**Potential Future Research Directions for D-MASE**

Format:

* Tentative title
* Problem statement: one paragraph defining the problem we are trying to solve and explaining the motivation for why we would want to solve that problem
* Work to be done: a list describing what work we would have to do to get there and classify the effort needed as low, medium, or high.
* Available literature: if possible, explain how this fits into the overall literature/research space.

1. **“An Investigation of Automatic Partitioning of Power Systems for Decentralized Applications”**

Problem statement:

Usually the boundaries for power system areas are defined by ownership, and power system algorithms have been developed with that in mind. However, now with the industry paradigm shifting away from centralized operations towards a decentralized model, it makes sense to no longer identify system boundaries only by ownership. For example, it may be beneficial from a computational perspective to divide up a system such that there is minimal communication needed across the boundaries between different decentralized entities. An automatic method to accomplish this would aid in the investigation of how different partition (or prosumer) numbers/boundaries impact the performance of decentralized power system applications. This type of partitioning could be done offline and rerun occasionally when there are significant topology changes.

Work to be done:

* Identify how we would want to break up a system. For example, keep partitions roughly same size? Minimize number of cut lines/communication lines?
* Determine what partitioning method(s) to use (spectral, Kernighan-Lin, etc. possibly talk to Jack Poulson).
* Write standalone code that takes as input system topology (maybe other parameters too) and outputs partitions based on how we want to break up the system.

Estimated level of effort:

Low-medium

Available literature:

Have seen 1-2 recent papers on this topic

1. **“Parallel Distributed Multi-area State Estimation using ADMM”**

Problem statement:

Distributed multi-area state estimation has been of great interest to the power industry due to the need to accurately estimate inter-area transactions (double check this). Many previously proposed approaches required a central entity to coordinate and facilitate this process. ADMM is an inherently decentralized algorithm that can converge to the central solution assuming there is sufficient communication between each control area (need to reexamine how my implementation is currently handling the communication). Previous work on ADMM-based D-MASE described the formulation of the problem along with how to handle error rejection in a decentralized way. However, it did not investigate the impact of the number of areas on overall performance.

Work to be done:

* Would be useful to have an automatic partitioning tool so that I could easily compare 2 vs 10 vs 100 vs 100 partitions without having to manually partition each test case.
* Would need pretty large systems to get noticeable speed improvements
* Option 1: truly parallelize using C++ and MPI
  + Set up Visual Studio and math libraries
  + Port centralized SE code from MATLAB to C++ (code and debug)
  + Port ADMM SE code from MATLAB to C++ (code and debug)
  + Refamiliarize myself with MPI and how to submit jobs to the cluster
  + Implement and debug the MPI code

Result: get truly accurate timings

Estimated level of effort:

High

* Option 2: fake parallelization in MATLAB
  + Find a way to estimate communication time between partitions
  + Time in MATLAB how long it takes each partition to do its own SE and take the max of those
  + Compare behavior of having different numbers of partitions
  + If automatic partitioner works, could try partitioning different ways and see if that has an estimated impact

Result: get an estimate of timings (and potentially evaluate if it’s worth truly parallelizing)

Estimated level of effort:

Low

Available literature:

There are papers on parallel state estimation, but I don’t think there are any specifically on ADMM-based D-MASE.

1. **“Reliable Distributed Multi-area State Estimation in the Presence of Communication Issues”**

Problem Statement:

The key to achieving an accurate distributed multi-area state estimation solution is to have sufficient communication between the control areas. However, there may be times when the communication channel between areas becomes unreliable. For example, communication errors could be introduced either unintentionally or maliciously through false data injection attacks. The problem we would want to solve is how to still achieve an accurate solution if there is a communication failure.

Work to be done:

* Look at the existing literature to see what’s been done in this area.
* Develop an algorithm to detect communication issues. Perhaps using some kind of statistical analysis?
* Develop an algorithm to fix the issues. For example, like extrapolating the previous iteration solution?

Estimated level of effort:

Medium

Available literature:

TBD

What Masoud is doing for his decentralized frequency control using ADMM problem:

* First proving controllability of this decentralized problem (for me, it would be observability) – paper Magnus and him submitted to CDC conference (no decision yet) – paper 1 for them
* Then actually implementing this in parallel MATLAB, looking at number of iterations (rather than the actual time), then assigning a time for how long it takes each iteration and a fixed time for how long the communication could potentially take, then using that as an indication of how long the overall problem would take to solve – paper 2 for them

1. **“Observability Constraints in Distributed Multi-Area State Estimation”**

Network observability analysis determines if a unique estimate can be found for the system state. In traditional centralized state estimation, global observability is required across the whole area in order for a unique solution to be found. In a decentralized architecture, now local observability is required for each partition in order for them to uniquely solve their own smaller state estimation problem. For a multi-area power system, there is no innate difference between viewing multiple control areas as one mega control area versus each as its own separate control area. However, the first will converge to a unique solution given global observability whereas the second one may not given global observability but not local observability (double check this). It would be good to solve one of the following problems.

Option 1:

Prove that only global but not local observability is required in order to have the ADMM based

Work to be done:

* Read up on observability analysis.
* ???

Estimated level of effort:

Medium

Available literature:

Several papers from Le Xie on relaxing local observability constraints

1. **“Integrating Transmission and Distribution State Estimators”**

Problem statement:

Traditionally the transmission grid and the distribution grid have been operated independently of one another. However, now with the distribution grid experiencing ever-higher penetrations of DG (such as roof-top solar), we are seeing effects at the transmission level. With rising industry interest in the development of distribution state estimators, it makes sense to consider how to integrate the outputs of distribution state estimators into transmission-level state estimators.

Work to be done:

* Read up on distribution state estimators. Lots of problems, including mostly current measurements.
* Find out what the outputs of distribution state estimators are.
* Implement a distribution state estimator.
* How to calculate distributed SE if partitions are at completely different scales?

Estimated level of effort:

High+

Available literature:

???

1. **“Extending static distributed multi-area state estimation to dynamic power system behavior”**

Problem statement:

Under normal operations, power system behavior is quasi-static, shifting slowly throughout the day. However, when power system events occur, the quasi-static assumption no longer holds true.

Work to be done:

* Read up on dynamic state estimators.

Estimated level of effort:

High

Available literature:

???

1. Conventional improvements to distributed state estimators: error checking, topology estimation, parametric estimation
2. ADMM methods are increasingly being used for power system applications. ADMM requires that objective functions be convex in order to guarantee convergence. Many power system objective functions may be nonconvex and/or nonlinear. In the case of state estimation, the polar formulation is not convex although the rectangular one is. Why doesn’t the polar formulation converge (double check this)? Is there some way to make it converge?
3. Estimate the state of specific types of equipment?