An Introduction to the Echo State Network and its Applications in Power System

Jing Dai[1], Ganesh K. Venayagamoorthy[2] and Ronald G. Harley[1]

[1] School of Electrical and Computer Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332, USA
E-mail: jingdai@gatech.edu; rharley@ece.gatech.edu

[2] Real-Time Power and Intelligent Systems Laboratory,
Department of Electrical and Computer Engineering,
Missouri University of Science and Technology, Rolla, MO 65409, USA
E-mail: gkumar@ieee.org

Abstract—**Echo State Network (ESN) is a new type of Recurrent Neural Network (RNN) proposed in recent years. The training process of ESN is easier and requires less computational effort than regular RNN which has the same size. Due to its high modeling capability of complex dynamic system, ESN has been used in various power system applications such as power system nonlinear load modeling and true harmonic current detection, wide area monitoring, intelligent control of an Active Power Filter (APF), overhead conductor thermal dynamics identification, wind speed or water inflow forecasting, etc. This paper introduces the basic concept and the offline and online training algorithms of the ESN in detail and reviews the state of the art of ESN applications in power systems.**

I. Introduction

Artificial Neural Networks (ANNs) are well known as effective universal function approximators. Different types of ANNs have been proposed in the past few decades and broadly used in nonlinear system modeling, intelligent neural control systems. The Multi-Layer Perceptron Network (MLP) and Radius Basis Function Network (RBF) have relatively simpler structures and training algorithms. However, their approximation abilities are limited due to their feedforward structure, especially for the modeling of complex dynamic systems. Recurrent neural networks (RNNs) have advantages over feedforward neural networks in terms of modeling dynamic systems with their dynamic memory and time embedding capabilities; however, the application of RNNs has been limited due to their high computational requirement during training. Recently, the Echo State Network (ESN) [1] has been introduced as a novel approach for designing RNNs. Since it was developed from RNNs, it has the modeling capability of a recurrent network but requires simpler training process. ESN has been successfully applied for simple function approximation, system identification and direct adaptive control.

This paper introduces the basic concept and the training algorithms of ESNs, and reviews their applications in power systems. In section II, the basic concept of ESNs and their online and offline training algorithms are introduced. The various applications of ESNs in power systems, in both modeling and control, are then reviewed in Section III.

II. Echo State Network

Echo state network is a special type of recurrent neural networks. In order to maintain the high modeling capability of ordinary recurrent neural networks, a large (e.g. 100 hidden neurons) RNN is used as a "Dynamic Reservoir (DR)" in the hidden layer of the ESN, which can be excited by suitably presented input and/or feedback of output. The architecture of the ESN is shown in Fig. 1 [1]. Due to the complex structure of the DR, the ESNs have high capability for modeling complex dynamic systems. However, in order to limit the required computational effort during training, a novel type of both online and offline training algorithms has been proposed, which are summarized in the following sections.

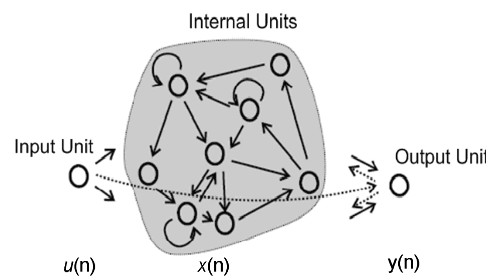


Fig. 1. ESN architecture.

A. Offline Training Algorithm of the ESN

The steps of the offline training algorithm for the ESN [1] are summarized below:

1) Generate a recurrent neural network following certain rules to ensure its "echo state property"

Three weight matrices should be generated. They are the input weight matrix W^{in}, internal weight matrix W and the output feedback matrix W^{back}. Once W^{in}, W and W^{back} have been generated, they will not change during the entire training process.

The echo state property is related to the algebraic properties of the weight matrix W; however, there is no known necessary and sufficient algebraic condition which allows one to decide whether the network has the echo state property, given W^{in}, W and W^{back}. However, there are certain conditions which increase the possibility of the RNN having the echo state property. Usually W is generated by following the principles described below:

- Generate a sparse matrix W_0 and make sure the mean value of all the weights in it is about zero.
- Normalize W_0 to a matrix W_1 with unit spectral radius as:

$$W_1 = \frac{W_0}{|\lambda_{\max}|}, \tag{1}$$

where $\lambda_{\max}$ is the spectral radius of W_0.

