Fault-Tolerant Indirect Adaptive Neurocontrol for a Static Synchronous Series Compensator in a Power Network With Missing Sensor Measurements

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

Abstract—Identification and control of nonlinear systems depend on the availability and 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 (referred to as missing sensor measurements in this paper). This paper proposes a novel fault-tolerant indirect adaptive neurocontroller (FTIANC) for controlling a static synchronous series compensator (SSSC), which is connected to a power network. The FTIANC consists of a sensor evaluation and (missing sensor) restoration scheme (SERS), a radial basis function neuroidentifier (RBFNI), and a radial basis function neurocontroller (RBFCN). The SERS provides a set of fault-tolerant measurements to the RBFNI and RBFCN. The resulting FTIANC is able to provide fault-tolerant effective control to the SSSC when some crucial time-varying sensor measurements are not available. Simulation studies are carried out on a single machine infinite bus (SMIB) as well as on the IEEE 10-machine 39-bus power system, for the SSSC equipped with conventional PI controllers (CONVC) and the FTIANC without any missing sensors, as well as for the FTIANC with multiple missing sensors. Results show that the transient performances of the proposed FTIANC with and without missing sensors are both superior to the CONVC used by the SSSC (without any missing sensors) over a wide range of system operating conditions. The proposed fault-tolerant control is readily applicable to other plant models in power systems.

Index Terms—Fault-tolerant neurocontrol, missing sensor restoration, particle swarm optimization, radial basis function (RBF) network, static synchronous series compensator (SSSC).

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 so-called flexible alternating current (ac) transmission system (FACTS) devices [1]. It is able to rapidly provide both capacitive and inductive impedance compensation independent of the line current. By coupling an additional energy storage system to its direct current (dc) terminal, the SSSC can also provide simultaneous active power compensation, which further enhances its capability in power flow control, power oscillation damping and improving transient stability [1]–[3].

Power systems containing power-electronics-based FACTS devices are large-scale nonlinear, nonstationary, multivariable systems with dynamic characteristics over a wide range of operating conditions. Conventionally, linear control techniques are used to design FACTS controllers from a linearized system model with fixed parameters at a specific operating point [1]–[3]. Final settings are made using field tests at one or two operating points. In a real power system, the SSSC and the associated power network cannot be accurately modeled as a linear system with fixed and known parameters. Therefore, at other operating points or in the event of a major disturbance, the linear controllers may not be able to guarantee acceptable performances or stability. The drawbacks of using linear controllers to control a nonlinear system can be overcome by using neural-network-based nonlinear intelligent control techniques. The artificial neural networks (ANNs) are good at identifying other operating points or in the event of a major disturbance, the linear controllers may not be able to guarantee acceptable performances or stability. The drawbacks of using linear controllers to control a nonlinear system can be overcome by using neural-network-based nonlinear intelligent control techniques. The artificial neural networks (ANNs) are good at identifying other operating points or in the event of a major disturbance, the linear controllers may not be able to guarantee acceptable performances or stability. The drawbacks of using linear controllers to control a nonlinear system can be overcome by using neural-network-based nonlinear intelligent control techniques. The artificial neural networks (ANNs) are good at identifying other operating points or in the event of a major disturbance, the linear controllers may not be able to guarantee acceptable performances or stability. The drawbacks of using linear controllers to control a nonlinear system can be overcome by using neural-network-based nonlinear intelligent control techniques. The artificial neural networks (ANNs) are good at identifying
A fault-tolerant control system is capable of detecting the faults and canceling the effects of the faults or attenuating them to an acceptable level. This improves system reliability, maintainability and survivability. In principle, in order to achieve fault tolerance, system redundancy is necessary [5]. 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.

State estimation is commonly used to identify state variables that are not accessible for direct measurements, and therefore can be modified to restore missing sensor data [6]. This technique is based on the analysis of the system model and the redundancy of system state variables. By deriving closed-form solutions for the variables corresponding to the missing sensors, the lost data are explicitly represented by the remaining available data. However, for many systems, this model-based method converges slowly and the closed-form solutions can be unfeasible. Moreover, accurate system models are usually unavailable in real system applications.

The use of autoassociative neural network (autoencoder) [7]–[9] offers an alternative approach for missing sensor restoration. The autoencoder is a feedforward multilayer neural network. It is trained to perform an identity mapping, where the network inputs are reproduced at the output layer. The network contains a hidden “bottleneck” layer which has fewer nodes than the input and output layers. The dimensionality reduction through the input-to-hidden layer enables the network to extract significant features in data, without restriction on the character of the nonlinearities in the data (nonlinear feature extraction). Hence, the hidden layer captures the correlations between all input data. On the other hand, the dimensionality expansion through the hidden-to-output layer enables the network to reproduce the high-dimensional inputs at the output layer. If one or more input data are missing, the correlations between the input data established by the autoencoder can be used to search for the optimal estimates of the missing data.

Based on an autoencoder, a missing sensor restoration algorithm (MSR) was designed in [9] and [10] to restore missing sensor data, which were constant values at steady state. However, the MSR relied on an external sensor evaluation scheme [9] or a sensor monitor [10] to evaluate the integrity of sensor data and detect which sensor or sensors are missing. In real system applications, such a sensor evaluation scheme or sensor monitor might be difficult to implement. Moreover, [9] and [10] did not specifically address the case when multiple sensors were missing; only one sensor measurement was assumed to be missing in the test. To restore multiple missing sensor measurements, the input space of the autoencoder in [9] and [10] must be sufficiently high in order to provide the required degree of data redundancy. The resulting MSR must search in a multidimensional space with a high-dimensional autoencoder in order to restore all missing data. This results in a time-consuming search. Consequently, the MSR is not fast enough and cannot be used directly for online applications, such as the real-time control problem in this paper, when multiple sensors are missing.

This paper, therefore, proposes a novel online sensor evaluation and (missing sensor) restoration scheme (SERS). It employs several cascaded autoencoders to capture the correlations between the redundant sensor measurements (through training). This structure enables the SERS to work online to evaluate the integrity of the crucial time-varying sensor measurements which determine the behavior of the SSSC controllers. If one or more sensor measurements are missing, the SERS can quickly detect which sensor or sensors are missing. Based on the correlations established by each autoencoder between the missing sensor data and the remaining healthy sensor data, the SERS utilizes the particle swarm optimization (PSO) algorithm [11]–[13] to quickly online restore the missing sensor measurements. The proposed SERS is independent of system models and is fast and efficient for online detection and restoration of multiple time-varying missing sensor measurements.

Based on the SERS, this paper proposes a novel fault-tolerant indirect adaptive neurocontroller (FTIANC) for controlling an SSSC connected to a power network. The FTIANC consists of a SERS, a radial basis function neuroidentifier (RBFNI) and a radial basis function neurocontroller (RBFNC). 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 [14]. The SERS works online to provide a set of fault-tolerant complete sensor measurements for the RBFNI and the RBFNC. This guarantees a fault-tolerant control for the SSSC.

Simulation studies are carried out for the SSSC connected to a single-machine infinite bus (SMIB) power system as well as the IEEE 10-machine 39-bus power system. Results show that the proposed FTIANC improves the dynamic performance and reliability of the SSSC and the power network over a wide range of system operating conditions.

II. SSSC AND POWER NETWORK MODEL

Fig. 1 shows an SSSC with its controllers connected to a SMIB power system. The generator (with 500-MW power rating) is modeled together with its automatic voltage regulator (AVR), exciter, and turbine governor dynamics taken into account [15]. The generated power is transmitted to a power system through three 230-kV three-phase transmission lines, which represent the different loops between the generator and the power system. The impedances of the three lines are \( Z_1 = 0.02 + j0.4 \) per unit (p.u.), \( Z_2 = 0.03 + j0.6 \) p.u., and \( Z_3 = 0.04 + j0.8 \) p.u. (on 500-MW, 230-kV bases), respectively. A three-phase balanced electric load draws a constant active power of \( P_L = 0.1 \) p.u. with a constant power factor of 0.85 from the sending-end bus of the 230-kV transmission lines. The SSSC is connected at the receiving end of line 3 through a series injection transformer for dynamic power flow control. It is modeled as a detailed switching-level voltage source converter with an energy storage system coupled with the dc-link. The system is simulated in the PSCAD/EMTDC environment [16].

