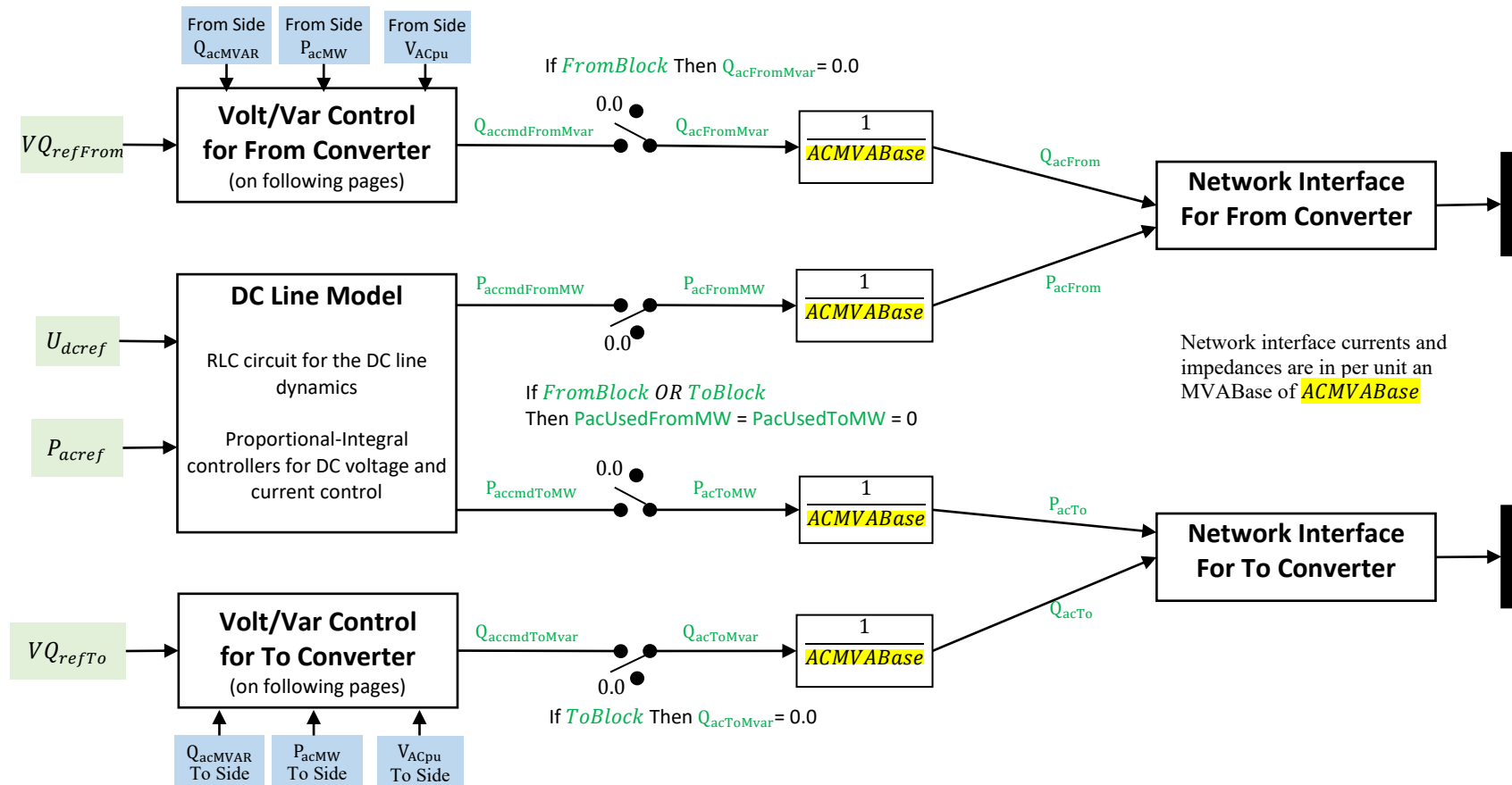


VSC DC Line VHVDC2

VSC DC Line Generic Model VHVDC2

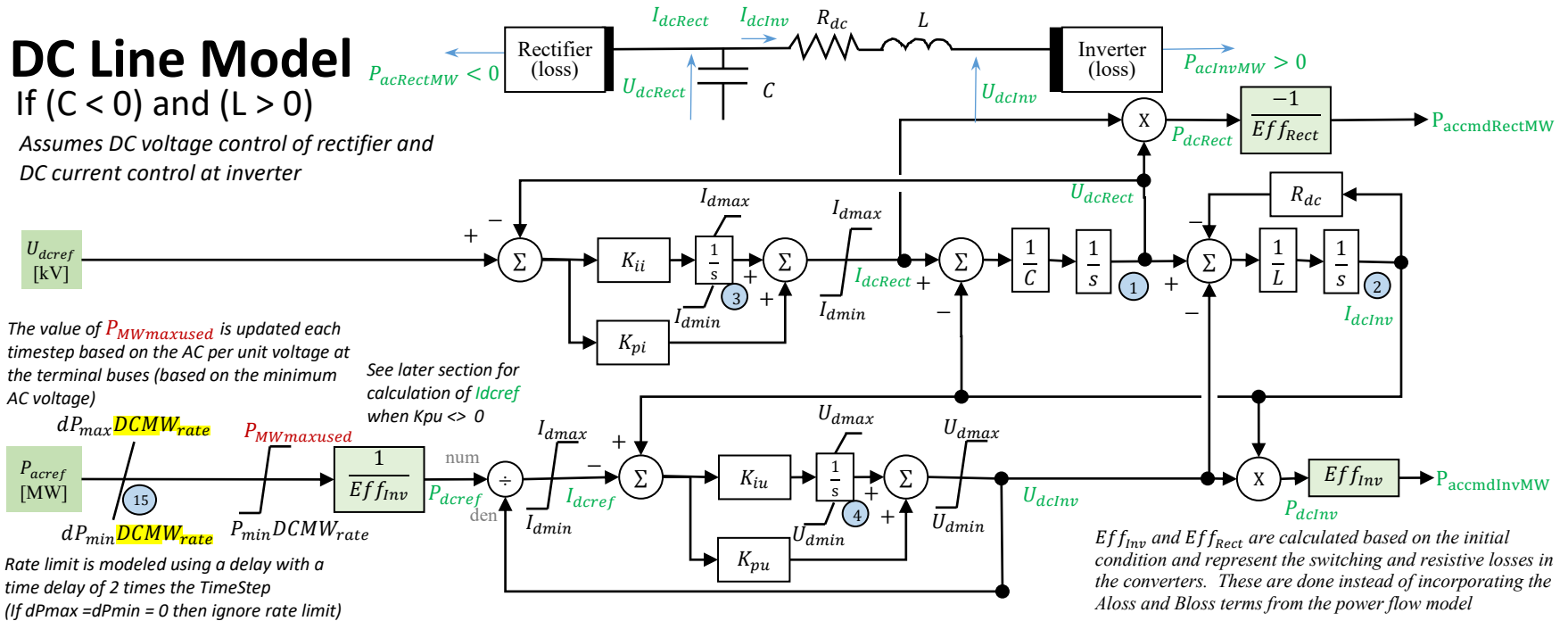
Input parameters for converters for these models are given at the *From* and *To* end of the VSC DC Line model as defined in the power flow data record. The parameters related to the DC line model on the next page refer to either the *Rectifier* or *Inverter* side. As part of initializing this model we determine side is the Rectifier and Inverter based on the initial condition direction of the MW flow. The Inverter is the side that is injecting AC power into the AC network, while the Rectifier is absorbing power from the AC network.