- Scale W_1 to W, as

$$W = \alpha W_1, \tag{2}$$

978-1-4244-5098-5/09/$26.00 ©2009 IEEE

where $\alpha < 1$.

2) Feed the teacher input and teacher output data (training data) to the ESN

When the training data is fed to the ESN, it will activate the dynamics within the dynamic reservoir. At each sampling step, compute the internal dynamic reservoir states according to equation (3):

$$x(n+1) = \tanh(u(n+1)W^{\text{in}} + x(n)W + y(n)W^{\text{back}}) \quad (3)$$

where u is the input vector, x is the vector of internal units and y is the output vector.

Then, collect the states at time n as a new row in a matrix *DR*.

Since at sampling step $n = 0$, $x(0)$ and $y(0)$ are not defined, the following initial conditions are used:

- Initialize the network state arbitrarily, e.g. to the zero state $x(0)=0$;
- Set $y(0)=0$.

3) Wash out the initial memory in the DR

Since the arbitrarily generated network states (e.g. zero state) contains an initial memory which is not caused by the input, do not collect information from times $n = 1, \cdots, n_0$. (n_0 is different according to different systems and the length of input sequence.)

By time $n = n_0 + 1$, it is safe to assume that the effects of the arbitrary starting state have die out and that the network states are a pure reflection of the teacher-forced input and output.

So the first n_0 rows of the states matrix *DR* are removed to obtain a new matrix DR^{forget}.

4) Compute the output weights

The first n_0 rows of the teacher output sequence $y(n)$ are also removed to obtain a new matrix $Teacher^{\text{forget}}$.

Then the output weight is calculated using equation (4):

$$W^{\text{out}} = (Pseudoinverse(DR^{\text{forget}}) \bullet Teacher^{\text{forget}})^T \quad (4)$$

B. Online Training Algorithm of the ESN

The online training algorithm of the ESN proposed in [2] is described as follows:

1) Generate a recurrent neural network

Different from the offline training algorithm, there are **four** initial weight matrices that should be generated. Besides the input weights W^{in}, internal weights W and the feedback weights W^{back}, the output weights W^{out} is also randomly generated at the beginning.

The rules for generating the internal weights W are the same as those described in the offline training algorithm.

2) Calculate the states in the DR

This step is the same as described in the offline training algorithm. Equation (3) is used for the calculation of the states in the dynamic reservoir. However, it is no longer required to collect the state at time n as a new row of a state matrix DR.

3) Compute the estimated output of the ESN

Now the output $\hat{y}(n)$ of the ESN is calculated by equation (5):

$$y(n+1) = x(n+1)W^{\text{out}}(n) \quad (5)$$

where $W^{\text{out}}(n)$ denotes the output weight matrix at time step n.

4) Update the output weights

The estimated output $\hat{y}(n+1)$ is compared with the actual output $y(n+1)$ collected from the RTDS and the error vector e_y is calculated as:

$$e_y = y(n+1) - \hat{y}(n+1) \quad (6)$$

The output weights are updated according to equation (7), as described in [2]:

$$W^{\text{out}}(n+1) = W^{\text{out}}(n) + \eta x(n+1)^{\text{T}} e_y(n+1) + \gamma x(n)^{\text{T}} e_y(n) \quad (7)$$

where η is the learning gain and γ is the momentum gain, and each one is in the range of [0, 1], just like the parameters used in other types of neural network training schemes.

The output weights are updated such that the mean squared training error (MSE) is minimized.

$$MSE = \frac{1}{r}\sum_{n=1}^{r}(y(n) - \hat{y}(n))^2 = \frac{1}{r}\sum_{n=1}^{r}(y(n) - \sum_{i=1}^{L} W_i^{out} \cdot x(n))^2 \quad (8)$$

where r is the length of the Input/Output sequence used for training and $x(n)$ contains the states within the dynamic reservoir.

