# Multi-Terminal DC Line Dynamic Model for the Pacific DC Intertie

## Jamie Weber

lauf@powerworld.com
217-384-6330 ext. 13

2001 South First Street
Champaign, Illinois  61820
+1 (217) 384.6330

support@powerworld.com
http://www.powerworld.com

# PDCI Model

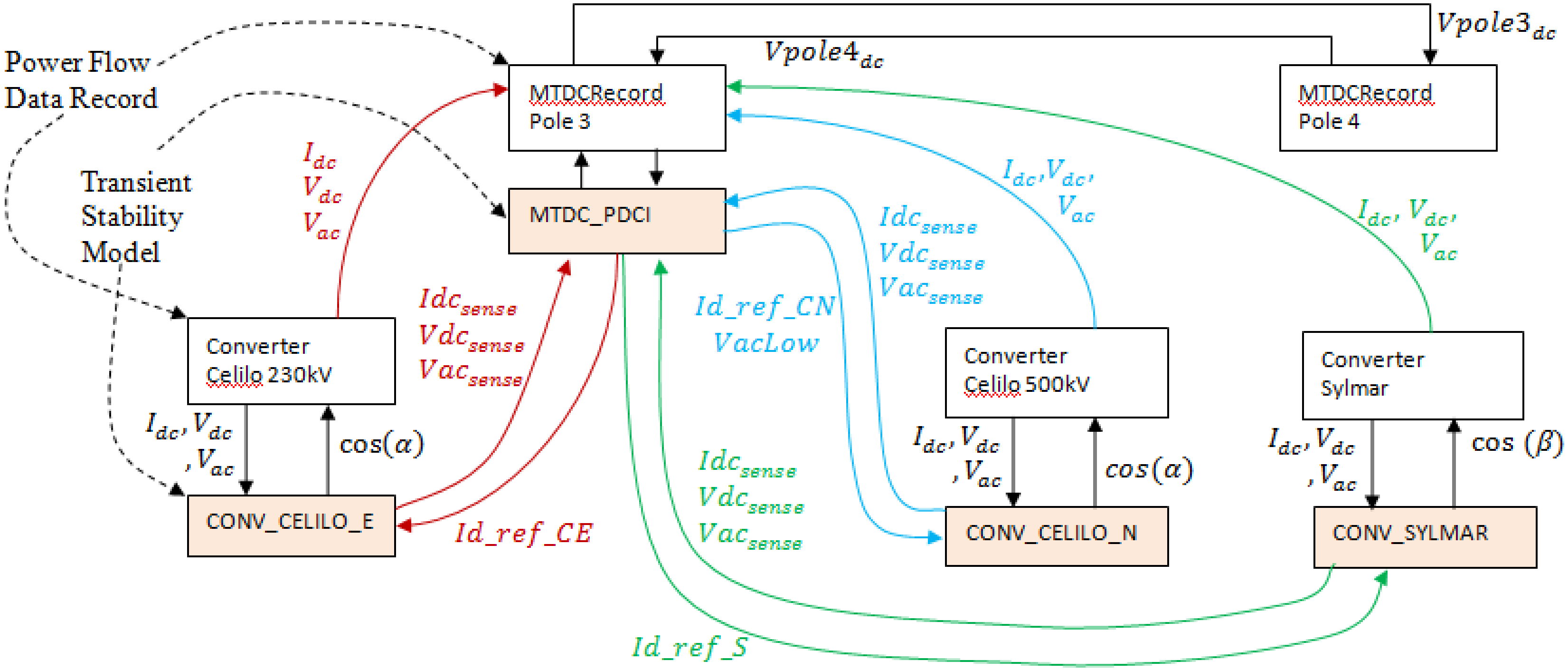
- Documentation
  - Existing text files contained the actual user-defined model implementation of the PDCI model (*pdci_ns3.p* and *pdci_sn3.p*)
  - Dmitry Kosterev provided partial documentation describing the pdci_sn3.ps file. Most helpful for the newer CELILO 500 kV converters which had recently been upgraded
- PowerWorld took the actual 1,300 lines of code in the *pdci_ns3.p* file and determined the block diagram being modeled for this important device (also looked through *pdci_sn3.p*)
- The *pdci_ns3.p* code encompasses a model for controlling the firing angle on two converters at Celilo and one converter at Sylmar

# PowerWorld Implementation and User Experience

- Code Implementation
  - Simulator’s internal code has been written to make the interaction of the dynamic multi-terminal DC line model and converters with the network boundary equations generic
    - This will make adding new DC line models much easier.
    - Also will permit the creation of an interface to a user-defined multi-terminal DC line model
- In Simulator Version 16 patches, the user need only check a box asking that this model be used
  - Simulator will look for the PDCI in the case and automatically include the dynamic model if appropriate
  - All parameters of the model are hard-coded then
- Version 17 will allow the user to explicitly add the dynamic models and also see the internal states of these models if desired
  - All parameters of the model will remain hard-coded for model
  - May change this eventually if desired

# Implementation Overview in Simulator

- Assign dynamic model **MTDC_PDCI** to the multi-terminal DC Line record
- Assign appropriate dynamic converter models to the various DC converters: **CONV_CELILO_E**, **CONV_CELILO_N**, **CONV_SYLMAR**



# MTDC_PDCI

- The model is assigned to one Multi-Terminal DC record.
  - For the PDCI, two separate MTDC records are modeled, one for each pole of the PDCI
- Inputs
  - **From DC Converters**: States of the sensed DC Current, DC Voltage, and AC Voltage at each converter
  - **AC Network**: Direct access to network boundary equation AC voltages is also used
  - **Other MTDC_PDCI**: There is some feedback between the two poles in the DC voltage measurement
- Outputs
  - Feeds a reference current to each DC Converter model: *id_ref_CN*, *id_ref_CE*, and *id_ref_S*
  - Also feeds a flag for *VacLow* as needed to the CONV_CELILO_N

# MTDC_PDCI:
# Low Voltage Detection

Direct readings of AC voltage at converter buses

$If\ (Vac_{Celilo230} < 0.5)\ OR\ (Vac_{Celilo500} < 0.5)\ Then\ \boldsymbol{VacLowCelilo} = TRUE$

$If\ (Vac_{Celilo230} > 0.6)\ AND\ (Vac_{Celilo500} < 0.6)\ Then\ \boldsymbol{VacLowCelilo} = FALSE$

