# Feedback and Feedforward Repetitive Control of Single-phase UPS Inverters – an Online Particle Swarm Optimization Approach

Bartlomiej Ufnalski, Lech M. Grzesiak
Warsaw University of Technology

**Abstract**

**The paper presents a novel repetitive control of the constant-amplitude constant-frequency (CACF) voltage-source inverter (VSI) suitable for uninterruptible power supply (UPS) applications. The control system consists of three parts: a repetitive feedback controller, a repetitive disturbance feedforward controller and a non-repetitive full-state feedback controller. The repetitive feedback part is synthetized using the recently developed plug-in direct particle swarm repetitive control (PDPSRC) strategy. This approach is free from long-term stability issues encountered in the classical iterative learning control (ILC) schemes. The repetitive disturbance feedforward part is the extension of a non-repetitive disturbance feedforward controller. The repetitiveness of the load current has been harnessed to reduce the influence of the delay introduced by the digital control system. This makes the disturbance feedforward path more effective in comparison to the standard one. The non-repetitive controller is employed here just to increase damping in the system which is naturally significantly underdamped. A standard technique of full-state feedback and pole placement has been used to calculate its gains. The resulting control system has been tested with the help of numerical simulations.**

## 1. Introduction

A repetitiveness of the process to be controlled is often encountered in various areas of power electronics. The constant-amplitude constant-frequency voltage-source inverter can serve as a popular example of such a power electronic device. Under a steady load level and a non-varying load type, the control process becomes repetitive. Yet this repetitiveness is ignored in most control schemes. The main reason for this are widely-known stability issues related to the pass-to-pass operation of classical ILC algorithms [1][2][3][4][5]. It is then necessary to use additional low-pass filters to stabilize the system. Usually the acceptable, i.e. needed to ensure stability of the system, pass band of such filters appears to be relatively narrow. This in turn significantly deteriorates the effectiveness of the disturbance rejection. The classical approach to repetitive control is illustrated in Fig. 1. This type of the repetitive controller is sometimes called a plug-in one. Other controllers depicted in Fig. 1 are just to indicate that in the case of high performance systems an accompanying controller acting in the along-the-pass direction is needed. The name thus comes from its ability to be plugged into an existing non-repetitive controller. It should be noted that most repetitive techniques, including ILC and repetitive control (RC), can be analyzed in the uniform framework discussed in [6] and surrounded in Fig. 1 by the dotted line. The classical repetitive part of a control law is as follows

$$u_{\mathrm{RC}}(p,k) = \boldsymbol{Q}_{\mathrm{RC}}(z^{-1})u(p,k-1) + \boldsymbol{L}_{\mathrm{RC}}(z^{-1})e(p,k-1)\,, \quad (1)$$

where $\boldsymbol{Q}_{\mathrm{RC}}$ and $\boldsymbol{L}_{\mathrm{RC}}$ are zero-phase-shift filters, $p$ is the integer time index along the pass and $k$ is the integer pass index.

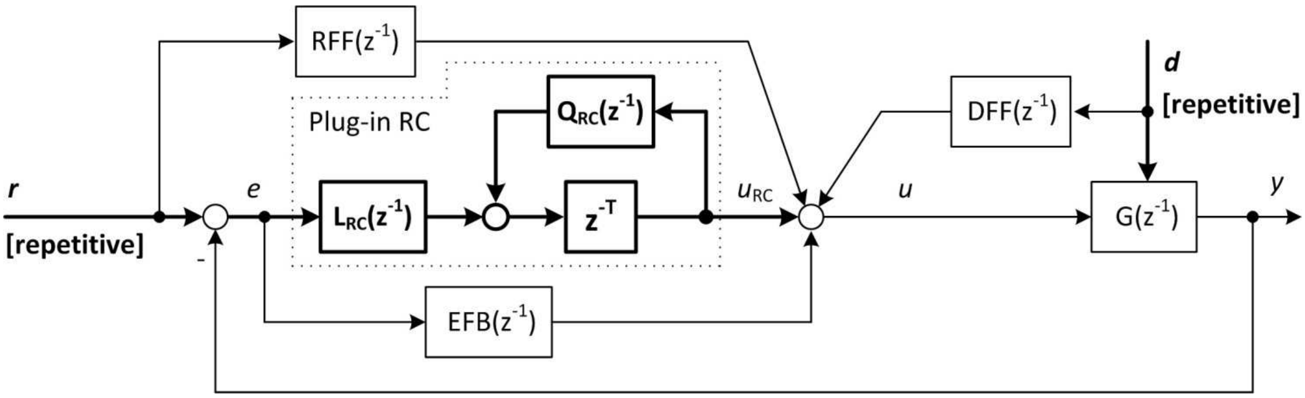


Fig. 1. A schematic diagram of a control system consisting of: plug-in type repetitive controller, error feedback (EFB) controller, reference feedforward (RFF) and disturbance feedforward (DFF) paths.

