

Over Excitation Limiter BASOEL2



Model supported by PSSE

Following are Initialized at start of simulation

```
OEL1Active = OEL2Active = OEL3Active = FALSE
OEL1Limit  = OEL2Limit  = OEL3Limit  = FALSE
OEL1Timer  = OEL2Timer  = OEL3Timer  = 0    // timer is used to accumulate time either to set or reset OELRef
```

Update Each Time Step

```
// This local procedure is called three times (once for each timer)
LocalProcedureForTimer( Inputs: theOEL, theOET, IncrementTime; Outputs: theOELLimit, theOELActive, theOELTimer)
// use 0.999 multiplier to make it act above the limit until it drops 0.1% below the limit
// This helps prevent a numerical oscillation if the system is coming to a steady state near theOEL threshold
If theOELLimit then
    theOELLimit = (State1 > theOEL*0.999)
Else
    theOELLimit = (State1 > theOEL)
EndIf
If (    theOELLimit) and (not theOELActive) then
    theOELTimer = theOELTimer + IncrementTime // count up
ElseIf (not theOELLimit) and (    theOELActive) then
    theOELTimer = theOELTimer - IncrementTime // count down
EndIf
If theOELTimer <= 0 Then
    // When timer drops below 0 then it will reset the respective Active flag
    theOELTimer = 0
    theOELActive = FALSE // reset the Timer
ElseIf theOELTimer >= (theOET) then
    // When timer goes above the theOET time for the respective timer then the Active Flag is set
    If theOET > 0.000001 Then theOELTimer = theOET
    Else theOELTimer = 0.000001
    EndIf
    theOELActive = TRUE // Timer has expired so OELRef has an output and VOELout may be non-zero
EndIf
```

IncrementTime = time difference between now and the previous time step

```
LocalProcedureForTimer ( OEL1, OET1, IncrementTime, OEL1Limit, OEL1Active, OEL1Timer )
LocalProcedureForTimer ( OEL2, OET1 + OET2, IncrementTime, OEL2Limit, OEL2Active, OEL2Timer )
LocalProcedureForTimer ( OEL3, OET1 + OET2 + OET3, IncrementTime, OEL3Limit, OEL3Active, OEL3Timer )
```

Calculation of VOELout

```
If (OELFlag <> 0) AND
( (OEL1Active AND OEL1Limit) // VOELout is only used if at least
OR (OEL2Active AND OEL2Limit) // one of the 3 timers is
OR (OEL3Active AND OEL3Limit) // BOTH active AND the OELsens is above the OEL limit threshold
)
Then Calculate VOELout from the block diagram
Else VOELout = 0
```

Calculation of OELRef

```
OELRef = OEL1
If OEL2Active AND (OELRef > OEL2) then OELRef = OEL2
If OEL3Active AND (OELRef > OEL3) then OELRef = OEL3
```