1 page (what is findings, what did you do, why did you do this, who cares)

**What are the findings:**

The findings are that Gridlab-d’s IEEE 1547 checks on its inverter objects do the job of turning off when they perceive and overvoltage. However, the big problem is that when an inverter object runs these checks, they read in a different voltage when it turns back on than if it had just stayed on during an overvoltage condition. In theory, this should not be happening, because when we put back solar into the system, the voltage on the inverter node should be the same as the voltage on that node if the same solar power had been there for an extended period of time. However, this is not the case. The inverter when running IEEE 1547 checks turns itself off at times it should not turn off, namely in times where without the checks the inverter’s voltage should be below overvoltage. It is also important to note that if I put deltamode on a model, even if I do not put in the checks, it behaves differently than if I have not.

Another finding is that the weather data, the tmy2 data is in terms of hours, so you cannot expect a change in solar power in less than an hour on a given system.

A few findings about the inverter object: all it does is check for the inverter’s voltage. At every timestep, whether in event or deltamode, it converts the voltage to p.u. to see if the inverter hits any over/undervoltage conditions. It sees how long it’s been in an over/undervoltage condition, and if the inverter has been in it for too long, the checks will turn the inverter off. If the inverter is off, the object will use its 1547 checks to see how long it has been off due to an over/undervoltage condition; when a set time is up, the inverter will turn itself back on. It’s important to note that only the inverter turning itself back on, or a huge power surge will trigger Gridlab-d’s deltamode, not an over/undervoltage condition.

One more finding from the Gridlab-d sourceforge forum is that if we set up our load objects using constant\_current\_A, B or C, we should expect different behaviors from this object whether it is in event or deltamode. <https://sourceforge.net/p/gridlab-d/discussion/842562/thread/6c3b393b/>. This was to explain our voltage differences that I talked about above.

**What I did:**

I modified the IEEE13 node feeder model from gridlab-d to handle solar power on m680 with an inverter. I made IEEE13\_pv4.glm, IEEE13\_pv5.glm, IEEE13\_pv6.glm, IEEE13\_pv7.glm, which represent IEEE13 with deltamode and IEEE1547 checks activated, IEEE13 with nothing activated, IEEE13 with just deltamode activated, and IEEE13 with just the checks and deltamode for the inverter only activated. Each meter objects in these models had two recorders each: one for voltage, one for current. The outputs of the recorders are read into pandas\_experiment.ipynb. I made graphs with this ipython notebook, and posted them not only in a in-group presentation, but also on Gridlab-d’s sourceforge forums, where I have been corresponding with Frank Tuffner.

Here, Frank has been giving me several suggestions. The first was to switch over from the main stable gridlab-d build to a branch called feature/730, which solved an error I was having with the inverter turning off but still feeding in current. It also allowed me to run IEEE1547 checks without deltamode. The second thing I did from Frank was to change all my loads into ZIP loads, as per the suggestion above. Unfortunately, they did not solve my problems. Look to my post on sourceforge for more information.

I also added print statements inside of the inverter.cpp source code, so I could see what was going on. I used the function gl\_output, so I could feed the print statements into an output file for further observation. This is how I was able to find out that the inverter object was feeding current into the IEEE13 feeder when it wasn’t supposed to.

**Why did I do it:**

I tried looking into Gridlab-d’s IEEE1547 checks because I saw that Gridlab-d is a monolith of a code, and without a proper understanding of it, any changes I make could come out with unintended consequences. People, who knew the system made the IEEE1547 checks. So I assumed this was the right way. I made four files, one control, one with just deltamode, one with deltamode and IEEE1547 checks, and one with just IEEE1547 checks because I wanted to see the effect of IEEE1547 on the system, if deltamode itself makes a difference on the system, and if the combination of IEEE1547 and deltamode made the system more correct or not, since IEEE1547 works in less-than-second timeframes, and the only way you can get to that is through deltamode.

I added print statements in the system so I could figure out how gridlab-d worked, when it triggered deltamode, and if the inverter behavior, when it goes into deltamode and when it turns on and off, made sense with the data I was seeing on csv files and in my ipython notebook. I made the ipython notebook so I can use the python pandas library to easily read csv output files and plot them. It made data analysis a lot easier.

I took the advice from Frank Tuffner, because he is one of the main contributors to Gridlab-d according to Sourceforge. I especially took his advice on turning all loads into ZIP loads because I thought that would solve the voltage difference problem once and for all, and thus make gridlab-d useful for battery analysis on a high pv-penertration grid, and other reasons that I have stated in my in-group powerpoint presentation.

Learnings from the project:

1. Take your time to understand what the code is
2. observations:  
   The no-deltamode guy decreases his voltage after he hits the sun  
   Then stays there till the next hour  
   The deltamode guy stays where he is at for 5 minutes, then h goes down, then back up, then down, then back up.   
   timestep does not make much of a difference  
     
   Deltamode without inverter checks and deltamode with checks takes different times to iterate.   
     
   just inverter checks does not seem to iterate, but brings high voltage to a check first. Not sure, or maybe what we are seeing is a voltage at a tiny timestep after it converges   
     
   After it shuts off in delta  
     
   enable deltamode vs running deltamode (can you always run deltamode)  
     
     
     
   Take aways general:   
   one year or two just to document this if another undergrad wants to come on to the lab. (find a way to make this more fun) (write unit tests or pandas) (write tools to understand)   
   document this (time frame, one or two years)   
     
   reformating code: (maybe just modularize functions) (make this more user-friendly) (if you see a large chunk of code, then make it a seperate helper function)  
     
   Take it very slow.   
     
   Takeaway:  
   pass on  
   One page, guide a student to continue,

Notes: the vdata\_no1547 and the vdata\_new (deltamode with 1547) are exactly the same, because solar does not go above 1.10 there

Ok, long story short, the recommendation online is still not good.

Who would be the idea person who would work on this?

Someone who probably has more power grid experience rather than coding experience because they could interpret the data: but at least someone who has dealt with c before, or else they would get lost on data types and other sorts of things.