Comparison of TDNN and RNN Performances for Neuro-Identification on Small to Medium-Sized Power Systems

Diogenes Molina, Jiaqi Liang, Ron Harley
Department of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, USA
ddmolina@gatech.edu, jliang@gatech.edu,
rharley@ece.gatech.edu

Ganesh K. Venayagamoorthy
Real-Time Power and Intelligent System Laboratory
Missouri University of Science and Technology
Rolla, MO 65409, USA
gkumar@ieee.org

***Abstract*— For Artificial Neural Networks (ANN) to become more widely used in power systems and the future smart grids, ANN based algorithms must be capable of scaling up as they try to identify and control larger and larger parts of a power system. This paper goes through the process of scaling up an ANN based identifier as it is driven to identify increasingly larger portions of a power system. Distributed and centralized approaches for scaling up are taken and the pros and cons of each are presented. The New England/New York 68-bus power network is used as the test bed for the studies. It is shown that while a fully-connected (centralized) ANNs is capable of identification of the system with appropriate accuracy, the increase in the training times required to obtain an acceptable set of weights becomes prohibitive as the system size is increased.**

Keywords-power system identification; artificial neural networks; time delay neural network; recurrent neural network

I. Introduction

Artificial Neural Networks (ANNs) have been used for identification and control of a variety of dynamical systems. In the power and energy area they have been applied to problems such as load and wind speed forecasting [1]-[2], controller coordination [3], eigen-value extraction [4], wide area control [5], power system protection [6], power system identification[7], etc. However, for ANNs to become more widely used in power systems and provide smart solutions to the future smart grids, it is important to evaluate different approaches, architectures, and training algorithms that have the potential to lead to successful neuro-identification and neuro-control of realistically sized power systems.

A power system is a dynamical system composed of thousands of elements that span over vast geographical regions, are affected by a variety of events and disturbances, and with widely varying operating regimes. Identifying and controlling such a system will require ANN based algorithms that are capable of: achieving acceptable performance under different operating conditions with new smart grid ingredients (generalizing), as larger and larger parts of the power system are considered (scaling up).

Several schemes have been proposed in the literature for improving ANN generalization capabilities. Bayesian regularization [8] and early stopping [9] are examples of such schemes.

Scalable ANN architectures such as Cellular Neural Networks (CNN) are described in [10] for power system terminal voltage predictions with encouraging results. In [11] the dimension of the input to the ANN is reduced by means of sub-setting and feature space transformation methods, leading to reductions of the size of the ANN required for identifying a power system, therefore improving scalability. Others have developed hardware and software platforms specifically aimed at handling large size ANN [12]-[13]. Depending on the application, the development of such specialized implementations might prove to be unfeasible. This paper goes through the process of scaling up ANNs to identify increasingly larger portions of a power system using two approaches, distributed and centralized.

The paper is organized as follows: section II provides some general concepts. Section III compares two ANN architectures commonly used for non-linear system identification using a single machine infinite bus power system. The comparisons allow a more informed selection of the architecture to use for scaling up. Section IV goes through the process of scaling up using a distributed and a centralized approach. The 68-bus New England/New York power system is used for the study. Conclusions are provided in Section V.

II. Background

A. General ANN Concepts

Several ANN architectures have been developed for dynamical system identification. Two commonly used topologies are the Time-Delay Neural Network (TDNN) and the Recurrent Neural Network (RNN). Topological characteristics give these networks the capability to process temporal information. Also, internal non-linear elements as well as sets of trainable weights provide these networks with the capability to replicate non-linear input-output mappings. These two features coupled with the availability of effective parameter adaptation algorithms (training algorithms) make the TDNN and the RNN good candidates for non-linear dynamic system identification [14].

This work was supported by the National Science Foundation under EFRI Grant #0836017

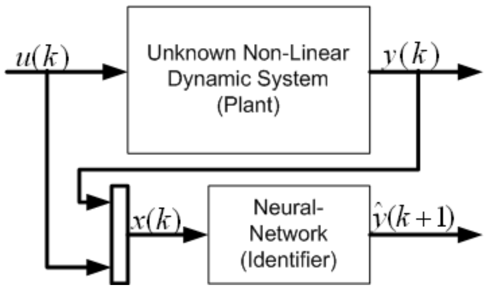

Figure 1. ANN identification of unknown dynamic non-linear system

The identification task for the ANN illustrated in Fig. 1 can be summarized as: given a sequence of present inputs and outputs of an unknown plant, the identifiers are expected to provide one-step-ahead predictions of the plant outputs.

1) Time Delay Neural Network (TDNN)

The TDNN is a feed-forward network consisting of three layers: input, hidden, and output layers. Fig. 2 illustrates a TDNN. Note that *d* time delays are applied to the input vector *x(k)* and presented as inputs to the network. These time-delayed inputs provide the network with temporal information about the system being identified [15]. The hidden layer activation function is a design parameter, which in this study was selected to be the tan-sigmoid function. Other commonly used options are radial basis functions, log-sigmoid, etc. The matrices W_x, W_y, and the bias vectors b_1, b_2 are the variable parameters that are updated following a particular training algorithm in order to reproduce or mimic the input-output mapping of the plant. The output of the TDNN is given by (1).

$$\hat{y}(k+1) = f\left(W_x \times [x(k) \quad x(k-1) \quad \cdots \quad x(k-d)]^T + b_1\right) \times W_y + b_2 \tag{1}$$

2) Recurrent Neural Network (RNN)

The RNN architecture selected for this study has the structure of the Elman network as illustrated in Fig. 3. Note the presence of a time-delayed feedback loop between the output of the hidden layer and its input. This loop provides the network with memory, thus greatly improving its capability to represent dynamical systems [16]. The output of the RNN is given by (2).

