Autonomous Agents and Cooperation for the Control of Cascading Failures in Electric Grids

Paul Hines, *Student Member, IEEE,* Huaiwei Liao, *Student Member, IEEE,*
Dong Jia, *Member, IEEE,* and Sarosh Talukdar, *Fellow, IEEE*

Abstract—**A power system can be thought of as a stochastic hybrid system: a Finite State Machine whose states involve continuous variables with uncertain dynamics. Transitions in this machine correspond to outages of generation and transmission equipment. A cascading failure corresponds to a series of such transitions whose net effect is a blackout. We present evidence that the probability of cascading failures is subject to phase transitions—large and abrupt changes that result from only small changes in system stress. We suggest a network of distributed, autonomous agents to reduce the ill effects of cascading failures. These agents improve their decisions by cooperating (sharing goals and exchanging information with their neighbors). Results from experiments on the IEEE 118 bus test case are included.**

I. Introduction

POWER systems are described by continuous and discrete variables in deterministic and stochastic relations. In other words, they are stochastic hybrid systems [1,2,3]. The discrete variables include circuit breaker positions and transformer tap settings. The continuous variables include voltages and currents. The uncertainties accrue from modeling deficiencies, hidden failures in protection devices, and random disturbances, such as lightning strokes.

Much has been written about the continuous dynamics of power systems [4]. This work focuses on the combination of discrete and continuous dynamics.

A. Goals

The overall (global) problem of managing a power system has many objectives, including the minimization of damage to equipment and interruptions of service. Stressful operating conditions bring these objectives into conflict. Existing practices always resolve these conflicts in favor of the equipment. For instance, the Aug. 14, 2003 blackout cost billions of dollars in service interruptions, but caused negligible equipment damage [5, 6].

The global, multi-objective, power system operating problem is tackled by decomposing it into thousands of sub-problems, and assigning each to a control-agent. Existing decompositions are far from perfect: locally correct solutions are often globally wrong (the best solutions of the sub-problems are not the best solution of the global problem).

We have two goals. First, to allow for better tradeoffs among conflicting objectives during cascading failures. And second, to find problem decompositions for which local correctness implies global correctness in the control of cascading failures.

B. Cascading failures

A cascading failure is a progression of equipment outages, one propagating another. Each outage can be thought of as a state transition in a stochastic hybrid system. The cascade usually begins with a bizarre disturbance. The resulting dynamics produce violations of the operating constraints and threaten equipment. The protection system acts to remove the threatened equipment from service. Further violations occur, precipitating still more equipment outages, and so on.

Bizarre disturbances, though rare, occur frequently enough to give the distribution of cascading failures and the resulting blackouts a fat tail [7].

We seek to shorten cascading failures by eliminating constraint violations before they can trigger the protection system.

C. Distributed model predictive control

The autonomous agents that we have designed use distributed model predictive control (DMPC) to accomplish this goal. We choose to use distributed agents because they can act quickly and are robust to failures. We design the agent network by formulating the global control problem into an optimization problem and decomposing it into sub-problems, one for each agent. The agents work on their sub-problems in parallel, using whatever information they are

This work is supported in part by ABB corporate research, NSF, and the Carnegie Mellon Electricity Industry Center.

Paul Hines is with the Eng. and Public Policy Dept., Carnegie Mellon U., Pittsburgh, PA 15213 USA (e-mail: phines@cmu.edu).

Huaiwei Liao is with the Elect. and Computer Eng. Dept., Carnegie Mellon U., Pittsburgh, PA 15213 USA (e-mail: hliao@andrew.cmu.edu).

Dong Jia is with the Elect. and Computer Eng. Dept., Carnegie Mellon U., Pittsburgh, PA 15213 USA (e-mail: djia@andrew.cmu.edu).

