ECE 6320 Fall 2019: Example Projects

Dan Molzahn

August 2019

This document provides some examples of projects for ECE 6320. You are also welcome to propose your own project on any topic related to power systems operations.

All teams must meet with the instructor near the start of the semester in order to discuss their project, and two brief status updates will be due during the semester.

Evaluations: Your projects will be evaluated based on two components: 1) A written report in two-column IEEE format.[1] The written project report is due at the start of the final exam period. 2) An oral presentation to the class during the final exam period.

Project Deadlines:

- September 2019 (Date TBA): Meet with the instructor to discuss the project.
- October 18, 2019: Send a brief (less than one page) update to the instructor regarding the status of your project.
- November 15, 2019: Send a second brief (less than one page) update to the professor regarding the status of your project.
- December 11, 2019 (11:20 am – 2:10 pm): Written project reports are due at the start of the final exam period. Oral project presentations will occur during the final exam period. There is no final exam for this course.

[1] See the IEEE Transactions Articles Template at the following link for further information: `https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/authoring-tools-and-templates/ieee-article-templates/templates-for-transactions/`

1 Implementation and Analysis of Distributed Slack Variants of Power Flow Algorithms

The power flow problem discussed in class uses a single slack bus with specified voltage magnitude and phase angle ($V_{slack} = 1$ and $\theta_{slack} = 0^\circ$) and unspecified active and reactive power injections (P_{slack} and Q_{slack}). A more general power flow formulation called a *distributed slack* can better model the actual behavior of typical power systems.

In a distributed slack formulation, all buses are classified as either PV or PQ buses. While the PQ buses are treated the same as in the traditional power flow formulation (specified active and reactive power injections), the PV buses have different behavior. Specifically, the distributed slack formulation of the power flow problem proportionally allocates the role of balancing the active power injections among all of the generators rather than a single slack bus. This is accomplished by modeling the power injections at each PV bus i as:

$$P_i = P_i^{\bullet} + \alpha_i \, \delta_P,$$

where α is a specified "participation factor" vector with non-negative entries that sum to one, $P_i^{\bullet}$ is the specified nominal active power injection at bus i, and δ_P is a new variable that is shared among all of the active power injection equations. Another distributed slack formulation explicitly models the change in losses in order to compute the active power injections P_i rather than adding the variable δ_p.

In the distributed slack formulation, note that the phase angle reference $\theta_i = 0^\circ$ is enforced at a single "reference" bus i and all generator buses have specified voltage magnitudes. Also note that the distributed slack formulation simplifies to the slack bus formulation discussed in class by choosing $\alpha_i = 1$ for a single PV bus i and $\alpha_k = 0$ for all other PV buses k.

Tasks for this project include:

- Performing a literature review on formulations of distributed slack power flow problems.
- Implementing various power flow algorithms (e.g., Newton-Raphson, Fast Decoupled Power Flow, Gauss-Seidel) for the distributed slack formulation of the power flow equations.
- Integrating these implementations into MATPOWER such that the user can easily choose a distributed slack formulation as an additional option when solving power flow problems. Dr. Ray Zimmerman at Cornell University, who maintains MATPOWER, would likely be able to assist with this task.
- Comparing the performance and solution characteristics of the distributed slack and traditional power flow formulations on a variety of test cases.

2 Implementation and Study of Optimal Multiplier Variants of Power Flow Algorithms

As will be discussed in class, Newton-Raphson and other iterative methods for solving the power flow equations may diverge and thus fail to provide a solution. Divergence of iterative power flow algorithms can be addressed using so-called "Optimal Multiplier" approaches that adjust the size of the step taken at each iteration of the algorithm. Power flow algorithms that use an optimal multiplier are guaranteed to be *non-divergent*. (Roughly speaking, the iterates cannot grow arbitrarily far apart.) While this does not guarantee that the algorithm will *converge* to a power flow solution, empirical results show that optimal multiplier methods are particularly helpful for obtaining solutions in heavily stressed systems where other power flow algorithms diverge.

Previous research has developed power flow algorithms with optimal multipliers that use voltage phasor representations in both polar and rectangular coordinates. A summary and empirical comparison of two optimal multiplier algorithms using each of these coordinate choices is presented in the following paper:

> J.E. Tate and T.J. Overbye, "A Comparison of the Optimal Multiplier in Polar and Rectangular Coordinates," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1667-1674, November 2005.

