01 Jan 2006

Fault-Tolerant Control for SSSC using Neural Networks and PSO

Wei Qiao

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Ronald G. Harley

Recommended Citation

W. Qiao et al., "Fault-Tolerant Control for SSSC using Neural Networks and PSO," *Proceedings of the IEEE PES Power Systems Conference and Exposition, 2006*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/PSCE.2006.296441

Fault-Tolerant Control for SSSC Using Neural Networks and PSO

Wei Qiao, *Student Member, IEEE*, Ronald G. Harley, *Fellow, IEEE*, and Ganesh K. Venayagamoorthy, *Senior Member, IEEE*

Abstract**-- This paper presents a fault-tolerant indirect adaptive neuro-controller (FTNC) for controlling a static synchronous series compensator (SSSC), which is connected to a power network. The FTNC consists of a sensor evaluation and restoration scheme (SERS), a radial basis function neuro-identifier (RBFNI) and a radial basis function neuro-controller (RBFNC). The SERS is designed using the auto-associative neural networks (auto-encoder) and the particle swarm optimizer (PSO). This FTNC is able to provide efficient control to the SSSC when single or multiple crucial sensor measurements are unavailable. The validity of the proposed FTNC model is examined by simulations in PSCAD/EMTDC environment.**

Index Terms**--Fault-tolerant control, neural networks, particle swarm optimization, static synchronous series compensator**

I. Introduction

THE static synchronous series compensator (SSSC), using a voltage source converter to inject a controllable voltage in quadrature with the line current of a power system, belongs to the family of flexible ac transmission system (FACTS) devices. It is able to rapidly provide both capacitive and inductive impedance compensation independent of the line current [1]. By coupling an additional energy storage system to the dc terminal, the SSSC can also provide simultaneous active power compensation, which further enhances its capability in power flow control, power oscillation damping and transient stability [1]-[3].

In terms of the control objectives, various control schemes, based on the conventional linear PI controllers, have been designed for the internal control of the SSSC [3]-[6]. In a previous work [6], the authors proposed a model reference indirect adaptive neuro-controller for the internal control of an SSSC. This neuro-controller was shown improved transient performance over the conventional linear PI controllers (CONVC).

However, control of nonlinear plants in power systems relies on the availability and the quality of sensor measurements. Measurements can be corrupted or interrupted due to sensor failure, broken or bad connections, bad communication, or malfunction of some hardware or software (these are referred as missing sensor measurements in this paper). If some sensors fail to provide the correct information, the controllers cannot guarantee the correct control behavior for the plant based on the faulty input data. Therefore, fault-tolerant measurements are an essential requirement for system control.

For many systems, certain degrees of redundancy are present among the data collected from various sensors. If the degree of redundancy is sufficiently high, the readings from one or more missing sensors may be able to be accurately restored from those remaining healthy sensor readings. Conventional methods in recovering missing sensor data are based on the analysis of the system model, e.g., the state estimation methods. The drawbacks of these methods have been discussed in [7], [8].

In a previous work [9], the authors proposed a fault-tolerant P-Q decoupled control scheme (FTCS) for an SSSC. This FTCS contains a suitably designed sensor evaluation and (missing sensor) restoration scheme (SERS) cascaded with a P-Q decoupled control scheme using conventional linear PI controllers. A brief review on fault-tolerant control was also given in [9].

This paper proposes a fault-tolerant indirect adaptive neuro-controller (FTNC) for the internal control of an SSSC connected a power network. This FTNC contains a SERS cascaded with a radial basis function neuro-identifier (RBFNI) and a radial basis function neuro-controller (RBFNC), as shown in Fig. 1. The RBFNI is trained to provide a dynamic predictive plant model at all times; this plant model is then used for training the RBFNC; the RBFNC in turn generates the control signals to drive the outputs of the actual plant to the desired values [6]. The SERS is used to evaluate the integrity of the crucial sensor measurements that determine the behaviors of the RBFNI and the RBFNC. If one or more sensors are missing, the SERS searches in its input space for the optimal estimates of the missing data. The restored values of the missing data from the SERS, together with the remaining data read directly from the healthy sensors, provide a set of complete inputs to the RBFNI and the RBFNC. This guarantees a fault-tolerant control for the SSSC. Simulation studies are carried out with single and multiple time varying current sensors missing in order to evaluate the performance of the proposed FTNC scheme.

This work was supported in part by the National Science Foundation, Washington, DC, USA, under grant ECS 0400657 and the Duke Power Company, Charlotte, North Carolina, USA.

Wei Qiao and Ronald G. Harley are with the Intelligent Power Infrastructure Consortium (IPIC) in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: {weiqiao, rharley}@ece.gatech.edu). Ronald G. Harley is also a professor emeritus at the University of KwaZulu-Natal, Durban, South Africa.

