DISTURBANCE PREDICTION FOR TIME VARYING SYSTEMS USING FEEDFORWARD NEURAL NETWORK

[1]M. Eswar Murali
[1]PG Scholar, Dept. of E.C.E
Jawaharlal Nehru Technological University
Kakinada, A.P

[2]Dr. R. Madhu
[2]Assistant Professor, Dept. of E.C.E
Jawaharlal Nehru Technological University
Kakinada, A.P

\

Abstract— In this paper, a feedforward control scheme with disturbance prediction capability is proposed which improves its performance under accurate prediction, but it deteriorates due to the inevitable prediction error. A recently developed repetitive controller is being used in this paper. A time-base generator has been used as the input signal for neural approximator. The dynamic optimization problem (DOP) of shaping the control signal is solvable with the help of a function approximator. Here the feed-forward neural network (FFNN) is used as the function approximator. A constant-amplitude constant-frequency voltage-source inverter along with an output LC filter, is assumed to be equipped with a disturbance load current sensor. This enable the implementation of disturbance feed-forward path as part of the non-repetitive subsystem. The robustness to a measured noise is also tested alongside. The comparison is based on the results of numerical experiments.

Keywords— Full state feedback control, Repetitive Control, Artificial neural network, Dual Feedforward Neural Network.

I. INTRODUCTION

A typical process control system, in most operations, works to regulate the output against the frequent and variable disturbances, while there is rarely a set-point change. The feedback controller starts to act against a disturbance only when it has resulted in output error and may give a poor output regulation, especially for a system with input delay. Disturbance rejection has become a major issue in control systems. To improve the performance, feedforward control is a common solution when the disturbance is measurable, as this control can act before the disturbance has resulted in output error. Disturbance rejection solely based on the feedback control tends to be poor when the disturbance propagates to the output quickly. With measurable disturbances, the feedforward control has a great potential for better performance. The feedback controller starts to act against a disturbance only when it has resulted in output error and may give a poor output regulation, especially for a system with input delay. One critical issue with feedforward control is its realizability. If the disturbance has less time delay to reach the output than that for the control signal, the feedforward controller will have time lead and is not physically realizable. Another unrealizable case is that the controller is not proper rational. The simplest approximation is to just use the gain of controller, which lose dynamic compensation function of the ideal controller. A dynamic proper rational controller appeals with a better approximation. Repetitive process control has gained noticeable attention during the last few years. This is mostly due to the increase in developing control systems characterized by nearly perfect reference tracking and disturbance rejection capabilities. The repetitiveness of a process to be controlled is common in many industrial systems. The repetitive control (RC) or iterative learning control (ILC) are often an appealing solution. Both RC and ILC can be analyzed in a uniform framework. The ILC mainly focuses on batch processes. It is common to assume that the initial state of can reset. ILC has gained acceptance in industrial robotics. Repetitive control is commonly used in continuous processes which is characterized by the initial state of each signal originating from the final state of the previous signal, i.e. resetting is not acceptable. However, the distinction between RC and an ILC is that it can be particularly relevant to control schemes that are developed for some continuous repetitive processes, and as such could be labelled as Repetitive Control, but at the same time the techniques that are adopted here are clearly of iterative nature and include reinforced learning.

A control task encountered in power electronic converters is sometimes related to continuous repetitive processes. Any constant-amplitude constant-frequency (CACF) converter may work here just to illustrate its action in the repetitive process. If a prime quality voltage wave shape is predicted in CACF VSI with an output LC filter no matter the nonlinear load current presence, the repetitiveness of the method is effective and should be exploited. The most common approach is to incorporate periodical terms among the controller to urge load harmonics selective rejection. However, there are two major problems that are related in the implementation of this scheme. First is its computational burden and the other is the problematic synthesis of periodic terms. That's why most of the CACF VSI controllers are still designed as non-repetitive types. Control tasks can be formulated again to pose the problem as an issue of optimization, and usually dynamic. The model predictive control (MPC) is a commonly acknowledged optimal scheme that is intended to shape the control signal according to the users given performance index. Here the model of the plant acts as a critic. The ILC method directly inspires the DOP formulation for repetitive procedures. The main objective is to shape the control signal according to the users input. An appropriate solver operates in online mode and tackles the DOP iteratively, where one iteration corresponds to a multiple number of passes and the plant handles the role of a critic. The task of stabilizing the system becomes more interesting if an external disturbance other than the controlled output is anticipated to enter the given system. Such type of disturbances are irresistible in power electronic converters which is prominently in the form of load current. Besides, it is even difficult to determine the upper band of such a disturbance, for example in the case of a diode rectifier it changes remarkably with the inductance of the rectifier. It is worthy in noticing that the online trained neurocontrollers are broadly discussed within the context of these non-repetitive control systems and several different motion control systems that are reported. Moreover, there are a few attempts to redefine these algorithms to make them suitable within the context of the repetitive control in power electronics. The performance of FFNN based controls is then compared numerically.

