

# Distributed Model Predictive Control for the Mitigation of Cascading Failures

Sarosh Talukdar, *Fellow, IEEE*, Dong Jia, *Member, IEEE*, Paul Hines, *Student Member, IEEE*, and Bruce H. Krogh, *Fellow, IEEE*.

**Abstract**—Most large blackouts are caused by cascading failures—sequences of equipment outages, one set of outages precipitating another. We study the application of distributed, autonomous agents for shortening such sequences. Each agent controls a single variable—the consumption of a load or the output of a generator. Each agent uses model predictive control and cooperates with its neighbors in making its decisions. Experiments using the IEEE 118-bus test case illustrate the effectiveness of this method.

## I. INTRODUCTION

IN model predictive control (MPC) strategies, a control agent solves an optimal control problem over a finite time horizon and applies the first control action from the optimal solution. At the next control step, the agent updates the state based on feedback information from the sensors and then resolves the optimal control problem [6-7]. A model incorporated as a set of constraints in the optimization problem predicts the system's response. MPC is used extensively to control large-scale manufacturing processes, particularly in the chemical industry [8-9]. The principal reasons for the success of MPC are: operating and control constraints are accommodated directly by incorporating them explicitly into the MPC optimization problem; and the optimization objective function can reflect real operating objectives, particularly economic factors. The principal drawback is that the computations may be too intense to close high-rate feedback loops.

For large-scale applications, distributed MPC has been proposed to reduce the amount of communication and computation that would be required to implement a centralized MPC algorithm [1-3]. Distributed implementations may also be more robust than centralized MPC, and in some applications certain restrictions on information flow may prohibit the implementation of a centralized controller. Different cooperation strategies have been proposed to guarantee feasibility and stability [3-5]. For example, to provide some information about anticipated

effects of interactions between subsystems, the distributed control agents may exchange predicted state trajectories [3-4] or sets of possible reachable states [5]. In the distributed MPC scheme studied in this paper, control agents take into account their influences on neighboring subsystems and anticipate the reactions of neighboring control agents by incorporating their neighbors' dynamics and objectives into local MPC problems. The agents' decisions might be different from those anticipated by their neighbors because the agents have different models. Each agent uses a local model of the system representing its local neighborhood in detail, but uses simplified models of more remote areas of the grid. The control agents cooperate to reduce the effects of these differences by exchanging solutions to their local problems with their neighbors.

We consider the application of distributed MPC to the problem of controlling cascading failures in a power system. While MPC has been applied to power system problems [15-16], ours is a new application of the distributed MPC method to the cascading failure problem. A cascading failure is typically caused by an initiating disturbance that results in unusual violations of normal limits on voltages, currents, and frequency. When these limit-violations persist, protection systems react to de-energize the equipment that the violations threaten. This can result in subsequent limit violations and additional protection system actions. Thus, the initial disturbance initiates a sequence of abrupt transitions in the system's topology that make up the cascading failure. A blackout, that is a large-scale interruption of demand, is a byproduct of such sequences of protective relay actions.

MPC control strategies may be effective against cascading failures. The objective is to shed small amounts of load and generation judiciously, thereby alleviating violations of operating limits before protection systems take actions that make the problem worse. In most cascading failures, there is a substantial time lag between the initiating disturbance and the first set of relay actions. In the 2 July 1996 event in the western US, after a series of several bizarre relay operations occurred in a period of about 1.5 sec, the system remained in a stressed condition for more than 20 sec before the final sequence of protection system actions commenced [10]. The Northeast American blackout of 14 August 2003 progressed substantially slower. The sequence of cascading events took more than one hour to complete. While it would have been very difficult for operators to react to control the WSCC

Manuscript received March 7, 2005. This work was supported in part by the Carnegie Mellon Electricity Industry Center, ABB Corporate Research, and the National Science Foundation.

S. Talukdar is with Carnegie Mellon U., Elect. and Computer Eng. Dept. and the Dept. of Eng. and Public Policy.

D. Jia is with Carnegie Mellon U., Elect. and Computer Eng. Dept.

P. Hines is with Carnegie Mellon U., Dept. of Eng. and Public Policy.

B. H. Krogh is with Carnegie Mellon U., Elect. and Computer Eng. Dept.

cascading failure, 20 seconds should be sufficient time for an MPC algorithm to operate.