Ganesh K. Venayagamoorthy is with the Real-Time Power and Intelligent Systems Laboratory in the Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO 65409-0249 USA (e-mail: gkumar@ieee.org).

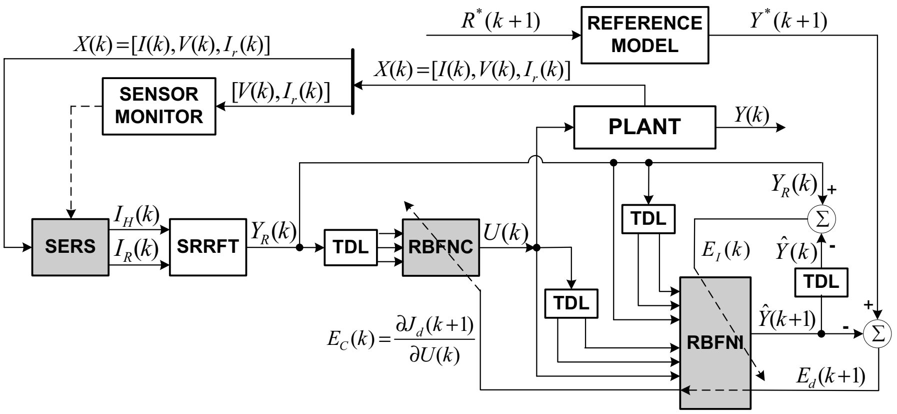


Fig. 1. Schematic diagram of the fault-tolerant indirect adaptive neuro-controller (FTNC) connected to the plant: $R^* = [Q^*, P^*]$, $Y^* = [i_q^*, i_d^*]$, $U = [v_{cq}, v_{cd}]$, $Y = [i_q, i_d]$, $Y_R = [i_{qR}, i_{dR}]$, $\hat{Y} = [\hat{\imath}_q, \hat{\imath}_d]$, $I = [i_a, i_b, i_c]$, $V = [v_{ca}, v_{cb}, v_{cc}]$, and $I_r = [i_{ra}, i_{rb}, i_{rc}]$. SRRFT means synchronously rotating reference frame transformation.

II. SSSC And Power Network Model

Figure 2 illustrates an SSSC with its internal controllers connected to a 160 MVA, 15 kV (L-L) single machine infinite bus (SMIB) power system [6]. The three three-phase transmission lines represent the different loops between the generator bus and the infinite bus. The SSSC is located at the receiving end of line 3. The system is simulated in PSCAD/EMTC environment.

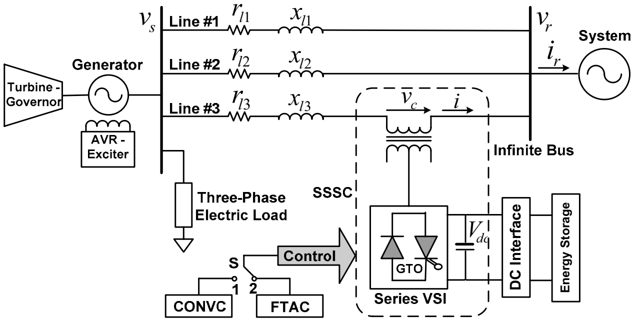


Fig. 2. SSSC in a SMIB power system

A *P-Q* decoupled power flow control scheme for SSSC as described in [6] is shown in Fig. 3. P^* and Q^* are desired reference values of the transmitted real power and reactive power at the receiving end of line 3, which are used to determine the reference values of *d*-axis component i_d^* and *q*-axis component i_q^* of the line current at the SSSC ac terminal. The instantaneous three-phase currents of line 3 are sampled and transformed into *d*-axis and *q*-axis components i_d and i_q by applying the synchronously rotating reference frame transformation (SRRFT). The actual *d-q* current signals are compared with the corresponding reference signals to generate the *d*-axis and *q*-axis current deviations, respectively, which are then passed through two PI controllers ($\mathrm{PI_d}$ and $\mathrm{PI_q}$, called CONVC). The outputs of the PI controllers in turn determine the modulation index and phase shift applied to the PWM module to drive the GTO thyristors of the inverter. The main objective of this SSSC is to control the transmitted real and reactive power at the receiving end of line 3.

