

NERC Lessons Learned:

“State Estimator Outages Requiring Tuning/Calibrating EMS Settings”
and
“Breaker Failure Due to Trip Coil Polarity”

Donnie Bielak
Reliability Engineering

- Title
 - State Estimator Outages Requiring Tuning/Calibrating EMS Settings
- Source of Lesson Learned
 - Reliability *First*
- Date Published
 - March 6, 2018

- Several entities experienced short SE outages due to the software solution not converging
- Troubleshooting techniques attempted to locate and remove erroneous data point inputs into EMS to help SE solve
 - This method failed and the respective EMS vendors were contacted

- In three events, operators received an EMS alarm during off hours that SE failed to converge
 - Appropriate entities assisted with monitoring as operators began troubleshooting
 - Additional support, diagnostics, and reboots were unsuccessful, which resulted in soliciting vendor support

- The vendors suggested setting changes, which caused SE to converge
 - The change was tested before implementing, in order to ensure there were no negative impacts to SE solution quality
- Event Analysis Process discovered the settings were part of the original SE installation and programmed for each specific entity's needs
 - These settings needed to be tuned when the topology of their system changes

- All three cases required a change to SE settings or parameters
 - Changes were tested before implementing to ensure a quality SE solution
 - The time to find a solution resulted in diminished situational awareness
- The entities relied on their RC for SE, including Real Time Contingency Analysis
- Any topology changes should have been halted while SE failed, such as switching solutions

- While relying on RC for assistance, it is important to **keep SE up** and providing base case states for downstream applications
- When significant changes are made to the EMS system, it may be necessary to **review SE parameters**
- In a test environment, input erroneous data on sub-transmission facilities or remove key points to **see what can cause issues with SE**

- Work with vendors to **develop a list of key settings** that need to be adjusted
- Develop key metrics to **identify when tuning would be beneficial** (such as number of iterations to solve, mismatch size, etc.)
- Develop in-house expertise to **reduce restoration time**

- Title
 - Breaker Failure due to Trip Coil Polarity
- Source of Lesson Learned
 - Midwest Reliability Organization
- Date Published
 - March 6, 2018

- A temporary phase-phase-ground fault occurred on a 115 kV line
 - The breaker at one end of the line was slow to operate
 - Breaker failure protection operated causing two 115 kV breakers of a three-breaker switching station to open
 - The failed breaker's two trip coils were damaged when they were energized to trip the breaker
- The switching station de-energized

- Each trip coil had opposing polarity
 - When energized simultaneously, cancellation of the magnetic flux occurred due to the opposing polarity, causing the coil armature to remain immobile
- Relay data showed that the coils were energized by separate primary and secondary relay trip contacts in less than 3 milliseconds

- The coils were replaced and tested to verify proper operation
- Simultaneous energization tests should be conducted on all breakers with multiple trip coils that share a common magnetic path
 - This should occur for the tripping time of the breaker
- One test method is a set up with at least two parallel contacts that could be used to energize the coils from their separate DC circuits

- If the coils are installed properly, the series 52a contacts should interrupt the coil currents
- If installed improperly, then the contacts will be called on to interrupt the currents
- Beware of inductive kick-back voltage while interrupting DC current flowing through the coils

- Common Assembly trip coils installed by a vendor **need to be verified**
 - Field wiring should also be verified to ensure coils operate properly
- **Test breakers** for proper tripping when coils are energized simultaneously