Sarosh Talukdar is with the Elect. and Computer Eng. Dept., Carnegie Mellon U., Pittsburgh, PA 15213 USA (e-mail: talukdar@ece.cmu.edu)

0-7803-8812-7/05/$20.00 ©2005 IEEE

able to collect locally. In this case, for each agent to make globally optimal or even globally feasible decisions it would need much more information than it could practically collect in real time. It would need to know the state of the entire grid and how every other agent will act in response to that state. To compensate for this information problem, we design the agents to use model predictive control (MPC) to solve their local problems.

Model predictive control is a repetitive procedure that combines the advantages of long-term planning (feed-forward control based on performance predictions over an extended horizon) with the advantages of reactive control (feedback using measurements of actual performance). At the beginning of each repetition, the state of the system to be controlled is measured. A time-horizon, stretching into the future, is divided into intervals. Models are adopted to predict the effects of control actions on system-states in these intervals. The predictions are used to plan optimal actions for each interval, but only the actions for the first interval are implemented. When this interval ends, the procedure is repeated. Ref. [8] provides an overview of MPC theory and practice for centralized applications.

MPC, because it uses optimization for making decisions, readily accommodates large numbers of complex constraints. Many other control techniques do not allow inequality constraints. Instead, they require the designer to approximate the effects of constraints with conservative assumptions.

The adaptation of MPC to distributed agents results in a two dimensional decomposition of the global problem. The MPC procedure is a temporal decomposition, and the distribution of the problem to autonomous agents is a spatial decomposition. The agent models decrease in fidelity with distance in both dimensions, resulting in the agent being able to make good decisions for the here and how and approximate predictions for more distant actions. DMPC is described in detail in Ref. [9].

D. *Agent cooperation*

If designed correctly, agents that cooperate will achieve solutions that are better, or at least no worse, than agents that act unilaterally. We define cooperation as exchanging useful information and sharing commensurate goals. We have found that enabling agents to cooperate allows for vastly improved decision making. This result is similar to that found by Talukdar and Camponogara [10], although the algorithm used for this application differs somewhat.

II. Measuring the Risk of a Cascading Failure

The outage of a component, such as a transmission line or generator, is a discrete event that suddenly changes a network's configuration. The dynamic response of the network to such changes can over-stress some of its remaining components. Thousands of relays spread throughout power networks, seek to prevent over-stresses from lasting long enough to cause harm.

$C(S)$, a cascading failure of length L and size |S|, can be thought of as an alternating sequence of equipment-outages and threshold-crossings:

$$C(S) = \{E_0, T_0, E_1, T_1, \ldots, E_L, T_L\} \qquad (1)$$

where:

E_0 is a set of one or more outages that initiated the cascade. E_0 is typically a multiple-contingency event.

T_n is the n-th set of threshold-crossings (constraint violations) in the sequence.

E_n is the n-th set of equipment-outages in the sequence.

T_L is the only empty set in the sequence, and signals its end.

$S = E_0 \vee E_1 \vee \ldots \vee E_L$ is the set of all the components lost during the cascade.

There are two principal difficulties in accurately predicting $C(S)$, given E_0:

- *Complexity:* The relays in power networks set thresholds on several types of variables, including impedance, voltage, current, and frequency. The more detailed models for calculating the dynamics of these variables require large computational and data-gathering efforts.
- *Uncertainty:* The responses of power networks to sudden changes are profoundly uncertain. Unknown load dynamics, imprecise parameter values, and hidden failures are among the reasons.

We will use a probability based measure to asses the severity of cascading failures. Consider the probability, P(y), of a cascading failure, where:

$$P(y) = \text{Probability}\ [\ C\ (|S| \geq y)\ \Big|\ E_0 \wedge H\ \wedge x\ ] \qquad (2)$$

where H is a set of hidden failures and x is a measure of network-stress such as average line-loading. In words, $P(y)$ is the probability that a random set of outages, E_0, will produce a cascading failure of size y or greater, given the current state of the system (H and x).