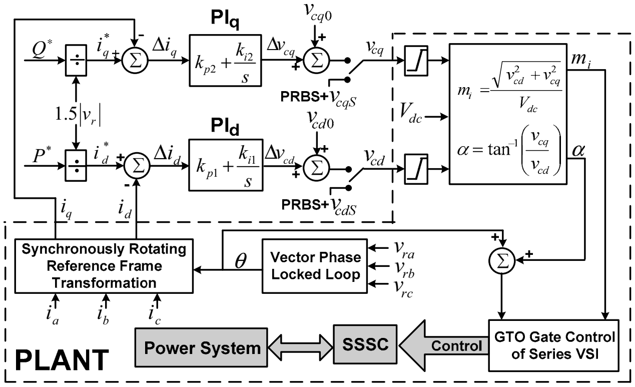


Fig. 3. *P-Q* decoupled control scheme for SSSC

III. Fault-Tolerant Indirect Adaptive Neuro-Controller

A. Overall Structure

A schematic diagram of the proposed FTNC connected to the plant (the dash-line block in Fig. 3) is shown in Fig. 1. It contains a SERS [9], a RBFNI and a RBFNC. The plant inputs and outputs are $U = [v_{cq}, v_{cd}]$ and $Y = [i_q, i_d]$, respectively. In this paper, i_d and i_q are two crucial variables to determine the behavior of the RBFNI and the RBFNC. These two *d*-axis and *q*-axis currents are calculated from the three-phase currents i_a, i_b, and i_c ($I = [i_a, i_b, i_c]$) of line 3 (Fig. 2), which are variables measured by the metering current transformers (called current sensors hereafter). The vector $V = [v_{ca}, v_{cb}, v_{cc}]$, consists of the three-phase ac-side injected voltages of the SSSC, measured by the metering potential transformers (called voltage sensors hereafter). The vector $I_r = [i_{ra}, i_{rb}, i_{rc}]$, measured by other current sensors, consists of the three-phase currents flowing from the infinite bus into the system. These two vectors, V and I_r, are irrelevant to the performances of the RBFNI and RBFNC and used to build the correlations with the variables in the vector I. Therefore, missing any measurement in the vector V or the vector I_r is not taken into account in this paper. The SERS only works under the condition that v_{ca}, v_{cb}, v_{cc} and i_{ra}, i_{rb}, i_{rc} are all available. This condition is determined by a sensor monitor. In practice, the sensor monitor can be designed by using the following relationships. During balanced operation, v_{ca}, v_{cb}, v_{cc}, and i_{ra}, i_{rb}, i_{rc}, should approximately satisfy the following equations.

$$v_{ca} + v_{cb} + v_{cc} = 0 \tag{1}$$

$$i_{ra} + i_{rb} + i_{rc} = 0 \tag{2}$$

If the system is under balanced operating conditions but the above relationships conflict, it indicates that one or more sensors are lost.

The SERS only evaluates the integrity of the crucial vector I. If the SERS identifies that one or more current sensors are missing, it is responsible for restoring all missing sensors. The output vector of the SERS, I_R, contains the restored sensor data; but I_H, contains other healthy sensor readings in the vector I. The variables, $[I_R, I_H]$, are transformed into the d-axis and q-axis current components, $Y_R = [i_{qR}, i_{dR}]$, by applying the SRRFT. In this paper, missing any of the three currents i_a, i_b and i_c results in the loss of both i_d and i_q. Therefore, the calculated currents i_{dR} and i_{qR} from the SRRFT block, by using the restored currents from the SERS, are then used by the RBFNI and the RBFNC as the estimated actual plant outputs for continuous on-line identification and control. If there is no sensor missing, the vector Y_R is exactly same as the actual plant output vector Y.

B. Design of RBFNI and RBFNC

The RBFNI and RBFNC are each a three-layer RBF network with the Gaussian density function as the activation functions in the hidden layer. The overall input-output mapping for the RBF network, $\hat{f} : X \in R^n \to Y \in R^m$ is

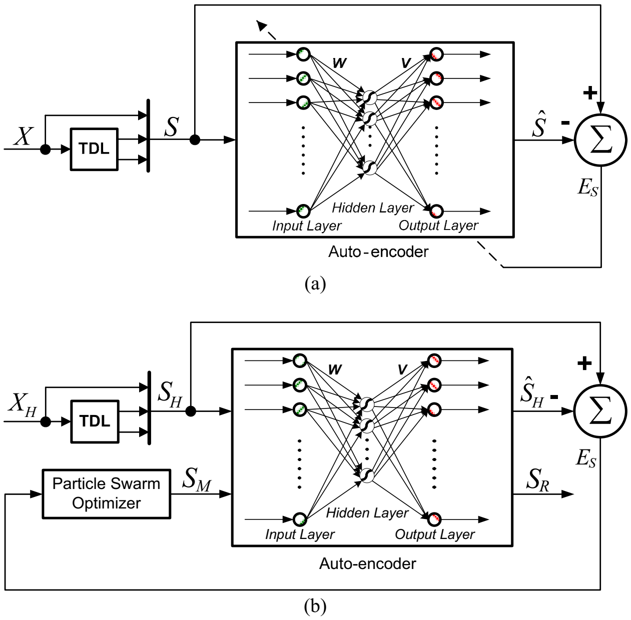
$$\hat{y}_i = b_i + \sum_{j=1}^{h} v_{ji} \exp\left(-\frac{\left\| x - C_j \right\|^2}{\beta_j^2} \right) \tag{3}$$

where x is the input vector, $C_j \in R^n$ is the center of the j^{th} RBF units in the hidden layer, h is the number of RBF units, b_i and v_{ji} are the bias term and the weight between hidden and output layers respectively, and $\hat{y}_i$ is the i^{th} output.

