

# Repetitive neurocontroller with disturbance feedforward path active in the pass-to-pass direction for a VSI inverter with an output LC filter

B. UFNALSKI\* and L.M. GRZESIAK

Faculty of Electrical Engineering, Warsaw University of Technology, 75 Koszykowa St., 00-662 Warsaw, Poland

**Abstract.** An enhancement to the previously developed repetitive neurocontroller (RNC) is discussed and investigated in the paper. Originally, the time-base generator (TBG) has been used to produce the only input signal for the neural approximator. The resulting search space makes the dynamic optimization problem (DOP) of shaping the control signal solvable with the help of a function approximator such as the feed-forward neural network (FFNN). The plant under consideration, i.e. a constant-amplitude constant-frequency voltage-source inverter (CACF VSI) with an output LC filter, is assumed to be equipped with the disturbance load current sensor to enable implementation of the disturbance feed-forward ( $p$ DFF) path as a part of the non-repetitive subsystem acting in the along the pass  $p$ -direction. An investigation has been undertaken to explore potential benefits of using this signal also as an additional input for the RNC to augment the approximation space and potentially enhance the convergence rate of the real-time search process. It is numerically demonstrated in the paper that the disturbance feed-forward path active in the pass-to-pass  $k$ -direction ( $k$ DFF) improves the dynamics of the repetitive part as well indeed.

**Key words:** repetitive control, feedforward neural network, dynamic optimization problem, repetitive disturbance rejection, voltage-source inverter, disturbance dual feedforward path.

## Nomenclature

2D	– Two-dimensional (here control system),	$p$ -path	– Along the pass controller path,
ANN	– Artificial Neural Network,	PDPSRC	– Plug-in Direct Particle Swarm Repetitive Controller,
BP (L–M)	– Error Back-Propagation learning/optimization algorithm (here the Levenberg-Marquardt algorithm),	RC	– Repetitive Control(ler),
CACF VSI	– Constant-Amplitude Constant-Frequency Voltage-Source Inverter,	RFF	– Reference Feed-Forward,
DDFF	– Disturbance Dual Feed-Forward path, i.e. acting in both $p$ - and $k$ -directions,	RMS	– Root Mean Square value of a signal (here calculated within one period of a reference signal),
DFF	– Disturbance Feed-Forward,	RMSE	– Root Mean Squared Error (here calculated within one period of a reference signal),
DOP	– Dynamic Optimization Problem,	RNC	– Repetitive Neuro-Controller,
EDO	– Evolutionary dynamic optimization,	SOP	– Static Optimization Problem,
FFNN	– Feed-Forward Neural Network (here of the multi-layer perceptron type),	TBG	– Time-Base Generator,
FSF	– Full-State Feedback (controller),	$w_{nm}^{(i)}$	– Neural weight connecting input/neuron $m$ with neuron $n$ in the $i$ -th layer,
ILC	– Iterative Learning Control(ler),	$\mathbf{w}^{(i)}$	– Matrix of neural weights in the $i$ -th layer,
IMP	– Internal Model Principle,	$\bullet^m$	– Measurement signal corrupted with noise,
$k$ DFF	– Disturbance Feed-Forward path in the $k$ -direction, i.e. acting in the pass-to-pass direction (a novel approach),	$\bullet^{\text{ref}}$	– Reference signal.
$k$ -direction	– Pass-to-pass direction,		
$k$ -path	– Pass-to-pass controller path,		
nonRC	– Non-repetitive part of controller (or signal),		
$p$ DFF	– Disturbance Feed-Forward path in the $p$ -direction, i.e. acting in the along-the-pass direction (a standard approach),		
$p$ -direction	– Along the pass direction,		

## 1. Introduction

The repetitiveness of a process to be controlled is innate to many industrial systems. Its presence creates new possibilities in developing control laws that take into account the 2D nature of a plant. Therefore, the repetitive control (RC) or iterative learning control (ILC) are often an appealing solution. It should be noted that the difference between RC and ILC is rather subtle and both of them can be analyzed in a uniformed framework [1]. The ILC community mainly focuses on batch processes and it is common to assume that the

\*e-mail: bartlomiej.ufnalski@ee.pw.edu.pl

initial state of each batch can (and usually is) reset. ILC has gained some acceptance in industrial robotics [2, 3] and the ABB robotic laser cutting solutions [4] can serve as an example. By contrast, RC is mainly used in continuous processes characterized by the initial state of each pass originating from the final state of the previous pass, i.e. no resetting is acceptable. Control tasks encountered in power electronic converters are sometimes related to continuous repetitive processes. Any CACF converter may work here as an illustrative example. Power electronic practitioners often attack the repetitiveness of a process with the help of multioscillatory (also known as multiresonant) controllers [5]. The internal model principle (IMP) is hence employed. However, more and more often the distinction between ILC and RC is blurred – this can be particularly pertinent to control schemes that are developed for continuous repetitive processes, and as such could be labelled as RC, but at the same time the techniques adopted are clearly of an iterative nature and incorporate some kind of learning, including reinforcement learning. Learning in this context means any ability to use past experiences from previous passes (or at least a single one) in making control decisions in the current pass, which should not be confused here with adaptive nonlinear operation [6].

If high quality voltage waveform is expected in CACF VSI with an output LC filter regardless of the nonlinear load current presence, the repetitiveness of the process is valuable and should be exploited. The most common approach is to include oscillatory terms within the controller to get load harmonics selective rejection [5, 7]. However, there are two major obstacles related to the implementation of that scheme: its computational burden and the problematic synthesis of oscillatory terms near the Nyquist limit. That is why many CACF VSI controllers are still designed as non-repetitive ones, e.g. [8].

There is an emerging group of control solutions for repetitive processes and this can be attributed to the growing computational power of the off-the-shelf microcontrollers. Any control task can be reformulated to pose the problem as an on-line, and usually dynamic, optimization problem. For example, the model predictive control (MPC) is a widely acknowledged optimal control scheme that is designed to shape a control signal according to a user-defined performance index (also called cost function, cost functional or fitness function). In MPC a model of the plant serves the role of a critic. The formulation of the DOP for repetitive processes is directly inspired by the ILC technique. The goal is to shape the control signal according to the user-defined functional. A relevant solver operates in on-line mode and tackles the DOP in an iterative manner, where one iteration of the solver corresponds to a total multiple of passes and a physical plant handles the role of the critic. The feasibility of such an approach to repetitive control has already been demonstrated using two types of DOP solvers. In [9–11] an ANN training algorithm has been used to solve online the DOP in RC whereas in [12–14] an evolutionary dynamic optimizer (EDO) based on a modified particle swarm optimizer (PSO) has been employed to solve the same problem and as a result the plug-in direct particle swarm controller (PDPSRC) has been proposed. It was decid-

ed at that time to label both of them as repetitive controllers, rather than ILCs, and this is continued throughout the present paper.

