POWER GRID FLOW CONTROL STUDIES AND HIGH SPEED SIMULATION

by

WILLIAM MICHAEL SIEVER

A DISSERTATION

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI–ROLLA

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

2007

Dr. Ann Miller, Co-advisor	Dr. Daniel Tauritz, Co-advisor
Dr. Bruce McMillin	Dr. Mariesa Crow

Dr. Donald Wunsch

PUBLICATION DISSERTATION OPTION

This dissertation consists of four articles which have been submitted for publication as follows:

Pages 4–17 have been published in the Proceedings of the 2006 Conference on Artificial Intelligence in Power Systems (AIESP).

Pages 18–26 are intended for publication in the Proceedings of the 2007 North American Power Symposium (NAPS).

Pages 27–47 are intended for publication in the Proceedings of the 2007 Conference on System of Systems Engineering (SoSE) in Service of Energy and Security.

Pages 48–74 have been submitted for publication in IEEE Transactions on Power Systems.

ABSTRACT

Flexible AC Transmission System (FACTS) devices, which use high-speed, high-power semiconductor technology to better control power grids, are expected to be vital components to regulate an increasingly overburdened and under regulated transmission system. The work presented here demonstrates that the most general of these devices, the Unified Power Flow Controller (UPFC), may be able to reduce the kinds of power flows that lead to cascading failures. A comprehensive plan to use the UPFC as a theoretical super set of FACTS devices is proposed, which can be used to significantly harden power grids against both intentional and natural failures. Finally, a set of techniques for high speed simulation is developed which can be used in conjunction with the proposed hardening plan to allow it to be scaled to actual systems.

ACKNOWLEDGMENTS

I would like to thank all my committee members for their aid and support. In particular the encouragement, guidance, and occasional kick in the keester from my advisors, Dr. Ann Miller and Dr. Daniel Tauritz, was vital to me during the development of the work presented here.

I would also like to thank Mr. Richard Steiner for providing insight into industrial power research. I also can not forget the encouragement, goading, and common commiseration of my friends, Ilker, Filiz, Scott, Chris, et al.

Finally, I need to acknowledge the people who taught me the most important things in life — my parents and my brother, who have been constant role models in my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION

1. INTRODUCTION

As recent blackouts around the world have shown, modern power grids have subtle vulnerabilities which make them susceptible to cascading failures, where a minor problem induces a domino effect which disables a significant portion of the system.

Society has become increasingly dependent on electric power systems as one of the most fundamental energy sources due, in part, to the tremendous reliability that the power industry has provided thus far. Unfortunately, electric power demand continues to grow while expansion of the transport system, the power lines that carry power from the generation facilities to consumers, has stagnated due to economic and environmental concerns.

The Electric Power Research Institute, an industry research consortium, has advocated the use of recent developments in high power semi-conductor technology in a family of new devices collectively called Flexible AC Transmission System (FACTS) devices. These provide new control modes as well as greater control accuracy and better response time than more traditional devices, which were often electromechanical in nature.

Although many of these devices are currently in use, the devices are primarily being used as replacements for older components and the new control capabilities are still being under utilized. The work included here comprises a number of components that are necessary to speed the adoption of this new technology. The first paper is novel in that it presents a simple metric to evaluate power grid health, then uses a simple model of a Unified Power Flow Controller (UPFC) to optimize the metric. Test results demonstrate that the metric outperforms a technique based on the maximum flow through a traditional graph. Most importantly, although the model used is simplistic, the paper provides empirical evidence of the benefit and extent to which a UPFC can control grid power flow.

The second paper uses a more advanced model of the UPFC that includes all of its control capabilities. This work again demonstrates empirical evidence of the degree of control achievable and, in addition, shows the modes of control capabilities that may be needed for power flow support. Although many UPFC models already exist, most enforce a specific mode of control, whereas the model being used here does not. Since the UPFC can be used in a

variety of control modes, this model represents a theoretical superset of the functionalities of many of the other FACTS devices. The paper provides more comprehensive evidence of the impact of FACTS devices on power flow control and provides a model to be used in more advanced power system simulation.

The third paper outlines an approach, based on game theory, to identify both critical grid vulnerabilities and a means to identify locations in the grid where FACTS technology (specifically UPFCs) can be used to harden the system against the vulnerabilities. Incremental use of these two techniques will lead to substantial hardening of the grid against blackouts. The hardening process relies on an underlying simulation which is capable of simulating extensive line failures. The details of the simulation are fully outlined. The UPFC is used primarily because it is unknown which form of power control is most beneficial *a priori*, but the UPFC coupled with the control approach outlined in the second paper represents a superset of the FACTS family. Although the study described will identify vital installation locations using hypothetical UPFCs, at the conclusion of the study each location can be evaluated to determine the minimum requirements and if the full capabilities of the UPFC are not required, a lesser device may be used instead.

The fourth paper presents a technique that utilizes computer algebra systems to simplify simulated systems to their essential operations. Essentially this is a way to "compile" a model into a high speed simulation. The technique provides advantages in both the software design and the run-time speed of simulation. The test cases demonstrate how the technique can speed up the evaluation of power system simulations by more than a factor of ten. The technique can be used in a variety of applications that require high speed or real time simulation including hardware-in-the-loop equipment evaluation, training, and situational awareness tools.

The technique presented in the third paper can identify ways to use FACTS technologies to substantially harden the grid against attack. Unfortunately, it requires five levels of nesting to: 1) incrementally harden the system, 2) use a genetic algorithm to identify either vulnerabilities (attacks) or UPFC placements to harden the system, 3) use a steady-state simulation to determine total power delivery, 4) use a numeric optimization technique to identify optimal UPFC settings for each time step, and 5) identify the actual time step values via steady-state simulation. The computational complexity of the entire process is on the order of product of

the complexity of each loop. I.e., $O(a_1 \cdot 2 \cdot (a_2 \cdot a_3 \cdot a_4 \cdot a_5 \cdot a_6))$, where the constants represent the computational complexity of each layer of nesting. Conservative values of each of these can be used to estimate the actual computational time required:

a_1 System is incrementally hardened 8 times. (Assuming up to 3 FACTS devices and each requires 2-3 iterations to fully harden the system)

a_2 Genetic Algorithms often require 200 generations of improvements to plateau.

a_3 Power system delivery requires 6 steps of simulation (5 failures occur)

a_4 Numeric Optimization requires approximately 7 iterations (based on estimates in the first paper)

a_5 Steady-State estimation takes 3 iterations (based on experience)

a_6 Each iteration of the steady state operation takes N seconds

Based on these values, an entire simulation will take approximately $403200N$ seconds, and this is a conservative value for a modest power system. Given the multiplicative dependency on the value of N, reducing N by 90% reduces the time of the entire process by 90% and may make a significant difference in the practical application of the proposed simulation.

Although it imposes limits on the degree of simulation, the high-speed simulation technique can be used to actually reduce the computation time required by 90%. The high-speed technique typically is resistant to topology changes, such as the installation of a UPFC or the removal of a line. However, allowing changes in UPFC configurations can be accomplished by using the UPFC model presented in the second paper. An inactive UPFC can be installed on every line and the genetic algorithm outlined in the third paper merely identifies which UPFC should be active. Moreover, the fourth paper already incorporates some guidelines to allow simulated line outages.

PAPER 1

Improving grid fault tolerance by optimal control of FACTS devices

Bill Siever, Ann Miller Department of Electrical and Computer Engineering

University of Missouri–Rolla

Rolla, MO 65409–0040

Email: {bsiever,milleran}@umr.edu

FAX: 573-341-4532

Daniel Tauritz Department of Computer Science

University of Missouri–Rolla

Rolla, MO 65409–0040

Email: tauritzd@umr.edu

FAX: 573-341-4501

Index Terms— **UPFC, FACTS, Gradient Descent**

Abstract— **One of the most promising applications of the family of electronic devices called Flexible AC Transmission System (FACTS) devices is to better regulate power flow in transmission grids. In particular, the Unified Power Flow Controller (UPFC) is the best choice for complete power flow control. By selecting proper installation locations and control techniques, UPFCs may be able to prevent the "domino effect," where a single fault leads to a widespread blackout. Due to installation costs, it is hoped that only a small number of devices will be needed to effectively regulate a large grid, however, selecting the optimal number of devices,**

identifying the best possible installation locations, and finding a technique for coordinated control of these device are still active areas of research. In this paper we provide empirical evidence that common optimization techniques may be used to identify control settings for UPFCs. The evidence indicates that the optimization techniques lend themselves to real-time use, as well as use during the planning phase to identify the best possible installation locations for UPFCs.

I. Introduction

In recent years, expansion of national power grids has been hampered by social, environmental, and economic constraints. During the same period of time, power demand has dramatically increased. This combination of increased demand and limited physical expansion, which is expected to continue for the foreseeable future, forces many components of the grid to operate at, or near, their operational limits. Since the power grid is essentially a free-flow network, when a power line fails, the power which it was carrying will be re-directed through other lines in the system. This may push the other lines past their operational capacity and may cause them to be disabled, either due to physical failure or by protective equipment. This second round of failures only exacerbates the problem and may lead to a "domino effect," known as a cascading failure, causing a widespread blackout like the 2003 blackout that affected large portions of north-eastern North America. In a May 2002 report to the President of the United States [1], the Department of Energy (DoE) referred to the over-burdened components as bottlenecks and provided a succinct summary of the significance of the problem:

> Our transmission infrastructure is at the heart of our economic well-being. Imagine an interstate highway system without storage depots or warehouses, where traffic congestion would mean not just a loss of time in delivering a commodity, but a loss of the commodity itself. This is the nature of the transmission infrastructure. That is why bottlenecks are so important to remove and why an efficient transmission infrastructure is so important to maintain and develop.

In addition to the increased congestion on the network, recent deregulation efforts have also introduced control problems. When power was strictly regulated and a single company controlled all three layers of the power system hierarchy (the generators, the long-distance transmission lines, and the local distribution systems), there was incentive to sacrifice efficiency at one level in order to improve efficiency and stability at another. Since deregulation, companies no longer have as much economic incentive to ensure stable power delivery. This is especially complicated in the transmission network where interconnected entities are responsible for power transfers that cannot be strictly controlled. Any transfer of power between two entities will inevitably lead to unwanted parallel loop flows through other parts of the grid which may substantially degrade the stability of a third party [2], [3].

All these problems are, in part, due to the free-flow nature of the power transmission grid: for the most part, power flows through the grid along the path of least resistance (Ohm's law). Most power systems contain elements that help regulate power flow such as phase changers, series compensation, and shunt compensation, but historically these devices were mechanically switched and may not be capable of reacting fast enough to prevent cascading failures. The need for better high-speed control of power flow led to an initiative at the Electric Power Research Institute to develop power-electronic based devices, employing high speed, high power semi-conductor technology, to help better regulate power flow. All these devices are collectively known as Flexible AC Transmission System (FACTS) devices. The family of FACTS devices includes high speed versions of traditional devices like phase changers, and series and shunt compensators, as well as devices based on a new technology, the voltage source converter (VSC). The most powerful of the VSC based devices is the Unified Power Flow Controller (UPFC). The UPFC can be attached between a bus and a power line to help control the phase angle, bus voltage, and line reactance. Because it can control each of these, the UPFC provides the most complete power flow control of any of the FACTS devices [3].

Effective economical use of UPFCs depends on minimizing the installation costs (essentially minimizing the number of UPFCs), while selecting installation locations that maximize system performance. In addition, a suitable control algorithm must be selected

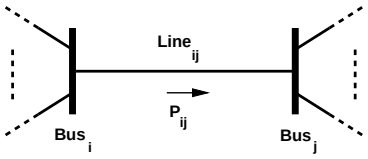
to ensure that multiple UPFCs are able to work cooperatively. These are interrelated problems — in order to select optimal locations, the control technique must be known a priori. The work presented here is primarily concerned with improving grid fault tolerance via UPFC control, so the goal of the control algorithm is to redirect power flow from the most overtaxed lines to under utilized lines.

To date, numerous control techniques have been proposed. Most of these fall into two categories: either they are primarily used for short term control of system dynamics or they are intended for long term (steady state) control of power flow. Generally these techniques can be used in conjunction, where the long term control identifies a set point for the steady state power flow and the short term control is used to maintain dynamic stability of the system about the set point and provide additional dampening for transients. Here we present empirical evidence that a common optimization technique may be acceptable for identifying long term set points that maximize fault tolerance by better distribution of power flow. These set points may then be used to help evaluate different potential installation locations. In addition, the evidence indicates that these techniques may be suitable for controlling the UPFCs on-line.

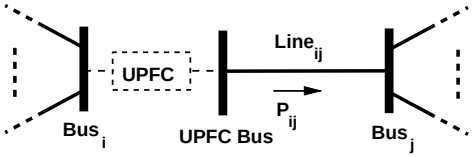
II. System Model

The power grid can be represented as a set of buses interconnected with lines of known series impedance. Each line also has a maximum rated power capacity ($Smax_{ij}$ for the line from Bus_i to Bus_j). Each bus in the system is associated with four state variables: real power, reactive power, voltage, and phase angle. At each bus, two of these variables have specified values and the other two are unknown (which are known and which are unknown depends on whether it is directly connected to a generator). The most common way to solve for the unknowns, which was used here, is the Newton-Raphson technique of computing load flow [4]. Once the unknown bus values are computed, line flows can be easily computed as well.

The UPFC's function in this work is to act as a means of forcing a specific amount of real power to flow through a line. By forcing power through a line, the remaining lines in the system adjust their power flow according to the physics of the system. The UPFC



(a) Original Power Line



(b) Virtual FACTS Device Inserted

Fig. 1. Original Line Model and Modifications for Simulated UPFC

was modeled as a mechanism which delivered real power to one of the power line's buses and drew a corresponding amount of real power from the other bus. Fig. 1(a) shows the original configuration of a line from Bus_i to Bus_j and Fig. 1(b) shows the representation of the same bus after a virtual UPFC has been inserted on Bus_i's side. In this case, the UPFC is assumed to be able to increase or decrease the real power flow through $Line_{ij}$ by 20% of the line capacity, $Smax_{ij}$. This is simulated by inserting a phantom UPFC bus and injecting the line's original power flow and the UPFC's modification of it ($\pm 20\%$ of $Smax_{ij}$) into Bus_j. A corresponding amount of power is deducted from Bus_i. Reactive power flow is maintained at the level present in the original system in a similar manner [5].

III. Optimization and Goals

The system performance can be measured against a number of factors, such as:

1) Number of lines with a power flow exceeding capacity
2) Number of lines with dangerously excessive power flow (20% or more over capacity)
3) Aggregate amount of power exceeding line capacities

Our goal is to mitigate the dangerously loaded lines that may lead to cascading failures, which can be achieved by balancing the overall power flow through the system, which minimizes line losses as well. Thus we chose a fitness measure based on each line's

percentage of its maximum capacity:

$$\sum_{lines} \left(w_{ij} \frac{S_{ij}}{Smax_{ij}} \right)^{2n} \tag{1}$$

This equation is based on a similar overload performance index used for ranking contingency severity [6]. This particular metric has a high penalty for lines that are at or over their capacity (when the fraction one or more). By varying n, the amount of disparity between overloads and near-overloads can also be adjusted, however all work presented here assumes that $n = 1$. A weighting factor, w_{ij}, can be used to rank the relative importance of different lines, but here we assume that all lines are of equal importance ($\forall i, j \ w_{ij} = 1$).

This measure was chosen for several reasons:

1) $|S_{ij}|$, the magnitude of the line's apparent power flow, reflects the current and voltage through the line, and hence the primary factor in thermal line failure.
2) It directly reflects the real control criteria — to evenly distribute power flow throughout the system. Better distribution of power flow leads to a lower score.
3) It is a continuous, scalar function in terms of the control variables, which allows the use of efficient gradient based search techniques.