The MATPOWER package does not currently have either the polar or rectangular optimal multiplier formulations. This project will implement both optimal multiplier formulations by modifying the Newton-Raphson algorithms in MATPOWER. Additionally, this project will replicate the numerical analyses conducted by Tate and Overbye in the paper above using recently released test cases that model large-scale power systems.[2] Finally, this project will study the impact that various initializations have on optimal-multiplier-based power flow algorithms by replicating the third problem in Homework 1 with both the polar and rectangular optimal multiplier algorithms.

To summarize, tasks for this project include:

- Implementing both polar and rectangular variants of the optimal-multiplier-based Newton-Raphson algorithm. These variants are provided in the Tate and Overbye paper cited above.
- Integrating these implementations into MATPOWER such that the user can easily choose these optimal multiplier formulations as additional options when solving power flow problems. Dr. Ray Zimmerman at Cornell University, who maintains MATPOWER, would likely be able to assist with this task.
- Replicating the numerical experiments in the Tate and Overbye paper cited above using recently released test cases.
- Repeating the third problem in Homework 1 with the optimal multiplier based algorithms to explore the impacts of various initializations.

[2] See the PGLib-OPF library of test cases, which are available at `https://github.com/power-grid-lib/pglib-opf`.

3 Implementation and Study of Power Flow Solvers in PowerModels.jl

Research projects lead by Los Alamos National Laboratory have developed a software package known as PowerModels.jl which provides a variety of functions related to power system operations.[3] PowerModels.jl is written in the Julia programming language, which has advantages in computational speed and modeling flexibility relative to Matlab, particularly with respect to formulating nonlinear optimization problems.

The PowerModels.jl package currently has limited functionality for solving power flow problems. This project would address these limitations by implementing and testing a variety of power flow algorithms in PowerModels.jl.

Tasks for this project include:

- Gaining experience with the Julia programming language and the PowerModels.jl package.
- Implementing various power flow algorithms (e.g., Newton-Raphson, Fast Decoupled Power Flow, Gauss-Seidel) in PowerModels.jl. Dr. Carleton Coffrin at Los Alamos National Laboratory, who maintains PowerModels.jl, would be able to assist with this task.
- Comparing the performance of the implemented algorithms with the existing power flow algorithms in PowerModels.jl and the algorithms in MATPOWER.

[3] `https://github.com/lanl-ansi/PowerModels.jl`

4 Benchmarking Holomorphic Embedding Methods for Solving Power Flow Problems

During class, we will briefly discuss a recently developed family of power flow solution algorithms known as "Holomorphic Embedding Methods". These algorithms exploit a complex variable representation of the power flow equations in order to apply solution techniques from the field of complex analysis. Holomorphic embedding algorithms claim a variety of theoretical advantages with respect to their convergence characteristics relative to traditional power flow algorithms. However, holomorphic embedding algorithms can suffer from poor numerical performance in practice.

For further details on Holomorphic Embedding Methods for solving power flow problems, refer to the following reference:

> A. Trias, "HELM: The Holomorphic Embedding Load-Flow Method. Foundations and Implementations," *Foundations and Trends in Electric Energy Systems*, vol. 3: no. 3-4, pp. 140-370, 2018.

Tasks for this project include:

- Reading the literature on holomorphic embedding methods to understand the associated implementation details.
- Implementing the holomorphic embedding method in Matlab.
- Integrating this implementation into Matpower such that the user can easily choose a holomorphic embedding formulation as an additional option when solving power flow problems. Dr. Ray Zimmerman at Cornell University, who maintains Matpower, would likely be able to assist with this task.
- Comparing the performance of the holomorphic embedding algorithm with the existing power flow algorithms in Matpower.

5 Learning Initializations to Iterative Algorithms

Many power systems algorithms iteratively update a solution until it satisfies a convergence tolerance. The performance of these algorithms has a strong dependence on the selected initialization. In some applications, a high-quality initialization can be obtained from a known solution to a related problem, such as the solution to the same system at a previous instant in time. However, high-quality initializations are often unavailable in other applications, such as contingency analyses. In these applications, engineers rely on intuition or "rules of thumb", such as a flat start ($V = 1$, $\theta = 0°$), for initializations. This may result in longer computational times or convergence failure.

