Song, Xinya; Cai, Hui; Zeng, Yuelin; Jiang, Teng; Schlegel, Steffen;
Westermann, Dirk:

Parameter tuning by neural network for digital twins of inverter

DOI: 10.22032/dbt.50142

URN: urn:nbn:de:gbv:ilm1-2021200141

Original published in:	*Power and Energy Student Summit 2019 / Wolter, Martin. - Magdeburg : Universitätsbibliothek, 2019. - (2019), p. 1-6. ISBN 978-3-944722-84-9 (Res electricae Magdeburgenses; 77)*
Original published:	2019-07-08
ISSN:	2628-1716
DOI:	10.24352/UB.OVGU-2019-086
[*Visited:*	2021-10-14]

Parameter Tuning by Neural Network for Digital Twins of Inverter

X.Song, H.Cai, YL. Zeng, T. Jiang, S. Schlegel, D. Westermann
song.xinya@tu-ilmenau.de

***Abstract*—This paper proposed the innovative methods based on the neural network (NN) to build the digital twins (DT) of the inverter model. The proposed methods can be divided into two groups: firstly, an online-parameterized tuner for the PI controller is formulated through backpropagation (BP) algorithm; and secondly an NN-based identifier is used to approximate the nonlinear functional dynamics of the targeted control loop of the inverter. The design of PI tuner is based on the deviation between the output of the model and the reference output from measurement data. Then, according to the difference, the tuner can calculate the appropriate parameter of the current and voltage controller to track the dynamic behaviour of the reference model. The NN identifier is, however, to replicate the dynamic character of the reference model by NN which is initially trained offline with extensive test data and afterward is applied to online tuning.**

Index Terms- neural network, digital twins, backpropagation, PI tuner, NN identifier

Introduction

A growing number of power inverters are being used to feed renewable energies into the grid. Because of the switching processes, the harmonic oscillations are also brought into the power system [1]-[3]. To guarantee the stability of the network and the success of its operation, it is crucial to investigate the interaction between the parallel-connected inverters and the distribution grid. The paper initially focuses on the modelling of the inverter and the network to simulate the possible dynamic in inverter and network. To enhance the precision of the model, the parameter tuning will be considered to build the digital twins (DT) [6] of the inverter based on the measurement data from the reference model. The original concept of DT could be dated back to 2002 in the presentation about the formation of a Product Lifecycle Management (PLM) from the University of Michigan, which it was referred to as the Mirrored Spaces Model. [7] The first adoption of Digital Twin as a conceptual basis used as a simulation of a vehicle or system that uses the best available physical models to mirror the life of its flying twin, which implies that a highly-detailed simulation model can reproduce its physical behaviour as close as possible. [6],[7]. In electrical and electronic area, they are, with the development of computer simulation, dependent on the simulation models for the understanding the dynamics of power system. The modelling technology has been evolved from an access to mathematical and computing experts to a standard tool in an engineer's portfolio, which is afterwards developed as a support tools to solve the problem like the validation and testing for system.

Because of its autonomous, adaptive and intelligent rectification of parameters, artificial intelligence (AI) is becoming increasingly significant [7]-[9]. This paper uses one of the AI methods, the neural network (NN) [10], to build, firstly, a PI tuner for the parameterization of the controller of the inverter model. Because of the adaptive character [10] of NN, the PI tuner can dynamically rectify the controller according to the response of measurement data. On this account, the NN will be built to tune the controller automatically according to the deviation of the targeted model and the measurement data of the reference model in the control loop of inverter. By using backpropagation algorithm (BP) [11], the gradient that is applied to the calculation of weights to be used in the network can be computed as to minimize the error between model and the measured data of reference model. For the sake of evaluation of this method, the parameter estimation, which also aims to rectify the parameter of the control loop of the inverter, will be presented in this paper.

Another application of NN in this paper is on account of its self-learning ability [12]. The performance of NN has increased significantly in new research domains such as neurobiology, bioinformatics. Moreover, neuropsychology has also become increasingly demanding, particularly at the interfaces of biology and medicine with computer science [10]-[12]. The aim is to create an exact replication of the natural networking of neurons and thus, for example, to draw inferences about the functioning of the brain. In contrast to the mapping of neurons, the technical application has no longer concentrated on the biological plausibility of the networking and activity of NN, instead in automation technology on the stability and real-time performance of the application. With the enormous measurement data, the NN is trained offline by using the training algorithm of supervised learning [11]. The NN can be regarded as an identifier, which could be used online to replicate the dynamic behaviour of reference inverter.

