Wide Area Monitoring in Power Systems Using Cellular Neural Networks

Bipul Luitel, *Student Member, IEEE* and Ganesh K. Venayagamoorthy, *Senior Member, IEEE*

***Abstract*—The demand of power and the size and complexity of the power system is increasing. Wide area monitoring and control is an integral part in transitioning from the traditional power system to a Smart Grid. However, wide area monitoring becomes challenging as the size of the electric power grid, and consequently the number of components to be monitored, grows. Wide area monitor (WAM) designed using feed-forward and feedback neural network architectures do not scale up to handle the growing complexity of the Smart Grid. In this paper, cellular neural network (CNN) is presented as a way to provide scalability in the development of a WAM for Smart Grid. The CNN based WAM is compared with multilayer perceptrons (MLP) based WAM on two different power systems. The results show that the CNN has better or comparable performance with, and scales up much better than, MLP.**

***Index Terms*—Backpropagation, CNN, Cellular Multilayer Perceptron, MIMO, Power system, Wide Area Monitor**

I. Introduction

Stability of electric power system depends on proper functioning of various power system components. Power system is a massively distributed network. Therefore, constant remote monitoring is necessary to assess the current state of these components. Based on this assessment, related control action is taken on the power system components in order to keep the system in stability. Wide area monitoring and control system (WAMCS) has, therefore, become critical for the power grid. However, with the addition of more distributed resources and micro-grids to a smart grid, the number of variables to be monitored will increase and wide area monitoring and control of such complex dynamic system will become a challenge.

Applications of wide area monitoring systems (WAMS) in power system for state estimation, disturbance identification and wide area PSS have been reported in literature [1], [2]. Various design aspects of WAMCS are studied in [3]. Unlike traditional methods of data acquisition and control, typically supervisory control and data acquisition (SCADA) methods that relied on remote terminal units (RTU) for data, WAMS utilize phasor measurement units (PMUs) for collecting data from the power system on a faster time-scale and hence can be used to monitor transient and dynamic response of the system [1]. A substation based dynamic state estimator has been used as WAMS in [4] that provides abilities to predict instabilities before they occur. Although these various techniques are being used and developed for wide area monitoring, there are still major challenges in their use for control. These challenges are related to extracting dynamics of the system without knowing the system model, mining and interpreting huge amount of data available from monitoring devices and assessment of the overall dynamics of the system based on wide area information [1]. It is even bigger challenge to make reliable control decisions under real-time constraints.

Computational intelligence (CI) techniques have shown promises in the field of wide area monitoring and control [5]. Since neural networks (NN) can be used to represent the dynamics of the system by training on the historical data of the system without having to know its actual model, they have shown promises in predictive control applications. NNs have been successfully implemented as state predictors and neurocontrollers [6] in the areas of wide area monitoring and control. Simultaneous recurrent neural network (SRN) and echo state network (ESN) based wide area monitor (WAM) has been demonstrated to be quite effective in performing predictive neuroidentification of distributed power systems for the purposes of accurate control [6], [7]. Radial basis function networks have been used for wide area monitoring with an adaptive critic designs based control in [8]. However, these feed-forward and feedback neural network architectures do not scale up to handle the growing complexity of the smart grid for wide area monitoring and control. As the number of variables increases, the number of neurons in the NN increases and so does the computational complexity. Therefore, it becomes challenging for the NN training algorithms to correctly learn the non-linear system dynamics.

A cellular neural network (CNN) overcomes this problem of scalability by dividing a huge network into sub-networks among different cells where each cell consists of a neural network that deals with fewer variables and hence fewer neurons and lesser computational complexity. These cells are interconnected in such a way that the connectivity and the dynamics of the actual power system is preserved. In this paper, WAM is developed using a multilayer perceptron (MLP) based CNN, also known as a cellular MLP (CMLP). The design is applied to two benchmark power systems for predicting the speed deviations of the generators. The development and

The authors are with the Real-Time Power and Intelligent Systems Laboratory, Missouri University of Science & Technology, Rolla, MO 65409 USA, Contact: {iambipul, gkumar}@ieee.org

The funding provided by the National Science Foundation, USA under the grants CAREER ECCS #0348221 and EFRI #0836017 is gratefully acknowledged. The research was also partially supported by IEEE CIS Walter Karplus graduate student research grant.

training mechanisms for CMLP are described and the results are compared with a WAM developed using a multiple-inputs multiple-outputs (MIMO) MLP. The remaining sections of the paper are arranged as follows: development of WAM using CMLP is described in Section II. CMLP training approach is described in Section III. Results and discussions are presented in Section IV, and conclusions in Section V.

II. Development of a CNN based WAM

Two test systems are considered for this study. Test System I is the 12-bus benchmark system shown in Fig. 1 [9]. It consists of three generators, one in each area. Test System II is the two-area four-machine system shown in Fig. 2 [10]. It consists of four generators, two in each area. The WAM is developed to predict the speed deviations ($\Delta\hat{\omega}$) of each generator in the system at time instant $k+1$ based on speed deviations ($\Delta\omega$) and deviation of the reference voltage (ΔV_{ref}) (shown in Fig. 3) of the generators at time instant k as the inputs.