IV. Optimization Procedure

The goal of optimization was to find valid power flow settings for UPFCs in a specific network configuration that minimizes the objective function being used (Eq. 1). As mentioned previously, the UPFCs were only allowed to modify power flow by up to $\pm 20\% of Smax_{ij}$, which represents the only optimization constraint. Any one of numerous constrained optimization techniques could be used, but we chose a relatively common form of sequential quadratic programming (SQP). The form of SQP used the following procedure:

1) Select a uniform random start point
2) Create a quadratic approximation of the search space using the BFGS method [7]
3) Determine the direction of steepest descent. If a local optimum or the maximum number of iterations is reached, stop

4) Use a line search technique until no longer able to "descend"

5) Update the quadratic approximation based on BFGS and goto step 3

Essentially a quadratic approximation is used to represent the objective function in terms of the control variables, then line search is used to find a local optima in a straight line. The quadratic approximation and search direction are updated and the process is repeated until a minimum is found. It is important to note that SQP methods are guaranteed to find a global optimum in super-linear time if the objective function is convex.

This is a sequential, iterative approach to optimization as opposed to using conventional Optimal Power Flow (OPF). OPF techniques would simultaneously solve for bus voltages, phase angles, UPFC control values, and constraints. The SQP approach here sequentially computes the UPFC control values based on objective function computations which compute the bus voltages and phase angles [8].

V. Experiments

All experiments were run on the IEEE 118 bus test system[1]. Since the objective of UPFC installation was to augment power flow and relieve network congestion, a heavily loaded system configuration was used (available upon request).

A. UPFC Impact on System

The first set of experiments was to determine the overall impact of a UPFC on a power grid. It was expected that a UPFC only effects a relatively small region of the power grid. This hypothesis was tested by placing a single UPFC in each of the possible installation locations and setting it to each of its limits ($+20\%$ and $-20\% of Smax_{ij}$), then measuring the change in power flow through each other line in the system. (This is not conclusive. In some cases increasing power flow through one circuit may cause power flows through other circuits to cancel, whereas a value between the UPFC limits may not induce the canceling power flow.). The results of this test are presented in Table I.

[1] `http://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm`

TABLE I
LINE AFFECTS FOR ALL POSSIBLE UPFC PLACEMENTS

% Dev of S	Max Lines Affected	Mean Affected	Std Dev
$> 1\%$	122	27.93	20.40
$> 5\%$	71	10.97	11.14
$> 10\%$	39	6.54	7.27
$> 15\%$	31	4.81	5.47
$> 20\%$	29	3.80	4.53
$> 25\%$	26	3.10	3.86

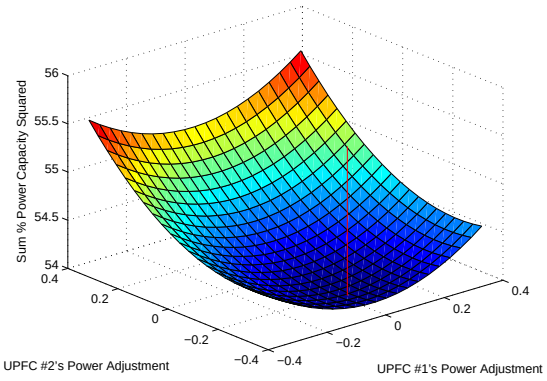
Note that typically fewer than 30 lines are even slightly impacted by the UPFC. Even in the worst case, only 122 of 186 lines are effected. This indicates that a complete load flow may not be necessary to determine the effect of a UPFC — instead it is possible to perform a load flow of only the buses which could be significantly effected by changing the UPFC's settings. If true, this reduces the complexity of the load flow computation which will be beneficial for any optimization techniques that depend on load flow computation.

B. Performance Metric State Space

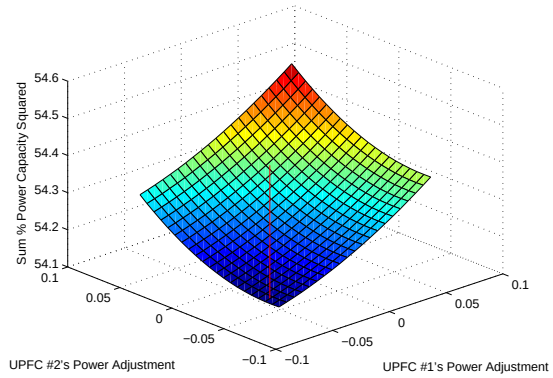
The second set of experiments was a test of the state space of the objective function being used (Eq. 1). In each of these tests, two UPFCs were installed at random locations in the system and a graph of the objective function was generated for all combinations of UPFC settings. Each UPFC was assumed to have 20 different control points in between its maximum and minimum value.

Figs. 2(a) and 2(b) show some typical examples of the state space of the objective function. As can be seen, the space seems to be "well behaved" and, consequently, a good candidate for SQP search techniques.

Although these results are promising, having more than two devices will lead to more complex search spaces which may have local minima. In order to test for multiple minima, Monte-Carlo style sampling was used. In these tests multiple UPFCs were randomly placed in the system and the SQP optimization procedure was performed using randomly chosen start points. If only a single optimum is present, then the algorithm should always



(a) Example of the effects on the objective function of UPFCs installed on lines 27 and 110



(b) Example of effects on the objective function of UPFCs installed on lines 5 and 54

Fig. 2. Examples of the State Space for Two Different UPFC Installation Locations. Vertical lines indicate the optimal settings.

converge to it. The system was tested with N UPFC devices where N was either 3, 5, 7, 9, 11, 13, 15, or 17. For each N, 100 different system configurations were randomly generated. Each system configuration was searched from 11 random starting points to try to find evidence of multiple minima. Each FACTS device was assumed to have 100 possible set points. The minimum step size of any of the UPFCs was used as the convergence criteria for the SQP optimization. Any solutions that had a Euclidean distance less than twice the mean step size were assumed to be equal (i.e., most of the FACTS devices are set at essentially the same setting). Using these criteria, all the minima found in each system were identical. Although this is not conclusive, it does provide a strong indication that with the given system and objective function, there are only global optima.

C. Real Time Control Issues

The use of SQP for real time control is heavily dependent on both the number of independent variables and the computational complexity of the objective function. In this case, each UPFCs control is an independent variable and the objective function is based on load flow computation. As has been previously mentioned, it may be possible to lower the complexity of the load flow computation by only computing the load flow for the part of the system affected by the UPFCs. This indicates that this technique may scale well to larger systems and, if the influence of multiple UPFCs mostly follows the law of superposition, then even multiple UPFC installations may benefit from reduced computation.

A second consideration is the number of load flows that must be computed. Since the SQP process uses a quadratic model of the objective function in terms of the control variables, it must perform repeated load flows to build and update this estimate. Fig. 3 shows the number of load flow calculations that were required for 100 random placements of 3, 5, 7, 9, 11, 15, or 17 devices. Since both the number and complexity of load flows seem to be bounded for any given size installation, it seems likely that SQP based minimization can be used in real time to ensure that at least a local minima is achieved.

The non-optimized version of load flow used here typically completed within 30ms on a 2GHz Pentium IV. This indicates that even for 17 UPFCs, the optimal long term settings could be found in under 15ms.

D. SQP vs. MaxFlow

An alternative version of UPFC control, utilizing a graph theory algorithm known as Max Flow, has been proposed [9], [10]. The MaxFlow algorithm was proposed to determine the maximum amount of "flow" between two points in a graph, where each arc of the graph has a maximum capacity. As applied to power systems, Max Flow can be used to determine the amount of power that can flow through each line without causing lines to exceed their capacity.

Although both MaxFlow and the technique presented here have their respective merits, the work presented here shows a substantial reduction in system stress. Generally the SQP

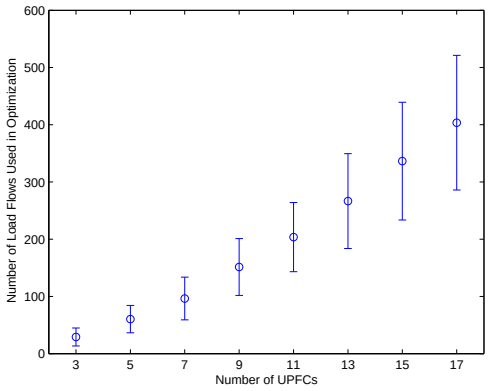


Fig. 3. Number of Load Flow Calls Used Per Optimization vs. Number of UPFCs. Central point is the mean and the error bars include the 95 percentile of 100 random samples.

optimization reduced the severity and number of overloaded lines more substantially than MaxFlow.

In order to compare the two systems, a test system was developed for each possible UPFC installation location. Each of these test cases was then sequentially subjected to all possible single line contingencies and the number and severity of overloaded lines were determined. A summary of the results, which can be seen in Tables II and III, indicate that SQP based optimization is better at reducing system stress. Table IV provides a summary of the pros and cons of each technique.

TABLE II
SUMMARY OF SQP CONTROL FOR EACH POSSIBLE UPFC PLACEMENT AND EACH SINGLE LINE CONTINGENCY

	Total Overloads	Overloads $> 10\%$	Overloads $> 20\%$
Maximum	26	19	14
Mean	0.6611	0.4585	0.3502
Std Dev	1.5928	1.2999	1.0722

TABLE III
SUMMARY OF MAX FLOW CONTROL FOR EACH POSSIBLE UPFC PLACEMENT AND EACH SINGLE LINE CONTINGENCY

	Total Overloads	Overloads $> 10\%$	Overloads $> 20\%$
Maximum	30	26	20
Mean	0.6937	0.4707	0.3605
Std Dev	1.6135	1.3080	1.0758

TABLE IV
COMPARISON OF MAX FLOW CONTROL AND SQP CONTROL

	Max Flow	SQP
Realism	Restricted to Real Power Flow only. Ignores reactive power and line losses.	Based on physical system power flow, implicitly includes reactive and losses.
Reliability	Guaranteed, but non-unique solution.	Depends on convergence of SQP and loadflow. May fail in extreme circumstances.
Speed	Simple and fast algorithm	Empirical evidence indicates suitable for on-line control
Resources	Global system knowledge	Global system knowledge

VI. FUTURE WORK

Although there was some study of the effects of a single UPFC on the grid leading to the conclusion that load flow only needs to be computed for the effected part of the grid, it would be beneficial to study if this remains true for systems with multiple UPFCs. It may be possible to approximate the combined effects by superposition of the effects of individual UPFCs.

The evidence presented indicates that the objective function being used here has only global minima, even when multiple UPFCs are being used. There are some questions that still need to be resolved before it is known whether SQP can be used in large systems:

1) Are local minima present, but ignored due to the quadratic approximation?
2) Does a quadratic model accurately approximate the objective function?
3) Are local minima prevalent in other systems?
4) Can it be proven that only global minima exist?

The answers to these questions may prove that SQP can be used in general to find global optima.

It is also important to compare SQP techniques to more common power system optimization techniques like Optimal Power Flow (OPF) [4]. Comparing both the computational complexity and quality of modeling may show the relative strengths and weaknesses of each technique.

Although the work presented here incorporated the effects of reactive power flow in the system, the UPFC's ability to control reactive power flow was neglected. The optimization process can be amended to include reactive power flow control, which should result in a further improvement in the quality of the system.

One of the biggest impediments of UPFC usage is the complexity of choosing optimal installation locations. As can be seen here, UPFCs can be used for both power flow compensation and for power flow restriction; they may be used on chronically congested lines in a restrictive manner or as means of increasing power flow through lightly loaded lines to draw power away from the congested area. This duality makes nearly every line in a power system a possible candidate for UPFC installation. In order to compare possible installation locations, the UPFCs control algorithm must already be known a priori. In addition, the size of the UPFC itself is a critical parameter. In the work presented here, it was assumed that the UPFC would be large enough to change the lines power flow by up to $\pm 20\%$ of the line's capacity ($Smax_{ij}$), but this is both unlikely and unnecessary. The form of SQP presented here may be used to evaluate the quality of potential UPFC locations because it is capable of finding the optimal set points for multiple UPFC locations and it may be extended to determine the optimal size of UPFC to install by including economic model indicating the costs of different control ranges. For instance, in one particular location a UPFC with only $\pm 7\%$ control may be sufficient and may only cost only a fraction of a larger UPFC with $\pm 20\%$ control that may be necessary in another location. An economic model may be used to select several small capacity devices rather than a few large devices or a combination of devices of various sizes.

VII. Conclusions

Based on the empirical evidence, SQP is a good choice for finding the optimal set points for systems with several UPFCs. The form of SQP used here, which uses load flow directly, has the ability to incorporate both real and reactive power flow measures in whatever optimality criteria is chosen. The optimality measure used, which was designed to improve fault tolerance via better power distribution, appears to be smooth and have

a single global optimum, even in systems with multiple UPFCs. If it can be proven that it is concave, then SQP based optimization techniques are guaranteed to find optimal set points. Both the size and number of load flow computations, on which SQP optimization relies, seem to have reasonable bounds. Based on these findings, SQP optimization appears to be a good candidate for real-time control of UPFC set points.

REFERENCES

[1] U. S. D. of Energy, "National transmission grid study," May 2002.

[2] E. J. Lerner, "What's wrong with the electric grid," *The Industrial Physicist*, Oct.-Nov. 2003.

[3] Y. H. Song and A. T. Johns, Eds., *Flexible AC Transmission Systems (FACTS)*. The Institution of Electrical Engineers, 1999.

[4] M. Crow, *Computational Methods for Electric Power System*. CRC Press, 2003.

[5] D. J. Gotham and G. Heydt, "Power flow control and power flow studies for systems with facts devices," *IEEE Trans. Power Syst.*, vol. 13, Feb. 1998.

[6] A. J. Wood and B. F. Wollenberg, *Power Generation, Operations, and Control*, 2nd ed. Wiley, 1984.

[7] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox For Use with Matlab*. The Math Works, Inc., 1999.

[8] E. Acha, C. Fuerta-Esquivel, H. Ambriz-Perez, and C. Angeles-Camacho, *FACTS: Modellins and Simulation in Power Networks*, 1st ed. Wiley, 2004.

[9] A. Armbruster, B. McMillin, and M. Crow, "Controlling power flow using facts devices and the max flow algorithm," in *Proc. of the International Conference on Power Systems and Control*, Abuja, Nigeria, Dec. 2002.

[10] A. Armbruster, M. Gosnell, B. McMillin, and M. Crow, "Power transmission control using distributed max flow," in *29th Annual International Computer Software and Applications Conference, 2005*, vol. 1, July 2005.

PAPER 2

Using a Simple UPFC Model to Identify Optimal UPFC Control Settings

W. M. Siever, D. R. Tauritz, *Member, IEEE,* A. Miller, *Senior Member, IEEE,*
M. L. Crow, *Senior Member, IEEE*

***Abstract*— Recent developments in semiconductor technology have lead to widespread research into Flexible AC Transmission System (FACTS) devices, new high-speed devices which offer unprecedented control capabilities. One of the most powerful of these devices, the Unified Power Flow Controller, is unique in that it is a superset of several of the lesser devices in the FACTS family. One primary area of investigation for applying FACTS technology is in the better regulation and control of steady-state power flow. In these applications the optimal means of compensation or regulation may not be known in advance, so traditional FACTS models may not be appropriate. A simple steady-state model is presented here which does not assume any control mode, and, when combined with traditional optimization techniques, can be used to identify which control modes may be most beneficial at a particular bus and may also aid the identification of where FACTS devices can be installed to maximize their potential benefits.**

***Index Terms*— FACTS devices, UPFC, Steady-State simulation, Modeling**

I. Introduction

Recent developments in semiconductor technology have lead to widespread research into Flexible AC Transmission System (FACTS) devices, new high-speed power devices which offer unprecedented control capabilities. One of the most powerful of these devices, the Unified Power Flow Controller, is unique in that it is a superset of several of the lesser devices in the FACTS family. It is hoped that appropriate application of FACTS technology will be able to overcome a decrease in power system security that has resulted from a combination of increasing demand, a reduction in transmission system construction, and the increased power transfers resulting from deregulation. It is hoped that FACTS technology can be applied to better control power flow and reduce the burden on congested transmission corridors by better utilization of under used transmission paths. In these types of applications a measure of quality may be available, but the optimal means of compensation or regulation may not be known in advance, so traditional FACTS models may not be appropriate.

