LOGIC BASED DESIGN OF OPTIMAL RECONFIGURATION STRATEGIES FOR SHIP POWER SYSTEMS

Harry G. Kwatny * **Edoe Mensah** *
Dagmar Niebur * **Gaurav Bajpai** ** **Carole Teolis** **

* *Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA*
** *Techno-Sciences, Inc.,11750 Beltsville Drive Beltsville, MD 20705, USA*

Abstract: A ship electric power system is characterized by a family of nonlinear differential algebraic equations indexed by the discrete modes of system operation. Mixed integer dynamic programming is used to compute optimal mode transition strategies that protect vital loads and maximize the amount of non-vital loads that are supplied.

Keywords: applications, hybrid systems, switching, optimal control, feedback design, nonlinear dynamics

1. INTRODUCTION

In a ship *integrated power system* (IPS), electricity supplies the propulsive power as well as the service loads thereby providing all of the energy needs of the ship. The IPS is particularly attractive because it can be made robust with respect to component failures through management of power flow, loads and energy storage devices.

A challenge is the design of a power management system (PMS). The PMS is intended to manage the discrete actions such as load shedding, cable switching, UPS activity, and the startup or shutdown of prime movers. The main goals are to ensure that power is supplied to vital loads during failure events and to maximize the amount of non-vital load that is supplied while maintaining adequate voltage levels on the load buses.

In this paper we describe the design of optimal feedback PMS. Our approach builds on the work of (Geyer *et al.*, 2002) that applies the framework of (Bemporad and Morari, 1999) to power systems. While interesting results are achieved for a simple power system, as noted in (Geyer *et al.*, 2002), the use of piecewise affine approximations has significant limitations when applied to the nonlinear differential-algebraic-equations (DAE) that describe power systems.

In an earlier publication (Kwatny *et al.*, 2006) we introduced an approach in which we use the nonlinear DAE model and employ mixed-integer dynamic programming to derive an optimal feedback control. In this paper we describe the computations in greater detail, highlighting new concepts for improving the efficiency of the dynamic programming computations. We illustrate the analysis with an example.

Our approach emphasizes the use of logical constraints in formulating the optimal control problem. Logical specifications are used to define the

[1] This research was supported by the Office of Naval Research Contract Number N00014-04-M-0285 and the National Science Foundation Contract Number ECS-0400391.

admissible transition behavior of the discrete system, to incorporate saturation of the continuous control, to characterize the algebraic constraints of the DAE model, and in the definition of the the cost function.

The remainder of the paper is organized as follows. Section 2 gives the problem definition and Section 3 describes an example that is used throughout the remainder of the paper. Section 4 briefly describes the process of converting from logical specifications to mixed integer inequalities, and Section 5 describes our approach to solving the optimization problem.

2. PROBLEM DEFINITION

2.1 Modeling

The system operates in one of m modes denoted $q_1, \ldots, q_m$. $Q = \{q_1, \ldots, q_m\}$ is the discrete state space. The discrete time difference-algebraic equation (DAE) describing operation in mode q_i is

$$\begin{aligned} x_{k+1} &= f_i\left(x_k, y_k, u_k\right) \\ 0 &= g_i\left(x_k, y_k, u_k\right) \end{aligned} \qquad i = 1, \ldots, m \tag{1}$$

where $x \in X \subseteq R^n$ is the system continuous state, $y \in Y \subseteq R^p$ is the vector of algebraic variables and $u \in U \subseteq R^l$ is the continuous control. Transitions can occur only between certain modes. The set of admissible transitions is $\mathcal{E} \subseteq Q \times Q$. It is convenient to view the mode transition system as a graph with elements of the set Q being the nodes and the elements of $\mathcal{E}$ being the edges. We assume that transitions are instantaneous and take place at the beginning of a time interval. So, if a system transitions from mode q_1 to q_2 at time k we would write $q(k) = q_1, q(k^+) = q_2$. We allow resets. State trajectories are assumed continuous through events, i.e., $x(k) = x(k^+)$, unless a reset is specified.