Note: Differences between VHVDC2 and VHVDC1 are highlighted in yellow

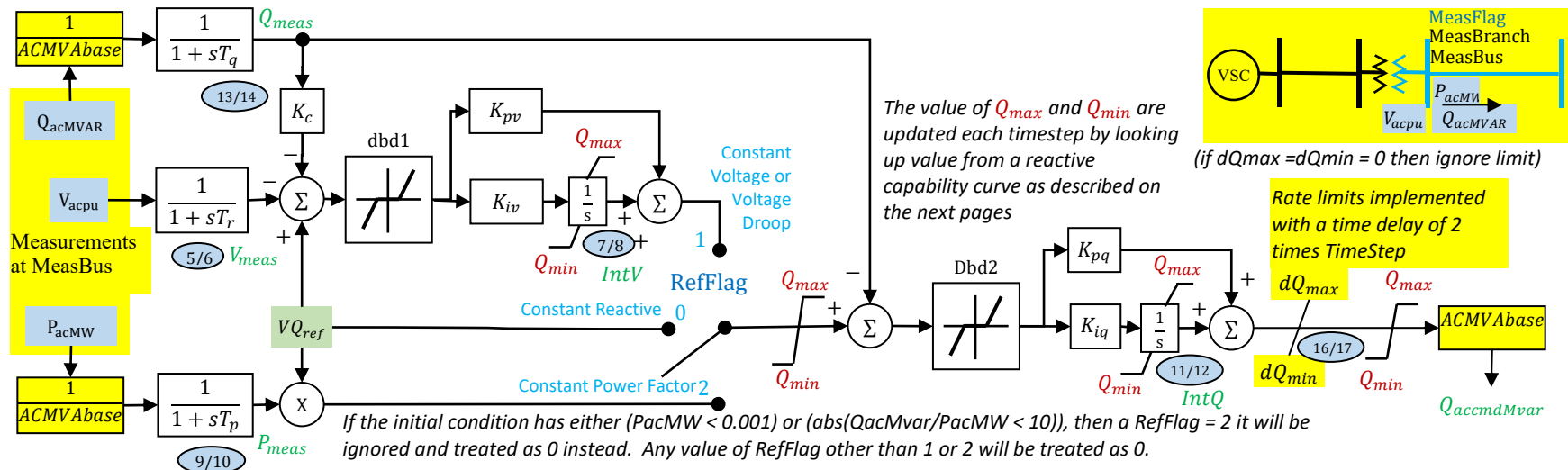


If $(C < 0)$ and $(L > 0)$

Assumes DC voltage control of rectifier and DC current control at inverter



Mvar Control for each Converter is the Same

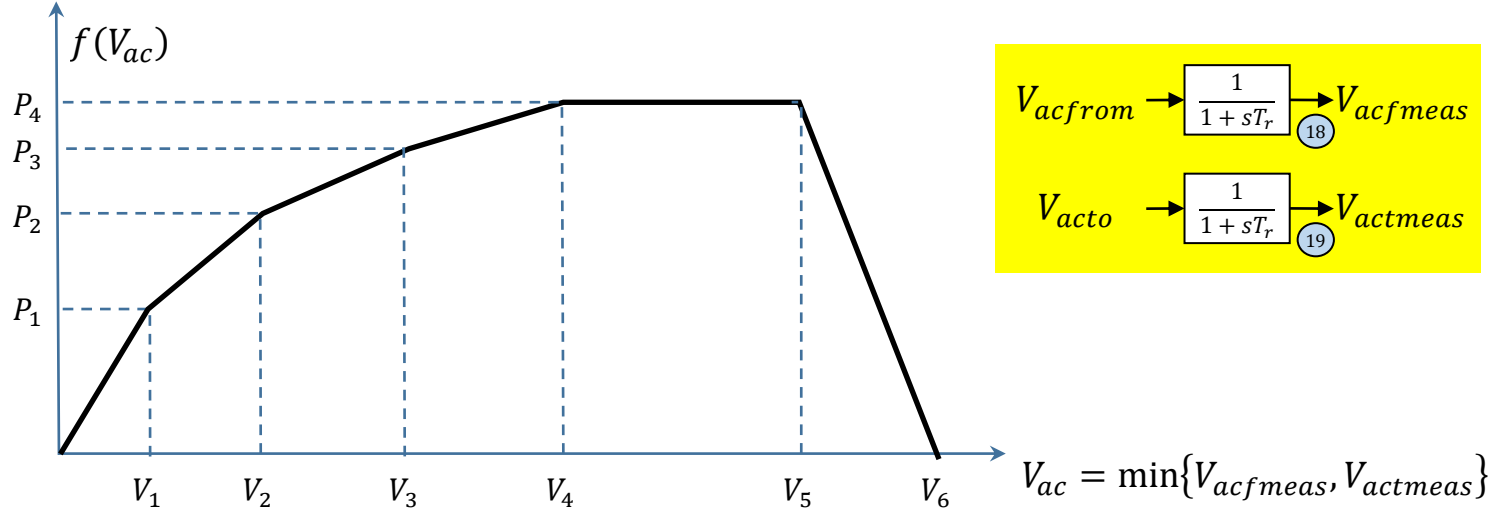


Real Power Limit

If **VPFlag** = 0 then this curve is ignored and **PMWMaxUsed** = **Pmax*****DCMWRate**.