One plausible method for improving iterative algorithms is to apply machine learning techniques in an attempt to improve the quality of initializations used in various applications. By learning how different initializations affect solution times, this method could help engineers appropriately initialize iterative algorithms in challenging situations. The large training dataset required to train a machine learning algorithm for this purpose can be obtained by off-line computations which repeatedly solve the relevant problem for a specific test case with varying parameter choices (e.g., different loading scenarios for the same system) and different initializations. The trained machine learning algorithm could result in substantial speed improvements in on-line computations. Potential applications of this method include iterative algorithms for power flow, optimal power flow, state estimation, etc., each of which would provide a variant of this project.

Tasks for this project include:

- Performing a literature review regarding the initialization of iterative algorithms for power system optimization and control.
- Creating a training dataset which contains the solution times and solution quality metrics associated with different initializations and loading scenarios.
- Selecting and implementing an appropriate machine learning algorithm.
- Training the machine learning algorithm on the training dataset.
- Evaluating the quality of the resulting initializations using additional scenarios beyond those in the training dataset.

6 Comparison of Distributed Optimization Algorithms for Optimal Power Flow Problems

Power systems have traditionally been optimized in a centralized fashion, i.e., all relevant information is sent to a centralized entity who solves an optimization problem and returns a setpoint to each controllable device. This approach may lack the scalability needed for operating future smart grids which need to consider orders of magnitude more controllable devices. Thus, there has been significant interest in transitioning to a distributed optimization paradigm where each controllable device makes decisions based on information communicated by a limited subset of neighboring devices. Distributed optimization algorithms attempt to reach the same solution as would be computed by a centralized entity. See the following paper for a survey of distributed optimization algorithms in a power systems context:

> D.K. Molzahn, F. Dorfler, H. Sandberg, S.H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems," *IEEE Transactions on Smart Grid*, vol. 8: no. 6, pp. 2941-2962, November 2017.

While a variety of distributed optimization algorithms have been proposed, there have been relatively few numerical comparisons of their characteristics. Such a comparison would help both researchers and practicing engineers understand the state of the art in this topic.

Tasks for this project include:

- Developing simple models of plausible communication infrastructures in future smart grids to serve as benchmark test cases.
- Implementing a selected set of distributed optimization algorithms.
- Applying the implemented algorithms to the benchmark test cases.
- Comparing the performance of each algorithm.

7 Machine Learning Techniques for Improving Semidefinite Relaxations of Optimal Power Flow Problems

As will be discussed in class, relaxations of optimal power flow problems have recently attracted significant attention from the research community. Convex relaxations simplify the AC power flow equations in optimal power flow problems in order to obtain formulations with various advantageous characteristics. See Section 4.1.1 of the following paper for further details regarding the semidefinite relaxation:

> D.K. Molzahn and I.A. Hiskens, "A Survey of Relaxations and Approximations of the Power Flow Equations," *Foundations and Trends in Electric Energy Systems*, vol. 4, no. 1-2, pp. 1-221, February 2019.

One particularly popular relaxation uses semidefinite programming to relax a rank-constrained formulation of the optimal power flow problem. Since solving this semidefinite programming relaxation can be computationally challenging, researchers have developed a method for exploiting the network sparsity. This method relies on row and column reordering techniques that are similar to those discussed during class in the context of Newton-Raphson power flow algorithms. Different reordering techniques can significantly impact the semidefinite relaxation's computational tractability.

State-of-the-art techniques for exploiting sparsity in semidefinite relaxations of optimal power flow problems use heuristic techniques that perform well on some benchmark test cases, but likely do not achieve the best possible computational performance.

In order to improve the computational performance of semidefinite programming relaxations, this project investigates the application of machine learning techniques in order to identify better row and column reordering techniques than are obtained through existing heuristic methods. Existing code based on the PowerModels.jl package has been developed in order to generate the data (computational times associated with various reordering choices) needed for training a machine learning algorithm. This project would apply various machine learning techniques to this data in order to improve the computational tractability of the semidefinite programming relaxations of optimal power flow problems.

Tasks for this project include:

- Familiarizing yourself with the PowerModels.jl package, particularly the semidefinite programming relaxation.
- Applying various machine learning algorithms to the training data (computational times associated with various reordering choices).
- Characterizing the performance of these algorithms on different test cases.

This project builds on an existing collaboration with Professor Bissan Ghaddar at the University of Waterloo, who will be able to give advice on the associated tasks.