C. Summary of Training Algorithm of the ESN

In both online and offline training algorithms of ESN, only the output weights are updated, which guarantees the low computational requirement of ESN. However, since all the other weights are randomly generated, the overall performance of ESNs is highly dependent on the randomly generated weights. Therefore, ESNs are normally pre-selected in both offline and online applications, so that the randomly generated weights can provide satisfying performances.

III. ESN Applications in Power System

The ESN has been used for many power system applications. These applications can be divided into two main categories: (1) Train an ESN as a nonlinear function approximator to identify certain features of a dynamic system. (2) Apply neurocontrol techniques such as Adaptive Critic Design [xx] approaches and use ESNs for control purposes. This section shows the state of the art of ESN applications in power system by five examples.

A. Nonlinear Load Modeling using ESNs

With the wide use of power electronic devices, harmonic currents are being injected into the power system, known as “harmonic pollution”. Although IEEE Standard 519 requires the utilities and customers to limit the amount of harmonic current and voltage, the practical evaluation is complicated, as it is difficult to separate the contributions from the utilities and customers. A neural network-based harmonic current prediction scheme was previously proposed by the authors to estimate the true harmonic current attributed to the nonlinearity of the load, instead of the distorted power supply. ESN is used for the detection of true harmonic contributions from nonlinear loads [3].

The neural network-based harmonic detection scheme is shown in Fig. 2.

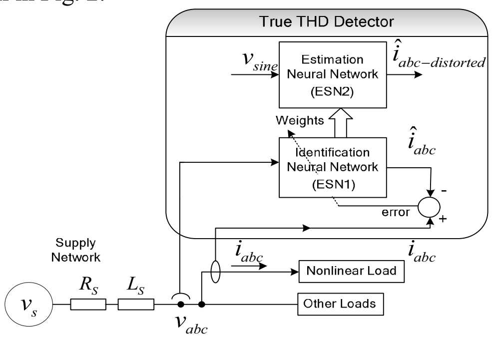


Fig.2. Neural network-based harmonic detection scheme.

The distorted power supply voltage v_{abc} and the resultant distorted line current iabc are measured to identify nonlinear characteristic of the load. The identification neural network (ESN1) is trained to learn the nonlinear characteristics. The estimation neural network (ESN2), which is an exact replica of ESN1 with trained and fixed weights, is used to predict the true current harmonics.

The identification neural network (ESN1) is trained with the distorted voltage v_{abc} as input and the distorted load current i_{abc} as output. After training, a mathematically generated sine wave is supplied to the estimation neural network ESN2, to predict the load current harmonics when the load is supplied by a clean sinusoidal voltage source. Ideally, ESN1 can learn the exact nonlinear characteristics of the load, and therefore the distortion present in the output of ESN2, $\hat{i}_{abc-distorted}$ can be considered as the exact current harmonics attributed to the nonlinearity of the load.

After training, the weights of ESN1 is fixed and tested. The testing result of the ESN is shown in Fig. 3. The output of ESN matches the actual current waveforms of the nonlinear load very well, which means the ESN has "leant" the nonlinear characteristics of the load.

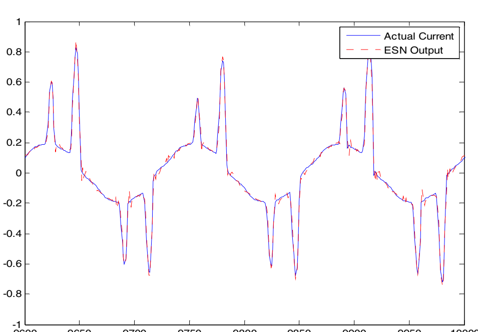


Fig.3. ESN training results in load modeling application

A comparison of the MLP, RNN and ESN for this application is also carried out in [3] and the results show that the ESN has a stronger learning ability since fewer training data is needed by the ESN. Moreover, the ESN has a simpler training process and less computational effort since only the output weight needs to be updated.

Offline training algorithm is used for this application. The training data (voltage and current) is first measured from the nonlinear load and stored. Then the offline training is carried out using the saved data.