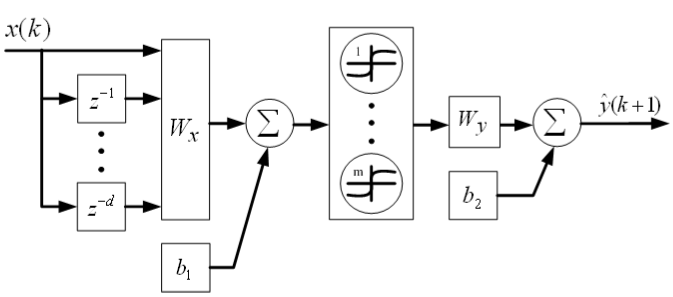

Figure 2. Time delay neural network architecture

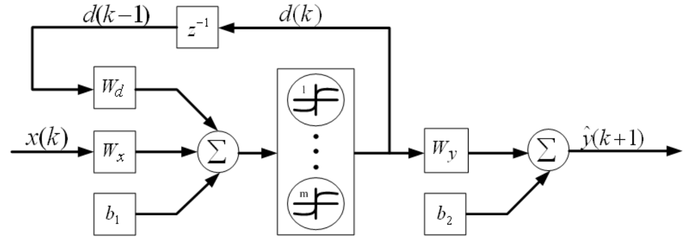

Figure 3. Recurrent neural network architecture

$$\hat{y}(k+1) = f\left(W_x \times x(k) + W_d \times d(k-1) + b_1\right) \times W_y + b_2 \tag{2}$$

One disadvantage of using an ANN with feedback loops is a great increase of the complexity and computational cost of training the network [15]. In both, the TDNN and the RNN, the process of training involves calculating the gradient of the error with respect to the parameters in the network; however, the recurrent structure causes changes in the RNN parameters to have an impact for all time steps after they occur. This prompts the utilization of computationally expensive methods such as Real-Time Recurrent Learning (RTRL), Back Propagation Trough Time (BPTT), or truncated BPTT to calculate the necessary error gradient [16].

3) Training Algorithms:

Several training algorithms available for use in MATLAB's Neural Networks Toolbox are investigated during the studies. Algorithms such as the Levemberg-Marquardt, Bayesian Regularization, etc, prove to be too computationally intensive for training larger networks. After a heuristic search, the scaled conjugate gradient algorithm presented in [17] was selected to train larger ANNs. Similar findings are provided in [18], where a more detailed study was completed of the performance of different training algorithms when applied to large feed-forward ANNs.

B. Performance Metrics

In the literature, no single performance for ANN evaluation has surfaced as the agreed upon universal standard. Therefore, three popular performance metrics are proposed here to evaluate the neural networks: the Root Mean Square Error (3), the Mean Absolute Error (4), and the Mean Absolute Percentage Error (5) since each of them emphasizes a different aspect of the identification results. A similar approach for comparing ANN performance was taken in [19]. Due to the squaring action, the RMSE emphasizes the presence of large errors. The MAE weighs all measurements equally, while the MAPE measures accuracy by dividing the predictions by the measured values of the plant output.

$$RMSE_j = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_{j,t}\right)^2} \tag{3}$$

$$MAE_j = \frac{1}{T}\sum_{t=1}^{T}\left|y_t - \hat{y}_{j,t}\right| \tag{4}$$

$$MAPE_j = \frac{100}{T}\sum_{t=1}^{T}\left|\frac{y_t - \hat{y}_{j,t}}{y_t}\right| \tag{5}$$

C. Training Methodology

ANN weight updates are limited to offline training, which is completed in the following sequence:

- For each generator, Pseudo-Random Binary Signals (PRBS) are injected at two points: the terminal voltage set point to the automatic voltage regulator, and the active power output set point to the governor. This procedure is expected to excite the dynamics of the power system.
- Input-output pairs are collected from the simulation as training data.
- Training data is imported into the Neural Network Toolbox in MATLAB where it is used by the **Scaled Conjugate Gradient** training algorithm to generate an acceptable set of network weights.
- Weights are stored in a text file for use during the testing stage.

D. Testing Methodology

The ANN parameters are held constant at all times during testing. Two rounds of evaluations are completed. The first round focuses on observing the generalization capabilities of the networks by measuring the performance of predictions at different operating points of a Single Machine Infinite Bus power network. The second round evaluates the scaling properties of the ANN. Training stage behavior, and ANN testing performance are observed as the number of parameters, inputs, and outputs are increased to identify larger and larger portions of the New England/ New York 68-bus test system.

The power system simulations are developed in DIgSILENT [21]. During testing, the different ANNs are run in parallel with the DIgSILENT power system simulations. The software implementation follows the approach shown in Fig. 4. This setup will enable power system neuro-control in the next phase of the research.

During the initialization stage, the weights generated during offline training are loaded into the ANN, and from that point on the ANN run in parallel with the DIgSILENT power system model. Pointer based data sharing between DIgSILENT and the ANN software avoids the need to transfer large amounts of data at each ANN time step, which results in a small reduction in the time it takes to run the DIgSILENT simulations after adding the neuro-identifiers.