8 Studying the Impact of Generator Capability Curves on AC Optimal Power Flow Problems

AC optimal power flow problems are typically formulated with "box" constraints that separately limit the active and reactive power outputs of the generators: $P_G^{min} \leq P_G \leq P_G^{max}$ and $Q_G^{min} \leq Q_G \leq Q_G^{max}$. More realistic generator capability models impose so-called "D-curve" limits that couple the generators' active and reactive power outputs in order to represent the generators' internal current magnitude limits. As described in the following paper, it is possible to infer the parameters required for D-curve limits using the box constraint data provided in typical power system datasets:

> B. Park, L. Tang, M.C. Ferris, and C.L. DeMarco, "Examination of Three Different ACOPF Formulations With Generator Capability Curves," *IEEE Transactions on Power Systems*, vol. 32: no. 4, pp. 2913-2923, July 2017.

This paper also performs a detailed analysis regarding how different D-curve formulations impact AC optimal power flow solutions. Since this paper was published in 2017, many new publicly available optimal power flow test cases have been developed. This project would extend the analyses in the Park, Tang, Ferris, and DeMarco (2017) paper to the PGLib-OPF library of test cases, which are available at `https://github.com/power-grid-lib/pglib-opf`.

Tasks for this project include:

- Performing a literature review of different formulations for modeling generator capability curves.
- Implementing the method described in the paper above in order to obtain D-curve parameters for the test cases in the PGLib-OPF library.
- Converting the D-curve formulations to a representation that is appropriate for input to MATPOWER. (See Section 6.4.3 of the MATPOWER manual, which is available at `https://matpower.org/docs/MATPOWER-manual-7.0.pdf`.)
- Performing similar numerical studies to provided in the Park, Tang, Ferris, and DeMarco (2017) paper using the PGLib-OPF test cases.

9 Integrating Sophisticated DC Power Flow Formulations in Power System Software Packages

Many commercial software packages for solving DC power flow and DC optimal power flow problems use loss approximation models and other variations from the "textbook" style assumptions used in academic software packages such as Matpower and PowerModels.jl. This makes it difficult for the research community to compare the performance of newly developed power flow and optimal power flow algorithms to the state-of-the-art methods used in industry. This project would help ameliorate this issue by implementing more sophisticated DC power flow formulations into academic software packages such as Matpower and PowerModels.jl. Descriptions of these more sophisticated DC power flow formulations are available in the following paper:

B. Stott, J. Jardim, and O. Alsac, "DC Power Flow Revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290-1300, August 2009.

Tasks for this project include:

- Performing a literature review of different DC power flow formulations.
- Implementing a selected set of these formulations in Matpower and/or PowerModels.jl such that the user can easily choose a various alternative DC power flow formulations as additional options when solving power flow problems. Dr. Ray Zimmerman at Cornell University would likely be able to assist with this task for Matpower. Dr. Carleton Coffrin at Los Alamos National Laboratory would be able to assist with this task for PowerModels.jl.
- Comparing the performance of the various formulations on a variety of test cases.

10 Studying the Set of Active Constraints in DC Optimal Power Flow Problems

In many power system optimization problems, researchers have observed that only a small fraction of the line flow constraints ever become active at the optimal solution, despite variations in the load profile and generation costs. In other words, only a small fraction of lines have flows at their limits. This observation has far-reaching implications not only for power system optimization, but also for the practical long-term planning, operation, and control of the system.

To formalize this observation, previous research has studied the set of constraints which can *potentially* be active for any choice of the cost function and loading within a specified range. Computing the set of potentially active constraints can be accomplished by solving optimization problems that maximize and minimize the flow on each line. If the maximum achievable line flow is less than the specified line flow limit, that limit can never be reached and is therefore redundant with respect to the remainder of the constraints. Further details on this approach are available:

> L.A. Roald and D.K. Molzahn, "Implied Constraint Satisfaction in Power System Optimization: The Impacts of Load Variations," `https://arxiv.org/abs/1904.01757`, April 2019.

Other complementary research in the following paper aims to learn the sets of constraints that are *actually* (as opposed to potentially) active in power system optimization problems as the loading varies within a specified range.

> Y. Ng, S. Misra, L. A. Roald and S. Backhaus, "Statistical Learning for DC Optimal Power Flow," *Power Systems Computation Conference (PSCC)*, June 2018.

The set of actually active constraints is necessarily contained in the set of potentially active constraints identified using the approach described in the Roald and Molzahn (2019) paper above. This project would investigate the gap between these sets for different test cases and different ranges of loads. In other words, this project aims to characterize the constraints that cannot be *a priori* eliminated as redundant versus the constraints that are actually active at typical solutions.

