Complex Valued Open Recurrent Neural Network for Power Transformer Modeling

A. Minin, Yu. Chistyakov, E. Kholodova, H.-G. Zimmermann and A. Knoll

Abstract**—**Application of artificial Neural Networks (NN) for power equipment modeling has been outlined in many papers, but conventional NN deal only real valued numbers. Complex Valued Open Recurrent Neural Networks (CVORNN) is significant expansion of NN application due to complex numbers usage, which is natural form of data in electrical engineering. Dealing with complex values is very useful in Wide Area Measurement Systems. Power transformer modeling using CVORNN is presented in this paper as it is one of the most common elements of power grid which has nonlinear behavior. Considering paper consists of two main parts: conventional modeling for getting transformer data and CVORNN application for studied transformer. First part of the paper describes nonlinearities which take place in transformer model: nonlinear magnetizing system, ambient temperature influence on windings and OLTC voltage stabilization. Typical day-long load curve is used for load simulation. The second part devoted to CVORNN basics and application. CVORNN is trained and tested using received in the first part data. Obtained results show that CVORNN is promising and convenient method for modeling in electrical engineering.

Keywords**—**Complex Valued Open Recurrent neural network, Transformer modeling, Power equipment modeling, On-load tap changer, Complex Valued Back Propagation.

I. Introduction

Detailed models of electrical power network elements are necessary for receiving accurate simulation data. In analytical models it is usually difficult to take into account all the factors that have influence on the power equipment. An approach which allows to build a model which may track a lot of real phenomena in dynamic regimes is modeling using neural networks. In [1], [2] modeling of a transformer with Real Valued Neural Networks (RVNN) is described. Such networks are also applied for the task of differential protection and transformer insulation modeling [3], [4]. The approach generally consists of two parts: firstly one generates data with analytical model; then a neural network is trained and tested. Comparison with analytical results shows applicability of an approach.

In the present paper the modeling of a power transformer system with nonlinearities (due to On-Load Tap Changer) is proposed as an extension of [5], using the approach based on Complex Valued Open Recurrent Neural Networks (CVORNN). As its name implies the difference from real-valued ones is in intrinsic capability to deal with the complex numbers instead of the real ones. Taking into account the common representation of electrical values as rotating vectors, which can be treated as complex numbers, this feature is useful and promising in frame of power grid elements modeling.

Like in approaches using RVNN, on the basis of analytical model simulation data, CVORNN-based model work is shown. Conventional method is described in the first part of the paper: temperature influence, on-load work, tap changer and result of the simulation by itself. The second part deals with CVORNN principles. Then, having simulated data from the analytical model, the CVORNN is trained and tested. The results are explained using various statistical measures.

II. Analytical Transformer Modeling

Well-known basic equations of a power transformer, supplemented by terms representing a nonlinear magnetizing system, influence of an ambient temperature and On-Load Tap Changer (OLTC) voltage stabilization system are considered to be an analytical model of a power transformer system in the paper. The system is considered to work on some changing load which corresponds to the evening load peak. Basic equations and nonlinear part are presented here.

A. Basic transformer principles and equations

In a basic transformer one of the windings, named a primary winding, is energized by an external voltage source. The alternating current flowing through a primary winding creates a variable magnetic flux in magnetic core. The variable magnetic flux in magnetic core creates electromotive force (EMF) in all windings, including primary. When current is sinusoidal absolute value of EMF is equal to the first derivative of a magnetic flux. EMF induces current in the secondary winding. Ideal transformer without losses is shown

Manuscript received December 18, 2011.

A. Minin is with Corporate Technology Department, Siemens LLC, Volynskiy lane 3, 191186, St. Petersburg, Russia also with the Dept. of Robotics and Embedded Systems at Technical University of Munich (e-mail: alexey.minin@siemens.com).

Yu. Chistyakov is with the Corporate Technology Department, Siemens LLC, Volynskiy lane 3, 191186, St. Petersburg, Russia (e-mail: yury.chistyakov@siemens.com).

E. Kholodova is with the Corporate Technology Department, Siemens LLC, Dubininskaya str. 96, 115093, Moscow, Russia (e-mail: elena.kholodova@siemens.com).

Dr. H.-G. Zimmermann is with Corporate Technology Department, Siemens AG, D-81730 Muenchen, Mch-P, R. 53.220 (e-mail: hans_georg.zimmermann@siemens.com).

Prof. Dr. Alois Knoll is with Technical University of Munich, head of Robotics and Embedded Systems department (e-mail: knoll@in.tum.de).

in Fig. 1.

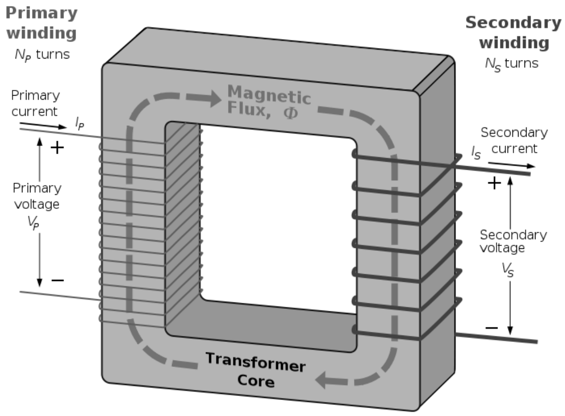

Fig. 1. Ideal transformer [6]. Electric and magnetic circuits are depicted. No losses in windings assumed on the picture.