Transitions are triggered by external *events* and *guards*. Events are of two types; either controlled – belonging to set Σ_s, or exogenous (occur spontaneously)–belonging to set Σ_e. A guard is a subset of the continuous state space X that enables a transition. A transition enabled by a guard might represent a protection device. Not all transitions have guards and some transitions might require simultaneous satisfaction of a guard and the occurrence of an event.

We consider each discrete state label, $q \in Q$, and each event, $s \in \Sigma_s$, $e \in \Sigma_e$ to be logical variables that take the values True or False. Guards also are specified as logical conditions. In this way the transition system can be defined by a logical specification (formula) $\mathcal{L}$.

For computational purposes it is useful to associate with each logical variable, say α, a binary variable or indicator function, δ_α, such that δ_α assumes the values 1 or 0 corresponding respectively to α being True or False. It is convenient to define the discrete state vector $\delta_q = [\delta_{q_1}, \ldots, \delta_{q_m}]$. Precisely one of the elements of δ_q will be unity and all others will be zero.

With the introduction of the binary variables we can replace the set of dynamical equations (1) with the single DAE:

$$\begin{aligned} x\left(k+1\right) &= f\left(x\left(k\right), y\left(k\right), \delta_q\left(k\right), u\left(k\right)\right) \\ &= \delta_{q_1} f_{q_1}\left(x\left(k\right), y\left(k\right), u\left(k\right)\right) + \cdots \\ &\quad + \delta_{q_m} f_{q_m}\left(x\left(k\right), y\left(k\right), u\left(k\right)\right) \\ 0 &= g\left(x\left(k\right), y\left(k\right), \delta_q\left(k\right), u\left(k\right)\right) \\ &= \delta_{q_1} g_{q_1}\left(x\left(k\right), y\left(k\right), u\left(k\right)\right) + \cdots \\ &\quad + \delta_{q_m} g_{q_m}\left(x\left(k\right), y\left(k\right), u\left(k\right)\right) \end{aligned} \tag{2}$$

Remark 2.1. (Power System DAE Models). Power systems are typically modeled by sets of semi-explicit DAEs as given by (1) – we will drop the explicit use of u_k to simplify notation. In any mode q_i the flow defined by (1) is constrained to the set $M_i \subset X \times Y$ defined by $0 = g_i(x_k, y_k)$. Ordinarily, it is assumed that M_i is a regular manifold. In rare instances it is possible to explicitly solve $g_i = 0$ for $y_k(x_k)$ and to reduce the DAE to an ordinary differential or difference equation (ODE). More generally, there exists a finite set of disjoint semi-algebraic sets in $X \times Y$ whose union contains M_i and for which there is an explicit characterization of y_k in each set. If the system is not too large, modern quantifier elimination tools enable the identification of these sets and the corresponding functions so that the algebraic condition $g_i = 0$ is replaced by the logical statement:

$$\wedge_{j \in J}\left(\mathcal{L}_j\left(x, y\right) \Rightarrow y = y_j\left(x\right)\right)$$

where each $\mathcal{L}_j$ is a logical specification and J is an index set.

We will do this in the example given below. The advantage of this process is that we can compute the continuous flow in any mode precisely.

2.2 The Control problem

The system is observed in operation over some finite time horizon T that is divided into N discrete time intervals of equal length. A control policy is a sequence of functions

$$\pi = \left\{\mu_0\left(x_0, \delta_{q0}\right), \ldots, \mu_{N-1}\left(x_{N-1}, \delta_{q(N-1)}\right)\right\}$$

such that $[u_k, \delta_{sk}] = \mu_k\left(x_k, \delta_{q_k}\right)$. Thus, μ_k generates the continuous control u_k and the discrete control δ_{sk} that are to be applied at time k, based on the state (x_k, δ_{qk}) observed at time k.

Consider the set of m-tuples $\{0,1\}^m$. Let Δ_m denote the subset of elements $\delta \in \{0,1\}^m$ that satisfy $\delta_1 + \cdots + \delta_m = 1$. Denote by Π the set of sequences of functions $\mu_k : X \times \Delta_m \to U \times \{0,1\}^{m_S}$ that are piecewise continuous on X.