UPFC Model

An ideal model of a UPFC consists of a voltage source connected to a source bus in shunt and another voltage source connected in series with a transmission line leaving the source bus, as seen in Fig. 1. The series injected voltage provides a means of controlling the real and reactive power transfer through the line while the shunt component can provide reactive compensation to the source bus and also provides the real power injected by the series component. The shunt component is essentially a voltage source converter (VCS), which can convert power from the source bus into DC and temporarily store it in a short term storage device. This power can then be returned to the bus to provide the reactive compensation or the series component of the UPFC can use some of the stored energy to inject real and reactive power into the line and thus modify the power flow through the line. The UPFC does not generate power, so any real power injected by the series source is provided by the DC converter. In essence the injected power is drawn by the shunt component. In practical implementations the magnitude of

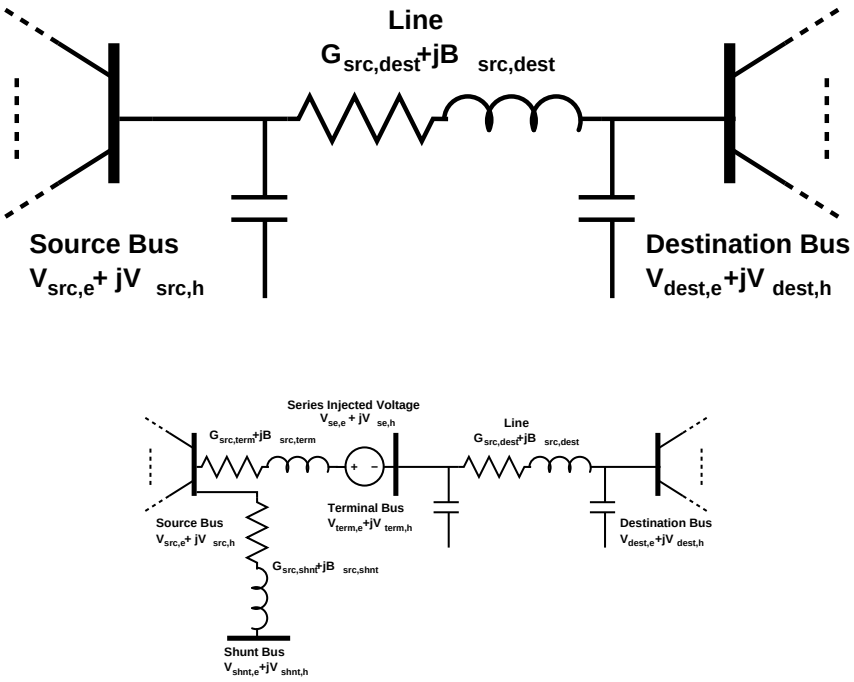


Fig. 1. A Line Model and the Same Line with a UPFC

the voltages are constrained by the limitations of the DC storage and current flow limitations.

The UPFC model can have twelve unique forms of steady-state control and typical system studies use a loadflow implementation that assumes that one particular mode will be used. In most cases, the power flow equations are augmented with the control mode equations and the loadflow solution automatically determines the values of series injected voltage and the shunt voltage to achieve the desired set point [2], [3], [1], [6]. I.e., the desired real and reactive power flows as well as the target reactive compensation are explicitly provided and the loadflow process identifies corresponding values of the series and shunt voltages and phases to achieve them.

This form of computing values is logical for estimating values in a system where UPFCs are currently installed and ideal target operating conditions are known. However, when new installation locations need to be identified, the evaluation criteria may not provide explicit values. In prior work by the authors, it was demonstrated that a simple two layer optimization process could be used to identify power flow set points for a measure of transmission system

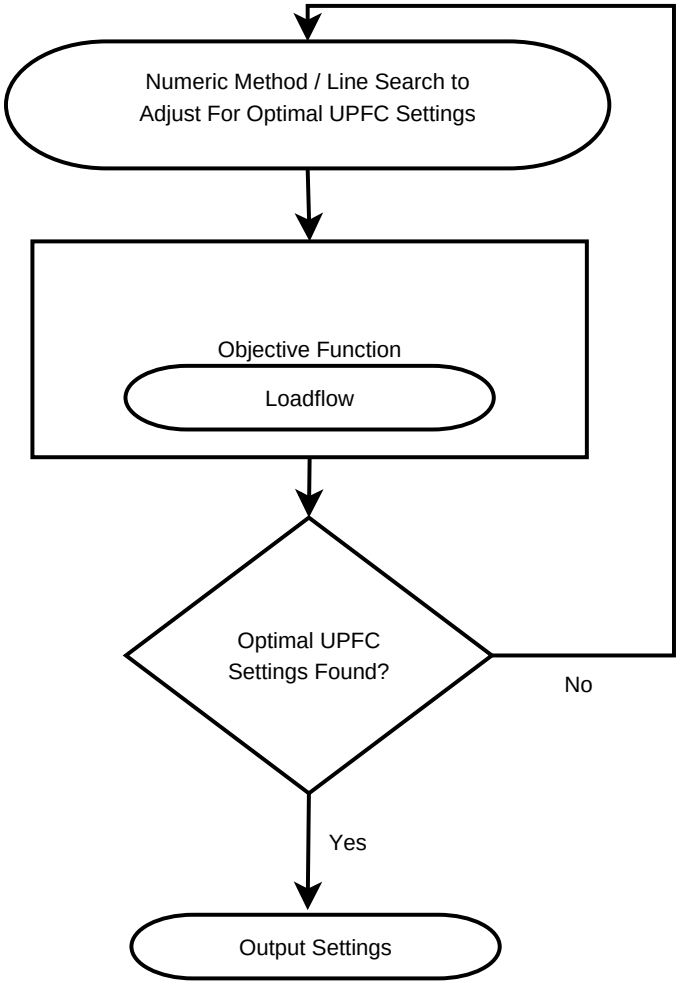


Fig. 2. Process to Identify Optimal UPFC Settings

health[5], [4]. The two layer approach, as shown in Fig. 2, was used because the objective being optimized was difficult to incorporate into traditional power flow techniques. In that work, UPFC settings were found that optimized a global objective function which was a metric of the balance of power flow. The metric itself was based on line flows, which in turn were computed from a standard loadflow algorithm. This work had two substantial conclusions: 1) UPFCs effects are very localized and typical only have a significant impact on a few lines, and 2) UPFCs can be used to reduce line overloads that are a consequence of contingencies. Although the basic conclusions of the study remain valid, only a small segment of the UPFC control capabilities were actually being utilized. The work presented here allows for the use of a more comprehensive UPFC model.

In order to evaluate where and what FACTS devices should be installed in a system, it is vital to know the impact such devices may have when being used optimally. Although the exact form of objective function may change, a global optimization criteria based on operating state is an important method of comparing different potential system configurations. Unfortunately, traditional loadflow techniques are intended purely for system identification and do not lend themselves to inclusion of complex optimization criteria.

The two layer process, on the other hand, may provide a means of optimizing any criteria provided. Moreover, if the optimization criteria is continuous, optimal control values for the UPFCs may be found via application of traditional gradient based optimization techniques. In the work here the desired control mode is unknown and may change depending on the conditions of the system or the objective, so a simpler model is used in which the shunt voltage, series voltage, and series phase angle are specified directly. The shunt phase angle is left free to ensure that the shunt can meet the power consumption demands of the series source. Although the two layer search could be used on the specified set points, the limits impose non-linear constraints whereas the direct search based on the voltages and phase angles can use constant constraints which allows the use of a simpler set of optimization techniques.

II. Model Inclusion in Loadflow

One of the biggest benefits of the simple UPFC model is its ease of use in steady-state models. Traditional loadflow implementations require only a minimum of changes to work with the technique. For each simulated UPFC, two new buses need to be created, one PV bus corresponding to the UPFC's shunt component and one PQ bus corresponding to the line connected terminals as seen in Fig. 1. Computational modifications need only be made to the construction of the Jacobian for lines corresponding to the shunt and series UPFC lines as well as the final power flow calculations for those buses. The equations needed to modify a rectangular loadflow implementation are provided in the Appendix.

III. Conclusions

The idea outlined here shows how traditional gradient-based optimization procedures can be used to identify optimal means of selecting control settings for FACTS devices. This simple technique could prove to be a very useful tool for evaluating where FACTS devices should be used. In addition, once it has been used to determine the optimal control settings for a variety of scenarios at a particular installation location, these settings can then be analyzed to determine which control modes are truly needed for control at that location.

Appendix

The equations presented here are derived as modifications to the standard load flow equations. A delta (Δ) indicates the change in power that results from the UPFC, so the values need to be added to the values already computed by default. By representing the shunt bus as a normal PV bus and the series injection as a modification to the original system, little modification is necessary to a standard loadflow procedure. Note that the shunt bus will maintain the set voltage value but adjust the phase angle so as to satisfy the power consumed by the series injection.

A. Modifications To Source Buses

1) Source Bus ΔP :

$$\begin{aligned} \Delta P_{src} &= -V_{src,e} G_{src,term} V_{se,e} + \\ & V_{src,e} B_{src,term} V_{se,h} - \\ & V_{src,h} G_{src,term} V_{se,h} - \\ & V_{src,h} B_{src,term} V_{se,e} \end{aligned}$$

2) Source BusΔQ:

$$\begin{aligned}\Delta Q_{src} &= V_{src,e}G_{src,term}V_{se,h} + \\ &\quad V_{src,e}B_{src,term}V_{se,e} - \\ &\quad V_{src,h}G_{src,term}V_{se,e} + \\ &\quad V_{src,h}B_{src,term}V_{se,h}\end{aligned}$$

3) Source Bus Derivatives - Jacobian Components:

$$\begin{aligned}\frac{\partial \Delta P_{src}}{\partial V_{src,e}} &= -G_{src,term}V_{se,e} + B_{src,term}V_{se,h} \\ \frac{\partial \Delta P_{src}}{\partial V_{src,h}} &= -G_{src,term}V_{se,h} - B_{src,term}V_{se,e} \\ \frac{\partial \Delta Q_{src}}{\partial V_{src,e}} &= G_{src,term}V_{se,h} + B_{src,term}V_{se,e} \\ \frac{\partial \Delta Q_{src}}{\partial V_{src,h}} &= -G_{src,term}V_{se,e} + B_{src,term}V_{se,h}\end{aligned}$$

B. Modifications to Terminal Buses

1) Terminal Bus ΔP:

$$\begin{aligned}\Delta P_{term} &= V_{term,e}G_{term,src}V_{se,e} - \\ &\quad V_{term,e}B_{term,src}V_{se,h} + \\ &\quad V_{term,h}G_{term,src}V_{se,h} + \\ &\quad V_{term,h}B_{term,src}V_{se,e}\end{aligned}$$

2) Terminal Bus ΔQ:

$$\begin{aligned}\Delta Q_{term} &= -V_{term,e}G_{term,src}V_{se,h} - \\ &\quad V_{term,e}B_{term,src}V_{se,e} + \\ &\quad V_{term,h}G_{term,src}V_{se,e} - \\ &\quad V_{term,h}B_{term,src}V_{se,h}\end{aligned}$$

3) Terminal Bus Derivatives - Jacobian Components:

$$\begin{aligned}
\frac{\partial \Delta P_{term}}{\partial V_{term,e}} &= G_{term,src}V_{se,e} - B_{term,src}V_{se,h} \\
\frac{\partial \Delta P_{term}}{\partial V_{term,h}} &= G_{term,src}V_{se,h} + B_{term,src}V_{se,e} \\
\frac{\partial \Delta Q_{term}}{\partial V_{term,e}} &= -G_{term,src}V_{se,h} - B_{term,src}V_{se,e} \\
\frac{\partial \Delta Q_{term}}{\partial V_{term,h}} &= G_{term,src}V_{se,e} - B_{term,src}V_{se,h}
\end{aligned}$$

4) Shunt Real Power Injection:

$$\begin{aligned}
\Delta P_{series} = {} & V_{term,e}G_{term,src}V_{se,e} - \\
& V_{term,h}B_{term,src}V_{se,e} + \\
& G_{term,src}V_{se,e}^2 - \\
& V_{se,e}V_{src,e}G_{src,term} + \\
& V_{se,e}V_{src,h}B_{src,term} + \\
& V_{term,e}B_{term,src}V_{se,h} + \\
& V_{term,h}G_{term,src}V_{se,h} + \\
& V_{se,h}^2G_{term,src} - \\
& V_{se,h}V_{src,e}B_{src,term} - \\
& V_{se,h}V_{src,h}G_{src,term}
\end{aligned}$$

5) Shunt Bus Derivatives - Jacobian Components:

$$\begin{aligned}
\frac{\partial \Delta P_{series}}{\partial V_{src,e}} &= -V_{se,e}G_{src,term} - V_{se,h}B_{src,term} \\
\frac{\partial \Delta P_{series}}{\partial V_{src,h}} &= V_{se,e}B_{src,term} - V_{se,h}G_{src,term} \\
\frac{\partial \Delta P_{series}}{\partial V_{term,e}} &= G_{term,src}V_{se,e} + B_{term,src}V_{se,h}
\end{aligned}$$

REFERENCES

[1] E. Acha, C.R. Fuerta-Esquivel, H Ambriz-Perez, and C. Angeles-Camacho. *FACTS: Modelling and Simulation in Power Networks*. Wiley, first edition, 2004.

[2] C.R. Fuerte-Esquivel and E. Acha. Unified power flow controller: a critical comparison of newton-raphson UPFC algorithms in power flow studies. *Generation, Transmission and Distribution, IEE Proceedings*, 144(5):437–444, 1997.

[3] C.R. Fuerte-Esquivel, E. Acha, and H. Ambriz-Perez. A comprehensive Newton-Raphson UPFC model for the quadratic power flow solution of practical power networks. *IEEE Transactions on Power Systems*, 15(1):102–109, 2000.

[4] W. Siever, R. P. Kalyani, M. L. Crow, and D. R. Tauritz. UPFC control employing gradient descent search. In *Proceedings of the 37th Annual North American Power Symposium*, pages 379–382, October 2005.

[5] William M. Siever, Daniel R. Tauritz, and Ann Miller. Improving grid fault tolerance by optimal control of facts devices. *Proceedings of First International ICSC Symposium on Artificial Intelligence in Energy Systems and Power*, February 2006.

[6] Xiao-Ping Zhang and K.R. Godfrey. Advanced unified power flow controller model for power system steady state control. In *Electric Utility Deregulation, Restructuring and Power Technologies, 2004. (DRPT 2004). Proceedings of the 2004 IEEE International Conference on*, volume 1, pages 228–233, 2004.

PAPER 3

Power Grid Protection via FACTS Devices

Bill Siever, Ann Miller, Daniel Tauritz Department of Electrical and Computer Engineering
University of Missouri–Rolla
Email: {bsiever,milleran,tauritzd}@umr.edu

Abstract— **A stable electricity supply is vital for modern society both because of its direct uses in heating, lighting, etc. and because electric transmission grids are a fundamental infrastructure on which many other vital infrastructures rely. Unfortunately, due to increasing power consumption, increased power transfers due to deregulation, and a decrease in new construction, power transmission lines are often operating near their operational limits. This leaves them in a fragile state where a few small failures, whether naturally occurring or intentional, could induce a cascading failure (a blackout). Recent development of high-speed, semi-conductor based devices provides a means of better power flow control. One of the most powerful of these devices, the Unified Power Flow Controller, can be used as a theoretical model to study how these devices can be used to improve power grid resilience. By developing an appropriate simulation and using genetic algorithms, critical vulnerabilities can be identified as well as fixed. These techniques can be used to iteratively harden the grid, making it less prone to blackout and better able to forestall or reduce the severity of unavoidable blackouts.**

Keywords: FACTS, UPFCs, Power Grid, Genetic Algorithms

I. Power System Conditions

Modern industrialized society has become dependent on electric power. In fact, in a recent report to the President of the United States, the U.S. Department of Energy said "Electricity is a cornerstone on which the economy and the daily lives of our nation's citizens depend. This essential commodity has no substitute " [7]. Not only does electric power directly provide heating, lighting and the power that drives manufacturing plants, but it is also a vital resource on which other infrastructures, including water distribution, sewage treatment and removal, emergency services, and traffic flow control, rely. Unfortunately, power grids all over the world are facing conditions which may jeopardize their ability to satisfy future demand for power as well as making them a target for terrorist attack.

