

Research Article

Inverse Optimal Control with Speed Gradient for a Power Electric System Using a Neural Reduced Model

Alma Y. Alanis,¹ Enrique A. Lastire,² Nancy Arana-Daniel,¹ and Carlos Lopez-Franco¹

¹ CUCEI, Universidad de Guadalajara, Apartado Postal 51-71, Col. Las Aguilas, 45079 Zapopan, JAL, Mexico

² CINVESTAV, Unidad Guadalajara, Apartado Postal 31-438, Plaza La Luna, 45091 Guadalajara, JAL, Mexico

Correspondence should be addressed to Alma Y. Alanis; almayalanis@gmail.com

Received 5 November 2013; Revised 30 January 2014; Accepted 30 January 2014; Published 16 March 2014

Academic Editor: Hamid R. Karimi

Copyright © 2014 Alma Y. Alanis et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presented an inverse optimal neural controller with speed gradient (SG) for discrete-time unknown nonlinear systems in the presence of external disturbances and parameter uncertainties, for a power electric system with different types of faults in the transmission lines including load variations. It is based on a discrete-time recurrent high order neural network (RHONN) trained with an extended Kalman filter (EKF) based algorithm. It is well known that electric power grids are considered as complex systems due to their interconnections and number of state variables; then, in this paper, a reduced neural model for synchronous machine is proposed for the stabilization of nine bus system in the presence of a fault in three different cases in the lines of transmission.

1. Introduction

Many physical systems, such as electric power grids, computer and communication networks, networked dynamical systems, transportation systems, and many others, are complex large-scale interconnected systems [1]. To control such large scale systems, centralized control schemes are proposed in the literature assuming available global information for the overall system. Another problem in complex large-scale interconnected systems is the effect of delays that typically are unknown and time-variable [2, 3]. While using control centralization has theoretical advantages, it is very difficult for a complex large-scale system with interconnections due to technical and economic reasons [4]. Furthermore, centralized control designs are dependent upon the system structure and cannot handle structural changes. If subsystems are added or removed, the controller for the overall system should be redesigned. Therefore decentralized control for interconnected power systems has also attracted considerable attention of researchers in the field of complex and large-scale systems like multiarea interconnected power systems. Besides, due to physical configuration and high dimensionality of interconnected systems, centralized control is neither economically feasible nor even necessary. These facts

motivate the design of decentralized controllers, using only local information while guaranteeing stability for the whole system [1].

The main issue in this paper is the analysis of a fault in the electric power system in different lines of transmission, the recurrent high order neural networks (RHONN) allow the identification of nonlinear systems, and then the RHONN model can be used for the controller design. Recently, some works have been published about synchronous generators in which reduced models have been proposed, such models are able to reproduce full order dynamics for synchronous generators [1, 5]. The system under study consists of three synchronous generators interconnected (nine bus system) and there are cases of study of power electric system, where a three-phase fault is introduced at the end of the line 7 [6]; in this paper, the analysis for the system is focused in other lines, at the end of buses 8 and 9, the fault is proposed and tested via simulation and the purpose is the production and distribution of a reliable and robust electric energy.

On the other hand, a model in discrete time has been proposed [7], in which a recurrent high order neural network has been incorporated to implement a control law as this reduced model allows the stabilization through the inverse optimal control law SG. In this work, a neural model of

the multimachine system is proposed, which results useful, because it is focused in the variable states that are more relevant for this paper: position, velocity, and voltage rotor [7]; further, the control law is implemented for the power electric system that consists of three interconnected synchronous generators. A solution is proposed for the destabilization problem of multimachine power electric system in the presence of a fault in one of its lines of transmission that occurs at 10 seconds of simulation. A system identification of the complete multimachine power electric system model (nine bus system) is presented through a neural reduced model and this allows the design of a neural inverse optimal SG control law. Finally the results obtained are shown, in which it can be seen that the control law stabilizes the system in presence of the fault in the three cases of fault that are presented.

In literature, there are works that report the parameter identification for synchronous machines for full order models [5] as well as for reduced order ones [8]; however, these models are for nominal condition; that is, they do not consider fault scenarios; in [1], a reduced order neural model is considered; however, it is developed for continuous time; nevertheless, the need to real-time implementations makes necessary the use of digital models, besides, in [9], has been developed a discrete-time neural controller, which is proposed for a single machine system. Then, the paper main contributions can be stated as follows: first a RHONN is used to establish a discrete-time reduced order mathematical model for a multimachine power electric system model. Then this neural model is used to synthesize an inverse optimal SG control law to stabilize the system and, finally, three fault scenarios are considered in order to illustrate the applicability of the proposed scheme.

2. Mathematical Preliminaries

2.1. Discrete-Time High Order Neural Networks. The use of multilayer neural networks is well known for pattern recognition and for static systems modelling. The NN is trained to learn an input-output map. Theoretical works have proven that, even with just one hidden layer, a NN can uniformly approximate any continuous function over a compact domain, provided that the NN has a sufficient number of synaptic connections [10]. To implement the neural network (NN) design, a RHONN is used [7] and this model turns out to be very flexible because it allows incorporating priori information to the model:

$$\hat{x}_i(k+1) = \omega_i^T z_i(x_i(k), u(k)), \quad i = 1, \dots, n, \quad (1)$$

where \hat{x}_i ($i = 1, 2, \dots, n$) is the state of the i th neuron and ω_i ($i = 1, 2, \dots, n$) is the respective online adapted weight vector. Now we define the vector:

$$z_i(x_i(k), u(k)) = \begin{bmatrix} z_{i_1} \\ z_{i_2} \\ \vdots \\ z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} \xi_{i_j}^{d_{i_j}(1)} \\ \prod_{j \in I_2} \xi_{i_j}^{d_{i_j}(2)} \\ \vdots \\ \prod_{j \in I_{L_i}} \xi_{i_j}^{d_{i_j}(L_i)} \end{bmatrix}. \quad (2)$$

L_i is the respective number of high-order connections, $\{I_1, I_2, \dots, I_{L_i}\}$ is a collection of nonordered subsets of $\{1, 2, \dots, n+m\}$, n is the state dimension, and m is the number of external inputs, with $d_i(k)$ being nonnegative integers and ξ_i defined as follows:

$$\xi_i = \begin{bmatrix} \xi_{i_1} \\ \vdots \\ \xi_{i_n} \\ \xi_{i_{n+1}} \\ \vdots \\ \xi_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ S(u_m) \end{bmatrix}. \quad (3)$$

$u = [u_1, u_2, \dots, u_m]^T$ is the input vector to the neural network and $S(\bullet)$ is defined by

$$S(\varsigma) = \frac{1}{1 + \exp(-\beta\varsigma)}, \quad \beta > 0, \quad (4)$$

where ς is any real value variable.

2.2. The EKF Training Algorithm. The best well-known training approach for recurrent neural networks (RNN) is the backpropagation through time learning [11]. However, it is a first order gradient descent method and hence its learning speed could be very slow [12]. Recently, Extended Kalman Filter (EKF) based algorithms have been introduced to train neural networks [7, 9, 13, 14]. With the EKF based algorithm, the learning convergence is improved [14]. The EKF training of neural networks, both feedforward and recurrent ones, has proven to be reliable and practical for many applications over the past years [14]. It is known that Kalman filtering (KF) estimates the state of a linear system with additive state and output white noises [15, 16]. For KF-based neural network training, the network weights become the states to be estimated. In this case, the error between the neural network output and the measured plant output can be considered as additive white noise. Due to the fact that the neural network mapping is nonlinear, an EKF-type is required (see [17] and references therein). The training goal is to find the optimal weight values which minimize the prediction error. The EKF-based training algorithm is described by [15]:

$$\begin{aligned} K_i(k) &= \rho_i(k) H_i(k) M_i^{-1}(k) \\ \omega_i(k+1) &= \omega_i(k) + \eta_i K_i(k) e_i(k) \end{aligned} \quad (5)$$

$$\rho_i(k+1) = \rho_i(k) - K_i(k) H_i^T(k) \rho_i(k) + \phi_i(k)$$

with

$$\begin{aligned} M_i(k) &= [\tau_i(k) + H_i^T(k) \rho_i(k) H_i(k)]^{-1} \\ e_i(k) &= x_i(k) - \hat{x}_i(k), \end{aligned} \quad (6)$$

where $\rho_i \in \mathfrak{R}^{L_i \times L_i}$ is the prediction error associated covariance matrix, $\omega_i \in \mathfrak{R}^{L_i}$ is the weight (state) vector, $x_i \in \mathfrak{R}$ is the i th plant state component, $\hat{x}_i \in \mathfrak{R}$ is the i th neural

state component, η_i is a design parameter, $K_i \in \mathbb{R}^{L_i \times m}$ is the Kalman gain matrix, $\phi_i \in \mathbb{R}^{L_i \times L_i}$ is the state noise associated covariance matrix, $\tau_i \in \mathbb{R}^{m \times m}$ is the measurement noise associated covariance matrix, and $H_{ij} \in \mathbb{R}^{L_i \times m}$ is a matrix, for which each entry (H_{ij}) is the derivative of one of the neural network output, (\hat{x}_{ij}), with respect to one neural network weight, (ω_{ij}), as follows:

$$H_{ij}(k) = \left[\frac{\partial \hat{x}_i(k)}{\partial \omega_{ij}(k)} \right]_{\omega_i(k) = \hat{\omega}_i(k+1)} \quad i = 1, \dots, n, \quad j = 1, \dots, L_i. \quad (7)$$