Equivalent circuit of generic transformer is shown in Fig. 2. Power losses are represented as resistances R_1 (primary) and R_2 (secondary), flux leakage – as reactances X_1 (primary) and X_2 (secondary). Iron losses caused by hysteresis and eddy currents in the core are proportional to the core flux and thus to the applied voltage. Therefore they can be represented by resistance R_m. To maintain the mutual flux in the core magnetizing current I_μ is required. Magnetizing current is in phase with the flux. Since the supply is sinusoidal, the core flux lags the induced EMF by 90° can be modeled as a magnetizing reactance X_m in parallel with the resistance R_m. R_m together with X_m are called magnetizing branch of the model. In case of open-circuit, current I_0 represents the transformer's no load current [7],[8].

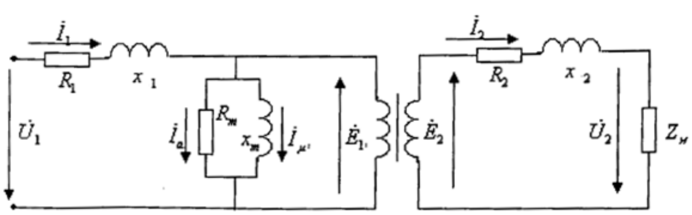

Fig. 2. Equivalent circuit of a transformer [8, p.53]

Analysis of circuit significantly simplifies if the circuit with magnetically connected windings will be replaced by an equivalent circuit, elements of which are electrically connected with each other (see Fig. 3). Here the number of turns in primary (N_1) and secondary (N_2) is equal, so the parameters of the transformer have to be changed in order to maintain all energy relations. The secondary winding is moved (or "referred") to the primary side utilizing the scaling factor:

$$N^2 = \left(\frac{N_1}{N_2}\right)^2 \tag{1}$$

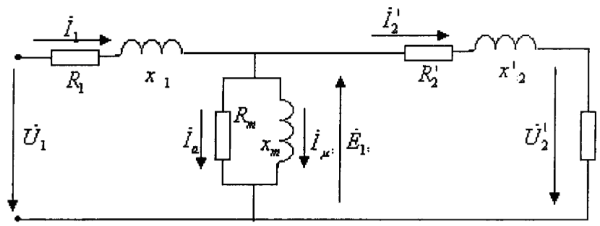

Fig. 3 Equivalent circuit of a transformer referred to the primary winding [2, p.54]

Finally, transformer equations can be written as follows [7]:

$$\begin{cases} U_1 = E_1 + I_1(R_1 + jX_1) = E_1 + I_1 Z_1 \\ E_1 = R_2' I_2' + jX_2' I_2' + U_2' \\ I_1 = I_0 + I_2' \end{cases} \tag{2}$$

where $U_1, E_1, I_1, R_1, X_1, Z_1$ – primary winding voltage, EMF, current, resistance, reactance and impedance, respectively. Secondary winding is described with similar values, but already referred to the first winding:

$$\begin{aligned} U_2' &= U_2 N, \quad I_2' = \frac{I_2}{N} \\ R_2' &= R_2 N^2, \quad X_2' = X_2 N^2 \end{aligned} \tag{3}$$

B. Transformer's parameters

Given transformer model is based on the transformer data of Russian transformer OMP-10/10 [9].

Table 1. Transformer Parameters

Parameter	Symbol	Value	Unit
Nominal power	S	10	kVA
Primary winding voltage	U_1, U_{hv}	10	kV
Frequency	f	50	Hz
Secondary winding voltage	U_2, U_{lv}	230	V
No-load current	I_{nl}	4.2	%
No-load power	P_{nl}	60	W
Short circuit power	P_{sc}	280	W
Short circuit voltage	U_{sc}	3.8	%

The detailed calculation of the equivalent circuit parameters is given below.

Using factory data from short circuit and no load tests (see Table 1) other transformer parameters were calculated:

Primary winding:

$$\begin{aligned} Z_1 &= \frac{U_{sc} \cdot U_1^2}{100 \cdot S} = \frac{3.8 \cdot 10000^2}{100 \cdot 10000} = 380\ Ohm \\ R_1 &= \frac{P_{nl} \cdot U_1^2}{S^2} = \frac{60 \cdot 10000^2}{10000^2} = 60\ Ohm \\ X_l &= \sqrt{(Z_1^2 - R_1^2)} = \sqrt{(3802 - 602)} = 375\ Ohm \end{aligned} \tag{4}$$

Secondary winding:

$$Z_2 = \frac{U_{sc} \cdot U_2^2}{100 \cdot S} = \frac{3.8 \cdot 230^2}{100 \cdot 10000} = 0.201\ Ohm$$

$$R_2 = \frac{(P_{nl} \cdot U_2^2)}{S^2} = \frac{60 \cdot 2302}{10000^2} = 0.032\ Ohm \qquad (5)$$

$$X_2 = \sqrt{Z_{22} - R_{22}} = \sqrt{0.2012 - 0.0322} = 0.198\ Ohm$$

$$I_0 = I_a + I_\mu = \frac{S \cdot I_{nl}}{U_1 \cdot 100} = \frac{10000 \cdot 4.2}{10000 \cdot 100} = 0.042\ A$$