For the purposes of this work, a *phase transition* occurs when a small change in the stress on the network produces a large and abrupt change in *P(y).*

In [11], experiments based on DC power flow model are conducted on two types of networks: a regular, square, 50x50 grid, and a simplified model of a 3357-node power network. Both networks demonstrate distinct phase transitions in *P(y)*. Experimental results for the 3357-node power network are shown in Fig. 1.

Given that phase transitions occur in simulated networks, we conjecture that the probability of a cascading failure in actual power networks, given a multiple contingency, is similarly subject to phase transitions. If this conjecture is true, it may be possible to develop on-line techniques for assessing the risk of cascading failures.

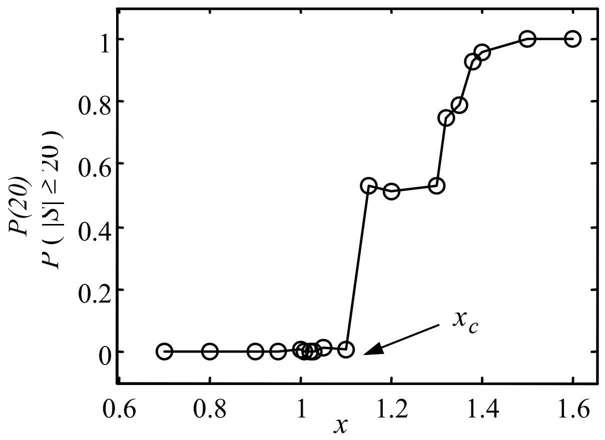

Fig. 1—*P(20)*, the conditional probability of cascading failures of size 20 or greater, plotted against x, the base load multiplier, for a 3357-node network. x_c marks the critical point at the beginning of a phase transition.

III. Controlling Cascading Failures

The global control problem that we use to control cascading failures is:

- to eliminate network violations before the protection system operates (causes a state transition) at minimum social cost.

This problem can be formulated as a non-linear programming problem using steady-state power network equations. We define this problem using the notation below. Let:

N be the index set of all the nodes in the network.

n be the index of the agent located at bus *n*.

Q be the index set of all the branches in the network.

V be a complex vector of node voltages. V_{nk} is the voltage at bus n at time step *k*.

I be a complex vector of node current injections. In is the injection at bus *n*.

G be a complex vector of generation power injections. For the sake of notational simplicity, we assume no more than one generator is located at each bus. It is fairly easy to incorporate multiple generators or loads, but doing so complicates the notation somewhat.

L be a complex vector of load powers. As above, we assume one load at each bus.

Y_{NN} be the complex node admittance matrix for all the nodes in the network.

Y_Q be the complex branch admittance matrix for the set of all branches in the network.

y_{nm} be the single element of the node admittance matrix that is the admittance between buses *n* and *m*.

Since many cascading failures are propagated by under/over-voltage violations at buses and over-current violations on transmission lines, we include these constraints in our formulation. These are not the only violations that can trigger protection system actions, but are a useful starting point for our design. The decision space is any combination of load and generation shedding. We assume that both load and generation can be shed in continuous quantities. The objective function is the total social cost of the control actions. Therefore the global, single period control problem can be written as follows.

$$\underset{G,L}{\text{minimize}} \sum_{n \in N} Cost_n \left(G_n - G_{n0}, L_n - L_{n0} \right) \tag{3a}$$

subject to:

$$I = Y_{NN} V \tag{3b}$$

$$G_n - L_n = V_n conj(I_n),\ n \in N \tag{3c}$$

$$\text{Re}\left(L_n / L_{n0}\right) = \text{Im}\left(L_n / L_{n0}\right),\ n \in N \tag{3d}$$

$$G_n^{\min} \le G_n \le G_n^{\max},\ n \in N \tag{3e}$$

$$0 \le L_n \le L_{n0},\ n \in N \tag{3f}$$

$$\left|V\right|^{\min} \le \left|V\right| \le \left|V\right|^{\max} \tag{3g}$$

$$\left|I_{nm}\right| = \left|y_{nm}(V_n - V_m)\right| \le \left|I_{nm}\right|^{\max},\ n,m \in N, n \ne m \tag{3h}$$