II. CHOOSING THE RIGHT LEARNING ALGORITHM:

The Levenberg–Marquardt backpropagation (BP) algorithm is initially used to guarantee rapid convergence of errors and better quality of voltage under stable circumstances. However, the resilient backpropagation, Broyden–Fletcher–Goldfarb–Shanno and quasi-Newton algorithm are very less computationally challenging learning algorithms. Choosing any optimization tool based on its performance is always hectic, the quality of the output waveform during steady and transient states is a requisite that needs to be investigated in the predefined system before selecting the candidates for its practical implementation. It is to be stressed out that these learning algorithms are to be used even in a noisy environment and the optimization task is progressive in nature. The following is due to the variable load conditions. There are different types of learning algorithms designed for training the neural networks. Consequently, only a commendable decision process is demonstrated and numerous recommendations are made.

III. REPETITIVE CONTROLLER BASED ON FEEDFORWARD NEURAL NETWORKS:

The main reason is to develop repetitive control schemes that inherently does not suffer from some long-term stability issues that are predominant and thus no special measures need to be taken to secure stable operation. By considering, the plug-in direct particle swarm controller cost function which is the combination of control error cost and the control signal dynamics cost. The latter component does not allow high-frequency oscillations. Although the FFNN based RC using the cost of a control error as the main objective, does not allow high-frequency oscillations due to the limited approximation capability of the FFNN with an already fixed number of neurons, a fixed activation function and constrained weights. It is very important to know that if the Feedforward neural networks activation functions obey a set of assumptions, then the network is a universal function approximator. It is clear that, on one hand, the limited number of neurons is required to prevent overlearning and on the other hand, it is quite challenging to select the complex network equally effective in the case of linear loads and also for highly nonlinear loads. It is clearly demonstrated that by adding a disturbance feedforward path in the *k*-direction improves the performance and makes selecting the number of neurons less challenging.

a) The *p*-direction controller:

The FFNN based repetitive controller, corresponding to any plug-in RC, needs to be accompanied by a p-direction controller if the dynamics is required to be shaped also along the signal, which could be the case in some applications of highly underdamped CACF VSI. Here, the full state feedback (FSF) controller is used for this purpose

$$u_{FSF} = -(k_{11} i_L^m + k_{12} u_C^m) \qquad (1)$$

Where {k_{11}, k_{12}} are the controller gains and $\{i_L^m, u_c^m\}$ are the measured state variables of the filter. Moreover, the FSF controller shapes the passage seen by the plug-in repetitive controller. By increasing damping makes the relevant approximation problem easier. Also the reference feedforward (RFF) path is represented as

$$u_{RFF} = k_{10} u_c^{ref} = (1 + k_{12}) u_c^{ref} \qquad (2)$$

It is introduced to keep the gain unity and the disturbance feedforward (*p*DFF) path is added to compensate it partially for the voltage drop

$$u_{pDFF} = k_{13} i_{load}^m = (\hat{R}_f + k_{11}) i_{load}^m \qquad (3)$$

Where $\hat{R}_f$ is the resistance of the LC filter. Then a significant identification error is assumed ($\hat{R}_f = 0.25 R_f$) in order to make the control errors produced by the non-repetitive path more realistic. The sum of these three paths represents the non-repetitive part of the overall control. Here, the equation (4) shows the non-repetitive control action.

$$u_{nonRC} = u_{RFF} + u_{FSF} + u_{pDFF} \qquad (4)$$

b) The *k*-direction controller:

The repetitive path includes a universal function approximator in the form of feedforward neural network along with a DOP-capable learning algorithm. Here the Levenberg-Marquard (L-M) training algorithm has been used. It is demonstrated in numerous studies that an FFNN trained in the online mode using back propagation (BP) methods such as L-M or resilient back propagations (RPROPs) can effectively control the non-repetitive processes. Noticeably, the well-documented practicality of the online trained neurocontrollers for non-repetitive processes in adjustable speed drives and generators has not been followed. By using an adequate cost function the repetitiveness of the process can be simply exploited by the neurocontroller. The following cost function is used for controlling the output voltage of the VSI

$$\boldsymbol{\varepsilon}_{ANN}^{\alpha,k_2}(\mathrm{k}) = \frac{k_2^2}{2}\sum_{p=1}^{\alpha}(u_c^{ref}(p) - u_c^m(k,p))^2 \qquad (5)$$

Where u_C^m denotes the measured output filter capacitor voltage, u_C^{ref} is the reference voltage and k_2 is the error scaling factor. It is to be noted that $\boldsymbol{\varepsilon}_{ANN}^{\alpha,k_2}$ is proportional to the mean squared error (MSE) calculated over the entire period of the reference signal, hence the repetitiveness of the process is directly incorporated into the functional definition. The time-base generator acts as the only input to the FFNN.

$$u_{FFNN}(p) = \omega_{10}^{(2)} + \sum_{n=1}^{N}\omega_{1n}^{(2)}\, v_n(p), \qquad (6)$$

Where

$$v_n(p) = f_1(\omega_{n0}^{(1)} + \omega_{n1}^{(1)} u_{TBG}(p)) \qquad (7)$$

u_{TBG} denotes the time-base generator signal, ω being the neural networks weights and f_1 expressing the activation function. It is common to use the hyperbolic tangent as the activation function with an insistence on the latter if the computational burden of the neurocontroller is to be considered for the final implementation of the algorithm. Keeping number of hidden neurons (N) constant and reducing iteratively. Thus equation (5) should then be seen as a function of the neural weights and the training algorithm is employed to continuously solve the DOP.

$$\varepsilon_{ANN}^{\alpha,k_2}(k) = \varepsilon_{ANN}(\omega^{(1)}(k), \omega^{(2)}(k)) \qquad (8)$$

The resulting output signal of the neural network is added with the non-repetitive path here a reference feedforward plus a full state feedback plus a disturbance feedforward to produce the reference signal for the pulse width modulator (PWM).

IV. DISTURBANCE DUAL FEEDFORWARD:

The concept of utilizing the data on the external disturbance in the *k*-direction tends to be real for the ILC or RC systems. The name '*k*-direction disturbance feedforward' would create an impact of inconsistency as the disturbance signal is allowed to pass through the repetitive part of the system that undoubtedly operates in the feedback mode. The *p*-direction disturbance feedforward that is being used here, is often termed for clarifying the static one in order to indicate that none of the derivatives of the disturbance signal are calculated. The word disturbance static feedforward should be interpreted as indicating that the disturbance latest accessible sample is used to alter the control signal. A similar perception can be made for the *k*-direction disturbance feedforward used here the most recent sample set of disturbance is passed through the FFNN to enhance a signal modifies the control signal directly. This is obvious when the training algorithm is infact redundant and could be halted at a steady state in the *k*-direction. The accessibility of the disturbance signal to add the non-repetitive *p*DFF has already been harnessed. The *p*DFF is of proportional type, i.e. it cannot compensate completely for the voltage drop whole along the inductive component (L_f). However, the resistance identification errors effect the accuracy of the *p*DFF whereas the potential *k*DFF cannot be affected by any such limitations since the *k*-path uses the physical plant in the relevant DOP. Choosing any optimization tool based on its performance is always hectic, the quality of the output waveform during steady and transient states is a requisite that needs to be investigated in the predefined system before selecting the candidates for its practical implementation. The training algorithm introduces a feedback action that is required to look over the FFNN when any new load conditions arise. It has been already known in the event of the repetitive neurocontroller, the resilient backpropagation algorithm – a regular winner for non-repetitive online-trained neurocontrollers over the repetitive controller. Many other learning algorithms can operate with the same effectiveness in terms of the time taken for execution and offers potentially faster convergence.