III. Selecting A Scalable ANN Architecture

This section evaluates the feasibility of the TDNN and the RNN for scaling up to identify larger power systems. Emphasis is given to observing the generalization capabilities of the networks by varying the operating point of the power system while keeping the weight parameters of the networks constant. The main motivation for completing this portion of the investigation is to determine whether the use of the more complex RNN results in a significant increase in accuracy and generalization capabilities of the ANN based identifiers.

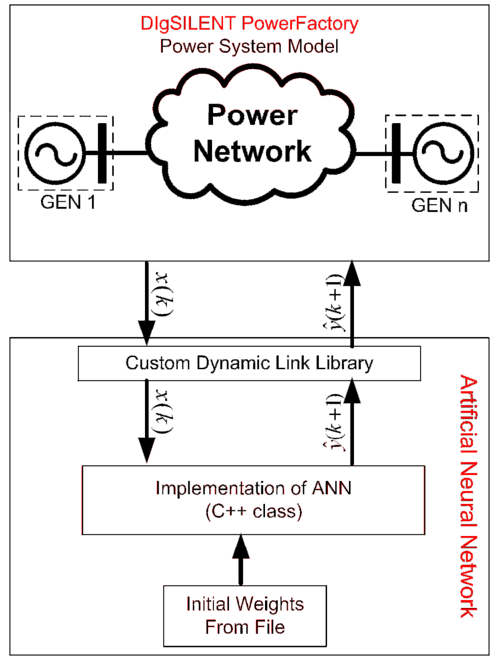

Figure 4. Implementation of ANN software for integration with DIgSILENT

The Single Machine Infinite Bus (SMIB) system shown in Fig. 5 is the test system used during the evaluations. The infinite bus is represented as an ideal voltage source, and the network is represented by algebraic equations (ignoring the network dynamics is common practice in transient stability studies [20]); therefore, the only dynamics in the system are given by the generator, its automatic voltage regulator, and its turbine and governor, i.e., the plant being identified. The goal of the neuro-identification is to learn the dynamics of the plant using only input-output data collected a-priori to produce one-step-ahead predictions of the terminal voltage and rotor speed. The input and output vectors depicted in Figs. 1-3 are defined in (6) and (7), where ω and u_t are the speed and terminal voltage of the machine, and u_t^{setp} and p^{setp} are the terminal voltage and power output references to the machine's automatic voltage regulator and turbine governor respectively.

$$x(k) = \left[\omega(k) \quad u_t(k) \quad u_t^{setp}(k) \quad p^{setp}(k)\right]^T \tag{6}$$

$$\hat{y}(k+1) = \left[\hat{\omega}(k+1) \quad \hat{u}_t(k+1)\right]^T \tag{7}$$

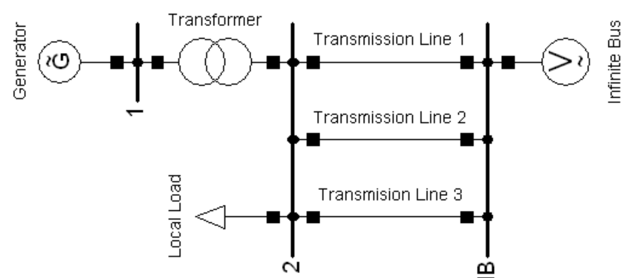

Figure 5. Single machine infinite bus (SMIB) power system

During testing the PRBS used to excite the dynamics of the system and generate the training data are no longer present in the system. The system events (or simulation conditions) used to test the performance of the ANN consist of lines opening and closing, increasing and decreasing the local load, and changing the power and voltage set point commands of the generator. The same set of events is applied at the five operating points labeled as A-E in Table I. The variation in operating points is produced by changing the steady state active power output and terminal voltage of the machine (terminal voltage changes directly result in changes in the reactive power output Q), as well as the infinite bus voltage.

The operating points labeled as T1 and T2 were used to generate the training data set, following the procedure outline in section II. T1 and T2 are expected to provide a rich enough data set for the neuro-identifiers to learn the dynamics of the system. Only PRBS signals are used to generate the training data (none of the system events used for testing are used to generate training data).

TABLE I SIMULATED OPERATING POINTS OF SMIB

Op. Point	T1	T2	A	B	C	D	E
u_t^{setp} [pu]	1.025	0.975	1.025	1.01	1.05	1.05	1.05
p^{setp} [pu]	0.714	0.071	0.714	0.071	0.714	0.929	0.929
Q [pu]	0.152	0.130	0.152	0.091	0.213	0.240	0.304
V_{IB} [pu]	1.00	0.95	1.00	1.00	1.00	1.00	0.95

Typical time-domain waveforms obtained during testing are shown in Fig. 6. The simulated power system outputs (red) as well as two sets of neuro-identifier outputs are shown: TDNN based (green) and RNN based (blue). Note that both identifiers perform with acceptable accuracy and as a result all curves in the figure overlap. Closer inspection of these waveforms revealed that the ANN outputs are in fact predictions. It is important to point to this fact because ANNs can sometimes learn to "cheat" and simply pass some of the inputs directly to the outputs, especially when the sampling time is small compared to the time constants in the system. For presentation purposes, the step-ahead predictions were delayed by one sample to allow them to overlap with the actual signal. Performance metrics were then calculated for the different ANN at each operating point using the simulation results. Figs. 7-9 summarize the results.