$If\ (Vac_{Sylmar} < 0.5)\ Then\ \boldsymbol{VacLowSylmar} = TRUE$

$If\ (Vac_{Sylmar} > 0.6)\ Then\ \boldsymbol{VacLowSylmar} = FALSE$

State 2 from the Celilo500 and Sylmar converter

$If\ (Vdc_{senseCelilo500} < 150)\ OR\ (Vdc_{senseSylmar} < 150)\ Then\ \boldsymbol{VdcLow} = TRUE$

$If\ (Vdc_{senseCelilo500} > 350)\ AND\ (Vdc_{senseSylmar} > 350)\ Then\ \boldsymbol{VdcLow} = FALSE$

$\boldsymbol{VacLow} = \boldsymbol{VacLowCelilo}\ OR\ \boldsymbol{VdacLowSylmar}$

$\boldsymbol{VlowFlag} = \boldsymbol{VacLow}\ OR\ \boldsymbol{VdcLow}$

**Average DC voltage across both poles of the MTDC**

## DC Voltage Measurement Freeze

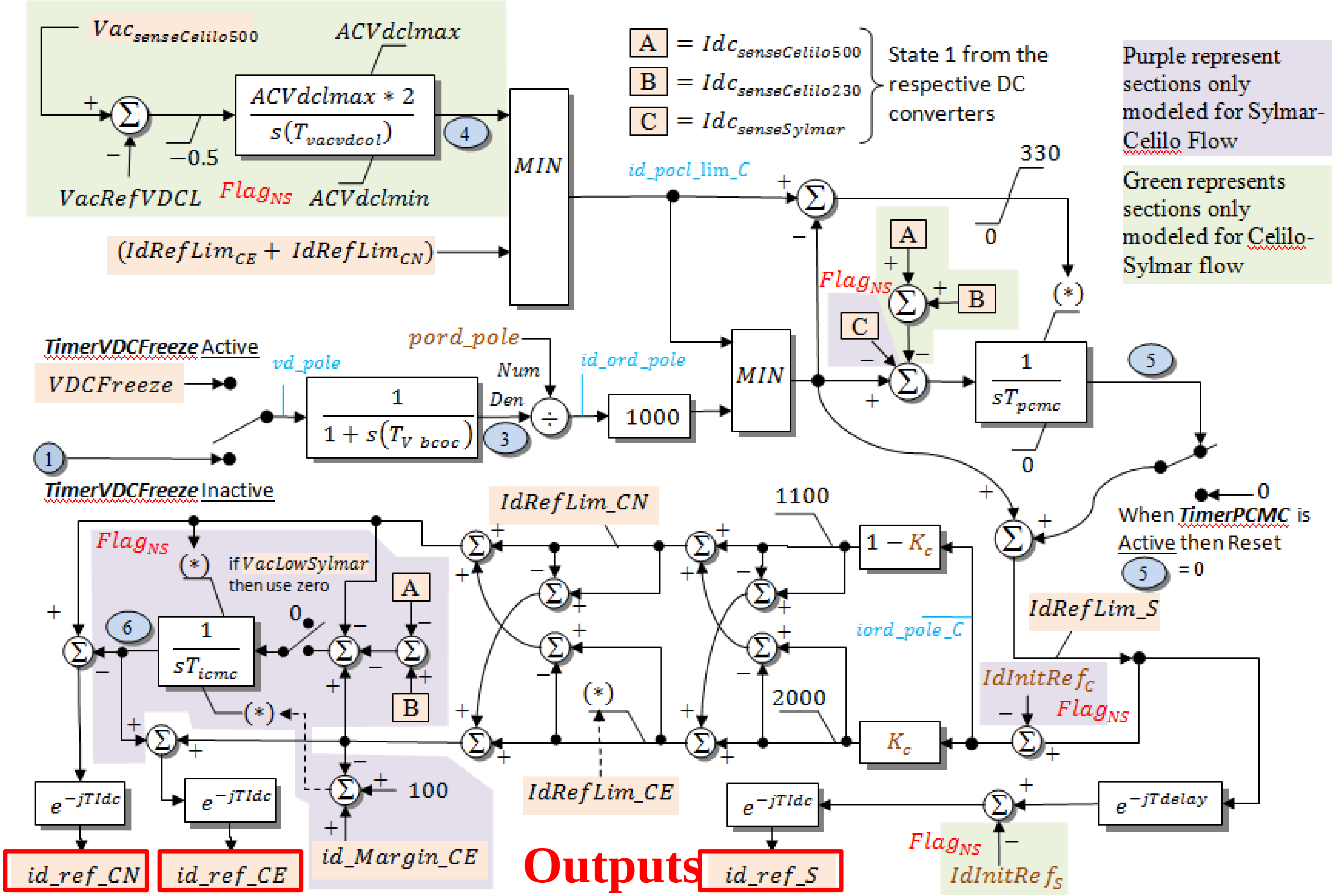
If ***VLowFlag*** = TRUE then activate a timer ***TimerVDCFreeze*** and set timer to ***Tdel***$_{friz}$. Continue to set this timer up to ***Tdel***$_{friz}$ as long as ***VLowFlag*** is TRUE. If ***VLowFlag*** become FALSE then start having the timer count down to zero. Once the ***TimerVDCFreeze*** reaches zero then make it inactive.

Whenever ***TimerVDCFreeze*** is inactive then
  Set ***VDCFreeze*** = ②
  and pass ① as the DC voltage to Current Order Allocation Calculation
Whenever ***TimerVDCFreeze*** is active then
  Pass the variable ***VDCFreeze*** as the DC voltage to the Current Order Calculation

## Pole Current Margin Compensator

If ***VLowFlag*** = TRUE then activate a timer ***TimerPCMC*** and set timer to ***Tpcmc_rst***. Continue to set this timer up to ***Tpcmc_rst*** as long as ***VLowFlag*** is TRUE. If ***VLowFlag*** become FALSE then start having the timer count down to zero. Once the ***TimerPCMC*** reaches zero then make it inactive.