The six (6) point piece-wise linear curve defined next allows an emulation of the reduction on the maximum allowable power reference (Pmax) as a function of the available ac voltage. This function acts to dynamically change Pmax as shown in the DC Line Model control. This is defined by the ten (10) parameters, P1, P2, P3, P4, V1, V2, V3, V4, V5 and V6.

$$P_{MWmaxused} = f(V_{ac}) * DCMW_{Rate} \text{ where } V_{ac} = \min\{V_{acfmeas}, V_{actmeas}\} = \min\{\text{state18}, \text{state19}\}$$



Pseudo Code for Real Power Limit

```

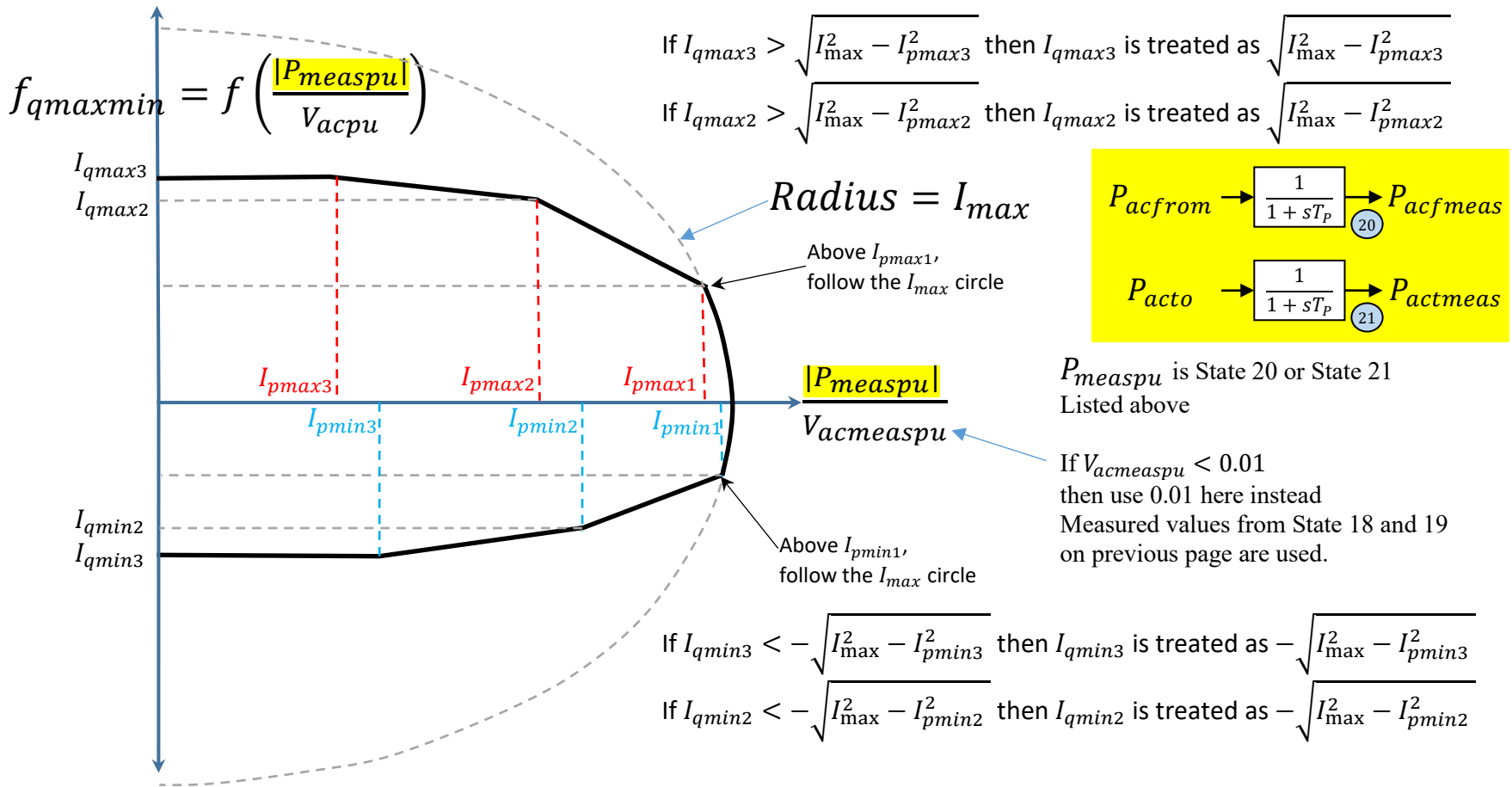
If VPFlag = 0 then
    PmaxUsed = Pmax
Else
    _Vfrom = measured From converter terminal AC voltage (State 18)
    _Vto   = measured To converter terminal AC voltage (State 19)
    If _Vfrom < _Vto then _V = _Vfrom
    else _V = _Vto
    if (_V <= 0) then PmaxUsed = 0
    elseif (_V <= v1) then PmaxUsed = 0 + ((p1 - 0)/(v1 - 0))*(_V - 0)
    elseif (_V <= v2) then PmaxUsed = p1 + ((p2 - p1)/(v2 - v1))*(_V - v1)
    elseif (_V <= v3) then PmaxUsed = p2 + ((p3 - p2)/(v3 - v2))*(_V - v2)
    elseif (_V <= v4) then PmaxUsed = p3 + ((p4 - p3)/(v4 - v3))*(_V - v3)
    elseif (_V <= v5) then PmaxUsed = p4
    elseif (_V <= v6) then PmaxUsed = p4 + ((0 - p4)/(v6 - v5))*(_V - v5)
    else
        PmaxUsed = 0
    endif
    if PmaxUsed > Pmax then PmaxUsed = Pmax
EndIf
PMWmaxUsed = PmaxUsed * DCMWRate