Usually ρ_i , ϕ_i , and τ_i , are initialized as diagonal matrices, with entries $\rho_i(0)$, $\phi_i(0)$, and $\tau_i(0)$, respectively.

3. Controller Design

Optimal control is related to finding a control law for a given system such that a performance criterion is minimized. This criterion is usually formulated as a cost functional, which is a function of the state and control variables. The optimal control problem can be solved using Pontryagin's maximum principle (a necessary condition) [18] and the method of dynamic programming developed by Bellman [19, 20], which can lead to a nonlinear partial differential equation called the Hamilton-Jacobi-Bellman (HJB) equation (a sufficient condition); nevertheless, solving the HJB equation is not a feasible task [21, 22].

3.1. Inverse Optimal Control via CLF. In this paper, the inverse optimal control and its solution by proposing a quadratic control Lyapunov function (CLF) are used [23] and the CLF depends on a fixed parameter in order to satisfy stability and optimality condition. A posteriori, the speed gradient algorithm is established to compute this CLF parameter and it is used to solve the inverse optimal control problem. Motivated by the favorable stability margins of optimal control systems, a stabilizing feedback control law is proposed, which will be optimal with respect to a meaningful cost functional. At the same time, it is desirable to avoid the difficult task of solving the HJB partial differential equation. In the inverse optimal control problem, a candidate CLF is used to construct an optimal control law directly without solving the associated HJB equation [24]. Inverse optimality is selected, because it avoids solving the HJB partial differential equations and still allows obtaining Kalman-type stability margins [21].

In contrast to the inverse optimal control via passivity approach, in which a storage function is used as a candidate CLF and the inverse optimal control law is selected as the output feedback, for the inverse optimal control via CLF, the control law is obtained as a result of solving the Bellman equation. Then, a candidate CLF for the obtained control law is proposed such that it stabilizes the system and a posteriori a meaningful cost functional is minimized.

In this paper, a quadratic candidate CLF is used to synthesize the inverse optimal control law. The following

assumptions and definitions allow the inverse optimal control solution via the CLF approach.

The full state of system

$$x(k+1) = f(x(k)) + g(x(k))u(k) \quad (8)$$

is measurable.

Definition 1 (inverse optimal control law). Let us define the control law [23]

$$u(k) = -\frac{1}{2}R^{-1}(x(k))g^T(x(k))\frac{\partial V(x(k+1))}{\partial x(k+1)} \quad (9)$$

to be inverse optimal (globally) stabilizing if

- (1) it achieves (global) asymptotic stability of $x = 0$ for system (8);
- (2) $V(x(k))$ is (radially unbounded) positive definite function such that inequality

$$\bar{V} := V(x(k+1)) - V(x(k)) + u(k)^T R(x(k))u(k) \leq 0 \quad (10)$$

is satisfied. When $l(x(k)) := \bar{V} \geq 0$, is selected; then $V(x(k))$ is a solution for the HJB equation

$$l(x(k) + V(x(k+1)) - V(k)) + \frac{1}{4}V^{T*}R^{-1}(x(k))g^T(x(k))V^* = 0, \quad (11)$$

where

$$V^{T*} = \frac{\partial V^T(x(k+1))}{\partial x(k+1)}, \quad V^* = \frac{\partial V(x(k+1))}{\partial x(k+1)}. \quad (12)$$

It is possible to establish the main conceptual differences between optimal control and inverse optimal control as follows.

- (i) For optimal control, the meaningful cost indexes $l(x(k)) \geq 0$ and $R(x(k)) > 0$ are given a priori; then, they are used to calculate $u(x(k))$ and $V(x(k))$ by means of HJB equation solution.
- (ii) For inverse optimal control, a candidate CLF $V(x(k))$ and the meaningful cost index $R(x(k))$ are given a priori, and then these functions are used to calculate the inverse control law $u(k)$ and the meaningful cost index ($x(k)$), defined as $l(x(k)) := -\bar{V}(x(k))$.

As established in Definition 1, the inverse optimal control problem is based on the knowledge of $V(x(k))$. Thus, it is proposed as a CLF $V(x(k))$, such that (1) and (2) are guaranteed. That is, instead of solving (11) for $V(x(k))$, it is proposed a control Lyapunov function $V(x(k))$ as

$$V(x(k)) = \frac{1}{2}x^T(k)Px(k) \quad (13)$$

for control law (9), in order to ensure stability of the equilibrium point $x(k) = 0$ of system (8), which will be

achieved by defining an appropriate matrix P . Moreover, it will be established that control law (9) with (13), which is referred to as the inverse optimal control law, optimizes a meaningful cost functional of the form:

$$J(x(k)) = \sum_0^{\infty} (l(x(k)) + u^T(k) R(x(k)) u(k)). \quad (14)$$

Consequently, by considering $V(x(k))$ as in (13), the control law takes the following form:

$$\alpha(x(k)) := u(k) = -\frac{1}{2}(R(x(k)) + P_2(x(k)))^{-1}P_1(x(k)), \quad (15)$$

where $P_1(x(k)) = g^T(x(k))Pf(x(k))$ and $P_2(x(k)) = (1/2)g^T(x(k))Pg(x(k))$. It is worth pointing out that P and $R(x(k))$ are positive definite and symmetric matrices; thus, the existence of the inverse in (15) is ensured.

3.2. Speed-Gradient SG Algorithm. Given that (15) P is redefined as $P(k)$ where $P_1(x(k)) = g^T(x(k))P(k)f(x(k))$ and $P_2(x(k)) = (1/2)g^T(x(k))P(k)g(x(k))$, this will allow us to compute a time variant value in time for $P(k)$, which ensures stability to the system (8) by means of the algorithm SG.

In [25] a discrete-time application of the SG algorithm is formulated to find a control law $u(k)$ which ensures the control goal:

$$Q(x(k+1)) \leq \Delta, \quad \text{for } k \geq k^*, \quad (16)$$

where Q is a control goal function, a constant $\Delta > 0$, and $k^* \in \mathbb{Z}^+$ is the time at which the control goal is achieved. Q ensures stability if it is a positive definite function.

Based on the SG application proposed in [25], the control law given by (15) is considered, with Δ in (16) a state dependent function $\Delta(x(k))$.

Consider the control law redefined for the speed gradient algorithm which at every time depends on the matrix $P(k)$. Let us define the matrix $P(k)$ at every time k as

$$P(k) = p(k)P', \quad (17)$$

where $P' = P'^T > 0$ is a given constant matrix and $p(k)$ is a scalar parameter to be adjusted by the SG algorithm. Then the control law is transformed as follows:

$$u(k) = -\frac{P(k)}{2} \left(R(x(k)) + \frac{P(k)}{2} P_1^* \right)^{-1} P_2^*, \quad (18)$$

where

$$P_1^* = g^T(x(k))P'g(x(k)), \quad P_2^* = g^T(x(k))P'f(x(k)). \quad (19)$$

The SG algorithm is now reformulated for the inverse optimal control problem.