Two conclusions are drawn from the SMIB evaluations. The first is arrived at heuristically: the time and memory requirements for training the RNN are considerably higher than for the TDNN. Increases in the order to tens of minutes in the time it takes to train the RNN compared to the TDNN were observed. Also, out-of-memory errors during RNN training were frequent. The second can be derived from Figs. 7-9: there is no noticeable increase in the accuracy or generalization capabilities of the neuro-identifier by using the more complex RNN. This second conclusion cannot be generalized; it is specific to the problem at hand. The RNN can now be discarded as a candidate for the next stage of testing since the increase in complexity does not result in any noticeable performance improvement. Therefore, only the TDNN is considered during the scaling evaluations.

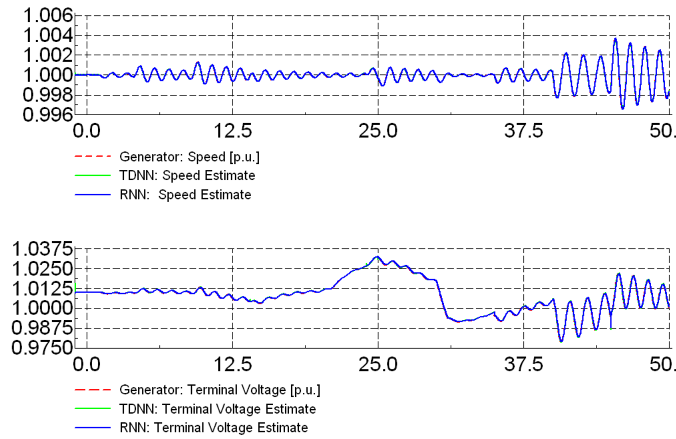

Figure 6. Actual and estimated rotor speed (top) and terminal voltage (bottom) of generator in SMIB using TDNN and RNN at operating point A

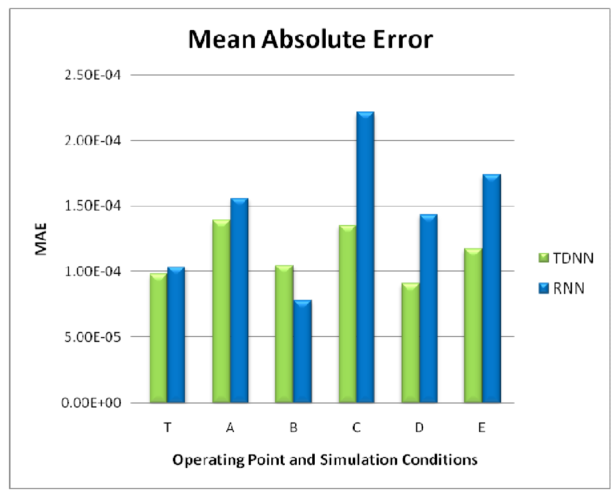

Figure 7. Mean Absolute Error at different operating points

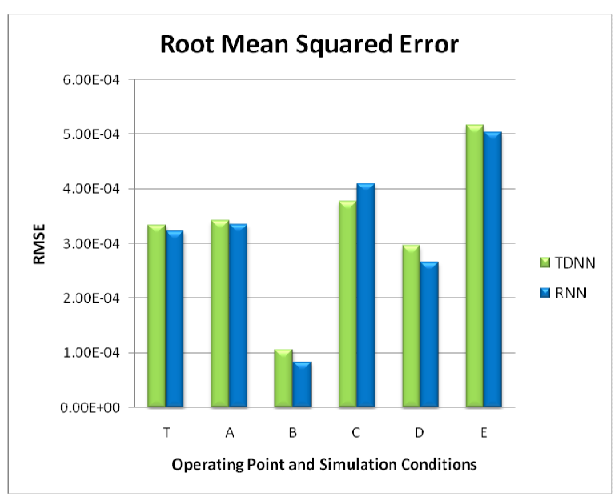

Figure 8. Root Mean Squared Error at different operating points

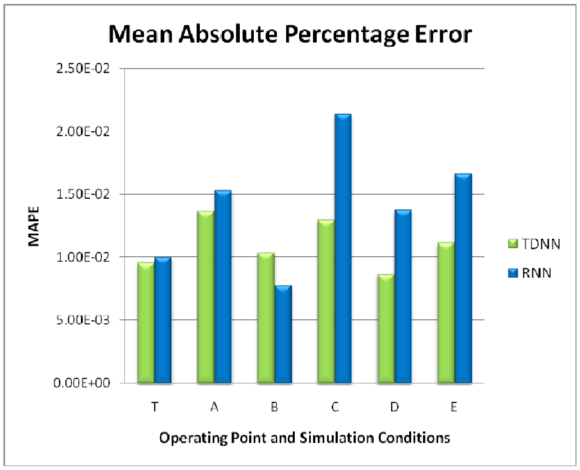

Figure 9. Mean Absolute Percentage Error at different operating points

IV. SCALING UP FOR NEURO-IDENTIFICATION OF THE NEW ENGLAND/NEW YORK 68-BUS TEST SYSTEM

The 68-Bus power system is utilized for the second stage of evaluations. The one line diagram of the system is shown in Fig. 10. It contains 16 generators, their automatic voltage regulators, turbines, and governors, as well as the power network with transmission lines, loads, etc. The power network is still represented by a set of algebraic equations, neglecting the fast dynamics of the lines, transformers, etc. The goal of the neuro-identification is to learn the dynamics of a specific number of generators in the system (this number is increased as the experiment progresses) to provide one-step-ahead predictions of their terminal voltages and rotor speeds. Simulations of this system are used to test the accuracy of the results obtained with the TDNN as it is scaled up using two approaches: a *distributed approach*, and a *centralized approach*.