B. *Wide Area Monitoring using ESN*

With deregulation and growth of the power industry, many power system elements such as generators, transmission lines, are driven to operate near their maximum capacity, especially those serving heavy load centers. Wide Area Controllers (WACs) using wide area or global signals can provide remote auxiliary control signals to local controllers such as automatic voltage regulators, power system stabilizers, etc. to damp out system oscillations. However, since the power system is highly nonlinear with fast changing dynamics, it is a challenging problem to design an online system monitor/estimator, which can provide dynamic intra-area and inter-area information such speed deviations of generators to an adaptive WAC continuously.

An ESN is used for the online design of a Wide Area Monitor (WAM) for a multimachine power system in [2]. A single ESN is used to predict the speed deviations of four generators in two different areas. The performance of this ESN WAM is evaluated for small and large disturbances on the power system.

The structure of the multimachine power system with a WAM is shown in Fig. 4. The two-area system consists of two fully symmetrical areas linked together by two transmission lines. Each area is equipped with two identical synchronous generators rated 20 kV/900 MVA. All the generators are equipped with identical speed governors and turbines. The loads are represented as constant impedances and split between the areas in such a way that Area 1 is transferring about 413 MW to Area 2.

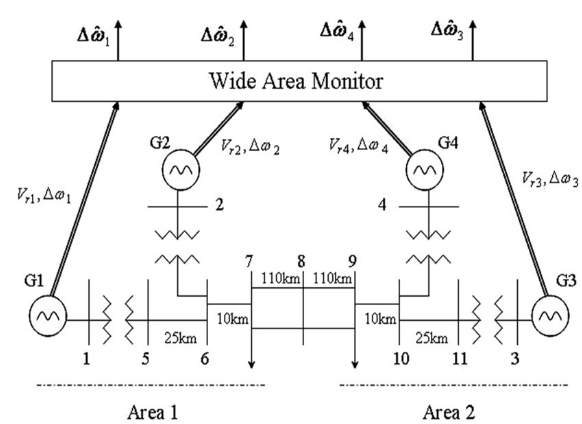


Fig.4. Two-area power system with a WAM predicting the speed deviations of generators G1, G2, G3 and G4.

The training of the ESN WAM is shown in Fig.5.

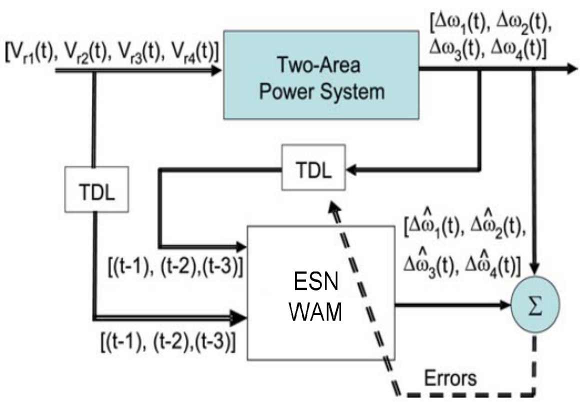
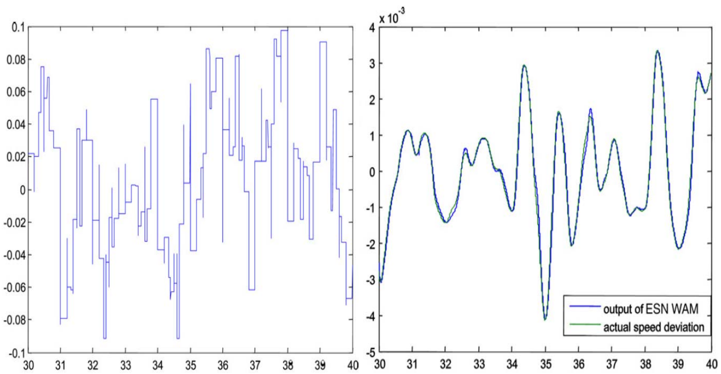


Fig.5. Wide area monitor for a two-area power system (TDL stands for timedelayed lines, in this case (t − 1), (t − 2) and (t − 3)).