Therefore, the name disturbance feedforward is used here in relation with the *k*-direction. The static *p*-direction DFF cannot fully compensate for the voltage drop across the inductor that is present in the output LC filter. In general, the *k*-direction DFF does not have that issue because it operates on the complete course of the disturbance signal and contains ample data needed to compensate for the mentioned voltage drop. Measured noise levels are also tested here. The FFNN based learning approach to repetitive control gives a significant flexibility that is absent in the classic iterative learning scheme in terms of the controller input signal. The time base generator (TBG) signal is passed to the FFNN. The converter output voltage u_{VSI} average voltage u_{VSI}^{avg} plotted and its relation is defined below

$$u_{VSI}^{avg}(t) = \frac{1}{T_s}\int_t^{t+T_s} u_{VSI}(\tau)\, d\tau, \qquad (9)$$

Where $t = \frac{k}{f^{ref}} + pT_s$ with f^{ref} denotes the reference voltage frequency and T_s is the sampling time. Then the repetitive part is switched on, initially without the kDFF path and with N = 10 (hidden neurons).

V. RESULTS AND DISCUSSION:

Firstly we check the error without using the repetitive control with number of hidden neurons N=10 and with 3% noise level. Figure 1 illustrates the performance of the controller with its non-repetitive part only.

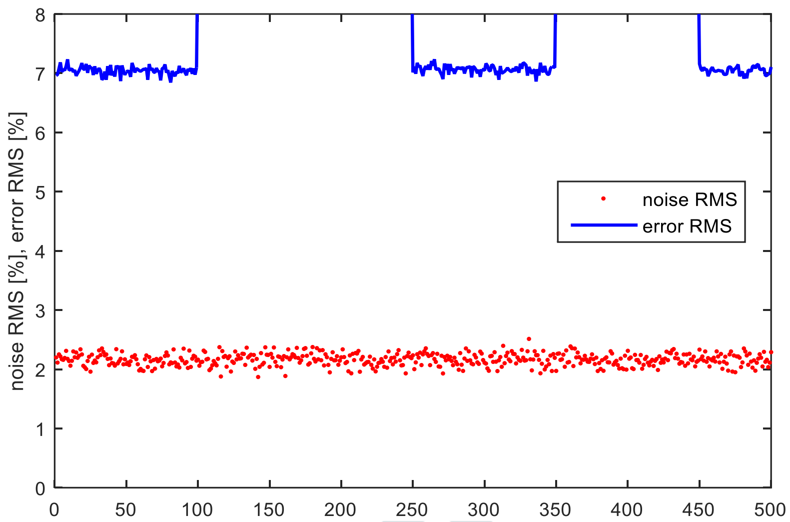

Fig 1: Performance without repetitive controller (RC).

As shown in figure 1 the error is very high which is usually not considered for application, this is due to the use of a non-repetitive controller. Now with the same parameters but using repetitive control the error reduces which is clearly demonstrated below.

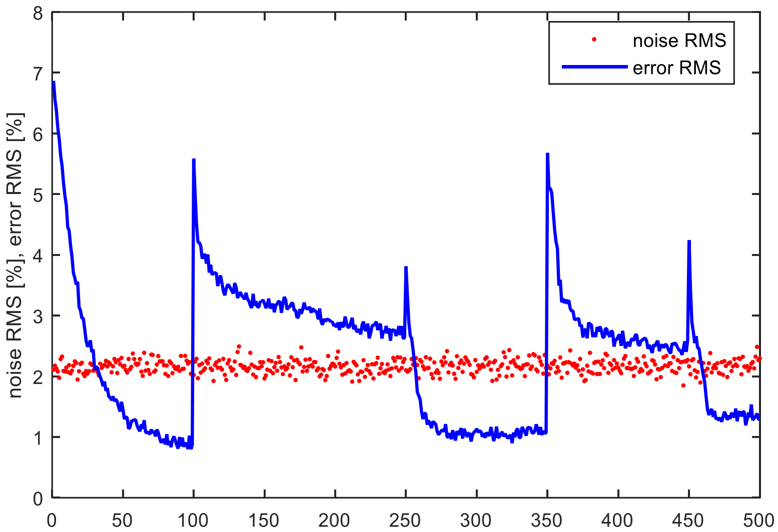

Fig 2: Error with repetitive controller and without *k*DFF path at 3% noise level.