A. *The Centralized Approach*

The centralized identifier shown in red in Fig. 11 receives measurements from across the system into a single location, thus it receives global information. In this approach, a single fully connected TDNN receives the speed, terminal voltage, terminal voltage set point, and active power output set point from all the generators being identified in the power system, and it uses them to generate step ahead predictions for the terminal voltage and speed of each of those generators.

When implemented for identification of all generators in the 68-bus system, the centralized approach results in a single TDNN with 192 inputs (64 present inputs plus two time delayed versions of each), 32 outputs (16 speeds and 16 terminal voltages), and 18032 trainable parameters.

Obtaining data from across the system into a single location has implications in terms of the communication requirements for implementing this strategy. Such requirements are not considered in this work but are paramount for real world implementation.

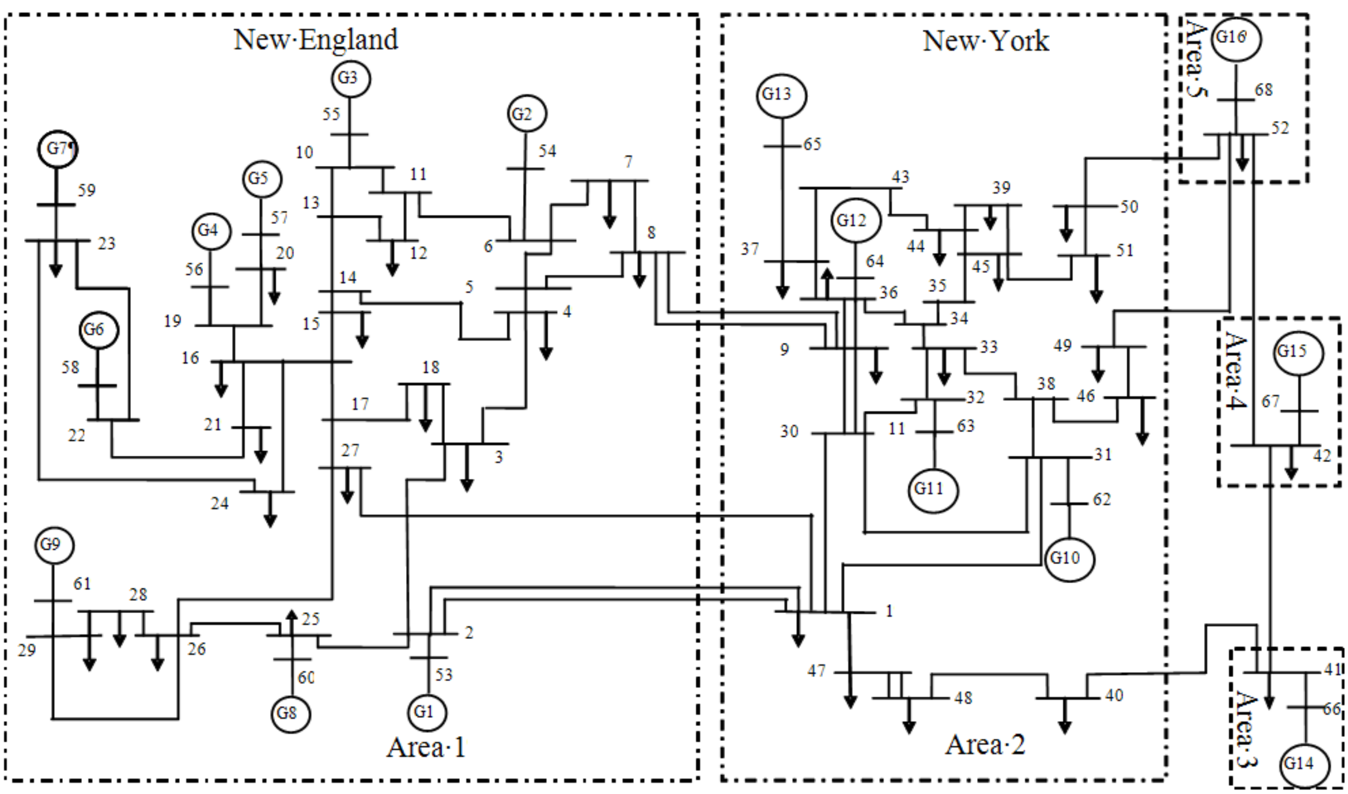

Figure 10. 68-bus New England/New York power system

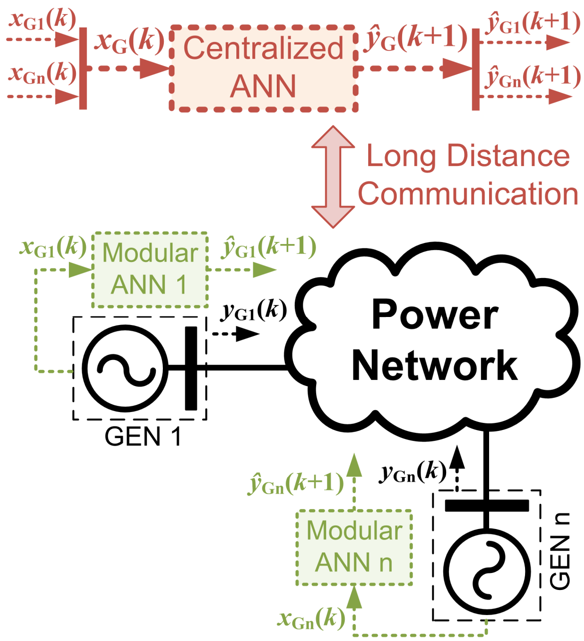

Figure 11. Centralized (red) and distributed (green) approaches for power system identification