# MTDC_PDCI:
# Current Order Allocation



Purple represent sections only modeled for Sylmar-Celilo Flow

Green represents sections only modeled for Celilo-Sylmar flow

**Outputs**

# MTDC_PDCI: Parameters and Initialization

- Parameters and initialization depends on flow direction on the PDCI (North to South) or (South to North)

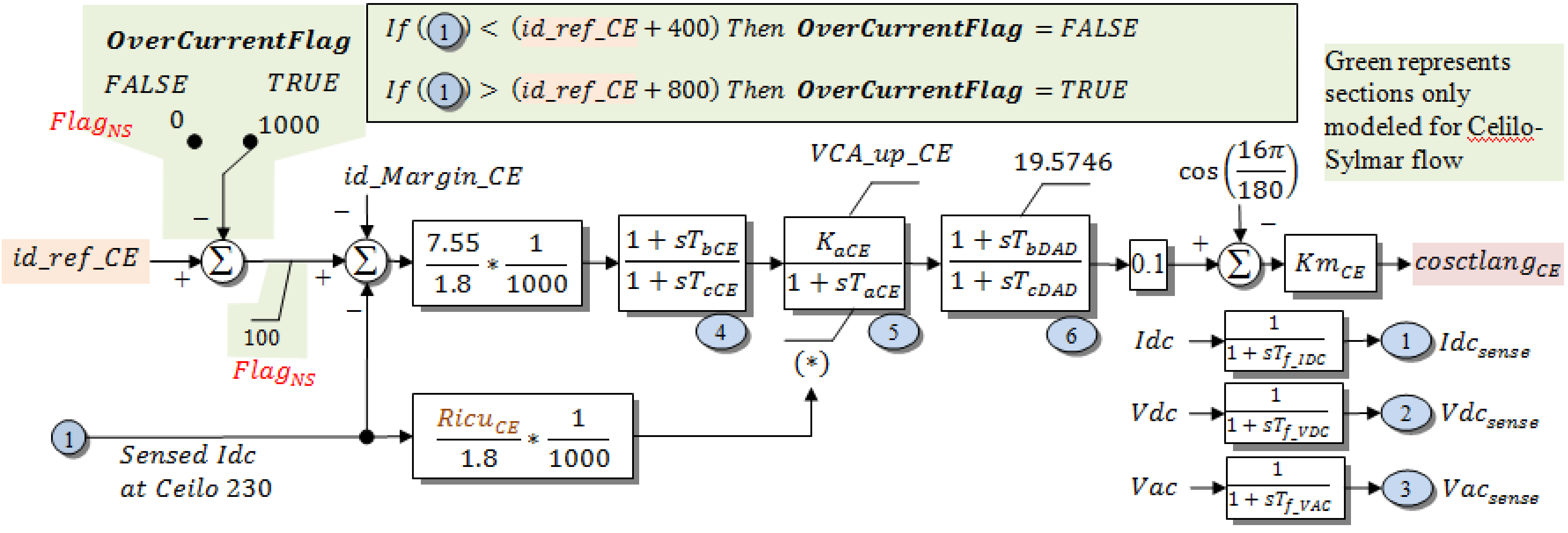
| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $T_{fpcmc\_rst}$ | 0.1 | |
| $T_{VDC\_friz}$ | 0.53 | |
| $Tdel_{friz}$ | 0.44 | |
| $VacRefVDCL$ | 228/230 | n/a |
| $ACVdclmax$ | 3100 | n/a |
| $ACVdclmin$ | 2400 | n/a |
| $T_{vcacvdcol}$ | 0.5 | n/a |
| | | |
| $IdRefLim_{CE}$ | Get from CONV_CELILO_E model | |
| $IdRefLim_{CN}$ | Get from CONV_CELILO_N model | |
| $IdRefLim_S$ | Get from CONV_SYLMAR model | |
| $T_{v\ bcoc}$ | 0.05 | |
| $T_{pcmc}$ | 0.5120 | |
| $T_{icmc}$ | n/a | 0.2500 |
| $Id\_margin\_CE$ | Get from CONV_CELILO_E model | |
| $TIdc$ | 0.5 cycles | |
| $Tdelay$ | 2 cycles | |

| Initialization Reference Values | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $pord\_pole$ | Initialize to Sum of Psched at two Celilo Converters | Initialize to Psched at the Sylmar Converter |
| $K_C$ | Initialize based on the following equation<br>$K_C = \frac{id\_ref\_CE}{id\_ref\_CE + id\_ref\_CN}$ | |
| $VacLowSylmar$ | FALSE | |
| $VacLowCelilo$ | FALSE | |
| $VdcLow$ | FALSE | |
| $VLowFlag$ | FALSE | |
| $IdInitRef_C$ | n/a | Initialize equal to State 14 to handle difference between $id\_ord\_pole$ and $id\_ref\_S$ |
| $IdInitRef_S$ | Initialize equal to State 14 to handle difference between $id\_ord\_pole$ and $id\_ref\_CN + id\_ref\_CE$ | n/a |

# CONV_CELILO_E

- Model assigned to the converter at the 230 kV bus at Celilo – the “Existing” Control
- Inputs
  - Reference current *id_ref_CE* from MTDC_PDCI
  - Network boundary equation converter values: Idc, Vdc, Vac
- Output
  - Cosine of the control angle (Alpha or Beta as appropriate)

# CONV_CELILO_E



Green represents sections only modeled for Celilo-Sylmar flow

# CONV_CELILO_E
# Parameters and Initialization

- Parameters and initialization depends on flow direction on the PDCI (North to South) or (South to North)