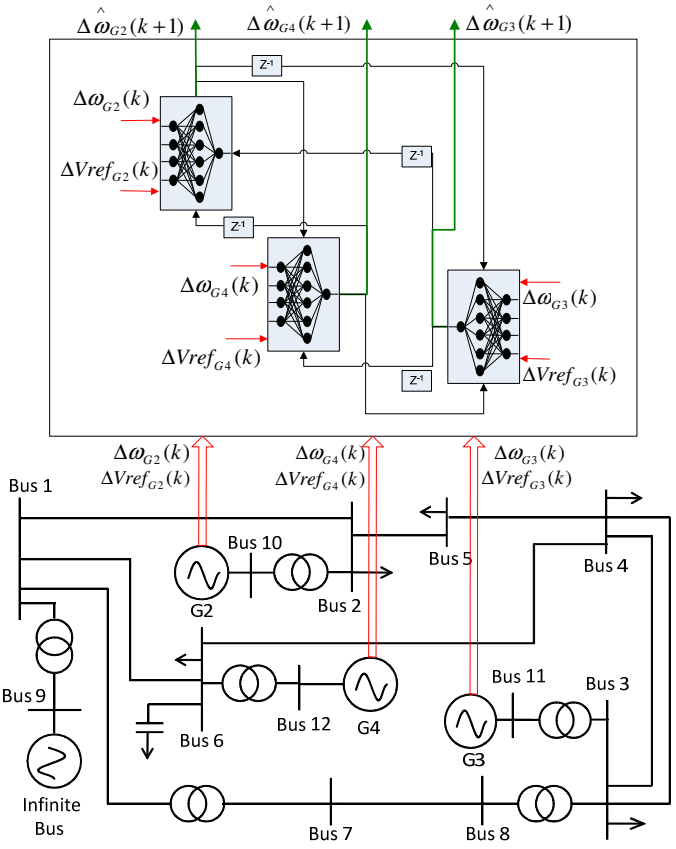

Fig. 1. CNN based WAM for Test System I (12-bus system).

The WAM is implemented using a CMLP where each cell of the network consists of an MLP and represents one generator of the power system. The cells are interconnected based on 'nearest-n neighbors' topology, which means previous sample outputs of n nearest neighbors of each cell are connected to the inputs of that cell. The "nearness" is defined as the electrical distance between the generators and is measured based on the length of the transmission lines

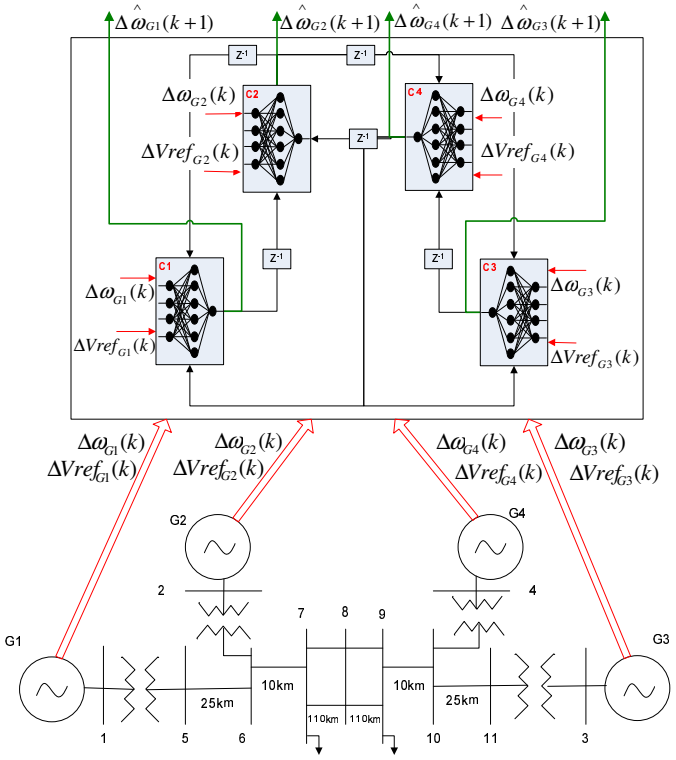

Fig. 2. CNN based WAM for Test System II(two-area four-machine system).

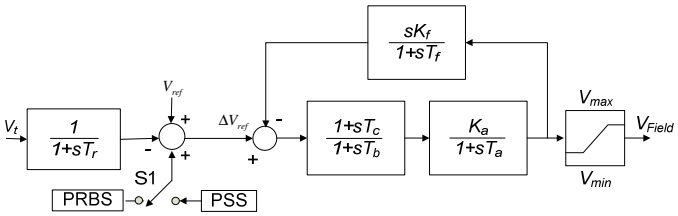

Fig. 3. Generator excitation system (showing application of PRBS and ΔV_{ref}).

separating the two generators. In this study, two nearest neighbors are considered for developing CMLP based WAM. For example in Fig. 2, two nearest neighbors of generator G1 are generators G2 and G4. This is represented in the CMLP by connecting the outputs of the cells C2 and C4 to the inputs of the cell C1. Similarly for G4, two nearest neighbors are G2 and G3 and hence outputs of the cells C2 and C3 are connected to the inputs of the cell C4. This topology allows for the scalability of the CMLP by keeping the size of the MLP in each cell to a minimum. The MLP in each cell consists of an input layer with four neurons, a hidden layer with six neurons and an output layer with a single neuron. This choice of number of neurons in the hidden layer is determined by trial and error and this paper does not compare and contrast against different number of neurons in the hidden layer. Therefore, it is not claimed to be optimal. The four inputs to the MLP in each cell consist of $\Delta V_{ref}(k)$ and $\Delta\omega(k)$ associated with the generator represented by the

cell and $\Delta\hat{\omega}(k)$ associated with the generators represented by the two nearest neighboring cells. The output of the CMLP is $\Delta\hat{\omega}(k+1)$ of the generator associated with the cell, where k is the sample index of the signal. This is explained in Fig. 2. A CMLP for the 12-bus system consists of an identical architecture with three cells and is shown in Fig. 1.

Fig. 4 shows the implementation of the WAM using a three layered feed-forward MLP for predicting the speed deviations of the three generators in the 12-bus system. It consists of six neurons in the input layer, 10 neurons in the hidden layer and three neurons in the output layer, one output representing the step-ahead predictions of speed deviation for each generator. The six inputs to the network are the two inputs ($\Delta\omega$,ΔV_{ref}) going into the WAM from each generator. The second test system is formulated similarly with eight input, 15 hidden and four output neurons for predicting the speed deviations of the four generators in the two-area four-machine system. and is shown in Fig. 5.

