Controlling Power Flow Using FACTS Devices and the Maximum Flow Algorithm

A. Armbruster B. McMillin
Department of Computer Science

M. L. Crow
Electrical and Computer Engineering
University of Missouri-Rolla, Rolla, MO 65409-0350

Abstract— **The bulk power system is the largest man-made network, and its sheer size makes controlling it an extremely difficult task. In this paper, a method to control the network using FACTS devices set to levels determined by a maximum flow (max-flow) algorithm from graph theory is described. The maximum flow algorithm is introduced and shown how it applies to the power system for FACTS device placement and scheduling. Finally, the appropriateness of the maximum flow algorithm for power flow control is discussed.**

KEYWORDS. *FACTS, Maximum Flow, Distributed Algorithms*

I. INTRODUCTION

The bulk power system grid is arguably the largest man-made interconnected network in existence. The sheer size of the power network makes control of the grid an extremely difficult task. Cascading failures are the most severe form of contingency that can occur in a power system. If not immediately mitigated, a contingency may lead to a second, a third, and so on until load must be shed to stabilize the system. A typical cascading failure involves transmission line overloads. In a system where there is a large directional power flow from one region to another, the loss of a single transmission line can instigate a cascading failure. If the first transmission line is lost, the power it carried must be shunted across the remaining parallel transmission lines. The shunted power may cause one or more of these transmission lines to overload. At some point, it is no longer possible to serve the load with existing network topology. If load is not shed in a timely manner, generators may trip as a result of under-frequency protective relays.

Distributed "flexible AC transmission system" (FACTS) controllers in the network can help alleviate the overload problem by directing extraneous power flow away from highly loaded lines. Grid control has historically been decentralized due to geographic and regulatory constraints. With the expansion of communication technologies, including optical and wireless communication, it is rapidly becoming possible to incorporate real-time, distributed, decision-making processes over a widespread area using a variety of information. Within this framework, controllers will be able to broadcast and receive operating status information from other entities throughout the system. Each entity must make computational decisions locally using state information obtained by messages from other processes.

The family of FACTS devices holds considerable promise as network-embedded controllers. The problem then becomes one for the identifying the sizing and placement of these controllers, and how to coordinate their action. In this paper, the FACTS devices execute a graph-theory-based maximum flow distributed algorithm to identify critical transmission corridors and adjust power flow to avoid cascading failures.

This paper describes the problem of where to place the FACTS devices to achieve maximum power flow performance and how to coordinate the actions of each device to prevent unproductive interactions. In section 2, the maximum flow algorithm is described as it applies to the power grid. Section 3 is a discussion of the placement of the FACTS devices and the appropriateness of the max-flow algorithm for the power flow. Section 4 discusses the conclusions and future directions for this work.

II. MAXIMUM FLOW AND THE POWER GRID

This section will give an introduction to the maximum flow algorithm, relate it to the power gird, and then discuss specifically how the algorithm is used to place and coordinate FACTS devices.

A. Introduction to Maximum Flow

The power flow in a power grid is modeled as a directed graph flow problem with a directed graph $G(N, A)$ modeling the power network. The set of nodes, N, corresponds to the busbars (or simply buses) of the power network. The

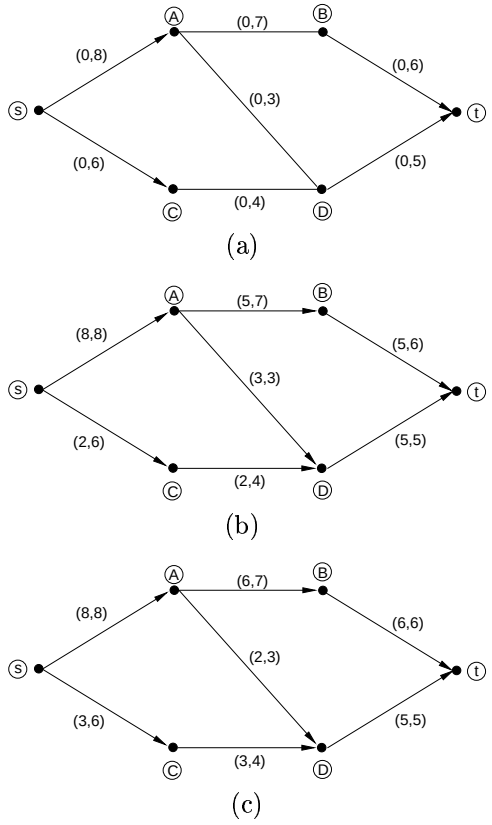

Figure 1 – Example Graph. Part (a) shows the showing each line with zero flow and variable capacity of x, $(0, x)$. Part (b) shows the graph after using forward labelings. Part (c) shows the network after the algorithm completed.