The purpose of this paper is to build a DT of the inverter by using the parameter tuning method. Accordingly, the structure of this work can be arranged as follows: In chapter II, the inverter model which will be tuned is built firstly based on the differential equations described in AC side and DC side. To maintain the targeted current and voltage, an overlaid voltage and an underlaid current control loop are designed. The initial parameter of the inverter model can be calculated by the optimum process, which will also be described in chapter II. Afterward, the theoretical fundamentals of parameter tuning strategy will be presented in chapter III. Then, the tuning process will be operated and the simulation results with the new parameter can be demonstrated in chapter IV. Finally, this paper concludes with a summary of the main points.

SIMULATION MODEL

A precise model for investigating the influence of the effect caused by the parallel-connected inverter is required [3]. With regards to the modeling concept [4] focused in this chapter, the model of inverter including its overlaid control loop is operated. To simplify the modeling procedure, the states and the topology considered are transformed from natural coordinates to αβ coordinates. Furthermore, the transformation from αβ coordinates to dq coordinates for the control of the states is executed [5].

Fig. 1 illustrates the topology of the inverter circuit, which connect with the voltage source of the grid. The circuit is shown in natural coordinates and includes a three-phase inverter with an intermediate voltage circuit, RL filter circle, and a capacity.

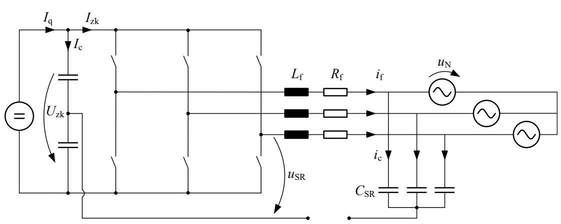

Fig. 1: Topology of Inverter model with the ideal switch

To model the dynamic process by using the differential equations, the system behavior is first converted to vector coordinates (αβ coordinates).

$$\vec{\underline{u}}_{SR} = R\cdot\vec{\underline{i}}_f + L\cdot\dot{\vec{\underline{i}}}_f + \vec{\underline{u}}_N \quad (1)$$

Here $R = R_f + R_N$, $L = L_f + L_N$

$$I_q = I_c + I_{zk} = I_{zk} + C\cdot\dot{U}_{zk} \quad (2)$$

To associate the AC and DC, the power relationship of both sides is used here:

$$P_{AC} = P_{DC} \quad (3)$$

with $P_{DC} = I_{zk}\cdot U_{dc}$, $P_{AC} = \frac{3}{2}\cdot\vec{\underline{i}}_f\cdot\vec{\underline{u}}_{SR}$, $\vec{\underline{u}}_{SR} = \vec{\underline{v}}\cdot\frac{U_{zk}}{2}$

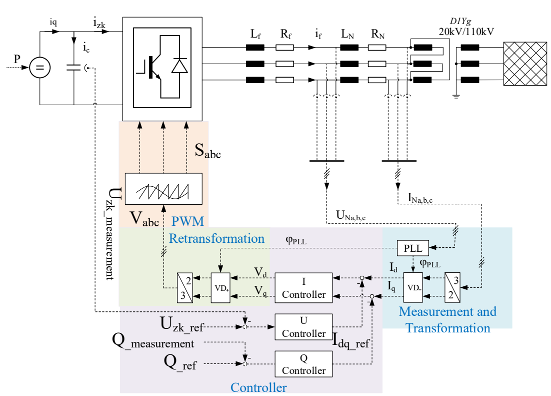

Fig. 2: Equivalent block diagram of inverter model

According to the above conditions, the power relationship can be rewritten:

$$I_{zk}\cdot U_{zk} = \frac{3}{2}\cdot\vec{\underline{i}}_f\cdot\vec{\underline{u}}_{SR} = \frac{3}{4}\cdot\vec{\underline{i}}_f\cdot\vec{\underline{v}}\cdot U_{zk} \quad (4)$$

Then,

$$I_{zk} = \frac{3}{4}\cdot(V_\alpha I_{f\alpha} + V_\beta I_{f\beta}) \quad (5)$$

From αβ coordinates to dq coordinates for controlling process: With these functions, the diagram can be shown with the control loop in Fig.2

PARAMETER TUNING STRATEGY

To research the interaction between inverter and grid, simulation will be the prior choice because of its reasonable price and safe research environment. This, however, exists in the difference between the real network and the simulation results, which can cause the imprecision of the analysis, see Fig.3.

