Enerdlg timer starts
- At 27.0 seconds the Enerdly timer expires and a permanent bypass is done.

The first 30 seconds of simulation an events that occur are summarized in the table below.

Time	Object	Model	Reason	Description
1.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
1.00	Branch 2 3 1	SCMOV	Ithresh	Transition from NORMAL to CAP+MOV because ItotAmp/(Icrated*Icappro) > Ithresh. Done Immediately
2.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
2.00	Branch 2 3 1	SCMOV	Imovlim	Timer Started: ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds
2.00	Branch 2 3 1	SCMOV	Icaplim	Timer Started: IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds
2.10	Branch 2 3 1	SCMOV	Imovlim	Transition decision from CAP+MOV to BYPASS because ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds . Schedule with delay Imovdly=1 seconds.
2.20	Branch 2 3 1	SCMOV	Icaplim	Transition decision from CAP+MOV to BYPASS because IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds . Schedule with delay Icapdly=0.8 seconds.
3.00	Branch 2 3 1	SCMOV	Icaplim	IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds , so branch series cap Bypass
3.10	Branch 2 3 1	SCMOV	Imovlim	ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds , so branch series cap Bypass (No change. Device was already Bypass)
4.00	Bus 3	Bus	User	Clear Fault
4.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp >= Iinsert, timer for Tinsert started
6.00	Branch 2 3 1	SCMOV	Iinsert	Transition decision from BYPASS to NORMAL because ItotAmp/Icrated < Iinsert for Tinsert seconds. Schedule Immediately
6.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp/Icrated < Iinsert for Tinsert seconds, so series cap Not Bypass
21.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
21.00	Branch 2 3 1	SCMOV	Ithresh	Transition from NORMAL to CAP+MOV because ItotAmp/(Icrated*Icappro) > Ithresh. Done Immediately
22.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
22.00	Branch 2 3 1	SCMOV	Imovlim	Timer Started: ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds
22.00	Branch 2 3 1	SCMOV	Icaplim	Timer Started: IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds
22.10	Branch 2 3 1	SCMOV	Imovlim	Transition decision from CAP+MOV to BYPASS because ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds . Schedule with delay Imovdly=1 seconds.
22.20	Branch 2 3 1	SCMOV	Icaplim	Transition decision from CAP+MOV to BYPASS because IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds . Schedule with delay Icapdly=0.8 seconds.
22.30	Branch 2 3 1	SCMOV	Enerlim	Timer Started: EnergyMOV > Enerlim for Enerdlg=1 seconds
23.00	Branch 2 3 1	SCMOV	Icaplim	IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds , so branch series cap Bypass
23.10	Branch 2 3 1	SCMOV	Imovlim	ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds , so branch series cap Bypass (No change. Device was already Bypass)
24.00	Bus 3	Bus	User	Clear Fault
24.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp >= Iinsert, timer for Tinsert started
26.00	Branch 2 3 1	SCMOV	Iinsert	Transition decision from BYPASS to NORMAL because ItotAmp/Icrated < Iinsert for Tinsert seconds. Schedule Immediately
26.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp/Icrated < Iinsert for Tinsert seconds, so series cap Not Bypass
26.00	Branch 2 3 1	SCMOV	Enerlim	Timer Started: EnergyMOV > Enerlim for Enerdlg=1 seconds
27.00	Branch 2 3 1	SCMOV	Enerlim	Transition decision from CAP+MOV to BYPASS because EnergyMOV > Enerlim for Enerdlg=1 seconds . Schedule Immediately.
27.00	Branch 2 3 1	SCMOV	Enerlim	EnergyMOV > Enerlim for Enerdlg=1 seconds , so branch series cap Bypass

At this point a permanent bypass has occurred and the SCMOV model itself would not undo this. However we have intentionally included a removal of the bypass event as a *user event* at 35 seconds. Doing this resets the EnergyMOV to 0.0 and related timers as you see in the yellow events below at 35.0 seconds. This then kicks off the same set events again at 41 seconds and then 61 seconds as shown next.