power flow between buses $n_i, n_j \in N$ is represented by an arc $a_{ij} \in A$. Each arc is assigned a weight, u_{ij}, denoting the maximum allowable power flow through that line, and subsequently, over the arc in the network. Initially, the weights are set as the steady state flow of an unconstrained power grid. A simple example of such a graph is shown in Figure 1 (a). Each arc pair gives (*assigned flow*, *maximum flow*), therefore the (a) graph shows that each arc has 0 units of assigned flow on each arc and is assigned a weight, or capacity. For the basic max-flow algorithm there are two special nodes, s and t, representing the source and termination (or sink), respectively. The figure does not show the directions of the arcs $A - B$, $A - D$, and $C - D$. Those directions were omitted because flow can theoretically travel in either direction. Those undirected arcs represent two directed arcs, one in each direction. When there is flow on one of the undirected arcs, a direction will be added to indicate the direction of the flow.

For simplicity, the algorithm of [1] will be used to find the maximum flow. The algorithm is shown in Figure 2. The algorithm works by successively assigning flow $f(a_{ij})$ to arcs along a directed path from s to t until no more flow can be added. For the first augmenting flow, the arcs along the path $s - A - D - t$ are labeled using 3 units of flow. Augmenting flow from $s - A$ again results in finding a path

Definitions:

- Forward Labeling of a_{ij}:
  If n_i is labeled and n_j is not and $u_{ij} > f(a_{ij})$, n_j gets labeled and $\Delta_{ij} = u_{ij} - f(a_{ij})$
- Backward Labeling of a_{ij}:
  If n_i is labeled and n_j is not and $f(a_{ij}) > 0$, n_j gets labeled and $\Delta_{ij} = f(a_{ij})$

Algorithm:

1. Assign an initial flow ($f(a_{ij}) = 0$ for all arcs in A)
2. Mark s labeled and all other nodes unlabeled.
3. Search for a node that can be labeled by either a forward or backward edge. If none found, flow is maximum, stop. If $n_j = t$ go to step 4, otherwise, repeat this step.
4. Backtrack the path computing the minimum Δ_{ij} used. If a_{ij} used a forward labeling, $f(a_{ij}) \leftarrow f(a_{ij}) + \Delta_{ij}$. If a_{ij} used a backward labeling, $f(a_{ij}) \leftarrow f(a_{ij}) - \Delta_{ij}$. Go to step 2.

Figure 2 – Ford and Fulkerson Algorithm for Max-Flow

along $s - A - B - t$, which uses the remaining 5 units of flow from $s - A$. In the next augmentation, the edge from s to A is at capacity, so a new path is searched along s to C. The flow is augmented on the path $s - C - D - t$, which accomodates 2 units of flow. The resultant graph is shown in Figure 1 (b). Searching for a path from $s - C$, power flow can be sent on $C - D$, but $D - t$ is at capacity and cannot accept more flow, thus the edge D to A must be searched, using a Backwards Labeling. The path continues along $A - B - t$. After the flows have been updated, the algorithm terminates because there are no more available paths to be found to the sink node. The final graph is in Figure 1 (c). By summing the arcs out of the source or into the sink, the maximum flow can be calculated, which in this case is eleven.

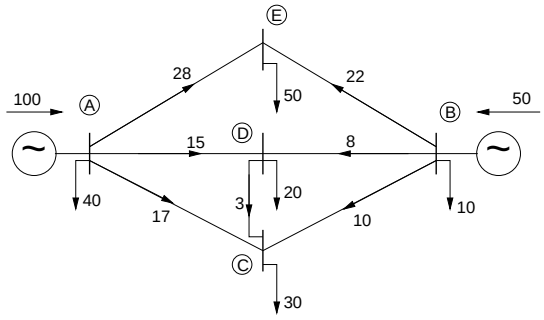

Figure 3 – Example power system network

B. *Modeling Power Flow Using Max-Flow*

As previously described, the power grid is modeled as a directed graph $G(N, A)$. An example of a power network modeled in this way is shown in Figure 3. The power system's active power loads are shown as arrows from each