The *Optimal Feedback Control Problem* is defined as follows. For each $x_0 \in X, \delta_{q0} \in \Delta_m$ determine the control policy $\pi^* \in \Pi$ that minimizes the cost

$$\begin{aligned} J_\pi\left(x_0, \delta_{q0}\right) = \\ g_N\left(x_N, \delta_{qN}\right) + \\ \sum_{k=0}^{N-1} g_k\left(x_k, \delta_{qk}, \mu_k\left(x_k, \delta_{qk}\right)\right) \end{aligned} \tag{3}$$

subject to the constraints (1) and the logical specification, i.e.,

$$J_{\pi^*}\left(x_0, \delta_{q0}\right) \leq J_\pi\left(x_0, \delta_{q0}\right) \quad \forall \pi \in \Pi \tag{4}$$

3. EXAMPLE

3.1 System Description

A relatively simple system that is known to exhibit interesting voltage stability characteristics is a single generator feeding an aggregated load composed of constant impedance loads and induction motors (Pal, 1993). By expanding this system to include a vital load with a UPS, as shown in Figure 1, we obtain one of interest to us.

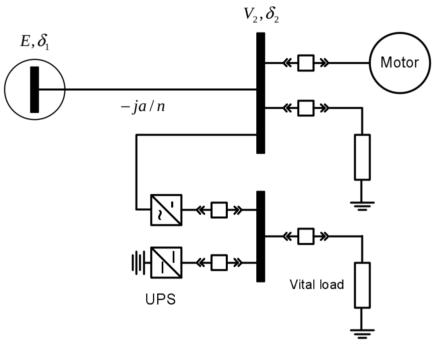

Fig. 1. System with UPS.

The primary means for voltage control is the field voltage. However, in the event of a transmission line fault it may be necessary to shed load in order to avoid a system collapse. This can be accomplished by dropping non-vital load in discrete blocks and, if necessary switching the vital load to battery supply.

We will assume that two blocks of non-vital load can be dropped independently by opening circuit breakers. Correspondingly, we use a load shed parameter $\eta \in \{0, \eta_1, \eta_2\}$ that denotes the fraction of load dropped.

The battery is connected to the DC load bus through a DC-DC converter. There are three possible UPS operating modes:

(1) Battery unconnected.
(2) Battery discharging; The battery and vital load are detached from the rest of the network. The battery supplies the load through a voltage controlled DC-DC converter set up to keep the load voltage constant.
(3) Battery charging; In this mode the battery is charged through a DC-DC converter operated in current controlled mode – the current is controlled to a specified value.

The overall system transition system is shown in Figure 2. It represents operational constraints that we impose on the system.

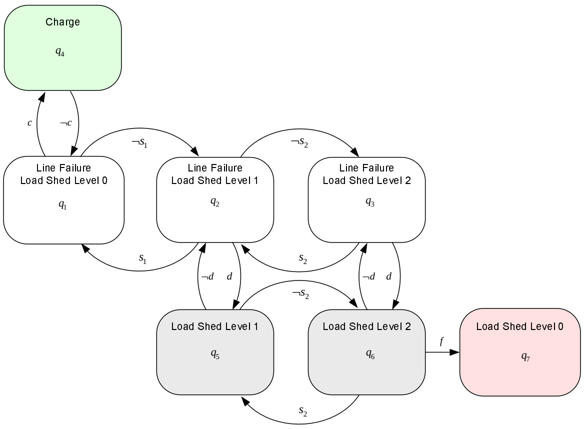

Fig. 2. Transition behavior for system with UPS.

3.2 Dynamics

3.2.1. Battery disconnected, modes q_1, q_2, q_3 The voltage regulated rectifier controls the voltage on vital load bus.We assume that the rectifier is power factor corrected so that from the AC side of the rectifier, the vital load looks like a constant power load with unity power fact, $P = P_v$, $Q = 0$.