Time	Object	Model	Reason	Description
35.00	Branch 2 3 1	TXLine	User	Not Bypass
35.00	Branch 2 3 1	SCMOV	Enerlim	Although model was bypass due to Enerlim, reset of EnergyMOV and Enerdly timer because another model or event has removed the Bypass
41.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
41.00	Branch 2 3 1	SCMOV	Ithresh	Transition from NORMAL to CAP+MOV because ItotAmp/(Icrated*Icappro) > Ithresh. Done Immediately
42.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
42.00	Branch 2 3 1	SCMOV	Imovlim	Timer Started: ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds
42.00	Branch 2 3 1	SCMOV	Icaplim	Timer Started: IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds
42.10	Branch 2 3 1	SCMOV	Imovlim	Transition decision from CAP+MOV to BYPASS because ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds . Schedule with delay Imovdly=1 seconds.
42.20	Branch 2 3 1	SCMOV	Icaplim	Transition decision from CAP+MOV to BYPASS because IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds . Schedule with delay Icapdly=0.8 seconds.
43.00	Branch 2 3 1	SCMOV	Icaplim	IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds , so branch series cap Bypass
43.10	Branch 2 3 1	SCMOV	Imovlim	ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds , so branch series cap Bypass (No change. Device was already Bypass)
44.00	Bus 3	Bus	User	Clear Fault
44.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp >= Iinsert, timer for Tinsert sta23.rted
46.00	Branch 2 3 1	SCMOV	Iinsert	Transition decision from BYPASS to NORMAL because ItotAmp/Icrated < Iinsert for Tinsert seconds. Schedule Immediately
46.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp/Icrated < Iinsert for Tinsert seconds, so series cap Not Bypass
61.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
61.00	Branch 2 3 1	SCMOV	Ithresh	Transition from NORMAL to CAP+MOV because ItotAmp/(Icrated*Icappro) > Ithresh. Done Immediately
62.00	Bus 3	Bus	User	Apply Fault with Impedance (3PB)
62.00	Branch 2 3 1	SCMOV	Imovlim	Timer Started: ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds
62.00	Branch 2 3 1	SCMOV	Icaplim	Timer Started: IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds
62.10	Branch 2 3 1	SCMOV	Imovlim	Transition decision from CAP+MOV to BYPASS because ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds . Schedule with delay Imovdly=1 seconds.
62.20	Branch 2 3 1	SCMOV	Icaplim	Transition decision from CAP+MOV to BYPASS because IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds . Schedule with delay Icapdly=0.8 seconds.
62.30	Branch 2 3 1	SCMOV	Transition	Timer Started: EnergyMOV > Enerlim for Enerdlg=1 seconds
63.00	Branch 2 3 1	SCMOV	Icaplim	IcapAmp/Icrated > Icaplim for IcapTup=0.2 seconds , so branch series cap Bypass
63.10	Branch 2 3 1	SCMOV	Imovlim	ImovAmp/Icrated > Imovlim for ImovTup=0.1 seconds , so branch series cap Bypass (No change. Device was already Bypass)
64.00	Bus 3	Bus	User	Clear Fault
64.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp >= Iinsert, timer for Tinsert started
66.00	Branch 2 3 1	SCMOV	Iinsert	Transition decision from BYPASS to NORMAL because ItotAmp/Icrated < Iinsert for Tinsert seconds. Schedule Immediately
66.00	Branch 2 3 1	SCMOV	Iinsert	ItotAmp/Icrated < Iinsert for Tinsert seconds, so series cap Not Bypass
66.00	Branch 2 3 1	SCMOV	Enerlim	Timer Started: EnergyMOV > Enerlim for Enerdlg=1 seconds
67.00	Branch 2 3 1	SCMOV	Enerlim	Transition decision from CAP+MOV to BYPASS because EnergyMOV > Enerlim for Enerdlg=1 seconds . Schedule Immediately.
67.00	Branch 2 3 1	SCMOV	Enerlim	EnergyMOV > Enerlim for Enerdlg=1 seconds , so branch series cap Bypass