In what follows, the basic FFNN-based RC is described and the enhancement to the approximation space used by the FFNN is proposed. The performance of both controllers is then compared numerically.

## 2. Long-term stability of the classic iterative learning control system – a critical review

There are also attempts to replace all oscillatory terms with a single model of any periodic signal

$$u(p, k) = u(p, k - 1) + k_{RC}e(p, k - 1), \quad (1)$$

where  $u$  denotes the control signal,  $e$  is the control error,  $k_{RC}$  is the controller gain,  $k$  is the iteration (pass, trial, cycle) index,  $p$  is the time index along the pass ( $1 \leq p \leq \alpha$ , where  $\alpha$  is the pass length) [15]. This is an example of the classic ILC law and has its background in IMP for 2D systems. Unfortunately, the control law of the form (1) is impractical for most real-life applications due to the stability issue in a long time horizon. It should be noted that (1) is nothing more than integral action in the  $k$ -direction. A perfectly repetitive signal is constant in this direction and integral action is sufficient to drive the control error to zero, at least theoretically. Any physical system is subject to some limitations. For example, the discussed VSI with the LC output filter cannot completely reject a disturbance load current if a certain value of its derivative over time is crossed. This happens due to a limited DC-link voltage and an intentional inductance of the output filter. This, in turn, implies that such a repetitive load current tends to destabilize the system. A high value of the derivative of a discontinuous current drawn by the diode rectifier can serve as an example. To robustify the system, it is necessary to introduce filtering to stop learning for higher frequencies

$$u(p, k) = \mathbf{Q}(z^{-1})u(p, k - 1) + k_{RC}\mathbf{L}(z^{-1})e(p, k - 1), \quad (2)$$

where  $\mathbf{Q}$  and  $\mathbf{L}$  are non-causal zero-phase-shift low-pass filters [16–18]. Nonetheless, to stabilize the system in the infinite time horizon in the presence of every possible repetitive disturbance,  $\mathbf{L}$  should have infinite attenuation in the stopband, which is unavailable for any practical digital filter. Alternatively,  $\mathbf{Q}$  can be designed to introduce a necessary forgetting of high frequency components. At the moment, there are no comprehensive theoretical analyses that deal with the stability of ILC under unknown repetitive disturbance. However some of our studies indicate that the filtering in (2) often only delays the build-up of oscillations in the control signal. Moreover, it is often true that a sufficiently strong filtering deteriorates the performance of the system to the point of being practically equivalent to standard 1D feedback control [19]. Obviously, the infinite time horizon is not required in real-life control systems. Hence, the minimum necessary level of filtering is always problem specific and its goal is to stabilize the system for a given number of repetitions. Also, not

all industrial repetitive processes are significantly disturbed by non-stationary exogenous phenomena, which makes some practical tasks much easier as only the reference tracking problem has to be addressed. On the other hand, if the system is disturbed by significant yet stationary exogenous phenomena, the learning can just be stopped by setting  $k_{RC}$  equal to zero after the initial drop of control error to prevent any further overlearning. The ILC approach gained some acceptance in robotics due to the usually moderate number of required repetitions, i.e. up to hundreds of thousands, and often stationary disturbances. In contrast, power electronics practitioners deal with repetitive processes that can require more than tens of millions of stable repetitions and are often subject to non-stationary disturbances, e.g. CACF VSI operating continuously for one month under different load types, which is why the control law (2) has not yet gained much (if any) acceptance among power electronics control engineers.

### 3. FFNN-based repetitive controller

The main motivation is to develop repetitive control schemes that innately do not suffer from long-term stability issue and thus no special measures have to be taken to ensure stable operation. For example, the PDPSRC cost function is the combination of the control error cost and the control signal dynamics cost [12, 20]. The latter component does not allow for the high-frequency oscillations build-up. Also the FFNN-based RC [9, 10], although using the cost of a control error as the only objective, does not produce high-frequency os-

cillations thanks to the limited approximation capability of the FFNN with a fixed number of neurons, fixed activation functions and a constrained weight space. It is important to note that if the FFNN's activation functions obey an explicit set of assumptions, then the network is a universal function approximator, provided that sufficiently many hidden neurons are available [21]. It is then clear that, on the one hand, the limited number of neurons is essential to prevent overlearning. On the other hand, it is challenging (or even impossible in some approximation spaces) to select the complexity of the network equally effective in the case of linear loads as well as for highly nonlinear loads. It is demonstrated here that adding a disturbance feedforward path in the  $k$ -direction improves the performance and also makes selecting the number of neurons less challenging.

**3.1. The  $p$ -direction controller.** The proposed system is sketched in Fig. 1. The FFNN-based repetitive controller, similarly to any plug-in RC, needs to be accompanied by a  $p$ -direction controller if the dynamics requires to be shaped also along the pass, which could be the case in some applications of the highly underdamped CACF VSI (see parameters in Table 1). In this study, the full state feedback (FSF) controller is used for this purpose

$$u_{FSF} = -(k_{11}i_L^m + k_{12}u_C^m), \quad (3)$$

where  $\{k_{11}, k_{12}\}$  are the controller gains and  $\{i_L^m, u_C^m\}$  are the measured state variables of the output LC filter. Moreover, the FSF shapes the dynamics seen by the plug-in repetitive con-