The RBFNI is used to provide a dynamic predictive plant model at all times. This model is then used for training the RBFNC. The plant inputs $U = [v_{cq}, v_{cd}]$ and outputs $Y_R = [i_{qR}, i_{dR}]$ at time k, k-1 and k-2 are fed into the RBFNI to estimate the plant output $\hat{Y} = [\hat{i}_q, \hat{i}_d]$ at time k+1. The difference between the actual output vector Y and the estimated output vector $\hat{Y}$ at time k forms the error vector $E_I(k)$, which is then used to train the RBFNI before the next sampling instant. The reference model utilizes the reference inputs R^* to generate the desired plant outputs Y^* at each time step, which are used to guide the plant outputs $Y = [i_q, i_d]$ to a desired steady state set point. In this paper, $R^* = [Q^*, P^*]$ are used as the reference inputs; thereby Y^* are calculated to be the constant values $[i_q^*, i_d^*]$ at each time step. The RBFNC is used to replace two conventional PI controllers (PI$_\text{d}$ and PI$_\text{q}$) in Fig. 3. The inputs of the RBFNC are the plant outputs at time k-1, k-2 and k-3. It in turn generates the control signals as the plant inputs in order to drive the plant outputs to the desired values. The detailed design and training process for the RBFNI and RBFNC has been discussed in [6].

C. Missing Sensor Restoration Algorithm (MSR)

Figure 4 shows the structure of a MSR block [7], [8]. It consists of a dynamic auto-associative neural network (auto-encoder) and a particle swarm optimizer (PSO).



(a)

(b)

Fig. 4. Overall structure of MSR: (a) Training phase of the auto-encoder. (b) On-line restoration of missing sensor data.

1) Auto-Encoder (Fig. 4(a)): The auto-encoder is a multi-layer perceptron (MLP) neural network with butterfly structure [7], [8]. It has the same number of inputs and outputs, but the number of neurons in the hidden layer is less than that of the inputs. This particular structure creates a bottleneck in the feedforward path of the auto-encoder, enabling it to capture the correlations between the redundant inputs. The inputs of the auto-encoder, S, consist of the vector, X, at the present time step as well as at the previous two time steps (*i.e.*, $S(k) = [X(k), X(k\text{-}1), X(k\text{-}2)]$). The use of the time-delayed inputs enables the auto-encoder to capture the auto-correlations of each variable in its input vector X.

The auto-encoder is firstly trained without any missing sensor. During the training, the two PI controllers (PI$_\text{d}$, PI$_\text{q}$) are deactivated as shown in Fig. 3 and the steady state plant inputs v_{cqS} and v_{cdS} are disturbed by pseudorandom binary signals (PRBS) from an external source at each time step k, given by

$$PRBS\_v_{cd}(k) = 0.1 \cdot |v_{cdS}| \cdot [rand2(k) + rand3(k) + rand5(k)]/3 \tag{4}$$

$$PRBS\_v_{cq}(k) = 0.1 \cdot |v_{cqS}| \cdot [rand2(k) + rand3(k) + rand5(k)]/3 \quad (5)$$

where *rand*2, *rand*3 and *rand*5 are uniformly distributed random numbers in [-1, 1] with frequencies 2 Hz, 3 Hz and 5 Hz, respectively; $|v_{cdS}|$ and $|v_{cqS}|$ are the magnitudes of v_{cqS} and v_{cdS}, respectively. By feeding forward the data through the auto-encoder and adjusting its weight matrices (using backpropagation algorithm), W and V, the auto-encoder is trained to map its inputs to its outputs. The detailed description of the auto-encoder training process has been given in [9].

2) Particle Swarm Optimizer (PSO): The particle swarm optimizer [10]-[12] is an evolutionary computational algorithm. It searches for the optimal solution from a population of moving particles. Each particle represents a potential solution and has a position in the problem space represented by a position vector x_i. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector v_i. At each time step, a fitness function f representing a quality measure is calculated by using x_i as input. Each particle keeps track of its individual best position $x_{i,pbest}$, which is associated with the best fitness it has achieved so far. Furthermore, the best position among all the particles obtained so far in the swarm is kept track of as x_{gbest}. This information is shared by all particles. The PSO algorithm is implemented in the following iterative procedure to search for the optimal solution.

(i) Initialize a population of particles with random positions and velocities of M dimensions in the problem space.

(ii) Define a fitness measure function to evaluate the performance of each particle.

(iii) Compare each particle's present position x_i with its $x_{i,pbest}$ based on the fitness evaluation. If the current position x_i is better than $x_{i,pbest}$, then set $x_{i,pbest} = x_i$.