This paper is organized as follows. Section 2 formulates the optimization problems to be solved by each of a set of autonomous agents and how the agents can cooperate in solving their problems. Section 3 presents results of several simulation studies applying the distributed MPC strategy with cooperation to the 118 bus IEEE test case. These simulations illustrate the potential for avoiding cascading failures using distributed MPC and also illustrate some advantages of cooperation. The concluding section summarizes the work in this paper and describes directions for future research..

## II. THE PROBLEM

We wish to prevent equipment outages by driving all operating-constraint violations to zero before the protection system acts. Think of the process as a sequence of MPC problems, each of the form: given the present state of the system (including some constraint violations), and a deadline, develop a plan (the amounts of load and generation to be shed over a horizon extending from now to the deadline) to eliminate the violations before the deadline expires, and at minimum social cost. This problem can be formulated as a non-linear programming problem, using the steady state power network equations from an optimal power flow [9]. Let:

- $t_0, t_1, \dots, t_K$  be discrete points in time,  $t_0$  being the present time, and  $t_K$  being the deadline.
- $U_k$  be the value, at time  $t_k$ , of the network's continuous control variables,
- $X_k$  be the value, at time  $t_k$ , of the network's continuous state variables.
- $\mu$  be a next-state-predictor, such that  $X_{k+1} = \mu(U_k, X_k)$
- $G(U_K, X_K) \leq 0$  be the constraints to be met by the deadline,  $t_K$ ,
- $C(U_0, U_1, \dots, U_K)$  be the cost of the succession of control changes from  $U_0$  to  $U_1$  to... to  $U_K$ .

Then, each of the sequence of MPC problems has the form:

$$\begin{aligned} \text{(OP):} \quad & \text{Minimize} \quad C(U_0, U_1, \dots, U_K) \\ & U_0, \dots, U_K \\ \text{Subject to} \quad & X_{k+1} = \mu(U_k, X_k); \quad k = 0, 1, \dots, K \\ & G(U_K, X_K) \leq 0 \end{aligned}$$

Note that in this formulation, the value of  $X_0$  is measured, the value of  $U_0$  is to be calculated, and all the other  $X$ 's and  $U$ 's are predicted to take the long-term effects of  $U_0$  into account.

### A. Problem decomposition

Suppose there are  $N$  autonomous agents placed so there is an agent at each generator and load. The goals of this section are to decompose the overall problem, (OP), into sub-problems, (SPO<sub>n</sub>), such that: a) each sub-problem can be

assigned to an autonomous agent, b) each sub-problem is easier to solve than the overall problem, and c) the optimal solutions of the sub-problems constitute an optimal solution of the overall problem.

We seek to achieve these goals by: i) breaking the control vector into  $N$  disjoint parts, ii) making agent- $n$  responsible for calculating only one of these parts, iii) allowing agent- $n$  to assume that the other agents will calculate their parts optimally, and iv) using simplified models to predict the responses of those distant parts of the network that are relatively insensitive to agent- $n$ 's decisions.

Consider agent- $n$ . Let:

$Z_{nk}$  be the subset of  $U_k$ , such that  $U_k = [Z_{1k}, Z_{2k}, \dots, Z_{Nk}]$ , and  $Z_{nk}$  is assigned to agent- $n$

$Y_{nk}$  be the part of  $U_k$  that is not assigned to agent- $n$ . In other words,  $U_k = [Z_{nk}, Y_{nk}]$ .

If agent- $n$  acts independently, without communicating with the other agents, the sub-problem it must solve is:

(SP<sub>n</sub>): Predict what the other agents will do and how the network will respond, that is, predict  $Y_{n0}, \dots, Y_{nN}$  and  $X_1, X_2, \dots, X_K$ . Simultaneously, solve the optimization problem:

$$\begin{aligned} \text{Minimize} \quad & C(U_0, U_1, \dots, U_K) \\ & Z_{n0}, \dots, Z_{nK} \\ \text{Subject to} \quad & G(U_K, X_K) \leq 0 \end{aligned}$$

Assuming that the other agents act optimally, this problem can be rewritten as follows.