Definition 2 (SG goal function). Consider a time-varying parameter $p(k) \in \mathcal{P} \subset \mathfrak{R}^+$ with $p(k) > 0$ for all k , and \mathcal{P}

is the set of admissible values for $p(k)$ [23]. A nonnegative function $Q: \mathfrak{R}^n \times \mathfrak{R} \rightarrow \mathfrak{R}$ of the form

$$Q(x(k), p(k)) = V_{SG}(x(k+1)), \quad (20)$$

where $V_{SG}(x(k+1)) = -(1/2)x^T(k+1)P'x(k+1)$ with $x(k+1)$ as defined in (8), is referred to as SG goal function for system (8), with $Q(k(p)) := Q(x(k), p(k))$.

Definition 3 (SG control goal). Consider a constant $p^* \in \mathcal{P}$. The SG control goal for system (8) with (18) is defined as finding $p(k)$, so that the SG goal function $Q(k(p))$ [23], as in (20), fulfills

$$Q(k(p)) \leq \Delta(x(k)), \quad \text{for } k \geq k^*, \quad (21)$$

where

$$\Delta(x(k)) = V_{SG}(x(k)) - \frac{1}{p(k)}u^T(k)R(x(k))u(k) \quad (22)$$

with $V_{SG}(x(k)) = -(1/2)x^T(k)P'x(k)$ and $u(k)$ as defined in (18); $k^* \in \mathbb{Z}^+$ is the time at which the SG control goal is achieved.

Solution $p(k)$ must guarantee that $V_{SG}(x(k)) > (1/p(k))u^T(k)R(x(k))u(k)$ in order to obtain a positive definite function $\Delta(x(k))$.

To conclude, the SG algorithm is used to calculate $p(k)$ in order to achieve the SG control goal defined above.

Proposition 4. Consider a discrete-time nonlinear system of the form (8) with (18) as input [23]. Let Q be a SG goal function as defined in (2) and denoted by $Q(k(p))$. Let $\bar{p}, p^* \in \mathcal{P}$ be positive constant values and let $\Delta(x(k))$ be a positive definite function with $\Delta(0) = 0$ and let ϵ^* be a sufficiently small positive constant. Assume the following.

(i) There exist p^* and ϵ^* such that

$$Q(k(p^*)) \leq \epsilon^* \ll \Delta(x(k)), \quad (23)$$

$$\frac{1 - \epsilon^*}{\Delta(x(k))} \approx 1.$$

(ii) For all $p(k) \in \mathcal{P}$,

$$(p^* - P(k))^T \nabla(p) Q(k(p)) \leq \epsilon^* - \Delta(x(k)) < 0, \quad (24)$$

where $\nabla(p)Q(k(p))$ denotes the gradient of $Q(k(p))$ with respect to $p(k)$. Then, for any initial condition $p(0) > 0$, there exists a $k^* \in \mathfrak{R}^+$ such that the SG control goal (16) is achieved by means of the following dynamic variation of parameter $p(k)$:

$$p(k+1) = p(k) - \gamma_{d(k)} \nabla(p) Q(k(p)) \quad (25)$$

with

$$\gamma_{d(k)} = \gamma_c \delta(k) |\nabla(p) Q(k(p))|^{-2}$$

$$0 < \gamma_c \leq 2\Delta(x(k)), \quad (26)$$

$$\delta(k) = \begin{cases} 1 & \text{for } Q(p(k)) > \Delta(x(k)) \\ 0 & \text{otherwise.} \end{cases}$$

Finally, for $k \geq k^*$, $p(k)$ becomes a constant value denoted by \bar{p} and the SG algorithm is completed.

With $Q(p(k))$ as defined in (20), the dynamic variation of parameter $p(k)$ in (25) results in

$$p(k+1) = p(k) + \Theta^*, \quad (27)$$

where

$$\Theta^* = 8\gamma_{d(k)} \times \frac{f^T(x(k)) P' g(x(k)) R(x(k))^2 g^T(x(k)) f(x(k))}{(2R(x(k)) + p(k) g^T(x(k)) P' g(x(k)))^3} \quad (28)$$

which is positive for all time k if $p(0) > 0$. Therefore positiveness for $p(k)$ is ensured and requirement $P(k) = P^T(k) > 0$ para $V(x(k)) = (1/2)x^T(k)P(k)x(k)$ with $P(k) = P^T(k) > 0$ is guaranteed. When SG control goal (21) is achieved, then $p(k) = \bar{p}$ for $k \geq k^*$. Thus, matrix $P(k)$ in (18) is considered constant and $P(k) = P$ where P is computed as $P = \bar{p}P'$, with P' a design positive definite matrix. Under these constraints, we obtain

$$\alpha(x(k)) := u(k) = -\frac{1}{2}(R(x(k)) + P_2(x(k)))^{-1}P_1(x(k)), \quad (29)$$

where $P_1(x(k)) = g^T(x(k))Pf(x(k))$ and $P_2(x(k)) = (1/2)g^T(x(k))Pg(x(k))$.

3.3. Tracking Reference. In the case of tracking reference, the control law is defined as follows [23]:

$$u(k) = -\frac{1}{2}(R(x(k)) + P_2(x(k)))^{-1}P_1(x(k)), \quad (30)$$

where $P_1(x(k)) = g^T(x(k))Pf(x(k) - x_{\text{ref}}(k+1))$ and $P_2(x(k)) = (1/2)g^T(x(k))Pg(x(k))$.

4. Multimachine Power System Control

4.1. Multimachine Power System Complete Model. In this work, the proposed decentralized identification and control scheme is tested with the Western System Coordinating Council (WSCC) 3-machine, 9-bus system [6, 26]. The differential and algebraic equations which represent the i th generator dynamics and power flow constraints respectively [1, 6] are given by

$$\begin{aligned} \dot{x}_{1i} &= x_{2i} - \omega_s \\ \dot{x}_{2i} &= \left(\frac{\omega_s}{2H_i} \right) (T_{mi} - (\psi_{di}I_{qi} - \psi_{qi}I_{di})) \\ \dot{x}_{3i} &= \left(\frac{1}{T'_{d0i}} \right) (-x_{3i} - X_{dd}) \\ &\quad \times [I_{di} - X_{di}^* (x_{5i} + X_{dls})] + E_{fdi} \end{aligned}$$

$$\begin{aligned} \dot{x}_{4i} &= \left(\frac{1}{T'_{q0i}} \right) (-x_{4i} - X_{qq}) \\ &\quad \times [I_{qi} - X_{qi}^* (x_{6i} + X_{qls})] + E'_{di} \\ \dot{x}_{5i} &= \left(\frac{1}{T''_{d0i}} \right) (-x_{5i} + x_{3i} - (X'_{di} - X_{lsi}) I_{di}) \\ \dot{x}_{6i} &= \left(\frac{1}{T''_{q0i}} \right) (-x_{6i} - x_{4i} - (X'_{qi} - X_{lsi}) I_{qi}), \end{aligned} \quad (31)$$

where x_1 is the power angle of the i th generator in rad, x_2 is the rotating speed of the i th generator in rad/s, x_3 is the q -axis internal voltage of the i th generator in p.u., x_4 is the d -axis internal voltage of the i th generator in p.u., x_5 is the $1d$ -axis flux linkage of the i th generator in p.u., x_6 is the $2q$ -axis flux linkage of the i th generator in p.u., E_{fdi} is the excitation control input, and ψ_{di} and ψ_{qi} are the d -axis flux linkage and q -axis flux linkage of the i th generator in p.u., respectively; ω_s is the synchronous rotor speed in rad/s, I_{di} and I_{qi} are the d -axis and q -axis currents of the i th generator in p.u., and E'_{di} is the transient voltage in d -axis of the i th generator. Besides, (4.1) is complemented with

$$\begin{aligned} X_{di}^* &= \frac{X'_{di} - X''_{di}}{(X'_{di} - X_{lsi})^2}, & X_{qi}^* &= \frac{X'_{qi} - X''_{qi}}{(X'_{qi} - X_{lsi})^2} \\ X_{dd} &= X_{di} - X'_{di}, & X_{qq} &= X_{qi} - X'_{qi} \\ X_{dls} &= (X'_{di} - X_{lsi}) I_{di}, & X_{qls} &= (X'_{qi} - X_{lsi}) I_{qi} \end{aligned} \quad (32)$$

being parameters for each synchronous generator. It is important to consider that each machine model considered is a flux decay model (one axis model) given in [1, 6]; excitors and governors are not included in this model [1, 8].