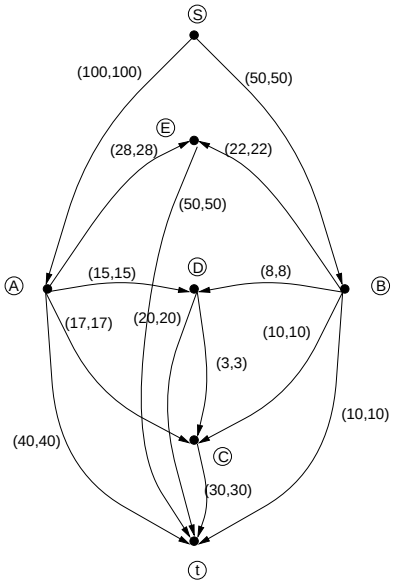

Figure 4 – Power network directed flow graph

bus, and the directed active power flows from steady state generation are shown for each transmission line. This power system may be equivalently represented as a directed graph as shown in Figure 4. Due to the power flow equations, each node in the graph must also satisfy $flow_{in} = flow_{out}$, except for the source and sink nodes. Note that all power flows from the source (s) to the termination (t). The termination may be considered the network "ground" node and the source is a "virtual source" node connecting all the power generators. The nodes s and t, together with G, form a graph $G'(N', A')$.

This model is useful for the case when a line is lost due to a failure, and the resulting power flow stresses the network. Too much load is drawn over lines of inadequate capacity and one-by-one the lines overload and trip off-line. A method is needed to rapidly re-balance the power by either re-directing the power flow across transmission corridors with greater capacity, or by shedding load to maintain power flow to the remaining loads. The solution to this problem can be approached by modeling it as determining the maximum flow in a directed graph [2].

C. *The Distributed Algorithm*

While the max-flow algorithm can be run from a central site, there are several problems with this centralized approach. The first is that communication with the central site may be compromised due to hardware failures or intrusions (hacking) into the communications network. The central site may also be subject to failure.

Since the power grid problem is inherently distributed, the FACTS devices can coordinate themselves by each running a distributed algorithm. Decentralized or parallel algorithms for the solution of max-flow have been proposed by many researchers [3].

Distributed Max-Flow Algorithm:

Assign an initial flow ($f(a_{ij}) = 0$ for all arcs in A)

for each Generator s_i**,** do in parallel

1. Mark s_i labeled.
2. Select an unlabeled outgoing arc n_j. If there are no more unlabeled outgoing arcs, stop.
3. Label node s_i and the arc with (s_i, Δ_{ij}).
4. Explore the arc a_{s_ij} with a probe(s_i, Δ_{ij}).
5. Wait for messages to return. If the return message is
   - path(s_i, Δ_{ij}), set $f(a_{s_ij}) \leftarrow f(a_{s_ij}) + \Delta_{ij}$
   - blocked(s_i): All paths to t are blocked by other searches. Unlabel the arc a_{s_ij}. Go to step 2.
   - echo(s_i): All paths to t along a_{s_ij} are filled to capacity. Go to step 2.

for each node, n_l

1. wait for a probe(s_i, Δ_{kl})
2. If the node is already labeled by s_i, return echo(s_i) to the sender of probe. Go to step 1
3. If the node is already labeled (by some s_j), return blocked(s_i) to the sender of the probe. Go to step 1
4. If the node is a sink node,
   - If a_{lm} used a forward labeling, $f(a_{lm}) \leftarrow f(a_{lm}) + \Delta_{lm}$.
   - If a_{lm} used a backward labeling, $f(a_{lm}) \leftarrow f(a_{lm}) - \Delta_{lm}$.

   clear labels for s_i and send path to sender of probe.
5. Select an unlabeled arc, a_{lm}. If there are no more unlabeled arcs that are not already at capacity, send echo(s_i) to the sender of the probe, n_k, and clear the label for s_i.
6. $\Delta_{lm} \leftarrow u_{lm} - f_{lm}$.
7. Label the arc and send a probe of $(s_i, \Delta{=}\min(\Delta_{kl}, \Delta_{lm}))$. If there are no more unexplored arcs that are not already at capacity,
8. Wait for messages to return. If the return message from node n_m is
   - path(s_i, Δ_{lm})
     - If a_{lm} used a forward labeling, $f(a_{lm}) \leftarrow f(a_{lm}) + \Delta_{lm}$.
     - If a_{lm} used a backward labeling, $f(a_{lm}) \leftarrow f(a_{lm}) - \Delta_{lm}$.