$$\begin{aligned} \text{(SPO}_{nk}\text{):} \quad & \text{Minimize} \quad C(U_0, U_1, \dots, U_K) \\ & \{Z_{nk}, Y_{nk}\}_{0 \leq k \leq K} \\ \text{Subject to} \quad & x_{nj,k+1} = M_{nj,k}(U_k, X_k); \\ & \quad \quad \quad j = 1, 2, \dots, J; \quad k = 0, 1, \dots, K-1 \\ & G(U_K, X_K) \leq 0 \end{aligned}$$

where:

- $x_{nj,k}$  is the state of the network at time,  $t_k$ , in region,  $R_j$ . In other words, the state of the entire network at time,  $t_k$ , is given by:  $X_k = \{x_{nj,k}\}_{0 \leq j \leq J}$
- $R_0, R_1, \dots, R_J$  are concentric and disjoint regions of the network, such that  $R_0$  is centered on agent- $n$ , and  $R_j$  is closer to agent- $n$  than  $R_{j+1}$
- $M_{nj,k}$  is a network-model such that  $x_{nj,k+1} = M_{nj,k}(U_k, X_k)$

Agent- $n$  predicts future states of the network with the aid of models,  $M_{nj,k}$ , that are centered at its location. The models are specific to the agent. They decrease in fidelity with both distance and time. In other words, each agent has its own suite of models; distant parts of the network are less accurately represented, as are time intervals towards the end of the time-horizon.

There is a tradeoff between the quality of the decisions agent- $n$  makes and the effort expended on predictions (of what the other agents will do and how the network will respond).

The more accurate the predictions, the closer the optimal solution of (SPO<sub>n</sub>) to the optimal solution of (OP). The cruder the predictions, the less the effort needed to make them.

Notice that agent-*n* solves the overall problem, but conditioned on the unique and simplified view of the network provided by its suite of models, {M<sub>njk</sub>}. (SPO<sub>nk</sub>) is simpler than (OP) because {M<sub>njk</sub>} is simpler than μ, especially for parts of the network that are far from agent-*n*. Of course, even though agent-*n* predicts the entire control vector, it implements only the part assigned to it, and only for the first time-interval.

### B. Cooperation

We will say that two agents cooperate if they share goals (objectives or constraints) and exchange information to better meet these goals.

Two obvious forms of cooperation are: a) for agents to tell their neighbors what they intend to do, and b) to pass along measurements that other agents may not be able to make. (Each agent can sense only a small part of the network; without help from its neighbors, it cannot be expected to obtain an accurate picture of what is happening in the entire network.)

These two forms of cooperation require the agents to communicate with their neighbors. But their optimization tasks become easier. Specifically, the task for agent-*n* becomes:

$$\begin{aligned}
(\text{SPC}_{nk}): \quad & \text{Minimize } C(U'_0, U'_{1,\dots}, U'_K) \\
& \{Z_{nk}\}_{0 \leq k \leq K} \\
& \text{Subject to } x_{nj1} = M_{nj1}(U'_0, X'_0); \\
& \quad j = 1, 2, \dots, J \\
& \quad x_{njk+1} = M_{njk}(U'_k, X'_k); \\
& \quad j = 1, 2, \dots, J; \quad k = 1, 2, 3, \dots, K-1 \\
& \quad G(U'_K, X'_K) \leq 0
\end{aligned}$$

where U'<sub>k</sub> and X'<sub>k</sub> are synthesized from agent-*n*'s own calculations and measurements as well as those supplied by its neighbors. There are many different ways of making these syntheses. As yet, we have tried very few and will need to do much more work, if we are to find the best ones.

### C. Application to the cascading failure problem

This section briefly describes how we apply the above general formulations to the cascading failure problem. For more details see [13].

The overall problem is adapted from a standard optimal power flow formulation [12]. The primary differences are that the cost function is the cost of load and generation shedding rather than generator fuel or bid costs and demand schedules.

For the decomposed problem, agent-*n* divides the network into four regions, R<sub>0</sub>—R<sub>3</sub>. R<sub>0</sub> contains the local node (bus *n*) where agent-*n* has direct control and measurement abilities. R<sub>1</sub>, called the local neighborhood, is the sub-network of

radius *r*<sub>1</sub> around bus *n*. Agent-*n* obtains constant measurements from buses within R<sub>1</sub> such that it maintains good models of this region. R<sub>2</sub>, named the extended neighborhood, extends to every bus within a radius of *r*<sub>2</sub> from bus *n*. In the extended neighborhood, the agent obtains infrequent (daily or weekly) measurements such that it can estimate the quantity of load and generation at these locations. This provides agent-*n* with crude approximations of the control abilities of agents at these remote locations. The remainder of the network falls into R<sub>3</sub>. Agent-*n* does not take any measurements or estimates of the state or control variables in R<sub>3</sub>. However, it estimates the configuration of the R<sub>3</sub> network by assuming that the remote branches are in some default state, for example given to by a system operator.