4.2. Reduced Neural Model of Multimachine Power System. The model mentioned above [1] is in continuous time and due to this fact, we proceed to discretize the states using Euler methodology; with the state variables discretized, the reduced neural model is proposed [7] as follows:

$$\begin{aligned} \hat{x}_1(k+1) &= f_1(k) \\ \hat{x}_2(k+1) &= f_2(k) \\ \hat{x}_3(k+1) &= f_3(k) + w_{34}u(k) \\ f_1(k) &= w_{11}(k)S(\hat{x}_1(k)) + w_{12}(k)S(\hat{x}_2(k)) \\ f_2(k) &= w_{21}(k)S(\hat{x}_1(k))^6 + w_{22}(k)S(\hat{x}_2(k)) \\ &\quad + w_{23}(k)S(\hat{x}_3(k)) \\ f_3(k) &= w_{31}(k)S(\hat{x}_1(k)) + w_{32}(k)S(\hat{x}_2(k)) \\ &\quad + w_{33}(k)S(\hat{x}_3(k)), \end{aligned} \quad (33)$$

$$\begin{aligned} f_1(k) &= w_{11}(k)S(\hat{x}_1(k)) + w_{12}(k)S(\hat{x}_2(k)) \\ f_2(k) &= w_{21}(k)S(\hat{x}_1(k))^6 + w_{22}(k)S(\hat{x}_2(k)) \\ &\quad + w_{23}(k)S(\hat{x}_3(k)) \\ f_3(k) &= w_{31}(k)S(\hat{x}_1(k)) + w_{32}(k)S(\hat{x}_2(k)) \\ &\quad + w_{33}(k)S(\hat{x}_3(k)), \end{aligned} \quad (34)$$

where \hat{x}_i estimates x_i ($i = 1, 2, 3$). Given the neural reduced model, the inverse optimal SG control law is applied to the reduced neural model to each synchronous generator, that is, in a decentralized way. Thus, the control law is established from (30) where the matrix P is given for different values for each fault as follows: in the case of the fault at the end of bus 7 ($100 \times I$, $5 \times I$, $20 \times I$), in the case of the fault at the end of bus 8 ($80 \times I$, $5 \times I$, $700 \times I$), and in the case of the fault at the end of bus 9 ($100 \times I$, $5 \times I$, $10 \times I$) for generators 1, 2, and 3 respectively, where I is an identity matrix of 3×3 .

From (33) ($x(k)$), $f(x(k))$, the control law for the neural network is defined as

$$\begin{aligned} g(x(k)) &= \begin{bmatrix} 0 \\ 0 \\ \omega_{34} \end{bmatrix} \\ f(x(k)) &= \begin{bmatrix} f_1(k) \\ f_2(k) \\ f_3(k) \end{bmatrix}. \end{aligned} \quad (35)$$

It is important to note, that [5] proves that low-order models are well-suited for stability analysis and feedback control design for industrial power generators. Moreover, the use of neural networks allows modelling system interconnections using only local information, as well as not modeled dynamics for the reduced model [1].

5. Preliminary Calculations for Faults

For the design of the fault, a system data preparation is required and the following preliminary calculations are taken from [6], considering the parameters of the generators given in Tables 7 and 8.

- (1) All system data are converted to a common base; a system base of 100 MVA is frequently used.
- (2) The loads are converted to equivalent impedances or admittances. The needed data for this step are obtained from the load-flow study. Thus if a certain load bus has a voltage \bar{V}_L , power P_L , reactive power Q_L , and current flowing into a load admittance $\bar{Y}_L = G_L + jB_L$, then

$$P_L + jQ_L = \bar{V}_L \bar{I}_L^* = \bar{V}_L [\bar{V}_L^* (G_L - jB_L)] = \bar{V}_L^2 (G_L - jB_L). \quad (36)$$

The equivalent shunt admittance at that bus is given by

$$\bar{Y}_L = \frac{P_L}{\bar{V}_L^2} - j \left(\frac{Q_L}{\bar{V}_L^2} \right). \quad (37)$$

- (3) The internal voltages of the generators $E_i \angle \delta_{i0}$ are calculated from the load-flow data. These internal angles may be computed from the pretransient terminal voltages $V \angle \alpha$ as follows. Let the terminal voltage be used temporarily as a reference, as shown in Figure 1. If $\bar{I} = I_1 + jI_2$, then, from the relation $P + jQ = \bar{V} \bar{I}^*$

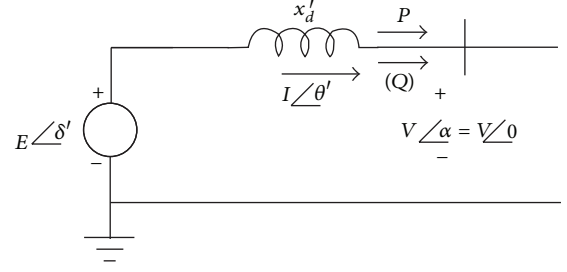


FIGURE 1: Generator representation for computing δ_0 .

it is possible to obtain $I_1 + jI_2 = (P - jQ)/V$. Since $E \angle \delta' = \bar{V} + jx_d' \bar{I}$, then

$$E \angle \delta' = \left(V + \frac{Qx_d'}{V} \right) + j \left(\frac{Px_d'}{V} \right). \quad (38)$$

The initial generator angle δ_0 is then obtained by adding the pretransient voltage angle α to δ' , or

$$\delta_0 = \delta' + \alpha. \quad (39)$$

- (4) The \bar{Y} matrix for each network condition is calculated. The following steps are usually needed.
 - (a) The equivalent load impedances (or admittances) are connected between the load buses and the reference node; additional nodes are provided for the internal generator voltages (nodes 1, 2, ..., n in Figure 2) and the appropriate values of x_d' are connected between these nodes and the generator terminal nodes. Also, simulation of the fault impedance is added as required, and the admittance matrix is determined for each switching condition.
 - (b) All impedance elements are converted to admittances.
 - (c) Elements of the \bar{Y} matrix are identified as follows: \bar{Y}_{ij} is the sum of all the admittances connected to node i , and \bar{Y}_{ij} is the negative of the admittance between node i and node j .
- (5) Finally, all the nodes except for the internal generator nodes are eliminated and obtain the \bar{Y} matrix for the reduced network. The reduction can be achieved by matrix operation recalling all the nodes that have zero injection currents except for the internal generator nodes. This property is used to obtain the network reduction as shown below. Let

$$I = YV, \quad (40)$$

where

$$I = \begin{bmatrix} I_n \\ 0 \end{bmatrix}. \quad (41)$$

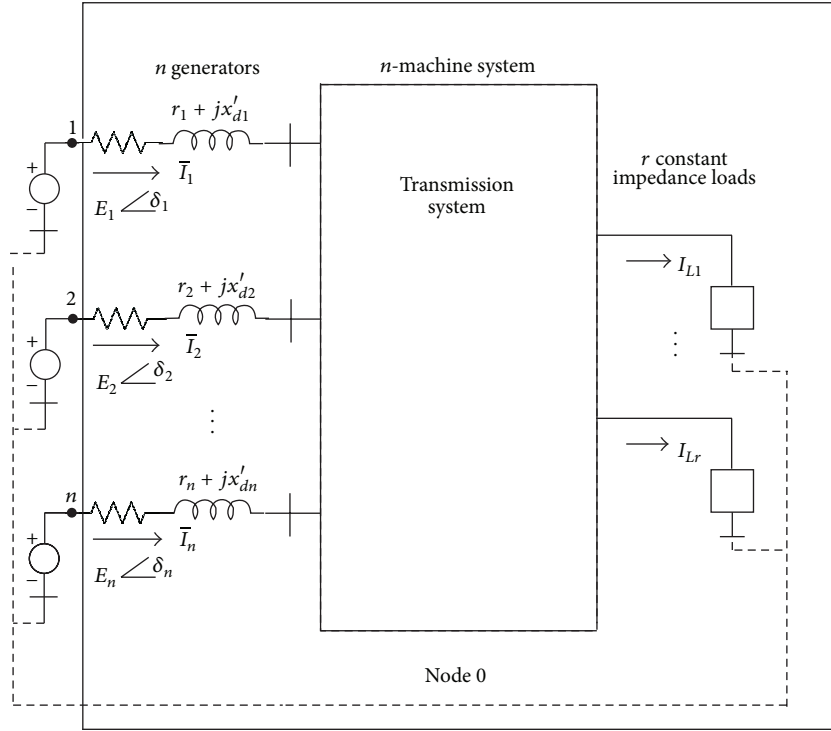


FIGURE 2: Representation of a multimachine system (classical model).