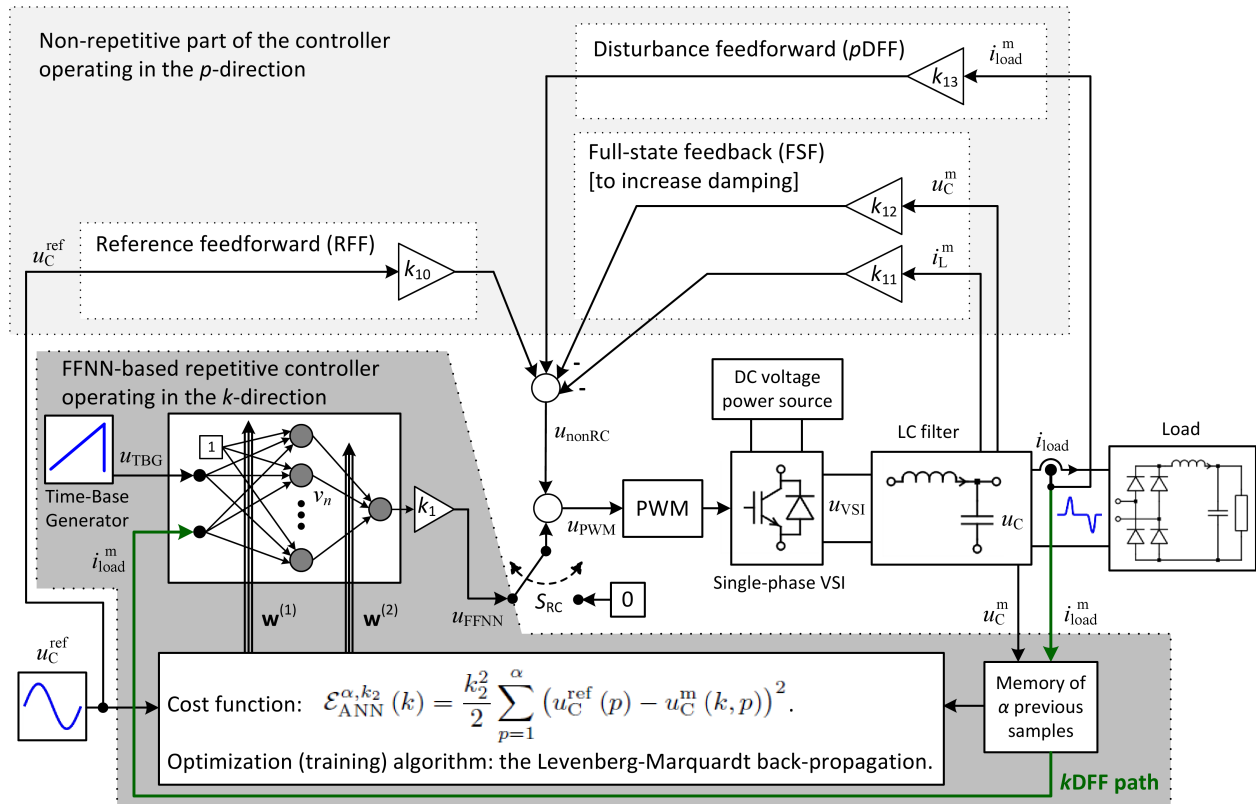


Fig. 1. Schematic diagram of the proposed system with a disturbance dual feedforward path (the block Load depicts an exemplary diode rectifier)

Table 1  
Selected parameters of the model

Component/Parameter	Description/Value
Inverter and measurement interface	
LC output filter	300 $\mu$ H, 160 $\mu$ F, $R_f = 0.6 \Omega$
Resonant frequency	726 Hz
Critical damping resistance	2.74 $\Omega$ (highly underdamped)
Reference output voltage	$f^{\text{ref}} = 50$ Hz, $U_{\text{RMS}}^{\text{ref}} = 230$ V, sinusoidal
Sampling time	$T_s = 100 \mu\text{s}$ ( $\alpha = 200$ points per pass)
Measurement noise	3% of 100 A or 325 V (band-limited white noise with 95% of samples within the range)
Test loads	
Load-1	Resistive: 4 kW
Load-2	Diode rectifier: 500 $\mu$ H, 3 mF, 6 kW, crest factor of ca. 2.5
FSF and $p$ DFF – the non-repetitive controller	
Closed-loop system damping	3 times higher than in the open-loop system
Identified filter resistance	$\hat{R}_f = 0.25R_f$ (significant identification error assumed to highlight the dynamics of the repetitive part)
Multilayer perceptron – the repetitive controller with $k$ DFF	
Number of hidden neurons	17 (or 7 for purposes of comparison)
Type of neurons	<code>tansig*</code> (excl. the output <code>purelin</code> neuron)
Training method type	Levenberg–Marquardt backpropagation (Matlab's <code>trainlm()</code> )
Learning parameters	default <code>trainlm()</code> settings, except <code>net.trainParam.epochs=1</code>
Weight constraints	Yes, in the interval $[-25,25]$
FFNN input(s) scaling	Yes, into the interval $[-1,1]$
Measurement and reference scaling	None
FFNN output signal gain	$k_1 = 100$ (by trial and error method)
Control error gain (scaling done before error backpropagation)	$k_2 = 1/100$ (by trial and error method)

\* the hyperbolic tangent (`tanh`).

troller and by increasing damping makes the relevant approximation task easier. Also the reference feedforward (RFF) path

$$u_{\text{RFF}} = k_{10}u_C^{\text{ref}} = (1 + k_{12})u_C^{\text{ref}} \quad (4)$$

is introduced to keep the unity gain for the zero frequency and the disturbance feedforward ( $p$ DFF) path is added to compensate partially for the resistive voltage drop

$$u_{p\text{DFF}} = k_{13}i_{\text{load}}^{\text{m}} = (\hat{R}_f + k_{11})i_{\text{load}}^{\text{m}}, \quad (5)$$

where  $\hat{R}_f$  is the identified resistance of the output LC filter. A significant identification error is assumed in this study ( $\hat{R}_f = 0.25R_f$ ) to make the control errors produced by the non-repetitive path more realistic. The sum of all these three paths

$$u_{\text{nonRC}} = u_{\text{RFF}} + u_{\text{FSF}} + u_{p\text{DFF}} \quad (6)$$

represents the non-repetitive part of the overall control structure.

**3.2. The  $k$ -direction controller (the basic approach).** The repetitive path includes a universal function approximator in the form of FFNN and an accompanying DOP-capable learning algorithm – here the Levenberg-Marquard (L-M) training method has been used. It has been demonstrated in numerous studies that an FFNN trained in the online mode using error backpropagation (BP) methods such as L-M or resilient

BPs (RPROPs) can effectively control non-repetitive processes [22–31]. Surprisingly, the well-documented usefulness of online trained neurocontrollers for non-repetitive processes in adjustable speed drives and generators has not been followed by a similarly rich literature on neurocontrollers for repetitive processes. The repetitiveness of the process can be easily exploited by the neurocontroller simply by using an adequate cost function.

For the purpose of controlling an output voltage of the CACF VSI, the following cost function is used

$$\mathcal{E}_{\text{ANN}}^{\alpha, k_2}(k) = \frac{k_2^2}{2} \sum_{p=1}^{\alpha} (u_C^{\text{ref}}(p) - u_C^{\text{m}}(k, p))^2, \quad (7)$$

where  $u_C^{\text{ref}}$  is the reference voltage,  $u_C^{\text{m}}$  denotes the measured output filter capacitor voltage and  $k_2$  is the error scaling factor. It should be noted that  $\mathcal{E}_{\text{ANN}}^{\alpha, k_2}$  is proportional to the mean squared error (MSE) calculated over the entire period of the reference signal ( $\alpha$  is again the pass length), hence the repetitiveness of the process at hand is directly incorporated into the functional definition. The basic FFNN [9, 10] has only one input connected to the time-base generator. The saw-shaped input signal is sufficient to turn the problem into constructing a relation of function type

$$u_{\text{FFNN}}(p) = w_{10}^{(2)} + \sum_{n=1}^N w_{1n}^{(2)} v_n(p), \quad (8)$$

where

$$v_n(p) = f_1 \left( w_{n0}^{(1)} + w_{n1}^{(1)} u_{\text{TBG}}(p) \right), \quad (9)$$

with  $u_{\text{TBG}}$  denoting the time-base generator signal,  $w$  being the ANN's weights (see Nomenclature section for details) and  $f_1$  expressing an activation function. It is common to use the hyperbolic tangent

$$f_1^{\text{tanh}}(y) = \frac{1 - e^{-2y}}{1 + e^{-2y}} = \frac{2}{1 + e^{-2y}} - 1 \quad (10)$$