(iv) If $x_{i,pbest}$ is updated, then compare each particle's $x_{i,pbest}$ with the swarm best position x_{gbest} based on the fitness evaluation. If $x_{i,pbest}$ is better than x_{gbest}, then set $x_{gbest} = x_{i,pbest}$.

(v) At iteration k, a new velocity for each particle is updated by

$$v_i(k+1) = w \cdot v_i(\mathrm{k}) + c_1\phi_1(x_{i,pbest}(k) - x_i(k)) + c_2\phi_2(x_{gbest}(k) - x_i(k)) \quad i = 1, 2, \cdots, N \quad (6)$$

(vi) Based on the updated velocity, each particle then changes its position according to the following equation.

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad i = 1, 2, \cdots, N \quad (7)$$

(vii) Repeat steps (iii)-(vi) until a criterion, usually a sufficiently good fitness or a maximum number of iterations is achieved. The final value of x_{gbest} is regarded as the optimal solution of the problem.

In (6), c_1 and c_2 are positive constants representing the weighting of the acceleration terms that guide each particle toward the *individual best* and the *swarm best* positions $x_{i,pbest}$ and x_{gbest}, respectively; ϕ_1 and ϕ_2 are uniformly distributed random numbers in [0, 1]; w is a positive inertia weight developed to provide better control between exploration and exploitation; N is the number of particles in the swarm. The velocity v_i is limited to the range $[-v_{max}, v_{max}]$. If the velocity violates this limit, it is set to the relevant upper- or low-bound value. The last two terms in (6) enable each particle to perform a local search around its individual best position $x_{i,pbest}$ and the swarm best position x_{gbest}. The first term in (6) enables each particle to perform a global search by exploring a new search space.

The multi-agent (particles) searching and information sharing mechanism in PSO enable a fast and efficient search for the optimal solution. In many cases, the PSO algorithm yields superior performance to other evolutionary computation algorithms, such as genetic algorithms. In this paper, the values of c_1 and c_2 in (6) are chosen as 2; the number of particles N is chosen as 20; the inertia constant w is fixed at 0.5. The fitness measure function f_i for each particle is defined as (Fig. 4 (b)):

$$f_i = \| E_S \| = \| S_H - \hat{S}_H(x_i) \| \qquad i = 1, 2, \cdots, N \quad (8)$$

where S_H represents the healthy sensor measurements; $\hat{S}_H$ represents the replicated healthy sensor data from the auto-encoder; $x_i = S_M$ represents the estimates of the missing sensor data. The objective of the PSO is to search for the optimal estimates of the missing sensor measurements which minimize the value of the fitness measure function.

3) Missing Sensor Restoration (Fig. 4(b)): It is assumed that some sensor data are missing only after the training of the auto-encoder is over. As a consequence, the outputs of the auto-encoder, $\hat{S}_H$, no longer match its inputs S_H when one or more sensor measurements are missing, and the error signal E_S becomes significant. In this case, the PSO module in the feedback search loop of the MSR is activated and only the healthy sensor data S_H are fed directly into the auto-encoder. The error signal, E_S, is then used by the PSO as a fitness signal to search the solution space for the optimal estimates of the missing sensor readings based on the correlations established by the auto-encoder between the healthy data and the missing data. In each iteration, the outputs of the PSO, S_M, which represent the estimated missing sensor data, are fed together with the healthy sensor data, through the auto-encoder to reduce the error E_S. Theoretically, good estimates of the missing data should drive the fitness signal from the auto-encoder to zero, indicating a perfect match. In real practice, once the error is below a pre-determined threshold value, the output of the auto-encoder, S_R, is regarded as a feasible guess.

The use of the auto-encoder does not need an explicit plant model. In addition, the PSO search algorithm is simple, fast, and efficient due to its multi-agent searching structure and information sharing mechanism. Therefore, the overall missing sensor restoration algorithm can quickly locate the optimal estimates of the readings from the missing sensors.

D. Design of SERS

The SERS in Fig. 5 consists of three parallel MSR blocks; each of them has the same structure as shown in Fig. 4 and only use one of the three current variables, i_a, i_b, and i_c, as input. Therefore, each MSR block evaluates the status of one current sensor measurement. If any MSR block determines that the current sensor (i_a, i_b or i_c) is missing, it will performs a one-dimensional search to restore the missing current. The variables i_{aR}, i_{bR} and i_{cR} represent the restored sensor readings from MSR1, MSR2 and MSR3, respectively. Since a necessary condition for the MSR to work is that the number of healthy inputs must equal or exceed the number of degrees of freedom in the hidden layer, in this application, the dimensions of the input, hidden and output layers of three MSR blocks are chosen to be 21-12-21. The output vector of the SERS, I_R, contains the total restored sensor measurements from all three MSR blocks; but I_H, contains other healthy sensor readings in the vector I.