Electric power is produced at large generating facilities and then "transmitted" over a system known as the transmission grid to regional distribution systems. The transmission grid, which consists of many long-distance, high-voltage lines and the buses to which they are connected, is really at the heart of the electric power industry. The transmission grid is the fundamental link between power producers and consumers, and, unfortunately, is becoming increasingly overburdened. Over the past decade demand for electricity has steadily increased and deregulation has spurred increased power transfers, but due to environmental, economic, and social concerns, the transmission grid has had relatively few upgrades. As a result of these events, many of the components are operating near their intended capacity. Prior to deregulation a top-down approach could be used to distribute power evenly through the entire system, however, due to deregulation there is incentive for transmission operators to operate as near capacity as possible. This allows excess power to be purchased from distant markets, but at the expense of reduced system stability margins. For the foreseeable future the power industry will be able to produce enough power to meet customer demands, however, the current transmission grid may be operating so close to its limits that a small failures could cause frequent blackouts.

Transmission grids have two features which make them prone to catastrophic failure. First, because transmission lines often cross vast, unmonitored space, they are susceptible

to both natural failures (ice, wind damage, tree contact) and intentional disruption (terrorist attack). Second, when a transmission line fails, the power which it was carrying flows over other lines. In a system where many components are already operating near their limits, the additional demands following a failure can cause other components to fail. The induced failures can, in turn, induce other failures that eventually lead to a domino effect that causes a blackout.

Power flow in the transmission grid today is largely dictated by Ohm's laws: power flows along the path of least resistance. Historically, the flow of power has been controlled by adjusting where the power is being generated and by "compensating" the lines, where electromechanical devices physically add or remove components to change line impedances. Although this was satisfactory when the grid was operating well below its maximum capacity, as the grid becomes increasingly overburdened it becomes vital to have better control over the flow of power to help mitigate cascading outages by directing power flow away from components that are near their failure point. By using automatic control algorithms and high-speed, accurate power flow control in key locations, it may be possible to mitigate or at least reduce the severity of cascading outages. Having this additional control may also be a vital element to defending power systems against deliberate physical and cyber attack.

The power grid is considered to be a significant target for terrorist attack because, due to its large scale, it is susceptible to a number of different attacks including: physical destruction of lines, physical control of a substation or generating facility, and cyber attacks on control and communication systems. Due to the sheer size of the system it is impossible to effectively protect all the physical components, and due to the complex control interactions of the different companies and components in the system a comprehensive cyber defense is also infeasible. In addition to being vulnerable, the power grid makes a tantalizing terrorist target due to the havoc that follows a short disruption. This was particularly evident following the August 2003 blackout that effected a significant region of North America. In addition to the financial losses incurred due to business closings, a number

of vital services including 911 service, sewage treatment, and water service were lost due to their reliance on electricity. Moreover, there is evidence that grid attacks are actively being investigated by terrorist organizations. In a statement to the joint subcommittee of the House of Representatives, Christopher Cox, a representative from the state of California, reported that "Al-Qaida computers seized in Afghanistan in 2001 had logged on to sites offering that offer [sic] software and programming instructions for the distributed control systems (DCS) and Supervisory-control and Data-acquisition (SCADA) systems that run power, water, transport and communications grids. [6]"

A. Unified Power Flow Controllers

In an effort to help better control power flow, the Electric Power Research Institute (EPRI) sponsored an initiative to develop a new class of power control devices called Flexible AC Transmission System (FACTS) devices. These devices, which are based on recent improvements in semi-conductor technology, can be used to help solve a variety of power control problems. By using the latest semi-conductor technology, these devices are able to control AC power in a substantially new way which is both faster and more precise than previous techniques.

One of the most powerful forms of FACTS device is the Unified Power Flow Controller (UPFC). As its name suggests, its primary role is to provide control over power flow. A UPFC is installed on a specific power line and provides almost total control over the power flowing through that line. Due to the nature of electric power flow, increasing or decreasing the flow through one line has an ancillary effect on the lines to which it is connected. This allows a single UPFC to have significant impact: it can be used to increase power flow through a line and potentially draw excess power away from an "upstream" lines that are operating over capacity, or it can restrict power flow and reduce the load on "down stream" lines. Due to high installation cost it is impractical to install more than a few UPFCs in a system, but a few devices cooperatively using their ancillary impact may provide enough regulation to redirect power flow and avoid or at least reduce the overload on critical lines.

Although UPFCs can be used to mitigate a variety of operating conditions, the work here focuses on finding ways to relieve or at least reduce the severity of cascading outages. The most significant cascading outages are a direct result of transmission lines carrying higher-than-normal amounts of current, which causes the metal to expand and sag. A failure occurs when the line either sags into contact with a ground source, such as a tree, or weakens to the point of that its own weight causes it to break, much like a fuse. Either of these failures is due to carrying above average current for a sustained period (several seconds to hours). Sagging into trees was the most significant contributor to the North American blackout of 2003 [7]. In that case, failure to maintain properly trimmed trees, rather than excessive line sage, was the major cause of failure.

UPFCs are studied here primarily because they offer a comprehensive means of power flow control, being able to control both real and reactive power flow as well as being able to regulate bus voltage. UPFCs have a total of twelve different forms of control [8] and represent a super set of the capabilities of other devices in the FACTS family as well as high speed versions of more traditional electromechanical means of control. For the work proposed here, the ideal control mode is unknown *a priori*, and, more importantly, may be different for different system vulnerabilities. I.e., under one type of failure the UPFCs may be best used for power flow regulation and in another scenario additional voltage support may be more important. A theoretical UPFC provides an ideal model of control capabilities because, with the appropriate control algorithm, it can seamlessly change control modes to suit the current situation. The plan presented here can be used to indicate where system vulnerabilities exist and further studies can easily identify the specific form of device (UPFC, other FACTS, or traditional means) to provide the best cost benefit for system defense.

B. Using UPFCs for CIP

The remainder of the paper looks at three inter-connected problems: 1) identifying the kinds of attacks the power grid is susceptible to, 2) finding installation locations that allow a few UPFCs to substantially reduce the likelihood of cascading failures, and 3) modeling the elements necessary to simulate both simple cascading failures and the control capabilities of

UPFCs. Each of these topics alone presents a complex problem and the approach presented here is not intended to be a panacea to solve all power grid vulnerabilities. Instead, a simplistic approach is outlined to explore the feasibility and potential impact of the use of FACTS devices for a specific type of system vulnerability. Following the feasibility study, detailed analysis can be performed to determine the real-world applicability of the results. Essentially the technique can be used to provide some recommended solutions to a very complicated problem, which system engineers can then evaluate and refine.

The approach presented here is based on a game-theory model of attackers and defenders and requires iterative cycles of simulated attacks. As such, a power system simulation that models the most significant features of both cascading failures and UPFCs is required. The simulation will be used in two ways: the first will identify the attacks to which the system is highly vulnerable and the second is used to find ways for UPFCs to mitigate the attacks. By repeating the two cycles it will be possible to incrementally harden the system against the most probable attacks and failures.

II. Iterative Hardening

The proposed technique for identifying system vulnerabilities and potential ways to rectify them is based on a basic game theory approach similar to that proposed in [1]. In this approach, two distinct games are played: one by an attacker and one by a defender.

A. The Attacker's Game

The goal of the attacker's game is to identify the brittle areas in the network. The actual goal of the attacker is to cause the most damage with the least effort. Ideally the attacker will select a few lines that will cause a total blackout. The attacker's game can be thought of as a simple discrete maximization problem, such as:

$$\arg\max_{\alpha} F\left(\emptyset, \beta\right) - F\left(\alpha, \beta\right) + G\left(\alpha, \beta\right) \tag{1}$$

where:

α is the attack plan, a schedule of what lines to remove and when to remove them

β is a set of parameters for the power system, including load profiles

$F(\alpha, \beta)$ is a function that simulates the power system and determines the total amount of power delivered

$G(\alpha, \beta)$ is a reward function for encouraging the simpler attacks

The term $F(\emptyset, \beta) - F(\alpha, \beta)$ measures the amount of power delivery lost due to the attack. The reward function, $G(\alpha, \beta)$, may also be dependent on the degree of power loss, so a three step plan may be preferred to a two step plan if the increase in damage is substantial. This maximization problem represents the typical intent of a malicious attack (maximal damage with minimal effort).

Power system parameters and operating conditions are nearly impossible to predict in advance, so the game can either assume that: 1) the attacker will try to take advantage of a peak load time and assume a specified worst case β, or 2) that β can be considered a random variable and, at the expense of considerably more computation, the expectation can be used:

$$\arg\max_{\alpha} \; E\left[F(\emptyset, \beta) - F(\alpha, \beta) + G(\alpha, \beta)\right]$$

(Note that the value of $F(\emptyset, \beta)$ is constant)

Exhaustive search can be used to find the most significant attacks on small systems, but unfortunately the problem search space grows exponentially with the attack size, so it is infeasible for large systems. At this time, there are no efficient techniques known for optimal search of this problem, however some important observations can be made:

1) It is expected that changing a single element of an attack may make the attack incrementally better, but there is no known method of identifying which change is optimal without exhaustive search
2) It is expected that mixing elements of two good attacks may yield a better attack

A technique known as a genetic algorithm (GA) is an ideal candidate for searching large combinatorial search spaces that exhibit these properties. GAs are loosely based on the concept of Darwinian evolution. A problem solution is maintained as a list (in this case

of line outages). A population of several solutions is "evolved" by using two mechanisms inspired by biological systems: mutation and crossover. An iterative algorithm is used to improve (evolve) the population. At each iteration members of the population are measured to determine how good each is relative to the others (using Equation 1). Many of the weakest members of the population are discarded. Next the remaining members of the population are selected for crossover, which corresponds to sexual reproduction (and operation number two above), where new solutions are produced based on components from each parent. Generally enough offspring are produced to replace the discarded members so as to keep the population size fixed. Finally the mutation operation, typically a form of random change (such as suggested in number one above) is applied at random to the population. At this point the weakest members have been eliminated, some of the good solutions have been randomly combined and some small random changes have been introduced, all in the hope of finding a better solution. Now the entire population is ready to be evaluated again and begin a second round of discarding, crossover, and mutation. The cycle continues until a suitable termination criteria is achieved such as reaching a performance plateau.

In essence, the GA maintains a population of good solutions. At each iteration it takes the best candidate solutions and, with some small probability, either makes a random change in the solution or combines solutions into a new candidate for the next generation. This process is similar to what a person would do when faced with a large combinatorial problem: they would test and find several potential solutions and then either try to make minor variations to incrementally improve the outcome, or to combine good solutions together to try to form better solutions. The main advantage of the GA is that it is automated, unbiased, doesn't require tedious analysis by a person, and works much faster than a person could.

B. The Defender's Game

Since the end goal is to demonstrate that UPFCs can defend the system against the weaknesses identified by the attacker, the defender's goal is to minimize the system's

brittleness, which can also be expressed as a discrete maximization problem:

$$\arg\max_{\beta} \; E\left[F\left(\alpha,\beta\right)+H\left(\beta\right)\right] \tag{2}$$

Where $H(\beta)$ is a reward function that encourages using as few UPFCs as possible, the expectation is taken over a set of likely attacks, and only the components of β that correspond to UPFC locations can be changed. By maximizing Equation 2, the defender is selecting places to install UPFCs that maximize the amount of power delivered over all the attacks to which the system was weakest.

The set of potential attacks will be taken directly from the best solutions to the Attacker's Game, and the probability of their incidence can be based on the same ranking used by the attacker (their complexity, $G\left(\alpha,\beta\right)$) or may assume that the probability of attack is related to the amount of damage incurred ($F\left(\emptyset,\beta\right)-F\left(\alpha,\beta\right)$). The latter corresponds to a mini-max game, where the defender minimizes the damage done by the attacker's best possible attacks.

Selecting installation locations for UPFCs is also a combinatorial problem with no known, optimal solution, but, as with selecting attacks, random variation and combination of good solutions may yield better solutions, so, again, a genetic algorithm is a good mechanism to select installation configurations. This assumes that all the installed UPFCs operate optimally with respect to the performance criteria, which will be covered in a subsequent section.

C. Iterative Hardening

Defending against a single attack alone does not provide any significant improvement in fault tolerance if there are other attacks of nearly equal complexity and damage. The real goal is to demonstrate that a few well placed UPFCs can substantially harden power grids. Ideally a few well placed UPFCs will substantially increase the system's robustness.

The attacker's genetic algorithm is designed to find the simplest significant faults, while the defender's genetic algorithm is designed to install the fewest number of devices necessary

to significantly harden the grid against those attacks. Since the attacker's choices will change based on the system configuration, these two algorithms need to be run in an iterative cycle to incrementally improve the system.

The attacker's algorithm will provide an adequate source of attacks for each potential system configuration, while the defender's algorithm will continually improve the system defenses in order to escalate the complexity of significant attacks. As such, it is expected that both the reward functions will need to be "cooled" as the two algorithms alternate back and forth to allow for increasingly complicated attacks and increasing numbers of FACTS installations. Fig. 1 shows a flow chart of the sequence in which the two algorithms will be used and Fig. 2 shows the basic data flow.

III. Simulation

An accurate simulation of the power system, represented by the function $F(\alpha, \beta)$ above, is vital in order to achieve meaningful results. To be useful, the simulation must:

1) Be fast enough for repeated evaluations needed by a GA
2) Be able to simulate line failures
3) Be able to simulate all twelve UPFC control modes as well as install and remove UPFCs
4) Provide a distinct means of comparing the quality of different attacks and different defenses
5) Be as accurate a model as an actual attacker would be likely to use

The first four criteria can be met via modification to a traditional power system technique known as Loadflow. The fifth criteria is subjective, however the form of simulation presented here is based on common analysis techniques used by power engineers to determine system faults and is one of the most likely starting places for a vulnerability assessment. Moreover, all the information required for this type of attack simulation would be readily available to a potential attacker.