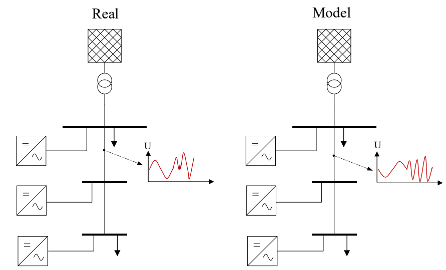

Fig. 3: Difference between the measurement data and the simulation results

On account of this, the parameterization should be applied to the model to tune the parameter. It is determined to decrease the deviation between the model and the measurement data from the grid. In this paper, two parameter tuning's methods will be used to build DT of inverter.

A. PI tuner through neural network

According to the BP algorithm, an illustration of the relevant procedure for the setup of the neural network for the parameterization of the PI controller is then presented. The form for the digital PI controller [33] is as follows:

$$u(k) = u(k-1) + K_p\cdot\big(e(k) - e(k-1)\big) + K_i\cdot e(k) \quad (6)$$

The structure of applied neural network in PI tuner can be shown in Fig. 4.

There are three inputs, two outputs and five hidden neurons. The states for further computations are collected in the input layer. The input variable is defined as:

$$O_i^{(1)} = x(i), i = 1,2,3 \quad (7)$$

Between the input layer and the hidden layer, the input states are transmitted by the weighted factor. Then, the input value of the hidden layer is calculated:

$$net_j^{(2)}(k) = \sum_{i=0}^{3} W_{ij}\cdot O_i^{(1)}, j = 1,2,\dots,5 \quad (8)$$

Fig.4 obviously reveals that the output of the hidden layer is determined by function f_1 , which is called the activation function. The output value after the activation function is defined as:

$$O_j^{(2)}(k) = f_1(net_j^{(2)}(k)), j = 1,2,\dots,5 \quad (9)$$

Similarly, the input and output value of output layer are:

$$net_l^{(3)}(k) = \sum_{i=0}^{3} W_{jl}\cdot O_j^{(2)}, l = 1,2 \quad (10)$$

$$O_{\mathrm{l}}^{(3)}(k) = f_2\left(net_{\mathrm{l}}^{(3)}(k)\right), \mathrm{l} = 1,2 \tag{11}$$

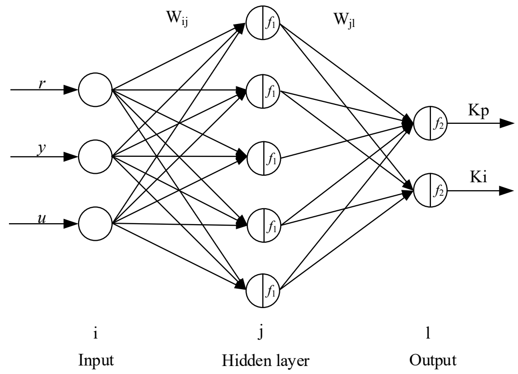

Fig. 4: Structure of neural network in PI tuner

To search for the optimized parameters, the gradient regression method is used. Before the computation, the evaluation function must be determined in the first place. In this paper, the evaluation function for the optimization process is described as:

$$E(k) = \frac{1}{2}(r(k) - y(k))^2 \tag{12}$$

The weight matrix is rectified by the gradient regression method through the (24):

$$\Delta W_{\mathrm{jl}}(k) = -\eta \cdot \frac{\partial E(k)}{\partial W_{\mathrm{jl}}} + \alpha \cdot \Delta W_{\mathrm{jl}}(k-1) \tag{13}$$

Here, η is the learning rate and α means the inertial parameters. From the equation, the change of weight matrix has a correlation with the evaluation function. Consequently, in order to calculate the new weight matrix, the following differential equation should be computed first:

$$\frac{\partial E(k)}{\partial W_{\mathrm{jl}}} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_{\mathrm{l}}^{(3)}(k)} \cdot \frac{\partial O_{\mathrm{l}}^{(3)}(k)}{\partial net_{\mathrm{l}}^{(3)}(k)} \cdot \frac{\partial net_{\mathrm{l}}^{(3)}(k)}{\partial W_{\mathrm{jl}}} \tag{14}$$