By observing the figures 1 and 2 the RMSE is comparatively low in the case with repetitive controller. Without repetitive controller the error is at its highest peak. So using RC is definitely worthy in control processes. Similarly with disturbance dual feedforward path the error seems to be much reduced and the system performance can be achieved to the best level. The behavior of the controllers output voltage and disturbance load currents are also given below. In fig 3 the steady state behavior of the controller for the given inputs is determined.

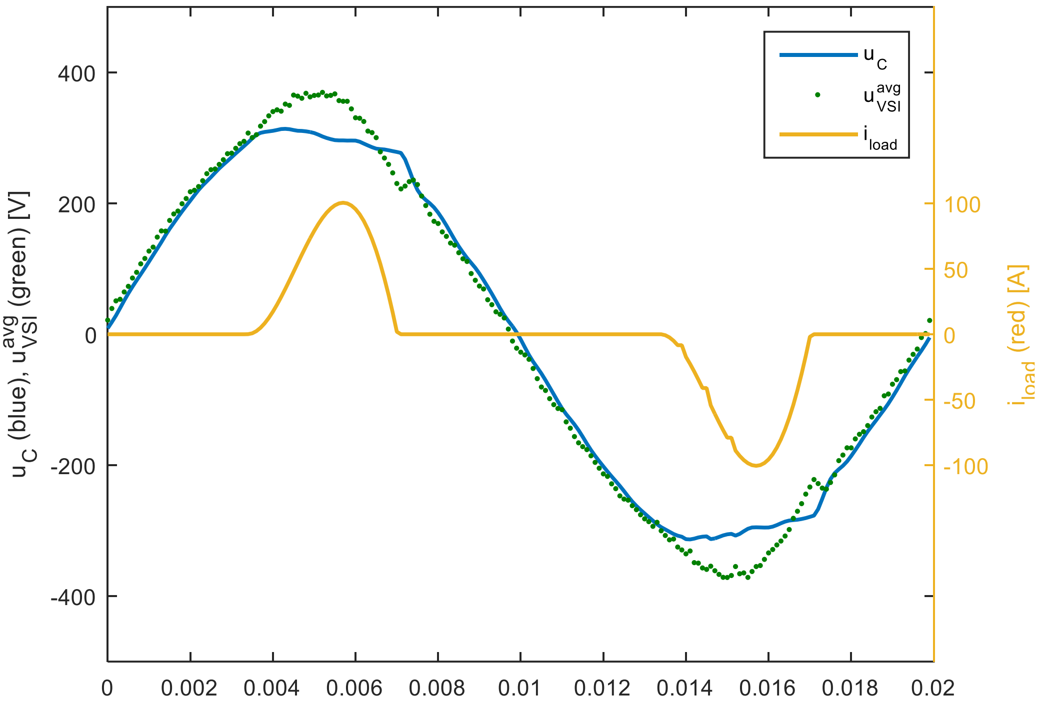

Fig 3: Steady state behavior of controller determining average voltage and load current.

Now with number of hidden neurons 10 and with disturbance dual feedforward path the RMSE settles way beyond the *k*DFF disabled system. Figure 4 demonstrates the performance of the controller with the both DFF paths. It can be observed that it is possible for the RMSE value to drop below the noise RMS value because its low-pass filtering action inherently present in the online trained neural network.