```

Reactive Capability Curve

If $DFlag = 0$ then only the I_{max} circle is enforced and the piecewise-linear curve is ignored.

The current capability of the converters is modeled by a simple capability-curve as next and defined by the eight (8) parameters, I_{max} , I_{pmax1} , I_{pmax2} , I_{pmax3} , I_{qmax2} , I_{qmax3} , I_{qmin2} and I_{qmin3} . Thus, independently, on each converter the current I_p (active current being injected into the grid) is calculated at every time step and thus based on the current value of I_p , I_{qmax}/I_{qmin} is determined from this capability-curve, and subsequently Q_{max}/Q_{min} is calculated as the value of $I_{qmax} \times V_{pu}$ and $I_{qmin} \times V_{pu}$. Thus, real and reactive power are independently controlled on either side of the HVDC link, within the current ratings of the converter. Note that inherently the HVDC link is always in P-priority2 – that is, real power always takes precedence.



Pseudo Code for Reactive Capability Limit Initialization to catch problematic input data

```
// catch degenerate bad input data that may cause sqrt of negative
If Ipmax1 > Imax then Ipmax1 = Imax
If Ipmin1 > Imax then Ipmin1 = Imax
if Ipmax2 > Ipmax1 then Ipmax2 = Ipmax1
if Ipmin2 > Ipmin1 then Ipmin2 = Ipmin1
if Ipmax3 > Ipmax2 then Ipmax3 = Ipmax2
if Ipmin3 > Ipmin2 then Ipmin3 = Ipmin2
```

Pseudo Code for Reactive Capability Limit

```
_V = Converter Terminal AC voltage directly. This is NOT the state 5 or 6 from the converter block diagram
_P = Converter Measured Power in per unit (State 20 or 21)
// In VHVDC1 values were in MW, but VHVDC2 measured P is in per unit
if _V < 0.01 then _Ip = abs(_P/0.01)
Else
    _Ip = abs(_P/_V)
EndIf
if _Ip >= Imax then
    _Iqmax = 0
    _Iqmin = 0
Else
    IqmaxOnCircle = sqrt(Imax*Imax - _Ip*_Ip)
    If DFlag = 0 then
        _Iqmax = +_IqmaxOnCircle
        _Iqmin = -_IqmaxOnCircle
    Else
        _Iqmax1 = +sqrt(Imax*Imax - Ipmax1*Ipmax1)
        _Iqmin1 = -sqrt(Imax*Imax - Ipmin1*Ipmin1)

        if (_Ip >= Ipmax1) then _Iqmax = +_IqmaxOnCircle
        elseif (_Ip >= Ipmax2) then _Iqmax = _Iqmax1 + ((Iqmax2 - _Iqmax1)/(Ipmax2 - Ipmax1))*(_Ip - Ipmax1)
        elseif (_Ip >= Ipmax3) then _Iqmax = Iqmax2 + ((Iqmax3 - Iqmax2)/(Ipmax3 - Ipmax2))*(_Ip - Ipmax2)
        else
            _Iqmax = Iqmax3
        endif

        if (_Ip >= Ipmin1) then _Iqmin = -_IqmaxOnCircle
        elseif (_Ip >= Ipmin2) then _Iqmin = _Iqmin1 + ((Iqmin2 - _Iqmin1)/(Ipmin2 - Ipmin1))*(_Ip - Ipmin1)
        elseif (_Ip >= Ipmin3) then _Iqmin = Iqmin2 + ((Iqmin3 - Iqmin2)/(Ipmin3 - Ipmin2))*(_Ip - Ipmin2)
        else
            _Iqmin = Iqmin3
        endif

        // also limit to within the circle in case the user puts in weird input data
        if _Iqmax > +_IqmaxOnCircle then _Iqmax = +_IqmaxOnCircle
        if _Iqmin < -_IqmaxOnCircle then _Iqmin = -_IqmaxOnCircle
    EndIf
EndIf

QmaxUsed = _Iqmax * _V
QminUsed = _Iqmin * _V
// In VHVDC1 values were in MW and Mvar so we multiplied by MWRate. This is not done in VHVDC2
```