Pseudorandom binary signal (PRBS) ΔV_{r1}, ΔV_{r2}, ΔV_{r3} and ΔV_{r4} are applied to Generator G1, G2, G3 and G4 respectively to train the ESN WAM. Figure 6-(b) shows the actual speed deviations and ESN WAM estimated speed deviations of the generators G1 with the ESN output weights fixed after 30 seconds of continuously training (3000 input and output samples). It is clear from this figures that the ESN WAM is able to estimate the speed deviations of G1 sufficiently well indicating that it has learned the power system dynamics.



(a) Voltage deviation applied to Generator G1

(b) Actual and ESN WAM estimated speed deviation of G1

Fig.6. Forced training input and testing results of generator G1.

Online training is carried out in this application so that the ESN WAM can be used to monitor the power system dynamics in real time.

C. *Overhead Conductor Thermal Dynamics Identification using ESN*

Dramatic reductions in sensor, computing and communications costs, coupled with significant performance enhancements has increased the possibility of realizing widely and massively distributed Power Line Sensor Networks (PLSNs) to monitor utility asset status for enhancing line reliability and utilization. One of the important applications of such a PLSN is to evaluate the overhead power line dynamic current capacity down to 'per span' level of granularity. Due to the inherent non-linearity of overhead conductor thermal behavior, it is usually quite complex to directly calculate the conductor temperature. Therefore the prediction for the conductor dynamic thermal behavior becomes difficult. The ESN is proposed to applied to predict the overhead conductor thermal dynamics in real-time in [4]. The well trained ESN model is used to predict the dynamic thermal behavior, and thus to evaluate the dynamic current capacity of the line under current ambient weather conditions.

The thermal dynamic model of the overhead conductors is shown in Fig.7.

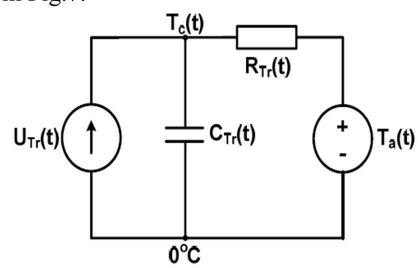


Fig.7. Thermal dynamic model of the overhead conductor.

The temperature of the overhead conductor $T_c(t)$ can be considered as a nonlinear function of the Ambient Temperature $T_a(t)$ and the Line Current $I(t)$, i.e. $T_c(t) = f(T_a(t), I(t))$ .An ESN is trained to learn this nonlinear function using data measured from the overhead conductors in a real power system. The training algorithm of this ESN thermal dynamic identifier is shown in Fig.8.

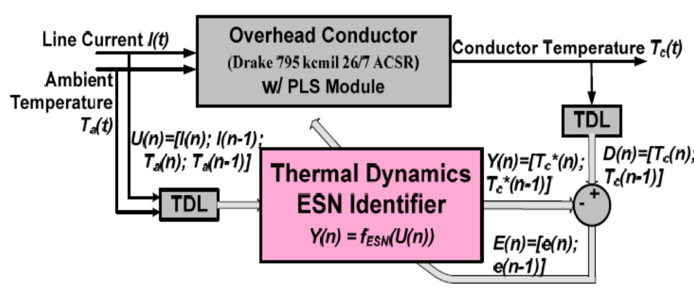
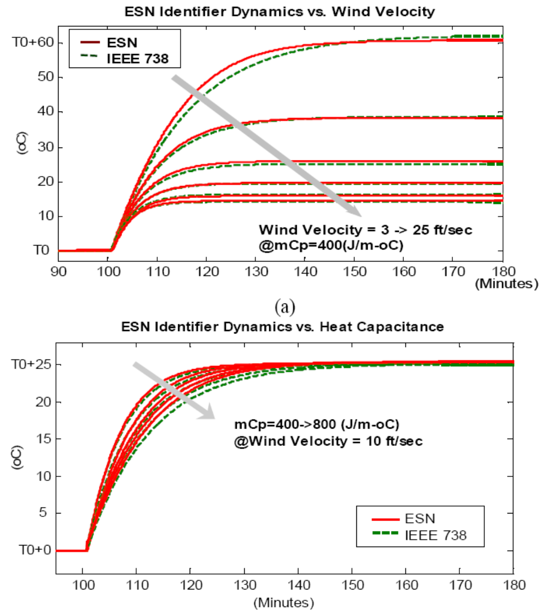


Fig.8. Overhead Conductor Thermal Dynamics Identifier.