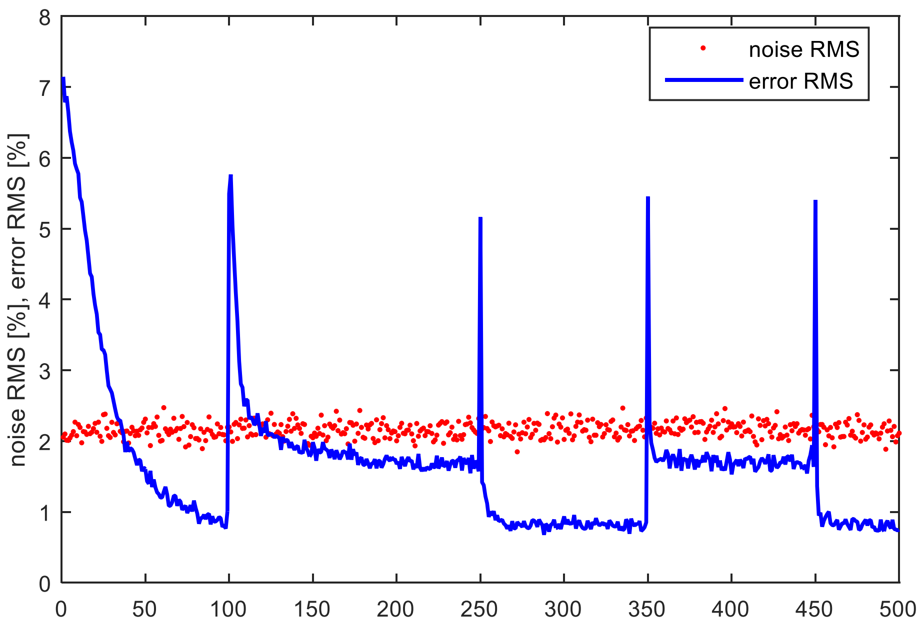

Fig 4: Error with repetitive controller and *k*DFF path and noise level 3%.

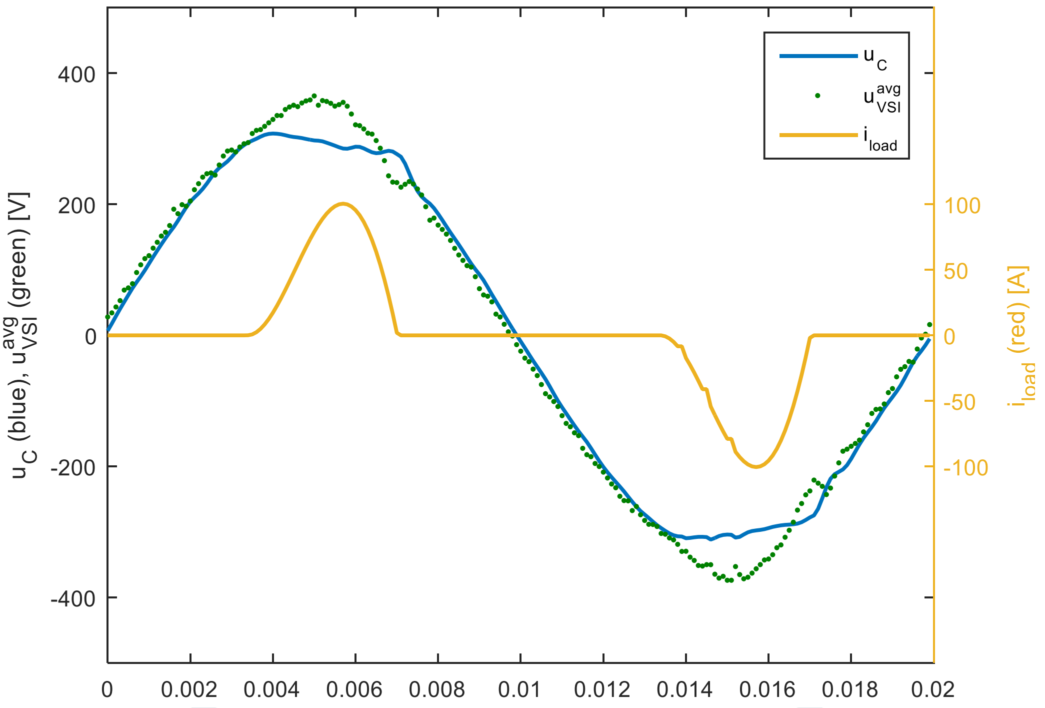

Fig 5: Steady state behavior of controller determining average voltage and load current.

By comparing the figures 2 and 4 the performance seems to be improved in the case of *k*DFF enabled path. Till now we have conducted with a measured noise level of 3% under controlled conditions. Now by testing the system with some random measured noise levels and observe the error response when the noise level is increased. The measured noise level is then changed to be 5% and conducted the test for both *k*DFF enabled and disabled systems.