The SSSC is controlled by a P–Q decoupled power flow control scheme using two conventional PI controllers (Pld and Pq, called CONVC) as described in [14]. In Fig. 2, \( P^* \) and \( Q^* \) are
the desired reference values of the transmitted real and reactive powers at the receiving end of line 3, which are used to determine the reference values of the \( d \)-axis component \( i_d^* \) and the \( q \)-axis component \( i_q^* \) of the line current at the SSSC ac terminal, respectively. The instantaneous three-phase currents of line 3, \( i_a, i_b, \) and \( i_c \), 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) [17]. 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 the two PI controllers. The outputs of the PI controllers, in turn, determine the modulation index \( m \) and phase shift \( \alpha \) applied to the pulse-width modulation (PWM) module to drive the gate turnoff (GTO) thyristors of the inverter. The main objective of this SSSC is to control the transmitted active and reactive power at the receiving end of line 3. The reference values \( P^* \) and \( Q^* \) can be determined by the results of the power flow calculation at a specific operating point to achieve some form of optimal operation (e.g., optimal power flow) of the network, while considering the operating limits of the SSSC.

III. MISSING SENSOR RESTORATION USING AUTOENCODER

An autoencoder can learn the data correlations through inspection of historical data. Once trained, data correlations established by the autoencoder can be used by some search algorithms (e.g., PSO in this paper) to restore missing data if the data dependency is sufficiently strong. The unique point of convergence of these search algorithms using the autoencoder for missing sensor restoration rests on the concepts of contractive and nonexpansive mapping [8], [18].

A. Missing Sensor Restoration Algorithm

By extending the missing sensor restoration (MSR) algorithm in [9] and [10], this section presents an MSR that is suitable for the restoration of time-varying missing sensor measurements, as shown in Fig. 3. It consists of an autoencoder and a PSO.

1) Autoencoder [Fig. 3(a)]: The autoencoder used in this paper is a three-layer feedforward neural network with the sigmoidal nonlinearity in the hidden layer. It is trained to capture the correlations between the redundant inputs. The overall
input–output mapping for the autoencoder $\hat{g} : S \in R^p \rightarrow \hat{S} \in R^p$ is

$$\hat{S}_i = \hat{g}(S, W, V_i) = V_i \cdot d(S, W) = \sum_{j=1}^{q} V_{ij}d_j(S, W_j)$$ (1)

where $p$ is the dimension of the input and output vectors; $q$ is the number of the hidden-layer neurons; $S$ is the input vector; $\hat{S}_i$ is the $i$th output; $W$ and $V$ are input and output weight matrices, respectively; and $d_j(S, W_j)$ is the sigmoid activation function of the $j$th hidden-layer neuron, given by

$$d_j(S, W_j) = \frac{1}{1 + e^{-a_j}}$$ (2)

where

$$a_j = W_j \cdot S = \sum_{i=1}^{p} W_{ji}S_i.$$ (3)

Suppose the vector $X = [x_1, x_2, \ldots, x_n]$ consists of the measured time-varying sensor data at each time sampling $k$. In power systems, the time-varying variables are generally periodic and in the sinusoidal form, given by

$$x_i(k) = A_i \sin(\omega_i k + \phi_i), \quad i = 1, \ldots, n.$$ (4)

Each periodic time-varying variable $x_i(k)$ is autocorrelated and its feature is determined by the magnitude $A_i$, the angular frequency $\omega_i$, and the phase angle $\phi_i$. Autocorrelations can be used to extract the significant features buried in a periodic time-varying signal, and therefore, are useful to restore the missing time-varying sensor measurements. The autocorrelation of each variable in the vector $X$ can be captured by the autoencoder using the time-delayed inputs. As shown in Fig. 3(a), the inputs of the autoencoder $S$ consist of the vector $X$ at the present time step as well as at the previous two time steps (i.e., $S(k) = [X(k), X(k-1), X(k-2)]$). The autoencoder is trained to reproduce its input data at its output layer. Once trained, the cross correlations between different sensor data as well as the autocorrelations of each sensor data in the vector $X$ are established by the autoencoder.

There exist other complex structures, instead of the simple three-layer structure, for the autoencoder. For instance, Kramer [19] proposed a four-layer autoassociative neural network for missing sensor detection and restoration; and Hsieh [20] used a five-layer autoassociative neural network for nonlinear principal component analysis. Generally, the autoassociative networks with more complex structures are able to solve more sophisticated autoassociation problems, and therefore, can provide better performance for missing sensor detection and restoration. However, the use of complex autoassociative neural networks increases the computational cost, and consequently, is not suitable for the applications of real-time system identification and control.

2) Missing Sensor Restoration [Fig. 3(b)]: After training the autoencoder, the input data of the autoencoder are reproduced at its output layer. If one or more sensor measurements are missing, the outputs of the autoencoder in Fig. 3(a) $\hat{S}$ no longer match its inputs $S$, and the error signal $E_S$ becomes significant. However, it should be pointed out that although the autoencoder can detect sensor missing based on the value of the error signal $E_S$, it cannot identify which sensor or sensors are missing. This problem is solved by the SERS proposed in the next section. At this stage, it is assumed that the information of which sensor or sensors are missing is available. In this case, the PSO [11]–[13] module in the feedback search loop of the MSR is activated and only the healthy sensor data $S_H$ are fed directly into the autoencoder. The PSO iteratively searches the solution space for the optimal estimates of the missing sensor readings based on the correlations established by the autoencoder between the healthy data and the missing data. At each iteration, the estimated missing sensor data by the PSO $S_M$ are fed together with the healthy sensor data, through the autoencoder to reduce the value of the following fitness measure function $f$, defined by:

$$f = || E_s || = || S_H - \hat{S}_H ||$$ (5)

where $S_H$ represents the actual healthy sensor data; $\hat{S}_H$ represents the reproduced healthy sensor data from the autoencoder; and

$$\hat{S}_{Hj}(S_M) = \hat{g}(S_H, S_M, W, V_j) = \sum_{l=1}^{q} V_{jl}d_l(S_H, S_M, W_l)$$ (6)

is the $j$th variable in the vector $\hat{S}_H$. Theoretically, good estimates of the missing data should drive the fitness signal $||E_S||$ from the autoencoder to zero, indicating a perfect match. In the real application, once the error is below a predetermined threshold value, the output of the autoencoder $S_R$ is regarded as the best estimates of the missing sensor data.

The use of the autoencoder does not need an explicit plant model. In addition, because of some attractive features, e.g., simple implementation, small computational load, and fast convergence, the PSO algorithm can provide a fast and efficient
search for the optimal solution. In many cases, it yields superior performance to other evolutionary computation algorithms, such as genetic algorithms [21]. Therefore, the overall missing sensor restoration algorithm can quickly find the optimal estimates of the readings from the missing sensors.

B. Convergence of the MSR

The unique convergence of the MSR can be shown through the concepts of contractive and nonexpansive mapping [8], [18]. An operator \( \Theta \) mapping \( \mathbb{R}^N \to \mathbb{R}^N \) is contractive if for any vectors \( x, y \in \mathbb{R}^N \) it follows that \( ||\Theta x - \Theta y|| \leq ||x - y|| \). According to Banach fixed-point theorem, if \( \Theta \) is a contractive mapping, then there exists a unique fixed point \( x_0 \) for which \( \Theta x_0 = x_0 \); if \( \Theta \) is nonexpansive, then there may exist a plurality of fixed points \( x_0 \) for which \( \Theta x_0 = x_0 \).