The sub-problem for agent-*n* at time *t*<sub>0</sub> is given in (2.1) below.

$$\text{minimize}_{\{\delta_{M1}, \dots, \delta_{MK}\}} \sum_{k=1}^K e^{-\rho k} c_M^T \delta_{Mk} \quad (2.1a)$$

subject to (for *k*=1...*K*):

$$|I_{Bk}| \cong |I_{Bk-1}| + D_{BM} \delta_{Mk} \leq f(I_{B,0\dots k-1}) |I_B|^{\max} \quad (2.1b)$$

$$\sum_{g \in G_M} \delta_{gk} = \sum_{l \in L_M} \delta_{lk} \quad (2.1c)$$

$$RR_g \leq \delta_{gk} \leq 0, \quad g \in G_M \quad (2.1d)$$

$$-u_{M0} \leq \sum_{k=1}^K \delta_{Mk} \leq 0 \quad (2.1e)$$

where:

- M* is the index set of all control variables that the agent includes in its problem. This includes only the control variables in R<sub>0</sub>, R<sub>1</sub>, and R<sub>2</sub>;
- δ<sub>Mk</sub> is the vector of predicted/calculated control variables changes for time *t*<sub>k</sub>;
- c* is a vector of costs associated with load and generation reductions;
- ρ is a discount factor such that 0 < ρ < 1;
- B* is the set of all branches (transmission lines) for which the agent has current measurements. *B* includes all of the branches in R<sub>0</sub>, and R<sub>1</sub>, and those branches in R<sub>2</sub> for which the agent has data from the cooperation process;
- D*<sub>BM</sub> is a portion of the branch current distribution factor matrix calculated using the DC load flow approximations;
- f* is a function that evaluates to a scalar and sets the amount by which the given state variable must be reduced during a given period;
- G*, *g* represent the set of all generator locations in the control vector and an index into that set;
- RR*<sub>g</sub> is the ramp rate for generator *g*: i.e. the amount by which the generator can be reduced between time steps.

Equation (2.1b) gives the combined state variable (branch

current) prediction and limits for each time period ( $t_k$ ). This constraint includes a scaling function  $f$  that adds some slack to the constraints so that the violations need not be entirely eliminated during the first period, but can be eliminated gradually over the time horizon. We also add ramp rate constraints on the generators (2.1d) since there are natural limits to how fast a generator can decelerate, and constraints (2.1e) to ensure that the agent does not enact or predict more load or generation shedding than that which is feasible.

According to our earlier definition of cooperation as sharing goals and exchanging useful information, an agent that merely solves (2.1) and acts is not cooperative. Such an agent uses an overlapping objective function (2.1) but does not exchange useful information with its neighbors before taking action. 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 neighborhood ( $R_l$ ). Consider two agents: agent- $n$  and agent- $m$ . Agent- $n$  is near a violation that agent- $m$  should react to, but agent- $m$  is unaware of the problem because it lies just outside of agent- $m$ ’s neighborhood (but not agent- $n$ ’s). If agent- $n$  solves its problem and calculates that agent- $m$  should act, and then shares the important violation data with agent- $m$ , agent- $m$  will likely be able to make better decisions about its local control actions. If agent- $m$  replies and shares its local data with agent- $n$ , agent- $n$  may also be able to improve its solution. 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.

The simulations in section three reflect results from this method of cooperation.

### III. VERIFICATION

In this section we describe the results of simulations designed to evaluate this method. The following experiments are specifically designed to determine the relationship between agent performance and communication abilities. The below results apply to simulations on the IEEE 118 bus test case, though similar results obtain using other networks. The 118 bus case was modified slightly from the original to match its properties to those of a typical contemporary power system.