Other values:

$$\cos\phi_0 = \frac{P_{nl}}{U_1 \cdot I_0} = \frac{60}{10000 \cdot 0.042} = 0.143$$

$$R_0 = \frac{S}{U_1 \cdot I_0 \cdot \cos\phi_0} = \frac{10000}{10000 \cdot 0.042 \cdot 0.143} = 166.5 \qquad (6)$$

$$X_0 = \frac{S}{U_1 \cdot I_0 \cdot \sin\phi_0} = \frac{10000}{10000 \cdot 0.042 \cdot 0.143} = 14.06$$

C. *Influence of ambient temperature on the transformer*

In order to get more realistic model, windings' resistance dependence from temperature have been introduced:

$$R = R_{nom}(1 + \alpha(T - 20)) \qquad (7)$$

where R is calculated winding resistance

R_{nom} – nominal winding resistance

α – temperature coefficient

T – temperature

Transformer windings are assumed to be made from copper and corresponding temperature coefficient $\alpha = 3.8 \cdot 10^{-3}\ K^{-1}$ is used.

Temperature variation is assumed as in Fig. 5 within 12 hours time range from 12:00 till 24:00 (from 0:00 p. m. till 12:00 p. m.). Sun peak happens in the daytime, then the temperature decreases due to weather change for the worse.

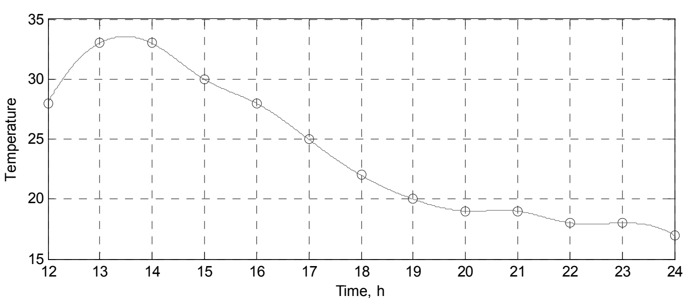

Fig. 4. Temperature variation

D. *Load curve*

Implemented transformer model works on some specified load which should be treated as equivalent impedance of some power system, supplied by the transformer. Peak on the load curve (Fig. 5) corresponds to evening peak of a typical household load. For being more realistic, small noise to the load profile is added.

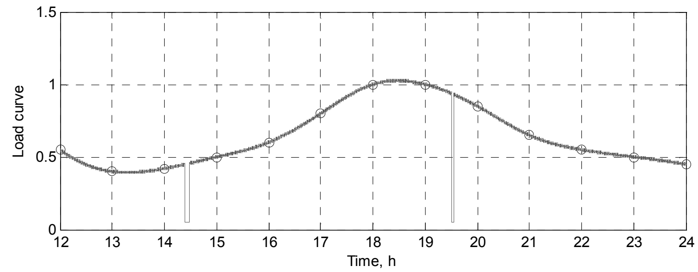

Fig. 5. Load variation [10, p. 47]

Moreover, in some points of simulation models of faults are added. Introduced extreme regimes help to test OLTC stabilization system and ability of CVNN to handle with such nonlinear data.

E. *On-load tap changer*

The transformer under consideration is equipped with On-Load Tap Changer (OLTC) mechanism on the primary winding. Range of voltage variation is equal to $\pm 15\%$ with 2.5% step.

F. *Simulation results*

Analytical modeling is carried out in MATLAB, where all mentioned above peculiarities are implemented.

Transformer is assumed to work with nominal input AC voltage, having voltage output equals to 230 V (RMS) consequently.

Simple voltage control algorithm with OLTC is applied. The aim is to keep secondary voltage on the nominal level in spite of load and temperature fluctuations. During the simulation RMS value of secondary voltage $U_{2\mathrm{RMS}}$ is calculated over each electrical cycle using integral formula. Then it is compared with predefined quality margins (220 and 240 V) and corresponding control action (OLTC switching) is undertaken.

Results, obtained from the simulation are presented in Fig. 6, where all voltages, currents, temperature, load and OLTC position are presented. Fig.7 shows $U_{2\mathrm{RMS}}$ voltage value during the simulation.

As it can be seen, temperature variation has low influence on the secondary current. At the same time, load variation has significant influence on the results being the main reason for OLTC switchings.

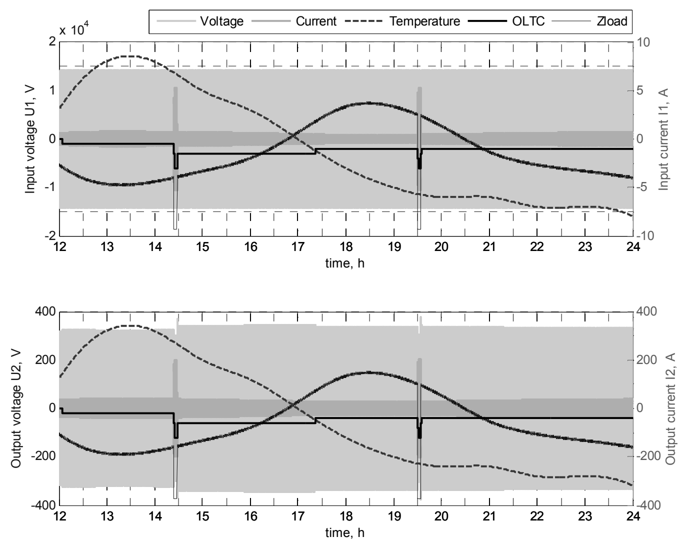

Fig. 6. Results of the simulation. Because of introduced variation of load, temperature and OLTC control currents and secondary voltage vary in time.