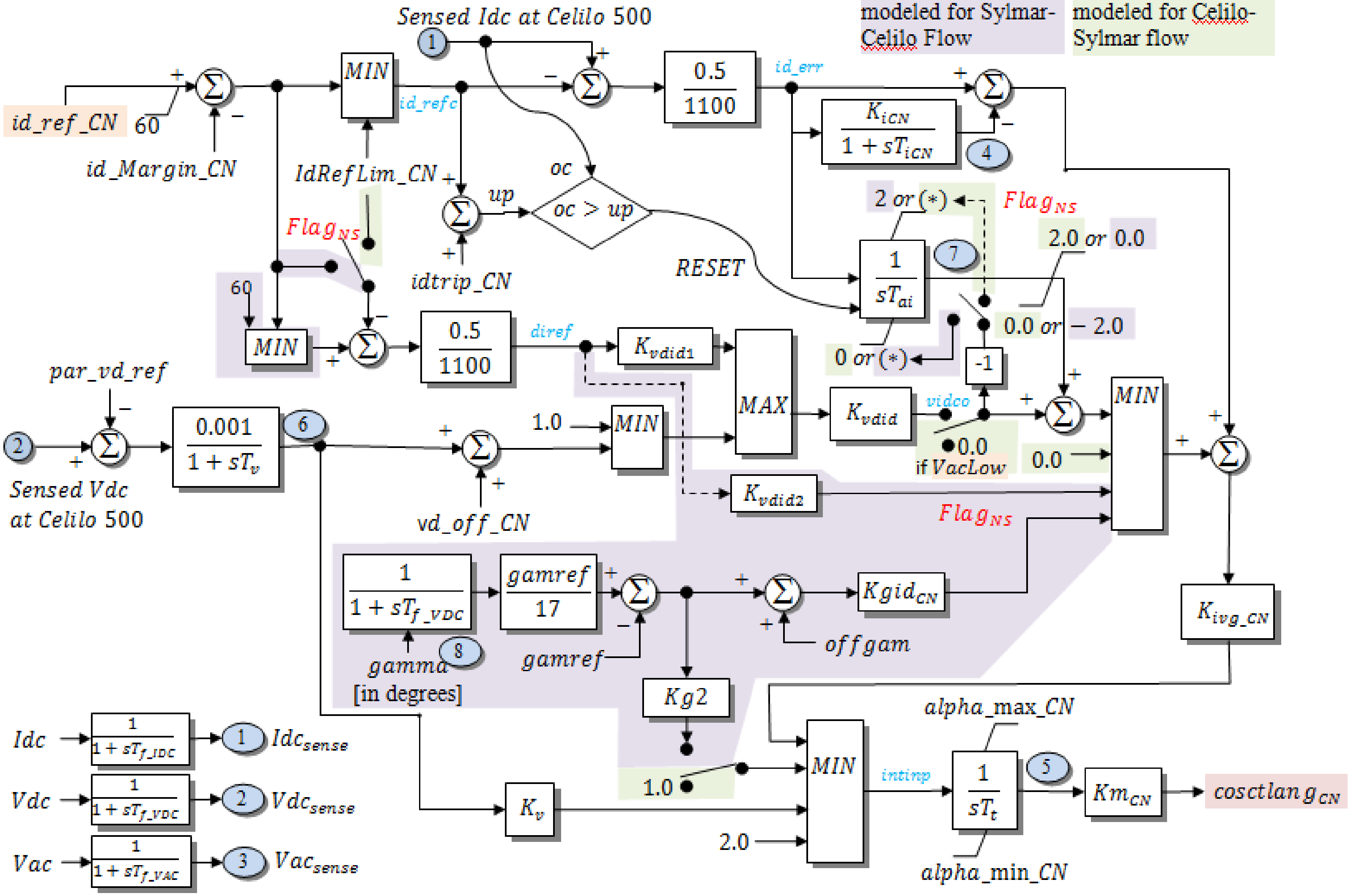
| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $T_{bCE}$ | 0.0323 | |
| $T_{cCE}$ | 0.0323 | |
| $K_{aCE}$ | 330 | 82.5 |
| $T_{aCE}$ | 8.1500 | |
| $VCA\_up\_CE$ | 20 | 8.5 |
| $T_{bDAD}$ | 0 | 1/240 |
| $T_{cDAD}$ | 1/60 | |
| $Km_{CE}$ | 1 | -1 |
| $id\_margin\_CE$ | 0 | 180 |

| Initialization Reference Values | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $Ricu_{CE}$ | 2.2 | Initialize so State 5 is at its lower limit<br>$Ricu_{CE} = \frac{④ * 1.8 * 1000}{①}$ |
| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
| $IdRefLim_{CE}$ | 2160 | |
| $T_{f\_IDC}$ | 0.004 | 0.0084 |
| $T_{f\_VDC}$ | 0.004 | 0.0084 |
| $T_{f\_VAC}$ | 0.0333 | |

# CONV_CELILO_N

- Model assigned to the converter at the 500 kV bus at Celilo – the "New" Control
- Inputs
  - Reference current *id_ref_CN* from MTDC_PDCI
  - VacLow from MTDC_PDCI
  - Network boundary equation converter values: Idc, Vdc, Vac
- Output
  - Cosine of the control angle (Alpha or Beta as appropriate)

# CONV_CELILO_N



Purple represent sections only modeled for Sylmar-Celilo Flow

Green represents sections only modeled for Celilo-Sylmar flow

# CONV_CELILO_N
# Parameters and Initialization

- Parameters and initialization depends on flow direction on the PDCI (North to South) or (South to North)