A well-trained autoencoder constructs a nearly nonexpansive mapping \( \hat{g} \) between its input vector \( S \) and output vector \( \hat{S} \) because of \( \hat{S} = \hat{g}(S) \approx S \). It has been shown in [18] that there exists a unique point of convergence for a well-trained autoencoder given an “operating point” defined by the set of healthy sensors. This convergence point should be reached regardless of how the missing sensors are initialized. A necessary condition for autoencoder to work is that the number of healthy inputs must equal or exceed the number of degrees of freedom in the hidden layer [19].

IV. FAULT-TOLENTIT ADAPTIVE NEUROCONTROLLER

The schematic diagram of the FTIANC connected to the plant (the dashed line block in Fig. 2) is shown in Fig. 4. The FTIANC consists of an SERS, an RBFNC, and an RBFNC. 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.

In Fig. 4, \( U = [v_{ca}, v_{qr}] \) and \( Y = [i_d, i_q] \) are the plant input and output vectors, respectively. In this paper, \( i_d \) and \( i_q \) are two crucial variables to determine the behavior of the RBFNI and the RBFNC. The values of \( i_d \) and \( i_q \) are calculated from the three-phase currents \( i_a, i_b, \) and \( i_c \) (line 3 (Fig. 1), which are time-varying variables measured by the metering current transformers (called current sensors in this paper). Therefore, in this paper, \( i_a, i_b, \) and \( i_c \) are three crucial measurements; missing any of them results in the loss of both \( i_d \) and \( i_q \).

The vector \( V = [v_{ca}, v_{db}, v_{qr}] \) consists of the injected three-phase voltages of the SSSC, measured by the metering potential transformers (called voltage sensors in this paper). 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 the RBFNC, but are used to build the correlations with the variables in the vector \( I \). The use of the extra measurements \( V \) and \( I_r \) to form the input vector \( X = [I, V, I_r] \) of the SERS provides necessary data redundancy in order to restore two or three missing currents in the vector \( I \).

The SERS is designed by using the MSR algorithm and the features of the three-phase variables in power systems. It evaluates the integrity of the crucial vector \( I \), which determines the behavior of the RBFNI and the RBFNC. If one or more sensor data in \( I \) are missing, the SERS detects which sensor or sensors are missing. This information is then used by the PSO to search for the best estimates of all missing sensor data. In this paper, a small population of particles (five particles) is used in the PSO algorithm to reduce the computational cost of the PSO search algorithm for real-time implementation. The output vector of

![Fig. 4. Schematic diagram of the FTIANC connected to the plant. SRRFT denotes synchronously rotating reference frame transformation.](image-url)
the SERS $I_R$ contains the restored missing sensor data; however, $I_H$ contains other healthy sensor data in the vector $I$. The variables $[I_R, I_H]$ are transformed into the $d$-axis and $q$-axis current components $Y_R = [\dot{i}_{dR}, \dot{i}_{qR}]$ by applying the SRRFT. In this application, missing any of the three currents $i_d$, $i_q$, and $\dot{i}_r$ results in the loss of both $i_d$ and $i_q$. Therefore, the calculated currents $\dot{i}_{dR}$ and $\dot{i}_{qR}$ from the SRRFT block, by using the restored currents from the SERS, are then used by the RBFN and the RBFNC as the restored actual plant outputs for continuous online identification and control. This provides a fault-tolerant control for the SSSC. If there is no sensor missing, the vector $Y_R$ is exactly the same as the actual plant output vector $Y$.

B. Design of the RBFN and the RBFNC

1) Radial Basis Function Neural Network (RBFNN): The RBFN and the RBFNC are each a three-layer RBFNN with the Gaussian density function as the activation functions in the hidden layer. The overall input–output mapping for the RBF network $\tilde{f}: X \in \mathbb{R}^n \rightarrow Y \in \mathbb{R}^m$ is

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

where $x$ is the input vector, $C_j \in \mathbb{R}^n$ 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 $|| \cdot ||$ denotes the Euclidean norm.

2) Design of the RBFN: The RBFN is developed using the nonlinear autoregressive moving average with exogenous inputs (NARMAX) model [22]. As shown in Fig. 4, the plant inputs $U = [v_{ca}, v_{cq}]$ and outputs $Y_R = [\dot{i}_{dR}, \dot{i}_{qR}]$ at time $k$, $k - 1$, and $k - 2$ are fed into the RBFN to estimate the plant output $\dot{Y} = [\dot{i}_{d}, \dot{i}_{q}]$ at time $k + 1$. The RBFN is trained to provide a dynamic predictive plant model at all times; this model is then used for training the RBFNC.

The RBFN is pretrained offline using a suitably selected training data set collected from two sets of training [14]. The first set is called forced training, in which the two PI controllers ($P_{id}$ and $P_{iq}$) are deactivated, and the small pseudorandom binary signals (PRBS), given by

$$\text{PRBS}_{v_{ca}}(k) = 0.1 \cdot |v_{ca}|$$

$$\text{PRBS}_{v_{cq}}(k) = 0.1 \cdot |v_{cq}|$$

are imposed from an external source and added to the steady-state plant inputs $v_{ca}$ and $v_{cq}$ as the forced disturbances of the plant as well as the RBFN at each time step $k$, as shown in Fig. 2. In (8) and (9), rand2, rand3, and rand5 are the uniformly distributed random numbers in $[-1, 1]$ with frequencies 2, 3, and 5 Hz, respectively; $v_{ca}$ and $v_{cq}$ are the magnitudes of $v_{ca}$ and $v_{cq}$, respectively. The second set is called natural training, in which the PRBS is removed and the system is exposed to the natural disturbances and faults in the power network. The forced training and the natural training are carried out at several different operating points to form the training data set, given by

$$\Delta = \{X, Y\} = \left\{ \bigcup_{i=1}^{m} \Delta_{F,i}, \bigcup_{i=1}^{m} \Delta_{N_{ij}} \right\}$$

where $\Delta$ is the entire training data set selected from $m$ operating points; $X$ and $Y$ are the input data sets of the RBFN and the corresponding output data sets of the plant, respectively; $\Delta_{F,i}$ is the subset collected from the forced training at the operating point $i$; and $\Delta_{N_{ij}}$ is the subset collected from the natural training caused by the $j$th natural disturbance event at the operating point $i$. The selected training data set ensures that the RBFN can track the system dynamics over a wide operating range.

The performance of the RBFNNs relies on a set of parameters, including the number of RBF units, the RBF centers, widths, and output weights. Given the number of RBF units, the locations of the RBF centers are determined by a $k$-means clustering algorithm [23] using the data from the training data set $\Delta$. After locating the RBF centers, a good method to determine the RBF widths is the $p$-nearest neighbors heuristic [24], in which the width $\beta_k$ of the $k$th RBF unit is given by

$$\beta_k = \left( \frac{1}{p} \sum_{i=1}^{p} ||C_i - C_k||^2 \right)^{1/2}$$

where $C_i$ are the $p$-nearest neighbors to the center $C_k$. In this paper, $p$ is chosen as the same as the number of RBF units $h$ in the hidden layer. After determining the RBF centers and widths, the output weights of the RBFNN are then calculated by the singular value decomposition (SVD) method [22]. However, the widths given by (11) are still nonoptimal and can be optimized to achieve an optimal RBFNN with fewer RBF units and better performance [25]. Following the method in [25], the RBF widths are optimized by the PSO algorithm for a given number of RBF units. By training the RBFN with optimized widths over the training data set $\Delta$, the performances of the RBFN using different number of RBF units are compared. It is found that any further increase of the RBF units over 25 gives negligible further improvement of the RBFNN performance. Therefore, the dimensions of the input, hidden, and output layers of the RBFNN are $12 \times 25 \times 6$.