In (14),
$\frac{\partial net_{\mathrm{l}}^{(3)}(k)}{\partial W_{\mathrm{jl}}} = O_{\mathrm{j}}^{(2)}, \frac{\partial u(k)}{\partial O_1^{(3)}(k)} = e(k) - e(k-1), \frac{\partial u(k)}{\partial O_2^{(3)}(k)} = e(k),$
The change of the weight matrix can be determined by the equations mentioned above:

$$\Delta W_{\mathrm{jl}}(k) = \alpha \cdot \Delta W_{\mathrm{jl}}(k-1) + \eta \cdot \delta_{\mathrm{l}}^{(3)} O_{\mathrm{j}}^{(2)}(k) \tag{15}$$

with $\delta_{\mathrm{l}}^{(3)} = e(k) sgn(\frac{\partial y(k)}{\partial u(k)}) \cdot \frac{\partial u(k)}{\partial O_{\mathrm{l}}^{(3)}(k)} \cdot f'_2(net_{\mathrm{l}}^{(3)}(k))$
Through the same procedure, the changes of the weights of the hidden layer is as follows:

$$\Delta W_{\mathrm{ij}}(k) = \alpha \cdot \Delta W_{\mathrm{ij}}(k-1) + \eta \cdot \delta_j^{(2)} O_i^{(1)}(k) \tag{16}$$

$$\delta_j^{(2)} = f'_1\left(net_j^{(2)}(k)\right) \cdot \sum_{l=1}^{2} \delta_l^{(3)} \cdot W_{jl}(k), \tag{17}$$

According to the mentioned above, two PI controllers are used to stabilize the voltage and current, which means that two PI tuners will be used to parameterize the inverter. Each tuner picks up the measured data from the demonstrator and the output of model of each control loop. The output of each tuner is the optimized K_{p} and K_{i}, see Fig.5.

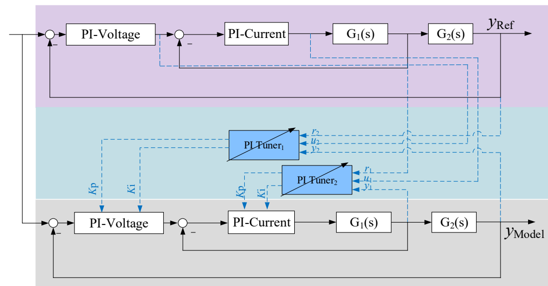

Fig. 5: Structure of neural network in PI tuner

B. Neural network(supervised learning)

In supervised learning, an external supervisor is used to provide the network with the required response to each input or the deviation of the actual from the correct output [10]-[12]. According to the deviation, the modification of the network is carried out via the learning rules. The size of the errors determines the weight configuration, which is varied and adapted to the current learning progress. After updating the adjusted weight parameters, a performance improvement should be achieved. Another goal is to minimize the network error as much as possible. The general graphical representation of the supervised learning procedure is shown in Fig.6

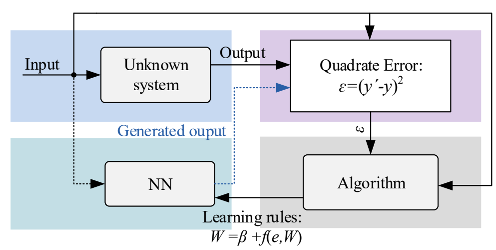

Fig. 6: Structure of supervised learning

Based on the theoretical explanation mentioned above, the voltage control loop and the current control loop of the reference model can be replicated by the NN identifier, the general idea of which can be observed in Fig.7. To replicate the reference model, three neural networks are designed to imitate the voltage controller, the current control loop and the controlled plant of the overlaid control loop. Based on the measurement data of the input and the output from the reference model, these three NN will be trained firstly according to the supervised learning process. The selection of the number of hidden neurons of NN has no exact criterium. In this work, the NN with 15 hidden neurons, after testing, exerts the best performance. To build the DT, the objective is that the state variables of the model are the same as in the reference model.

The simulation results by NN-identifier will be presented in the following chapter. For the sake of comparing these methods, the Root-Mean-Square-Error (RMSE) [10] of the state will be calculated to justify the similarity between the simulation results and the measurement data, see Table 1.