Preventing Numerical problems based on L and C inputs

When modeling the DC line dynamics by entering non-zero L and C parameters numerical problems can occur. Care must be taken to choose properly tuned values for K_{ii} , K_{pi} , K_{iu} , and K_{pu} . In order to attempt to prevent some of these problems, if the value of inductance is too small we will ignore the L and C circuit simulation. This means that you can not model a capacitor without also including the inductor. In addition very small values of C will be ignored and treated as zero. Finally, the natural frequency of the LC circuit is calculated and if it is too high, then either special sub-timestep integration techniques are used or the C value is ignored and C is treated as zero.

```
If L < 5 then // If L is less than 5 millihenry, then ignore DC line equations and treat L and C as 0.
    L = 0
    C = 0
ElseIf C < 5 then // If Capacitance is less than 5 MicroFarad, then treat C as 0.0
    C = 0
Else // If Natural Frequency of the circuit is too high for the integration timestep, then treat C as 0.0
    NaturalFrequency = 1/(2*pi*sqrt(L*1E-3 * C*1E-6))
    Depending on Frequency, special integration techniques may be used, or C may be treated as 0.0
```

Calculation of Idcref value when calculating State Derivatives

When updating state derivatives, the value of I_{dcref} is needed appears to require the value of U_{acinv} to be known. If the value of $K_{pu} = 0$, then the value of U_{acinv} is the output of an integrator which means it can NOT change instantaneously. In this situation, the value of I_{dcref} is calculated simply as you would expect.

```
Idcref = Pdcref/Udcinv = Pdcref/State4
```

However, if $K_{pu} \neq 0$, then the numerical simulation will allow the DC voltage to change instantaneously. One option would be to use the previous timestep's U_{acinv} value, however this is not necessary, and we can instead solve algebraic equations as follows.

```
Pdcref = Idcref*Udcinv
UdcInv = Kpu*(-Idcref + State2) + State4
```

Now substitute the second equation into the first equation to get

```
Pdcref = Idcref*( Kpu*(-Idcref + State2) + State4)
```

Expand terms to get

```
Pdcref - Idcref*(-Kpu*Idcref + Kpu*State2 + State4) = 0
```

Group terms for **Idcref**

```
Pdcref -(Kpu*State2 + State4)*Idcref + Kpu*sqr(Idcref) = 0
```

Use quadratic equation to solve this.