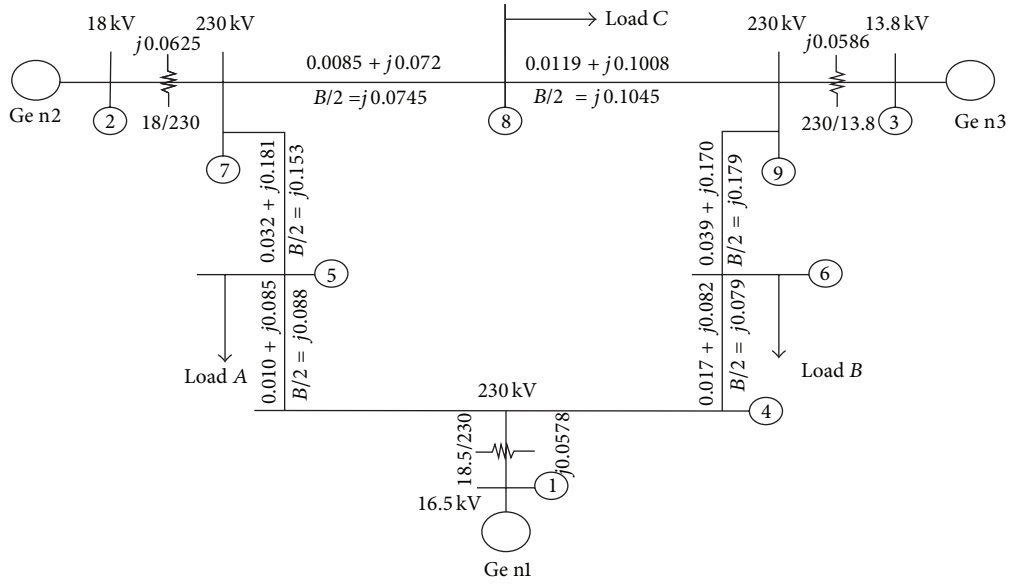


FIGURE 3: Nine bus system.

Now the matrices Y and V are partitioned accordingly to get

$$\begin{bmatrix} I_n \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{nn} & Y_{nr} \\ Y_{rn} & Y_{rr} \end{bmatrix} \begin{bmatrix} V_n \\ V_r \end{bmatrix}, \quad (42)$$

where the subscript n is used to denote generator nodes and the subscript r is used for the remaining nodes. Thus for the

network in Figure 2, $V_n \in \mathfrak{R}^{n \times 1}$ and $V \in \mathfrak{R}^{r \times 1}$. Expanding (42),

$$I_n = Y_{nn} V_n + Y_{nr} V_r, \quad 0 = Y_{rn} V_n + Y_{rr} V_r \quad (43)$$

from which we eliminate V_r to find

$$I_n = (Y_{nn} V_n - Y_{nr} Y_{rr}^{-1} Y_{rn}) V_n. \quad (44)$$

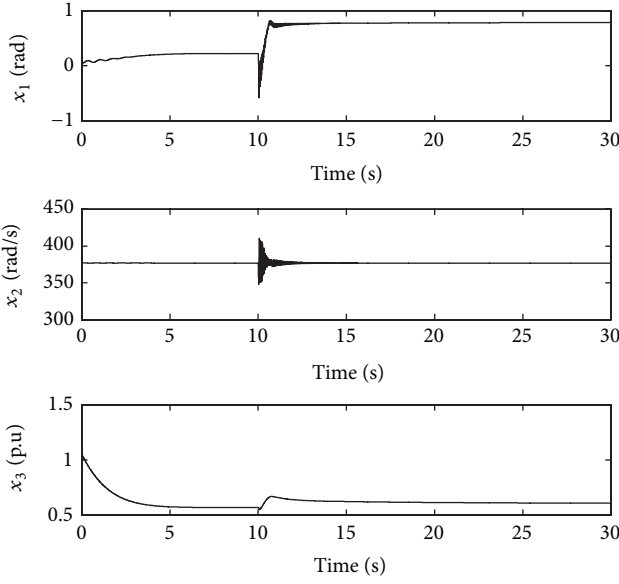


FIGURE 4: Generator 1 response with a fault at bus 7.

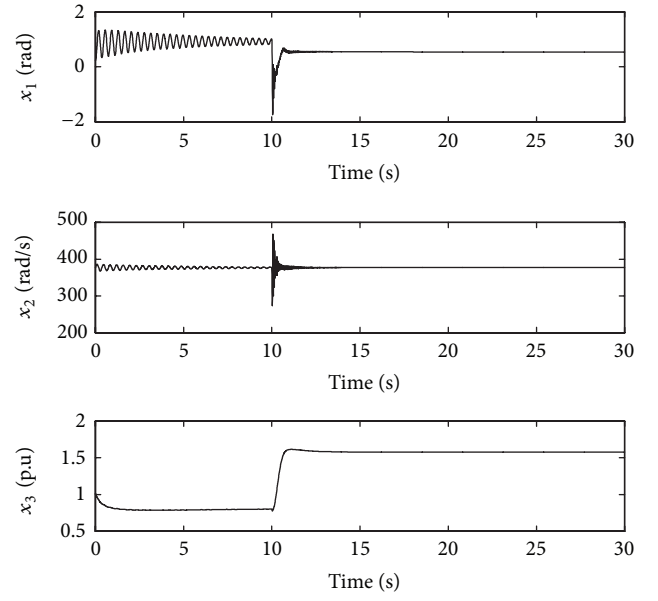


FIGURE 6: Generator 3 response with a fault at bus 7.

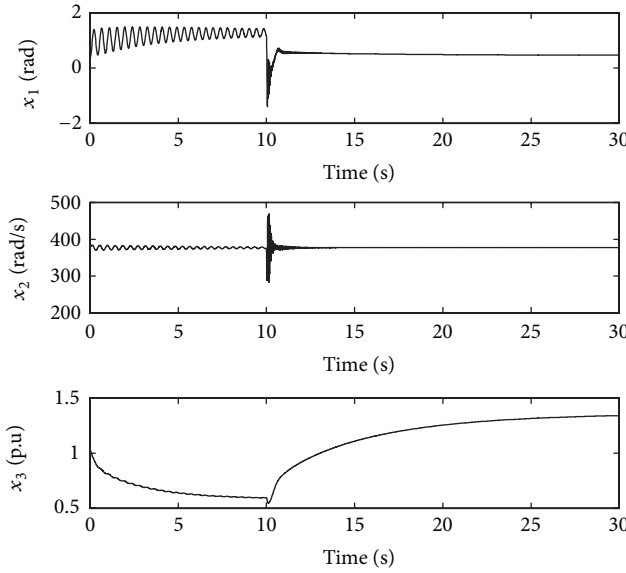


FIGURE 5: Generator 2 response with a fault at bus 7.

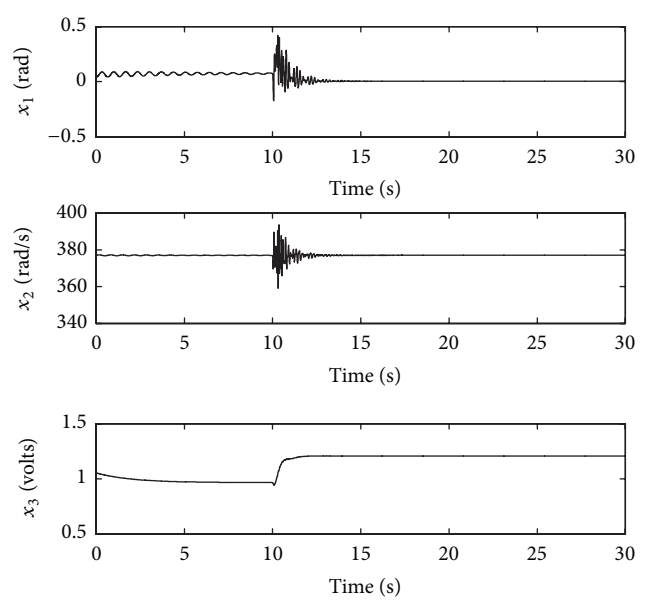


FIGURE 7: Generator 1 response with a fault at bus 8.