Figure 9 shows the prediction results of the ESN thermal dynamic identifier under different weather conditions and heat capacitance.

This proposed method removes the need for a thermal model, as well as the need to measure all the various ambient weather conditions, which helps to reduce the monitoring costs dramatically. This will benefit the massive deployment of the proposed power line sensor modules along power lines down to a “per-span” level of granularity.



(b)

Fig.9. ESN step response under variant weather conditions: (a) Wind velocity; (b) Heat capacitance.

D. *Wind Speed Forecast and Water Inflow Forecasting using ESNs*

In [5], the ESN is used to forecast hydropower plant reservoir water inflow, which is fundamental information to the hydrothermal power system operation planning. A database of average monthly water inflows of Furnas plant, one of the Brazilian hydropower plants, is used as source of training and test data. The results show that the ESN provides pretty good results for one-step ahead water inflow forecasting, providing valuable information for the system operator.

In [6], the ESN is used to forecast hourly wind speed in North Brazil in a period of 24 hours ahead, trying to provide this fundamental information for the operation planning of the electrical wind power system.

These two applications are both examples of using the ESN for modeling. The offline training algorithm is used in both applications.

E. *Adaptive Controller Design using ESNs*

Applications A, B, C, and D described above in Section II are all examples of using ESN for modeling purposes. The ESN can also be used for control purpose just like other types of neural networks.

In [7], an ESN-based harmonic current identification scheme for indirect adaptive neurocontrol of an Active Power Filter (APF) in a multiple-reference frame is proposed. APF has been used to compensate harmonics and reactive current drawn by nonlinear loads. The performance of the APF is highly dependent on its control strategy. A typically power system is a large-scale, nonlinear and non-stationary system and APF controllers are traditionally designed from a linearized system model at a specific operating point. However, when the operating condition changes or large disturbance occurs, the controller performance degrades. The drawbacks of using linear controllers to control a nonlinear system can be overcome by using neural-network-based nonlinear intelligent control techniques. Since ESN has been used as an effective system identifier with much faster training speed than other Recurrent Neural Networks (RNNs), it is suitable for such real-time control applications.

The proposed online system identifier using an ESN is implemented on an Innovative Integration M67 card consisting of the TMS320C6701 processor to identify the load harmonics in a typical power system. The APF and the power system are simulated using a Real-Time Digital Simulator (RTDS) system. The testing results of a real-time implementation show that the ESN is capable of providing fast and accurate system identification for the indirect adaptive neurocontrol of an APF.

The active filter application is shown in Fig. 10. The current injected by the active filter can compensate for the harmonic current drawn by the nonlinear load, leaving the source current nearly sinusoidal.

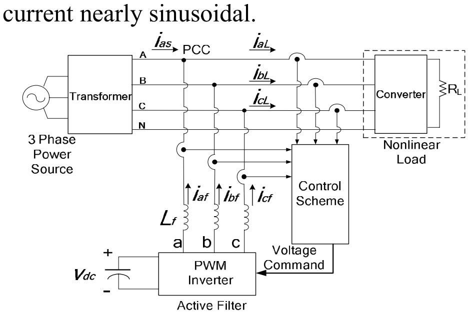
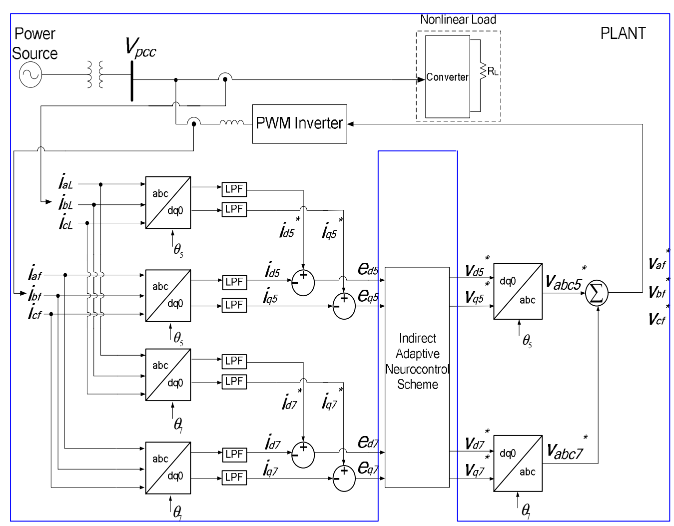
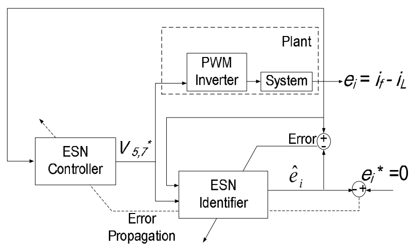