or the Elliott function

$$f_1^{\text{Elliott}}(y) = \frac{y}{1 + |y|} \quad (11)$$

as the activation function with an emphasis on the latter if the computational burden of the neurocontroller needs to be considered for the final implementation of the training algorithm in a cost-effective microcontroller. The relevant discussion takes place in [32]. The functional (7) should then be seen as a function of the neural weights

$$\mathcal{E}_{\text{ANN}}^{\alpha, k_2}(k) = \mathcal{E}_{\text{ANN}}(\mathbf{w}^{(1)}(k), \mathbf{w}^{(2)}(k)) \quad (12)$$

and the training algorithm is employed to continuously solve the DOP of the form:

$$\begin{aligned} &\text{reduce iteratively} && \mathcal{E}_{\text{ANN}}(\mathbf{w}^{(1)}(k), \mathbf{w}^{(2)}(k)) \\ & && \mathbf{w}^{(1)}, \mathbf{w}^{(2)} \\ &\text{subject to} && \text{system nonlinearities,} \\ & && \text{system uncertainties,} \\ & && \text{system nonstationarity,} \\ & && N = \text{const,} \\ & && \text{constrained weight space,} \end{aligned} \quad (13)$$

where  $N$  is the number of hidden neurons (here organized into a single layer). The resulting output signal of the neural network is summed with (6) to produce reference signal for the modulator

$$u_{\text{PWM}} = u_{\text{nonRC}} + u_{\text{FFNN}}. \quad (14)$$

One strength of such an approach is that it uses the physical plant as the critic. It is also important to recall that the training algorithm is always on. In consequence, the action of the controller in the  $k$ -path considers plant nonlinearities such as converter saturation and PWM dead-time, uncertainties such as the  $R_f$  identification error in the  $p$ -path, and obviously the nonstationarity coming from variable load conditions. The  $k$ -path also compensates for the phase shift inherent to the proposed  $p$ -path, which is important if the synchronization is needed.

#### 4. Disturbance Dual Feed-Forward

It is assumed here that the system is equipped with a load current sensor. The availability of the disturbance signal has already been harnessed to add the non-repetitive  $p$ DFF. The  $p$ DFF is of the proportional type, i.e. it cannot completely

compensate for the voltage drop across the inductive component ( $L_f$ ). Moreover, the accuracy of the  $p$ DFF is affected by the resistance identification errors whereas potential  $k$ DFF would not be affected by such limitations since the  $k$ -path uses the physical plant as the critic in the relevant DOP. One of the key design decisions in any ANN-based solution is related to choosing a robust approximation space. The effectiveness of an ANN-based controller or estimator is highly influenced by the proper selection of input signals. There are no explicit rules on how the spanning vectors should be selected. Nevertheless, their correlation with the ANN desired output signal could serve as one of the hints [33]. Also, auxiliary information on the state of a plant can be passed to the neurocontroller in order to enhance its performance [34]. That information is supplementary in the way that it is not required to include those paths to secure stable operation and the desired steady-state performance. Nevertheless, such additional signals, if appropriately selected, can noticeably influence transients or simplify a tuning procedure, i.a. by making a neurocontroller less prone to inadequate ANN sizing.

The basic topology of the original RNC assumes that the ANN's input nodes are fed only by the TBG. This makes the DOP well-posed but not always well-conditioned. To make (9) well-conditioned, the number of neurons  $N$  in (13) would need to vary with the load type, i.e. should reflect the complexity of the load current shape. It is not uncommon in static optimization problems (SOPs) to enrich the training procedure with a growing and/or pruning algorithm. But such solutions for DOPs are rather immature and, to the best of the authors' knowledge, have not yet been developed for DOP-based controllers. In the case of neurocontrollers, such an architectural dynamic optimization would be very challenging. The neurocontroller would have to have both growing and pruning algorithms active at the same time and the executed architectural modifications should run seamlessly, i.e. no auxiliary performance evaluations would be allowed and growing/pruning would have to ensure smooth transitions on the ANN's output signal. It has then been decided that at the current stage of the research the problem will be converted into well-conditioned by introducing disturbance feedforward also in the  $k$ -direction (depicted as  $k$ DFF in Fig. 1). The equation (9) is modified accordingly into

$$v_n(p) = f_1 \left( w_{n0}^{(1)} + w_{n1}^{(1)} u_{\text{TBG}}(p) + w_{n2}^{(1)} i_{\text{load}}^m(p, k) \right), \quad (15)$$

to accommodate the new ANN's input signal. The theoretical perfect control signal would be non-negligibly correlated with the load current, hence the above candidate modification is not accidental. The resulting control structure with the initial  $p$ DFF and added  $k$ DFF has been named a disturbance dual feed-forward (DDFF).

The concept of utilizing the information on the external disturbance in the  $k$ -direction seems to be original for the ILC/RC systems. The proposed name ' $k$ -direction disturbance feedforward' needs to be justified. The name could create an impression of inconsistency as the disturbance signal

is passed to the repetitive part of the system that undeniably operates in the feedback mode. The  $p$ -direction disturbance feedforward used here, often termed for clarification the static one to indicate that no derivative of the disturbance signal is calculated [35], has a well-established name. It is important to note that the  $p$ DFF path is always impaired due to the delay introduced by the digital implementation of the controller. The term ‘disturbance static feedforward’ should be interpreted as indicating that the most recent available sample of the disturbance is used to modify the control signal. A similar interpretation can be made for the  $k$ -direction disturbance feedforward proposed here – the most recent complete pass of the disturbance is passed through the FFNN to produce a signal that directly modifies the control signal. This is clearly apparent at a steady state in the  $k$ -direction when the training algorithm is in fact superfluous and could be stopped. No  $k$ -direction derivative of the disturbance signal is present here at the inputs of the neuro-controller, hence this feedforward path could also be regarded as a static one. The training algorithm introduces the feedback action needed to relearn the FFNN when new load conditions arise. Therefore, the name DFF is used here also in relation to the  $k$ -direction. Nevertheless, it has been decided not to use the term static in connection to the  $k$ -direction in order to avoid potential confusion. The static  $p$ DFF cannot completely compensate for the voltage drop across the inductor present in the output filter. In contrast, the  $k$ DFF does not have that limitation because it operates on the complete period of the disturbance signal and contains sufficient information needed to compensate for the above-mentioned voltage drop.