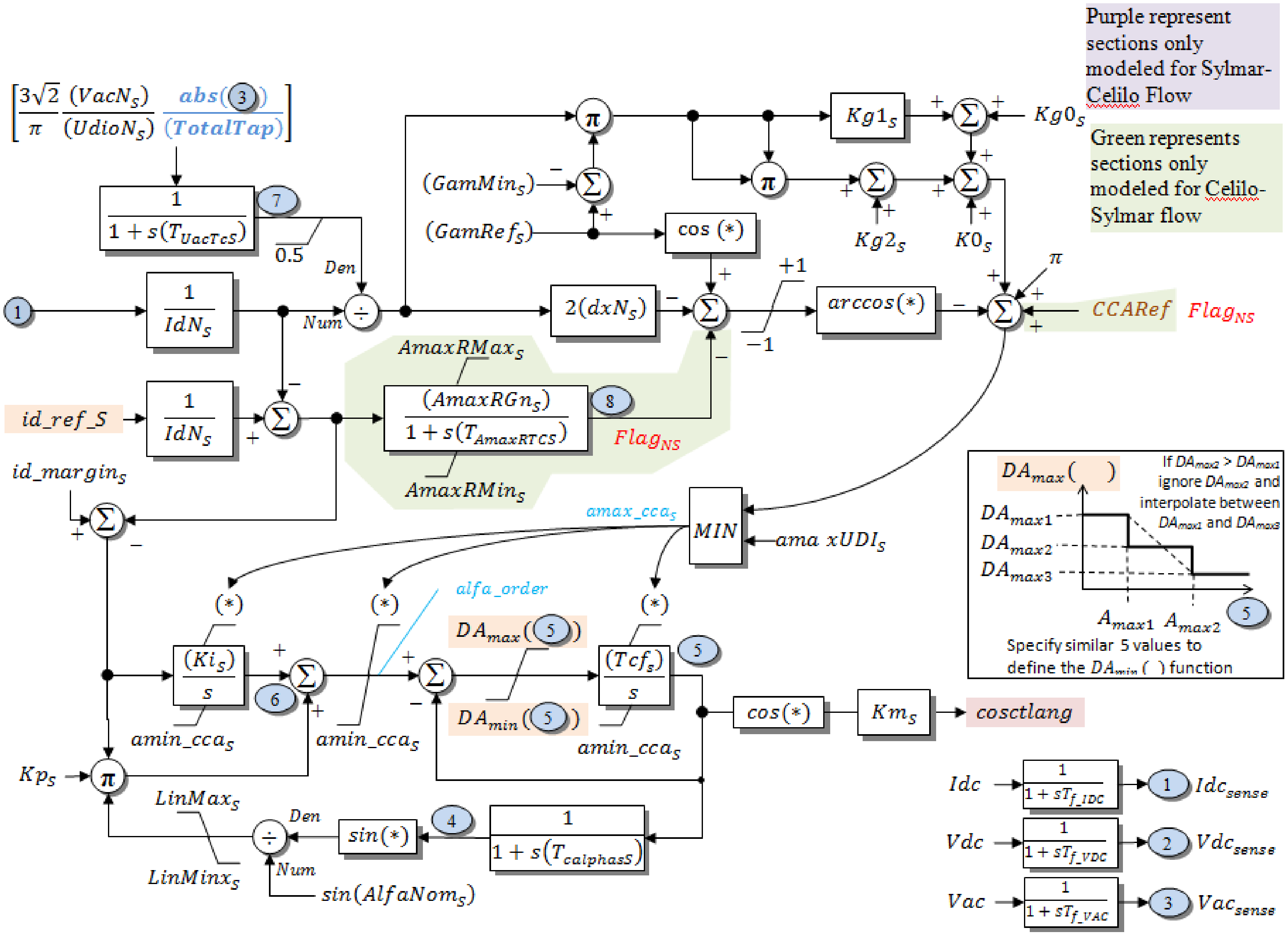
| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $id\_Margin\_CN$ | 0 | 150 |
| $IdRefLim\_CN$ | 1650 | |
| $idtrip\_CN$ | +110 | -110 |
| $id\_min\_min$ | n/a | 60 |
| $K_{iCN}$ | 2/3 | |
| $T_{iCN}$ | 1/120 | |
| $T_{ai}$ | 0.1660 | |
| $K_{vdid1}$ | 0.4400 | |
| $K_{vdid}$ | +0.7575 | -0.7575 |
| $T_v$ | 0.0138 | |
| $par\_vd\_ref$ | 550 | |
| $K_v$ | -0.4000 | |
| $vd\_off\_CN$ | 0.100 | 0.066 |

| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $K_{ivg\_CN}$ | -0.264 | +0.26400 |
| $T_t$ | -1/360 | +1/360 |
| $alpha\_max\_CN$ | +1.000 | +1.91260 |
| $alpha\_min\_CN$ | -1.992 | +0.34700 |
| $Km_{CN}$ | -0.5 | 0.5 |
| $K_{vdid2}$ | n/a | -0.333 |
| $gamref$ | n/a | 0.38428 |
| $Kg2$ | n/a | 0.6 |
| $Kgid_{CN}$ | n/a | 0.11694 |
| $offgam$ | n/a | -0.42743 |
| $T_{f\_IDC}$ | 0.004 | 0.0084 |
| $T_{f\_VDC}$ | 0.004 | 0.0084 |
| $T_{f\_VAC}$ | 0.0333 | |

# CONV_SYLMAR

- Model assigned to the converter at the Sylmar
- Inputs
  - Reference current *id_ref_S* from MTDC_PDCI
  - Network boundary equation converter values: Idc, Vdc, Vac
- Output
  - Cosine of the control angle (Alpha or Beta as appropriate)

# CONV_SYLMAR



Purple represent sections only modeled for Sylmar-Celilo Flow

Green represents sections only modeled for Celilo-Sylmar flow

If $DA_{max2} > DA_{max1}$ ignore $DA_{max2}$ and interpolate between $DA_{max1}$ and $DA_{max3}$

Specify similar 5 values to define the $DA_{min}$( ) function

# CONV_SYLMAR
# Parameters and Initialization

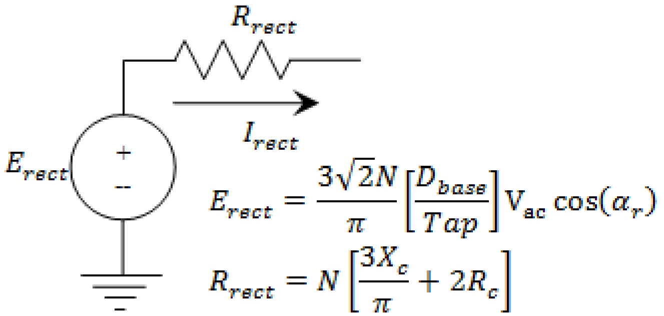
- Parameters and initialization depend on flow direction on the PDCI (North to South) or (South to North)

| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $VacN_S$ | 230.0 | |
| $UdioN_S$ | 286.7 | |
| $T_{UacTcS}$ | 0.1 | |
| $IdN_S$ | 3100.0 | |
| $GamMin_S$ | 15 π/180 | |
| $GamRef_S$ | 17 π/180 | |
| $K0_S$ | -0.018846501 π/180 | |
| $Kg0_S$ | 0.0 π/180 | |
| $Kg1_S$ | -0.601288000 π/180 | |
| $Kg2_S$ | -0.029795800 π/180 | |
| $dxN_S$ | 0.090 | |
| $AmaxRGn_S$ | 0.150 | n/a |
| $T_{AmaxRTCS}$ | 0.002 | n/a |
| $AmaxRMax_S$ | 0.050 | n/a |
| $AmaxRMin_S$ | -0.050 | n/a |
| $id\_margin_S$ | 0.03 | 0.00 |
| $amaxUDI_S$ | 200 π/180 | |
| $amin\_cca_S$ | 95 π/180 | 5 π/180 |
| $T_{f\_IDC}$ | 0.004 | 0.0084 |
| $T_{f\_VDC}$ | 0.004 | 0.0084 |
| $T_{f\_VAC}$ | 0.0333 | |
| $Km_{CN}$ | -1 | 1 |