The main aim of the conducted analytical modeling is to generate data of the transformer with nonlinearities in order to train and test complex valued neural network in the next section. Time scaling has been used (12 hours is represented by 60 seconds of power device simulation). Such an assumption makes the task of CVNN even harder since the change of loads and temperature happens slower in reality. See next sections for details.

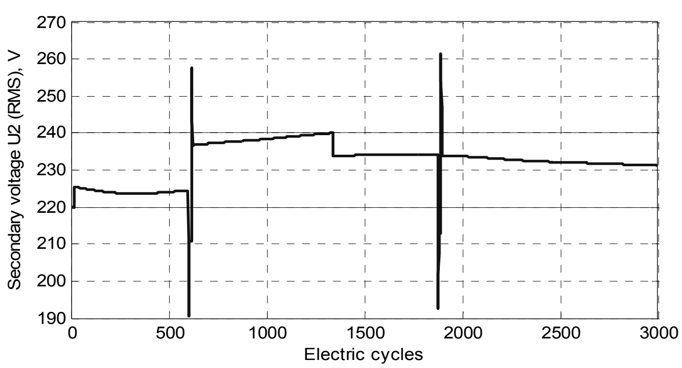

Fig. 7. Modeled OLTC voltage control increases quality of supply facilitating to keep secondary voltage in defined margins (220 and 240 V). Margins violation happens in case of heavy faults because of limited OLTC switching range.

III. Complex Valued Neural Networks

Complex Valued Neural Network (CVNN) (see [11] and [12]), an essential extension of a traditional real-valued neural network, is a universal approximator of data, where inputs of network, weights and transition functions are from the complex values domain.

The basics of CVNN are discussed in the paper. The task for the NN is to find the mapping from inputs into outputs ($\mathbb{C} \to \mathbb{C}$) so that the selected inputs propagated through the neural network can lead to the set of expected values (targets). The CVNN can be trained with nearly the same methods which are used for the Real Valued NN with some modifications which we will discuss below.

A. Feed Forward Path

The feed forward path is the same for the CVNN as it is discussed in many papers for the Real-Valued Neural Network (RVNN) (see [13] and [14]). It is shown with gray arrows at the Fig. 8. Network inputs propagate through the input layer (netin0 in Fig. 8) of the network, and then go to the first hidden layer as its inputs (*netout$_0$*). Then the inputs are to be multiplied with the weights matrix W_1 which consists of complex numbers. After this linear algebra operation a transition function f should be applied. The procedure is repeated iteratively. After the information goes out of the network (*netout$_2$*) the network output – *observation* has to be compared against *expectation* or *target* (see Fig. 8) using the error function. The difference between RVNN and CVNN starts when one is trying to calculate the approximation error.

B. Error function discussion

The approximation error can be defined as follows, see [17]:

$$E = \frac{1}{T}\sum_{t=1}^{T}\Big(y_t\left(w,x\right) - \widehat{y_t}\Big)\overline{\Big(y_t\left(w,x\right) - \widehat{y_t}\Big)} \to \min_{w} \quad (8)$$

where the bar above the values means complex value conjunction (changes the sign of the imaginary unit). This error is a non analytical function, which means that there is no derivative defined. The definition of the analyticity of the function can be given with the Cauchy-Riemann conditions [17] which are not fulfilled by our error function. Moreover it is not needed, that out error function makes a mapping from $\mathbb{C} \to \mathbb{C}$ since at the complex plane we cannot formulate minimization task. It is needed that concerned error function does the mapping from $\mathbb{C} \to \mathbb{R}$. Then it has a Taylor expansion can be as follows(9):

$$E(w + \Delta w) = E(w) - \eta g^T \overline{g} + \frac{\eta^2}{2} g^T G \overline{g} \quad (9)$$

In case (9) exists minimization of the error function can be done. This is true if:

$$\Delta w = -\eta \cdot \overline{g},\ \overline{g} = \overline{\frac{dE}{dw}} \quad (10)$$

where η is the so called learning rate – a real valued constant, which serves the following needs. Taking it relatively small all the members of the Taylor expansion of the order different from one can be ignored. Note that gradients conjunction is very important due to the need in the existence of the Taylor expansion for the network training.

Thus by applying the training rule (10) the existence of Taylor expansion (9) is guaranteed and a minimization of the error function(8) can be done. Doing the iterative changes of the weights according to (10) one can train the CVNN to reproduce the expectation values having only the input information.

To calculate the gradients, mentioned in(10), one should use the so called Complex Valued Error Back-Propagation (CVEBP). Calculation of the gradients can be efficiently utilized with the error back propagation algorithm presented at the Fig. 8. Calculation of the CVEBP is to be done during the backward path.

C. Backward Path

Here, in order to calculate all partial derivatives of the error with respect to the weight one has to propagate the derivative of the error, which is in our case:

$$dE = \left(y - \hat{y}\right) \tag{11}$$

back through the network till the input layer (first proposed in [18]). The back propagation calculations are marked with black arrows at Fig. 8.