3) Design of the RBFNC: The RBFNC is used to replace two conventional PI controllers in Fig. 2. As shown in Fig. 4, the RBFNC uses the plant outputs $Y_R = [\dot{i}_{dR}, \dot{i}_{qR}]$ at time $k - 1$, $k - 2$, and $k - 3$ as its inputs, and then generates the control signals $U(k) = [v_{ca}(k), v_{cq}(k)]$ as the plant inputs to drive the plant outputs to the desired values.

The RBFNC is first pretrained offline to learn the dynamics of the CONVC. This ensures that the whole system, consisting of the FTIANC and the plant, remains stable. Similar to the pretraining of the RBFN, the pretrained data set for the RBFNC is collected from two sets of training, forced training, and natural training. During the forced training, the following forced PRBS:

$$\text{PRBS}_{P_{in,ref}}(k) = 0.05 \cdot |P_{in,ref}|$$

is applied to the plant model. The second set is called natural training, in which the PRBS is removed and the system is exposed to the natural disturbances and faults in the power network.
is added to the reference value of the turbine input power to disturb the plant (with switch S2 closed in Fig. 1). In (12), $P_{in,ref}$ is the reference value of the turbine input power; $r_{ad12}$, $r_{ad13}$, and $r_{ad15}$ are the same as those in (8) and (9). The forced and natural trainings are carried out at several different operating points to form the pretraining data set, given by

$$B = \{ S, Z \} = \left\{ \bigcup_{i=1}^{m} B_{Fi}, \bigcup_{i=1}^{m} \bigcup_{j=1}^{n} B_{Nij} \right\} \quad (13)$$

where $B$ is the entire pretraining data set selected from $m$ operating points; $S$ is the output data set of the plant and also the input data set of the RBFNC; $Z$ is the input data set of the plant and also the output data set of the CONVC; $B_{Fi}$ is the subset collected from the forced training at the operating point $i$; and $B_{Nij}$ is the subset collected from the natural training caused by the $j$th natural disturbance event at the operating point $i$. The selected pretraining data set ensures that the RBFNC can track the CONVC dynamics over a wide operating range. During this stage, the plant is controlled by the CONVC instead of the RBFNC (with switch S1 in position 1 in Fig. 1) because the RBFNC has not yet learned the correct control behavior. Following the approach for the RBFNI, the parameters of the RBFNC, including the number of RBF units, the RBF centers, widths, and output weights, are determined offline using the pretraining data set $B$. Consequently, the dimensions of the input, hidden, and output layers of the RBFNC are $6 \times 15 \times 6$.

Once the RBFNC has learned the dynamics of the CONVC, the RBFNC is applied to control the plant (with switch S1 in position 2 in Fig. 1) and the output weights of the RBFNC are adapted further online to achieve better performance. Online training of the RBFNC takes place with the trained RBFNI in cascade with the reference model, as shown in Fig. 4. The reference model utilizes the reference inputs $P^*$ to generate the desired plant outputs $Y^*$ at each time step, which are used to guide the plant outputs $Y = [i_d, \delta]$ to a desired steady-state set point. In this paper, $P^* = [P^*, Q^*]$ are used as the reference inputs; thereby, $Y^*$ are calculated to be the constant values $[i^*_d, \delta^*]$ at each time step. The error signal $E_d(k + 1) = (1/2)[|E_d(k + 1)|^2]$, where $E_d(k + 1)$ is the difference between the desired outputs $Y^*$ of the reference model and the estimated output $\hat{Y}$ of the RBFNC at time $k + 1$, is propagated back through the RBFNI (without changing its weights) to form the error vector $E_C(k)$. It is then used to train the RBFNC before the next sampling instant. During this stage, the output weights of the RBFNC can also be adapted online to the unexpected operating conditions where it has not been pretrained offline [4]. The online training is carried out at different operating points by applying the forced and the natural trainings as well. When a desired performance is achieved, the training stops and the RBFNC with fixed parameters is then used to control the plant.

### C. Design of the SERS

1) **Overall Structure of the SERS**

The SERS consists of three cascaded MSR blocks with different priorities as shown in Fig. 5. Since the vectors $V$ and $I_r$ are irrelevant to the performances of the RBFNI and the RBFNC, the three MSR blocks are only used to evaluate and restore the three crucial sensor measurements in the vector $J$. However, the sensor data in $V$ and $I_r$ provide the necessary data redundancy among the inputs of the SERS. Therefore, a necessary condition for the SERS to work is that $v_{ca}, v_{db}, v_{ec}, i_{ca}, i_{db},$ and $i_{ec}$ are all available. This condition is determined by the following relationships. Because the transmission systems of a power network normally operate under a nearly balanced three-phase condition, the three-phase voltages $v_{ca}, v_{db},$ and $v_{ec},$ and the three-phase currents, $i_{ca}, i_{db},$ and $i_{ec}$ should approximately satisfy the following:

$$v_{ca} + v_{db} + v_{ec} = 0 \quad (14)$$

$$i_{ca} + i_{db} + i_{ec} = 0. \quad (15)$$

The real systems are not ideally balanced. A realistic expression for (14) and (15) can be written as follows, which are usually true at the transmission level where an SSSC would be connected:

$$|v_{ca} + v_{db} + v_{ec}| < \sigma_1 \quad (16)$$

$$|i_{ca} + i_{db} + i_{ec}| < \sigma_2 \quad (17)$$

where $\sigma_1$ and $\sigma_2$ are the predetermined small threshold values.

If the system is under normal balanced operating conditions, but (16) or (17) are not satisfied, it indicates that one or more sensors in $V$ or $I_r$ are missing. However, if $v_{ca}, v_{db}$ and $v_{ec}$ are all missing, there might be $v_{ca} = v_{db} = v_{ec} = 0$, and therefore, (16) is satisfied. To distinguish this case from the case of no missing sensor, another equation is used, given by

$$|v_{ca}| < \sigma_1 \quad \text{and} \quad |v_{db}| < \sigma_1 \quad \text{and} \quad |v_{ec}| < \sigma_1. \quad (18)$$

Based on (16) and (18), there exist three scenarios of the sensor data in the vector $V$: if (16) is satisfied but (18) is not satisfied, there is no sensor missing; if (16) is not satisfied, there are one or more sensors missing; if (16) and (18) are both satisfied, $v_{ca}, v_{db},$ and $v_{ec}$ are all missing. A similar equation can be used for the vector $I_r$, given by

$$|i_{ca}| < \sigma_2 \quad \text{and} \quad |i_{db}| < \sigma_2 \quad \text{and} \quad |i_{ec}| < \sigma_2. \quad (19)$$

However, the transmission systems of a power network may experience unbalanced operations, although they rarely happen. During unbalanced operations, both (16) and (17) may not be satisfied even though there is no sensor missing. The SERS should not identify these cases as sensor missing. Therefore, it is necessary for the SERS to distinguish the unbalanced operations from the balanced operations. This issue will be discussed in Section IV-C4.

Equations (16)–(19) are implemented by the module “equations” (16)–(19) as shown in Fig. 5) to evaluate the integrity of the sensor data in $V$ and $I_r$. If any sensor data in $V$ and $I_r$ are missing, they send a signal to block the three MSR modules. Otherwise, they send unblock signal to the three MSR modules, which are then activated for sensor evaluation and missing sensor restoration.