Let δ_1, δ_2 denote the voltage angles at bus 1 and 2. Define the relative angle $\theta_2 = \delta_2 - \delta_1$. The network equations are

$$\begin{aligned} P_v = a\,E\,V_2 \sin\theta_2 - c\,V_2^2 \\ 0 = a\,E\,V_2 \cos\theta_2 + d\,V_2^2 \end{aligned} \tag{5}$$

Where P_v is the power consumed by the vital load and $c - j\,d$ is the admittance of the non-vital aggregate load.

The field voltage E is used to control the load bus voltage V_2 to its desired nominal value of 1. If we ignore the exciter dynamics, then (5) allows the determination of the field voltage that yields the desired load bus voltage provided the resultant E is within its strict limits, $0 \leq E \leq 2$. It is always the upper limit that is the binding constraint. This implies two possibilities for satisfying (5): either $V_2 = 1$ or $E = 2$. These are:

$$V_2 = 1,\; E = \frac{\sqrt{\left(c + P_v\right)^2 + d^2}}{a}, \quad 0 < P_v \tag{6}$$

$$E = 2,$$
$$V_2 = \sqrt{\frac{2a^2 - cP_v - \sqrt{4a^4 - 4a^2cP_v - d^2P_v^2}}{c^2 + d^2}},$$
$$0 < P_v < 2a^2\left(\sqrt{c^2 + d^2} - c\right) \tag{7}$$

Once the excitation system saturates there is an upper limit to P_v, as seen in (7). This is the voltage collapse bifurcation point. Also, these relations are only good for $P_v > 0$. When $P_v = 0$ we have

$$V_2 = \frac{a}{\sqrt{c^2 + d^2}} E \tag{8}$$

Equation (6) (non-saturated field) does approach the proper limit as $P_v \to 0$, but the Equation (7) (saturated field) does not. This is as it should be.

Remark 3.2. (Network Solution). As discussed in Remark 2.1 we can express the network constraints in terms of the logical constraint

$$\mathcal{L}_0 = (V_2 = 1 \Rightarrow E = z_1) \wedge (E = 2 \Rightarrow V_2 = z_2) \tag{9}$$

where z_1, z_2 are defined via (6), (7), and (8).

3.2.2. Battery Charging, mode q_4 The battery model is composed of a differential equation describing the battery 'state of charge' σ and an output map that gives the battery terminal voltage v_b as a function of the state of charge.

$$\frac{d}{dt}\sigma = \frac{1}{C} i, \; v_b = f(\sigma), \; 0 \le \sigma \le 1$$

where i is the battery charging current and C is the battery effective capacitance. The DC-DC converter operates in current control mode so the battery is charged with constant current, $i = i_c$. While charging we have:

$$\frac{d\sigma}{dt} = \frac{i_c}{C}$$

Because the AC-DC rectifier maintains constant V_3, from the AC side of the rectifier, charging looks like an additional constant power load, $P_c = V_3 i_c$. The network supplies both the vital load and the power to charge the battery. Thus, the network relation is given by Equations (6) and (7) with P_v replace by $P_v + P_c$.

3.2.3. Battery Discharging, modes q_5, q_6 The vital loads and battery are separated from the rest of the system and draw no power from the network. Consequently the the relationship between E and V_2 is given by Equation (8). The DC-DC converter now maintains constant voltage on bus 3, so that the battery current is $i = -P_v/V_3$ and

$$\frac{d\sigma}{dt} = -\frac{P_v}{C\,V_3}$$

In the following study we take $C = 0.5$ and $P_v = 10$.

3.2.4. Induction Motors If we neglect the small stator resistance and inductance and assume a large magnetizing inductance, the equivalent circuit for an induction motor consists of a series rotor resistance and inductance R_r, X_r. Define the slip $s = (\omega_0 - \omega_m)/\omega_0$ and let P_m denote the mechanical load power. Then the motor dynamics take the form

$$\dot{s} = \frac{1}{I_m\omega_0^2}\left(P_m - V_s^2 \frac{R_r s(1-s)}{R_r^2 + s^2X_r^2}\right) \tag{10}$$