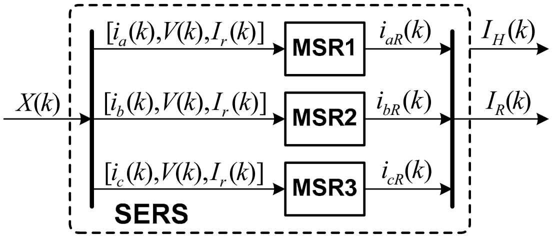


Fig. 5. Structure of the sensor evaluation and restoration scheme (SERS): $X = [I,\ V,\ I_r]$, $I = [i_a, i_b, i_c]$, $V = [v_{ca}, v_{cb}, v_{cc}]$, and $I_r = [i_{ra}, i_{rb}, i_{rc}]$.

The entire sensor evaluation and restoration process of the SERS is implemented in two stages: sensor evaluation (stage I) and missing sensor restoration (stage II). In stage I, each MSR evaluates the status of one current measurement (i_a, i_b or i_c) in its input vector by checking the value of the Euclidean norm of the error signal $||E_S||$ of the auto-encoder as shown in Fig. 4. At normal operating conditions, with a well-trained auto-encoder, $||E_S||$ should be acceptably small (In real applications, a threshold value can be specified depending on the system properties). If one or more current sensors are missing, the outputs of the corresponding auto-encoders no longer match their inputs and the values of $||E_S||$ become significant.

Table 1 gives all eight cases of the status of i_a, i_b and i_c which can be determined in stage I. The positive sign, +, indicates that the value of $||E_S||$ of the corresponding MSR (MSR1, MSR2 or MSR3) is significant; while the negative sign, -, indicates that the value of $||E_S||$ of the corresponding MSR is below a pre-specified threshold value. If the SERS identifies that one or more current sensors are missing, the procedure goes to stage II, in which each MSR block with missing current is activated to restore the missing sensor data. Table 1 shows the restored missing sensor by each MSR in each case during this stage.

TABLE I
SENSOR EVALUATION AND MISSING SENSOR RESTORATION

Case No.	Missing Sensors	Stage I: Sensor Evaluation			Stage II: Restored Sensors		
		MSR1	MSR2	MSR3	MSR1	MSR2	MSR3
0	none	-	-	-			
1	i_a	+	-	-	i_{aR}		
2	i_b	-	+	-		i_{bR}	
3	i_c	-	-	+			i_{cR}
4	i_a, i_b	+	+	-	i_{aR}	i_{bR}	
5	i_b, i_c	-	+	+		i_{bR}	i_{cR}
6	i_a, i_c	+	-	+	i_{aR}		i_{cR}
7	i_a, i_b, i_c	+	+	+	i_{aR}	i_{bR}	i_{cR}

The use of parallel structure to design the SERS is based on the following reasoning. 1) This structure enables the SERS to evaluate the status of the crucial sensor measurements and determine which sensor or sensors are missing, instead of relying on a sensor evaluation scheme in [7] or a sensor monitor in [8]. 2) Each MSR only searches in a one-dimensional space to restore one missing sensor reading for any of the seven cases, which is faster than using only one MSR [7], [8] to search in a multi-dimensional space in order to restore multiple missing sensor measurements. 3) The required degree of data redundancy for restoring one missing sensor is lower than that of restoring multiple missing sensors. 4) This structure is simple and three MSR blocks are implemented in parallel to save searching time.

IV. Simulation Results

The dynamic performance of the proposed FTNC is evaluated at two different operating points by applying three-phase short circuit and missing sensor tests.

A. Tests at the Operating Point Where Controllers are Designed

The RBFNC is trained and the CONVC is tuned at a specific operating condition (called OP-I), where the generator operates with a pre-fault rotor angle of 42.6°. A three-phase short circuit is applied to the receiving end of line 2 at $t = 15$ s and 100 ms thereafter, line 2 is cleared out from the system. Three missing sensor tests are then applied from $t = 15.1$ s during this post-fault transient state: 1) Case I – i_b missing; 2) Case II – i_b and i_c missing; 3) Case III – i_a, i_b and i_c missing.

Figures 6, 7 and 8 show the results of the rotor angle δ for Cases I, II and III, respectively. These results show that the damping control of the FTNC is more efficient than that of the CONVC during the post-fault transient state. During the first swing after the fault is applied, the FTNC is already providing significant damping compared to that provided by the CONVC. By continuously training the RBFNI and the RBFNC, the FTNC drives the plant successfully and quickly to a new operating point with a rotor angle δ = 46.3° at the steady state. Moreover, comparing the curves by FTNC with and without missing sensors, the transient performance of the FTNC only degrades slightly due to missing sensor data. However, comparing the curves CONVC and the curves FTNC with missing sensor or sensors, the transient

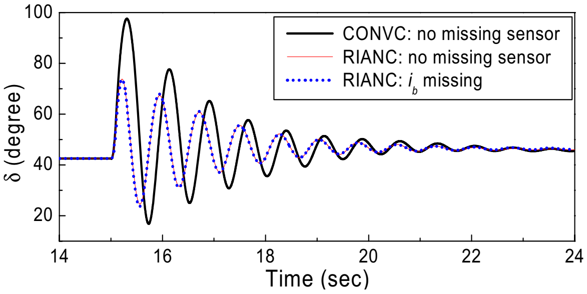


Fig. 6. A 100 ms three-phase short circuit at 15.0 s at OP-I; Case I - i_b missing from 15.1 s.