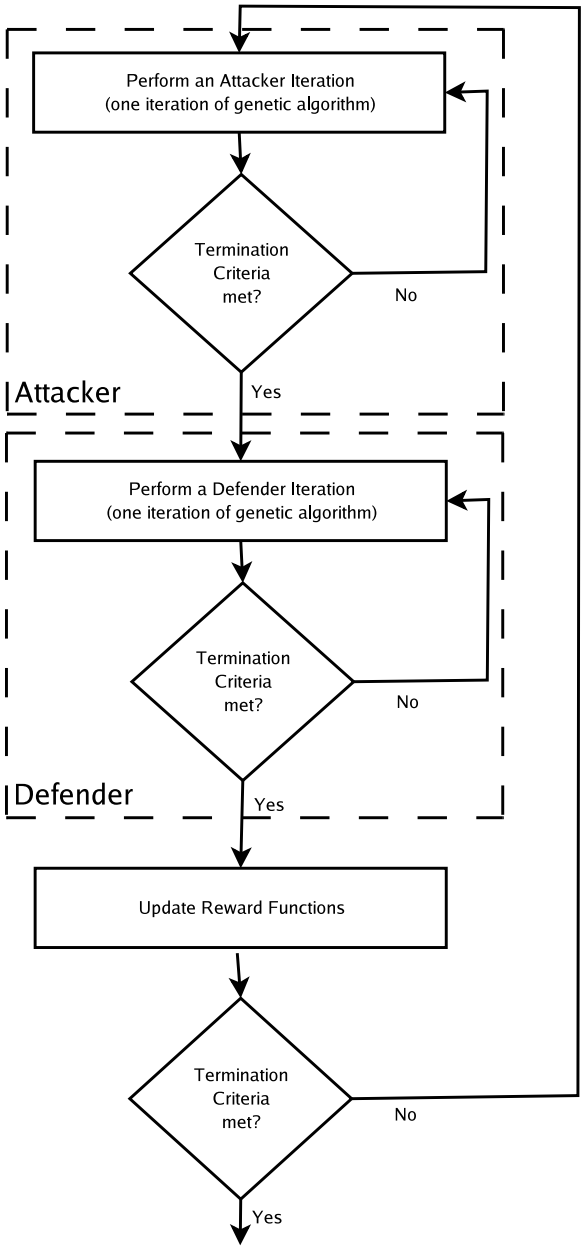


Fig. 1. Computational Flow Chart of Attacker and Defender Algorithms

A. Power System Steady-State Model

The most straight forward approach to steady-state power system simulation is a technique known as Loadflow. Loadflow models a power system as a collection of buses, which can be either a generator, a power customer, or both, and a set of power lines connecting the buses together. At each bus there are four state variables, and, depending on the specific combination of generators and power consumers at the bus, two of the state variables are

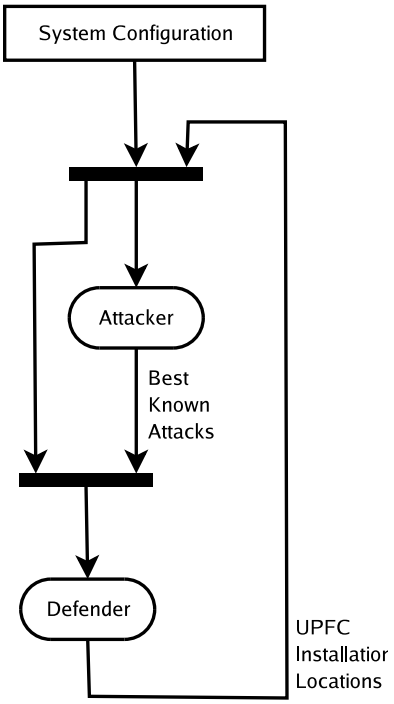


Fig. 2. Data Flow of the Attacker and Defender Algorithms

known and the other two are unknown. Loadflow is merely a technique that solves for the unknown state variables. The four state variables are:

P_j the real power load at bus j

$Q_{j,}$ the reactive power load at bus j

$v_{e,j}$ the real component of the voltage at bus j, $Re\{V_j\}$

$v_{h,j}$ the reactive component of voltage at bus j, $Imag\{V_j\}$

Note that the voltage is a sinusoidal signal and can be represented in polar form with a magnitude, $|V_j|$, and a phase angle (relative to a reference bus), $\angle V_j$. For the version of Loadflow used here the voltage is converted to rectangular form, $v_{e,j} + iv_{h,j}$. There are three types of buses in the system, and the type of bus indicates which variables are known and which are unknown:

Generator Buses Generator buses are directly connected to a large generator. It is assumed that the power, P_j, and voltage, v_j, at these buses is constant due to the generator. The reactive power supplied by the generator, Q_j, and the phase angle of the voltage, δ_j, are unknown. Note that there are practical limits on the amount of

reactive power a generator can supply, which are enforced by the simulation described here.

Load Buses Load buses represent the bulk of the buses in a system, which have a known real power load, P_j, and a known reactive power load, Q_j. Generally these represent the load being used by customers but may also represent power being injected into the system that can not be explicitly represented as a generator (discussed later). The voltage, v_j, and phase angle of the voltage, δ_j, are unknown.

Slack Bus The slack bus is a special generator in the system which is used: 1) as a reference against which all other phase angles are measured ($\delta_j = 0$ by definition), and 2) as a supply for additional real power to make up for system losses. At all other buses a known power is either injected or withdrawn, however the power lines themselves require power to operate. The slack bus represents a "free" source of real power (the slack) to make up for the power consumed by the transmission system itself, known as system losses.

Power systems are governed by Kirchhoff's power laws, which ensure that the sum of the power at a bus is zero. I.e., the power that enters the bus must also leave the bus. Kirchhoff's laws for an AC transmission grid can be represented as one set of equations for the real component of power and a second set for either the reactive component or voltage depending on the type of bus. All buses except the slack bus must have balanced real power:

$$P_j - v_{e,j} \sum_{k}^{buses} \left(g_{j,k} v_{e,k} - b_{j,k} v_{h,k}\right) + v_{h,j} \sum_{k}^{buses} \left(g_{j,k} v_{h,k} + b_{j,k} v_{e,k}\right) = 0 \tag{3}$$

and the load buses must also have balanced reactive power:

$$Q + v_{e,j} \sum_{k}^{buses} \left(g_{j,k} v_{h,k} + b_{j,k} v_{e,k}\right) + v_{h,j} \sum_{k}^{buses} \left(g_{j,k} v_{e,k} - b_{j,k} v_{h,k}\right) = 0 \tag{4}$$

while the generator buses have a constant voltage:

$$|V_j| - \left(v_{e,j}^2 + v_{h,j}^2\right) = 0 \tag{5}$$

where:

$g_{j,k}$ the conductance from bus j to bus k

$b_{j,k}$ the susceptance from bus j to bus k

Note that subscript e indicates the real component of a complex variable and the subscript h indicates the imaginary component of a complex variable. Equation 3 assures everything but the slack bus meets Kirchhoff's law for real power. Equation 4 ensures that load buses meet Kirchhoff's law for reactive power as well, while Equation 5 ensures that generators, which generate an unknown amount of reactive power and thus violate Equation 4, operate at a fixed voltage.

Since each bus has an equation for real power (P_j) and either an equation for reactive power (Q_j) or an equation for voltage magnitude ($|V_j|$), there are a total of $2N$ quadratic equations and $2N$ unknowns for a system with N buses. The system of equations is generally solved via the Newton-Raphson method, which uses the first-order Taylor series approximation of (3), (4), and (5) to iteratively update an estimate of the values of the unknown variables.

The Newton-Raphson method starts with an initial guess of state variables, which is either based on a prior known state or specified nominal values. The iterative process then updates these values until the error in the equalities of (3), (4), and (5) are within an appropriate error tolerance.

A basic Loadflow algorithm which relies on the Newton-Raphson technique is shown in Fig. 3. The Newton-Raphson technique is commonly used in power systems for a variety of reasons:

1) State variables are generally close to either a known or a nominal value, so it is easy to select an "initial guess" for state variables
2) The technique generally has quadratic convergence, and hence only requires a few iterations
3) The power flow equations are sinusoidal in nature and are well behaved with regard to minor perturbations

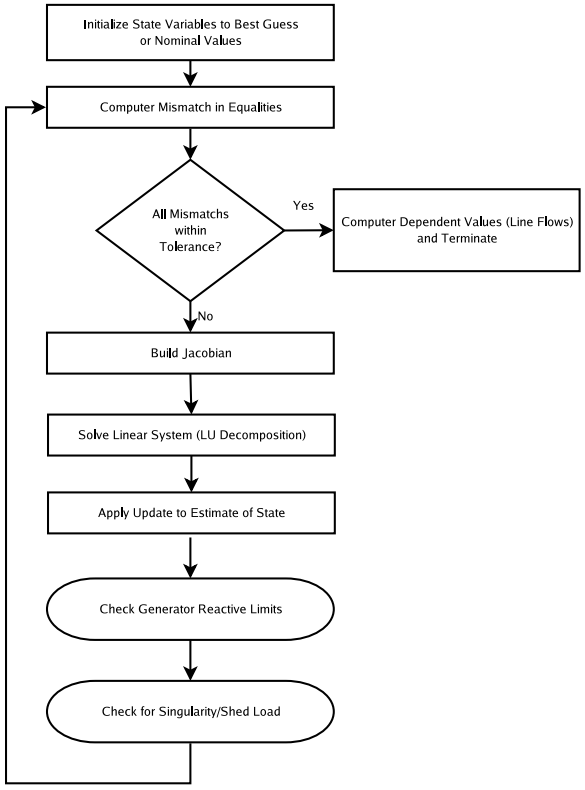


Fig. 3. The Newton-Raphson Loadflow Estimation Process

4) In the Newton-Raphson method, the power flow equations are a sparse, linear system and the underlying techniques, such as using LU decomposition, are computationally efficient

Generators produce both real power, which can be used for real work, and a form of oscillating power called reactive power. Reactive power is a vital component of AC power systems and may be either consumed or produced by the power lines themselves, as well as by generators or customers. As power lines fail, other lines begin to transfer the excess power and may require additional reactive power to do so. The generators in the system both produce or consume reactive power to ensure that the total reactive power in the system is balanced, however each generator has a limit on the amount of reactive power it can supply or absorb. Since reactive power demands change as the system loading changes during outages, it is vital to be able to honor the reactive generation limits of the system's generators. A common method to enforce these limits is to monitor the reactive power

each generator is supplying on each iteration of the Newton-Raphson loop. If a generator exceeds either the minimum or maximum reactive generation limit, the generator bus is changed to a load bus with P_j set to correspond to the power injected by the generator, Q_j set to correspond to the maximum amount of reactive power that the generator can absorb or consume depending on which limit was exceeded and the voltage. The voltage v_j then becomes an unknown variable.

B. Detecting and Enforcing Convergence

As lines are removed from the system, two significant problems may occur, either of which can prevent traditional Loadflow techniques from working: islanding and exceeding system capacity.

Islanding is where separate "islands" develop which effectively separate the system into multiple independent systems. Typically when this occurs at least one of the newly formed systems will be unable to meet the equality constraints. There are three possible cases: 1) islands that lack a swing bus and have no mechanism to compensate for the real-power losses in the lines, 2) islands that have load but no generation can not satisfy customer demand, and 3) islands that have generation but no load have a surplus of power with no consumers.

Islanding can be easily detected and corrected via graph traversal. A simple mechanism starts from an arbitrary bus and recursively visits all unvisited buses to which it is connected, marking each as visited. If, upon completion, any unvisited buses exist then the visited group represents a new island, and the process is repeated with the first unvisited bus. This process is repeated until all nodes are assigned to islands. When complete, islands with only generators or only loads are discarded. Any remaining islands that lack a slack bus are modified so that the largest generator in each becomes a slack bus.

Exceeding Capacity is when the system is not physically able to transmit power in a way that satisfies all the constraints (the power flow constraints of equations 3, 4, and 5 and the generator reactive limits).

In many cases the constraint equations cannot be met because the system no longer has the physical ability to carry enough power to satisfy the load being demanded. When this happens the original assumptions about the known variables are incorrect and no values of state variables can meet the constraint equations, so the original assumptions on load and generation must be changed to bring the system back to a solvable state. In systems losing transmission facilities the most common problem is having a load bus who's lines can not carry enough power to satisfy the specified demand, P_j. To bring the system back to a solvable state some of the load must be shed (reduce P_j). In the framework devised here, the attackers must assume, as in typical min-max game theory, that the defender will make optimal choices with the resources available. Thus ideally both the attackers and defenders will assume that only the minimum amount of load necessary will be shed to bring the system back to a solvable state.

A mechanism for optimal updates of the state variables, which can also be used to detect an ill-conditioned system, was proposed in [2]. The authors noticed that, when using the rectangular formulation of power flow as given previously, the complete Taylor series expansion only requires three terms. Moreover, these terms have a particularly efficient form and, most importantly, an exact solution can be found via the use of an appropriately chosen scalar multiplier. The optimal multiplier is easy to compute and provides a substantial improvement in system solvability.

In [3], Overbye notes that the solvable region of the state space is separated from the unsolvable region by a border on which the Jacobian used in the Newton-Raphson process becomes singular. When the system is solvable, the optimal multiplier remains near unity. Overbye also shows that infeasible systems can be detected by monitoring the magnitude of the optimal multiplier [3] . When it is sufficiently small, no state assignments will be able to satisfy the load demands of the system and load shedding must be performed.

In [4], an extension of [3], a technique was proposed to bring the system back to an optimal solvable point with a load shedding technique that maximizes the amount of demand

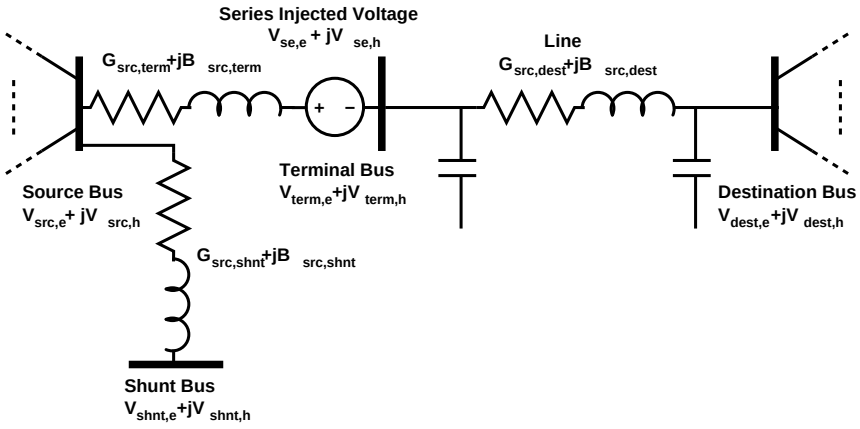


Fig. 4. UPFC Model

that can be met. This optimal load shedding relies on the use of the optimal multiplier technique to bring unsolvable systems back to the solvable boundary.

Although this optimal form of load shedding may not be in use on a given power system, it is unlikely that an attacker would know the exact load shedding capabilities and procedures, so they would assume a conservative case. By using the optimal load shedding, the attacker's mini-max perception, i.e. that the system will be as well defended as possible, is maintained.

C. UPFC Model

A perfect model of a UPFC consists of a voltage source connected to a bus in shunt and another voltage source connected in series with a line as seen in Fig. 4. The only constraints imposed on the model are the magnitude of the shunt voltage source, which is typically near the magnitude of the source bus, and that the real power injected or consumed by the series source must be supplied by the shunt source, which ensures there is no net real power injected into the system.

The typical UPFC model has twelve unique forms of control and typical Loadflow implementations assume that one particular mode will be used [8]. In the work here, the desired control mode is unknown and may change depending on the conditions of the system, so a simpler model is used in which the shunt voltage, series voltage, and series phase angle are specified directly. The shunt phase angle is left free to ensure that the shunt can meet the power consumption demands of the series source.

The UPFC model used here is novel in two respects: 1) it doesn't assume the control mode, allowing for the UPFC to change operating modes in different simulated scenarios to achieve optimal control for each, and 2) the rectangular coordinate system is used to comply with the optimal multiplier method, which is used to allow for optimal load shedding.

The optimal settings for the UPFCs can be found via simple optimization of a metric that will ensure maximal power delivery prior to failure. Prior work has shown that sequential quadratic programming is sufficient to directly find UPFC settings for a simpler model, however it is expected that the same technique will apply to this more general model [5].

D. Line Failure

Line failures, the prime component of cascading outages, occur because of excessive current overheating power lines, which eventually sag to the point that they either contact a ground source or a break. A simple line model has two parameters for each line: 1) a maximum current rating which it can safely carry and 2) a maximum ampacity, or cumulative current, that can be carried when the current rating is exceeded. The time until a line fails can be calculated based on the results of the Loadflow. For each line which is exceeding its current rating, the failure time is the amount of "remaining ampacity" divided by the current through the line. The "remaining ampacity" continually diminishes until either the line fails, or the line is no longer exceeding capacity and has had suitable time to "cool" to relieve the excess heat generated.

E. Simulation Overview

The full power system simulation overview can be seen in Fig. 5. As can be seen, there are a variety of nested loops for selecting optimal UPFC settings, removing naturally failing lines, and removing attacked lines. In addition, the Loadflow itself is a loop performing a significant amount of computation.