#### A. Simulation model description

We model the network using a standard, AC power flow network 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. The initial condition of the network is calculated with an optimal power flow algorithm [12]. One agent is placed at each bus and has the capabilities specified in section 2. Table 3.1 summarizes the important model input parameters and assumptions.

TABLE 3.1—VERIFICATION MODEL INPUT PARAMETERS

<i>Input</i>	<i>Description</i>
Network data	Modified from the IEEE 118 bus test case
Load shedding costs	Randomly assigned between \$500/MW and \$1500/MW
Generator shedding costs	Assigned uniformly at \$30/MW
Solution horizon ( $K$ )	4 time steps
External neighborhood radius ( $r_e$ )	10 branches
Local neighborhood radius ( $r_l$ )	Varies between 1 and 6 branches
External data estimation error	15% coefficient of variation ( $\delta_e/x$ )
Initiating disturbances	Chosen randomly from a set of 100 violation inducing double branch outages

A simulation is initiated by choosing a disturbance, a local neighborhood radius ( $r_l$ ), and allowing agents to sample data from the pre-fault condition of their external networks. During each simulation time step the agents solve their local problems, and implement the required local control action. After the agents have finished their calculations, the affect of agent control actions is calculated using an AC power flow routine. For every disturbance/radius combination this procedure was repeated for both cooperative agents and unilateral agents.

To prevent a zone three or a time over-current relay operation, a violation must be eliminated fairly quickly (1-2 seconds). In order to minimize the risk of a line sagging and causing a fault, longer time delays (seconds to minutes) will likely be acceptable. In this paper we use four period simulations and define  $f$  such that agents seek to reduce current magnitude violations linearly from 130% of the limit in the first period to 100% in the final period (figure 5.1 shows this control goal). If a violation persists past the original planning horizon the agent continues to act to reduce the violation below the threshold.

#### B. Results

What follows are results from 771 simulations using the above procedure and sampled using the assumptions in table 3.1. Additional details about these results and the associated simulations are given in [13].

Each simulation is repeated for agents with and without cooperation. Figure 3.1 shows the trajectory of the most severe violation resulting from a typical disturbance for both cooperative and unilateral agents. Figures 3.2 and 3.3 show the relationship between the quantity of communication (internal neighborhood size) and two measures of performance: control error and completion time. The control error is the area of the space between the control goal and the actual trajectory (see Figure 3.1). For the cooperative agents this area is quite small, while for the unilateral agents it is rather large. The completion time is the number of time

iterations required to reduce the violation to no more than 5% above the constraint. For the cooperative case, the completion time is 4, whereas the non-cooperative agents do not eliminate the violation within the time horizon.

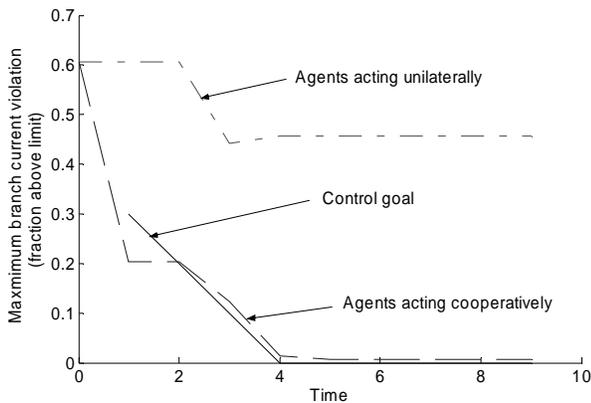


Figure 3.1—A typical violation trajectory resulting from a single simulation.