| Parameter | Celilo-Sylmar (North to South) | Sylmar - Celilo (South to North) |
|---|---|---|
| $Ki_S$ | 180 π/200 | |
| $Kp_S$ | 20 π/180 | |
| $T_{calphasS}$ | 0.003 | |
| $AlfaNom_S$ | 17.5 π/180 | |
| $LinMax_S$ | 1.000 | |
| $LinMinx_S$ | 0.300 | |
| $Tcf_S$ | 1000 | |
| $DA_{max1}$ | 10 π/180 | 90 π/180 |
| $DA_{max2}$ | 2 π/180 | 999 π/180 |
| $DA_{max3}$ | 1 π/180 | 5 π/180 |
| $A_{max1}$ | 110 π/180 | 0 π/180 |
| $A_{max2}$ | 140 π/180 | 120 π/180 |
| $DA_{min1}$ | -5 π/180 | -6 π/180 |
| $DA_{min2}$ | -5 π/180 | -8 π/180 |
| $DA_{min3}$ | -5 π/180 | -15 π/180 |
| $A_{min1}$ | 0 π/180 | 35 π/180 |
| $A_{min2}$ | 10 π/180 | 70 π/180 |
| $CCARef$ | Initialize so State 32 is at the upper limit | n/a |
| $IdRefLim_S$ | 3100 | |
| | | |

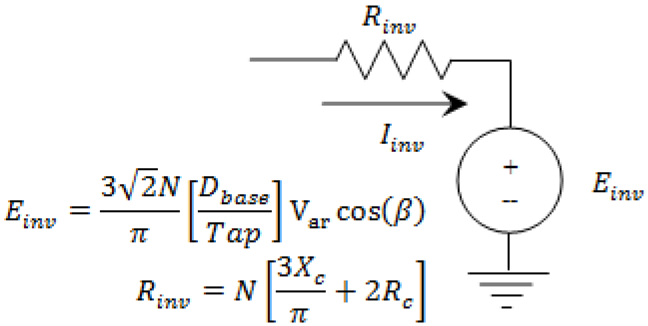
# Handling of the interaction with Network Boundary Equations

- DC Converter equations
  - Written in terms of Alpha at rectifiers
  - Written in terms of Beta at inverters
    - “Beta” is different than the Gamma traditional used when writing the static power flow equations
  - Equations are as follows
    - Also force currents to be positive



$$E_{rect} = \frac{3\sqrt{2}N}{\pi}\left[\frac{D_{base}}{Tap}\right]\mathrm{V_{ac}}\cos(\alpha_r)$$

$$R_{rect} = N\left[\frac{3X_c}{\pi} + 2R_c\right]$$



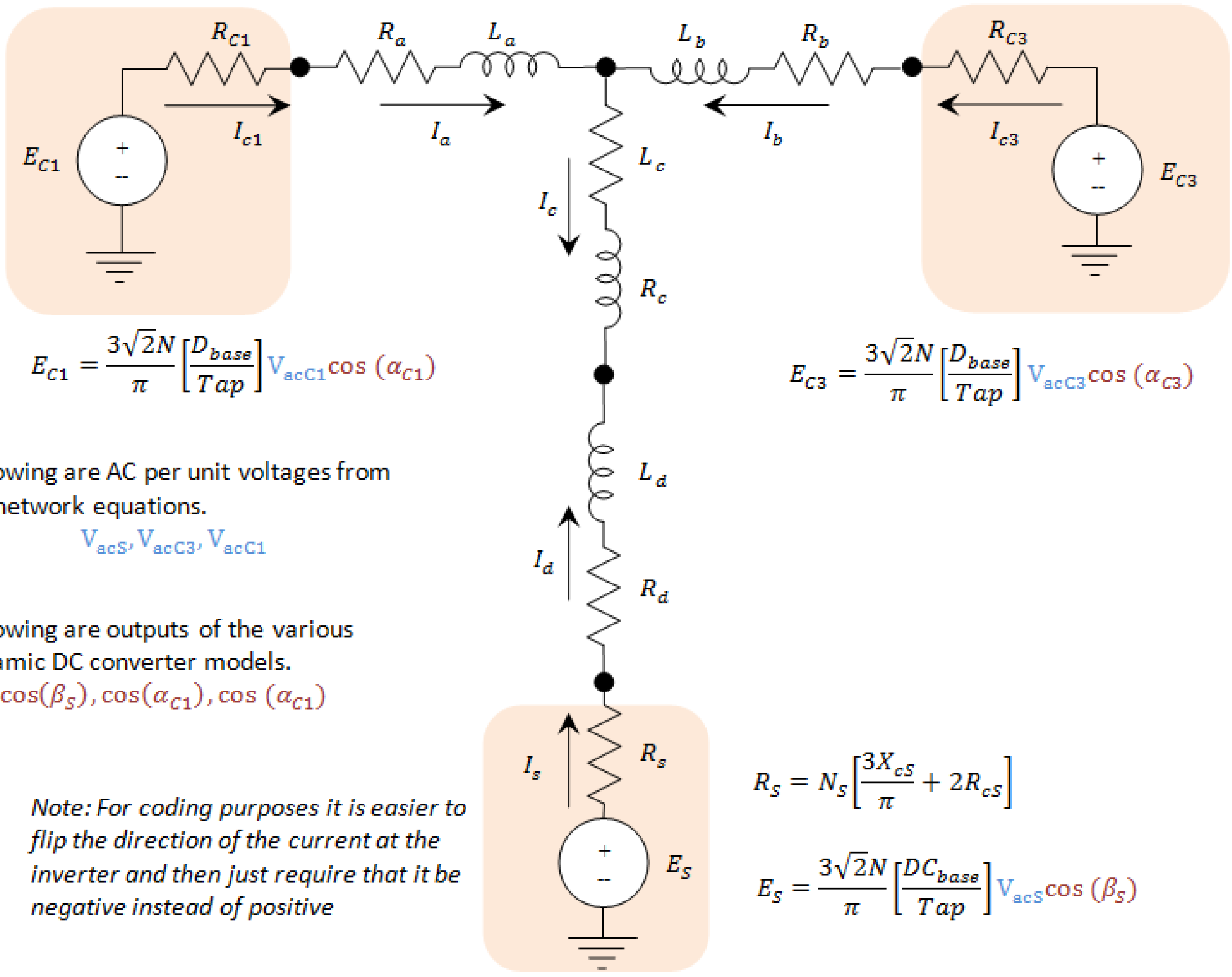
$$E_{inv} = \frac{3\sqrt{2}N}{\pi}\left[\frac{D_{base}}{Tap}\right]\mathrm{V_{ar}}\cos(\beta)$$

$$R_{inv} = N\left[\frac{3X_c}{\pi} + 2R_c\right]$$

# DC Network Model

$$R_{C1} = N_{C1}\left[\frac{3X_{cC1}}{\pi} + 2R_{cC1}\right]$$

$$R_{C3} = N_{C3}\left[\frac{3X_{cC3}}{\pi} + 2R_{cC3}\right]$$



$$E_{C1} = \frac{3\sqrt{2}N}{\pi}\left[\frac{D_{base}}{Tap}\right]V_{acC1}\cos\left(\alpha_{C1}\right)$$

$$E_{C3} = \frac{3\sqrt{2}N}{\pi}\left[\frac{D_{base}}{Tap}\right]V_{acC3}\cos\left(\alpha_{C3}\right)$$

Following are AC per unit voltages from the network equations.