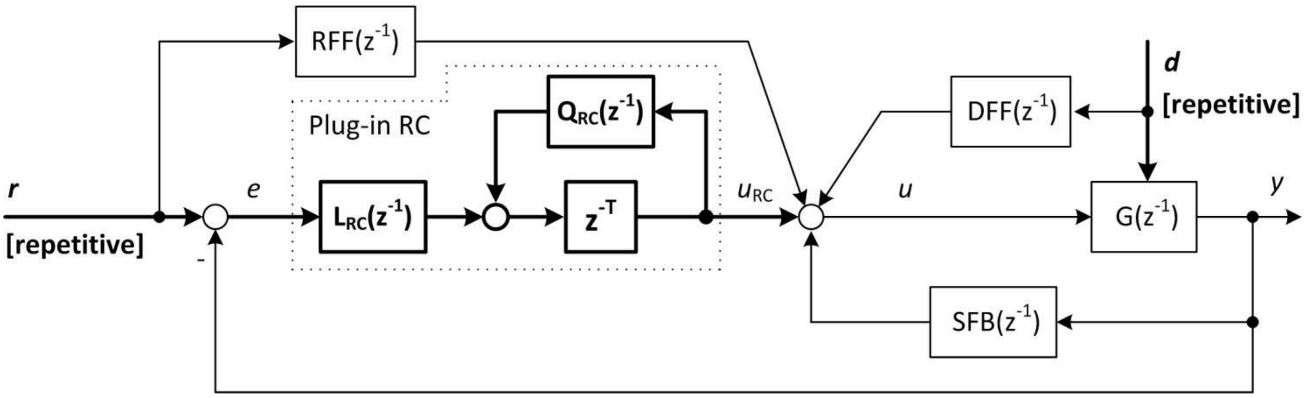


Fig. 2. A schematic diagram of a control system consisting of: plug-in type repetitive controller, state feedback (SFB) controller, reference feedforward (RFF) and disturbance feedforward (DFF) paths.

The plug-in type RC can be used in many other arrangements not covered by the schematic diagram in Fig. 1. For example, the state feedback (SFB) can be used to shape the dynamics of the plant seen by the RC as in Fig. 2 and the RC could be the only one exploiting the control error. Our idea is to alter two controllers in the scheme from Fig. 2 before using this control topology to synthetize a control system for the VSI with the LC output filter. First of all, the classical plug-in RC is to be replaced by the plug-in direct particle swarm repetitive controller (PDPSRC) to fully exploit the available bandwidth. The PDPSRC has been proposed recently as the alternative solution to the classical ILC schemes and is described in detail in [7]. Additionally, a modification to proportional disturbance current feedforward path is proposed here. This part of the control system is usually designed as non-repetitive one even if the disturbance is repetitive. However, our tests indicate that it could be beneficial to use the repetitiveness of the disturbance to compensate delays introduced by measurement signal conditioning, digital implementation of control and a pulse width modulator (PWM). The overall delay is usually around two sample periods. The detailed analysis is provided in [8]. This delay reduces the effectiveness of the DFF path. The performance of the developed repetitive disturbance feedforward (RDFF) controller does not deteriorates with this delay as far as a repetitive disturbance is present.

## 2. The plant to be controlled

The particle swarm repetitive controller under consideration does not require any plant identification stage. However, proposed here accompanying controllers acting in the $p$-direction require knowledge of the simplified linear model of the plant at the offline tuning stage. It has been assumed that the PWM inverter and measurement transducers all introduce unity gains. This simplification does not limit the generality of the discussion – they can always be forced to have resulting unity gain by scaling its physical gains in software procedures related to PWM and ADCs (analog-to-digital converters) handling. A continuous model is then as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{f}}(t)=\boldsymbol{A}\boldsymbol{x}_{\mathrm{f}}(t)+\boldsymbol{B}u(t)+\boldsymbol{E}i_{\mathrm{load}}(t), \tag{2}$$