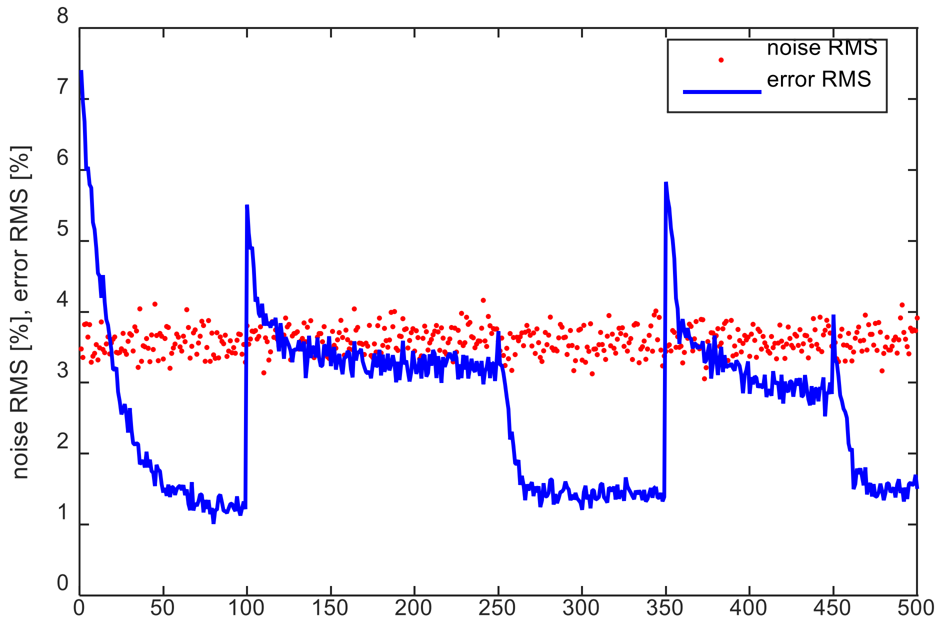

Fig 7: Error with repetitive controller and without *k*DFF path at 5% noise level.

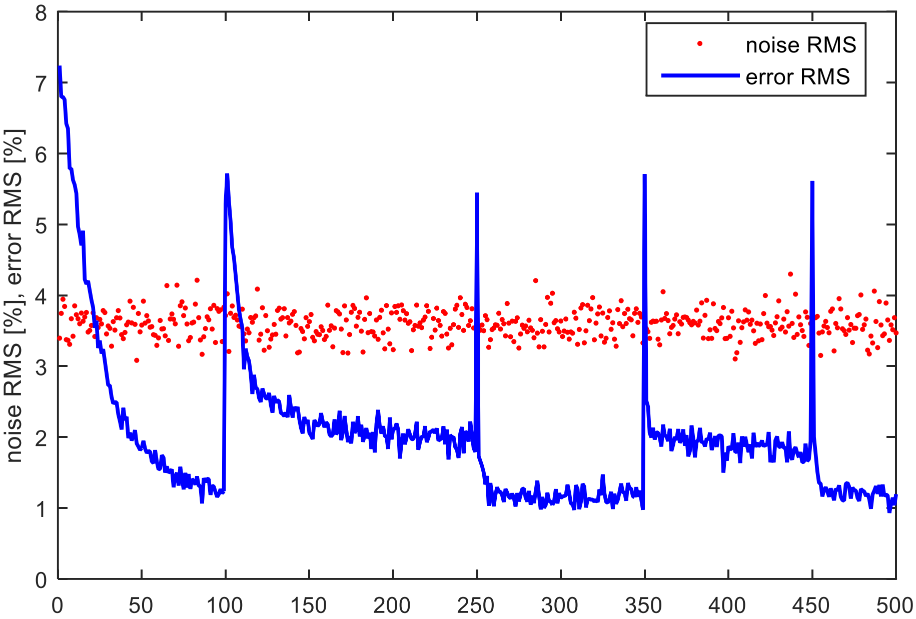

Fig 6: Error with repetitive controller and with *k*DFF path at 5% noise level.

By comparing the figures 4 and 6 the error is increased because of a slight raise in the input noise level but the error changes rapidly. From the figures 2 and 7 error detoriarates with increased noise level. Using kDFF path is always recommended as the system maintains a descent output i.e. the performance is improved in this case. So by using dual feedforward path the system stability can be improved. Similarly the system can be tested by increasing or decreasing the number of neurons. Moreover, in the *k*DFF case this number can be much smaller and thus the resulting control algorithm has appreciably lower computational complexity and memory burden.