     clear labels for s_i, set $\Delta_{kl} \leftarrow \Delta_{lm}$, and send path$(s_i, \Delta_{kl})$ to sender of probe.
   - blocked(s_i), all paths to t are blocked by other searches, clear labels for s_i and send blocked(s_i) to sender of probe.
   - echo(s_i): All paths to t are filled to capacity. Go to 5.

Figure 5 – Distributed Algorithm Using Multiple Sources for Max-Flow

In this application, it is unlikely that a processor would exist at each node (bus bar); however, processors would

exist at the FACTS devices. Processors exchange messages with each other over a communication network. For the power grid, this paper assumes each FACTS device and each generator has a processor and the communication links parallel the lines. With multiple source flow, the algorithm explores augmenting flow paths, simultaneously, from each generator by passing probe and response messages over the communications links. Winning paths are selected for the resulting power flow on a "first discovered" basis. Each FACTS devices runs the distributed maximum flow algorithm shown in Figure 5 which is a modification of the one from [4], which is based upon the one from [1].

The control algorithm uses the distributed maximum flow algorithm with the arc capacities initially set to the steady state, unconstrained power flow values. If the loads cannot be satisfied with these capacities, then changes are made to try to satisfy the load. First, all of the arc capacities are increased to the actual corresponding power line capacities. The maximum flow algorithm is then restarted with the sink nodes as source nodes and source nodes as sink nodes. The maximum flow algorithm changes slightly, as well. When the arc flows are modified, the negative value of the Δ is added to or subtracted from the arc flow. The other change to the algorithm is in the calculations of the Δ values. When searching from the sink to the source, $\Delta_{ij} \leftarrow u_{ji} - f_{ji}$.

III. FACTS Placement

The implementation of the max-flow algorithm on the power grid will be accomplished with the use of FACTS devices to enforce the desired flow across each line. The FACTS devices are assumed to be a series device, such as a UPFC, an SSSC, or a TCSC, that can control the active power flow across the line. The objective is to find the optimum placement of the minimum number of FACTS devices to realize the distributed max-flow algorithm. To properly control the network, the devices need to be arranged in a way that the grid remains stable under all possible single line contingencies. The approach is based on an enumeration of single contingencies. The algorithm of Figure 6 describes the proposed method of determining the placement of each FACTS device.

In this approach, each contingency is analyzed separately using a "greedy" strategy. As each contingency is applied, a FACTS device is placed on the line that is the most overloaded as a result of the outage. The amount of overload can be chosen as absolute value of overload (in MW), as a percentage of the base value, or as a percentage of the maximum capacity of the line. The criteria used will result in slightly different enumeration of the FACTS devices, but not surprisingly, those lines which are chronically overloaded will surface regardless of the criteria used. A FACTS device is then placed on the line with the greatest overload and its capacity is set to the value determined by the max flow algorithm. At this point, it is assumed that the FACTS device has suitable controls such that it will be able to regulate the active power flow to the desired value. In the power flow, the FACTS device is modeled as in [5]. This process is repeated for the given contingency until the maximum number of desired FACTS devices have been placed. Then the algorithm is repeated for the next contingency. The algorithm of Figure 6 is applied to the 118 bus test system shown in Figure 7.

Algorithm for FACTS Placement:

Select a line to be outaged and the desired number of FACTS (F_{max});
Execute Max-Flow on the graph without the selected line;

1. Let $F \leftarrow 0$
2. $F \leftarrow F + 1$
3. Compare the actual electrical flow (including all FACTS devices) to the Max-Flow values;
4. Choose the *most* overloaded line;
5. Place a FACTS device on that line and set the flow to the Max-Flow value;
6. If $F \leq F_{max}$, go to 2.

Figure 6 – Algorithm for determining the placement of FACTS devices for each contingency.