The costs associated with shedding load (from 3a) are the social costs that would be incurred from the interruption of electrical service. If some loads are deemed more important to the system than others, these varied costs can be incorporated into (3a). The costs associated with reducing generation (also in 3a) come either from the potential equipment damage that rapid deceleration could cause, or the amount that would have to be paid to an independent power producer for such emergency control. Equality constraint (3b) defines the voltage-current relationships in the network. Equality constraint (3c) expresses conservation of energy at each node. Equality constraint (3d) forces the system to shed real and reactive load in equal proportions. Inequality constraints (3e) and (3f) describe the extent to which loads and generation can be adjusted. The final inequality constraints (3g and 3h) define the measures used to identify violations (bus voltage, and line currents), as well as the limits on those measures. This formulation can be extended to include constraints on the dynamic system, such as frequency or generator "out-of-phase" limits, but such extensions are beyond the scope of this paper.

Simulations using IEEE test networks indicate that voltage and current violations can be eliminated by solving this problem and implementing the resulting control actions.

A. Decomposition

As discussed in the introduction we solve the global problem (3) by decomposing it in two dimensions. First, we decompose the problem in space by placing an agent with local control abilities at each bus in the network. Second, we decompose the problem in time using MPC.

During normal operation an agent collects measurements from its neighborhood and waits for network violation. Operators control the grid using standard procedures.

Occasionally (perhaps daily or weekly) the agent will collect data from more remote agents in order to maintain a rough model of more remote portions of the network. When the agent finds a violation it immediately calculates an adjustment to its local control variables using its local DMPC problem and the network information that it has collected. It then adjusts its local control variables by shedding load or generation and monitors the effect on the violation. After a short time delay it repeats this procedure until the violation is eliminated. Fig. 2 gives a high level view of the system design.

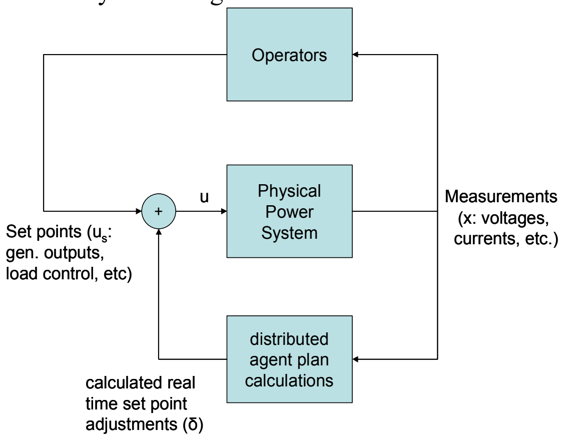
Operators
Physical Power System
u
+
Measurements (x: voltages, currents, etc.)
Set points (u$_s$: gen. outputs, load control, etc)
distributed agent plan calculations
calculated real time set point adjustments (δ)

Fig. 2—Feedback diagram of the system showing how the agent network and operators interact. Under normal circumstances agents make no adjustments to operator set points. If a violation occurs, agents work together to eliminate the violations by adjusting control set points.

B. DMPC formulation

To find good solutions to the global problem (3), the agents must solve local MPC problems that preserve the important relationships of (3), and are relatively insensitive to remote measurement errors. We have found that linear-difference approximations of the global formulation can achieve this property. In some applications, a Taylor series expansion of the global problem could work well, but for non-linear problems calculating the complete gradients can require more data than an agent is able to collect. To work around this problem we build a linear-difference formulation using standard DC load flow approximations.