3.2.5. Load Shedding We assume discrete load shedding blocks and define η to represent the fraction of load shed. Thus η can assume a finite number of values $0 \le \eta < 1$. The non-vital load admittances, taking into account the load shedding parameter, are:

$$c = (1-\eta)\,c_0, \; c_0 = \left(\frac{1}{R_L} + \frac{R_r s}{R_r^2 + s^2X_r^2}\right) \tag{11}$$

$$d = (1-\eta)\,d_0, \; d_0 = \left(\frac{X_r s^2}{R_r^2 + s^2X_r^2}\right) \tag{12}$$

Equation (10) represents the aggregated motor dynamics, and the load admittance is given by the last two equations, (11), (12). The system data is $R_L = 2, \; R_r = 0.25, \; X_r = 0.125, \; a = 1$ (nominal) , $I_m\omega_0^2 = 4$.

4. LOGICAL SPECIFICATION TO IP FORMULAS

The first step in solving the optimal control problem is to transform the logical constraints into a set of inequalities involving binary variables and possibly real variables, so-called *IP-formulas.* The idea of formulating optimization problems using logical constraints and then converting them to IP formulas has a long history. This concept was used as a means to incorporate qualitative information in process control (Tyler and Morari, 1999), and generally introduced into the study of hybrid systems in (Bemporad and Morari, 1999).

McKinnon and Williams (1989) suggested a sequence of transformations that brings a logical specification into a set of IP-formulas. Li *et al.* (2000) present a systematic algorithm for doing this. We have modified that implementation in order to obtain simpler and more compact IP formulas. [2]

If all of the guards are linear (set boundaries are composed of linear segments), then the IP formulas form a system of linear constraints involving the binary variables $\delta_q, \delta_{q^+}, \delta_s$, respectively, the

[2] A *Mathematica* package containing the transformation functions and the examples contained herein is available at http://www.pages.drexel.edu/ hgk22/Hybrid.htm

discrete state before transition, the discrete state after transition, the exogenous events. They also involve a set of auxiliary binary variables, d, introduced during the transformation process, and the continuous state variables, x. With x, δ_q, δ_s given, these inequalities typically provide a unique solution for the unknowns δ_{q^+} and d.

Example 4.3. (IP Formulas for UPS System). Four logical constraints need to be converted to IP formulas:

(1) the network specification, $\mathcal{L}_0$, Equation (9)
(2) the transition specification, $\mathcal{L}_1$, of Figure 2
(3) the excitation shedding specification

$$\mathcal{L}_2 = (V_2 = 1 \wedge 0 < E < 2) \vee (E = 2)$$

(4) the load shedding specification

$$\mathcal{L}_3 = (q_1^+ \Rightarrow \eta = 0) \wedge (q_2^+ \Rightarrow \eta = 0.4) \wedge (q_3^+ \Rightarrow \eta = 0.8)$$

The corresponding IP formulas are generated automatically. We don't display them here because of space limitations. All of the inequalities derived from $\mathcal{L}_1$ involve only binary variables while some of those derived from $\mathcal{L}_0$, $\mathcal{L}_2$ and $\mathcal{L}_3$ involve both binary and real variables. The latter also contain auxiliary binary variables d_i introduced during the conversion process. All of the inequalities are linear in all variables.

5. CONSTRUCTING THE OPTIMAL SOLUTION

We apply Bellman's principle of optimality: suppose $\pi^* = \{\mu_1^*, \ldots, \mu_{N-1}^*\}$ is an optimal control policy. Then the sub-policy $\pi_i^* = \{\mu_i^*, \ldots, \mu_{N-1}^*\}$, $1 \le i \le N-1$ is optimal with respect to the cost function (3).

