

Document Purpose / Intent

An extended term case may require the insertion of new, or additional, modeled generation resources. There is currently no known method in PST to increase inertial generation during a simulation. Additionally, there seemed to be minor confusion/forgetfulness as to how PST currently handles tripping a generator. A closer look into what happens when machines are tripped off may provide some insight into a method to ‘un-trip’, or insert, generators.

How generators are currently tripped in PST

During simulation initialization, `g.mac.mac_trip_flags` is initialized as a row vector of zeros that correspond to the `mac_con` array, and `g.mac.mac_trip_states` variable is set to zero. To trip a generator, a `mac_trip_flag` is set to 1 via the user generated `mac_trip_logic` code. The `mac_trip_logic` is executed in the `initStep` function which alters `g.mac.mac_trip_flags` to account for any programmed trip. Specifically, a 0 in the `g.mac.mac_trip_flags` row vector is changed to a 1 to signify a generator has tripped.

The `g.mac.mac_trip_flags` vector is summed in the `networkSolution` (`networkSolutionVTS`). If the resulting sum is larger than 0.5, the line number connected to the generator in the `g.line.line_sim` is found and the reactance is set to infinity ($1e7$). The reduced y matrices are then recalculated and used to solve the network solution via an `i_simu` call.

If derivatives of the tripped machine are not set to zero (as the current VTS method does), the generator’s speed increases, mechanical power output eventually drops to zero/near-zero, and the attached exciter `Efd` appears to approach 1.

Performance Note If a machine is tripped, altered reduced Y matrices are generated every simulation step. This repeated action could be reduced via use of globals.

- create new globals handling the Y matrices selected during the network solution.
- create an `old_mac_trip_flags`
- compare `old_mac_trip_flags` to `mac_trip_flags`
- if there has been a new trip, update associated Y matrix globals
- if there hasn’t been a new trip, but a machine has tripped, use the stored Y matrix

This is merely a calculation reduction and not required, but may result in a noticeable speed up.

Initial 'Un-trip' Thoughts

If the trip flag is changed from 1 to zero - the reactance would return to normal, essentially reconnecting the generator to the system. However, the calculated states and derivatives would be no longer be synchronized and probably cause all sorts of issues/transients.

The machine model may be re-initialized via the use of the 0 flag and setting i to the internal number of the generator to re-initialize. This would require knowing what type of machine model is tripped so the proper function call is used. The effects of such action is untested - a simple test case should be created and attempted.

Basically: create small system with multiple generators, trip a generator, allow system to regain steady state, reconnect generator by setting machine trip flag back to zero. Observe results and attempt to resolve issues.

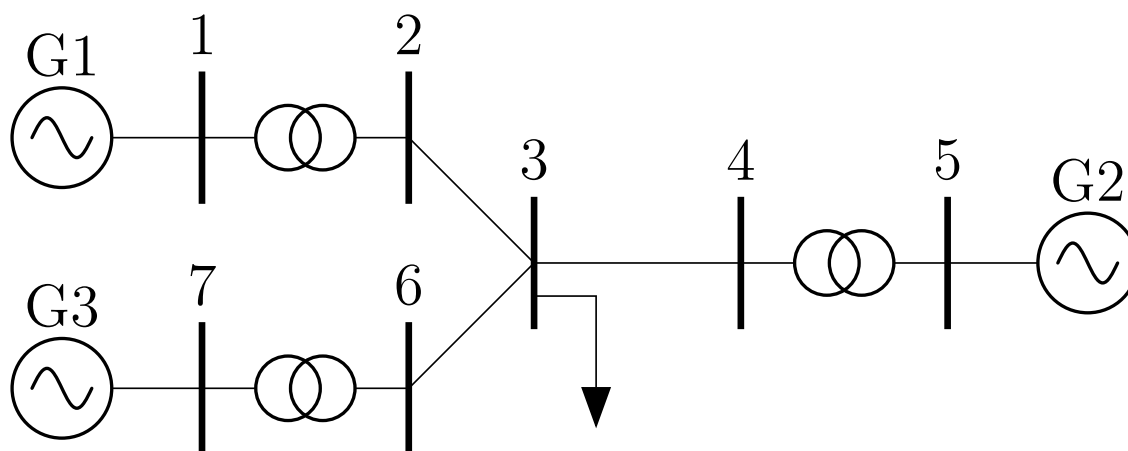
'Un-trip' Method 1 - Alter Aggregate Generator

This method includes altering a machine's base during simulation to imitate the addition/reduction of an aggregate generator. It would require recalculation of all `mac_pot` values used in model calculations. Additionally, the monitoring of what is actually occurring may be obfuscated by the fact that any gains in capacity would be added to the existing aggregate value. This method was not predicted as being viable due to foreseen complexity/unavoidable transients.