This algorithm results in the placement of FACTS devices as shown in Figure 8. The 118 bus system has 179 lines. The horizontal axis indicates which line has been outaged. The dots indicate the line where a FACTS device should be place for that particular outage based on percent overload. Note that the figure has a diagonal tendency. This indicates that the overloaded lines tend to be near the outaged line, which is the intuitive result. However, there are also lines that are chronically overloaded, indicated by the FACTS placement on certain lines denoted by the frequency of dots in the vertical direction. A summary of the chronically overloaded lines are given in Table 1. This table lists the line number (corresponding to Figure 8) and the corresponding sending and receiving end bus numbers (corresponding to Figure 7) of the fourteen most chronically overloaded lines. Note that many of the chronically overloaded lines are in close proximity to one another, frequently caused by the same re-directed power flow as a result of the line outage. By using FACTS and the maximum flow algorithm, it is often possible to mitigate adjoining line overloads simultaneously.

The maximum flow algorithm indicates the amount of active power flow that should be carried by all lines in the system. However, this is impossible to physically enforce, even if each line in the system had a FACTS device on it. Therefore the critical lines are identified using the algorithm of Figure 6, and the maximum flow across these lines is enforced by the use of FACTS devices. It is theorized that once the critical lines are set, the remaining active power flow in the system will tend to flow according to the

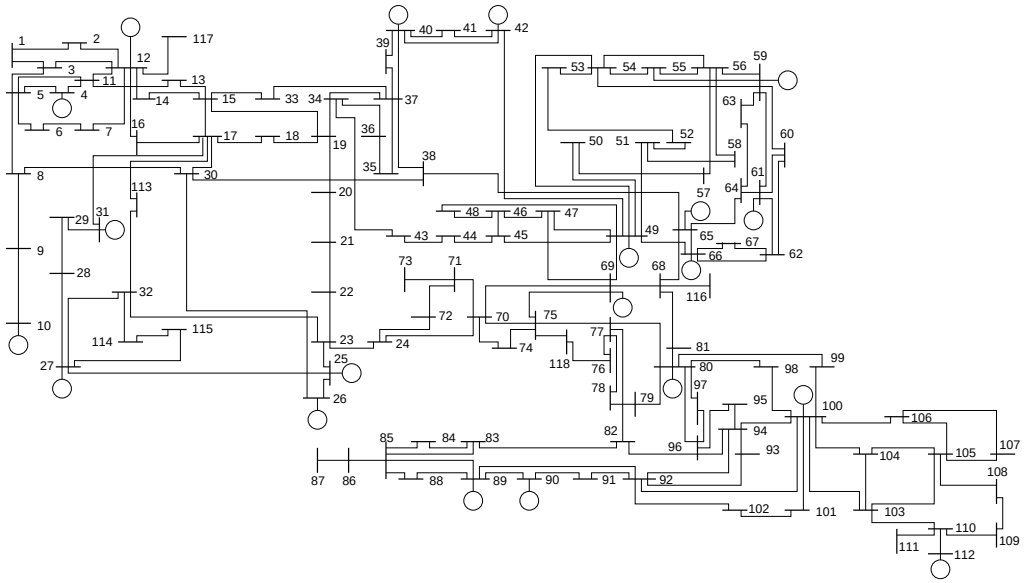

Figure 7 – IEEE 118 bus test system

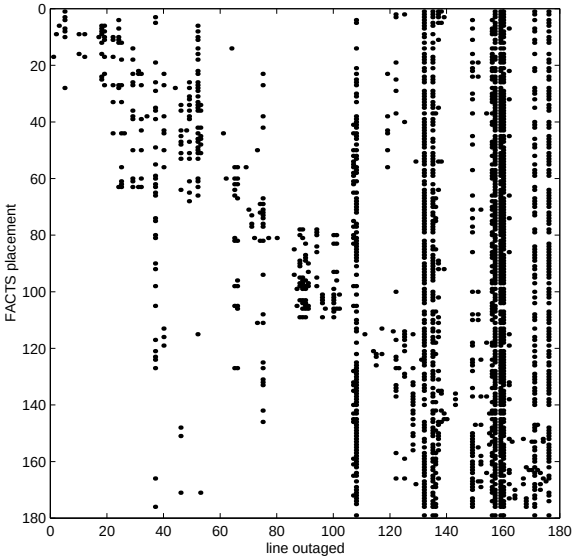

Figure 8 – FACTS placement for each line outage in the 118 bus system

Table 1. Chronically Overloaded Lines

#	i	j	#	i	j
37	23	24	149	91	92
107	65	66	156	94	96
108	65	68	157	94	100
132	80	96	159	96	97
135	80	99	160	98	100
136	82	83	171	105	106
137	82	96	176	109	110

Table 2. FACTS Placement and Line Overload Results

No. of FACTS	No. of Overloads
none	123
1	34
2	23
4	19
5	16
6	15
7	12

maximum flow pattern.