Let us denote the optimal cost of the trajectory beginning at x_i, δ_{qi} as $J_i^*(x_i, \delta_{qi})$. It follows from the principle of optimality that

$$J_{i-1}^*\left(x_{i-1}, \delta_{q(i-1)}\right) = \min_{\mu_{i-1}} \left\{ g_{i-1}\left(x_{i-1}, \delta_{q(i-1)}, \mu_{i-1}\right) + J_i^*(x_i, \delta_{qi}) \right\} \tag{13}$$

Equation (13) provides a mechanism for backward recursive solution of the optimization problem. To begin the backward recursion, we need to solve the single stage problem with $i = N$. The end point x_N, δ_{qN} is free, so we begin at a general terminal point

$$J_{N-1}^*\left(x_{N-1}, \delta_{q(N-1)}\right) = \min_{\mu_{N-1}} \left\{ \begin{array}{c} g_{N-1}\left(x_{N-1}, \delta_{q(N-1)}, \mu_{N-1}\right) + \\ g_N\left(f_{N-1}, \delta_{q^+(N-1)}\right) \end{array} \right\} \tag{14}$$

Once the pair μ_{N-1}^*, J_{N-1}^* is obtained, we compute μ_{N-2}^*, J_{N-2}^*. Continuing in this way we obtain

$$J_{N-i}^*\left(x_{N-i}, \delta_{q(N-i)}\right) = \min_{\mu_{N-i}} \left\{ \begin{array}{c} g_{N-i}\left(x_{N-i}, \delta_{q(N-i)}, \mu_{N-i}\right) \\ +J_{N-i+1}^*\left(f_{N-i}, \delta_{q^+(N-i)}\right) \end{array} \right\} \tag{15}$$

for $2 \le i \le N$.

We need to solve (15) recursively backward, for $i = 2, \ldots, N$ after initializing with (14). We begin by constructing a discrete grid on the continuous state space. The discrete space is denoted $\bar{X}$. At each iteration the optimal control and the optimal cost are evaluated at discrete points in $Q \times \bar{X}$. To continue with the next stage we need to set up an interpolation function to cover all points in $Q \times X$.

We exploit the fact that the system is highly constrained and all of the constraints are linear in binary variables. The basic approach is as follows:

(1) Before beginning the time iteration:
  (a) Separate the inequalities into binary and real sets, binary formulas contain only binary variables, real formulas can contain both binary and real variables.
  (b) For each $q \in Q$, obtain all feasible solutions of the binary inequalities; a list of possible solutions of pairs $\left(\delta_{q^+}, d\right)$.
  (c) Define projection $\bar{X} \to \bar{X}_{\mathcal{P}}$ where $\bar{X}_{\mathcal{P}}$ is the subspace of real states actually appearing in the real equations.
  (d) For each $x_{\mathcal{P}} \in \bar{X}_{\mathcal{P}}$
    (i) pre-screen the binary solutions to eliminate those that do not produce solutions to the real inequalities - typically a very large fraction is dropped
    (ii) for every feasible combination of binary variables obtained above, solve the real inequalities for the real variables
  (e) Lift real solutions to entire $\bar{X}$.
(2) For each i,
  (a) For each pair $(q, x) \in Q \times \bar{X}$
    (i) enumerate the values of the cost to go using the feasible sets of binary and real variables
    (ii) select the minimum

In step 1b above the number of solutions corresponding to each q can be very large because there are numerous redundant solutions associated with nonactive transitions. Thus, we add additional logical constraints that specify the inactive transitions. Step 1c exploits the fact that some real states do not appear in the real formulas. Because a large fraction of the binary solutions do not lead to real solutions, the pre-screening in step 1d.i is very effective in reducing computing time. Finally, we note that the inequalities are independent of the stage of the dynamic programming recursion. Thus, step 1d, which is by far the most intensive computational element of the optimization is done only once before the recursion step 2a begins.

Example 5.4. (Optimal Control). We seek an optimal control policy that minimizes the cost function

$$J = \sum_{k=0}^{N-1} \left(\begin{array}{c} \|V_2(k) - 1\|^2 + r_0 \|\sigma - 1\|^2 \\ +r_1 \|\eta_L(k)\|^2 \end{array} \right)$$

subject to the system constraints. In the following we take $r_0 = 1, r_1 = 1/25$.