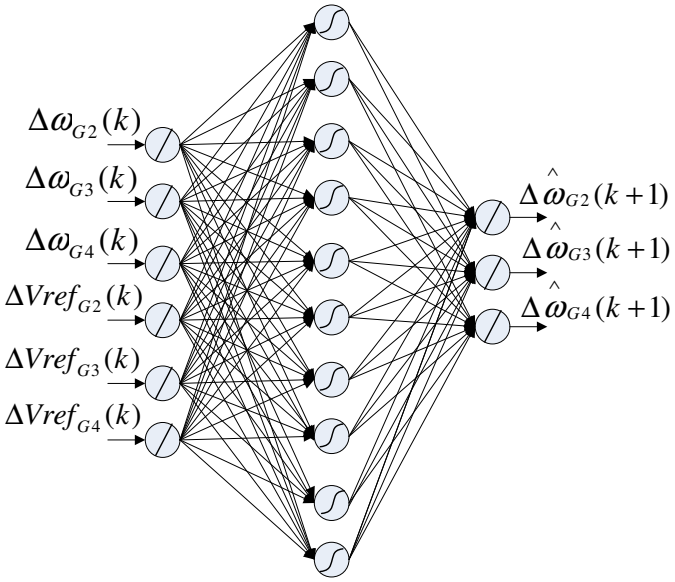

Fig. 4. Implementation of WAM using MLP for Test System I (12-bus system).

III. CNN Training

The neural networks are trained online using backpropagation algorithm [11]. In this approach, weights of the neural network are updated after every sample is passed through the network. After all the samples are covered, this process is repeated for as many passes through the network as required to achieve better convergence, as explained in [11]. Values of various parameters involved in training are listed in Table I. The training data is collected from the test systems designed on RSCAD and simulated on a Real-time Digital Simulator [12]. During the forced training, all of the generators are simultaneously perturbed using a pseudo-random binary signal (PRBS) (shown in Fig. 6) applied to the excitation system of the generators. The deviation of the generator

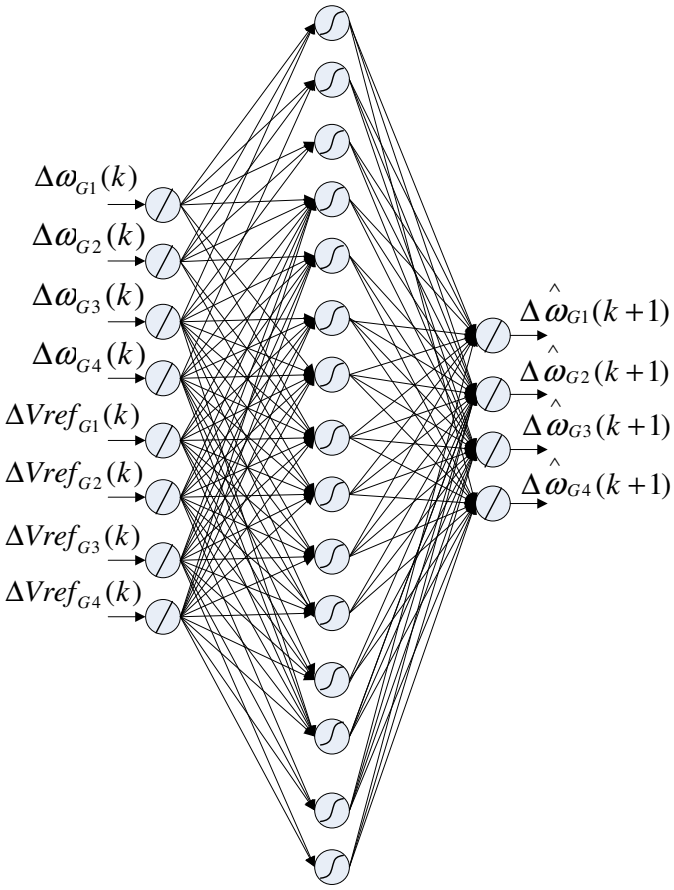

Fig. 5. Implementation of WAM using MLP for Test System II (two-area four-machine system).

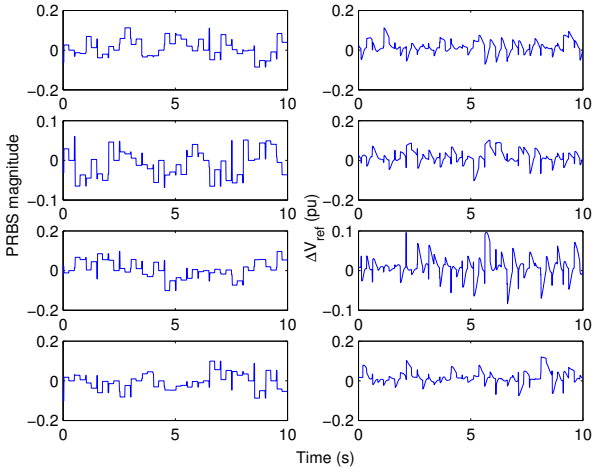

Fig. 6. PRBS signals applied to the four generators and the resulting ΔV_{ref}.

speed as a result of the PRBS perturbation is recorded along with the reference voltage applied to the generator excitation system (ΔV_{ref} in Fig. 3). The MIMO MLP is trained using these two signals of each generator as the inputs.

TABLE I
PARAMETERS USED FOR TRAINING MLP

Trials	50
Number of passes	100
Learning Rate (μ)	0.005
Momentum Gain (δ)	0.001

In case of CMLP, each cell is treated as an "object" and therefore, all of the cells are simultaneously trained with similar parameters. Since no parallel hardware/software platform is used in this study, the cells are trained sequentially, one cell after the other. However, the property of their parallel implementation is still maintained. The parallel training approach of each cell object of the CMLP is explained further.

