Lab 2 Report

Authors: Forrest Iandola, Huy Vo

1. Our BPCU is functional block that maps the generator status (healthy or unhealthy) to contactor status (open or close). It is implemented in Simulink by using Stateflow. We modeled the system as follows. There is one parallel state for each switch B1-12. Each of which can be in either Off or On state. There is an additional parallel state, S, that controls what operation the switches can do. In the Listen state, the controller detects changes in generator status. On a change, it moves to the Off state allowing any of the B1-12 to go from On to Off. Then it moves to the On state allowing any of the B1-12 to go from Off to On.

This extra S state satisfies a couple of the requirements. It satisfies G1 because it starts out in the Off state, preventing any of the switches to immediately switch to the On state. It also ensures G3 and G4. This is because the while S is in the Off state, switches are only allowed to be opened. Then in the On state, switches are only allowed to be closed. This, along with the logic in B1-12, ensures that it is never the case that multiple AC sources are connected in parallel and that both APUs are never inserted into the system.

G2 is trivial. Examine our logic.

The controller switches at a rate of 25ms. In the worst case scenario, a generator fails as we leave S’s On state. Then 25ms later we go to the Off state because the system has detected a change. Then 25ms later we go to the On state. So allowing for any wiring delay, in the worst case, buses can only remain unpowered for 50ms to 75ms. This value can be adjusted by changing the rate of the state machine.

G6 is trivial. See the bus logic.

G7 and G8 are handled by our bus logic.

1. Reading through the lab document and in designing the BPCU, w came up with a few interesting scenarios. We tested optimal condition (all generators healthy), one failure (either left generator or right generator), and worst case (both main generators off, both auxiliaries working and only one auxiliary working).
2. Simulink was not as prohibitive to our lab as Ptolemy II was. Wiring and text interface (biggest source of trouble with Ptolemy) were very intuitive and well designed. We wanted to use Statemate in order to better model our system but there was a very large learning curve. We were eventually able to figure it out after reading through quite a few tutorials.