We describe the optimal controller for a line fault that results in a line admittance of $a = 0.375$. This is a severe fault, but one that is manageable. The state space includes the 7 discrete states (modes) and two continuous states induction motor slip, s, indicative of power, and battery state, σ, that represents the fractional battery charge. For computational purposes, the continuous state is discretized $s \in \{.1, .2, .3, .4, .5\}$ and $\sigma \in \{.25, .5, .75.1.0\}$, and the feedback control is computed in terms of these 140 states. In implementation an interpolation function is used for the continuous states. [3] The optimal solution with a 25 step horizon is computed in about 9 minutes on laptop with a 1.1 Ghz Pentium M. It takes only a few seconds longer for a 50 step horizon because the most difficult computations are done only once.

Figures 3, 4 and 5 illustrate a particular feedback trajectory in which the initial battery state of charge is 0.1 and the initial slip is 0.

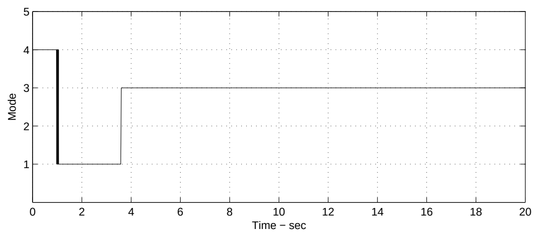

Fig. 3. Because of the low battery charge an initial switch into charging mode 4 occurs before load is dropped, modes 2 and 3.

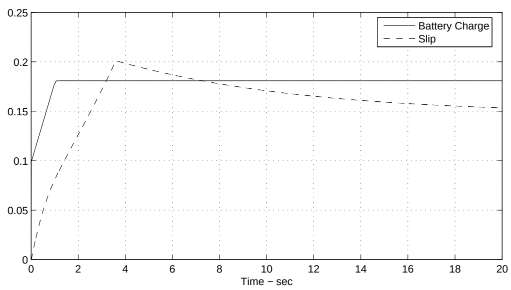

Fig. 4. The battery initially charges, but increasing slip, and hence electrical power, eventually requires load shedding.

6. CONCLUSIONS

We have described an approach to designing optimal feedback reconfiguration strategies for a class of power systems. The essential feature of our approach is the use of logical specifications to characterize various system constraints. Our approach to solving the mixed integer dynamic programming problem that results is described. An example is given that illustrates the problem of optimal load shedding as a means of responding to line faults.

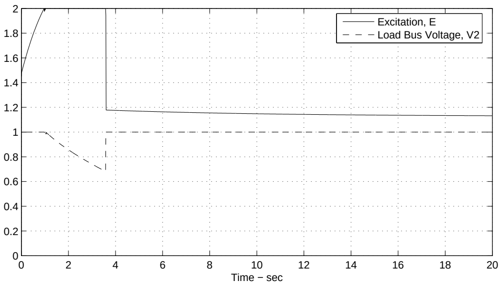

Fig. 5. After about 1 second the excitation saturates and load bus voltage drops. Load voltage regulation is re-established following load shedding.

REFERENCES

Bemporad, Alberto and Manfred Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.

Geyer, T., M. Larsson and M. Morari (2002). Hybrid control of voltage collapse in power systems. Technical Report AUTTe-02-0461. Automatic Control laboratory, ETH Zentrum.

Kwatny, H. G., E. Mensah, D. Niebur and C. Teolis (2006). Optimal power system management via mixed integer dynamic programming. In: *2006 IFAC Symposium on Power plants and Systems*. Kananaskis, Canada.

Li, Q., Y. Guo and T. Ida (2000). Modelling integer programming with logic: Language and implementation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **E83-A**(8), 1673–1680.

McKinnon, K. and H. Williams (1989). Constructing integer programming models by the predicate calculus. *Annals of Operations Research* **21**, 227–246.

Pal, M. K. (1993). Voltage stability: Analysis needs, modelling requirement, and modelling adequacy. *IEE Proceedings - C* **140**(4), 279–286.

Tyler, M. L. and M. Morari (1999). Propositional logic in control and monitoring problems. *Automatica* **35**(4), 565–582.

[3] Because of space limitations we omit simulations. In our work we simulate in SIMULINK for which the controller table data structure is automatically generated.