### 5. Numerical experiment results

Both control strategies (basic and modified) have been tested in exactly the same load conditions to make results comparable, and the proposed scenario is as follows [36]:

- a) the FFNN is initialized with near zero weight matrices  $\mathbf{w}^{(1)}$  and  $\mathbf{w}^{(2)}$  to produce near zero  $u_{\text{FFNN}}(p)$  within the entire first pass ( $1 \leq p \leq \alpha$ ), i.e. no pre-tuning, e.g. for no-load conditions, is introduced,
- b) no-load conditions are assumed for the first 1 s (50 passes at 50 Hz),
- c) next the diode rectifier load (ca. 6 kW, current crest factor ca. 2.5) is applied for 6 s (300 passes at 50 Hz),
- d) then the diode rectifier load is switched off and the resistive load of ca. 4 kW is switched on for 2 s (100 passes at 50 Hz),
- d) later the resistive load is switched off and the diode rectifier load is switched on again for 2 s (100 passes at 50 Hz),
- f) finally the diode rectifier is switched off and the initial no-load conditions are applied once again.

Key parameters of the system are summarized in Table 1. Figures 2 and 3 illustrate the performance of the non-repetitive

part of the controller, i.e. for the switch  $S_{\text{RC}}$  in Fig. 1 set permanently to zero. The average voltage  $u_{\text{VSI}}^{\text{avg}}$  plotted in the selected figures is related to the converter output voltage  $u_{\text{VSI}}$  labeled in Fig. 1 as follows

$$u_{\text{VSI}}^{\text{avg}}(t) = \frac{1}{T_s} \int_t^{t+T_s} u_{\text{VSI}}(\tau) d\tau, \quad (16)$$

where  $t = \frac{k}{f^{\text{ref}}} + pT_s$  with  $f^{\text{ref}}$  denoting the reference voltage frequency and  $T_s$  being the sampling time. Then the repetitive part is switched on (the switch  $S_{\text{RC}}$  put in the position of ‘ $u_{\text{FFNN}}$ ’), initially without the  $k$ DFF path and with  $N = 17$  hidden neurons. The performance improves as shown in Figs. 4, 5 and 6. Next, the  $k$ DFF part is activated, i.e. the

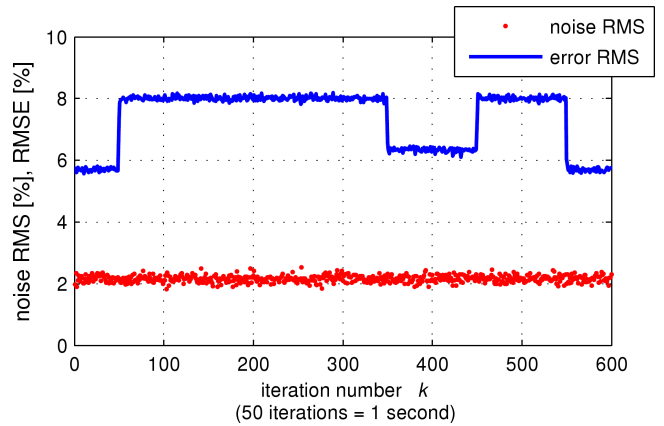


Fig. 2. Poor performance of the FSF+RFF+ $p$ DFF non-repetitive controller: the evolution of the RMSE – the measurement noise RMS given as some sort of referencing point

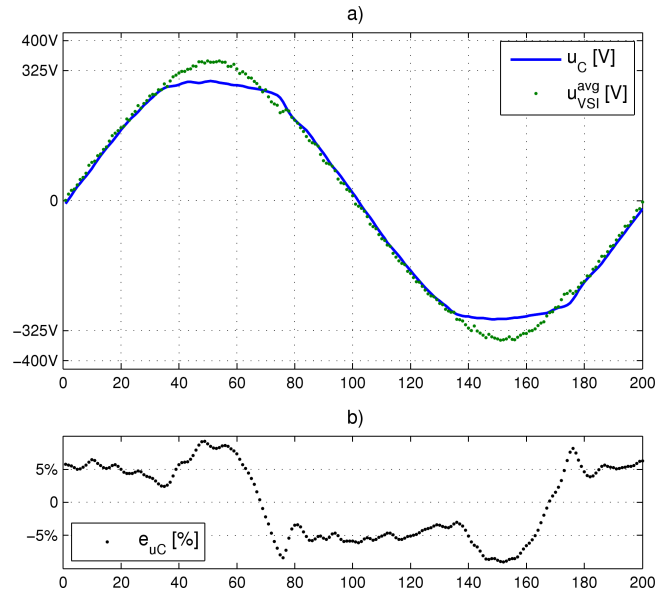


Fig. 3. Poor performance of the FSF+RFF+ $p$ DFF non-repetitive controller: significantly disturbed output voltage under the diode rectifier load (a) – the control error (b) and the converter average voltage (a) also provided

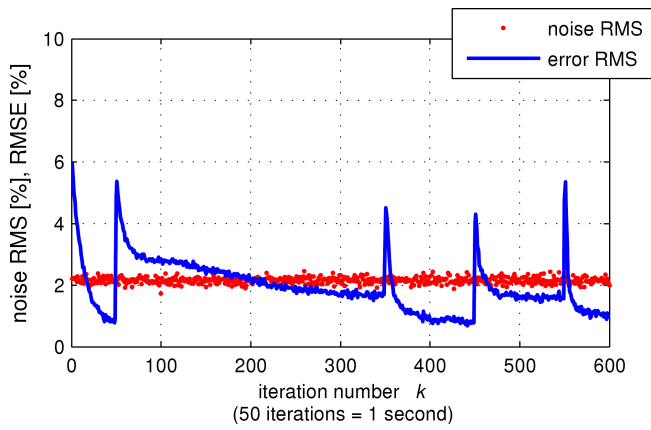


Fig. 4. Evolution of the RMSE for the repetitive neurocontroller composed of 17 neurons and without the  $k$ DFF path

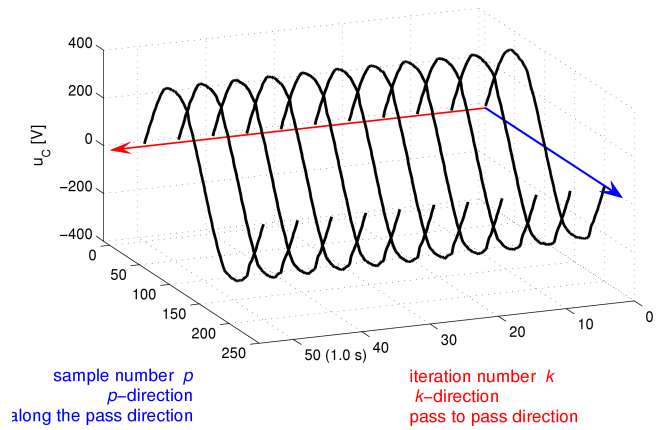


Fig. 5. Evolution of the output voltage in the  $k$ -direction after switching the diode rectifier on while keeping the  $k$ DFF path inactive