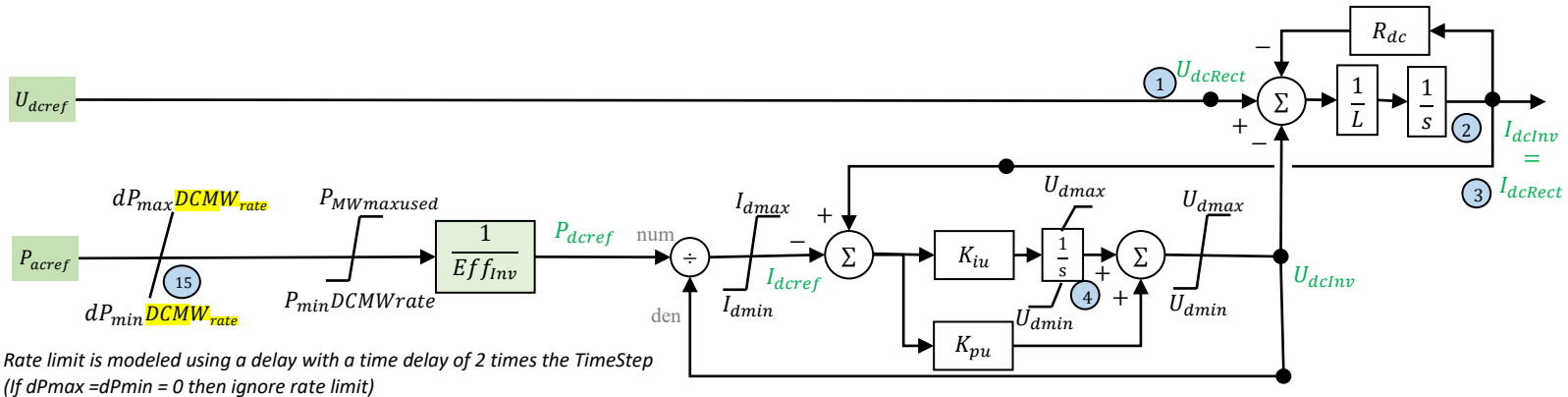
```
_Argument = sqr(Kpu*State2 + State4) - 4*Pdcref*Kpu
// Negative value possible for silly input data. Just switch to give maximum power transfer instead
If _Argument < 0 then _Argument = 0
// Use the negative value because we want the low current/high voltage solution
Idcref = 1/(2*Kpu)*(Kpu*State2 + State4 - sqrt(_Argument))
```

Finally, enforce limits

```
If Idcref > Idmax then Idcref = Idmax
ElseIf Idcref < Idmin then Idcref = Idmin
```

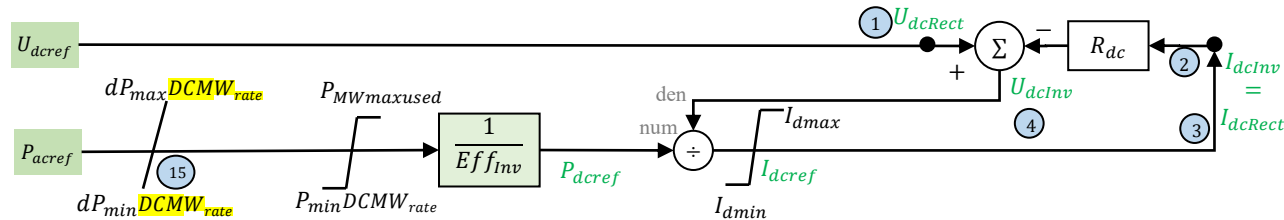
DC Line Model Equations when C=0 and L>0

U_{dcRect} , I_{dcRect} become algebraic states and the DC line equations simplify to following



DC Line Model Equations when C=0 and L=0

U_{dcRect} , I_{dcRect} , U_{dcInv} , I_{dcInv} become algebraic states and the DC line equations simplify to following



The algebraic equations here to solve are

$$P_{dcref} = U_{dcinv} * I_{dcRef}$$

$$U_{dcinv} = U_{dcRef} - R_{dc} * I_{dcRef}$$

Group terms for I_{dcRef}

$$P_{dcref} - U_{dcRef} * I_{dcRef} + R_{dc} * \text{sq}(I_{dcRef}) = 0$$

Use quadratic equation to solve this.

$$\text{Argument} = \text{sq}(U_{dcRef}) - 4 * P_{dcref} * R_{dc}$$

// Negative value possible for silly input data. Just switch to give maximum power transfer instead

If $\text{Argument} < 0$ then $\text{Argument} = 0$

// Use the negative value because we want the low current/high voltage solution

$$I_{dcRef} = 1 / (2 * R_{dc}) * (U_{dcRef} - \text{sqrt}(\text{Argument}))$$

Finally, enforce limits

If $I_{dcRef} > I_{dmax}$ then $I_{dcRef} = I_{dmax}$

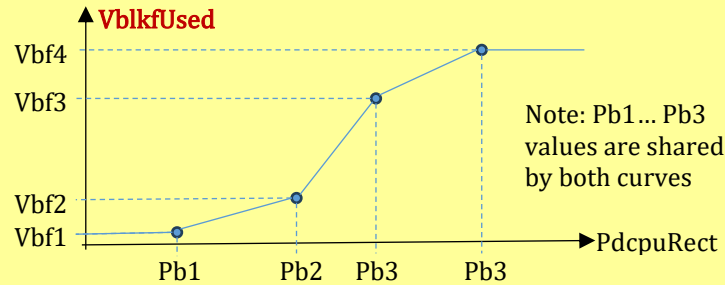
ElseIf $I_{dcRef} < I_{dmin}$ then $I_{dcRef} = I_{dmin}$

Converter Blocking Code for each Converter

Simple emulation of blocking is done based on the AC per unit voltage at each converter. The following logic is evaluated at each time step of the simulation for the FromEnd Converter and ToEnd Converter.