The matrix $(Y_{mn}V_n - Y_{nr}Y_{rr}^{-1}Y_{rn})$ is the desired reduced matrix $Y \in \mathbb{R}^{n \times n}$, where n is the number of the generators. The network reduction illustrated by (43)-(44) is a convenient analytical technique that can be used only when the loads are treated as constant impedances. If the loads are not considered to be constant impedances, the identity of the load buses must be retained. Network reduction can be applied only to those nodes that have zero injection current.

Once the preliminaries calculations are made to obtain the Y matrix for each fault in the correspondent bus, the network reduction for each fault is applied. For the first case of the analysis, the fault occurs at bus 7 and then the correspondent Y matrix is obtained as shown in Tables 9, 10,

and 11 included at the Appendix. Then the network reduction of Y matrix is applied and is defined as in Table 1.

For the second case of the analysis, the fault occurs at bus 8 and then the correspondent Y matrix is obtained as shown in Tables 12, 13, and 14 included at the Appendix after the network reduction of Y matrix is realized to obtain the reduced networks defined as in Table 2.

For the third case of the analysis, the fault occurs at bus 9 and then the correspondent Y matrix is obtained as shown in Tables 15, 16 and 17 included at the Appendix after the network reduction of Y matrix is realized to obtain the reduced networks defined as in Table 3.

TABLE 1: Reduced Y Matrices at bus 7.

Type of network	Node	1	2	3
Pre-fault	1	$0.846 - j2.988$	$0.287 + j1.513$	$0.210 + j1.226$
	2	$0.287 + j1.513$	$0.420 - j2.724$	$0.213 + j1.088$
	3	$0.210 + j1.226$	$0.213 + j1.088$	$0.277 - j2.368$
Faulted	1	$0.657 - j3.816$	$0.000 + j0.000$	$0.070 + j0.631$
	2	$0.000 + j0.000$	$0.000 - j5.486$	$0.000 + j0.000$
	3	$0.070 + j0.631$	$0.000 - j0.000$	$0.174 - j2.796$
Fault cleared	1	$1.181 - j2.229$	$0.138 + j0.726$	$0.191 + j1.079$
	2	$0.138 + j0.726$	$0.389 - j1.953$	$0.199 + j1.229$
	3	$0.191 + j1.079$	$0.199 + j1.229$	$0.273 - j2.342$

TABLE 2: Reduced Y Matrices at bus 8.

Type of network	Node	1	2	3
Pre-fault	1	$0.938 - j2.798$	$0.325 + j1.588$	$0.251 + j1.315$
	2	$0.325 + j1.588$	$0.436 - j2.694$	$0.230 + j1.123$
	3	$0.251 + j1.315$	$0.230 + j1.123$	$0.296 - j2.325$
Faulted	1	$0.736 - j3.569$	$0.082 + j0.535$	$0.063 + j0.534$
	2	$0.082 + j0.535$	$0.146 - j4.128$	$0.006 + j0.058$
	3	$0.063 + j0.534$	$0.006 + j0.058$	$0.122 - j3.115$
Fault cleared	1	$0.850 - j3.252$	$0.334 + j1.346$	$0.075 + j0.569$
	2	$0.334 + j1.346$	$0.687 - j2.061$	$0.032 + j0.148$
	3	$0.075 + j0.569$	$0.032 + j0.148$	$0.124 - j3.111$

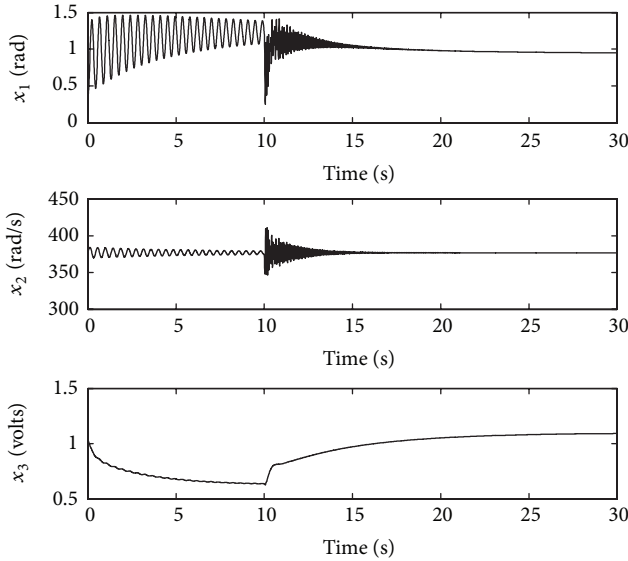


FIGURE 8: Generator 2 response with a fault at bus 8.

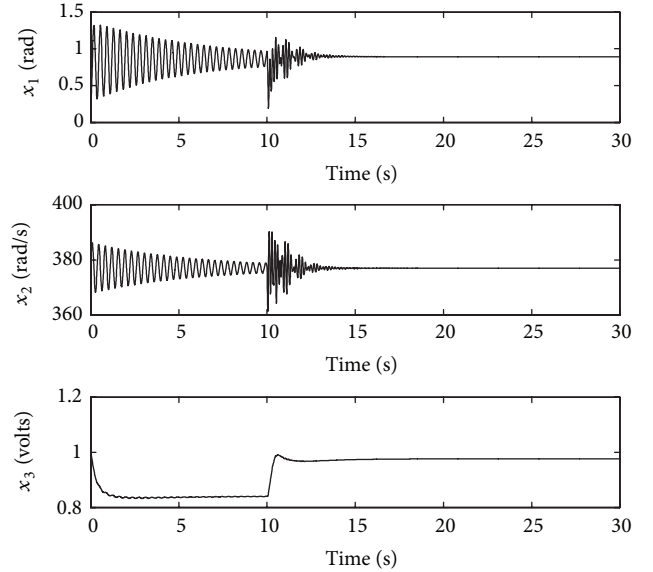


FIGURE 9: Generator 3 response with a fault at bus 8.

6. Fault Simulation

The power electric system used in this paper is presented in Figure 3. It corresponds to the nine bus system. Figure 3 also includes the bus interconnection and the related parameters in the transmission lines. Data for simulation is given in Tables 7 and 8, respectively [6], where the modeling of the

system is explained and the related parameters for each synchronous generator are described.

In this paper, the 18 state variables related to 3 synchronous generators are stabilized, using the neural reduced model [7], reaching stabilization for the system with the fault in three different lines of transmission, for simulation the sample time is fitted to 0.005 ms.

TABLE 3: Reduced Y Matrices at bus 9.

Type of network	Node	1	2	3
Pre-fault	1	$0.938 - j2.798$	$0.325 + j1.588$	$0.251 + j1.315$
	2	$0.325 + j1.588$	$0.436 - j2.694$	$0.230 + j1.123$
	3	$0.251 + j1.315$	$0.230 + j1.123$	$0.296 - j2.325$
Faulted	1	$0.727 - j3.735$	$0.135 + j0.787$	$-0.004 + j0.001$
	2	$0.135 + j0.787$	$0.263 - j3.377$	$-0.002 + j0.000$
	3	$-0.004 + j0.001$	$-0.002 + j0.000$	$-0.010 - j4.168$
Fault cleared	1	$1.271 - j1.980$	$0.290 + j1.247$	$0.102 + j0.344$
	2	$0.290 + j1.247$	$0.380 - j2.957$	$0.149 + j0.702$
	3	$0.102 + j0.344$	$0.149 + j0.702$	$0.209 - j2.853$

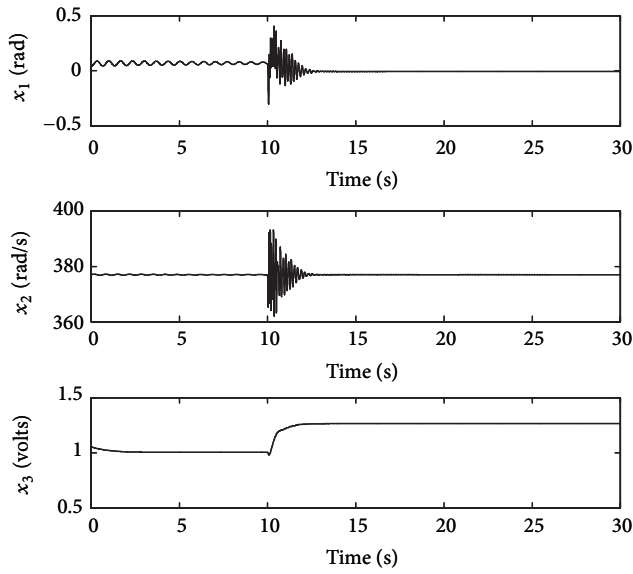


FIGURE 10: Generator 1 response with a fault at bus 9.