The resulting sub-problem for agent *n* is therefore:

$$\underset{\Delta_{MK}}{\text{minimize}} \sum_{k=k_0}^{K} e^{-\rho k} c_M{}^T \Delta_{Mk} \tag{4a}$$

subject to:

$$\sum_{g \in G_M} \delta_{g,k} = \sum_{l \in L_M} \delta_{l,k} \tag{4b}$$

$$x_{M,k} = x_{M,k-1} + D_{MM} \delta_{M,k} \tag{4c}$$

$$x_{\min} - f_1(k) \le x_{M,k} \le x_{\max} + f_2(k) \tag{4d}$$

$$\delta_{\min} \le \delta_{Mk} \le 0 \tag{4e}$$

$$-u_{M,k_0} \le \sum_{k=k_0}^{K} \delta_{M,k} \le 0 \tag{4f}$$

where:

M is the index set of all nodes (or control variables) that agent *n* includes in it sub-problem. $M \subset N$

K is the final time step in the control time horizon.

k is the current time period.

u is a vector of control variables.

δ is a vector of control variable adjustments such that $\delta_k = u_k - u_{k-1}$.

$\Delta_{M,K}$ is the full plan for all control variables (*M*) and time periods (*K*) in agent *n*'s sub-problem.

c is a vector of costs associated with control variable reductions.

ρ is the discount factor used in the MPC cost function.

$x_{M,k}$ is a vector of state variables in agent *n*'s problem. $x_{M,k}$ indicate violations that could trigger a cascading failure when outside of known bounds (x_{min}, x_{max}).

f_1, f_2 are slack functions that allow the agent to eliminate the known violations iteratively over time instead of all at once.

D is a matrix of state variable sensitivity factors.

The output of an agent's problem is a control plan for every control variable in M and for the entire time horizon (K time steps). The plan includes an action to take locally and immediately, as well as estimates of what the local agent will do in the future. The plan also includes estimates of what other agents will do both now and in the future. During each iteration the plan is updated and the control horizon reduced. For example, if each period is 2 seconds long, the plan might be 10 sec. long during the first period, 8 sec. long during the second period, 6 sec. during the third, etc. A typical control plan is illustrated in Fig. 3.

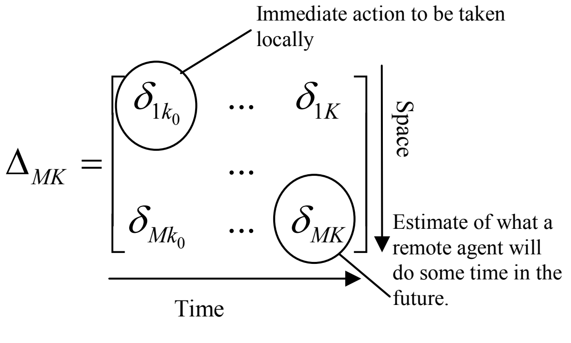
Immediate action to be taken locally
$\Delta_{MK} =$
δ_{1k_0}
...
δ_{1K}
...
δ_{Mk_0}
...
δ_{MK}
Space
Estimate of what a remote agent will do some time in the future.
Time

Fig. 3—Diagram of the 2 dimensional control plan calculated by agent *n*.

C. Cooperation

We have found that, if properly designed, cooperative agents can vastly outperform agents acting unilaterally. Cooperation can take a variety of forms. According to our earlier definition of cooperation as sharing goals and exchanging useful information, an agent that merely solves (4) and acts locally is only partially cooperative. Such an agent uses an overlapping objective function (4a) but does not exchange useful information with its neighbors before taking action. In order to improve performance we have studied a variety of designs for cooperative agents.

The algorithm presented in this paper is based on our finding that agents with only local information can overlook important data located just outside the agent's local area. Consider two agents: A and B. A is near a violation that B should react to, but B is unaware of the problem because the problem lies just outside of B's neighborhood (but not A's). If A solves its local problem, calculates that B should act, and then shares the important violation data with B, B will likely be able to make better decisions about its local control actions. If B replies and shares its local data with A, A may also be able to improve its solution.