A lookup function is used for specifying the blocking ac voltage as a function of the per unit DC power at the initial condition. Note that both curves use the same x-axis values and . Not that Pb values must be non-decreasing. There are no restrictions on Vb* values. It is not required to input all 4 pairs of values. The first Pb value for which $Pb[k] < Pb[k-1]$ will signify the end of the Pb/Vb pairs.

VblkfUsed is calculated at initialization. It is a function of the initial P in per unit on the dcMWRate at the rectifier end

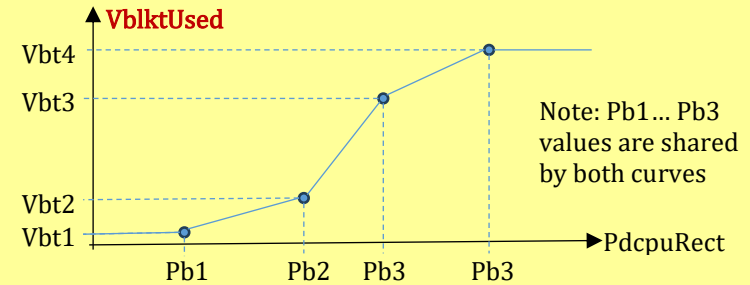


```

_V = Measured Vpu at the From End (State 5)
if _V < VblkfUsed
    FromBlocked = true
elseif FromBlocked
    if _V < Vunblk
        FromTimerActive = false
    elseif (plldelay < TimeStep)
        FromTimerActive = false
        FromBlocked = false
    elseif (not FromTimerActive)
        FromTimerActive = true
        FromTimerUnblock = PresentTime
    elseif (PresentTime - FromTimerUnblock) > plldelay
        FromTimerActive = false
        FromBlocked = false
    endif
endif
endif

```

VblkktUsed is calculated at initialization. It is a function of the initial P in per unit on the dcMWRate at the rectifier end



```

_V = Measured Vpu at the To End (State 6)
if _V < VblkktUsed
    ToBlocked = true
elseif ToBlocked
    if _V < Vunblk
        ToTimerActive = false
    elseif (plldelay < TimeStep)
        ToTimerActive = false
        ToBlocked = false
    elseif (not ToTimerActive)
        ToTimerActive = true
        ToTimerUnblock = PresentTime
    elseif (PresentTime - ToTimerUnblock) > plldelay
        ToTimerActive = false
        ToBlocked = false
    endif
endif
endif

```

A Pb/Vb table may look as follows

Pb1 = 0.2	Vb1 = 0.20	Pb1 = 0.2	Vb1 = 0.20
Pb2 = 0.6	Vb2 = 0.30	Pb2 = 0.6	Vb2 = 0.30
Pb3 = 0.8	Vb3 = 0.30	Pb3 = 0.0	Vb3 = 0.30 // These are ignored because Pb3 < Pb2
Pb4 = 0.9	Vb4 = 0.35	Pb4 = 0.0	Vb4 = 0.35

When a converter terminal is blocked, then that converter will be treated as a constant current injection of 0.0 immediately. The other side of the VSC DC Line will have its real power command set to 0.0 immediately, but the reactive power command will continue to be used.

In addition, when either converter is blocked, the inputs to the DC line dynamics “Pacref” and “Idcref” will be immediately treated as though they are a value of 0.0. This means that if the from side converter blocks the from side AC power is instantly set to 0.0. At the TO side converter however, the real power is set to 0.0 immediately, but the reactive power will continue to be injected at that side. The DC line dynamics will continue to simulate however and thus the DC current will transition down to zero as the DC line dynamics are simulated and Pacref value forces the DC current to zero.

Network Interface Equations

The network interface equations for VHVDC2 is modeled as a constant power injection.