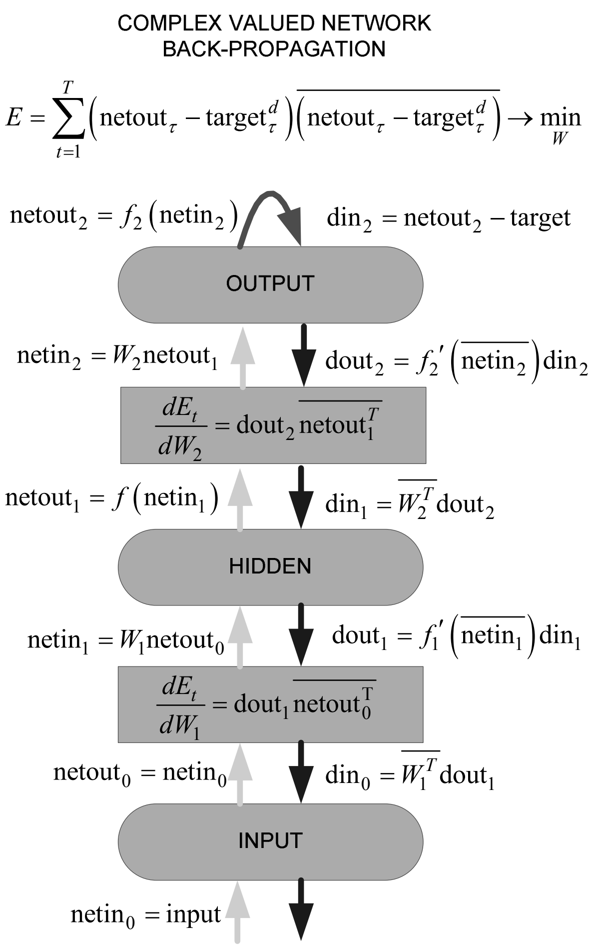

Fig. 8. Complex Valued Back-Propagation. Notations: *netin* - layer inputs, *netout* - layer outputs, *din* -layer derivative input, *dout* -layer derivative output, W_i are network weights, arrows show the information flow. The figure depicts the locality of the BP algorithm and independence of the BP from the network architecture.

One can see from the Fig. 8 that the back propagation for the CVNN has changes related to values conjunctions. These conjunctions appear due to the Wirtinger calculus used to calculate the derivatives of the errors (see [16] and [17]).

The Wirtinger derivative can be defined in the following way:

$$\begin{cases} \frac{\partial E}{\partial z} \triangleq \frac{1}{2}\left(\frac{\partial E}{\partial a} - i\frac{\partial E}{\partial b}\right), z = a + ib \\ \frac{\partial E}{\partial \overline{z}} \triangleq \frac{1}{2}\left(\frac{\partial E}{\partial a} + i\frac{\partial E}{\partial b}\right), \overline{z} = a - ib \end{cases} \tag{12}$$

Using such approach, by considering the functions which are doing the mapping from $\mathbb{R}^2 \to \mathbb{C}$ the derivatives for error function as well as for transition functions *f* (see Fig. 8) can be calculated.

D. Transition functions discussion

The transition functions is a problem in case of CVNN since there is a Liouville theorem, which says, that in case the functions is analytic (which means it has a derivative defined) over the whole complex plane and is bounded it is a constant over the whole complex plane. This theorem prohibits analytic and bounded functions which is bad for neural networks since these are the main requirements to the transition functions. Nevertheless this problem can be overcome by using the functions which are not analytical in the Liouville sense, but still have a Wirtinger derivative defined, these are the so called engineered functions:

$$\left| \begin{array}{l} f(z) = \tanh(z) + i\tanh(z) \\ f(z) = \tanh(z) \qquad\qquad , \\ f(z) = \tanh(r)e^{i\phi} \\ \phi = \arctan\left(\frac{\mathrm{Im}(z)}{\mathrm{Re}(z)}\right) \\ r = \sqrt{\mathrm{Re}(z)^2 + \mathrm{Im}(z)^2} \end{array} \right. \tag{13}$$

Some of functions from (13) are analytic in Wirtinger sense and are bounded, some are non analytic but we can use the normal derivative taking into account the unboundness of it. One can still use normal functions like *sin, tanh, log* but take into account, that these functions have singular points and are not bounded which can be rather dangerous for numerical stability of computations. Practical experiments have showed that usage of analytic functions is more appropriate, but can cause numerical explosions of the calculations. Therefore a combination of functions is more suitable from a computational point of view.

E. Recurrent Open Architecture Neural Network discussion

In the current paragraph the so called Complex Valued Open Recurrent Neural Network (CVORNN) is going to be discussed. The network aims to describe the state space approach for modeling of the systems with memory; the example is displayed in Fig. 9 below:

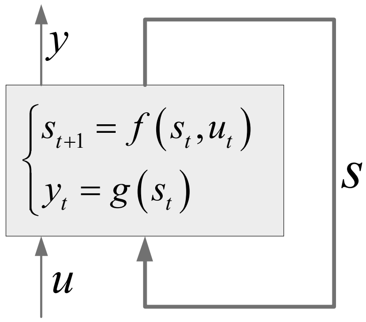

Fig. 9. Dynamic system which describes the transformer.