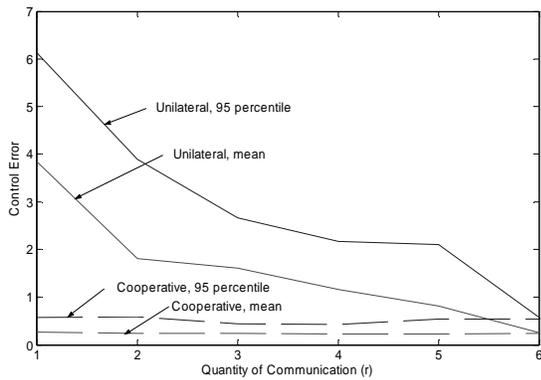


Figure 3.2—The relationship between  $r_l$  and control error

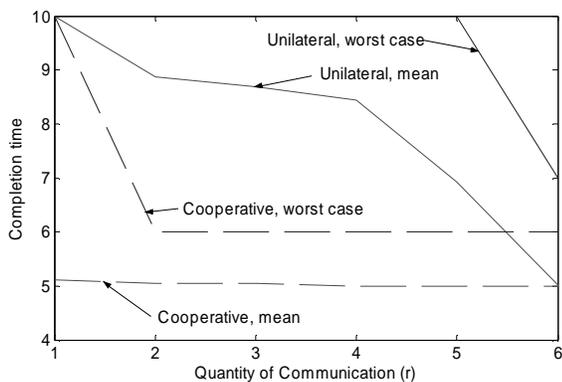


Figure 4.3—The relationship between  $r_l$  and completion time. On average the cooperative agents require one additional time step to remove their violations, independent of the local neighborhood size.

#### IV. CONCLUSIONS

This paper presents a distributed MPC strategy for alleviating operating conditions in power systems that could lead to cascading failures. Experiments suggest that the strategy is effective so long as the time between MPC

iterations is sufficiently large that the network nearly arrives at a steady state before the next control action occurs. If the generator actions can be accomplished quickly, this condition should hold. Tests using a dynamic power system simulator may provide additional insights. The experiments also demonstrate the value of even simple cooperation schemes in agent networks. Without cooperation, the communication required to obtain acceptable performance may be beyond what can be expected from existing technology.

We are currently investigating a number of more elaborate cooperation schemes.

#### REFERENCES

- [1] X. Wang, C. Lu, and X. Koutsoukos. "DEUCON: distributed end-to-end utilization control for real-time systems." Tech Report WUCSE-04-57, Washington University in St. Louis, MO., 2004.
- [2] Hines, P., H. Liao, D. Jia, and S. Talukdar. "Autonomous agents and cooperation for the control of cascading failures in the electric grid." accepted to the 2005 IEEE International Conference on Networking, Sensing and Control, Tucson, March 2005.
- [3] W.B. Dunbar and R.M. Murray. "Receding horizon control of multi-vehicle formations: a distributed implementation." *Proceedings of the Conference on Decision and Control*. 2004.
- [4] D. Jia and B.H. Krogh. "Distributed model predictive control." *Proceedings of the American Control Conference*, 2001.
- [5] D. Jia and B.H. Krogh. "Min-max feedback model predictive control for distributed control with communication." *Proceedings of the American Control Conference*, 2002.
- [6] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. "Constrained model predictive control: stability and optimality," *Automata*, vol. 36, pp. 789-814, 2000.
- [7] J. B. Rawlings. "Tutorial Overview of Model Predictive Control." *IEEE Control Systems Magazine*. vol. 20, no. 3. pp. 38-52, 2000.
- [8] E.F. Camacho and C. Bordons. "Model predictive control in the process industry" *Advances in Industrial Control*. 1995.
- [9] S.J. Qin and T.A. Badgwell. "An overview of industrial model predictive control technology," In J.C. Kantor, C.E. Garcia, and B. Carnahan, editors, *Chemical Process Control - V*, pp. 232-256. 1996.
- [10] WSCC Operations Committee. *Western Systems Coordinating Council Disturbance Report*. WSCC, Sept. 1996.
- [11] US-Canada Power System Outage Task Force. *Final Report on the August 14, 2003 Blackout in the United States and Canada*. April 2004.
- [12] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*. Wiley & Sons, New York, 1996.
- [13] Zimmerman, R. D. and D. Gan. "MATPOWER User's Manual, Version 2.0." PSERC, Dec. 1997. online: [www.pserc.cornell.edu/matpower](http://www.pserc.cornell.edu/matpower).
- [14] P. Hines and S. Talukdar. "Controlling Cascading Failures with Cooperative Autonomous Agents." Submitted to the *International Journal of Critical Infrastructure Systems*. March, 2005.
- [15] M. Larsson and D. Karlsson. "Coordinated system protection scheme against voltage collapse using heuristic search and predictive control." *IEEE Transactions on Power Systems*. Aug. 2003.
- [16] J.Y. Wen, Q.H. Wu, D.R. Turner, S. J. Cheng, and J. Fitch. "Optimal coordinated voltage control for power system voltage stability." *IEEE Transactions on Power Systems*. May 2004.