Each cell consists of four inputs viz. actual reference voltage applied to the generator excitation system $\Delta V_{ref}(k)$, actual speed deviation of the generator $\Delta\omega(k)$ and the predicted speed deviations of the nearest two generators, $\Delta\hat{\omega}_{k1}(k)$ and $\Delta\hat{\omega}_{k2}(k)$. For every sample of the input data $I(k)$, each cell produces a step-ahead predicted output $O(k+1)$. Therefore, for any input data of size $1, 2, \ldots, k, \ldots, K$ discrete samples, and W_n and V_n be the input and output weight matrices respectively of the MLP in n^{th} cell, then the output of each cell is given by:

$$\begin{aligned} O_n(k) &= \Delta\hat{\omega}_n(k) \\ &= f\left(I_n(k-1), W_n(k), V_n(k)\right) \end{aligned} \quad (1)$$

Thus, $I_n(k) = [\Delta V_{refn}(k) \;\; \Delta\omega_n(k) \;\; \Delta\hat{\omega}_{n1}(k) \;\; \Delta\hat{\omega}_{n2}(k)]$ uses the predicted output of the previous sample in case of the neighboring cells $n1$ and $n2$. This helps the parallelization of the cell objects, as long as the calculation of each sample output is synchronized among the different cells. In MATLAB, this is achieved by training each cell sequentially for every sample. After the output is calculated for each cell, the weights of each cell are updated before calculating the output for the next sample. This process of online training of a CMLP using backpropagation is shown in the flowchart of Fig. 7. The part in the flowchart surrounded in dark box shows the process that can be implemented in parallel irrespective of the number of cells when a suitable platform is available.

IV. Results and Discussions

A. Test System I

For Test System I, only one operating point is considered. A MIMO MLP is trained and tested on the same data set using the architecture explained above. The same training data is also used to train a CMLP consisting of three cells. Fig. 8 shows the actual versus predicted speed deviations of the three generators obtained from CMLP. The mean absolute error (MAE) between the actual and the predicted outputs for the two networks are calculated for comparison. The average and standard deviation of the MAE obtained over 50 trials are presented in Table II.

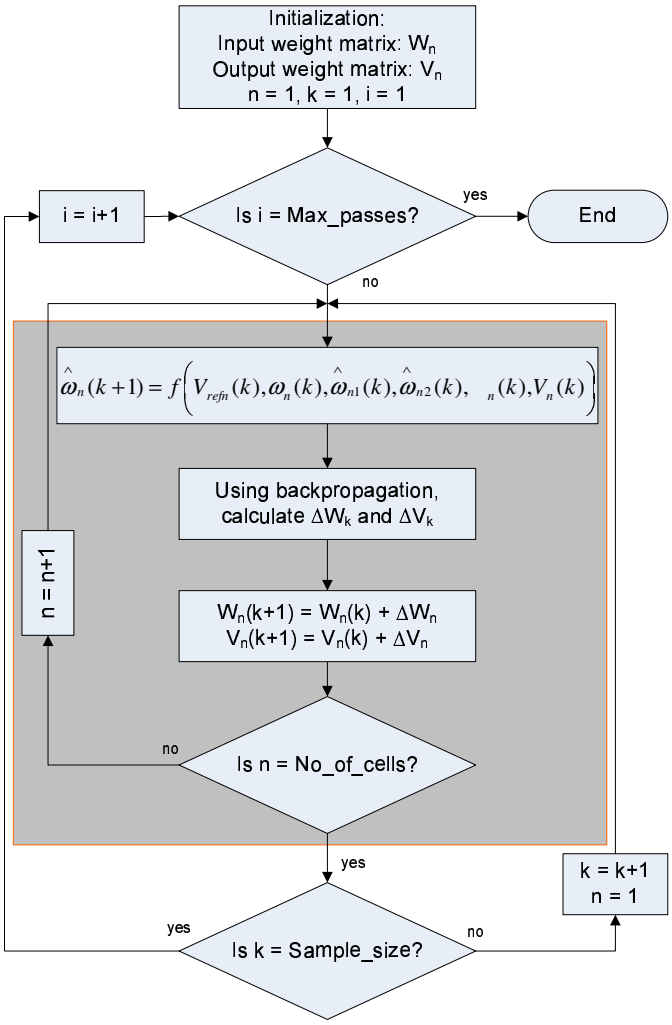

Fig. 7. Flowchart for training of CMLP using backpropagation.

B. Test System II

Different operating points shown in Table III are considered for the Test System II. A CMLP consisting of four cells is trained on OP_1 and tested on OP_1, OP_2 and OP_3. These operating points are different to each other in the amount of power transfer between the two areas of the test system. Testing data is also obtained for operating point OP_4 by causing a 10-cycle 3-phase to ground fault on bus 8 of the test system during OP_1 steady state conditions. Similarly, operating point OP_5 is obtained by causing a line outage on one of the two transmission lines between the buses 7 and 8 in the test system.

Fig. 9 shows the convergence diagram for the four outputs of the MIMO MLP. Similar convergence diagram for the CMLP is shown in Fig. 10. These diagrams show how the mean squared error (MSE) between the actual and the predicted outputs decreases over multiple passes of the training data through the network. The testing outputs obtained from the CMLP for the five operating points are shown in Figs. 11 to

TABLE II
COMPARISON OF MLP AND CMLP IN TEST SYSTEM I

	G2		G3		G4	
	MLP	CMLP	MLP	CMLP	MLP	CMLP
Avg.	0.018403	**0.017512**	0.016520	**0.015853**	**0.014042**	0.014383
Std.	0.001395	**0.000860**	0.001298	**0.000559**	0.000949	**0.000677**
Winner	0	1	0	1	0	1