This system is developing to its new state s_{t+1} through the non linear function f with its external inputs u_t and its current state s_t. In order to produce the outputs the system takes its current state s_t and applies a non linear function g to it. This system can be used for transformer modeling since its dynamics is driven by the external inputs, like input voltage and current and its internal state. We can easily translate the description given above to the neural network case, see Fig. 10 below:

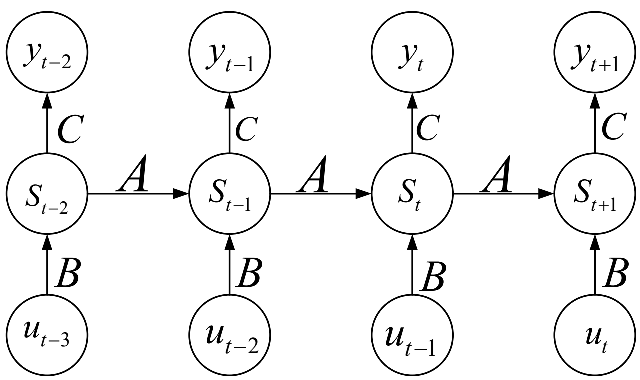

Fig. 10. Complex Valued Open Recurrent Neural Network. *A*,*B* and *C* are weights matrixes, *u* – are external drives (inputs), *s* – are internal states and *y* – are outputs (observables). Arrows show the information flow.

The network in Fig. 10 is recurrent through its weights. The weights *A, B* and *C* are the so called shared weights proposed by Zimmermann [15]. This means that *A, B* and *C* are always the same at each iteration of the network (in Fig. 8 it would mean that $W_1=W_2$). The network is called open since it has external drives. One can see that such network is perfectly suited for the time series modeling. Here time series obtained from the modeling were putted to the input side of the network (I_1, U_1, T, Z_{load}) and at the output side U_2 and I_2 were putted. Then the training of the network with the Complex Valued Gradient Descent minimization discussed above has been done. To obtain the prediction of the U_2 and I_2 we just continue the feed forward path of the network by iterative application of matrix *A*, *B* and *C* to the states and inputs respectively. In case the system is autonomous (develops on its own) then *B=0,* such architecture is called Complex Valued Closed Recurrent Neural Network (CVCRNN). Moreover in case one will apply the Teacher Forcing to train such architecture he will end up with the so called Historical Consistent Neural Network. This network is very interesting due to many nice properties it has. A more detailed description of this network can be found in paper [16]. Here we just note that HCNN can be used for "black box" modeling of the dynamical systems in case the only available information is outputs of the dynamical system. We do not apply this network, since in our problem we have the inputs and the outputs of the transformer. In case we would only have the outputs (like it would be for the Generator) we would use the Complex Valued Historical Consistent Neural Network. Note that we use a very long neural networks where the amount of states is equal to the size of data we have, e.g. if we have 1000 data points, the amount of states in the network will be equal to 1000.

F. NN Training for the Given Problem

The inputs for the network are the following parameters: input current, input voltage, load and temperature, the outputs (desired values) which are interested here are the output current (I_2) and the output voltage (U_2). According to the transformer model described above a set of *60000* patterns is generated. One can see that there are two peaks at the generated data, which correspond to the faults in a grid. For the training *2000* data points around the first peak (this is so called Training set) were taken. Then *1000* data points around the second peak were taken to see how well the network generalizes (the so called Validation set). For this experiment the network had *50* state neurons, *5* inputs and *2* outputs. Learning rate has been chosen to be $\eta = 0.01$. The amount of epochs for training is equal to *2000.* One epoch of training means the network has been applied in forward direction, derivative of the error has been calculated and back propagated. Then gradients have been calculated and weights matrixes *A, B* and *C* have been updated. *1000* epochs means that this sequence has been done for *1000* times.

In Fig. 11 below one can see the error decay during the training:

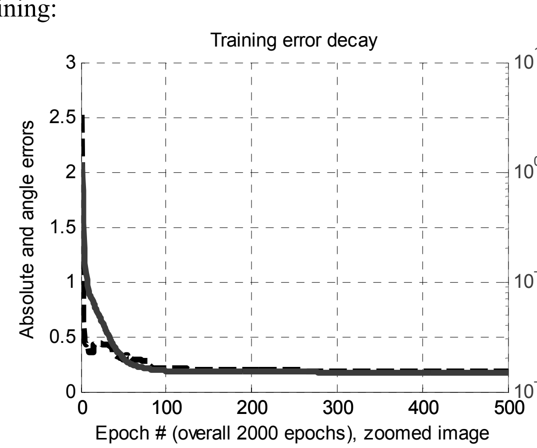

Fig. 11. Error decay for the absolute part of the error function and for the angle part of the error function.

The following statistics for the training set are used to check the quality of data approximation:

- Root mean squared error (*RMSE*)
- Correlation coefficient (*R*)
- Determination coefficient (R^2)

The results of modeling are shown in Fig. 12, where time characterizes time steps at which measurements are made. The statistics for the training set is not presented since it is nearly ideal as it should be at the training set. Here one can see that expectation values almost coincide with the observation ones and only differ slightly at the bump area; statistical coefficients are also close to their corresponding best values (Table 2). Fig. 13 and Fig. 14 show Validation set results for the whole time period and for the specified area with a leap of current I_2 and voltage U_2.