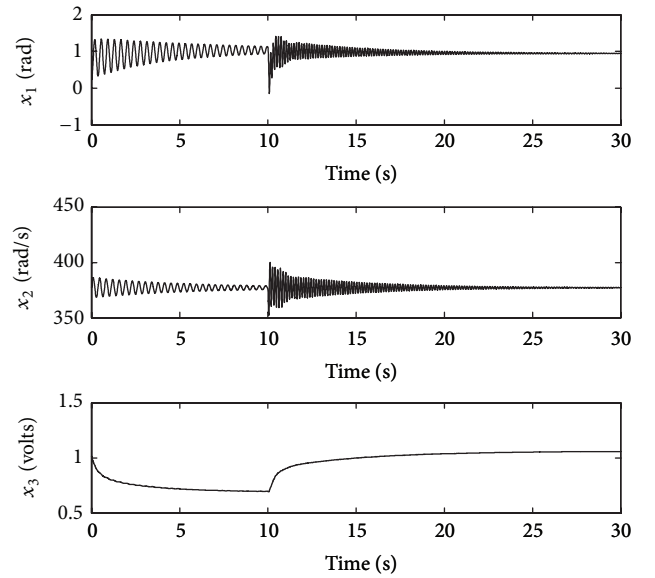


FIGURE 12: Generator 3 response with a fault at bus 9.

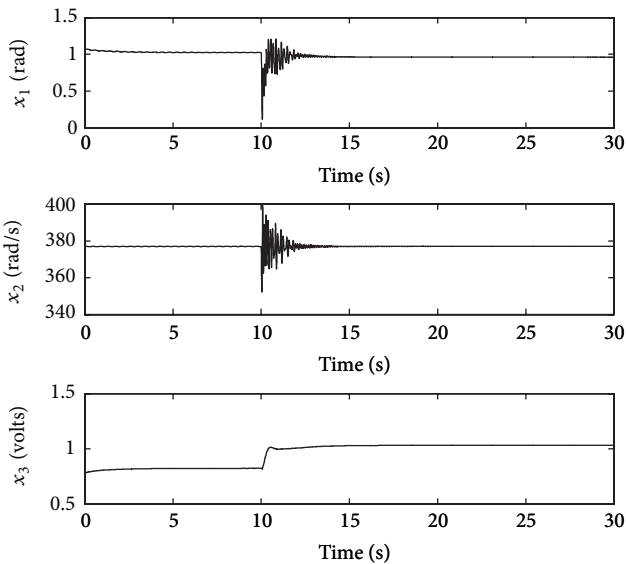


FIGURE 11: Generator 2 response with a fault at bus 9.

TABLE 4: Admittance loads.

Load	Admittance
A	$\bar{y}_{L5} = 1.2610 - j0.5044$
B	$\bar{y}_{L6} = 0.8777 - j0.2926$
C	$\bar{y}_{L8} = 0.9690 - j0.3391$

TABLE 5: Initial conditions of the generators.

Initial conditions	Generator 1	Generator 2	Generator 3
x_{01}	0.0396	0.3444	0.23
x_{02}	377	377	377
x_{03}	1.056	1.0502	1.0170
x_{04}	0	0.622	0.624
x_{05}	1.0478	0.7007	0.7078
x_{06}	-0.0425	-0.7568	-0.7328

There are three cases contemplated in the system simulation.

- (1) The fault occurs near bus 7 at the end of the lines 5–7. Results are depicted in Figure 4 for generator 1, Figure 5 for generator 2, and Figure 6 for generator 3.

TABLE 6: References for the system.

References	Generator 1	Generator 2	Generator 3
$x_{1\text{ref}}$	0.0396	0.3444	0.23
$x_{2\text{ref}}$	377	377	377
$x_{3\text{ref}}$	0.5	1.0502	1.0170

(2) The fault occurs near bus 8 at the end of the lines 8-9. Results are depicted in Figure 7 for generator 1, Figure 8 for generator 2, and Figure 9 for generator 3.

(3) The fault occurs near bus 9 at the end of the lines 6-9. Results are depicted in Figure 10 for generator 1, Figure 11 for generator 2, and Figure 12 for generator 3.

For the cases above mentioned, the fault is incepted at 10 seconds of simulation and then it is possible to see that the system has a prefault state (before 10 seconds), a fault state (at 10 seconds), and a postfault state (after 10 seconds). The admittances for the loads are given in p.u. in Table 4.

The initial conditions for the system are given in Table 5.

It is important to note that initial conditions of the generators are defined by their respective parameters [1]; however, in order to test the NN approximation capabilities, it is common to use signals that can represent a wide range of frequencies; then, it is possible that plant signals can exhibit a high frequency behavior [10].

The control goal is to stabilize the power electric system and this is why the references given for each state variable of the neural reduced model for the multimachine system are proposed as in Table 6.

7. Conclusions

In this paper a SG discrete-time inverse optimal controller is synthesized for a reduced order neural model to stabilize a multimachine power electric system in the presence of a fault at line 7, at line 8, and at line 9; from simulation results, it can be seen that the proposed controller allows stabilizing the state in an efficient way in the three different cases, allowing the system stabilization after the fault occurs. As future work authors are considering the stability analysis including the neural decentralized controller, besides the analysis of control delay for closed loop system.

Appendix

In this appendix, parameters used for simulations are presented. Tables 7 and 8 show the parameters for generators and transmission lines, respectively. Tables 9, 10 and 11 display the Y matrix of network with fault near to bus 7 for prefault, fault, and fault cleared conditions. Tables 12, 13 and 14 show the Y matrix of network with fault near to bus 8 for prefault, fault, and fault cleared conditions. Tables 15, 16 and 17 present the Y

TABLE 7: Parameters of the generators.

Parameter	Generator 1	Generator 2	Generator 3
H (sec)	23.6400	6.4000	3.0100
T_m (pu)	0.7160	1.6300	0.8500
T'_{d0} (sec)	8.9600	6.0000	5.8900
T''_{d0} (sec)	0.2000	0.3000	0.4000
T'_{q0} (sec)	0.3100	0.5350	0.6000
T''_{q0} (sec)	0.2000	0.3000	0.4000
X_d (pu)	0.1460	0.8958	1.3125
X'_d (pu)	0.0608	0.1198	0.1813
X''_d (pu)	0.0200	0.0500	0.0800
X_q (pu)	0.0969	0.8645	1.2578
X'_q (pu)	0.0969	0.1969	0.2500
X''_q (pu)	0.0200	0.500	0.0800
X_{ls} (pu)	0.0336	0.0521	0.0742

TABLE 8: Parameters of the transmission lines.

Bus i	Bus j	R_{ij}	X_{ij}	G_{ij}	B_{ij}
1	4	0.000	0.1184	0.000	-8.4459
2	7	0.000	0.1823	0.000	-5.4855
3	9	0.000	0.2399	0.000	-4.1684
4	5	0.0100	0.0850	1.3652	-11.6041
4	6	0.0170	0.0920	1.9522	-10.5107
5	7	0.0320	0.1610	1.1876	-5.9751
6	9	0.0390	0.1700	1.2820	-5.5882
7	8	0.0085	0.0720	1.6171	-9.7843
8	9	0.0119	0.1008	1.1551	-9.7843
5	0	0.000	0.000	1.2610	-0.2634
6	0	0.000	0.000	0.8777	-0.0346
8	0	0.000	0.000	0.9690	-1.1601
4	0	0.000	0.000	0.000	0.1670
7	0	0.000	0.000	0.000	0.2275
9	0	0.000	0.000	0.000	0.2835

matrix of network with fault near to bus 9 for prefault, fault, and fault cleared conditions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the support of CONACYT Mexico, through Projects 103191Y, 106838Y, and 156567Y. They also thank the very useful comments of the anonymous reviewers, which help to improve the paper. First author also thanks the scholarship LOreal-AMC (Mexican Academy of Sciences) for woman in science.