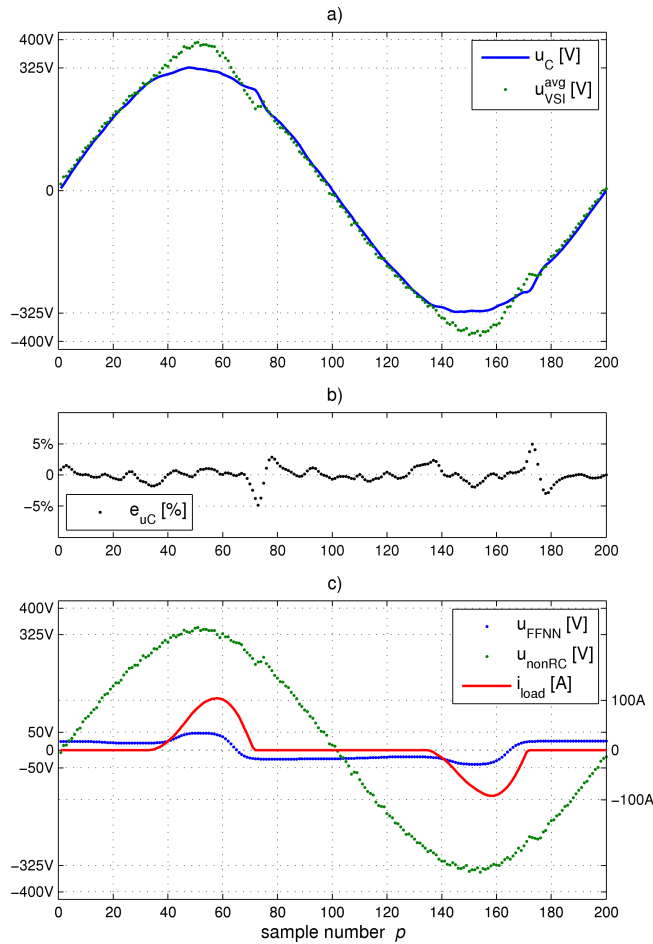


Fig. 6. Steady-state behavior of the  $k$ DFF-disabled controller under the diode rectifier load: a) the LC filter output voltage, the converter average voltage and b) the control error, c) the ANN's output signal, the non-repetitive path output and the load current

DDFF control algorithm is running. Figures 7, 8 and 9 demonstrate the performance of the controller with the DDFF paths. It can be observed that it is possible for the root mean squared error (RMSE) to drop below the noise RMS value because of

the low-pass filtering action inherently present in the online trained neural network having (7) as the cost function. The  $k$ DFF path introduces noticeably faster convergence of the overall controller, which could be read from the inset inside

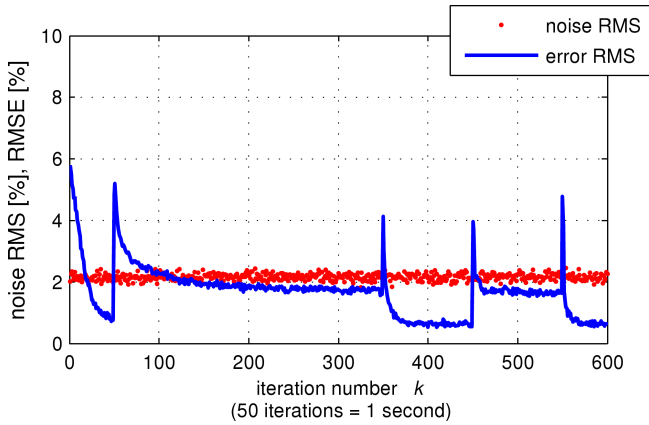


Fig. 7. Evolution of the RMSE for the repetitive neurocontroller composed of 17 neurons and with the  $k$ DFF path activated

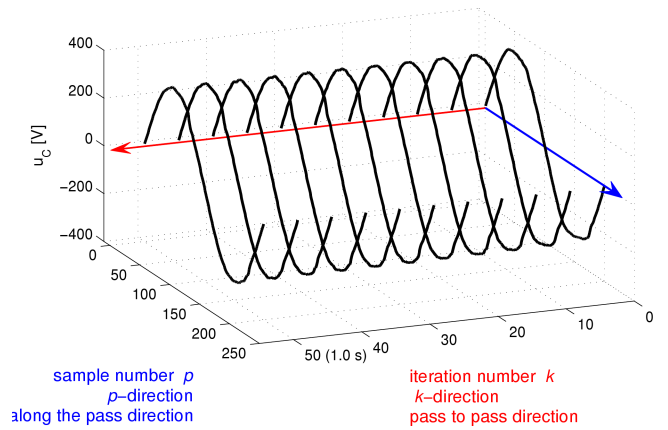


Fig. 8. Evolution of the output voltage in the  $k$ -direction after switching the diode rectifier on and for the  $k$ DFF path being activated

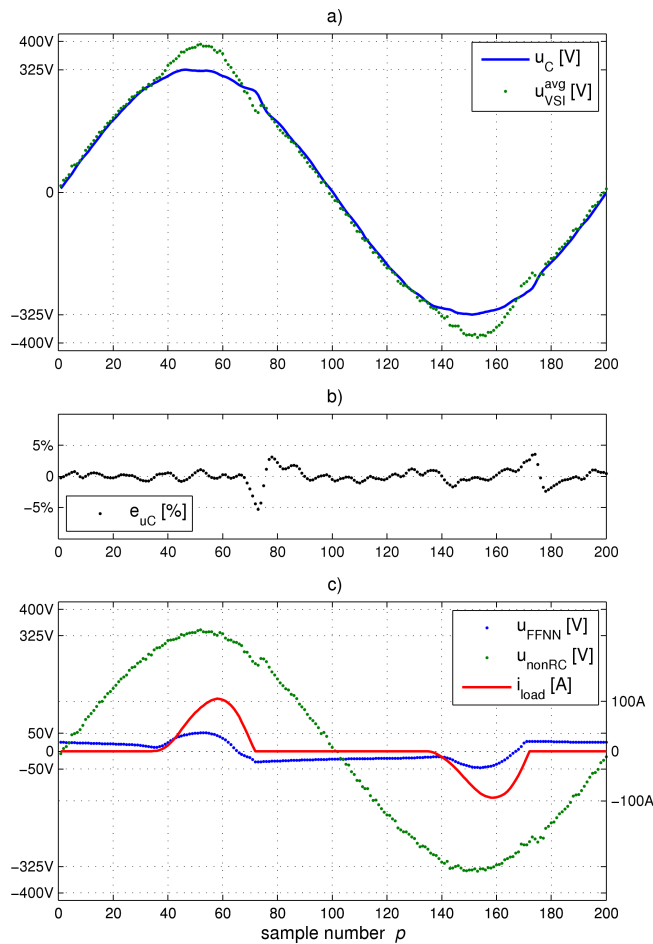


Fig. 9. Steady-state behavior of the  $k$ DFF-enabled controller under the diode rectifier load: a) the LC filter output voltage, the converter average voltage and b) the control error, c) the ANN's output signal, the non-repetitive path output and the load current