Each MSR has the same structure as shown in Fig. 3 and only performs a 1-D search to restore one missing sensor measurement. Therefore, the input current vector $I_1$ of MSR1 only consists of one current measurement (i.e., $I_1 = i_a \text{ or } i_b \text{ or } i_c$). If
$I_1$ is missing, it is restored by MSR1 and the restored value $I_{1R}$ is then used as the healthy input for MSR2. Consequently, the input current vector $I_2$ of MSR2 consists of two current measurements: one is the same as $I_1$, and the other is any one of the two currents not being used by MSR1. Finally, the input current vector $I_3$ of MSR3 consists of all of the three current measurements $i_a$, $i_b$, and $i_c$. In addition, the input vector of each MSR also contains the voltage vector $V$, which provides the required redundancy for missing sensor restoration. In this design, the three current vectors at the inputs of three MSR blocks are $I_1 = [i_a, i_b, i_c]$, and $I_3 = [i_a, i_b, i_c]$, respectively. The variables $I_{1R}$, $I_{2R}$, and $I_{3R}$ represent the restored sensor data from MSR1, MSR2, and MSR3, respectively. If several MSR blocks end up restoring the same missing sensor data, the finally restored value is chosen from the MSR with the highest priority. The use of the cascading structure to design the SERS is based on the following reasoning:

1) This structure enables the SERS itself to evaluate the status of the crucial sensor measurements and to detect which sensor or sensors are missing, instead of relying on a sensor evaluation scheme in [9] or a sensor monitor in [10].

2) Each MSR only searches in a 1-D space to restore one missing sensor measurement, which is faster than only using one MSR [9], [10] to search in a multidimensional space in order to restore multiple missing sensor measurements. In this application, each MSR converges within 20 iterations to restore one missing sensor measurement.

3) The required degree of data redundancy for restoring one missing sensor is lower than that of restoring multiple missing sensors for each MSR, and therefore, fewer sensor data need to be used.

The performance of the MSR relies on the data dependency at its input. Higher dependency among input data means better performance of the MSR. During balanced operation, the three-phase currents $i_a$, $i_b$, and $i_c$ approximately satisfy the following:

$$i_a + i_b + i_c = 0$$

Therefore, a strong dependency is present among the three current variables $i_a$, $i_b$, and $i_c$, and this relationship is used for designing the MSR3 block.

It is necessary to use three additional variables $I_r = [i_{ra}, i_{rb}, i_{rc}]$ as the inputs of MSR1 in order to provide enough redundancy among the input data of MSR1 (Other additional variables instead of $I_r$ can be used). Assuming that $i_a$ is missing, Fig. 6 shows the actual value $i_a$, the restored values $i_{aR1}$ by MSR1 without $I_r$ as inputs ($i_{aR1}(4)$) and with $I_r$ as inputs ($i_{aR1}(7)$), as well as the corresponding estimation errors $e_{a1}(4)$ and $e_{a1}(7)$ ($e_{a1} = ||i_a - i_{aR1}||$), respectively. These results clearly show that the performance of MSR1 is improved significantly by using three additional variables $I_r = [i_{ra}, i_{rb}, i_{rc}]$.

Fig. 7 compares the performances of the three MSR blocks. Assuming that $i_a$ is missing and MSR1 uses $I_r$ as inputs, Fig. 7 shows the actual value $i_a$, the restored values $i_{aR1}$ from MSR1, $i_{aR2}$ from MSR2, and $i_{aR3}$ from MSR3, as well as the corresponding estimation errors $e_{a1}$, $e_{a2}$, and $e_{a3}$ ($e_{a1} = ||i_a - i_{aR1}||$, $e_{a2} = ||i_a - i_{aR2}||$, and $e_{a3} = ||i_a - i_{aR3}||$), respectively. In this test, the PSO in all three MSR blocks uses the same fixed number of iterations to search for the estimated value of the missing sensor $i_a$ at each time step. These results clearly indicate that MSR3 has the best performance, and the performance of MSR2 degrades a little compared to MSR3 but it is better than MSR1. Moreover, MSR2 and MSR3 use less sensor data than MSR1. Therefore, the priority of restoring the same missing sensor decreases in the following order: MSR3, MSR2, and MSR1.
To determine the number of hidden-layer neurons of the autoencoder in each MSR, three requirements should be considered. First, a necessary condition for the autoencoder to work is that the number of healthy inputs must equal or exceed the number of degrees of freedom in the hidden layer. Second, the number of hidden-layer neurons should be as few as possible to reduce the real-time computational cost. Finally, to reproduce the autoencoder’s inputs at its output layer, the number of hidden-layer neurons must be sufficient. Based on these requirements and the discussions in the previous paragraphs, the dimensions of the input, hidden, and output layers of the autoencoders in MSR1, MSR2, and MSR3 are chosen to be $21 \times 12 \times 21$, $15 \times 10 \times 15$, and $18 \times 12 \times 18$, respectively. The output vector of the SERS $I_R$ contains the total restored missing sensor measurements from all three MSR blocks; however, $I_R$ 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 $i_{dR}$ and $i_{qR}$ by applying the SRRFT.

2) Training of the Autoencoders: The training of the autoencoder in each of the three MSR blocks requires that all sensor data in $V, I_T$, and $f$ are available. The integrity of the sensor data in $V$ and $I_T$ is evaluated by the module “equations (16)–(19)” as shown in Fig. 5, while the integrity of the sensor data in the crucial vector $I$ can be pre-evaluated by two equations similar to (16)–(19), given by

$$|i_\alpha + i_\beta + i_\gamma| < \sigma_3$$

(21)

$$|i_\alpha| < \sigma_3 \text{ and } |i_\beta| < \sigma_3 \text{ and } |i_\gamma| < \sigma_3.$$  

(22)

If (21) is satisfied but (22) is not satisfied, there is no sensor in $I$ missing; otherwise, some sensors in $I$ are missing. The three autoencoders are continuously trained online simultaneously without any missing sensor. By feeding forward the data through the autoencoder and adjusting its weight matrices (using backpropagation algorithm) $W$ and $V$, the autoencoder is trained to map its inputs to its outputs as shown in Fig. 3(a). After training for every $N_T$ time steps, the weights of each autoencoder are frozen for $N_E$ time steps to evaluate the convergence of the autoencoder. During the evaluation, if the error $||E_S||$ of each autoencoder is beyond a specified threshold value $\mu$ at any evaluation time step, the training resumes for next $N_T$ time steps. Otherwise, if the error $||E_S||$ of each autoencoder is below the threshold value $\mu$ during the entire $N_E$ time steps, the training stops and the autoencoder is used for sensor evaluation and missing sensor restoration. If the system changes to a new operating point, the error $||E_S||$ might be beyond the threshold value $\mu$ again. In this case, if there is no sensor missing [determined by (16)–(19), (21), and (22)], the training resumes to adapt to this new operating point.

3) Sensor Evaluation and Missing Sensor Restoration: The entire sensor evaluation and missing sensor restoration procedure of the SERS is implemented in four stages as shown in Fig. 8. In the first three stages, the SERS evaluates the status of the three current measurements $i_\alpha, i_\beta$, and $i_\gamma$ by checking the value of the error signal $||E_S||$ of each autoencoder as shown in Fig. 3. During a normal operating condition, with a well-trained autoencoder, $||E_S||$ should be acceptably small. (In real applications, a threshold value can be specified depending on the system properties.) If one or more sensors are missing, the outputs of the autoencoder no longer match its inputs and the value of $||E_S||$ becomes significant.

The sensor evaluation process is illustrated in Table I, in which the positive sign “+” indicates that the value of $||E_S||$ of the corresponding MSR is significant, while the negative sign “−” indicates that the value of $||E_S||$ of the corresponding MSR is below a prespecified threshold value $\varepsilon$. Table I gives all eight cases of the status of $i_\alpha, i_\beta$, and $i_\gamma$, which can be determined in only three stages as follows. Stage I indicates that there is no restoration action from any MSR and all the MSR blocks are only used to check the value of $||E_S||$. Stage II indicates that MSR1 is activated to restore the missing current $i_\alpha$, and all the MSR blocks are also used to check the value of $||E_S||$. Stage III indicates that MSR2 is activated to restore the missing current $i_\beta$ or $i_\gamma$, and all the MSR blocks are also used to check the value of $||E_S||$. In each stage, the restored missing data is used as the estimated healthy data required by the next stage.