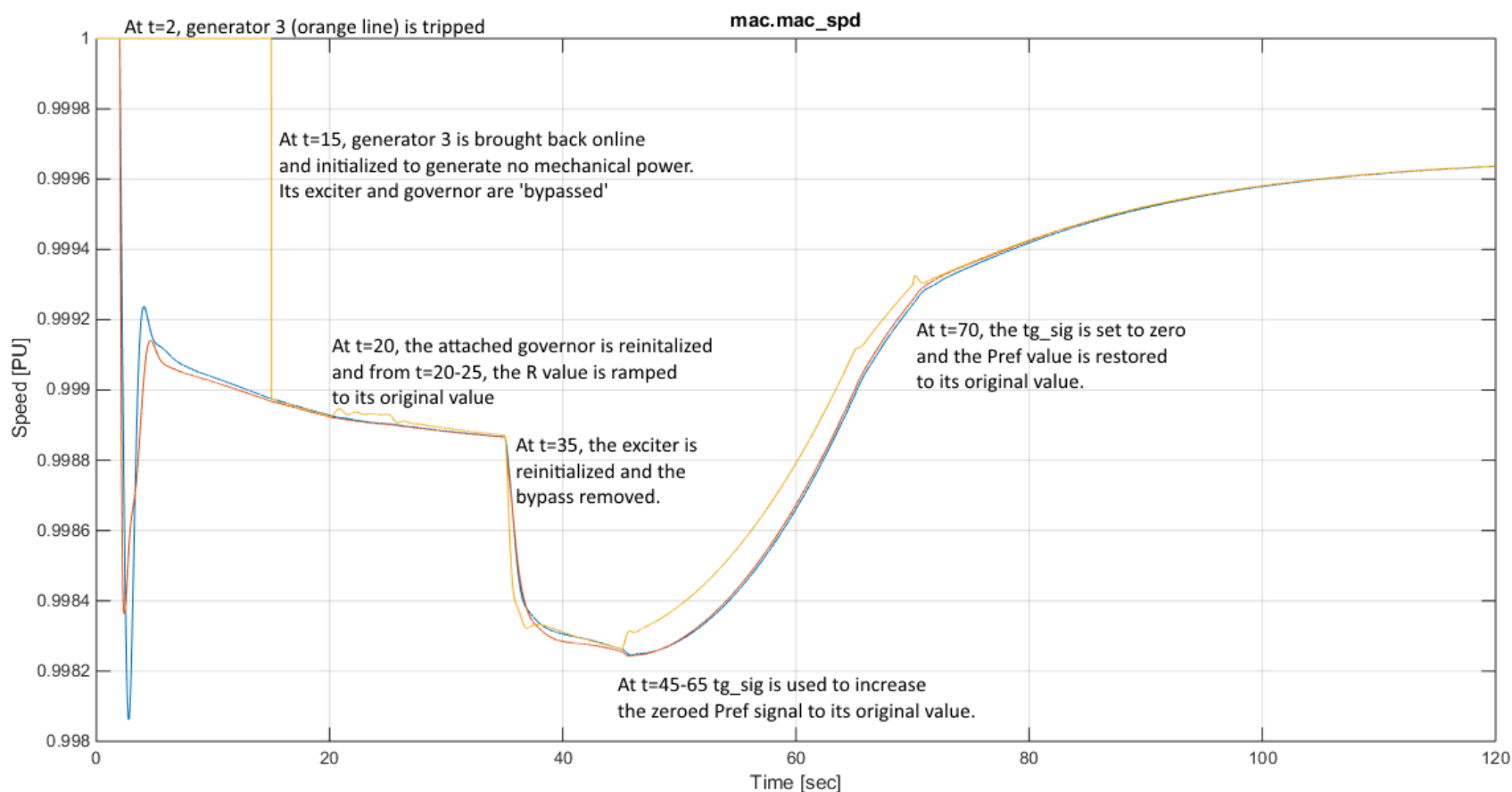
'Un-trip' Method 2 - Stand alone Generator

Account for all generators that may, or may not, be added into the system during creation of the system data file (case). Handle model re-initializations and bypasses as required to mimic actual procedure of connecting a generator to a synchronous system.

Test System A simple 3 machine system was created where all machines have governors, exciters, and pss. Parameters are essentially the same except for MVA base for each generator which was 500, 200, and 100 for generator 1, 2, and 3, respectively



Initial Experimental Results Using the previous notes as a guide, a 3 machine scenario was created where generator 3 will trip off, then attempt to be reconnected with minimal transient behavior. Initial results appear promising. All transient behavior may not be eliminated, but could possibly be reduced by resetting set point values and then ramping to the new desired setting (most notably the exciter reference voltage).



The exciter ramping up field voltage (and generating vars) caused the speed dip at $t=35$. The exciter is required to be connected before mechanical power is ramped, else voltage collapse occurs and the machine 'runs away'. The machine speeds do not return to 1 as the mechanical powers do not fully restore. The P_{ref} setting of the tripped governor appears to not end at the 0.5 it is set to, or the Pref value is scaled somewhere in the model that has been missed.

Various Other Result Plots