Fig.10. An APF connected to a typical power system.

The overall scheme for the adaptive neurocontrol of the APF in multiple-reference frames is shown in Fig. 11.



(a)



(b)

Fig. 11. ESN-based indirect adaptive control scheme.

An AC-DC power electronic converter supplying an adjustable resistance is used as a nonlinear load, which injects harmonic components into the load currents i_{aL}, i_{bL}, i_{cL}. The 5th and 7th harmonics in the load current, which are the major current harmonics present, are extracted using multiple-reference frames, in this case one reference frame for the 5th harmonic and another for the 7th harmonic. Each reference frame contains an *abc*-to-*dq* transform but uses an appropriate transformation angle which rotates at a specific multiple of the fundamental frequency. For example, the 5th harmonic reference frame (HRF) converts the 5th harmonic components in i_{aL}, i_{bL}, i_{cL} to dc current components i_{d5}^*, i_{q5}^* (HRF). A low pass filter then extracts these dc currents by eliminating all the higher frequency components. The i_{af}, i_{bf}, i_{cf} currents are also transformed into the 5th HRF, and their *d*, *q* components i_{d5}, i_{q5} are each compared with the i_{d5}^*, i_{q5}^* respectively, to form the two errors e_{d5}, e_{q5}. The 7th harmonic currents are processed in a similar way using a 7th HRF. The e_{d5}, e_{q5}, e_{d7}, e_{q7} are used by the neurocontrol scheme and fed to the neurocontroller to control the voltage command of the PWM inverter. When the errors are eliminated by the neurocontroller, the APF injects the correct 5th and 7th current harmonics to cancel the current harmonics caused by the nonlinear load, and hence no 5th and 7th current harmonics are injected into the power network.

The connections and the flow of the data between the DSP and the RTDS are shown in Fig. 12-(a). Figure 12-(b) shows the online training algorithm of the ESN identifier. At each time step, the ESN identifier predicts the outputs, which is the error vector between the harmonics of the load currents and those of the current injected by the shunt active filter, with the voltages transformed by the 5th and 7th reference frames as the inputs. Then, the errors between the actual values and these predicted values are calculated and used to update the weights of the ESN identifier. As the entire implementation is under real-time operation, the DSP and the RTDS can emulate the actual performance of the ESN identifier in practical applications.

The online training results of an ESN identifier with 50 internal units is shown in Fig. 13. The four outputs estimated by the ESN identifier can accurately track the desired values obtained from the RTDS simulation.

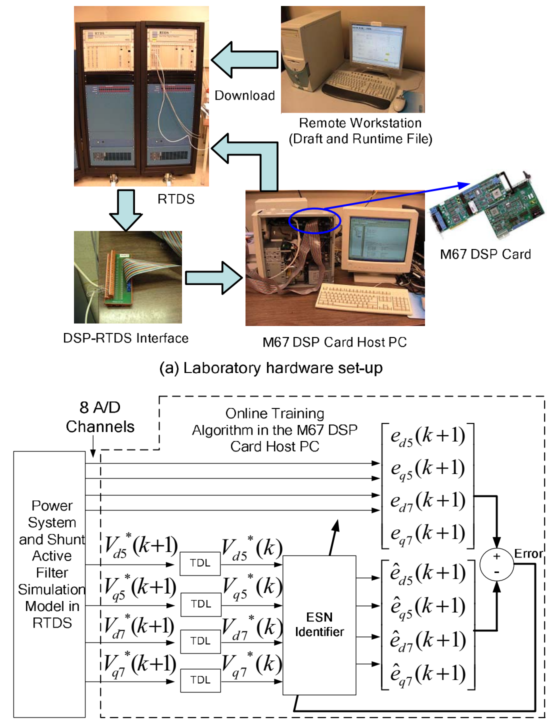


Fig.12. Laboratory hardware setup for the online training of the ESN identifier.