In principle, this question could be studied for AC optimal power flow problems. However, the are subtle issues that arise due to the non-convexity induced by the AC power flow equations. Therefore, this project could focus on DC optimal power flow problems such that the associated optimization problems are linear programs.

Tasks for this project include:

- Implementing the algorithm in the first paper listed above in order to compute the set of potentially active constraints by solving optimization problems that maximize and minimize the power flows on each line.
- Writing code that solves DC optimal power flow problems for varying loads and generation cost functions and then records the corresponding sets of active line flow constraints.
- Comparing and visualizing the sets of lines that are potentially active versus the sets of lines that are actually active for different ranges of loads and cost functions for a variety of test cases.

11 Best Practices for Recovering Solutions from the Output of a Power Flow Relaxation or Approximation

As we will discuss in class, there has recently been significant research interest in developing new relaxations and approximations of the AC power flow equations in order to better handle the non-convexities that these equations induce in power system optimization and control problems. These relaxations and approximations do not enforce the exact relationships between the voltage phasors and the power flows in order to achieve a more tractable (convex) optimization problem. This tractability comes at the cost of accuracy with respect to the original AC power flow equations. In particular, the voltage phasors resulting from a power flow relaxation or approximation may be inconsistent with the power flows.

This inconsistency is problematic for many applications that could benefit from power flow relaxations or approximations. Accordingly, this project will explore different ways to recover a solution from a power flow relaxation or an approximation that is as close as possible to satisfying the AC power flow equations.

As an illustrative example, the solution to a DC optimal power flow problem gives a representation of both the active power injections P and the voltage angles θ. These quantities do not give a consistent solution to the AC power flow equations due to the approximations inherent to the DC power flow formulation. In order to recover a pair of voltage angles and active power injections, one could either 1) compute active power injections by plugging the values of θ into the power flow equations (with 1 per unit voltage magnitudes) or 2) compute the phase angles associated with the power injections by solving a power flow problem with specified values of P from the DC optimal power flow solution. This project would study which of these options performs better in practice for the DC power flow approximation (and/or similar options for other power flow approximations and relaxations).

Tasks for this project include:

- Familiarizing yourself with the PowerModels.jl package, which implements a variety of power flow relaxations and approximations.
- Implementing code that quantifies the accuracy resulting from different methods for recovering AC power flow solutions from various power flow relaxations and approximations.
- Comparing the performance of the different solution recovery techniques on a variety of test cases.

12 Evaluating AC Feasibility for Solutions to DC Transmission Switching Problems

While it may seem counterintuitive, power systems can sometimes be operated more efficiently if certain transmission lines are removed from the system since lines with relatively high impedances can restrict the amount of power flows on neighboring lines.[4] In these situations, the operational cost can be reduced by "transmission switching", i.e., opening the breakers on certain lines to remove them from the system. Previous research has studied the transmission switching problem (i.e., identifying which transmission lines to remove in a given scenario), often using the DC power flow approximation instead of the more accurate AC power flow equations. Use of the DC power flow approximation results in a "mixed-integer linear programming" (MILP) formulation, with discrete variables associated with the decision to remove a line and continuous and linearly constrained variables representing the DC approximation of the power flow equations. See the following paper for an early survey on the transmission switching problem:

> K.W. Hedman, S.S. Oren, and R.P. O'Neill, "A Review of Transmission Switching and Network Topology Optimization," *IEEE Power and Energy Society General Meeting*, July 2011.

Despite the widespread use of the DC transmission switching formulation, more recent work observes that the solutions obtained from transmission switching are often infeasible with respect to network models that use the AC power flow equations. See, for instance, the studies described in the following paper:

> C. Coffrin, H. L. Hijazi, K. Lehmann and P. Van Hentenryck, "Primal and Dual Bounds for Optimal Transmission Switching," *Power Systems Computation Conference (PSCC)*, Wroclaw, Poland, June 2014.

This project aims to provide further analysis and confirmation of this observation by evaluating whether systems with lines removed according to the DC transmission switching problem have feasible solutions to AC optimal power flow problems.

Tasks for this project include:

- Formulating the DC transmission switching problem. A reference implementation available in PowerModels.jl may be helpful in developing this formulation.
- Solving the DC transmission switching problem using a mixed-integer linear programming solver for a variety of test cases and operating scenarios.
- Attempting to find an AC feasible solution to the network after removal of the lines associated with the DC transmission switching problem by solving the associated AC optimal power flow problem.
- Summarizing the results to determine how frequently an AC power flow solution is obtainable and whether it indeed results in lower-cost operation compared to the original network.