IV. Conclusions

By combining a simplistic UPFC model, which is amenable to optimal control, with a simulation that is capable of simulating the power system's state as components fail, a simple

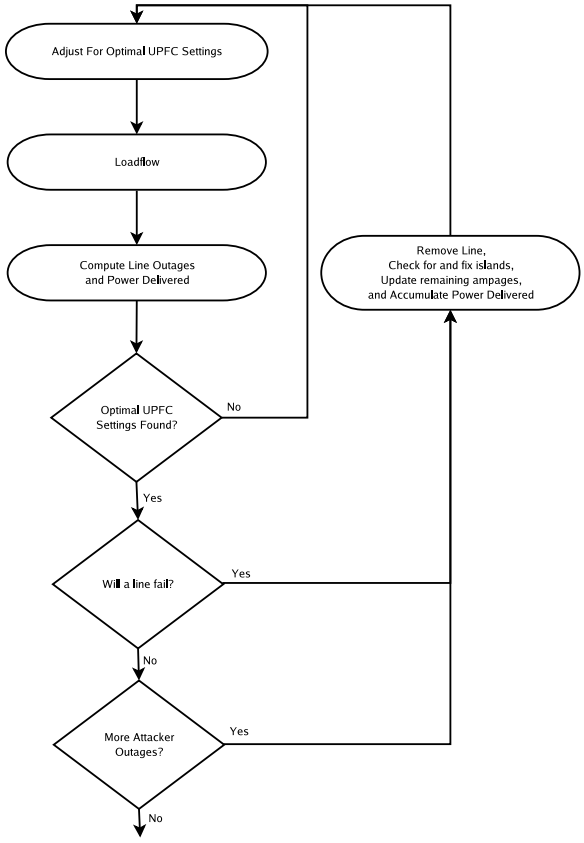


Fig. 5. Power System Simulation for Cascading Failures

line failure estimate, and genetic algorithms, it may be possible to provide a rudimentary plan for significantly improving the robustness of power grids. Genetic algorithms will use the simulation to both identify and repair weaknesses in the system. This combined approach uses the super set of functionality provided by a general model of the UPFC to identify and fix weaknesses.

REFERENCES

[1] V. M. Bier. *Game-Theoretic and Reliability Methods in Counter-Terrorism and Security*, chapter 3, pages 23–42. World Scientific Publishing Co, 2005.

[2] S. Iwamoto and Y. Tamura. A load flow calculation method for ill-conditioned power systems. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(4):1736–1743, April 1981.

[3] T.J. Overbye. A power flow measure for unsolvable cases. *IEEE Transactions on Power Systems*, 9(3):1359–1365, 1994.

[4] T.J. Overbye. Computation of a practical method to restore power flow solvability. *IEEE Transactions on Power Systems*, 10(1):280–287, 1995.

[5] William M. Siever, Daniel R. Tauritz, and Ann Miller. Improving grid fault tolerance by optimal control of facts devices. *Proceedings of Artificial Intelligence in Energy Systems and Power, 2006*, 1, February 2006.

[6] United States. Congress. House. Select Committee on Homeland Security. Subcommittee on Cybersecurity, Science, and Research and Development. *Implications of power blackouts for the nation's cybersecurity and critical infrastructure protection : joint hearing of the Subcommittee on Cybersecurity, Science, and Research and Development and the Subcommittee on Infrastructure and Border Security of the Select Committee on Homeland Security, House of Representatives, One Hundred Eighth Congress, first session, September 4, 2003 and September 23, 2003.* U.S. G.P.O., 2005.

[7] United States Department of Energy. National transmission grid study, May 2002.

[8] Xiao-Ping Zhang and K.R. Godfrey. Advanced unified power flow controller model for power system steady state control. In *Proceedings of the 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, 2004.*, volume 1, pages 228–233, 2004.

PAPER 4

Symbolic reduction for high-speed power simulation

W. M. Siever, D. R. Tauritz, *Member, IEEE,* A. Miller, *Senior Member, IEEE,* M. L. Crow, *Senior Member, IEEE,* B. M. McMillin, *Member, IEEE*, and S. Atcitty

***Abstract*— High speed simulations of power transmission systems, which often rely on solving linear systems of equations, are an increasingly important tool for training, testing equipment, on-line control, and situational awareness. Such simulations, however, suffer from two major problems: 1) they can be too computationally demanding to simulate large, complex systems within appropriate time constraints, and 2) they are difficult to develop and debug. Prior work has shown how computer algebra systems and symbolic computation can be used to help reduce both problems. In this paper, we 1) review common concepts in power system simulations, 2) summarize prior use of symbolic computation in power system simulation, 3) explore the advantages and disadvantages achieved via symbolic techniques, 4) extend the techniques to solve linear systems via *a priori* symbolic LU decomposition, and 5) demonstrate the advantages of symbolic techniques on a transient event simulation of the IEEE 118-bus test system, which runs in one tenth the time of an equivalent traditional (sparse matrix) approach.**

Supported in part by DOE/Sandia under contract number 291871, in part by NSF MRI award CNS-0420869 and CSR award CCF-0614633, and in part by the UMR Intelligent Systems Center. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

***Index Terms*— Power system transient stability, Symbol manipulation, Program compilers, Automatic programming, Software engineering, Real time systems, Nonlinear differential equations**

I. Introduction

A. High-Speed Power Simulation

Real-time and faster-than-real-time power system simulation is a vital tool for training, testing equipment, on-line control, and situational awareness:

- *Training*: As systems become more complex, local operations can have unforeseen consequences, therefore operators must be trained on accurate, real-time simulations of full-scale power systems [1], [2].
- *Testing Equipment*: Hardware-in-the-loop (HIL) testing, which interfaces real equipment to a simulation of a larger system, allows engineers to quickly evaluate equipment's real-time response to a variety of system operating conditions without the time or expense required of a large physical test bed. In order to evaluate new control technologies such as FACTS devices that are targeted at improving system-wide control, accurate, high-speed system simulations must be developed [3].
- *Situational Awareness*: Power system simulation is also a vital tool for state estimation and situational awareness. As the power transmission system has become increasingly overburdened, the need for complete knowledge of the state of the grid has become paramount. In fact, the U.S.-Canada Power System Outage Task force listed an "inadequate situational awareness" as one of the primary factors that lead to the 2003 North American blackout [4]. Such simulations need to operate faster than real time in order to properly predict the consequences of future events early enough to take preventative actions.

Achieving real-time simulations, however, suffer from two major problems: 1) they can be too computationally demanding to simulate large, complex systems within appropriate time constraints, and 2) they are difficult to develop and debug. Prior work has shown

how computer algebra systems and symbolic computation can be used to help reduce both problems [5]. In this paper, we extend the preliminary work of [5], with two significant new contributions:

1) the technique is extended to dynamic simulation, and
2) the LU decomposition is factored symbolically, which has limited applicability but yields a ten fold increase in run-time, making real-time simulation possible.

B. Dynamic Simulation Characteristics

Many forms of power system simulation share three distinct components: 1) modeling via differential equations, 2) numerical integration of differential equations, and 3) solving large, sparse systems.

Differential equations of the form:

$$\dot{x_t} = F(x_t, y_t) \tag{1}$$

are often used to model the dynamic components of a power system such as the rotating masses of synchronous machines [6]. In Equation (1), the rate of change of the state variable x at time t can be computed solely from the known state variables, x_t and y_t, at time t. Numerical integration is commonly applied to these equations to estimate the values of state variables at the next discrete step of the simulation:

$$x_{t+\Delta t} = x_t + \int_t^{t+\Delta t} F(x_t, y_t) \tag{2}$$

Algebraic manipulation can be used to convert equations like Equation (2) into an equality constraint:

$$\begin{aligned} H(x_{t+\Delta t}, y_{t+\Delta t}) &= x_{t+\Delta t} - x_t - \int_t^{t+\Delta t} F(x_t, y_t) \\ &= 0 \end{aligned} \tag{3}$$

Usually a numerical integration technique, such as trapezoidal integration, is used to find an algebraic approximation of the integral. If trapezoidal integration is used for the integral in Equation (3), the equality constraint can be expressed in terms of Equation (1):

$$\begin{aligned} H(x_{t+\Delta t}, y_{t+\Delta t}) &= x_{t+\Delta t} - x_t \\ &\quad -\frac{\Delta t}{2}\left(F(x_t, y_t) + F(x_{t+\Delta t}, y_{t+\Delta t})\right) \\ &= 0 \end{aligned}$$

In addition to the differential equations, there are often equality constraints of the form $G(x, y) = 0$. The most common equality constraints in power systems are the power flow equations that ensure that the power into and out of a bus are equal. The equality constraints on the next time step, $G(x_{t+\Delta t}, y_{t+\Delta t}) = 0$, can be combined with the equality constraints on the dynamic variables imposed by Equation (3), $H(x_{t+\Delta t}, y_{t+\Delta t}) = 0$, into a single system:

$$K(x_{t+\Delta t}, y_{t+\Delta t}) = \begin{bmatrix} H(x_{t+\Delta t}, y_{t+\Delta t}) \\ G(x_{t+\Delta t}, y_{t+\Delta t}) \end{bmatrix} = 0 \tag{4}$$

The resulting system consists of n equations, which are often non-linear, and n unknown variables. In power systems, the most common technique for solving these systems is the Newton-Raphson method, which is based on the first-order Taylor series of the system of equations [7]. Here, the first order Taylor series expansion is:

$$K(x_{t+\Delta t} + \Delta x_{t+\Delta t}, y_{t+\Delta t} + \Delta y_{t+\Delta t}) = K(x_{t+\Delta t}, y_{t+\Delta t}) + \frac{\partial K(x_{t+\Delta t}, y_{t+\Delta t})}{\partial \left[x_{t+\Delta t}, y_{t+\Delta t}\right]} \begin{bmatrix} \Delta x_{t+\Delta t} \\ \Delta y_{t+\Delta t} \end{bmatrix} = 0 \tag{5}$$

The Newton-Raphson method starts with an initial estimate (or guess) of the state in the next time step, $[x_{t+\Delta t}, y_{t+\Delta t}]^T$, and computes an error in the estimate, $[\Delta x_{t+\Delta t}, \Delta y_{t+\Delta t}]^T$,

via a rearrangement of Equation (5):

$$\underbrace{\begin{bmatrix} \Delta x_{t+\Delta t} \\ \Delta y_{t+\Delta t} \end{bmatrix}}_{State\,Error} = -\left[\underbrace{\frac{\partial K(x_{t+\Delta t}, y_{t+\Delta t})}{\partial\left[x_{t+\Delta t}, y_{t+\Delta t}\right]}}_{Jacobian}\right]^{-1} \underbrace{K(x_{t+\Delta t}, y_{t+\Delta t})}_{Mismatch} \tag{6}$$

Recall that each equation in the system is an equality constraint that will become zero when the correct state values are used. The "mismatch" is the current value of each of these equations and represents the error in the equality constraint. If any components of the mismatch exceed the error tolerance, the Jacobian, which is the matrix of the first derivative of each constraint with respect to each state variable, is computed. Next, the system is solved, typically via LU decomposition, to find the error in the state estimate, $[\Delta x_{t+\Delta t}, \Delta y_{t+\Delta t}]^T$. Then the error is added to the current state estimate:

$$\begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \end{bmatrix} = \begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \end{bmatrix} + \begin{bmatrix} \Delta x_{t+\Delta t} \\ \Delta y_{t+\Delta t} \end{bmatrix} \tag{7}$$

This entire process is repeated until either all components of the mismatch vector are within the error tolerance, the number of allowable iterations is exceeded, or an error condition, such as a singular matrix, occurs. Errors may occur if the initial guess is not close enough to the actual state values or if the system has no valid solution (no values of state variables will satisfy the constraints). In practice the process is usually limited to a few iterations and the condition number of the Jacobian is monitored. If the matrix is ill-conditioned or the maximum number of iterations is exceeded, it is assumed that the system is unsatisfiable.

One of the most important observations regarding Equation (4) in power systems is that the interconnections between buses are sparse. Therefore most of the equations only involve a few state variables and, consequently, the Jacobian is sparse. Exploiting this sparsity has been the key factor to improving the run-time performance of power system simulations.

C. Engineering Perspectives

Creation of a high-speed power system simulation requires expertise in both power engineering and software engineering. From the power engineer's perspective, the simulation's specifications consists of components that describe how the power system is being modeled:

- Dynamic models that are appropriate for the phenomena and time scales being studied
- Constraint equations
- Parameters for the equations, such as impedances
- Initial state values
- Integration techniques
- Mechanisms to estimate the values of the next state
- Maximum allowable numerical error in calculations

From the software engineer's perspective, the specification needs to describe when, how, and what needs to be done:

- Modules representing significant computational blocks and describing all computations
- Data flow description indicating module inputs, outputs, pre conditions, and post conditions
- Flow charts describing the order in which computational blocks are applied
- Timing requirements
- Error handling specifications and methods

D. Traditional Approaches

A traditional approach to a real-time simulation would be to create and test a single monolithic piece of software. To do this the software engineer needs to directly encode the power engineering knowledge into modules of the system. This either requires developers who are experts in both software engineering and power system modeling, or constant interaction between the experts in both domains. The former may be difficult to find and costly and the latter is fraught with potential miscommunication that can introduce time consuming errors. Ideally the two domains should be decoupled: the power engineer should

be free to change models, integration techniques, and parameters without significant impact on the software engineer and the software engineer should be free to modify techniques and improve computational speed without impacting the simulation results.

In order to meet the timing demands, the software engineer will make choices based on the computational "bottlenecks" in the simulation. The two most computationally demanding components of systems of this form are the construction of the Jacobian and solving the system in Equation (6). Due to the sparse nature of the system, the software engineer is likely to choose a sparse representation for the Jacobian matrix, which stores only non-zero values as well as an indicator of the corresponding row and column being used. For even moderately sparse systems, sparse storage can save considerable computer memory. Sparse techniques can save considerable compute time as well. For instance, a typical matrix multiplication of two $n \times n$ matrices using a non-sparse representation requires on the order of n^3 floating point operations, whereas a sparse representation requires at most mn operations, where m is the number of non-zero elements. Although the algorithms may not be practical, the best known theoretical values for each technique, which are $n^{2.38}$and $m^{0.7}n^{1.2} + n^{2+o(1)}$ respectively, still show significant computational advantage of the sparse representation for even moderately sparse systems [8].

For a non-sparse system, Jacobian construction cost is typically a factor of the number of elements in the matrix, n^2, while solving the system is on the order of $\frac{2n^3}{3} + 3n^2$(via LU decomposition). Based on this, it can be assumed that solving Equation (6) generally represents the most significant computational task for the simulation and, as such, is the most important consideration for meeting timing demands. Developing high speed solvers for sparse systems requires considerable expertise in both numerical methods and the specific computer architecture being used, so many simulations rely on outside packages, such as UMFPack, which is the sparse solver used in Matlab™, or SuperLU [9], [10]. UMFPack, which uses the Unsymmetrical-pattern MultiFrontal technique, is fast because it re-orders systems so as to maintain maximum sparsity without sacrificing numerical stability. Although UMFPack's specialty is solving non-symmetric systems, when the system

is highly symmetric and has non-zero diagonals, as is often the case in power system simulations, UMFPack resorts to a well known graph based algorithm, AMD (Approximate Minimum Degree), for arranging the matrix to minimize calculation [11], [9]. UMFPack solves a system by first analyzing the sparsity pattern to identify the best reordering, then it proceeds to perform a numeric factorization, and finally actually solves the system via LU decomposition. If numerical values change but the sparsity pattern remains fixed, only the last two steps need to be repeated to solve the new system.

Generally, the mechanism used to represent sparse matrices is based on the sparse solver being used. One of the most common techniques, and that used by UMFPack, Matlab™, and SuperLU, is the compressed column form. This technique stores elements by using three arrays. Two of the arrays, Ar and Av, are synchronized with each other: the contents of one indicates the row of a non-zero element and the contents of the other indicates its value. I.e., $Ar[3]$ indicates the row that corresponds to the value in $Av[3]$. The third array, Ac, is used to indicate where each successive column begins in Av and Ar. For instance, $Ac[3]$ gives the index into Av and Ar for the first element corresponding to the third column of the matrix, and $((Ac[4]) - 1)$ indicates the index of the last element of the third column. If an entire column is empty then two successive elements of Ac will have the same value. If there are m non-zero values in an $n \times n$ matrix, then both Ar and Av have $m \times 1$ elements and Ac has $n \times 1$ elements. Fig. 1 shows an example of a 3×3 sparse matrix and the corresponding vectors for Av, Ar, and Ac.