B. The distributed approach

The distributed approach shown in green in Fig. 11 uses only local measurements. In contrast to the single TDNN used for identification using the centralized approach, the distributed method uses multiple modular TDNN, one for each generator being identified. The inputs to each of the modular TDNN are the speed, terminal voltage, terminal voltage reference set point, and active power set point for one of the generators being identified.

When implemented for identification of all generators in the 68-bus system, the distributed approach results in 16 independent TDNN, each with 12 inputs (4 present inputs plus two time delayed versions of each), 2 outputs (one speed and one terminal voltage), and 77 trainable parameters.

The decentralized structure relaxes the communication requirements; however, it also has implications with respect to the usability of the resulting identifiers when the identification is being done with global control goals in mind. The localized structure of each of the distributed ANN prevents them from learning the coupling relationships among the different generators in the system, i.e. how the inputs to one generator affect the outputs of other generators. Such control implications limit the usability of this approach and are not discussed further in this paper.

C. Comparisons

One of the main concerns for implementing neuro-identification algorithms for realistically sized power system is the effect of having a large number of weights for the ANN. Such an increase in the number of parameters has a strong negative impact on the effectiveness of the training algorithms to find appropriate solutions for the ANN weight parameters. Having more weights increases the dimension and complexity of the solution space, potentially producing a larger number of local minima where the training algorithms can get stuck, while also increasing the time it takes to find an acceptable solution.

Fig. 12 shows the increase of ANN parameters and the time it takes to train the networks for the distributed (dotted) and centralized (solid) approaches as the number of generators being identified grows. The TDNN parameters used to obtain the results in Figs. 12 are shown in Table II.

For the centralized approach, the number of neurons in the hidden layer is increased linearly with the number of generators being identified (5 neurons are added to the hidden layer for each extra generator added to the identification). While there is no assurance that this is the most efficient method to scale up the hidden layer of the TDNN, simulation results show that acceptable accuracy can be obtained in this manner.

TABLE II PARAMETERS FOR SCALING UP THE TDNN

Neurons per Generator	**Outputs per Generator**	**Inputs per Generator**	**Delays per Input**
5	*2*	*4*	*2*

These graphs take into account the actual training time, but ignore the time it takes to prepare the data, setup the experiment, etc., which can have a noticeable impact on the overall time it takes to get the neuro-identification algorithms up and running.

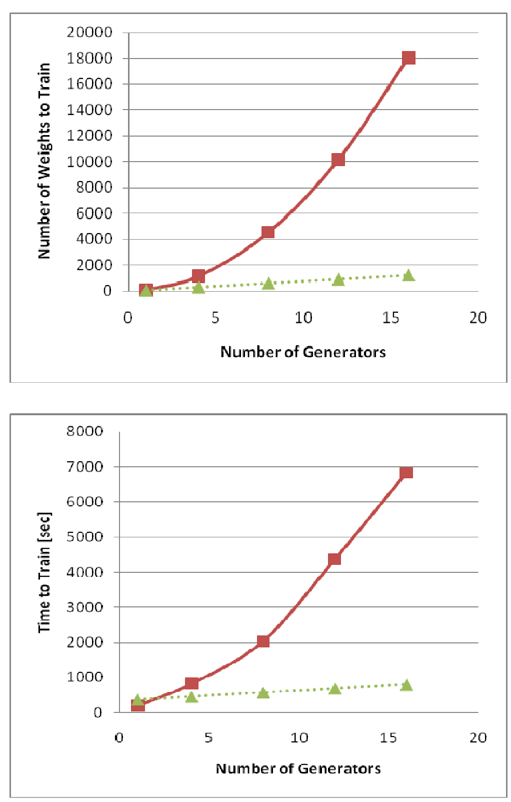

Figure 12. Increase of weights (top) and time it takes to train (bottom) as more generators are identified using centralized (red-solid), and distributed (green-dotted) approaches

A set of system events is defined to observe the accuracy of the different approaches. Fig. 13 presents the results obtained for the actual and predicted values for one of the generators (G1 in Fig. 10) after applying the following sequence of events to the 68-bus test system:

- Open one of the transmission lines connecting busses 8 and 9 @ t = 1 [sec]
- Place a solid 3-phase fault in the middle of one of the transmission lines connecting busses 1 and 2 @ t=14.95 sec (critical clearing time ≈ 60 msec)
- Open the faulted transmission line connecting busses 1 and 2 @ t = 15 sec
- Add 1130.4 MW and 95.8 MVar by ramping up the loads at busses 15, 16, 20, 21, 27 by 50% @ t = 30 sec
- Increase the terminal voltage set-point of the AVR on G1 by 5% @ t = 45 [sec]
- Increase the power set-point on generators G1 trough G9 by 10% to account for added loads@ t = 60 [sec]