[4] This phenomonon also occurs in other contexts as well, for instance, as "Braess's Paradox" in traffic systems, where closure of certain roads can actually improve overall traffic flow.

13 State Estimation with Various Power Flow Approximations

The state estimation problem seeks to determine the best estimate of the system state based on an available set of noisy measurements. An optimization formulation of this problem minimizes a norm of the error between the measured values and the estimated values of various quantities. The AC power flow equations that are typically used to model the network result in computational challenges for a variety of solution algorithms. In particular, the AC power flow equations cause the state estimation problem to be non-convex, potentially having multiple "spurious" solutions in addition to the actual system state. A poor initialization may result in a state estimation algorithm converging to a spurious solution, leaving system operators without accurate knowledge of the actual system state.

In order to bypass the challenges associated with the AC power flow equations, this project investigates the feasibility of using relaxations and approximations of the power flow equations in state estimation problems. The convex optimization problems that result from the use of relaxations and approximations do not have spurious solutions and can be more computationally tractable than non-convex formulations that use the AC power flow equations. However, relaxing or approximating the power flow equations can induce inaccuracies in the resulting solutions.

This project would explore the trade-offs associated with various relaxations and approximations of the power flow equations with respect to computational speed and solution quality for state estimation problems.

Tasks for this project include:

- Formulating and implementing a solution algorithm for state estimation problems that use the AC power flow equations.
- Selecting a subset of power flow relaxations and/or approximations. Formulating and implementing solution algorithms for state estimation problems that use these relaxations and/or approximations.
- Developing a set of test cases in order to compare the state estimation solutions from various formulations. These test cases can be adopted from existing test cases[5] by modeling a set of measurements in the system.
- Solving the state estimation formulations for the test cases and comparing the results.

[5]See the PGLib-OPF library of test cases, which are available at `https://github.com/power-grid-lib/pglib-opf`.

14 Studying Spurious Solutions to State Estimation Algorithms

The state estimation problem seeks to determine the best estimate of the system state based on an available set of noisy measurements. An optimization formulation of this problem minimizes a norm of the error between the measured values and the estimated values of various quantities. The AC power flow equations that are typically used to model the network result in computational challenges for a variety of solution algorithms. In particular, the AC power flow equations cause the state estimation problem to be non-convex, potentially having multiple “spurious” solutions in addition to the actual system state. A poor initialization may result in a state estimation algorithm converging to a spurious solution, leaving system operators without accurate knowledge of the actual system state.

Some previous analyses suggest that the prevalence of spurious solutions to state estimation problems decreases with the addition of measurements to the problem:

> R.Y. Zhang, J. Lavaei, and R. Baldick, ”Spurious Local Minima in Power System State Estimation,” to appear in *IEEE Transactions on Control of Network Systems*.

This project would build an empirical framework for identifying spurious solutions to state estimation problems. This framework would solve state estimation problems using a variety of initializations on numerous test cases with a range of operational conditions and levels of measurement noise. The goal of this project is to identify any patterns regarding which scenarios are most likely to have spurious state estimation solutions. For instance, are spurious solutions more common in problems with many measurements that are very noisy or in problems with few measurements that have low noise?

Tasks for this project include:

- Formulating and implementing a solution algorithm for state estimation problems that use the AC power flow equations.
- Creating a framework that enables solution of state estimation problems using many different initializations, loading conditions, and measurement characteristics for a range of test cases. High-performance computing resources may be helpful for this purpose in order to expand beyond the computational capabilities of a personal computer. Test case development should leverage existing publicly available system models.[6]
- Analyzing the results to identify and characterize the prevalence of spurious solutions to state estimation problems.

[6]See the PGLib-OPF library of test cases, which are available at `https://github.com/power-grid-lib/pglib-opf`.

15 Your Proposed Project Here!

You are welcome to propose your own project on any topic related to ECE 6320. Feel free to stop in during office hours or to your idea with the instructor after class. If you want to proceed with your project idea, please send a brief project description to the instructor (`molzahn@gatech.edu`) that will be added to this document.

One potentially relevant resource for developing a project idea is an extensive set of large-scale realistic power system test cases developed by the GRID DATA projects funded by the Department of Energy. More details of regarding these projects are available at `https://arpa-e.energy.gov/sites/default/files/documents/files/GRIDDATA_ProgramOverview.pdf`.