TABLE 9: Y matrix of pre-faulted Network near to bus 7.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - 30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	$-1.1876 + j5.9751$	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	$j5.4855$	0	0	$-1.1876 + j5.9751$	0	$2.8047 - j24.9311$	$-1.6171 + j13.6980$	0
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$3.7412 - j23.6423$	$-1.1551 + j9.7843$
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	$-1.1551 + j9.7843$	$2.4371 - j19.2574$

TABLE 10: Y matrix of faulted Network near to bus 7.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	0.0100	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	0
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	0.0100	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	0.0100	0	0	0.0100	0	0.0100	0.0100	0
8	0	0	0	0	0	0	0.0100	$3.7412 - j23.6423$	$-1.1551 + j9.7843$
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	$-1.1551 + j9.7843$	$2.4371 - j19.2574$

TABLE II: Y matrix of fault cleared Network near to bus 7.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$2.6262 - j11.8675$	0	0	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	$j5.4855$	0	0	0	0	$1.6171 - j18.9559$	$-1.6171 + j13.6980$	0
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$3.7412 - j23.6423$	$-1.1551 + j9.7843$
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	$-1.1551 + j9.7843$	$2.4371 - j19.2574$

TABLE 12: Y matrix of pre-faulted Network near to bus 8.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	$-1.1876 + j5.9751$	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	$j5.4855$	0	0	$-1.1876 + j5.9751$	0	$2.8047 - j24.9311$	$-1.6171 + j13.6980$	0
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$3.7412 - j23.6423$	$-1.1551 + j9.7843$
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	$-1.1551 + j9.7843$	$2.4371 - j19.2574$

TABLE 13: Y matrix of faulted Network near to bus 8.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	$-1.1876 + j5.9751$	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	$j5.4855$	0	0	$-1.1876 + j5.9751$	0	$2.8047 - j24.931$	0.0010	0
8	0	0	0	0	0	0	0.0100	0.0010	0.0010
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	0.0010	$2.4371 - j19.2574$

TABLE 14: Y matrix of fault cleared Network near to bus 8.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	$-1.1876 + j5.9751$	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	$j5.4855$	0	0	$-1.1876 + j5.9751$	0	$2.8047 - j24.9311$	$-1.6171 + j13.6980$	0
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$2.5861 - j13.8580$	0
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	0	$2.4371 - j19.2574$

TABLE 15: Y matrix of pre-faulted Network near to bus 9.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	$-1.1876 + j5.9751$	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	$-1.2820 + j5.5882$
7	0	$j5.4855$	0	0	$-1.1876 + j5.9751$	0	$2.8047 - j24.9311$	$-1.6171 + j13.6980$	0
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$3.7412 - j23.6423$	$-1.1551 + j9.7843$
9	0	0	$j4.1684$	0	0	$-1.2820 + j5.5882$	0	$-1.1551 + j9.7843$	$2.4371 - j19.2574$

TABLE 16: Y matrix of faulted Network near to bus 9.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	0.0100
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0.0100
5	0	0	0	$-1.3652 + j11.6041$	$3.8138 - j17.8426$	0	$-1.1876 + j5.9751$	0	0.0100
6	0	0	0	$-1.9422 + j10.5107$	0	$4.1019 - j15.4225$	0	0	0.0100
7	0	$j5.4855$	0	0	$-1.1876 + j5.9751$	0	$2.8047 - j24.9311$	$-1.6171 + j13.6980$	0.0100
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$3.7412 - j23.6423$	0.0100
9	0	0	0.0100	0	0	0.0100	0	0.0100	0.0100

TABLE 17: Y matrix of fault cleared Network near to bus 9.

Node	1	2	3	4	5	6	7	8	9
1	$-j8.4459$	0	0	$j8.4459$	0	0	0	0	0
2	0	$-j5.4855$	0	0	0	0	$j5.4855$	0	0
3	0	$-j4.1684$	0	0	0	0	0	0	$j4.1684$
4	$j8.4459$	0	0	$3.3074 - j30.3937$	$-1.3652 + j11.6041$	$-1.9422 + j10.5107$	0	0	0
5	0	0	0	$-1.3652 + j11.6041$	$2.6262 - j11.8675$	0	$-1.1876 + j5.9751$	0	0
6	0	0	0	$-1.9422 + j10.5107$	0	$2.8199 - j9.8343$	0	0	0
7	0	$j5.4855$	0	0	0.0100	0	$1.6171 - j18.9559$	$-1.6171 + j13.6980$	0
8	0	0	0	0	0	0	$-1.6171 + j13.6980$	$3.7412 - j23.6423$	$-1.1551 + j9.7843$
9	0	0	$j4.1684$	0	0	0	0	$-1.1551 + j9.7843$	$2.4371 - j19.2574$

References

- [1] V. H. B. Benitez, *Decentralized continuous time neural control [Ph.D. thesis]*, Unidad Guadalajara, Cinvestav, Mexico, 2010.
- [2] H. Li, X. Jing, and H. R. Karimi, "Output-feedback-based H_∞ control for vehicle suspension systems with control delay," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 436–446, 2014.
- [3] X.-L. Zhu, B. Chen, D. Yue, and Y. Wang, "An improved input delay approach to stabilization of fuzzy systems under variable sampling," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 330–341, 2012.
- [4] S. Huang, K. K. Tan, and T. H. Lee, "Decentralized control design for large-scale systems with strong interconnections using neural networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 805–810, 2003.
- [5] M. A. Arjona, R. Escarela-Perez, G. Espinosa-Perez, and J. Alvarez-Ramirez, "Validity testing of third-order nonlinear models for synchronous generators," *Electric Power Systems Research*, vol. 79, no. 6, pp. 953–958, 2009.
- [6] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, IEEE Press, New York, NY, USA, 1994.
- [7] A. Y. Alanis, E. N. Sanchez, and F. O. Tellez, "Discrete-time inverse optimal neural control for synchronous generators," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 697–705, 2013.
- [8] T. Weckesser, H. Johansson, and J. Østergaard, "Impact of model detail of synchronous machines on real-time transient stability assessment," in *Proceedings of the IREP Symposium-Bulk Power System Dynamics and Control*, 2013.
- [9] A. Y. Alanis, N. Arana-Daniel, and C. Lopez-Franco, "Neural-PSO second order sliding mode controller for unknown discrete-time nonlinear systems," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 3065–3070, 2013.
- [10] S. Haykin, *Kalman Filtering and Neural Networks*, John Wiley & Sons, New York, NY, USA, 2001.
- [11] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270–280, 1989.
- [12] C.-S. Leung and L.-W. Chan, "Dual extended Kalman filtering in recurrent neural networks," *Neural Networks*, vol. 16, no. 2, pp. 223–239, 2003.
- [13] A. Y. Alanis, M. Lopez-Franco, N. Arana-Daniel, and C. Lopez-Franco, "Discrete-time neural control for electrically driven nonholonomic mobile robots," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 7, pp. 630–644, 2012.
- [14] L. A. Feldkamp, T. M. Feldkamp, and D. V. Prokhorov, "Neural network training with the nprKF," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '01)*, pp. 109–114, USA, July 2001.
- [15] R. Grover and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1992.
- [16] Y. Song and J. W. Grizzle, "The extended Kalman filter as a local asymptotic observer for discrete-time nonlinear systems," *Journal of Mathematical Systems, Estimation, and Control*, vol. 5, no. 1, pp. 59–78, 1995.
- [17] A. S. Poznyak, E. N. Sanchez, and W. Yu, *Differential Neural Networks for Robust Nonlinear Control*, World Scientific, Singapore, 2001.
- [18] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko, *The Mathematical Theory of Optimal Processes*, Interscience, New York, NY, USA, 1962.
- [19] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.
- [20] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1962.
- [21] M. Krstić and H. Deng, *Stabilization of Nonlinear Uncertain Systems*, Springer, Berlin, Germany, 1998.
- [22] R. Sepulchre, M. Janković, and P. V. Kokotović, *Constructive Nonlinear Control*, Springer, Berlin, Germany, 1997.
- [23] F. O. Ornelas-Tellez, *Inverse optimal control for discrete-time nonlinear systems [Ph.D. thesis]*, Unidad Guadalajara, Cinvestav, Mexico, 2011.
- [24] R. A. Freeman and P. V. Kokotović, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*, Birkhäuser Boston, Cambridge, Mass, USA, 1996.
- [25] A. L. Fradkov and A. Yu. Pogromsky, *Introduction to Control of Oscillations and Chaos*, vol. 35, World Scientific, Singapore, 1998.
- [26] Power System Dynamic Analysis-Phase I, "Electric power research institute," EPRI Report EL-484, 1977.

