Memorandum

March 4, 2019

TO: WECC REMTF

FROM: Deepak Ramasubramanian, EPRI; dramasubramanian@epri.com
CC: Anish Gaikwad, Evangelos Farantatos, EPRI; Pouyan Pourbeik, PEACE®

SUBJECT: **A METHOD OF IMPLEMENTATION OF CURRENT LIMITS IN REGC_B AND REGC_C**

This document describes a method developed by EPRI of enforcing the current limit at every time step of the simulation through iterations of the network solution. At a high level, the flow chart of this implementation is shown below in Figure 1.

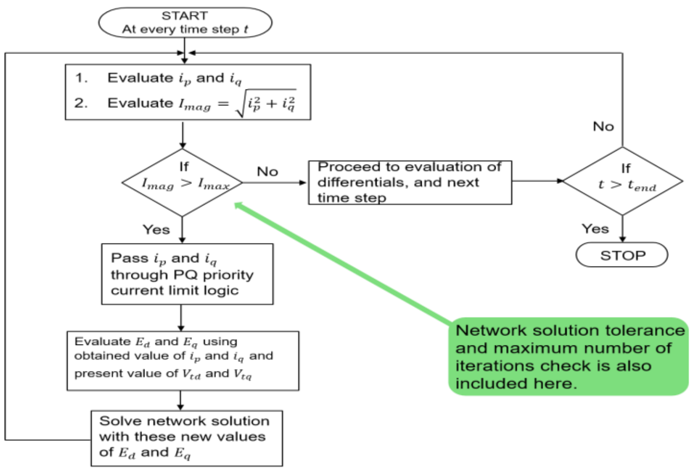

Figure 1: Flowchart for implementation of current limits through iterations of the network solution.

Where, ip and iq are the direct and quadrature components of current injected into the network by the converter source, Ed and Eq are the direct and quadrature axis components of the voltage behind reactance of the converter source, and Vtd and Vtq are the direct and quadrature axis components of the terminal voltage.

The pseudo code of this implementation at every time step is detailed below. Here, it is assumed that the network solution has already been solved once using the current injections obtained at the previous time step of the integration process. Thus, the value of the terminal voltage for the present time step is assumed to be available.

Step 1: Let δ be the angle of orientation of the source synchronous frame of rotation with respect to the network synchronous frame of rotation from the previous time step.

Step 2: Vtq=(-Vr*sin(δ))+(Vi*cos(δ))
Vtd=(Vr*cos(δ))+(Vi*sin(δ))

Here, Vr and Vi are the real and imaginary components of the terminal voltage on the network frame of reference.

Step 3: Solve for ip and iq in the equation (Ed+jEq) – (ip+jiq)*(re+jXe) = Vtd + jVtq

Step 4: Evaluate value of the magnitude of current Imag = sqrt((ip*ip) + (iq*iq))

Step 5: If Imag > Imax

Depending on the orientation of the dq axis, iq could be a negative value while denoting supply of reactive current. Care must be taken to take this into consideration while applying the limits.

Step 5a: If REEC is in P priority, then Ipmax = 0.9*Imax

Step 5a-i: if (ip > Ipmax) then ip = Ipmax;
if (ip < ~~0.0~~-Ipmax) then ip = -Ipmax~~0.0~~

Step 5a-ii: Iqmax = sqrt((Imax*Imax) – (ip*ip))
If (iq > Iqmax) then iq = Iqmax
If (iq < -(Iqmax*0.9)) then iq = -Iqmax*0.9

Step 5b: If REEC is in Q priority, then Iqmax = 0.9*Imax

Step 5b-i: If (iq > Iqmax) then iq = Iqmax
If (iq < -(Iqmax*0.9)) then iq = -Iqmax*0.9

Step 5b-ii: Ipmax = sqrt((Imax*Imax) – (iq*iq))
if (ip > Ipmax) then ip = Ipmax;
if (ip < ~~0.0~~-Ipmax) then ip = -Ipmax~~0.0~~

Step 6: Solve for the value of voltage behind reactance
Ed = Vtd + (ip*re) – (iq*Xe); Eq = Vtq + (iq*re) + (ip*Xe)

Step 7: Ed, Eq, and δ are used to define the current injection to solve the network solution and then return to Step 1. If the Imag <= Imax, then it is expected that the network solution will converge within a few iterations.

Commented [P1]: We probably need to add a new parameter to REGC_B called *pqflag* instead of relying on pricking it up from the downstream model.

Commented [RD2R1]: Possibly in addition to adding the parameter, should a data check also be added to ensure that the value of flag is same in both models? I understand that this not an issue of the model and is should be up to the user to ensure that parameterization is consistent.

Commented [P3]: We need to allow for energy storage; may have to add the Ke factor here from reec_d.

Commented [RD4R3]: True. Again, would there be a need for a data check to ensure consistency of the value of the parameter used?

A note regarding the variables Ed and Eq. These are not the same as the variables Ed and Eq in the REGC_B block diagram. In the pseudo code, Ed and Eq refer to the variables used to define the components of a voltage source in the user defined model in GE-PSLF™.

Thus, if Step 3 of the pseudo code is being run for the first time in a particular time step, then Ed and Eq would hold the same value as state variables s4 and s3 respectively. However, in subsequent iterations, the variables hold values defined in Step 7 of the pseudo code.

Note regarding the multiplier factor value of 0.9. In the EPRI implementation, this value has been hardcoded for two reasons:

1. It is assumed that the conditions for injection of current should always have a non-zero value for both ip and iq. Thus, Ipmax and Iqmax were assumed to always take a maximum value less than Imax. Appropriately, the VDL blocks of REEC_A were also parameterized. However, such an implementation is a specific implementation and not the norm. Thus, for the official software versions of REGC_B and REGC_C, it is recommended to remove this factor, essentially allowing for Ipmax = Imax when in P priority, and Iqmax = Imax when in Q priority. However, if it is deemed necessary to allow for a condition wherein ip and iq should always be non-zero, then, the value of 0.9 can be implemented as a user input rather than being hardcoded in the model.
2. It is also assumed that when the model works in the 2^{nd} and 3^{rd} quadrant of the P-Q plane (essentially in a leading power mode when using generator convention), there would be an added restriction on the maximum absolute value of reactive current. Thus, the minimum limit of iq is set as 90% of Iqmax. However, this is again a specific implementation and not the norm. Thus, again, for the official software versions of REGC_B and REGC_C, this factor can be removed.