Typical ADMM consensus problem:

Want to solve in a decentralized way.

Simplified formulation since :

Centralized WLS state estimation (non-convex function since h(x) is nonlinear, Gauss-Newton doesn’t guarantee convergence – not even local convergence like in Newton’s method):

Decentralized WLS state estimation using ADMM for partition *i* (non-convex for which ADMM does not guarantee convergence):

Recall:

Back to the decentralized WLS formulation:

Using the Gauss-Newton method, expand the nonlinear function into its Taylor series around , neglecting the higher-order terms.

Neglecting the second-derivative terms,

Polar AC power flow:

Rectangular AC power flow:

?

Derive rectangular line flow:

In order for ADMM to converge, all of the areas should be referenced to one slack bus. Otherwise the averaging process won’t converge.

Assume there exists two arbitrary areas A and B. Each area has its own slack bus, slack a and slack b. Assume slack a is the global reference bus. How do you reference area B’s slack bus to area A’s slack bus?

1. Run SE for Area A, and run SE for Area B.
2. Look at the common buses (overlapped boundary buses) between Areas A and B. Take th\_a – th\_b for those overlapped buses, and take the average (in general, there should only be a few tielines).
3. Add that average to all of the angles for Area B. That’s very straightforward in polar form. But how do you do that in rectangular form?

,

,

The voltage for any bus in Area A is . The voltage for any bus in Area B is

To change the reference bus for all of the buses in Area B, the new voltage would be

The previous way has some issues. For example, what if an area has no overlap with the global slack area? In that case, you need to reference each area to its neighbor until you reach the global slack area. That process requires a central coordinator. What if instead you let a pair of two areas adjust to each other, then adjust their neighbors until they all arrive at an agreement? Here’s how that would work.

,

,

The voltage for any bus in Area A is . The voltage for any bus in Area B is Let

To change the reference bus for all of the buses to halfway between areas A and area B, the new voltages would be and

Current D-MASE process:

1. Area 1-4 does its own small SE.
2. They exchange their boundary estimates, take the average, and use that average to calculate their next step.

The problem is that they can never converge, because each area has its own slack bus.

New D-MASE process:

1. Area 1-4 does its own small SE.
2. Assuming Area 1 has the global reference (slack) bus, adjust Areas 2-4 so that they are all referenced to Area 1’s slack.
3. Then exchange boundary estimates, take the average, and use that average to calculate their next step.
4. Do you need to adjust it back to each area’s slack?

Motivations for D-MASE:

* Control areas/ISOs want to jointly estimate their combined state so that they can accurately determine the interchange across the areas.
* One issue is that they have different slack (reference) buses. How do they calculate that power flow? Of course they have meters, but meters can have issues.
* Another motivation is if in the future, distribution areas have their own state estimators, and then you want to jointly solve the problem of state estimation across a ton of distribution areas.

Some future ideas:

* Xiaochen said that there are people doing partitioning for OPF. Look at those papers, and see what they did. Also how they motivated their problem.
* Jouni asked if I was sure that rectangular SE problem is definitely convex. I don’t remember seeing a rigorous proof on this. It’s the fundamental criteria to proving that the ADMM algorithm will converge. Look into this.

Questions to answer for D-MASE:

* Is it true that ADMM only converges for the rectangular state estimation formulation and not the polar one?
  + Show that the rectangular SE formulation is convex using rigorous math
* Do you really need sensors if you have a distributed SE problem than if you have a central SE problem?
* In order to have all areas converge in a decentralized way:
  + Solution 1: install a PMU in each partition
    - Becomes optimal PMU placement problem
    - But how would you incorporate those PMU measurements? The issue is the error covariance
  + Solution 2: have each neighboring area add a correction factor (slow but effective)