with

$$\boldsymbol{A}=\begin{bmatrix}-\frac{R_{\mathrm{f}}}{L_{\mathrm{f}}} & -\frac{1}{L_{\mathrm{f}}}\\ \frac{1}{C_{\mathrm{f}}} & 0\end{bmatrix} \tag{3}$$

$$\boldsymbol{B}=\begin{bmatrix}\frac{1}{L_{\mathrm{f}}}\\ 0\end{bmatrix} \tag{4}$$

$$\boldsymbol{E}=\begin{bmatrix}0\\ -\frac{1}{C_{\mathrm{f}}}\end{bmatrix} \tag{5}$$

and

$$\boldsymbol{x}_{\mathrm{f}}=\begin{bmatrix}i_{\mathrm{L}}\\ u_{\mathrm{C}}\end{bmatrix}, \tag{6}$$

where: $L_{\mathrm{f}}$, $R_{\mathrm{f}}$, $C_{\mathrm{f}}$ denote inductance, resistance and capacitance of the output filter, $i_{\mathrm{load}}$ represents the disturbance current drawn by the load, $i_{\mathrm{L}}$ stands for the inductor current and $u_{\mathrm{C}}$ is the output capacitor voltage to be controlled. Selected parameters of the VSI with the LC filter are collected in Table 1. A schematic diagram of the plant and the proposed control system are sketched in Fig. 3. All depicted subsystems are discussed in following sections.

Table 1. Parameters of the converter

| Parameter | Symbol across the paper | Value |
|---|---|---|
| Filter inductance | $L_{\mathrm{f}}$ | 300µH |
| Filter capacitance | $C_{\mathrm{f}}$ | 160µF |
| Filter resistance | $R_{\mathrm{f}}$ | 400mΩ |
| Reference voltage frequency | $f^{\mathrm{ref}}$ | 50Hz |
| Reference voltage RMS value | – | 230Vrms |
| Sampling/PWM period | $T_{\mathrm{s}}$ | 100µs |
| Measurement noise | – | ca. 1% |

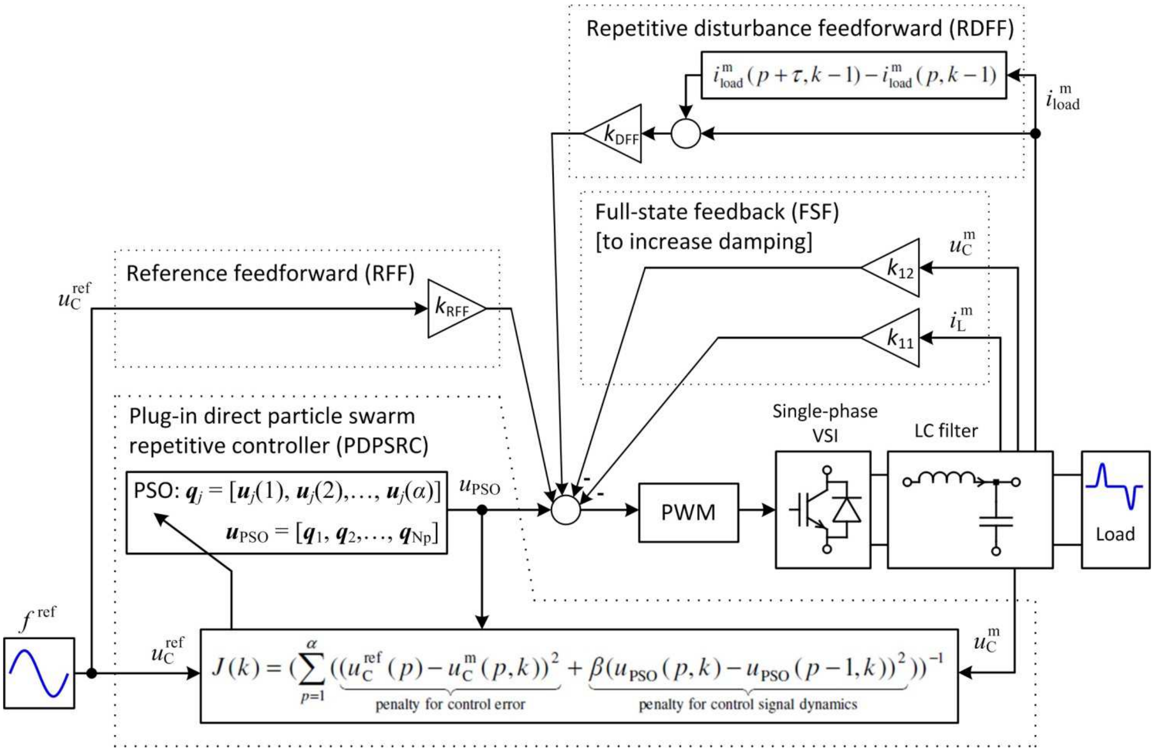


Fig. 3. A schematic diagram of the proposed repetitive control system.