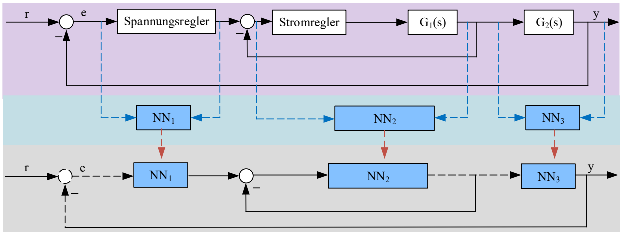

Fig. 7: Concept of application of NN identifier to build DT

PARAMETER TUNING AND SIMULATION

In this chapter, the validation process is firstly operated to testify whether there exists the deviation between the simulation results and the measurement data exists. The difference between the model and the reference model is investigated by comparing the dynamic behaviour. From this difference, the parameter tuning methods are correspondingly used to rectify the parameters in order to build the DT to replicate the dynamic process of the reference model. The general idea for the parameterization can be shown in Fig.8. As the explanation above, the parameters in the model are rectified according to the deviation between the simulation model and the measurement data from the reference model.

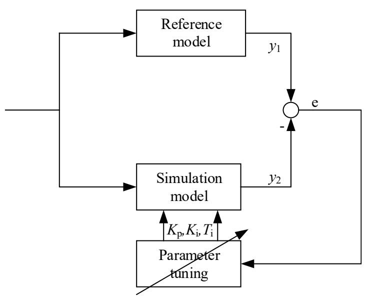

Fig. 8: General ideal of parameter tuning

By changing the operation point of the inverter, the active and reactive power curve of the reference model and the simulation model at PCC are presented as follows.

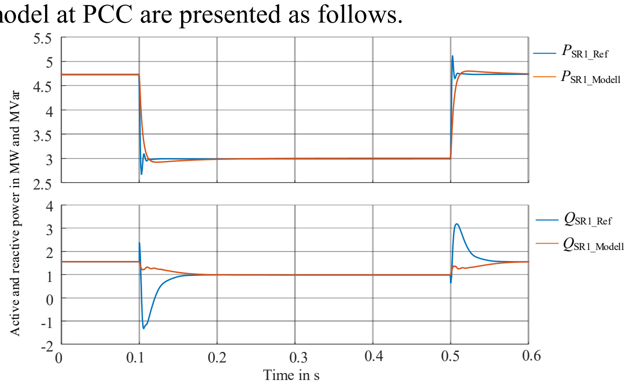

Fig. 9: Validation with active and reactive power at PCC

From Fig.9, it is obvious to see that the deviation between the reference model (blue) and the simulation model (orange) is sized during the changing process than at the steady period. Additionally, at the balancing procedure, the reference model can, however, reach the new operating point more quickly, which means that the parameter or the structure of the reference model differs from the simulated model. Except for state of the grid, the control loop of the converter should also be observed. According to the explanations in the front chapter, the amplitude of the voltage is determined by the regulated DC link voltage and the magnitude of the current is also defined by the current control loop. The following diagram illustrates the output of the overlaid voltage loop.
The actual value of the DC link voltage in Fig.10 clearly shows at 0.1s and 0.5s that the response time of the reference model is slower than the simulated model. In this regard, the DC link voltage of the reference model needs about 0.3s to reach the new steady state after the operating point changes, while the model needs only about 0.02s. It should be noted that the dynamic difference between these two models exists from 0.1s when the change occurs.

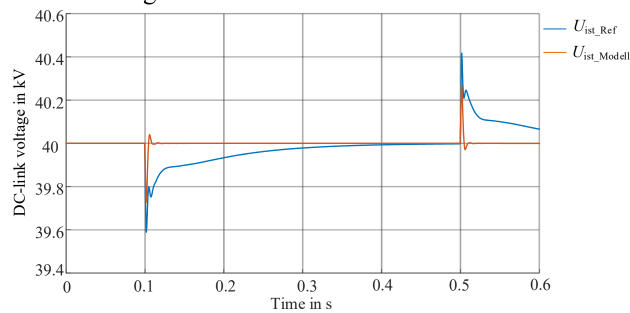

Fig. 10: Validation with DC link voltage of inverter

As exhibited in Fig.11, it should be pointed out that the amplification of the current controller is too small, which leads to the lower peak at the changing point