One of the hidden costs of sparse representation is the access time, which is the time required to find the element in the computer's memory (either to use in a computation or replace with a new value). Traditional dense matrix representations have an access time that is a factor of the number of dimensions. In power system simulations, where matrices are usually one or two dimensional corresponding to the number of lines at each bus, access time can be considered constant. In the case of the compressed column format, the average complexity is based on the average number of elements in a column. Pseudo-code to access a specific element in a compressed column sparse matrix is given in Algorithm 1. Since the

$$\begin{bmatrix} a_{1,1} & 0 & a_{1,3} \\ a_{2,1} & 0 & 0 \\ a_{3,1} & 0 & a_{3,3} \end{bmatrix}, \quad \begin{aligned} Av &= \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & a_{1,3} & a_{3,3} \end{bmatrix}^T \\ Ar &= \begin{bmatrix} 1 & 2 & 3 & 1 & 3 \end{bmatrix}^T \\ Ac &= \begin{bmatrix} 1 & 4 & 4 & 6 & \end{bmatrix}^T \end{aligned}$$

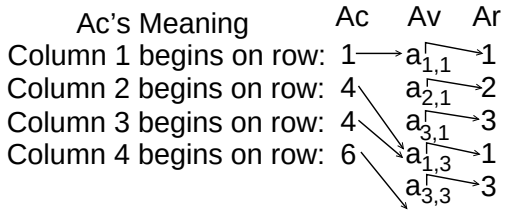


Fig. 1. Example of Compressed Column Storage of a 3×3 Matrix. Vector Av contains the values of the non-zeros and vector Ar contains the row index of the corresponding non-zero. Successive elements of vector Ac contain the index into Av and Ar for the corresponding column. I.e. $Ac[3]$ is the index into Av and Ar for the first non-zero element of the third column.

average column has $\frac{m}{n}$ non-zero entries, the complexity is on the order of $c \cdot log_2(\frac{m}{n})$ times *more* complicated than using an element from a standard two dimensional array, where c is a constant that takes into account the cost of the looping and conditional mechanisms in Algorithm 1. (Note: To make code more efficient it is common to "reuse" matrices rather than destroy and recreate them, however sparse matrix behavior is counterintuitive: although it depends on implementation, due to the access cost it is often more efficient to rebuild the sparse matrix rather than reuse it.)

II. Symbolic Computation and Code Generation

Symbolic computation has already been proposed as a means of improving both the design process and the run-time speed of power system simulations [5], [12]. In [12], the primary emphasis was on decoupling the system models from the generation of the simulation. In that work, a variety of symbolic processing programs were used to produce a simulation based on model specifications. The model specifications could easily be changed and a new simulation generated to conform to the new specifications.

In [5], as here, symbolic computation is used to reduce a system to its essential mathematical operations. The resulting set of equations was turned into an independent, high-speed simulation. In this method, the power engineer specifies the models of the system

Algorithm 1 Finding the storage location of an element in a compressed column sparse matrix

{Objective: Find the storage location of element *i*,*j*}
{I. Find the start and end of elements for column *j*}
$StartIndex \leftarrow Ac[j]$
$EndIndex \leftarrow Ac[j+1] - 1$
{II. Search for row *i* within the column's elements}
while $StartIndex \leq EndIndex$ **do**
  $Midpoint \leftarrow StartIndex + \lfloor \frac{EndIndex - StartIndex}{2} \rfloor$
  if $i > Ar[Midpoint]$ **then**
    $StartIndex \leftarrow Midpoint + 1$
  else if $i < Ar[Midpoint]$ **then**
    $EndIndex \leftarrow Midpoint - 1$
  else
    return $Av[Midpoint]$
  end if
end while
return $\emptyset$ {Failure: Return null}

as equations. The software engineer develops a program capable of factoring out these equations into their simplest form. For instance, assuming the goal is to solve a specific system, such as $b = Ax$, as fast as possible, where the matrix A and the vector x are known, one might simply write out and solve the equations:

$$\begin{aligned} b_1 &= a_{1,1}x_1 + \cdots + a_{1,n}x_n \\ &\vdots \\ b_n &= a_{n,1}x_1 + \cdots + a_{n,n}x_n \end{aligned} .$$

If a sparse system is being used, then only the simplest form of the equations need be used. For example, in a seven by seven system where it is known that $\{a_{1,1}, a_{1,4}, a_{1,5}, a_{1,6}, a_{1,7}\}$ are all zero, a simplified form for b_1 would be $a_{1,2}x_2 + a_{1,3}x_3$.

Another advantage of symbolic pre-processing is that all computations are specified symbolically, so the computer algebra system can compute derived values, such as derivatives, directly from the original equations prior to run time. Whereas, in the traditional approach the model equations are in one module, perhaps computing the mismatch, and the derivatives of these equations, which are derived by hand, are in another module. By using a single instance of model equations and computational techniques for computing derivatives, the

chance of human error is reduced. In addition, maintainability is improved because any modifications to the model automatically propagate to all equations derived from the model.

The advantages already observed in prior literature are summarized here:

- Computer algebra systems are more consistent and less prone to error than human computation, they are also more amenable to unit testing because they do not rely on power system specific knowledge
- Changes in fundamental equations propagate through to derived and dependent equations
- Indexes for array elements can be pre-computed, eliminating some run-time computations
- Sparse representations can be neglected, which significantly reduces overhead as will be explained in the following section
- Using a high level language to reduce the system equations may improve portability to new computer systems

III. Extension to Symbolic LU Decomposition

Prior work has used symbolic computation for both its software design advantages and its computational advantages, however, the solution of the linear system, which is typically the most computationally time consuming element, has been left to traditional sparse solvers such as UMFPack. Although it is of limited applicability, *a priori* symbolic LU decomposition is able to significantly improve the performance of simulations by both reducing the number of operations performed and by taking better advantage of performance enhancing features of modern processors such as instruction pipelines and caches.

The most significant computational speedups achieved by symbolic LU decomposition all stem from the elimination of both looping and conditional execution, which are used extensively by both the LU decomposition and the subsequent forward and backward substitution. To demonstrate the overhead required by these techniques, a simple, non-sparse version of the forward substitution process is given in Algorithm 2.

Algorithm 2 Pseudo-code for forward substitution

{Solve Ly=b for y}
Ensure: L is an $n \times n$ lower triangular matrix
Ensure: b is $n \times 1$ vector
$y[1] \leftarrow \frac{b[1]}{L[1,1]}$
{Solve for each y}
for $i = 1$ to n **do**
$sum \leftarrow 0$
for $j = 1$ to $i - 1$ **do**
$sum \leftarrow sum + \frac{L[i,j]}{y[j]}$
end for
$y[i] \leftarrow \frac{b[i]-sum}{L[i,i]}$
end for

In the symbolic approach, this system can be reduced to just the essential mathematical operations. For a 4×4 system, $Ly = b$, the equations are:

$$y[1] = b[1]/L[1,1]$$

$$y[2] = (b[2] - L[2,1] \cdot y[1])/L[2,2]$$

$$y[3] = (b[3] - L[3,1] \cdot y[1] - L[3,2] \cdot y[2])/L[3,3]$$

$$y[4] = (b[3] - L[4,1] \cdot y[0] - L[4,2] \cdot y[1] - L[4,3] \cdot y[2])/L[4,4]$$

An examination of this will show that there are four floating point divides, six floating point multiplies, and six floating point subtractions for a total of sixteen floating point operations. In addition, there are fourteen single dimensional index operations and ten two dimensional index operations. This is essentially the absolute minimum number of operations required for the computation. Using *a priori* symbolic reduction allows for the system equations to be represented in their simplest possible form, which eliminates the overhead of the looping mechanisms in Algorithm 2. This is particularly significant because for each multiply two additional operations are being performed: the index must be both incremented *and* the bounds checked.

Sparse representations can take advantage of the sparsity pattern of a particular row, but still have the additional overhead of a loop which must traverse the non-zero elements of the row and perform a boundary check, thus they still have a multiplicative increase in computational complexity over symbolically reduced system. For instance, the computational complexity for finding the value of a single row element, y, for a symbolically reduced system can be expressed as a function of the number of non-zero elements in that row, k. If the symbolic system requires $O(k)$ compute time, the sparse system will require $c \cdot O(k)$ compute time, where c is greater than 1.

In addition to the extra time required by indexing, the sparse representation fails to take full advantage of modern CPU pipelines. Modern CPUs feature a pipeline structure which tries to perform "speculative execution" by predicting which instructions should be executed next and executing them prior to their actual sequential turn. Since most programs consist of long sequences of instructions which are often independent of one another, this often yields significant performance improvements by essentially doing multiple operations nearly simultaneously. Sparse systems, however, often have short loops for traversing elements. Due to the intermittent sparsity of the matrix, in many cases the speculative execution mechanism will mis-predict whether the loop continues or terminates, which will cause much of the speculative execution effort to be discarded.

The symbolic approach, on the other hand, will require a significantly longer list of instructions and consequently a larger program. Thus, more instructions may need to be retrieved from memory, which may create a bottleneck. Most modern CPUs, however, are optimized for sequential execution and include burst memory transfers which help keep up with the CPU's execution speed.

Sparse techniques also require memory storage for not only the non-zero elements, but the row and column data as well (matrices Av, Ar, and Ac introduced earlier). This increases storage requirements and computation time due to the retrieval of row and column data. Symbolic approaches eliminate both the additional space and the time required to access the additional data.

Modern CPU architectures also include a hierarchy of memories, the lowest level of which are called caches. The caches represent a small, temporary repository for frequently used information. Modern manufacturing techniques can produce high-capacity ram very cheaply, but due to a variety of design considerations the main ram is considerably slower than the processors capacity to work with data. The cache memories are considerably more expensive per bit than the main ram, but operate much faster. Since most programs work repeatedly with a modest amount of data, having a small cache for this "highly utilized data" provides a tremendous speed boost with a small price increase. CPUs also include features that try to predict when the less frequently used data will be used in advance so it can be requested and returned from ram before it is actually needed. If the request is not early enough, then the CPU will "stall" while waiting for the data to arrive. Caches are often divided into two parts: one for the frequently used instructions and another for the frequently loaded data. Another advantage of the symbolic technique is that, due to its lesser data requirements, it has better utilization of the data cache. I.e. the cache can be nearly filled with actual numerical data rather than numerical data and indexing data required by sparse storage.

The most significant disadvantages of *a priori* factoring are its *a priori* computational overhead, its numerical insensitivity, and its inflexibility to topology changes. Symbolic manipulation requires more computational resources and time than the equivalent numeric operations, so the process of symbolic LU decomposition adds considerable time to the "compilation" process. In addition, once the system is factored it is assumed that the sparsity pattern will remain unchanged. If a pivotal element becomes zero during system simulation, the simulator may fail under a scenario that would be solvable with a dynamic LU solver. An additional consequence is that the topology of the simulated system is relatively inflexible. Although numeric values of the non-zero elements can be changed and some non-zero elements may even be allowed to go to zero, no new non-zero elements can be introduced. For many uses of high-speed simulation of power systems, these last two factors can be mitigated. In most uses of high-speed simulation some of the non-zero elements may

change their value (corresponding to a component failure), however no new non-zeros are introduced, which would correspond to new equipment being installed on the system. In addition, the intended course of the simulation and the expected failures are specified in advance, so it is possible to have different static LU solvers (produced by symbolic reduction) for each potential run-time configuration.

Symbolic LU Summary

A priori symbolic LU reduction has several distinct advantages for high-speed simulation:

1) indexing operations are pre-computed, requiring fewer run-time computations
2) the extra overhead of looping and boundary checking is eliminated, which significantly improved the processor's pipeline utilization
3) sparse element access time is reduced to a constant time
4) data memory requirements are reduced, which improves data cache utilization

and some disadvantages:

1) very limited capacity to change pivots to compensate for dynamic numerical conditions
2) a significant increase in code space is required and may create a memory bottleneck
3) considerable *a priori* computation is required to simplify the system
4) only limited changes in system topology can be made

IV. Example Application: Transient Simulation

Simulations utilizing differential algebraic systems, such as those presented in Section I-B, are commonly used to determine system response to transient events, such as a temporary ground fault. Generally, the goal is to determine if synchronism will be maintained following the transient event. When the simulation is started all state variables (bus voltage magnitudes and phase angles, generator rotor angles and frequencies) are in steady-state. At some time a transient event, or contingency occurs, and the state variables are estimated for each time-step, Δt. The simplest complete form of transient simulation, which will be used here, uses a classical model to represent the generators and a one-line (single phase) representation for the buses and lines of the transmission system [7], [6].

The classical generator model represents a voltage phasor, $\bar{E}_i = e_i \angle \delta_i$, connected via a reactance to a terminal bus which has a voltage phasor, $\bar{V}_j = v_j \angle \theta_j$. The terminal buses are connected by transmission lines with each other and with non-generating buses. The classical model of a synchronous generator is based on the angular acceleration of its rotor:

$$\dot{\omega}_i = \frac{1}{M_i}\Big(P_{m_i} - \frac{e_i v_i}{x'_{d_i}} \sin(\delta_i - \theta_j)\Big) \tag{8}$$

and the rate of change of its rotor angle with respect to the system frequency:

$$\dot{\delta}_i = \omega_i - \omega_s \tag{9}$$

Where:

i	The index of a specific generator
j	The index of the terminal bus associated with generator i
M_i	The inertia constant for generator i
P_{m_i}	The mechanical power for generator i
e_i	The internal voltage magnitude of generator i
x'_{d_i}	The transient reactance of generator i
v_j	The magnitude of the voltage at bus j, $\lvert\bar{V}_j\rvert$
ω_s	The synchronous rotor frequency ($377\frac{rad}{s}$ here)
ω_i	The rotor frequency of generator i
δ_l	the rotor angle of generator l, $\angle\bar{E}_l$
θ_l	the phase angle at bus l, $\angle\bar{V}_l$

Each bus has a voltage phasor, $\bar{V}_j$, as well as real and reactive loads. The power flow equations ensure that the net real and reactive power flow out of each bus, including the load on the bus itself, is zero. I.e.:

$$0 = P_j - v_j \sum_{k}^{buses} v_k y_{jk} \cos(\theta_j - \theta_k - \phi_{jk}) \tag{10}$$

and:

$$0 = Q_j - v_j \sum_{k}^{buses} v_k y_{jk} \sin(\theta_j - \theta_k - \phi_{jk}) \tag{11}$$

where:

P_j the real load at bus j

Q_j the reactive load at bus j

y_{jk} the magnitude of the $(j,k)^{th}$ element of the bus admittance matrix

ϕ_{jk} the angle of the $(j,k)^{th}$ element of the bus admittance matrix

$buses$ the set of all non-generator buses

The generator frequencies and rotor angles, $[\omega, \delta]$, represent the dynamic state components (x_t) of the system whereas the bus values, $[v, \theta]$, represent the instantaneously changing components of the system (y_t). The generator equations (Equations (8) and (9)) provide the differential updates $(\dot{x_t} = F(x_t, y_t))$, and the power flow equations (Equations (10) and (11)) provide the algebraic constraints on the system $(G(x_t, y_t) = 0)$.

A simplified flowchart of the simulation process can be seen in Fig. 2. Note that this is only the basic flow of the significant computations. It lacks error checking and application of the transient events.

IEEE 118-bus Test System

One of the most commonly used systems for power system studies is the IEEE 118-bus test system[1]. The 118-bus system is used here for comparison purposes because it is a realistic topology, of moderate size, and well studied. The 118-bus system has 118 buses and 20 generators, resulting in 276 state variables (V and θ at each bus and δ and ω at each generator) and consequently a Jacobian that is 276×276.