## 3. Swarm repetitive controller

A direct swarm controller for the repetitive process has been proposed in [7]. It performs online optimization of the shape of the control signal according to a user-defined performance index

$$J(k) = \Big(\sum_{p=1}^{\alpha}\big(\underbrace{(u_C^{ref}(p)-u_C^{m}(p,k))^2}_{\text{penalty for control error}} + \underbrace{\beta(u_{PSO}(p,k)-u_{PSO}(p-1,k))^2}_{\text{penalty for control signal dynamics}}\big)\Big)^{-1}, \tag{7}$$

where the superscript $\bullet^{m}$ denotes a measurement signal corrupted by the noise and $\alpha$ is the pass length. The plant itself serves as the critic and the optimization takes place during the regular operation of the system – there is no need to switch into some sort of offline mode. Particles directly store samples of the control signal

$$\boldsymbol{q}_j = [\boldsymbol{u}_j(1), \boldsymbol{u}_j(2), \boldsymbol{u}_j(3), \ldots, \boldsymbol{u}_j(\alpha)], \tag{8}$$

where $j$ is the particle's identification number, and as such they justify the name *direct*. The future control signal is constructed from individual solutions by concatenating all $N_p$ vectors

$$\boldsymbol{u}_{PSO} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \ldots, \boldsymbol{q}_{N_p}], \tag{9}$$

where $N_p$ is the number of particles in the swarm. After passing all $\alpha N_p$ control signal values to the PWM, which enables to rate all particles and constitutes one iteration for the swarm, the swarm is updated and the process is repeated until the converter is in operation.

A standard PSO (particle swarm optimization) algorithm [9] is suitable for solving offline optimization problems in stationary environments. In the discussed converter the non-stationarity of the environment is inherent due to varying load conditions. The PSO has to be modified to enable tracking of moving optimum. Two techniques have been combined to keep the swarm alive. First of all, the repelling of particles has been accommodated into the speed update rule by allowing negative value for $\delta$ [10]

$$\boldsymbol{v}_j(i+1) = c_1\boldsymbol{v}_j(i) + c_2 r_\mathrm{p}\delta(\boldsymbol{q}_j^\mathrm{pbest} - \boldsymbol{q}_j(i)) + c_3 r_\mathrm{g}\delta(\boldsymbol{q}^\mathrm{gbest} - \boldsymbol{q}_j(i)), \tag{10}$$

where: $c_1$, $c_2$ and $c_3$ are constants shaping the behavior of the particles (i.e. their inertia, cognitive and social weights), $r_\mathrm{p}$ and $r_\mathrm{g}$ are random numbers uniformly distributed between $0$ and $1$, the superscripts $\bullet^\mathrm{pbest}$ and $\bullet^\mathrm{gbest}$ denote personal best solutions and a global best solution found so far. A diversity of the swarm is then controlled by switching between attract ($\delta = 1$) and repel ($\delta = -1$) modes

$$\delta = \begin{cases} 1 & \text{if} \quad \delta < 0 \wedge D_\mathrm{dist} > D_\mathrm{thold} + \dfrac{h}{2} \\ -1 & \text{if} \quad \delta > 0 \wedge D_\mathrm{dist} < D_\mathrm{thold} - \dfrac{h}{2} \end{cases}, \tag{11}$$

when

$$D_\mathrm{dist} = \frac{1}{N_\mathrm{p}\sqrt{N_\mathrm{d}}} \sum_{j=1}^{N_\mathrm{p}} \sqrt{\sum_{n=1}^{N_\mathrm{d}} (q_{jn} - \overline{q}_n)^2}, \tag{12}$$

being one of the commonly employed diversity measures [10], crosses the threshold $D_\mathrm{thold}$. The hysteresis $h$ has been set in the experiment to $0$. The second technique for non-stationary environments forces particles to loose gradually their fitness. The personal fitness