To test this theory, the proposed method is applied to the 118 bus system for a varying number of FACTS devices. The results are summarized in Table 2. This Table indicates that in the base case (with no FACTS devices) there are 123 (out of a possible 179 total lines) line outages that produce at least one line overload. A line is considered to be overloaded if it carries more than 25% above the base case line flow. As FACTS devices are added, the number of cases of overloads decreases significantly. For a placement of four FACTS devices, there are only 19 cases of overload. This means, that out of a possible 179 line outages, only 19 of these outages now cause a line overload; 160 line outages (roughly 90%) do not produce a significant change from the base load flow case. Note that the application of just a single FACTS device reduces the number of overload cases from 123 (roughly 70%) to only 34 cases (20%). These results are achieved through proper placement of the FACTS device and the application of the maximum flow algorithm to determine the amount of active power flow through each FACTS device. Note, however, that there is also a point at which the benefit of applying more FACTS devices saturates – in this test system, that point is about five devices. The decrease in line overloads saturates after this point, where the decrease in overload cases is roughly one case per FACTS device.

IV. Conclusions

This paper presented a novel method for controlling the active power flow through the power system using FACTS devices. This method is based upon the graph-theoretic maximum flow algorithm to determine the scheduling of active power flow on each line in the system. The maximum flow algorithm can actively compensate for line outages by redirecting power flow to avoid cascading failures. The maximum flow algorithm is implemented by using FACTS devices to enforce the active power flow on critical corridors in the power system. The 118 bus system was used as a test bed to show the effectiveness of the proposed placement and scheduling algorithms. The use of the combined FACTS and maximum flow algorithm was able to significantly reduce the number of line overloads as the result of line outages.

Work in this area continues in developing a fault-tolerant distributed maximum flow algorithm that can be implemented in practice without the necessity of a centralized controller. Each FACTS device will be able to execute the maximum flow algorithm in real-time to determine its active power flow set point.

V. Acknowledgements

The National Science Foundation under has supported this work contracts DGE-9972752 and ECS-0085666 and the University of Missouri-Rolla Intelligent Systems Center.

References

[1] L. Ford and D. Fulkerson. Max Flow Through a Network. *Can. J. Mathematics*, pages 399–404, 1956.

[2] D. Fulkerson and G. Dantzig. Calculations of Maximum Flow in Networks. *Naval Logic Quart.*, 2:277–283, 1955.

[3] R. Ahuja and J. Orlin. A Fast and Simple Algorithm for the Maximum Flow Problem. *Operations Research*, 37(5):748–759, 1989.

[4] T.-Y. Cheung. Graph traversal techniques and the maximum flow problem in distributed computation. *IEEE Transactions on Software Engineering*, 1983.

[5] D. J. Gotham, G. T. Heydt, "Power flow control and power flow studies for systems with FACTS devices," *IEEE Transactions on Power Systems*, vol. 13, no. 1, pp. 60-65, February 1998.

A. Armbruster received his BS in Mechanical Engineering in 2000 and MS in Computer Science in 2001 from the University of Missouri-Rolla. He is currently pursuing a Ph.D. in Computer Science at UMR.

Bruce McMillin received the BS in Electrical and Computer Engineering and the MS in Computer Science from Michigan Technological University, Houghton, Michigan, in 1979 and 1985 respectively and the Ph.D. in Computer Science from Michigan State University, East Lansing, Michigan, in 1988 on Reliable Parallel Processing. Dr. McMillin has worked in both academia and industry. He is currently a Professor of Computer Science and research investigator in the Intelligent Systems Center at the University of Missouri at Rolla. During this time period he also spent a year on sabbatical at SUNY Stony Brook. His research interests include fault tolerance, security, parallel algorithms, software engineering, and distributed systems theory.

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 Associate Dean for Research and Graduate Affairs and a Professor of Electrical and Computer 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. She is the Vice-President for Education/Industry Relations of the Power Engineering Society and a recipient of an IEEE Third Millennium Medal.