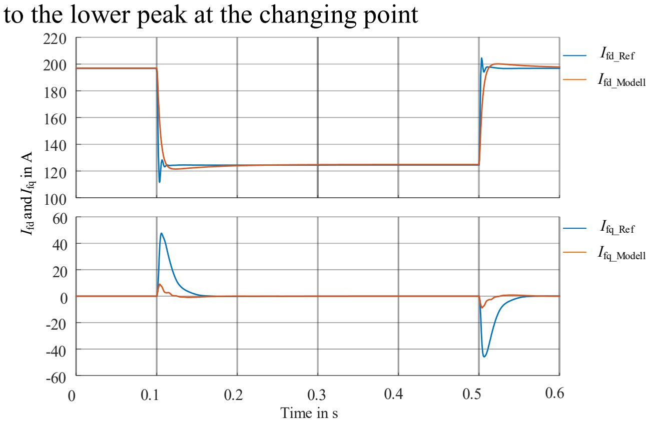

Fig. 11: Validation with output of current control loop on d and q axis of inverter

Besides, it can be observed that, on q axis, the integral parameter is too large, which has a smaller duration to reach the new steady point. Based on validation results, it is necessary to take the parameterization into account to reduce the deviation between them. As the explanation in chapter 3, four methods can be used into parameterization, and the simulation results after the tuning are in following.

1) *PI tuner*

In this paper, three controllers including DC link voltage controller, the current controller on axis d and current controller on axis q are parameterized separately. For this purpose, altogether six parameters are needed to be tuned in three controllers. The curve of the voltage controller is shown in Fig.12. It can be clearly seen that the controller parameters vary with the change in the DC link voltage (see Fig.12) and that the curves have the same dynamic behaviour.

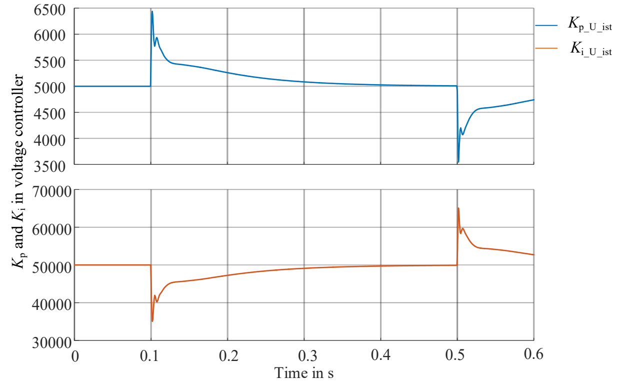

Fig. 12: $\boldsymbol{K_p}$ and $\boldsymbol{K_i}$ of voltage controller by PI tuner based on the validation

With PI tuner, the state of the model, after parameterization, is closer to the reference model (see Fig.13). It reveals the state of active and reactive power by changing the operating point of the simulation model and the measurement data. Compared to the validation results, the active power curve is more closely matched to the reference model.

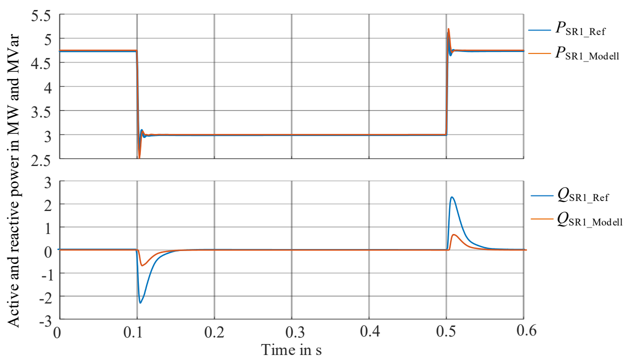

Fig. 13: Active and reactive power after parameter tuning by PI tuner

Despite the small deviation in the balancing process, the RMSE of the active power can reach 84.93%. However, the difference of the reactive power is bigger than that of the active power. Nevertheless, the behaviour of the reactive power in the dynamic process between 0.1s and 0.5s is identical.

As explained in Chapter 2, the dynamic behaviour of the three-phase voltage is determined by the DC link voltage. To build the DT, the DC link voltage dynamics between the parameterized model and the reference model should match with each other firstly. The DC link voltage plot is shown in Fig.14. This clearly indicates that the dynamic response between two curves (orange and blue line) is quite similar.