The eight cases can be separated into four groups. The first group contains cases 0 and 1, which can be determined directly in stage I. The second group contains cases 2 and 3, which can be determined by stages I and III. In stage I, both cases indicate that $i_\alpha$ is not missing but that $i_\gamma$ is missing, therefore moving to stage III. In stage III, $i_\gamma$ is restored by MSR2, therefore cases 2 and 3 can be distinguished by the sign of MSR3. The third group contains cases 4 and 5, which is determined by stages I and II. The fourth group contains cases 6 and 7, which must be determined by all three stages. In stage I, all of the four cases in groups 3 and 4 indicate the same results, i.e., $i_\alpha$ is missing, therefore going to stage II. In stage II, $i_\alpha$ is restored by MSR1. Group 3 is distinguished from group 4 because $i_\gamma$ is missing, which is indicated by the sign of MSR2. Thereafter, cases 4 and 5 in group 3 are separated directly by checking the sign of MSR3. A positive sign of MSR2 in stage II indicates that the cases fall into group 4. Therefore, the checking procedure goes to stage III, in which MSR2 restores the missing current $i_\gamma$. Finally, cases 6 and 7 are separated by the sign of MSR3.

If the SERS detects that one or more current sensors are missing, the procedure goes to the last stage IV, in which MSR3
is activated to restore one missing sensor. Other missing sensors (if they exist) take the values that are restored in the previous three stages. Table II shows the restored missing sensor by each MSR during stages II–IV in each case. It is important to note that in any stage, each MSR only performs a 1-D search to restore one missing sensor. In addition, one missing sensor may be restored by more than one MSR during the four stages, e.g., \( i_q \) in case 4 or \( i_c \) in cases 2 and 6. In order to select only one restored value for each missing sensor, the three MSR blocks are set with different priorities, as explained in Section IV-C1. Because MSR3 has the highest priority, it is always activated to restore one missing sensor in the last stage IV. If the same missing sensor is restored by more than one MSR, the finally restored value comes from the MSR with the highest priority. In Table II, the variables in the blank spaces with \( X \) represent the restored sensor values which are not used.

4) Unbalanced Operations: The balanced and the unbalanced operations can be distinguished by using (16), (17), and (21). If (16), (17), and (21) are all not satisfied, the system is under unbalanced operating conditions, and vice versa. However, if only one or two of (16), (17), and (21) are not satisfied, the system is operated at a balanced operating condition and some sensors are missing. Here, a reasonable assumption is that all the three vectors \( V, I_p, \) and \( I \) simultaneously containing missing sensors never happen.

Depending on the duration, the types of unbalanced operations can be divided into two main categories: long-term or steady-state unbalanced operations and short-term unbalanced operations [26]. The long-term unbalanced operations are mainly caused by unbalanced load, transformer with different single-phase units, untransposed transmission lines, etc. The effects of long-term unbalanced operations are normally small.
TABLE I

<table>
<thead>
<tr>
<th>Group No.</th>
<th>Case No.</th>
<th>Missing Sensors</th>
<th>Stage I: No action</th>
<th>Stage II: MSR1 action</th>
<th>Stage III: MSR2 action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>none</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$i_a$</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>$i_c$</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>$i_{a,b,c}$</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>$i_{a,b,c}$</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>Case No.</th>
<th>Missing Sensors</th>
<th>Restored Sensors</th>
<th>MSR1</th>
<th>MSR2</th>
<th>MSR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$i_a$</td>
<td>$i_a$</td>
<td>+</td>
<td>bst</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$i_c$</td>
<td>$i_c$</td>
<td>+</td>
<td>bst</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$i_{a,b,c}$</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>bst</td>
<td>bst</td>
</tr>
<tr>
<td>4</td>
<td>$i_{a,b,c}$</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>bst</td>
<td>bst</td>
</tr>
<tr>
<td>5</td>
<td>$i_{a,b,c}$</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>bst</td>
<td>bst</td>
</tr>
<tr>
<td>6</td>
<td>$i_{a,b,c}$</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>bst</td>
<td>bst</td>
</tr>
<tr>
<td>7</td>
<td>$i_{a,b,c}$</td>
<td>$i_{a,b,c}$</td>
<td>+</td>
<td>bst</td>
<td>bst</td>
</tr>
</tbody>
</table>

because the transmission system is still close to balanced operation [26].

The short-term unbalanced operations are mainly caused by unbalanced grid faults [26], including the single-phase-to-ground fault, phase-to-phase fault, etc. Under these conditions, the transmission systems experience a short-term strongly unbalanced operation (e.g., typically, 50–200 ms) during the fault, and return to the balanced three-phase operation after the fault is cleared. If some sensor data are missing while an unbalanced fault occurs, (16)–(19), (21), and (22) will not be applicable to evaluate the status of the sensor data during the short-term unbalanced fault. Therefore, the following scenarios must be considered separately: 1) some sensor data are missing during normal operations and no unbalanced fault occurs; 2) some sensor data are missing from a moment during normal operations and thereafter an unbalanced fault occurs; 3) some sensor data are missing from a moment during an unbalanced fault; and 4) some sensor data are missing after the unbalanced fault is cleared and no further faults occur. Because the SERS is designed for balanced operating conditions, there is no problem of the SERS’s operation for scenarios 1) and 4).

However, for scenario 2), the block signals from the “equations (16)–(19)” module as shown in Fig. 5 must be ignored during the short-term fault condition such that the SERS can continue to restore the missing sensors. The fault conditions can be detected by utilizing the signals from the protection devices that have been installed for the transmission lines. For scenario 3), the three MSR modules are blocked during the fault and unblocked after the fault is cleared. Because the fault only exists for a very short time, scenario 3) rarely happens. Even if it happens, shortly blocking the MSR modules does not have any notable effect on the entire system performance.

D. Example of the SERS Block—$i_a$ and $i_c$ Missing

Fig. 9 shows an example of the SERS block in the case of $i_a$ and $i_c$ missing; $V = [v_{ca}, v_{cb}, v_{cc}]$ and $I_c = [i_{ca}, i_{cb}, i_{cc}]$. The dashed line indicates that the restored value $i_{c,R}$ from MSR2 is discarded.
E. Discussions

The training and the operation of the SERS are independent of the RBFNI and the RBFNC. However, when some sensors are missing, the performances of the RBFNI and the RBFNC are dependent on the SERS. Because of the convergence of each MSR, a suitably designed and trained SERS can quickly and accurately restore the missing sensors data, and therefore, provides a set of high-performance fault-tolerant measurements to the RBFNI and the RBFNC. Consequently, the RBFNI provides a fault-tolerant robust identification for the input–output dynamics of the nonlinear plant, and the RBFNC provides a fault-tolerant robust control for the SSSC. On the other hand, the restored sensor data are not exactly the same as the healthy sensor data. As a result, the performance of the RBFNI and the RBFNC might degrade slightly. The dynamic performance of the entire FTIANC will be evaluated by simulation studies on a SMIB as well as a multimachine power system in Sections V and VI.

V. Simulation Results on the SMIB Power System

The dynamic performance of the proposed FTIANC is evaluated at two different operating points by applying three-phase and single-phase short circuits to the system in Fig. 1. During the simulations at each operating point, two and all of the three current sensors are assumed to be missing, respectively. In a practical system, if some sensors are missing, their values may be read as zeros, some noises or some uncertain error values. However, the forms of missing sensor readings do not affect the implementation of the SERS. Therefore, during the simulation, the sensor readings are simply set as zeros if they are missing.