With this design in mind the following is a cooperative control algorithm for agent *n*. During normal operation the agent gathers local data and exchanges information with other agents. If the agent becomes aware of a violation the agent follows the following procedure:

1. Solve (4) to obtain the control vector for the time period k_0 ($\delta_{M,k0}$).
2. Determine a set of agents (*Q*) that appear to require control action.
3. Compare solutions with those agents in set *Q*.
4. If a large discrepancy is found, exchange data with the agents with whom there exists a discrepancy. This results in agent *n* enlarging its problem boundary (*M*).
5. Re-solve (4) with the updated data.
6. Iterate from 2 until consensus is reached, or until a maximum number of iterations has occurred.

This rather simple cooperation algorithm was found to be quite effective. Each agent may begin with severely limited information but through the cooperation process the relevant agents obtain more detailed information about important aspects of the network. In our simulations we have found that agents reach consensus within one or two iterations. We limit this process to three iterations.

D. *Results*

We tested this method with and without cooperation using simulations on standard IEEE test systems. We assume that the network acts according to a standard, non-linear, power flow model with constant real/reactive power loads and constant power/voltage generators. The network is assumed to perform frequency regulation through a single slack bus. Control variable reduction costs of $1000/MW and $30/MW are assigned to loads and generators respectively. All agents use a four period (*K*=4) time horizon, but continue to act using a single period (*K*=1) formulation if one or more violations persist after the initial control horizon. For each simulation we choose a communication neighborhood radius, *r*. All of the buses that can be reached by traveling over no more than *r* branches are within an agent's neighbor set. Within this radius we assume that the agent is able to obtain perfect measurements constantly. An agent obtains no real-time measurements from the buses outside of its neighborhood, but can estimate the quantity of load and generation if needed. We model the effects of this estimation process using Gaussian noise with 15% coefficient of variation (CV). Each simulation begins with a violation inducing disturbance. The simulations in this paper are for branch-outage disturbances only.

Therefore, the simulation process proceeds as follows:

1. Choose a network and initial conditions.
2. Allow the agents to take noisy measurements such that they have estimates of the network control variables.
3. Inject a disturbance (one or more branch outages) that causes at least one violation.
4. Run a power flow to obtain the modified network state.
5. Set *k*=0, *K*=4.
6. Allow the agents to take measurements from their local neighborhoods.
7. Allow the agents to calculate control plans for the control horizon (*k*+1…*K*), and implement the portion for period *k*+1.
8. Increment *k* (and *K* if *k+1>K*).
9. Recalculate the power flow.
10. Repeat from 6 until all violations are eliminated, or until it is clear that the agents are unable to eliminate the remaining violations.

Fig. 4 shows typical simulated control trajectories for a violation on the IEEE 118 bus system. The control goal (as defined by f_1 and f_2) is indicated along with the resulting violation trajectories. The cooperative agents outperform agents acting unilaterally by a large margin.

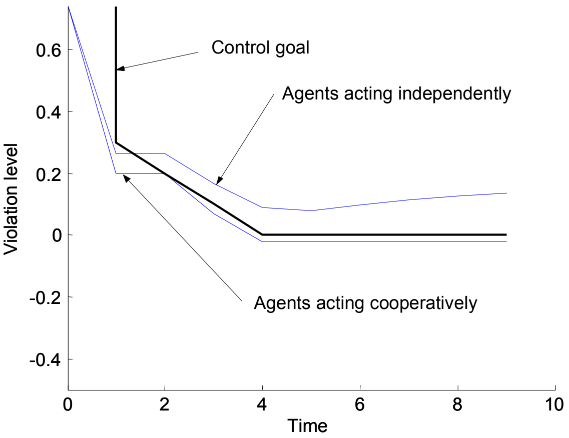

Fig. 4—Control trajectories for agents acting cooperatively and agents acting independently. This shows the results of an experiment on the IEEE bus with an outage on line 97. The communication radius (*r*) is two for both cases.

In order to determine an appropriate agent communication radius we performed a series of simulations with 15 severe disturbances and a range of communication radii (*r=2* to *r=10*). This allowed us to us to compare solution quality and quantity of communication. The results of these experiments are shown in Fig. 5.