Four different variations on transient event simulations were developed and then analyzed to compare the advantages and trade-offs of symbolic vs. sparse representation. In the first version symbolic decomposition was used for all operations (Jacobian construction and LU decomposition). In the second version symbolic decomposition was used for Jacobian

[1]http://www.ee.washington.edu/research/pstca/

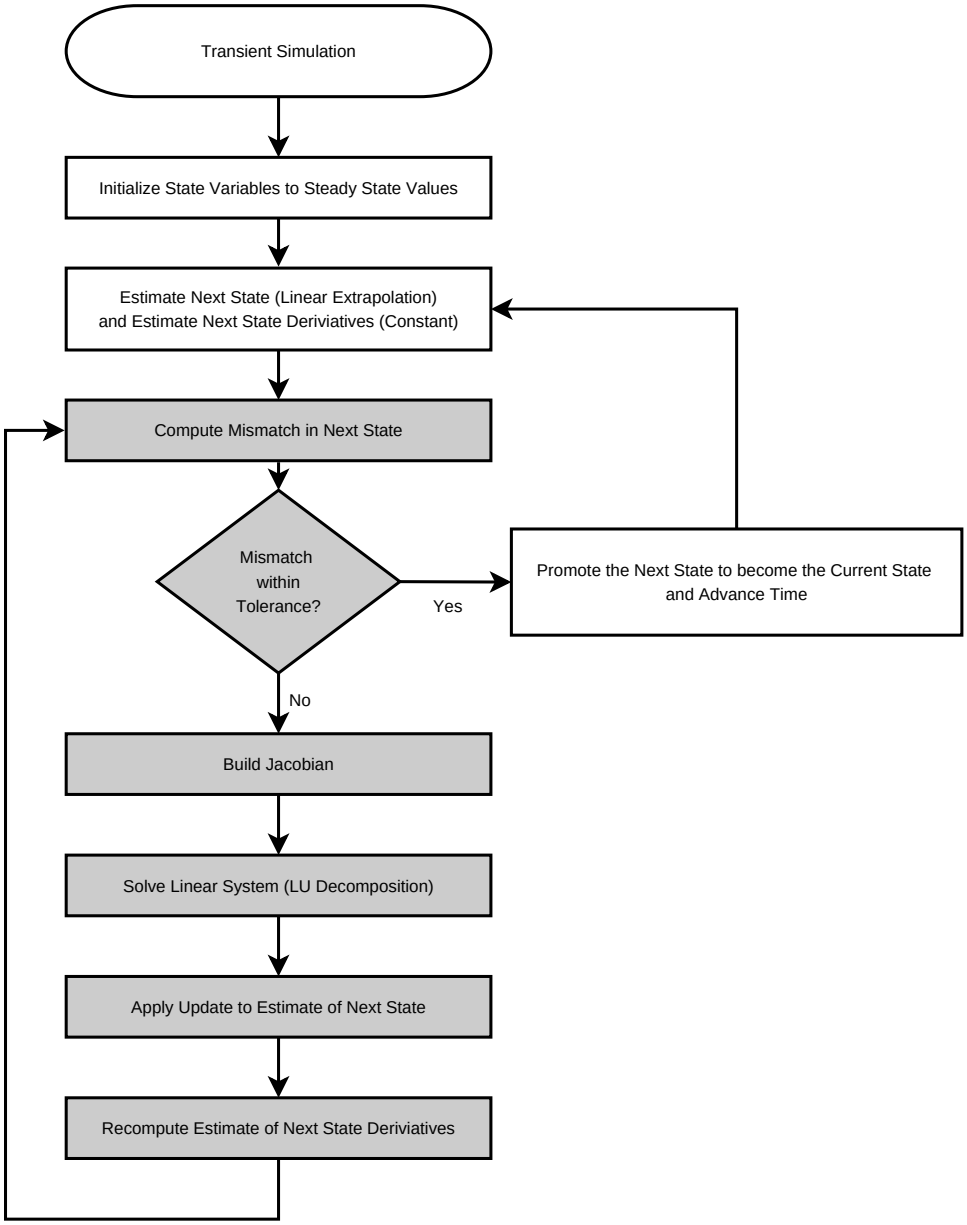


Fig. 2. Basic Transient Simulation Flow. The highlighted portion is the Newton-Raphson loop.

construction and UMFPack was used for dynamic LU decomposition. The third and fourth version both use the traditional sparse matrix approach for representing the Jacobian and LU decomposition. The third version uses UMFPack for LU decomposition while the fourth version uses SuperLU.

In order to compare the computational effort used by all four variants, Intel's VTune™ performance analyzer was used to measure resource utilization through a single iteration

of the Newton-Raphson loop, which is highlighted in Fig. 2. All four versions performed the same computational work: they started from the same state, performed essentially the same computations, and found state updates that were identical within the computer's precision; the only distinctions were the degree of *a priori* symbolic processing and the LU solution technique. The results of several interesting metrics are provided in Table I. Because individual raw results would vary from processor to processor and implementation to implementation, raw data, which is from an 1.6GHz Intel Duo Core processor, is provided only for the fully symbolic variant. All other data is normalized with respect to the fully symbolic version to contrast the computational differences of each approach.

The data in the table provides corroboration of all of the computational advantages of symbolic computation presented in Sections II and III:

- Symbolic LU decomposition provides the most significant speedup in execution time (over ten times faster than merely using symbolic decomposition for Jacobian construction). This is due primarily to eliminating overhead of indexing and bounds checking of sparse matrices
- Symbolic LU construction uses 55% fewer floating point operations than the sparse implementation and only 5.2% the total number of operations
- All the versions using sparse representation access data memory more than twenty times as much as the fully symbolic version
- The symbolic version, which used fewer instructions overall, consequently required fewer instruction fetches. However there was little re-use of instructions compared to the sparse approaches, so instruction cache utilization did not show a significant improvement
- For the system being used, the fully symbolic version had a small enough data footprint to almost completely fit in cache, reducing the penalty of data access
- Pipeline utilization is very high for the fully symbolic approach, indicating that all units of the processor are being fully utilized. The sparse techniques rely on short loops and

TABLE I

PERFORMANCE COMPARISON AMONG DECREASING LEVELS OF SYMBOLIC DECOMPOSITION FOR A SINGLE ITERATION OF THE NEWTON-RAPHSON METHOD ON A TRANSIENT SIMULATION OF THE IEEE 118 BUS TEST SYSTEM.

	Raw Data	Normalized		
Jacobian	Symbolic	Symbolic	Sparse	Sparse
LU	Symbolic	UMFPack	UMFPack	SuperLU
Run Time	0.325ms	11.1×	14.3×	23.2s×
Floating Point μOps Executed	125000 Insts	1.80×	2.68×	20.8×
Non-Floating Point μOps Executed	195000 Insts	19.1×	38.7×	97.9×
Instruction Fetches	179000 Fetches	16.2×	28.9×	52.5×
Data Memory References	98600 Refs	23.0×	44.3×	126×
Cache Instruction Misses	2000 Misses	1.22×	1.06×	0.560×
Cache Data Misses (Reads and Writes)	60 Misses	6.58×	11.8×	16.8×
Pipeline Efficiency	99.5%	0.793×	0.866×	0.692×

conditionals which depend on the sparsity pattern of the matrix, they under utilize the pipeline due to branch mis-predictions

Symbolic decomposition also achieves better optimization for the target computer architecture. It would be both difficult and time consuming to develop a symbolic reduction process that directly produces an executable. Instead, the decomposition process produces data in an intermediate format, the C language in this case. A traditional compiler was used to optimize the minimized set of equations for the target architecture. Fig. 3 shows an example of how the compiler has taken specific measures to optimize a set of equations for the underlying architecture. In this case the compiler made three significant optimizations: 1) it identified commonly used sub-expressions (I.e., $\sin\left(\theta_9 - \delta_{t+\Delta t,9} + \frac{\pi}{2}\right)$ in the Fig. 3) and only computed them once (a traditional approach would compute each multiple times), 2) where both the sine and cosine of an expression were used, a single function was called once to obtain both values, and 3) computation and I/O were interleaved in order to maximize processor utilization.

V. Conclusions

The basic ideas here are largely a continuation of [5], however two significant extensions are presented here: 1) the technique is extended to dynamic simulation, and 2) here even the LU decomposition is factored symbolically, which has limited applicability but yields a ten fold increase in run-time. The speed advantages of symbolic LU decomposition are primarily due to a reduction in the number of operations actually required for processing resulting from eliminating the loops of sparse representations, the reduction in memory access due to the elimination of the additional indexing of sparse matrices, better utilization of compiler optimization for the underlying architecture, and better utilization of the processor's pipeline.

References

[1] P. Forsyth, R. Kuffel, R. Wierckx, J.-B. Choo, Y.-B. Yoon, and T.-K. Kim, "Comparison of transient stability analysis and large-scale real time digital simulation," in *Power Tech Proceedings, 2001 IEEE Porto*, vol. 4, Porto, 2001.

```
                                  ; Generated Source Code:
                                  ;   j38 = −16.45 · V9 · E9 · sin(θ9 − δt+Δt,9 + π/2)
movsd   _2il0floatpacket.2 ,%xmm0 ; xmm0=π/2
addsd   −24256(%ebp),%xmm0        ; xmm0=π/2 + θ9
 ...                              ; interleaved instructions working on other
                                  ; operations
subsd   −24992(%ebp),%xmm0        ; xmm0=(π/2 + θ9) − δt+Δt
 call    __libm_sse2_sincos       ; The compiler identified that sine is used
                                  ; in j38 and cosine is used for j39.
                                  ; Both are computed simultaneously
                                  ; (xmm0=sin(... ), xmm1=cos(...))
movapd %xmm1,%xmm5                ; xmm5=cos(...)
                                  ; Generated Source Code:
                                  ;   j39 = 16.45 · V9 · E9 · cos(θ9 − δt+Δt,9 + pi/2)
movsd   %xmm0,−24216(%ebp)        ; Save sin( ... ) for use in j213 and j1189
 ...
movsd   %xmm5,−24208(%ebp)        ; Save cos( ... ) for use in j1185
 ...
movsd   %xmm4,304+j               ; Save the computed value of j38
 ...
movsd   %xmm2,312+j               ; Save the computed value of j39
```

Fig. 3. A Fragment of the Assembly Language for Construction of Two of the Elements of the Jacobian (j_{38} and j_{39}). The compiler has optimized for the target platform in a number of ways: using SSE2 instructions, only computing common sub-expressions once, efficient use of a single function for both sine and cosine, and interleaving of operations to balance memory access and compute time.

[2] K. Anupindi, A. Skjellum, P. Coddington, and G. Fox, "Parallel differential-algebraic equation solvers for power system transient stability analysis," in *Scalable Parallel Libraries Conference, 1993., Proceedings of the*, Mississippi State, MS, 1993, pp. 240–244.

[3] W. Ren, M. Steurer, and S. Woodruff, "Progress and challenges in real time hardware-in-the loop simulations of integrated ship power systems," in *Power Engineering Society General Meeting, 2005. IEEE*, June 2005, pp. 534–537.

[4] U.S.-Canada Power System Outage Task Force, "Final report on the August 14th blackout in the United States and Canada: Causes and recommendations," April 2004. [Online]. Available: https://reports.energy.gov/

[5] R. Bacher, "Computer aided power flow software engineering and code generation," *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 490–496, 1996.

[6] P. Anderson and A. Fouad, *Power System Control and Stability*. IEEE press, 1994.

[7] M. Crow, *Computational Methods for Electric Power System*. CRC Press, 2003.

[8] R. Yuster and U. Zwick, "Fast sparse matrix multiplication," *ACM Trans. Algorithms*, vol. 1, no. 1, pp. 2–13, 2005.

[9] T. A. Davis, "Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.*, vol. 30, no. 2, pp. 196–199, 2004.

[10] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, "A supernodal approach to sparse partial pivoting," *SIAM J. Matrix Analysis and Applications*, vol. 20, no. 3, pp. 720–755, 1999.

[11] P. R. Amestoy, Enseeiht-Irit, T. A. Davis, and I. S. Duff, "Algorithm 837: AMD, an approximate minimum degree ordering algorithm," *ACM Trans. Math. Softw.*, vol. 30, no. 3, pp. 381–388, 2004.

[12] F. Alvarado and Y. Liu, "General purpose symbolic simulation tools for electric networks," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 689–697, 1988.

W. M. Siever received his B.S. and M.S. in computer science in 1997 and 2000, respectively, from the University of Missouri - Rolla, where he is currently pursuing his Ph.D. in computer engineering. His interests include computer architecture, artificial intelligence, and common sense engineering.

D. R. Tauritz D. R. Tauritz is an Assistant Professor of Computer Science at the University of Missouri-Rolla. He received M.S. and Ph.D. degrees in Computer Science from Leiden University in the Netherlands in 1996 and 2002, respectively. His research and teaching interests include Evolutionary Algorithms, Artificial Intelligence, and computational methods for Critical Infrastructure Protection.

A. Miller is the Cynthia Tang Missouri Distinguished Professor of Computer Engineering at the University of Missouri, Rolla. Her technical interests include security and reliability of large scale networked systems. Miller has a Ph.D. in mathematics from Saint Louis University. She is a senior member of the IEEE and the IEEE Computer, Communications, and Reliability Societies.

M. L. Crow received her BSE degree from the University of Michigan, and her Ph.D. degree from the University of Illinois. She is presently the Dean of the School of Materials, Energy, and Earth Resources and is the F. Finley Distinguished Professor of Electrical Engineering

at the University of Missouri-Rolla. Her area of research interests have concentrated on developing computational methods for dynamic security assessment and the application of power electronics in bulk power systems.

B. M. McMillin is a Professor of Computer Science at the University of Missouri-Rolla. He received the Ph.D. in Computer Science from Michigan State University in 1988 and the BSEE and MSCS from Michigan Technological University in 1979 and 1985, respectively. His research focuses on Fault Tolerance, Distributed Computing, Formal Methods and Security for critical applications. He is a member of IEEE.

S. Atcitty received the B.S. and M.S. degrees in electrical engineering from New Mexico State University, Las Cruces. Currently, he is a Power Electronics Consultant with Energy Storage System Program at Sandia National Laboratories, Albuquerque, NM. His research interests include high-power electronics and applications in power systems.

SECTION

2. IMPACT AND FUTURE WORK

The four papers included contain a variety of components necessary for the further study and utilization of FACTS technology. Not only has an estimate of UPFC impact been presented, but a full plan to study their potential in steady-state control has been developed. In addition, a high speed power system simulator is provided which can substantially improve the effectiveness of the full study.

2.1. UPFCS

Future work in power system studies includes actual implementation of the simulation proposed and evaluation of its effectiveness. If it is found to be effective and provide guidelines for better utilization of FACTS technology, the basic simulation can be updated to include real topologies and more realistic line failure models.

2.2. HIGH-SPEED SIMULATION

The fundamental ideas behind high-speed power simulation via symbolic reduction can be applied to a number of research problems and expanded to provide a more comprehensive means of developing simulations.

First and foremost, a more comprehensive study of the use of static LU decomposition should be performed. Simulation work often relies on accurate solutions to linear systems that may become brittle with the form of static decomposition presented. Further investigation can determine a means of specifying which elements or combinations of elements should be allowed to be removed without impacting the numerical stability of the solution.

Parallelization is another significant area of extension. The technique can easliy be used to divide computation into fine grained, parallizable units. Such fine grain parallelization does not work well with many traditional parallel and distributed computational models, but may be ideal for recent delvelopments in general computation engines for high-end graphic and

physics processors. These processors typically utilize many parallel computational pipelines which could be used to provide siginificant computational speedup while using commercial, off-the-shelf parts.

VITA

William Michael Siever was born in Canton, Illinois on March 19, 1974. He was raised in the rural community of Havana, Illinois where he attended primary, middle, and high school. Upon graduation from high school, he enrolled in the University of Missouri-Rolla (UMR) to study computer science, for which he was awarded a Bachelor of Science in 1997. Following his B.S., he decided to pursue an M.S. in Computer Science at UMR, which was awarded in 2000. His Master's studies were primarily focused on computer engineering and artificial intelligence, with specific application to his thesis topic, robotic soccer. Upon completion of his M.S. in 2000, he chose to pursue a Ph.D. in Computer Engineering, which was awarded in May 2007. During the course of both his Ph.D. and his M.S. he was fortunate enough to be on teams that were finalists in both international programming competitions and hardware design competitions. He also had the opportunity to teach and assist with a number of courses, including computer organization and an introductory artificial intelligence course. His Ph.D. work, which has applications in system simulation and critical infrastructure protection, is a departure from his M.S. work.