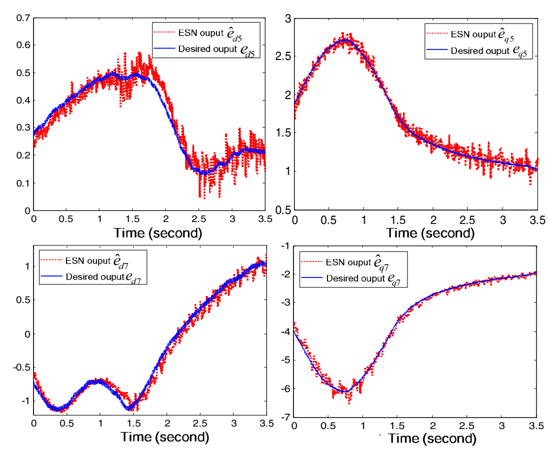


Fig. 13. Online training results of the ESN identifier with 50 internal units.

IV. Conclusions

This paper has introduced the basic concepts of ESN and their offline and online training algorithms. Several typical applications of ESN in power system have been reviewed. These applications have shown that ESN is an effective tool for nonlinear system modeling and control. The introduction of ESN and the review of its applications in power systems can be used as valuable references for future researches.

Acknowledgment

This work is supported by the National Science Foundation (NSF), USA, under grant 0836017 and grant ECS 0601521.

REFERENCES

[1] H. Jaeger, "Tutorial on Training Recurrent Neural Networks, Covering BPTT, RTRL, EKF and the 'Echo State Network' approach," GMD Report 159. German National Research Center for Information Technology, Sankt Augustin, 2002.

[2] G. K. Venayagamoorthy, "Online design of an echo state network based wide area monitor for a multimachine power system," *Neural Networks*, vol. 20, pp. 404-413, 2007

[3] Jing Dai, P. Zhang, J. Mazumdar, R. G. Harley and G.K. Venayagamoorthy, "A Comparison of MLP, RNN and ESN in Determining Harmonic Contributions from Nonlinear Loads," *Proc. IEEE 34th Annual Conf. on Industrial Electronics(IECON'08)*, Orlando, 2008, pp.3025 –3032.

[4] Yi Yang, R .G. Harley, D.M. Divan and T. G. Habetler "Overhead Conductor Thermal Dynamics Identification by using Echo State Networks," *Proc. IEEE Int. Joint Conf. on Neural Networks*, *IJCNN'09* Atlanta, GA, USA, July 2009.

[5] Rodrigo Sacchi, Mustafa C. Ozturk, Jose C. Principe, Adriano A. F. M. Carneiro and Ivan N. da Silva, "Water Inflow Forecasting using the Echo State Network: a Brazilian Case Study," *Proc. IEEE Int. Joint Conf. on Neural Networks*, *IJCNN'07* Orlando, Florida, USA, August 12-17, 2007

[6] A. A. Ferreira, T. B. Ludermir, R.R.B de Aquino, M. M. S. Lira and O.N. Neto, "Investigating the use of Reservoir Computing for forecasting the hourly wind speed in short –term," *Proc. IEEE Int. Joint Conf. on Neural Networks*, *IJCNN'08* Hong Kong, China, June 1-8, 2008

[7] Jing Dai, G.K. Venayagamoorthy and R. G. Harley, "Harmonic Identification using an Echo State Network for Adaptive Control of an Active Filter in an Electric Ship," *Proc. IEEE Int. Joint Conf. on Neural Networks*, *IJCNN'09* Atlanta, GA, USA, July 2009.