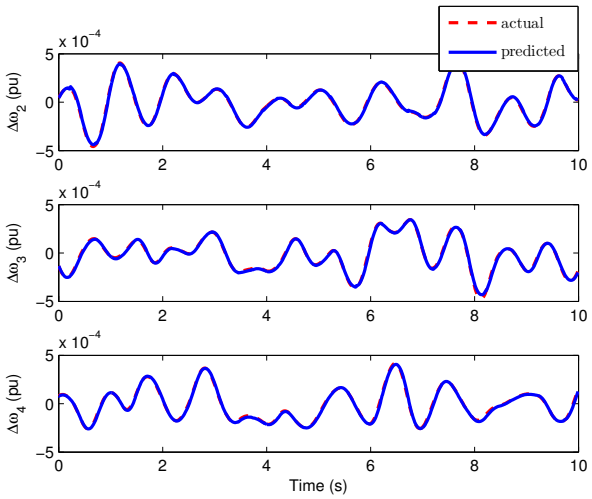

Fig. 8. Output of the CMLP based WAM for the 12-bus system

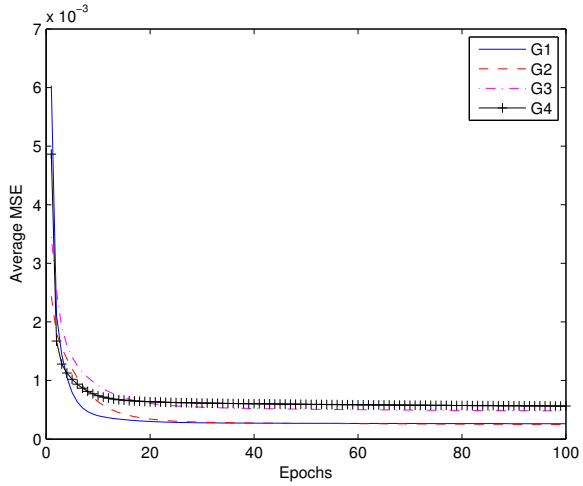

Fig. 9. Convergence of individual outputs of the MIMO MLP during training.

TABLE III
FIVE OPERATING POINTS CONSIDERED IN THE STUDY

	OP_1, OP_4, OP_5	OP_2	OP_3
Load 1 (MW)	950	556	950
Load 2 (MW)	1650	1469	944
$P_{area1 \Leftrightarrow area2}$ (MW)	253.2	302.9	80.45
$Q_{area1 \Leftrightarrow area2}$ (MVar)	22.68	57.2	-38.02
P_{G1} (MW)	705.6	573.8	579.5
Q_{G1} (MVar)	163.5	117.2	53.89
P_{G2} (MW)	705.5	537.7	579.1
Q_{G2} (MVar)	296	234.6	81.12
P_{G3} (MW)	441.5	309.5	314.4
Q_{G3} (MVar)	68.8	49.79	-31.56
P_{G4} (MW)	705.6	537.7	578.6
Q_{G4} (MVar)	169.8	140.1	-59.53

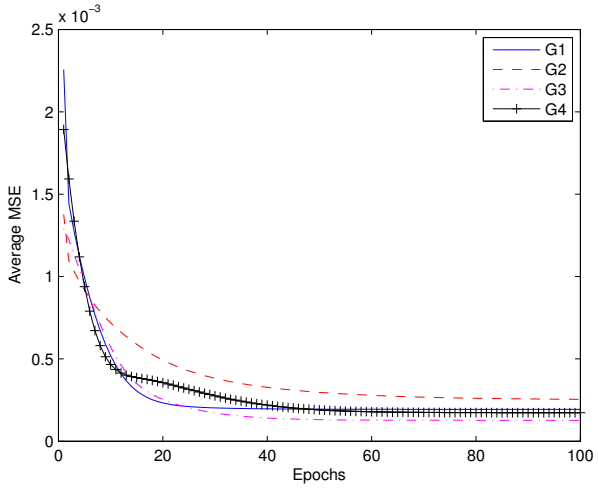

Fig. 10. Convergence of individual cells of the CMLP during training.

15. The comparison of absolute errors obtained using MLP and CMLP for OP_1 to OP_5 are shown in Figs. 16 to 20, respectively. The average and standard deviation of the mean absolute error (MAE) obtained by the two networks during testing on five operating points over 50 trials are shown in Table IV.

C. Analysis

Learning in CNN is a challenging task because of the connectivity between several cells that are learning concurrently. Since the predicted output from one cell is used as input(s) to other neighboring cell(s), errors due to poor training and hence false predictions of the NN in one cell can ripple through all of the cells and deteriorate the performance of the CNN. On the other hand it is also arguable that the NNs get trained even better due to the connectivity because the errors propagate through the network and each cell is trained actively (through its own training) and passively (through the training of its neighbors) as training algorithm on each cell tries to minimize the error at its output. This way, knowledge of the actual dynamics of the system is preserved not only on the individual neural networks at each cell, but also on the connectivity between the different cells of

TABLE IV
COMPARISON OF MLP AND CMLP FOR DIFFERENT OPERATING POINTS IN TEST SYSTEM II

		G1		G2		G3		G4	
		MLP	CMLP	MLP	CMLP	MLP	CMLP	MLP	CMLP
OP_1	Avg.	**0.010533**	0.011064	**0.012169**	0.013711	0.010201	**0.009517**	0.013105	**0.010210**
	Std.	0.000225	**0.000202**	**0.000491**	0.000587	0.000499	**0.000088**	0.000431	**0.000242**
OP_2	Avg.	0.012927	**0.008843**	0.014253	**0.011057**	**0.015672**	0.016951	0.019904	**0.013778**
	Std.	0.000560	**0.000330**	**0.000638**	0.001145	0.001045	**0.000227**	0.000825	**0.000245**
OP_3	Avg.	0.013716	**0.012222**	0.014582	**0.013466**	0.013608	**0.011166**	0.013869	**0.011375**
	Std.	**0.000050**	0.000252	**0.000145**	0.000626	0.000521	**0.000135**	0.000320	**0.000130**
OP_4	Avg.	0.003171	**0.001850**	0.006353	**0.005029**	0.007390	**0.001142**	0.008319	**0.002843**
	Std.	**0.000069**	0.000152	**0.000155**	0.000638	0.000122	**0.000023**	0.000737	**0.000213**
OP_5	Avg.	**0.002383**	0.004535	0.003849	**0.002275**	0.002516	**0.001739**	0.008739	**0.006740**
	Std.	**0.000581**	0.000649	**0.000358**	0.000814	0.000590	**0.000084**	0.000467	**0.000322**
Winner (of 5)		1	4	1	4	1	4	0	5