A. Test on a Three-Phase Fault 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 prefault rotor angle of 42.6°, output active power $P_t \approx 1.0$ p.u., and output reactive power $Q_t = 0.56$ p.u.; the transmitted active power and reactive power at the receiving end of line 3 are regulated by the SSSC at 0.45 p.u. and 0.22 p.u., respectively. At this operating point, about half of the generator’s output active power is transmitted by line 3. 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 tripped off from the system. After this grid fault, the system changes to a new operating condition with only two lines (lines 1 and 3) in service. Two missing sensor tests are then applied during this postfault transient state.

Case I—$i_b$ and $i_c$ Missing: From $t = 15.1$ s, the current sensors $i_b$ and $i_c$ are assumed to be missing and restored by the SERS. The restored values $i_bR$ and $i_cR$ are used with the healthy current $i_a$, together, to calculate $i_dR$ and $i_qR$ by applying the SRRFT.

Case II—$i_a$, $i_b$, and $i_c$ Missing: In this extreme case, for the same initial conditions as in Case I, all three current sensors $i_a$, $i_b$, and $i_c$ are assumed to be missing from 15.1 s.
QIAO et al.: FAULT-TOLERANT INDIRECT ADAPTIVE NEUROCONTROL FOR A SSSC IN A POWER NETWORK

Fig. 13. The 100-ms three-phase short circuit at 15 s at OP-I for Case II—\(i_a\), \(i_b\), and \(i_c\) missing from 15.1 s: \(\delta\) and \(V_s\).

Fig. 14. The 100-ms three-phase short circuit at 15 s at OP-II for Case I—\(i_b\) and \(i_c\) missing from 15.1 s.

\(i_b\) and \(i_c\) are assumed to be missing and restored by the SERS from \(t = 15.1\) s onwards. The restored values \(i_aRT\), \(i_bRT\), and \(i_cRT\) are used to calculate \(\deltaRT\) and \(\deltaRT\) by applying the SRRFT.

In each of the two cases, the two current components \(i_aRT\) and \(i_cRT\) are used by the RBFNI and the RBFNC as the actual plant outputs for online training, identification, and control. Figs. 10 and 12 show the actual values \(i_a\) and \(i_c\), the restored values \(i_aRT\) and \(i_cRT\) by the SERS, and the estimated values \(\hat{i}_a\) and \(\hat{i}_c\) by the RBFNI for Cases I and II, respectively. These results indicate that with a suitably designed SERS, the missing sensors are correctly restored, and therefore, provide a set of correct estimates \([\hat{i}_{aRT}, \hat{i}_{cRT}]\) of the plant outputs and a set of complete inputs to the RBFNI. As a consequence, the RBFNI tracks the transient dynamics of the actual plant outputs \(i_a\) and \(i_c\) with good precision.

Figs. 11 and 13 show the results of the rotor angle \(\delta\) and the 230 kV sending-end bus voltage \(V_s\) for Cases I and II, respectively. The curves CONVC indicate the system response under the condition that the SSSC is controlled by the CONVC. These results clearly show that the damping control of the FTIANC is more efficient than the CONVC during the postfault transient state. During the first swing after the fault is applied, the FTIANC is already providing significant damping compared to that provided by the CONVC. After the fault is cleared, the FTIANC drives the plant successfully and quickly to a new operating point with a rotor angle \(\delta = 46.3^\circ\) at the steady state. Moreover, comparing the curves by FTIANC with and without missing sensors, the control performance of the FTIANC only degrades slightly due to missing sensor data. However, the transient performance of the FTIANC with missing sensor measurements is still better than the CONVC used by the SSSC without any missing sensor. These results prove that the proposed FTIANC provides a fault-tolerant control for the SSSC.

B. Test on a Three-Phase Fault at a Different Operating Point

The transient performance of the FTIANC is now reevaluated at a different operating point (OP-II), where the prefault rotor angle of the generator changes to 50.1° (\(P_L = 1.0\) p.u. and \(Q_L = 0.59\) p.u.); \(P^*\) and \(Q^*\) are still chosen to be 0.45 p.u. and 0.22 p.u., respectively; however, line 1 is now kept 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 condition. A 100-ms three-phase short circuit is applied to the receiving end of line 2 at $t = 15$ s. Again, the same two missing sensor tests as in Section V-A are applied during this postfault transient state. Case I—from $t = 15.1$ s, the current sensors $i_a$, $i_b$, and $i_c$ are assumed to be missing and restored by the SERS. Case II—for the same initial conditions as in Case I, all three current sensors $i_a$, $i_b$, and $i_c$ are assumed to be missing and restored by the SERS from $t = 15.1$ s onwards.

Figs. 14 and 16 show the actual values $i_d$ and $i_q$, the restored values $i_{dR}$ and $i_{qR}$ by the SERS, and the estimated values $i_d$ and $i_q$ by the RBFNI for Cases I and II, respectively. Again, the plant outputs are correctly estimated by using the restored missing currents from the SERS as well as other healthy currents (if they exist). Consequently, the RBFNI tracks the transient dynamics of the plant with good precision.

Figs. 15 and 17 show the results of the rotor angle $\delta$ for Cases I and II, respectively. These results indicate that the CONVC fails to return the system back to the steady state after this large disturbance. However, the FTIANC still provides effective control, regardless if there are sensors missing or not. These results prove that the proposed FTIANC provides improved transient performance over the CONVC, and a fault-tolerant control for the SSSC over a wide system operating range.

C. Tests on a Single-Phase Fault at OP-II

In power system transient studies, three-phase short circuits are commonly used to evaluate the system transient performance and stability because they are the most severe faults in the power grid. However, in the real power system, most grid faults are unbalanced single-phase-to-ground faults. To further illustrate the robustness of the FTIANC, the system is now tested with a phase-A-to-ground fault at OP-II. This unbalanced fault is applied to the receiving end of line 2 at $t = 15$ s and is cleared after 150 ms. The system experiences an unbalanced operation during the fault, and returns to the balanced three-phase operation after the fault is cleared. Two missing sensor tests are applied from $t = 14$ s before the fault: Case I—two current sensors $i_b$ and $i_c$, missing, and Case II—all three current sensors $i_a$, $i_b$, and $i_c$ missing. Figs. 18 and 19 show the results of the generator rotor angle $\delta$ and the 230-kV sending-end bus voltage $V_s$ for Cases I and II, respectively. This single-phase fault causes a larger fault current in phase A than in phases B and C. The SERS has not been trained on this unbalanced condition, but it still provides the fault-tolerant measurements to the SSSC controllers. As a result, the control performance of the FTIANC both with and without missing sensors is superior to the CONVC without any missing sensor.

The proposed FTIANC is also tested with other types of unbalanced faults, including the phase-to-phase fault and two-phase-to-ground fault. The similar results as those in Figs. 18 and 19 are obtained. The FTIANC provides the effective control to the SSSC during various unbalanced faults even when two or three current sensors are missing. The simulation results in this section show that the FTIANC is able to provide a fault-tolerant robust control to the SSSC at any operating conditions in the transmission system, including the steady-state operation, balanced and unbalanced grid faults, and change of operating conditions.
VI. SIMULATION RESULTS ON THE IEEE 10-MACHINE 39-BUS SYSTEM

To demonstrate further the effectiveness of the proposed FTIANC, the IEEE 10-machine 39-bus New England system [27] as shown in Fig. 20 is now considered. An SSSC is connected to the bus 24 end of the transmission line 23–24 to regulate its power flows. This arrangement also improves the transient stability of this multimachine power system [28]. The same control schemes CONVC and FTIANC are applied to control the SSSC. In this study, G10 is modeled as a three-phase infinite source, while the other nine synchronous generators (G1–G9) are modeled in detail, with the turbine governor and AVR/exciter dynamics taken into account. The SSSC is represented by a detailed switching-level model.