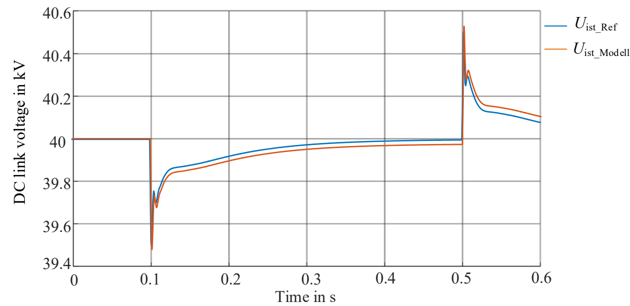

Fig. 14: DC link voltage after parameter tuning by PI tuner

In comparison with validation results, the output of the current control loop can better match the reference model after parameterization by the PI tuner (see Fig.15).

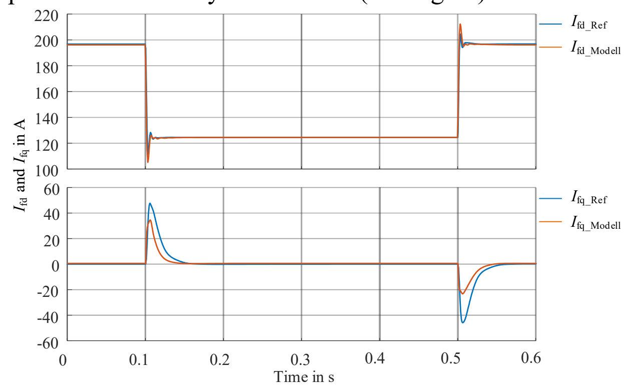

Fig. 15: $\boldsymbol{I_{fd}}$ and $\boldsymbol{I_{fq}}$ after parameter tuning by PI tuner

After the operation point changes, the behaviour during the dynamic process of the simulation model with the new parameters can better track the reference model. The figure below shows that despite the similar dynamic behaviour there exists still a deviation between these two curves. The maximum value of the orange line is about 36A, while at the same time the maximum value of the blue line is about 50A. As a result, it takes a longer balancing time of reference model to reach the steady state.

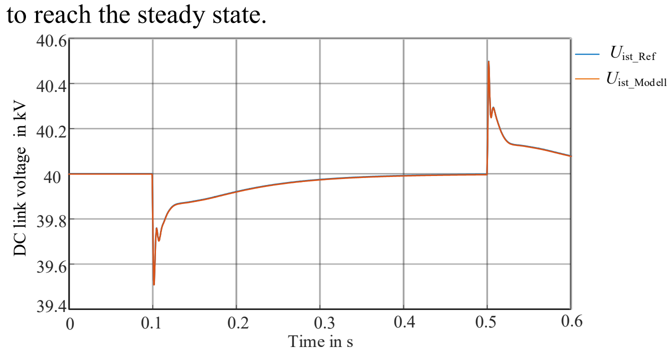

Fig. 16: DC link voltage after parameter tuning by NN

1) Neural network

Fig.16 shows the output of the voltage loop. The result illustrates that the two curves are identical, i.e. the voltage loop modeled by the NN is highly similar to the reference model. From the Tab.1, it can be also known that the RMSE of the

simulation model, which is built through the NN, can reach 99.3%.
Then, the underlaid current loop will be observed, which is simulated by the NN_2 (see Fig.7). To train these three NN, there are 200.000 data accumulated from the reference model. The output of NN_2 is shown in the following Fig.17.

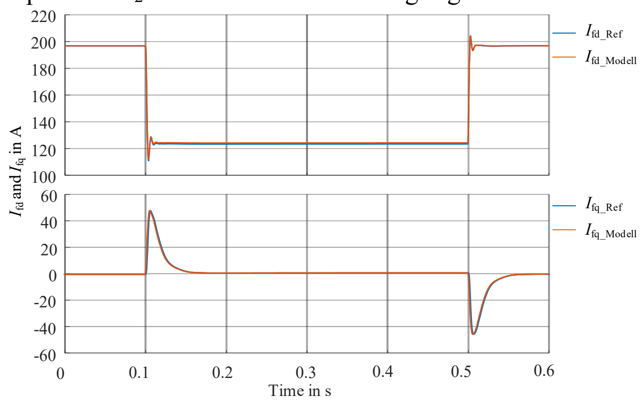

Fig. 17: $\boldsymbol{I_{fd}}$ and $\boldsymbol{I_{fq}}$ after parameter tuning by NN

To compare the results of these four methods, Tab.1 indicates us the RMSE of state and the output of the control loop, which represents the similarity grad between the reference model and the tuned model. From the table, the highest precise rests in the model tuned through NN. The similarity of the DT and reference model can reach over 97% on an average.