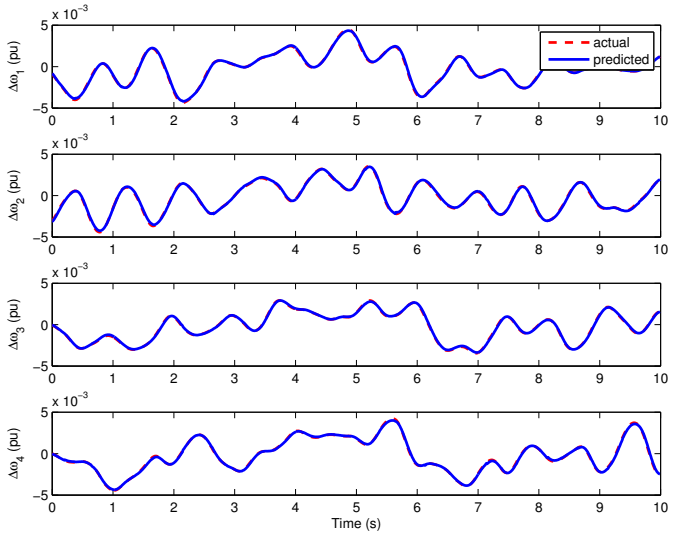

Fig. 11. Testing output of CMLP for operating point I (Test System II). Speed deviation of the generators is shown in y-axis against time in x-axis.

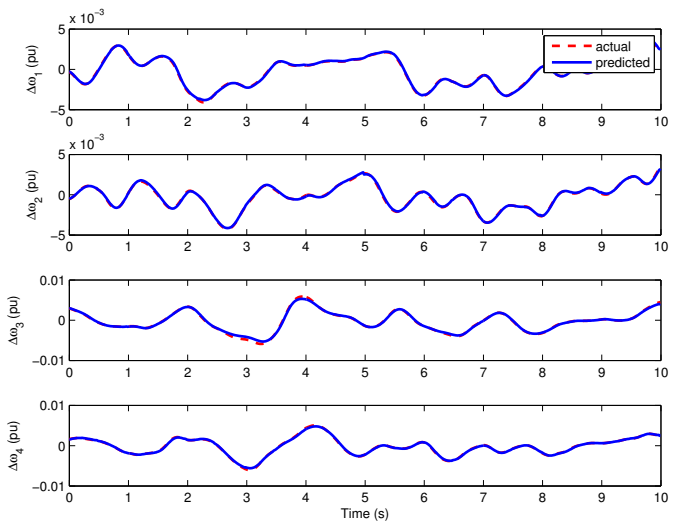

Fig. 12. Testing output of CMLP for operating point II (Test System II). Speed deviation of the generators is shown in y-axis against time in x-axis.

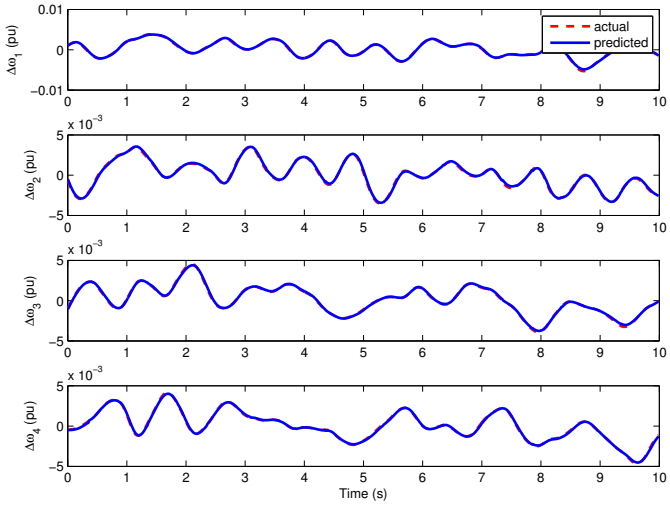

Fig. 13. Testing output of CMLP for operating point III (Test System II). Speed deviation of the generators is shown in y-axis against time in x-axis.

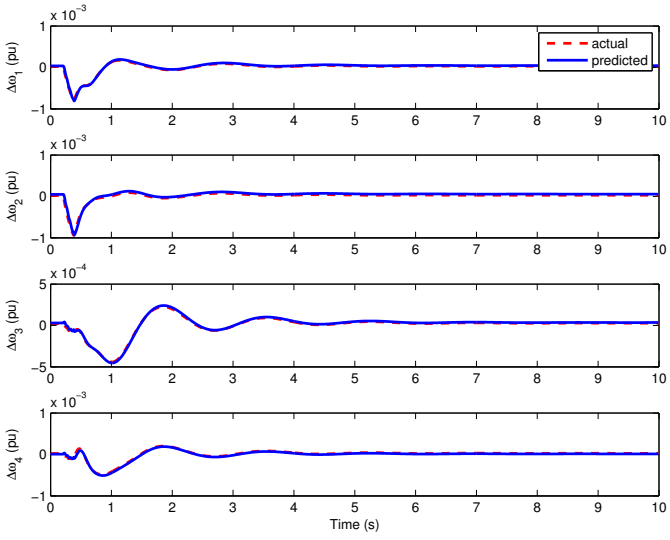

Fig. 14. Testing output of CMLP for operating point IV (Test System II). Speed deviation of the generators is shown in y-axis against time in x-axis.

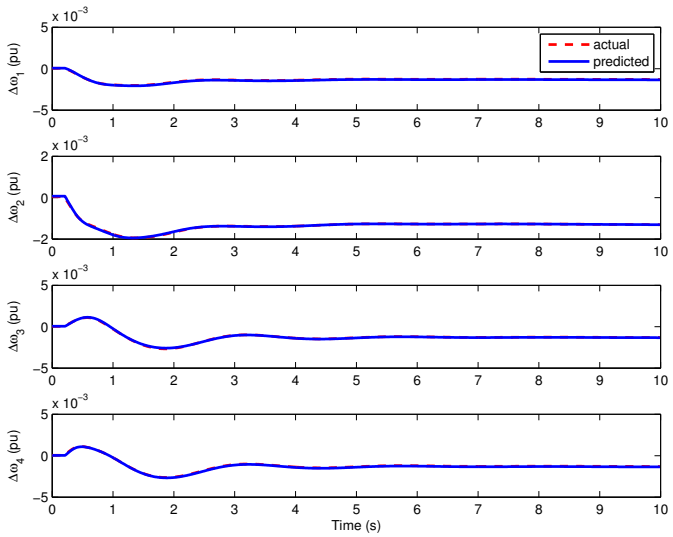

Fig. 15. Testing output of CMLP for operating point V (Test System II). Speed deviation of the generators is shown in y-axis against time in x-axis.