Fig. 13 demonstrates that both approaches to scaling up produce ANNs capable of predicting the behavior of the system with reasonable precision. Even though none of the events used for testing are presented during training, the maximum error observed during unfaulted operation is less than 0.5%. This shows that the training data generated by applying PRBS combined with the offline training procedure results in ANNs that learn the dynamics of the power system. During the fault, neither approach tracks the dynamics of the system, thus the error reaches large values. This inability to track the system during faults is expected and acceptable. Faults make drastic but short-lasting changes to the system. Note that the operating conditions of the power system vary significantly due to the events applied. During the simulation two critical transmission lines are permanently outaged, severely weakening the transmission corridors interconnecting the New England and New York portions of the system. Also, load increases and subsequent generator active power set point changes cause the power flows to be redistributed trough the network.

In spite of the positive results obtained during this work, one issue remains unresolved. Can the TDNN be scaled up to identify much larger portions of a power system? Fig. 12 points to the obvious disadvantages of the centralized approach. The computational burden resulting from the fast increase in weight parameters to be trained becomes prohibitive as the number of generators being identified grows. However, this centralized approach provides a more comprehensive model of the system since it takes into account the coupling relationships between generators. The computational burden resulting from the distributed approach increases much more gracefully, but the resulting identified model does not provide the coupling relationships between generators. For some applications, this might prove to be unacceptable.

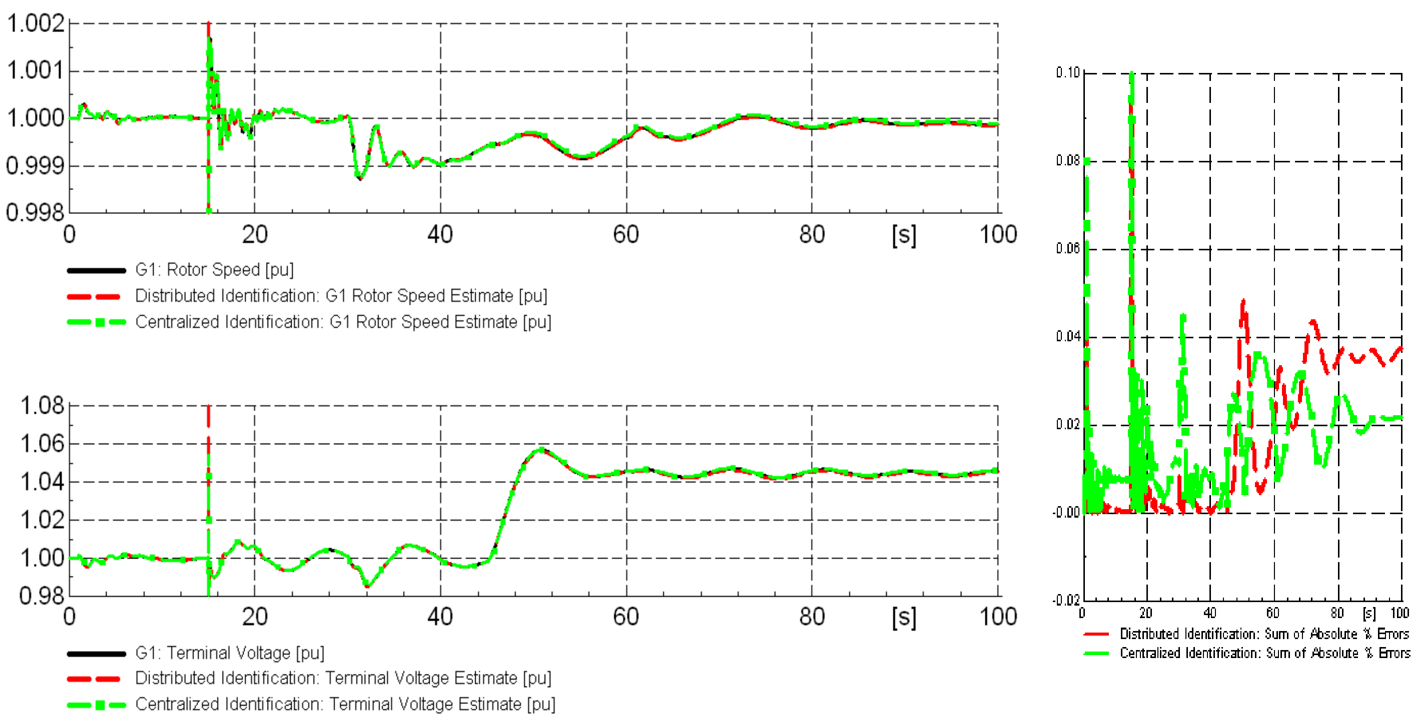

Figure 13. Actual and estimates of the rotor speed (top left) and terminal voltage (bottom left) of generator G1 in the 68-bus New England/New York power system. Sum of absolute % errors shown on right

V. Conclusions

This paper considered the identification of power systems using ANN. The capabilities of the TDNN and the RNN for identifying the non-linear dynamic behavior of a single machine infinite bus system operating at different operating points were studied. That portion of the study revealed that the increase in computational complexity resulting from utilizing the RNN did not offer noticeable improvements in the precision of the identified model at least for small power systems.