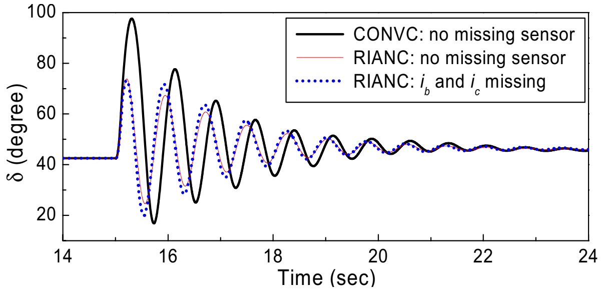


Fig. 7. A 100 ms three-phase short circuit at 15.0 s at OP-I; Case II - i_b and i_c missing from 15.1 s.

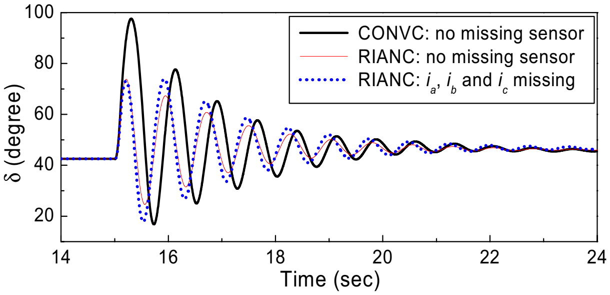


Fig. 8. A 100 ms three-phase short circuit at 15.0 s at OP-I; Case III - i_a, i_b and i_c missing from 15.1 s.

performances of the FTNC with missing sensor measurements are still better than those of the CONVC used by the SSSC without any missing sensor. In this sense, the proposed FTNC provides a fault tolerant robust control for the SSSC.

B. *Tests at a Different Operating Point*

The transient performance of the FTNC is now re-evaluated at a different operating point (OP-II), where the pre-fault rotor angle of the generator changes to 50.1°; line 1 is now open during this entire test. The parameters of the controllers are the same as those used in the test at OP-I, *i.e.*, the RBFNC has not been trained and the CONVC has not been tuned for OP-II; but the SERS has been trained for this operating point. A 100 ms three-phase short circuit is applied to the receiving end of line 2 at t = 15 s. Again, three missing sensor tests same as those in the previous subsection are applied during this post-fault transient state.

Figures 9, 10 and 11 show the results of the rotor angle δ for Cases I, II and III, respectively. These results indicate that the CONVC fails to drive the system back to the steady state after this large disturbance. However, the FTNC still provides the efficient control even if there are sensors missing or not. These results prove that the proposed FTNC provides improved transient performance over the CONVC and a fault-tolerant control for the SSSC over a wide range of operating conditions.

Under balanced operation, missing one sensor might be simply restored using the relationship $i_a + i_b + i_c = 0$. However, the use of SERS is still necessary because it identifies which sensor is missing. This can not be achieved by only using that relationship. Moreover, power systems might experience unbalanced operations. In this case, the relationship above cannot be used to restore the missing sensor. The use of the SERS to identify and restore the missing sensors under unbalanced operating condition has been discussed and the simulation results have been given in [9].

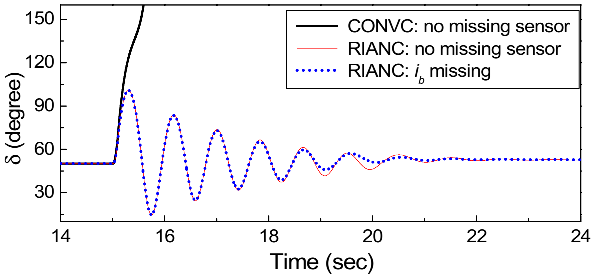


Fig. 9. A 100 ms three-phase short circuit at 15.0 s at OP-II; Case I - i_b missing from 15.1 s.

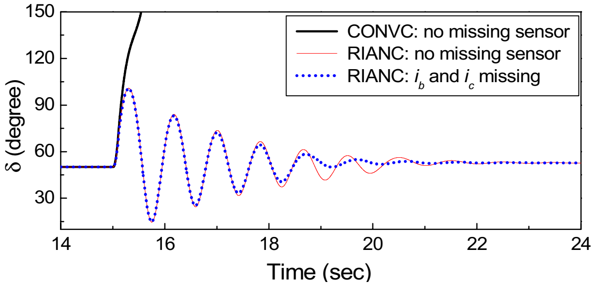


Fig. 10. A 100 ms three-phase short circuit at 15.0 s at OP-II; Case II - i_b and i_c missing from 15.1 s.