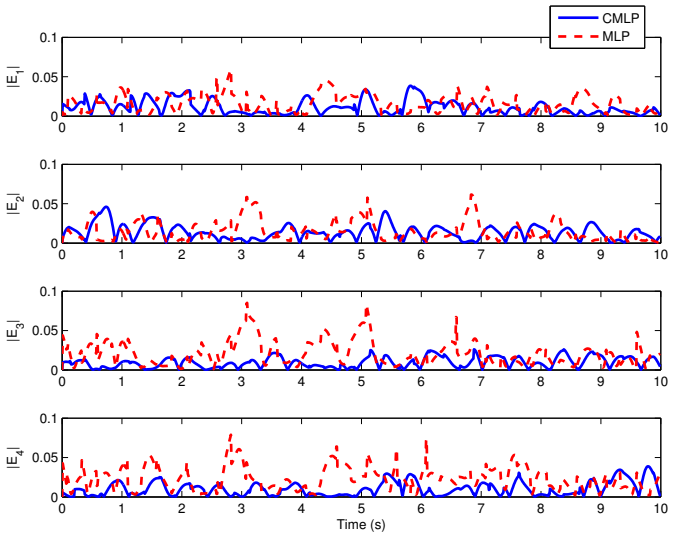

Fig. 16. Comparison of absolute errors obtained by MLP vs. CMLP for operating point I (Test System II).

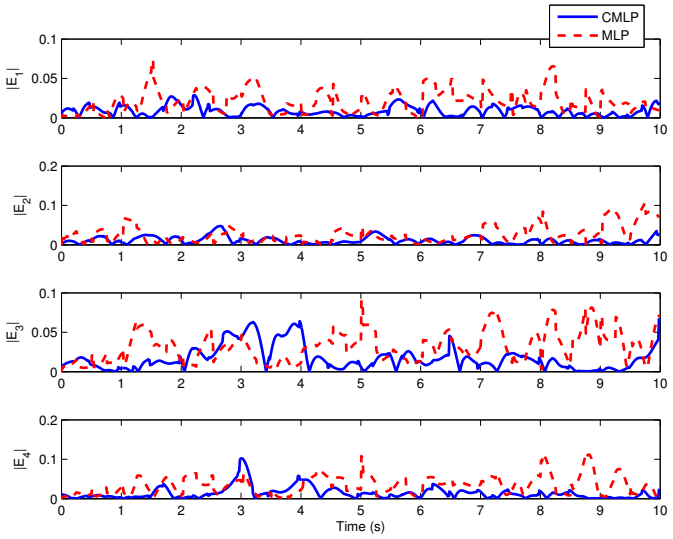

Fig. 17. Comparison of absolute errors obtained by MLP vs. CMLP for operating point II (Test System II).

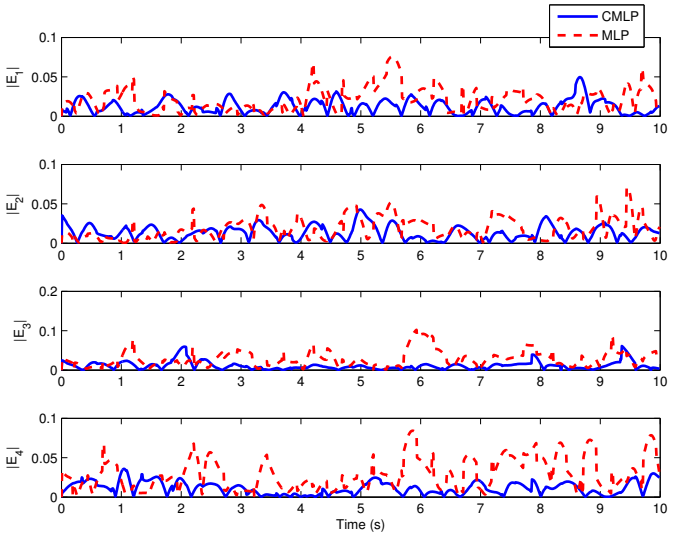

Fig. 18. Comparison of absolute errors obtained by MLP vs. CMLP for operating point III (Test System II).

the CNN during training. Moreover, the advantage of CMLP comes from its ability to scale up to a much larger system without significant impact on the performance. When the size of the network grows, the number of cells increases but the size of an MLP on each cell remains the same (as long as the nearest-n topology remains the same). For a CMLP with m cells with each cell having an MLP with N weights, the total number of weights in the network is mN. This is in contrast to a MIMO MLP where the number of neurons in the hidden layer needs to be increased significantly in order to obtain a satisfactory performance when the number of inputs and outputs increases. This causes the number of weights in an MLP to increase drastically as the size of the network grows, thus increasing the computational complexity leading to poor training and testing results. This is evident from even these two simple test systems where the number of weights in MLP increased from 90 to 180 between Test System I and II where as that of CMLP increased from 90 to 120. However, sequential training of CMLP takes a long time and can only be justified if implemented on a parallel hardware and/or software platforms such as general purpose GPU clusters, FPGAs or shared memory architectures.