The New England/New York test system was then used to evaluate the effects of scaling up TDNNs to identify larger systems. After a heuristic search for appropriate training algorithms, the scaled conjugate gradient training algorithm was selected on the basis of faster convergence when applied to large networks. During testing, it was observed that the networks were capable of achieving acceptable accuracy in the identification task even for the largest case when 16 generators were identified by a single network, but the training times quickly rose with the size of the system being identified. The increase in training time observed when using the distributed approach was much more gradual; however, with this approach it is impossible for the resulting ANN to capture the dynamics and coupling relationships among all the generators in the system, and this has serious negative implications in terms of neuro-control.

Different network architectures than those studied in this paper will be necessary for neuro-identification and subsequent neuro-control of realistically sized power systems to become a reality. ANN architectures with improved properties for scaling up are currently being explored to address the issues found in this paper.

Acknowledgment

This work was supported by the National Science Foundation under EFRI Grant #0836017.

References

[1] T. Senjyu, H. Takara, K. Uezato, T. Funabashi, "One-hour-ahead load forecasting using neural network," *IEEE Transactions on Power Systems*, vol. 17, no. 1, pp. 113-118, Feb 2002.

[2] R.L. Welch, S.M. Ruffing, G. K. Venayagamoorthy, "Comparison of feedforward and feedback neural network architectures for short term wind speed prediction," *Proc. of the International Joint Conference on Neural Networks, 2009*, pp.3335-3340, 14-19 June 2009.

[3] T.T. Nguyen, R. Gianto, "Neural networks for adaptive control coordination of PSSs and FACTS devices in multimachine power system," *IET Generation, Transmission & Distribution*, vol. 2, no. 3, pp. 355-372, May 2008.

[4] J. Liang, J. Dai, G.K. Venayagamoorthy, R.G. Harley; , "Dynamic system eigenvalue extraction using a linear echo state network for small-signal stability analysis - a novel application," *Proc. of the International Joint Conference on Neural Networks, 2010*, pp.1-8, 18-23 July 2010

[5] S. Mohagheghi, G.K. Venayagamoorthy, R.G. Harley, "Optimal Wide Area Controller and State Predictor for a Power System," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 693-705, May 2007.

[6] J. Upendar, C.P. Gupta, G.K Singh, G. Ramakrishna, "PSO and ANN-based fault classification for protective relaying," *IET Generation, Transmission & Distribution*, vol. 4, no. 10, pp. 1197-1212, October 2010.

[7] S. Mohagheghi, G.K. Venayagamoorthy, R.G. Harley, "A dynamic recurrent neural network for wide area identification of a multimachine power system with a FACTS device," *Proceedings of the 13th International Conference on Intelligent Systems Applications to Power Systems*, pp.6 pp., 6-10 Nov. 2005

[8] David J. C. MacKay, "A Practical Bayesian Framework for Backpropagation Networks," *Neural Computation*, vol. 4, no. 3, May 1992, pp. 448-472.

[9] Lutz Prechelt, "Automatic early stopping using cross validation: quantifying the criteria", *Neural Networks*, vol. 11, no. 4, pp. 761-767, June 1998

[10] L.L Grant, G.K. Venayagamoorthy, "Cellular Multilayer Perceptron for Prediction of Voltages in a Power System," *Proc. of the 15th International Conference on Intelligent System Applications to Power Systems, 2009*, pp.1-6, 8-12 Nov. 2009

[11] S. Muknahallipatna, B.H. Chowdhury, "Input dimension reduction in neural network training-case study in transient stability assessment of large systems," *Proc. of the International Conference on Intelligent Systems Applications to Power Systems, 1996*, pp.50-54, 28 Jan-2 Feb 1996

[12] D. Aberdeen, J. Baxter, R. Edwards, "92¢ /MFlops/s, Ultra-Large-Scale Neural-Network Training on a PIII Cluster," *Proc. of the ACM/IEEE 2000 Conference on Supercomputing*, pp. 44- 44, 04-10 Nov. 2000

[13] J. Lu, M. Yang, N. Bourbakis, D. Goldman, "High-performance neural network training on a computational cluster," *Proc. of the Seventh International Conference on High Performance Computing and Grid in Asia Pacific Region*, vol., no., pp. 467- 472, 20-22 July 2004

[14] K.S. Narendra, K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol.1, no.1, pp.4-27, Mar 1990

[15] S. Haykin, "Neural Networks: a comprehensive foundation," 2nd ed., Prentice Hall, 1998. ISBN-13: 978-0132733502.

[16] D. Prokhorov, G. Puskorius, and L. Feldkamp, "Dynamical neural networks for control," in J. Kolen and S. Kremer (Eds.), "A Field Guide to Dynamical Recurrent Networks," *IEEE Press*, 2001. ISBN-13: 978-0780353695.

[17] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, vol. 6, no. 4, pp. 525-533, 1993

[18] U. Seiffert, "Training of Large-Scale Feed-Forward Neural Networks," *Proc. of the International Joint Conference on Neural Networks 2006*, pp.5324-5329,

[19] Gong Li, Jing Shi, "On comparing three artificial neural networks for wind speed forecasting", *Applied Energy*, vol. 87, no. 7, July 2010, pp. 2313-2320.

[20] P. Kundur, "Power system stability and control," New York: McGraw-Hill, 1994. ISBN-13: 978-0070359581.

[21] DigSILENT PowerFactory Vesion 14 Manual, DigSILENT GmbH, 2008