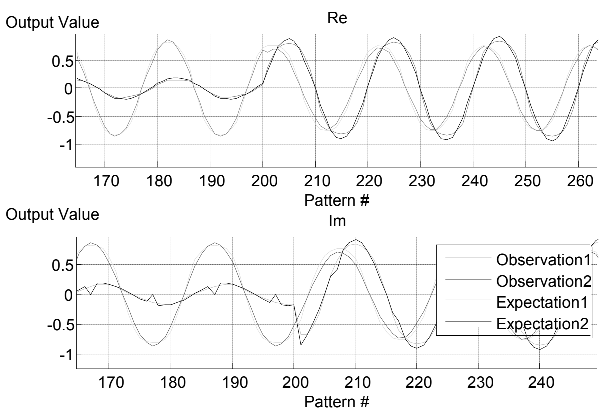

Fig. 12. Results of the transformer modeling for the subset of data containing leaps. One can see the real part of the network outputs and the actual values of I_2, U_2 on the training set. Zoomed image, since otherwise one cannot see the short circuit moment. One can see that network reacted perfectly, but this is training set.

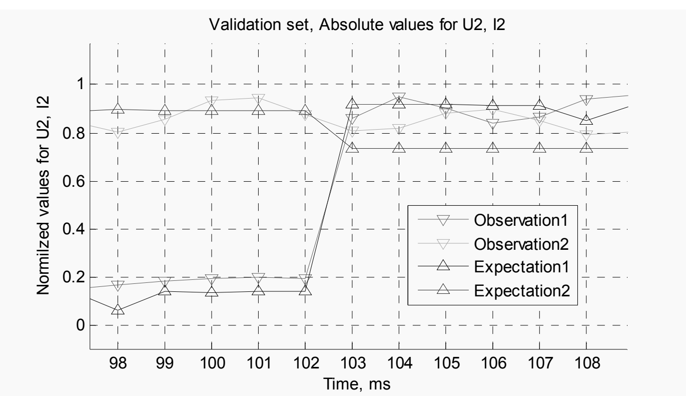

Fig. 13. Results of the transformer modeling for the subset of data containing leaps. One can see the real part of the network outputs and the actual values of I_2, U_2 on the Validation set. Zoomed region to see the short circuit moment better. Absolute values for voltage and current.

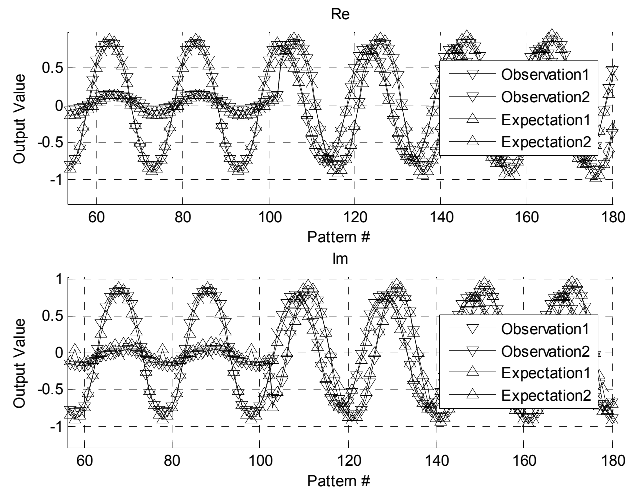

Fig. 14. Results of the transformer modeling for the subset of data containing leaps. One can see the real part of the network outputs and the actual values of I_2, U_2 on the Validation set. Zoomed region to see the short circuit moment better. Real and Imaginary values for voltage and current.

One can see the moment where a fault is happened (region between the 100 and 104 patterns at the Fig.14 and Fig. 13). Now we will zoom this region to see it better (see Fig. 15). One can see that network was able to predict such change of the sin like function of the current and voltage. This also makes a perfect statistics which is available at the Table 2. Such results are only possible with the CVORNN network due to the following reasons: the system under the modeling us recurrent, since it has memory e.g. the current state of the transformer depends on the previous states of the transformer. Another reason is that one cannot model the transformer with the real valued network in the same way we did, since one will have to separate the real and imaginary parts of the current and voltage and feed this information as a separate inputs. In this case a hypothesis was taken, that imaginary part and real part of the complex inputs are independent and do not interact while neural network processing which is not the case to our point of view. The complex valued signals interact inside the neural network which makes the modeling more physical since now we model waves interaction inside the complex weights of the neural network. Such approach can be considered as “back to analogous” modeling. Using the complex valued networks we can model such things like interference of the waves etc. The presented results can be easily extended for other electrical elements modeling like Phasors, Generators, Drives etc.

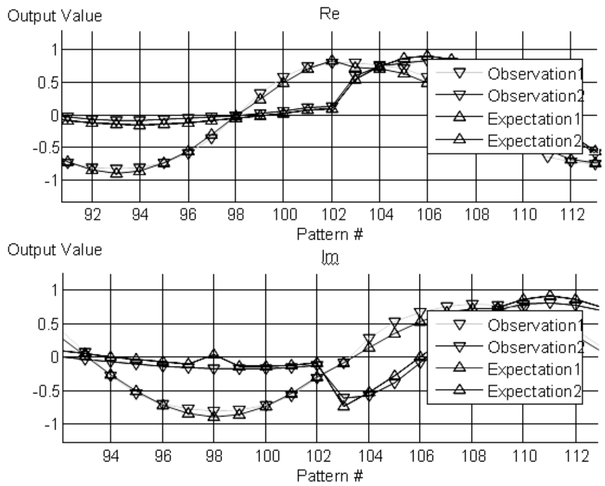

Fig. 15. Results of the transformer modeling for the subset of data containing leaps. One can see the real part of the network outputs and the actual values of I_2, U_2 on the Validation set. Strongly zoomed region to see the short circuit moment better. Real and Imaginary values for voltage and current.