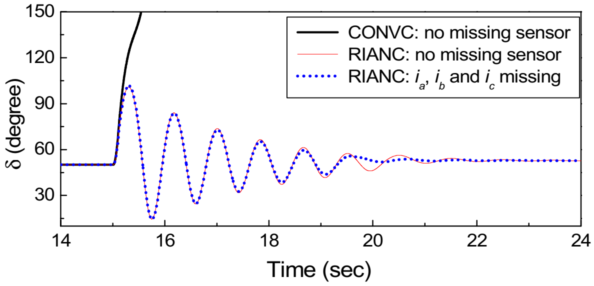


Fig. 11. A 100 ms three-phase short circuit at 15.0 s at OP-II; Case III - i_a, i_b and i_c missing from 15.1 s.

V. Conclusion

This paper has proposed a fault-tolerant indirect adaptive neuro-controller (FTNC) for the internal control of an SSSC, which combines a suitably designed sensor evaluation and (missing sensor) restoration scheme (SERS), a RBF neuro-identifier (RBFNI) and a RBF neuro-contrroller (RBFNC). The SERS is designed using the auto-associative neural networks (auto-encoder) and the particle swarm optimizer (PSO). This FTNC is able to provide efficient control to the SSSC when some crucial sensor measurements are unavailable.

Simulation studies are carried out at two operating conditions for the CONVC and the FTNC without any missing sensor, as well as for the FTNC with single and multiple phase current sensors missing; results show that the transient performances of the proposed FTNC with or without missing sensor measurements are both superior to the conventional linear PI controllers used by the SSSC without any missing sensor over a wide range of system operating conditions.

VI. References

[1] L. Gyugyi, C. D. Schauder, and K. K. Sen, "Static synchronous series compensator: a solid-state approach to the series compensation of transmission lines," *IEEE Trans. Power Delivery*, vol. 12, no. 1, Jan. 1997, pp. 406-417.

[2] P. F. Ribeiro, B. K. Johnson, M. L. Crow, A. Arsoy, and Y. Liu, "Energy storage systems for advanced power application," *Proceedings of the IEEE*, vol. 89, no. 12, Dec. 2001, pp. 1744-1756.

[3] L. Zhang, M. L. Crow, Z. Yang, and S. Chen, "The steady state characteristics of an SSSC integrated with energy storage," in *Proc. 2001 IEEE Power Engineering Society Winter Meeting*, Jan. 28-Feb. 1, 2001, Columbus, OH, USA, vol. 3, pp. 1311-1316.

[4] B. A. Renz, *et al*, "AEP unified power flow controller performance," *IEEE Trans. Power Delivery*, vol. 14, no. 4, pp. 1374-1381, Oct. 1999.

[5] Bruce S. Rigby and R. G. Harley, "An improved control scheme for a series capacitive reactance compensator based on a voltage-source inverter," *IEEE Trans. Industry Applications*, vol. 34, no. 2, Mar./Apr. 1998, pp. 355-363.

[6] W. Qiao and R. G. Harley, "Indirect adaptive internal neuro-control for a static synchronous series compensator (SSSC) connected to a power system," in *Proc. the 31st Annual Conference of the IEEE Industrial Electronics Society*, Nov. 6-10, 2005, Raleigh, NC, USA, pp. 50-55.

[7] M. A. El-Skarkawi and Robert J. Marks II, "Missing sensors restoration for system control and diagnostics," in *Proc. the 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, Aug. 24-26, 2003, Atlanta, GA, USA, pp. 338-341.

[8] W. Qiao, Z. Gao, and R. G. Harley, "Continuous on-line identification of nonlinear plants in power systems with missing sensor measurements," in *Proc. 2005 International Joint Conference on Neural Networks*, July 31-Aug. 4, 2005, Montreal, QC, Canada, pp. 1729-1734.

[9] W. Qiao, R. G. Harley and G. K. Venayagamoorthy, "A fault-tolerant P-Q decoupled control scheme for static synchronous series compensator," to be presented at the *IEEE PES 2006 Annual Meeting*, Montreal, QC, Canada, June 18-22, 2006.

[10] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE International Conference on Neural Networks*, Nov. 27-Dec. 1, 1995, Piscataway, NJ, USA, vol. 4, pp. 1942-1948.

[11] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. 1998 IEEE International Conference on Evolutionary Computation*, May 4-9, 1998, Piscataway, NJ, USA, pp. 69-73.

[12] M. Clerc and J. Kennedy, "The particle swarm – explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 1, Feb. 2002, pp. 58-73.