In this application, the crucial sensor measurements are the three-phase currents $i_a$, $i_b$, and $i_c$ of line 22–24. The other two sets of measurements, which are the injected three-phase voltages of the SSSC $v_{ca}$, $v_{cb}$, and $v_{cc}$ and the three-phase voltages of bus 24 $v_{24a}$, $v_{24b}$, and $v_{24c}$, are irrelevant to the performance of the SSSC controllers. They are used by the SERS to provide the required data redundancy in order to restore the crucial missing sensor measurements.

A three-phase short circuit is now applied to the bus 22 end of line 21–22 at $t = 50$ s, and is cleared after 150 ms. The same missing sensor tests as for the SMIB system (Cases I and II) are now applied, in which the sensor data are assumed to be missing from $t = 50.15$ s. Figs. 21 and 22 show the angular speeds of the generators 6 and 7, $\omega_6$ and $\omega_7$, for Cases I and II, respectively. Because the proposed SERS and FTIANC are independent of the system models, these results are similar to those in the SMIB system. The dynamic performance of the FTIANC is superior to the CONVC used by the SSSC. By using the fault-tolerant measurements from the SERS, the FTIANC provides effective control to the SSSC even when two or all crucial current sensor measurements are missing. Moreover, the control performance of the FTIANC with missing sensors is almost the same as that without any missing sensors. Therefore, the proposed FTIANC improves the transient performance, stability, and reliability of the entire power system. Due to the space constraints, the results at the other operating conditions are not presented because they are similar to those in Figs. 21 and 22, and the SMIB system.

VII. CONCLUSION

Control of nonlinear systems depends on the availability and quality of sensor measurements. Measurements are inevitably
subject to faults that can be caused by sensor failure, broken or bad connections, bad communication, or malfunction of some hardware or software. Therefore, fault tolerance is an essential requirement for system control. This improves system reliability, maintainability, and survivability.

This paper has proposed an FTIANC for an SSSC FACTS device. This FTIANC consists of a suitably designed SERS, an RBFNI, and an RBFNC. It is able to provide effective control to the SSSC when single and multiple crucial time-varying sensor measurements are missing.

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 actual plant outputs to the desired values. The SERS employs the autoassociative neural networks as autoencoders to capture the correlations between the redundant time-varying sensor measurements (through training). This enables the SERS to evaluate the integrity of the crucial time-varying current sensor measurements which determine the behaviors of the RBFNI and the RBFNC. If the SERS detects that one or more crucial sensor measurements are missing, it utilizes the particle swarm optimization algorithm and the correlations established by the autoencoders between the missing sensor data and the remaining healthy sensor data to quickly online restore the missing sensors. The restored missing sensor data are then combined with the healthy sensor data to provide a set of complete inputs for the RBFNI and the RBFNC. This guarantees a fault-tolerant control for the SSSC.

Simulation studies are carried out for the SSSC connected to an SMIB power system as well as the IEEE 10-machine 39-bus power system. The proposed FTIANC with and without missing sensor measurements is compared to a conventional PI control scheme used by the SSSC without any missing sensor measurements. Results show that the FTIANC improves the dynamic performance and reliability of the SSSC and the power network over a wide range of system operating conditions. The proposed fault-tolerant control is readily applicable to other plant models in power systems.

REFERENCES


Wei Qiao (S’05) received the B.Eng. and M.Eng. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 1997 and 2002, respectively, and the M.S. degree in high-performance computation for engineered systems from Singapore–MIT Alliance (SMA), Singapore, in 2003. Currently, he is working towards the Ph.D. degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

From 1997 to 1999, he was an Electrical Engineer at the China Petroleum & Chemical Corporation (SINOPEC). His current research interests include power system modeling, dynamics, control and stability, wind energy generation and grid integration, FACTS devices, and the application of computational intelligence in power systems. He is the coauthor of over 30 papers in refereed journals and international conferences.

Mr. Qiao was the recipient of the first prize in the student paper and poster competition of the IEEE Power Engineering Society General Meeting, Montreal, QC, Canada, in June 2006.


In 1971, he was appointed to the Chair of Electrical Machines and Power Systems at the University of Natal, Durban, South Africa, where he has been a Professor of Electrical Engineering, the Department Head, and the Deputy Dean of Engineering. Currently, he is the Duke Power Company Distinguished Professor at the Georgia Institute of Technology, Atlanta. His current research interests include the dynamic behavior and condition monitoring of electric machines, motor drives, power systems and their components, and controlling them by the use of power electronics and intelligent control algorithms. He is the coauthor of about 400 papers in refereed journals and international conferences and three patents. Altogether, ten of his papers attracted prizes from journals and conferences.

Prof. Harley is a Fellow of the British Institute of Electrical Engineers. He is also a Fellow of the Royal Society in South Africa, and a Founder Member of the Academy of Science in South Africa formed in 1994. During 2000 and 2001, he was one of the IEEE Industry Applications Society’s six Distinguished Lecturers. He was the Vice-President of Operations of the IEEE Power Electronics Society (2003–2004) and Chair of the Atlanta Chapter of the IEEE Power Engineering Society. He is currently Chair of the Distinguished Lecturers and Regional Speakers program of the IEEE Industry Applications Society. He received the Cyril Veinot Award in 2005 from the Power Engineering Society for “outstanding contributions to the field of electromechanical energy conversion.”

Ganesh Kumar Venayagamoorthy (S’91–M’97–SM’02) was born in Jaffna, Sri Lanka. He received the B.Eng. degree (with honors) from Abubakar Tafawa Balewa University, Nigeria, in March 1994 and the M.S. and Ph.D. degrees in engineering from the University of Kwazulu-Natal (UKZN), South Africa, in April 1999 and April 2002, respectively.

Currently, he is an Associate Professor of Electrical and Computer Engineering and the Director of the Real-Time Power and Intelligent Systems Laboratory at Missouri University of Science and Technology, Rolla. He was a Visiting Researcher at the ABB, Corporate Research, Sweden, in 2007. His research interests are in the development and applications of computational intelligence for real world applications including power systems stability and control, FACTS devices, alternative sources of energy, sensor networks, and evolvable hardware. He has published two edited books, three book chapters, 50 refereed journals papers, and 190 refereed international conference proceeding papers.

Dr. Venayagamoorthy is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT. He is a Senior Member of the South African Institute of Electrical Engineers (SAIEEE). He is also a member of the International Neural Network Society (INNS), The Institution of Engineering and Technology, U.K., and the American Society for Engineering Education. Currently, he is the IEEE St. Louis Computational Intelligence Society (CIS) and IAS Chapter Chairs, the Chair of the Working Group on Intelligent Control Systems, the Secretary of the Intelligent Systems subcommittee and the Vice-Chair of the Student Meeting Activities subcommittee of the IEEE Power Engineering Society, and the Chair of the IEEE CIS Task Force on Power System Applications. He has organized and chaired several panels, invited and regular sessions, and tutorials at international conferences and workshops. He was a recipient of the 2007 ONR Young Investigator Program Award, the 2004 NSF CAREER Award, the 2006 IEEE Power Engineering Society Walter Fee Outstanding Young Engineer Award, the 2006 IEEE St. Louis Section Outstanding Section Member Award, the 2005 IEEE Industry Applications Society (IAS) Outstanding Young Member Award, the 2006 SAIEEE Young Achievers Award, the 2004 IEEE St. Louis Section Outstanding Young Engineer Award, the 2003 INNS Young Investigator Award, five prize papers from the IEEE IAS and IEEE CIS, a 2006 UMR School of Engineering Teaching Excellence Award, and a 2005 UMR Faculty Excellence Award. He is listed in the 2008 edition of Who’s Who in the World.