Table 2. Quality of testing

Real/Imaginary parts	*RMSE*	*R*	R^2
I_2	$6\cdot10^{-3}/8\cdot10^{-3}$	0.99/0.99	0.98/0.97
U_2	$2\cdot10^{-3}/2\cdot10^{-3}$	0.99/0.99	0.99/0.98

Obtained results show general capability of CVORNN to deal with complex values of electrical power tasks. Research has to be further developed for proving ability to be used in electrical engineering software.

IV. Conclusion

The work of a transformer is modeled with a complex valued neural network. Data for testing and training of a CVNN have been generated; the neural network has been trained and tested. Injection of nonlinearities and adding noise in the analytical model for generating data made the model more realistic in comparison with previous research [5]. Two different architectures were compared on the transformer data.

From obtained results the following conclusions can be formulated:

- Statistical data (Table 2) show that quality of validation is rather high and Open Recurrent architecture gives good results.
- Developed method allows handling of unexpected events such as faults with rather high accuracy.
- CVRONN method allows representing power variables in "natural" complex form for handling data coming from Wide Area Measurement Systems.

The attractive feature is the possibility to model each grid device individually, just teaching the CVORNN with measured data from particular device.

Significant end-use of the approach could consist in integration of obtained CVRONN -based transformer model in power engineering simulation software packages after further development and training on the data from real device.

References

[1] Minchev, S.V., Neural networks for modeling of dynamic systems with hysteresis. B Proceedings First International IEEE Symposium Intelligent Systems. IS'02: First International IEEE Symposium "Intelligent Systems". Varna, Bulgaria, c. 42-47.

[2] Minchev, S.V., & Venkov, G.I., Modeling of nonlinear dynamic systems via discrete-time recurrent neural networks and variational training algorithm. 2004 2nd International IEEE Conference on "Intelligent Systems". Proceedings. Varna, Bulgaria, c. 105-108.

[3] J. Pihler "Learning of Open-Loop Neural Networks with Improved Error Backpropagation Methods", International conference on Circuits, Systems, Signals, Malta, 2010.

[4] L. Ekonomou, P.D. Skafidas, D.S. Oikonomou "Transformer oil's service life identification using neural networks A cascade of artificial neural networks to predict transformers oil parameters", 8th WSEAS International Conference on Electric Power Systems, High Voltages, Electric Machines (Power '08).

[5] Yury S. Chistyakov, Elena V. Kholodova, Alexey S. Minin, Hans-Georg Zimmermann, Alois Knoll "Modeling of electric power transformer using complex-valued neural networks" at the 2011 IEEE International Conference on Smart Grid and Clean Energy Technologies, China, 2011, in press.

[6] Power transformer: basic principles [Online]. Available: http://en.wikipedia.org/wiki/Power_transformer#Basic_principles

[7] A. I. Voldek, *Electricheskie mashiny,* St. Petersburg, Energiya, 1978

[8] S. G. German-Galkin, G. A. Kardonov, *Electricheskie mashiny. Laboratornie raboti na PK*, St. Petersburg, Korona print, 2003.

[9] Orion OMP transformers catalog [Online]. Available: http://www.orion-nt.ru/cat/transf/tr-hmel/tr_omp1.htm

[10] A. Lykin, "Elektricheskie sistemy i seti", Moscow, Logos, 2008.

[11] H. Leung, S. Haykin, "The complex back propagation", IEEE Transactions on Signal Processing., Vol.39, No.9, September 1991., pp. 2101 – 2104.

[12] T. Kim, T. Adali, "Fully complex multi-layered perceptron network for nonlinear signal processing", VLSI Signal Processing 32, pp. 29-43, 2002.

[13] V. Mladenov, E. Zirintsis, C. Pavlatos, V. Vita, L. Ekonomou "Application of Neural Networks for on-line calculations". 9th WSEAS International Conference On Applied Computer Science, 2009.

[14] L. Ekonomou, D.S. Oikonomou "Application and comparison of several artificial neural networks for forecasting the Hellenic daily electricity demand load". 7th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering And Data Bases (AIKED'08), University of Cambridge, UK, Feb 20-22, 2008.

[15] Zimmermann H.-G., Minin A., Kusherbaeva V., "Historical consistent complex valued recurrent neural network", ICANN 2011, Part I, LNCS 6791, pp. 185–192, 2011.

[16] Zimmermann H.-G., Minin A., Kusherbaeva V., "Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms", Proc. of ESANN 2011, pp. 216-222, 2011.

[17] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory". IEEE Proceedings, F: Communications, Radar and Signal Processing, 130(1):1116, 1983.

[18] R. Neuneier, H.G. Zimmermann, "How to train neural networks, neural networks: tricks of the trade", Springer, pp. 373-423, 1998