The tabulated data also show that the performance of CMLP is better than that of MLP for both Test Systems I and II. The lower values of average MAE show better performance of CMLP over MLP and the lower values of standard deviation show its consistency in maintaining that

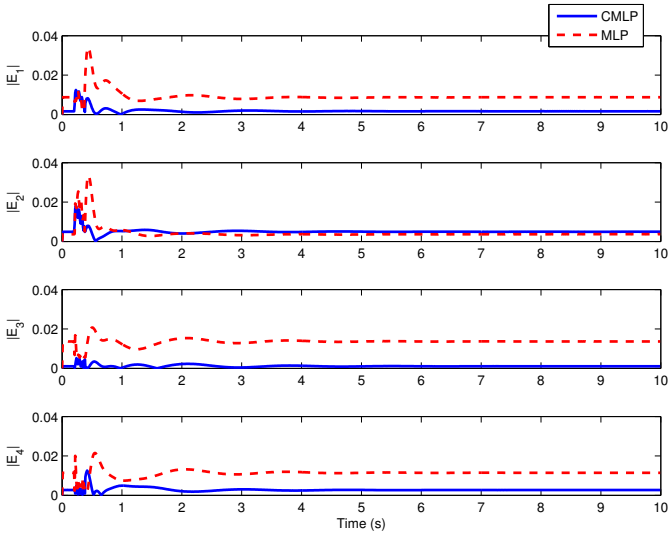

Fig. 19. Comparison of absolute errors obtained by MLP vs. CMLP for operating point IV (Test System II).

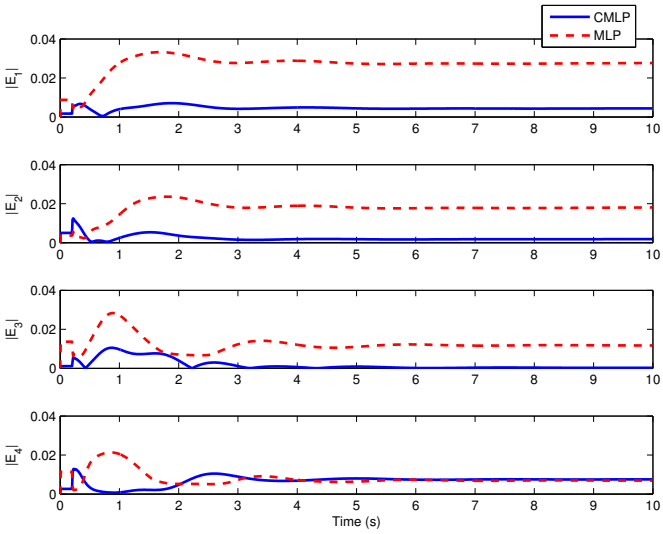

Fig. 20. Comparison of absolute errors obtained by MLP vs. CMLP for operating point V (Test System II).

performance. The 'winner' row in the tables is to quantitatively express the results for assessing which NN architecture has better performance. The lower values of average MAE is given priority in deciding the winning architecture for each output. If the difference of average MAE is less than 5% of each other, then it is considered as a tie and hence the standard deviation is considered to be the tie-breaker. For Test System I, CMLP has a better performance than MLP on all the outputs. For Test System II, CMLP has better performance in four operating points for outputs G1, G2 and G3 and in all five operating points for output G4.

V. Conclusion

In this paper, CMLP based state predictors are used for implementing wide area monitor for a multi-machine power system. The CMLP is developed using an MLP on each cell that represents a generator of the power system. The different cells of the CMLP are interconnected using 'nearest-n neighbors' topology so that the size of MLP in each cell is reduced. This ensures that the complexity of the CMLP increases linearly with the size of the power system being implemented. Therefore, the proposed architecture becomes highly scalable. Results obtained by this approach are presented and are shown to be comparable to or better than implementation of WAM using a single MIMO MLP structure in terms of performance as well as number of weights. However, sequential training of a CMLP takes longer than an MLP and is a challenging problem, especially when the size of the network grows. Therefore, its implementation can only be justified in parallel hardware and/or software platforms. Implementation of CMLP on a larger power system and its implementation on GPU cluster will be the authors' future work in this area.

References

[1] Y. Xue, "Some viewpoints and experiences on wide area measurement systems and wide area control systems," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008, pp. 1–6.

[2] I. Kamwa, J. Béland, G. Trudel, R. Grondin, C. Lafond, and D. McNabb, "Wide-area monitoring and control at Hydro-Québec: Past, present and future," in *Power Engineering Society General Meeting, 2006. IEEE*. IEEE, 2006, p. 12.

[3] M. Zima, M. Larsson, P. Korba, C. Rehtanz, and G. Andersson, "Design aspects for wide-area monitoring and control systems," *Proceedings of the IEEE*, vol. 93, no. 5, pp. 980 –996, May 2005.

[4] S. Meliopoulos, G. Cokkinides, R. Huang, E. Farantatos, S. Choi, and Y. Lee, "Wide area dynamic monitoring and stability controls," in *Bulk Power System Dynamics and Control (iREP) - VIII (iREP), 2010 iREP Symposium*, 2010, pp. 1–8.

[5] G. K. Venayagamoorthy, "Potentials and promises of computational intelligence for smart grids," in *Proc. IEEE Power & Energy Society General Meeting (PES '09)*, 26–30 July 2009, pp. 1–6.

[6] S. Ray and G. Venayagamoorthy, "Real-time implementation of a measurement-based adaptive wide-area control system considering communication delays," *IET Generation Transmission and Distribution*, vol. 2, no. 1, pp. 62–70, 2008.

[7] G. Venayagamoorthy, "Online design of an Echo State Network based wide area monitor for a multimachine power system," *Neural Networks*, vol. 20, no. 3, pp. 404–413, 2007.

[8] S. Mohagheghi, G. Venayagamoorthy, and R. Harley, "Optimal wide area controller and state predictor for a power system," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 693–705, 2007.

[9] S. Jiang, U. Annakkage, and A. Gole, "A platform for validation of facts models," *Power Delivery, IEEE Transactions on*, vol. 21, no. 1, pp. 484 – 491, 2006.

[10] M. Klein, G. Rogers, and P. Kundur, "A fundamental study of inter-area oscillations in power systems," *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 914–921, 1991.

[11] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550 –1560, oct. 1990.

[12] RTDS, "Real time digital simulator tutorial manual (rscad version)," RTDS Technologies, March 2008.