VI. CONCLUSION

The performance of the controller has been studied within the context of the pure sine wave inverter and a correct training algorithm is selected. The repetitive neurocontroller has been used and a modification was suggested to its approximation space. The resulting disturbance dual feedforward algorithm used here illustrates faster convergence than the controller without the *k*DFF path. Moreover, the number of neurons can be decreased due to the presence of the *k*DFF, which in turn makes the controller less prone to the overlearning phenomenon. It also reduces its computational complexity. In a noisy environment is to be noted that the approximation task available is quite challenging. This happens because the shape of the desired control signal acquires many features from the obtained load current waveform. The number of neurons is just a trade-off between the absence of excessive overlearning for linear loads and no-load conditions and an accurate control signal shaping for significantly nonlinear loads. The numerical experiments indicate that the *k*DFF-enabled controller with less than 10 neurons can effectively reduce disturbance caused by linear and nonlinear loads. By choosing the best learning algorithm the system can operate with better effectiveness in terms of execution time and offer potentially faster convergence rates which leads to less disturbance. Important aspects for future study includes the identification of its limits in any unknown noisy environments.

REFERENCES

[1] Xian Li, Qing-Guo Wang: Feedforward control with Disturbance prediction for Linear discrete-time systems, *IEEE Transaction on control systems technology*, 2018.

[2] Deng, H., Oruganti, R., and Srinivasan, D.: Neural controller for UPS inverters based on B-spline network, *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, 2008, pp. 899–909.

[3] Ufnalski, B. and Grzesiak, L. M.: A performance study on synchronous and asynchronous update rules for a plug-in direct particle swarm repetitive controller, *Archives of Electrical Engineering*, vol. 63, no. 4, 2014, pp. 635–646.

[4] Elci, H., Longman, R., Phan, M., Juang, J.-N., and Ugoletti, R.: Simple learning control made practical by zero-phase filtering: applications to robotics, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, 2002, pp. 753–767.

[5] J. Wall´en, "Estimation-based iterative learning control", *Ph.D. thesis*, Link¨oping University, Link¨oping, 2011.

[6] B. Ufnalski, “Plug-in direct particle swarm repetitive controller”, MATLAB Central, http://www.mathworks.com/matlabcentral/fileexchange/47847-plug-in-direct-particle-swarm-repetitive-controller (2014).

[7] A.Koerber and R. King, “Combined feedback–feedforward control of wind turbines using state-constrained model predictive control,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1117–1128, Jul. 2013.

[8] Ufnalski. B: *Repetitive Neurocontroller with Disturbance Feedforward*, 2014, url: www.mathworks.com/matlabcentral/fileexchange/47867-repetitive-neurocontroller-with-disturbance-feedforward.

[9] H. Zhong, L. Pao, and R. de Callafon, “Feedforward control for disturbance rejection: Model matching and other methods,” in *Proc. 24th Chin. Control Decis. Conf. (CCDC)*, May 2012, pp. 3528–3533

[10] B. Ufnalski, L.M. Grzesiak, and K. Galkowski, “Particle swarm optimization of an iterative learning controller for the single-phase inverter with sinusoidal output voltage waveform”, *Bull. Pol. Ac.: Tech.* 61 (3), 649–660 (2013).

[11] M. Kaminski and T. Orlowska-Kowalska, “Application of neural network with adaptive interaction for speed control of the drive system with elastic joint”, *IEEE Int. Symp. Industrial Electronics* (ISIE) 1, 1–6 (2013).

[12] Pajchrowski. T, Zawirski. K, and Nowopolski. K: A neural speed controller trained on-line by means of modified RPROP algorithm, *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, 2015, pp. 560–568.

[13] MathWorks: *Neural Network Toolbox MATLAB/Simulink)*, 2015, url: www.mathworks.com/help/nnet.

[14] MathWorks: *Nguyen-Widrow layer initialization function*, 2015, url: www.mathworks.com/help/nnet/ref/initnw.html.

[15] M. Kaminski, T. Orlowska-Kowalska, and K. Szabat, “Neural speed controller based on two state variables applied for a drive with elastic connection”, *16th Int. Power Electronics and Motion Control Conference and Exposition* (PEMC), 610–615 (2014).

[16] T. Pajchrowski, “Adaptive neural speed controller for servodrive trained online”, *18th Int. Conf. Methods and Models in Automation and Robotics* (MMAR) 1, 183–188 (2013).