Tab. 1: Comparison of simulation results after parameter tuning

	U_{DC}	U_{SR1}	I_{fd}	I_{fq}	P	Q
PI	82,56%	88,37%	81,45%	80,33%	84,93%	79,78%
NN	99,5%	99,1%	97,77%	98,8%	97,65%	98,76%

CONCLUSION

To investigate the impact caused by the decentralized renewable generation, the simulation is firstly taken into consideration before the application. Thus, this paper has firstly built an inverter model to research the existing dynamic interaction between the distribution grid and the inverter. By comparing the simulation result and the testing data, there is, however, the deviation between simulation results and the measurement data from the reference model, which can lead to the inaccuracy of the research results. To improve the precision of the model, the tuning method has been applied, see Table 2.

Tab. 2: Comparison of different parameter tuning's methods

Method	advantage	disadvantage
PI tuner	• online parameter tuning • intelligent and self-learning • little training data	• lowest precision • only tuning of K_p and K_i
NN	• highest precision of these four methods • better identifier than system identification • online tuning	• Huge amount of data for the training process • NN transfer function without tuning the exact parameter

PI tuner aims to tune the controller's parameter of the overlaid voltage control loop and the underlaid current control loop on d and q axis for the tracking of reference inverter model. In contrast to parameter estimation, the PI tuner can merely rectify the controller's parameter which result in a lower precision. NN identifier utilizes the neural network to replicate the dynamic behaviour by training the NN with the measurement data from the reference model. By contrast with the PI tuner, the NN identifier can effectively build the DT of the inverter, which is over 97% the same as the reference model.

REFERENCES

[1] A. Y. Liu, P. H. Lan, and H. H. Lin, "An IEEE 1547-Based Power Conditioner Test System for Distributed Energy Resources," *EPE*, vol. 05, no. 04, pp. 945–949, 2013.

[2] Tao Yang, "Development of dynamic phasors for the modelling of aircraft electrical power systems," Ph.D. dissertation, University of Nottingham.

[3] I. Hauer, A. Naumann, M. Stotzer, and Z. A. Styczynski, "Communication interface requirements during critical situations in a Smart Grid," in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, Berlin, Germany, op. 2012, pp. 1–7.

[4] M. Motoyama *et al.,* "Erratum to "Improving the Power Generation Performance of a Solar Tower Using Thermal Updraft Wind" [Energy and Power Engineering Vol. 6 No. 11 (October 2014) 362-370]," *EPE*, vol. 07, no. 06, pp. 255–257, 2015.

[5] F. Jenni and D. Wüest, Eds., *Steuerverfahren für selbstgeführte Stromrichter*. Zürich, Stuttgart: Vdf, Hochsch.-Verl. an der ETH Zürich; Teubner, 1995.

[6] F. Biesinger, D. Meike, B. Kras und M. Weyrich, „A Case Study for a Digital Twin of Body-in-White Production Systems General Concept for Automated Updating of Planning Projects in the Digital Factory" in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, Sep. 2018 - Sep. 2018, S. 19–26.

[7] F. Biesinger, D. Meike, B. Krass, M. Weyrich, "A Digital Twin for the Production Planning based on Cyber-Physiscal Systems",*CIRP ICME 18-12th CIRP Conference on Intelligent Computation in Manufacturing Engineering*, 2018.

[8] R. Rosen, G. von Wichert, G. Lo, K. D. Bettenhausen, "About The Importance of Autonomy and Digital Twins for the Future of Manufacturing." in ScienceDirect; IF AC Papers Online CONFERENCE PAPER ARCHIVE, pp. 567-572, 2015

[9] S. S. Haykin, *Neural networks: A comprehensive foundation / Simon Haykin,* 2nd ed. Upper Saddle River, N.J.: Prentice Hall; London : Prentice-Hall International, 1999.

[10] S. Kamalasadan, G. Swann, and A. A. Ghandakly, "A novel radial basis function neural network based intelligent adaptive architecture for power system stabilizer," in *Proc. North Amer. Power Symp.*, 2009,

[11] pp. 1489–1496. K. Reinisch, Analyse und Synthese kontinuierlicher Regelungs- und Steuerungssysteme 3. Aufl.- Berlin.

[12] A. Zell, *Simulation neuronaler Netze,* 4th ed. München [u.a.]: Oldenbourg, 2003.