Fig. 10. Also the performance of the  $k$ DFF-enabled controller does not deteriorate significantly with the reduction in the number of neurons, which is not the case with the  $k$ DFF-disabled controller. The relevant comparison for  $N = 7$  neu-

rons is shown in Fig. 11. This implies that in the DDFF scheme it is easier to guess an effective number of hidden neurons. Moreover, in the  $k$ DFF case this number can be much smaller and thus the resulting control algorithm has ap-



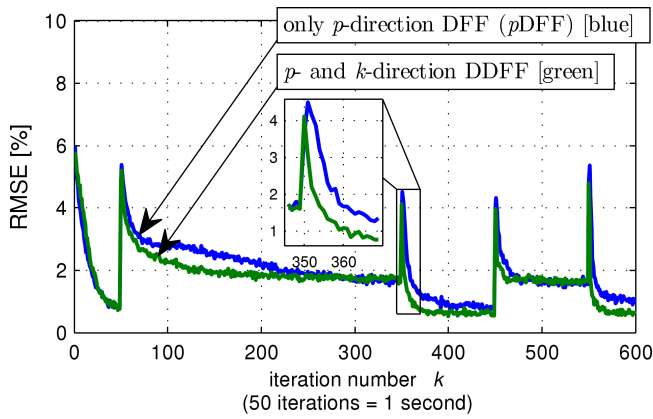


Fig. 10. Comparison of convergence rates (see slopes) and fitting capabilities (see steady-state values) for  $p$ DFF- and DDFE-equipped systems – the case of  $N = 17$  neurons

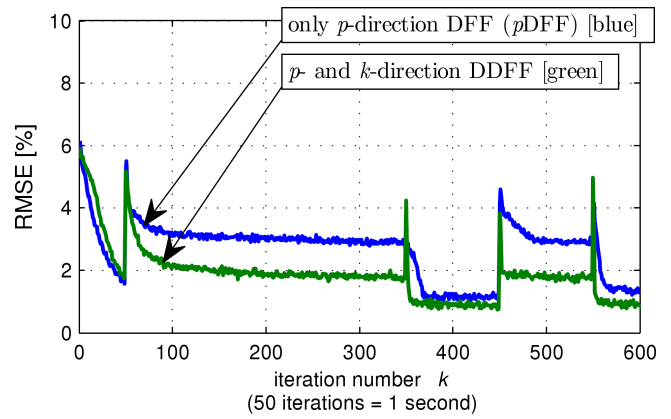


Fig. 11. Comparison of convergence rates (see slopes) and fitting capabilities (see steady-state values) for  $p$ DFF- and DDFE-equipped systems – the case of  $N = 7$  neurons

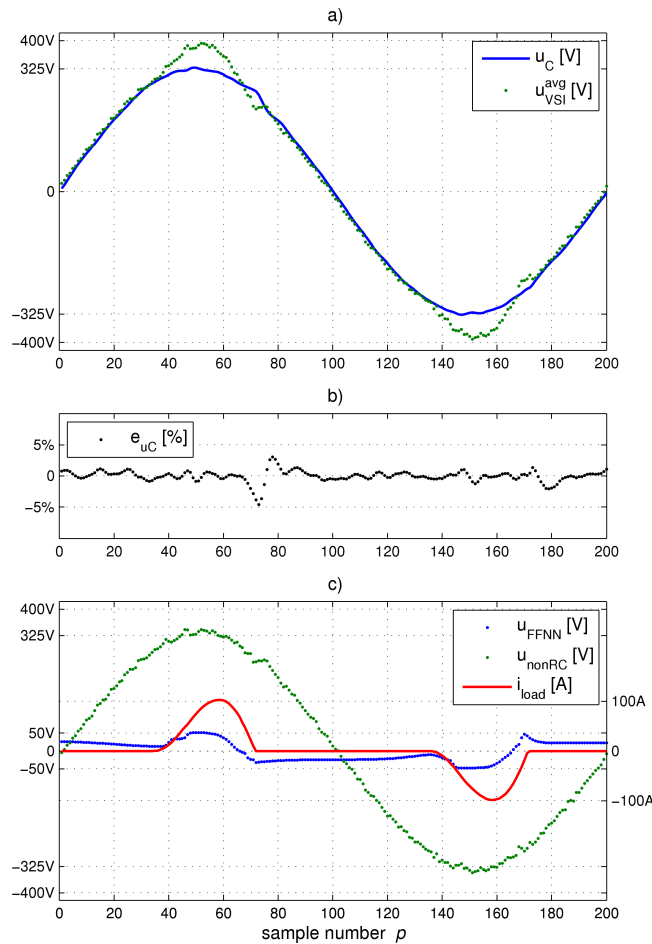


Fig. 12. Steady-state behavior of the controller after 100,000 iterations (passes) under the diode rectifier load – in this particular experiment one weight out of 29 has saturated: a) the LC filter output voltage, the converter average voltage and b) the control error, c) the ANN's output signal, the non-repetitive path output and the load current

preciably lower computational complexity and memory burden. To demonstrate the above-stated long-term stability, the neurocontroller has been kept under the diode rectifier load for one hundred thousand periods of the reference signal. Fig-

ure 12 clearly shows that the control signal  $u_{FFNN}$  does not feature any high-frequency oscillations – here in the case of  $N = 7$  hidden neurons.

## 6. Conclusions

The repetitive neurocontroller has been presented and a modification to its approximation space has been proposed. The disturbance feedforward is made active also in the pass-to-pass direction. The resulting disturbance dual feedforward algorithm manifests faster convergence than the controller without the  $k$ DFF path. Moreover, the presence of the  $k$ DFF path allows the number of neurons to be decreased, which in turn makes the controller less prone to the overlearning phenomenon and also reduces its computational complexity. It should be noted that the approximation task at hand is quite challenging from the generalization point of view in a noisy environment. This happens because the shape of the desired control signal inherits many features from the shape of the load current. The number of neurons is then a trade-off between an accurate control signal shaping for significantly nonlinear loads and an absence of excessive overlearning for linear loads and no-load conditions. The numerical experiments indicate that the  $k$ DFF-enabled neurocontroller with less than 10 neurons can effectively reject disturbance caused by linear as well as nonlinear loads. The proposed control scheme illustrates simultaneously that the disturbance feedforward path, quite common in non-repetitive control schemes, can be effectively incorporated also into the repetitive part of the controller. The latter has been already reported in the topical literature, e.g. [13,37], in the form of load current predictors that calculate future values in subsequent periods on the basis of the previous pass(es). This paper has demonstrated that it is also possible and beneficial to harness the measured disturbance in the DOP-solver-based repetitive controller for the CACF VSI with the output LC filter.

**Acknowledgments.** This research was partially supported by the statutory fund of the Electrical Drive Division within the Institute of Control and Industrial Electronics, the Faculty of Electrical Engineering, the Warsaw University of Technology, Poland.