$V_{acS}, V_{acC3}, V_{acC1}$

Following are outputs of the various dynamic DC converter models.

$\cos(\beta_S), \cos(\alpha_{C1}), \cos\left(\alpha_{C1}\right)$

*Note: For coding purposes it is easier to flip the direction of the current at the inverter and then just require that it be negative instead of positive*

$$R_S = N_S\left[\frac{3X_{cS}}{\pi} + 2R_{cS}\right]$$

$$E_S = \frac{3\sqrt{2}N}{\pi}\left[\frac{DC_{base}}{Tap}\right]V_{acS}\cos\left(\beta_S\right)$$

# What is different than a Power Flow Solution

- In static power flow solutions, DC converter control is instantaneous
  - Power Flow Firing Angles (Alpha and Gamma) are assumed to move instantaneously
  - PDCI model does not make this assumption, it models the dynamics of the firing angle control
- Power Flow solutions ignore the inductance of the DC transmission line also
  - In power flow DC currents change instantaneously
  - PDCI models inductance, so DC currents become state variables
  - Note: can also be capacitance in the DC transmission lines
    - We are <u>NOT</u> modeling in the PDCI presently.
    - For cable DC lines (underwater for instance), the capacitance may become large enough that modeling will be important.

# Traditional Explicit Numerical Integration Routines

- Always use the initial algebraic variables to back-solve and obtain the initial values of all dynamics states
- PowerWorld's transient stability tool then uses explicit integration (Euler or 2nd order Runga-Kutta)
  1. Use numerical integration (with a time-step) to update dynamic state
  2. Update algebraic variables such as the AC system voltage and angle (by solving network boundary equations)
  3. Go back to 2 and repeat until simulation finished

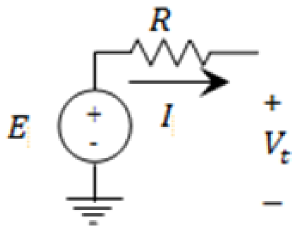
  - Multi-terminal DC simulation will be inserted between steps 1 and 2

# Implementation in Numerical Solution Engine of MTDC

1. Numerical integration → integrate the MTDC_PDCI, CONV_CELILO_E, CONV_CELILO_N, and CONV_SYLMAR model states
    - Updated variables are cos($\alpha$) and cos($\beta$)
2. Take the cos($\alpha$) and cos($\beta$) terms and use them to model a step change in the DC voltages seen by the DC network equations. Use numerical integration to solve for new DC voltages and DC currents
    - Updated variables are DC voltage and DC Currents
3. Solving normal AC network boundary equations except modify DC line equations to assume that cos($\alpha$) and cos($\beta$) , and DC currents are a constant. When network boundary equations are solved update the DC voltages
    - Updated variables are DC voltages
4. Back to Step 1 and repeat

# Step 2: Numerical Integration of DC Network Equations

- DC Converter Model
  - Constant angle so model as a constant voltage source

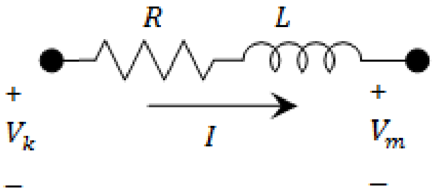


$$RI + V_t = E$$

or

$$I = 0$$

- DC Transmission Line Model
  - Model as RL circuit using trapezoidal rule



$$V_k(n) - V_m(n) - RI(n) = L\frac{dI}{dt}(n)$$

$$V_k(n) - V_m(n) + \left(-R - \frac{2L}{h}\right)I(n) = \left(+R - \frac{2L}{h}\right)I(n-1) - V_k(n-1) + V_m(n-1)$$

- DC Bus Equation
  - Just use Kirchoff's Current Law

$$I_a + I_b + I_c + \cdots = 0$$

# Step 2: Numerical Integration of DC Network Equations

- Solve the previous set of equations using sub-interval integration
- Assume at beginning that no converters are stuck at the zero current limit
- At each sub-interval, if the calculation yields a converter current with the wrong sign then redo the sub-interval replacing the DC converter equation with the equation I=0
- Assume that once a current goes to zero it remains zero during the remaining sub-interval time-steps