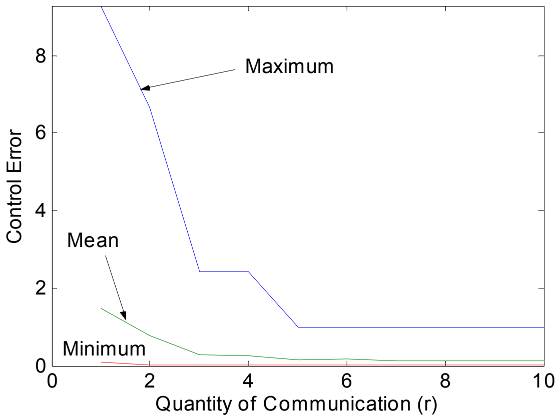

Fig. 5—Plot of control error vs. quantity of communication for the 15 disturbances simulated. Control error is measured by calculating the amount by which the action violation trajectory exceeds the control goal. Communication is in terms of the radius of an agent's neighbor set. The above graph shows the relationship for agents acting independently.

IV. Summary

Cascading failures can be thought of as state transition sequences in a stochastic hybrid system. For power networks, we have found evidence that the probability of a cascading failure increases sharply as loading increases past a critical point. We conjecture that such phase transitions occur along other trajectories as well. We propose controlling cascading failures by eliminating network violations before they trigger state transitions. We demonstrate that cooperative autonomous agents using Model Predictive Control can accomplish this goal without global knowledge. We also show that agent performance can be substantially enhanced through the use of cooperation.

The methods that we present here are in the early stages of development and therefore leave substantial room for improvement. Using our current methods, estimating the probability of a cascading failure requires thousands of simulations. We hope that improvements in this method will result in tools that are practical for real-time use. Agent performance can likely be improved by using a network model that approximates the system dynamics more closely. Additionally we expect that agent performance can be enhanced through the use of learning and additional forms of cooperation. Finally we hope to study some of the institutional barriers that grid control technologies must overcome in order to achieve more widespread adoption in restructured electricity systems.

References

[1] P.J., Antsaklis, "Special Issue on Hybrid Systems," *Proceedings of the IEEE*, Vol. 88, No. 7, July 2000.

[2] P.J., Antsaklis and A. Nerode, "Special Issue on Hybrid Control Systems," IEEE Transactions on Automatic Control, Vol. 43, No. 4, July 1998.

[3] G. Pola, M.L. Bujorianu, J. Lygeros and M.D. Di Benedetto, "Stochastic Hybrid Models: An Overview," Proc. of IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 2003), Saint-Malo, France, June 16-18, 2003.

[4] P. Kundur, *Power System Stability and Control*, McGraw Hill, New York, 1993.

[5] U.S.-Canada Power System Outage Task Force, *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations.* April 2004.

[6] ICF Consulting, *The Economic Cost of the Blackout,* 2004, online: www.icfconsulting.com.

[7] S. N. Talukdar, J. Apt, M. Ilic, L. B. Lave, and M. G. Morgan, "Cascading failures: Survival versus Prevention," *The Electricity Journal,* Nov. 2003.

[8] J. B. Rawlings, "Tutorial Overview of Model Predictive Control," *IEEE Control Systems Magazine*, vol. 20, no. 3. pp. 38-52, 2000.

[9] E. Camponogara, D. Jia, D., B.H. Krogh and S. Talukdar, "Distributed Model Predictive Control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44-52, 2002.

[10] S. Talukdar and E. Camponogara. "Collaborative Nets." *Proceedings of the 33rd Hawaii International Conference on System Sciences.* 2000.

[11] H. Liao, "Phase Transitions in the Probability of Cascading Failures," Proc. of the Carnegie Mellon Electricity Transmission in Deregulated Markets conference, December 15-16, 2004. online: www.ece.cmu.edu/~tanddconf_2004/.