## REFERENCES

- [1] Y. Wang, F. Gao, and F.J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control", *J. Process Control* 19 (10), 1589–1600 (2009).
- [2] J. Wallén, "Estimation-based iterative learning control", *Ph.D. thesis*, Linköping University, Linköping, 2011.
- [3] E. Rogers, K. Galkowski, W. Paszke, and D.H. Owens, "Two decades of research on linear repetitive processes part II: Applications", *Multidimensional Systems (nDS) 2013, Proc. 8th Int. Workshop on* 1, 1–6 (2013).
- [4] ABB Group, "ABB at the iREX 2013 – robotic laser cutting solutions", [abb.com](http://abb.com) (2013).
- [5] B. Ufnalski, L.M. Grzesiak, and A. Kaszewski, "Advanced control and optimization techniques in AC drives and DC/AC sine wave voltage inverters: selected problems", in: *Advanced and Intelligent Control in Power Electronics and Drives, Studies in Computational Intelligence*, eds. T. Orłowska-Kowalska, F. Blaabjerg, and J. Rodriguez, vol. 531, pp. 303–333, Springer, Berlin, 2014.
- [6] J. Farrell and W. Baker, "Learning control systems", in: *An Introduction to Intelligent and Autonomous Control*, eds. P.J. Antsaklis and K.M. Passino, Kluwer Academic Publishers, London, 1993.
- [7] B. Ufnalski, A. Kaszewski, and L.M. Grzesiak, "Particle swarm optimization of the multioscillatory LQR for a three-phase four-wire voltage-source inverter with an LC output filter", *IEEE Trans. Industrial Electronics* 62 (1), 484–493 (2015).
- [8] M. Monfared, "A simplified control strategy for single-phase UPS inverters", *Bull. Pol. Ac.: Tech.* 62 (2), 367–373 (2014).
- [9] B. Ufnalski and L.M. Grzesiak, "Artificial neural network based voltage controller for the single phase true sine wave inverter – a repetitive control approach", *Electrical Review (Przegląd Elektrotechniczny)* 89 (4), 14–18 (2013).
- [10] B. Ufnalski and L.M. Grzesiak, "Particle swarm optimization of an online trained repetitive neurocontroller for the sine-wave inverter", *Industrial Electronics Society, IECON 2013 – 39th Annual Conf. IEEE*, 6003–6009 (2013).
- [11] H. Deng, R. Oruganti, and D. Srinivasan, "Neural controller for UPS inverters based on B-spline network", *IEEE Trans. Industrial Electronics* 55 (2), 899–909 (2008).
- [12] B. Ufnalski and L.M. Grzesiak, "A plug-in direct particle swarm repetitive controller for a single-phase inverter", *Electrical Review (Przegląd Elektrotechniczny)* 90 (6), 6–11 (2014).
- [13] B. Ufnalski and L.M. Grzesiak, "Feedback and feedforward repetitive control of single-phase UPS inverters – an online particle swarm optimization approach", *Technical Report 1/2014*, Scientific Reports of the Cologne University of Applied Sciences, Köln, 2014.
- [14] B. Ufnalski and L.M. Grzesiak, "A performance study on synchronous and asynchronous update rules for a plug-in direct particle swarm repetitive controller", *Archives of Electrical Engineering* 63 (4), 635–646 (2014).
- [15] 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).
- [16] R.W. Longman, "Iterative/repetitive learning control: learning from theory, simulations, and experiments", in: *Encyclopedia of the Sciences of Learning*, pp. 1652–1657, Springer, New York, 2012.
- [17] H. Elci, R.W. Longman, M.Q. Phan, J.-N. Juang, and R. Ugoletti, "Simple learning control made practical by zero-phase filtering: applications to robotics", *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications* 49 (6), 753–767 (2002).
- [18] Y. Shi, "Robustification in repetitive and iterative learning control", *Ph.D. Thesis*, Columbia University, New York, 2013.
- [19] M.H.A. Verwoerd, "Iterative learning control – a critical review", *Ph.D. Thesis*, University of Twente, Enschede, 2005.
- [20] 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).
- [21] K. Hornik, "Approximation capabilities of multilayer feedforward networks", *Neural Networks* 4 (2), 251–257 (1991).
- [22] M. Kaminski and T. Orłowska-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).

- [23] M. Kaminski and T. Orłowska-Kowalska, "FPGA implementation of ADALINE-based speed controller in a two-mass system", *IEEE Trans. Industrial Informatics* 9 (3), 1301–1311 (2013).
- [24] T. Orłowska-Kowalska and M. Kaminski, "Adaptive neurocontrollers for drive systems: Basic concepts, theory and applications", *Advanced and Intelligent Control in Power Electronics and Drives, Studies in Computational Intelligence* 531, 269–302 (2014).
- [25] 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).
- [26] T. Pajchrowski and K. Zawirski, "Application of artificial neural network for adaptive speed control of PMSM drive with variable parameters", *COMPEL: Int. J. Computation and Mathematics in Electrical and Electronic Engineering* 32 (4), 1287–1299 (2013).
- [27] T. Pajchrowski, "Application of artificial neural network for speed control of servodrive with variable parameters", *Mechatronics 2013*, International Publishing, London, 693–700 (2014).
- [28] J. Sobolewski and L.M. Grzesiak, "Neuro-control system for converter based electrical energy source – test performed in laboratory setup with combustion engine emulator", *IEEE Int. Conf. Industrial Technology (ICIT)*, 1603–1608 (2013).
- [29] B. Ufnalski and L.M. Grzesiak, "Particle swarm optimization of artificial-neural-network-based on-line trained speed controller for battery electric vehicle", *Bull. Pol. Ac.: Tech.* 60 (3), 661–667 (2012).
- [30] L.M. Grzesiak, V. Meganck, J. Sobolewski, and B. Ufnalski, "On-line trained neural speed controller with variable weight update period for direct-torque-controlled AC drive", *12th Int. Power Electronics and Motion Control Conf. (EPE-PEMC)* 3, 1127–1132 (2006).
- [31] L.M. Grzesiak and J. Sobolewski, "Energy flow control system based on neural compensator in the feedback path for autonomous energy source", *Bull. Pol. Ac.: Tech.* 54 (3), 335–340 (2006).
- [32] D.L. Elliott, "A better activation function for artificial neural networks", *Institute for Systems Research (ISR) Technical Report 93-8*, CD-ROM (1993).
- [33] B. Ufnalski and L.M. Grzesiak, "Selected methods in angular rotor speed estimation for induction motor drives", *IEEE Int. Conf. Computer as a Tool (EUROCON)* 2, 1764–1771 (2007).
- [34] M. Kaminski, T. Orłowska-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).
- [35] B.W. Bequette, *Process Control: Modeling, Design, and Simulation*, Prentice Hall PTR, London, 2003.
- [36] B. Ufnalski, "Repetitive neurocontroller with disturbance feedforward", MATLAB Central, <http://www.mathworks.com/matlabcentral/fileexchange/47867-repetitive-neurocontroller-with-disturbance-feedforward> (2014).
- [37] K. Sozański, *Digital Signal Processing in Power Electronics Control Circuits*, Power Systems, Springer-Verlag, London, 2013.