# Step 2: Numerical Integration of DC Network Equations: Matrices

- The following is an example matrix setup for the PDCI

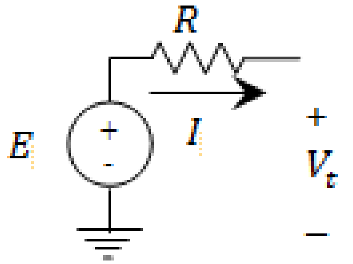
| | Ic1 | Ic3 | Is | Ic | Ia | Ib | Id | v3 | v4 | v7 | v8 | v9 | X | | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Celilo1 | Rc1 | | | | | | | | | 1 | | | $Ic1_{(n)}$ | | Ec1 |
| Celilo3 | | Rc3 | | | | | | | | | 1 | | $Ic3_{(n)}$ | | Ec2 |
| Sylmar1 | | | Rs | | | | | | | | | 1 | $Is_{(n)}$ | | Es |
| LineC | | | | (-Rc-2Lc/h) | | | | 1 | -1 | | | | $Ic_{(n)}$ | | (+Rc-2Lc/h)*$Ic_{(n-1)}$ - $v3_{(n-1)}$ + $v4_{(n-1)}$ |
| LineA | | | | | (-Ra-2La/h) | | | -1 | | 1 | | | $Ia_{(n)}$ | | (+Ra-2La/h)*$Ia_{(n-1)}$ - $v7_{(n-1)}$ + $v3_{(n-1)}$ |
| LineB | | | | | | (-Rb-2Lb/h) | | -1 | | | 1 | | $Ib_{(n)}$ | = | (+Rb-2Lb/h)*$Ib_{(n-1)}$ - $v8_{(n-1)}$ + $v3_{(n-1)}$ |
| LineD | | | | | | | (-Rd-2Ld/h) | | -1 | | | 1 | $Id_{(n)}$ | | (+Rd-2Ld/h)*$Id_{(n-1)}$ - $v9_{(n-1)}$ + $v4_{(n-1)}$ |
| KCL3 | | | | -1 | 1 | 1 | | | | | | | $v3_{(n)}$ | | 0 |
| KCL4 | | | | 1 | | | 1 | | | | | | $v4_{(n)}$ | | 0 |
| KCL7 | 1 | | | | -1 | | | | | | | | $v7_{(n)}$ | | 0 |
| KCL8 | | 1 | | | | -1 | | | | | | | $v8_{(n)}$ | | 0 |
| KCL9 | | | 1 | | | | -1 | | | | | | $v9_{(n)}$ | | 0 |

- If Celilo1 current becomes negative, then replace equation with the following

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Celilo1 | 1 | | | | | | | | | 0 | | | $Ic1_{(n)}$ | 0 |

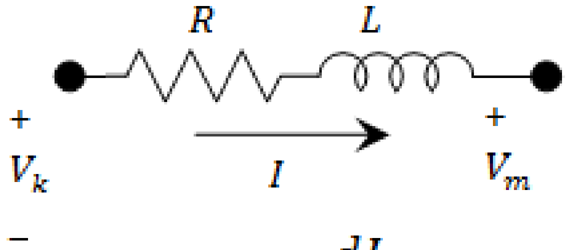
# Step 3: Solution of algebraic change in DC voltages

- DC Converter Model
  - Constant angle and constant current



$$V_t = E - RI$$

- DC Transmission Line Model
  - Model as RL circuit, but current is constant
  - dI/dt is the unknown variable



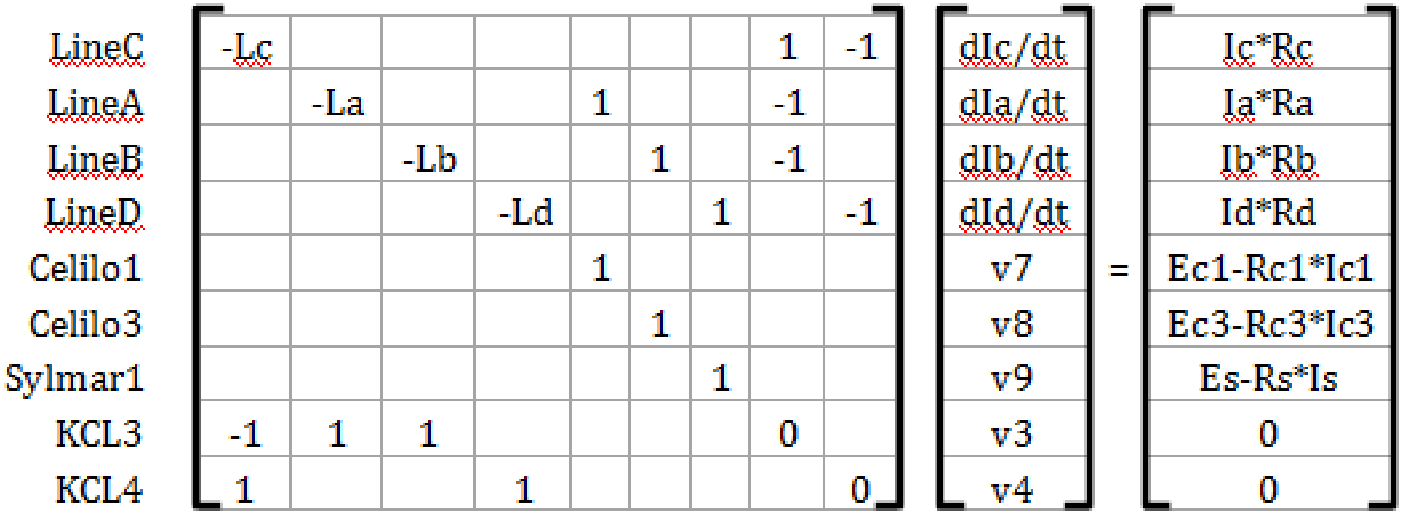
$$V_k - V_m - L\frac{dI}{dt} = RI$$

- DC Bus Equation
  - Just use Kirchoff's Current Law, but for the *derivatives*

$$\frac{dI_a}{dt} + \frac{dI_b}{dt} + \frac{dI_c}{dt} + \cdots = 0$$

# Step 3: Solution of algebraic change in DC voltages: Matrices

- The following is a sample of the matrix setup for the PDCI



$$\begin{array}{r} \text{LineC} \\ \text{LineA} \\ \text{LineB} \\ \text{LineD} \\ \text{Celilo1} \\ \text{Celilo3} \\ \text{Sylmar1} \\ \text{KCL3} \\ \text{KCL4} \end{array} \begin{bmatrix} -Lc & & & & & & & 1 & -1 \\ & -La & & & 1 & & & -1 & \\ & & -Lb & & & 1 & & -1 & \\ & & & -Ld & & & 1 & & -1 \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ -1 & 1 & 1 & & & & & 0 & \\ 1 & & & 1 & & & & & 0 \end{bmatrix} \begin{bmatrix} dIc/dt \\ dIa/dt \\ dIb/dt \\ dId/dt \\ v7 \\ v8 \\ v9 \\ v3 \\ v4 \end{bmatrix} = \begin{bmatrix} Ic*Rc \\ Ia*Ra \\ Ib*Rb \\ Id*Rd \\ Ec1-Rc1*Ic1 \\ Ec3-Rc3*Ic3 \\ Es-Rs*Is \\ 0 \\ 0 \end{bmatrix}$$

# Summary

- PowerWorld now includes the PDCI dynamic model
- This document can serve as a block diagram for what is implemented in the existing *pdci_ns.p* file helping with future updates to the model
- This document makes it more transparent to users what the model actually does
  - i.e you all can double-check our work!