$$P_j = J(\boldsymbol{q}_j^\mathrm{pbest}), \tag{13}$$

which is constant in the basic PSO if a better solution does not emerge, here evaporates at the constant rate $\rho$ according to the rule [11]

$$\begin{bmatrix} P_j(i+1) \\ \boldsymbol{q}_j^\mathrm{pbest} \end{bmatrix} = \begin{cases} \begin{bmatrix} \rho P_j(i) \\ \boldsymbol{q}_j^\mathrm{pbest} \end{bmatrix} & \text{if} \quad J(\boldsymbol{q}_j(i+1)) \le \rho P_j(i) \\ \begin{bmatrix} J(\boldsymbol{q}_j(i+1)) \\ \boldsymbol{q}_j(i+1) \end{bmatrix} & \text{if} \quad J(\boldsymbol{q}_j(i+1)) > \rho P_j(i) \end{cases}, \tag{14}$$

where $\rho < 1$ for maximization problem. All parameters of the swarm are given in Table 2.

Table 2. Parameters of the swarm

| Parameter | Symbol across the paper | Value |
|---|---|---|
| Dimensionality of the problem (number of samples per pass) | $N_\mathrm{d}$, $\alpha$ | 200 |
| Number of particles | $N_\mathrm{p}$ | 25 |
| Swarm update frequency | $f^\mathrm{ref}/N_\mathrm{p}$ | 2Hz |
| Inertia weight | $c_1$ | 0.73 |
| Cognitive weight | $c_2$ | 1.5 |
| Social weight | $c_3$ | 1.5 |
| Evaporation constant | $\rho$ | 0.97 |
| Diversity threshold | $D_\mathrm{thold}$ | 0.5/325 |
| Penalty factor | $\beta$ | 0.25 |

**4. Full-state feedback controller**

The considered plant is inherently stable. However, natural damping is very low since parasitic resistance of the circuit is far below the critical one. A full-state feedback controller has been implemented to increase the damping of the system seen by the repetitive controller. The resulting $p$-direction dynamics is as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{\mathrm{f}}(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}_{\mathrm{f}}(t) + \boldsymbol{B}u(t) + \boldsymbol{E}i_{\mathrm{load}}(t)\,, \tag{15}$$

where

$$\boldsymbol{K} = [k_{11}, k_{12}] \tag{16}$$

can be calculated using the pole placement procedure available in many control system developer environments (e.g. *place()* from control system toolbox for Matlab or Octave, *ppol()* in Scilab). In this particular numerical experiment the damping has been increased five times, i.e. the closed-loop poles have been placed five times farther to the left in the complex plane in respect to the plant natural poles. The reference feedforward gain for the closed-loop system is given by the formula

$$k_{\mathrm{RFF}} = 1 + k_{12}\,. \tag{17}$$

The analysis relevant to the reference input design procedure is available in [12]. Such a value for $k_{\mathrm{RFF}}$ introduces a unity gain for the zero frequency.

**5. Repetitive disturbance feedforward**

It is to note that the PDPSRC is immune to delays present in every digitally controlled PWM inverter. The most significant delays come from measurement signal acquisition and conditioning, a non-zero step size of the control algorithm (due to its digital implementation) and the modulation. A detailed analysis of all mentioned delays can be found in [8]. The swarm controller uses the physical plant as the critic, thus the fitness of the particles takes into account all delays present in the system. Consequently, the search algorithm by maximizing (7) produces the control signal that is optimal for the particular delays present in the digital control system. Our idea is to exploit the repetitiveness of the load current to reduce the influence of the delays also in the disturbance feedforward path. The employed here disturbance proportional feedforward path is aimed to compensate the resistive voltage drop. Therefore, its gain is as follows

$$k_{\mathrm{DFF}} = R_{\mathrm{f}} + k_{11}\,. \tag{18}$$

This compensation is of static form, i.e. it neglects the inductive voltage drop. A calculation of the inductive voltage drop requires differentiation of the noisy measurement signal and usually aggressive filtering is needed to make it practical. This filtering limits the effectiveness of the dynamic compensation. It has been decided to focus here on the static compensation. Nevertheless, the ongoing research envisages a possibility to use the below described technique to compensate also the delay linked with smoothing prior to the differentiation.

The effectiveness of the DFF is affected by the overall delay time between the occurrence of the physical disturbance and applying updated output average voltage $u_{\mathrm{VSI}}^{\mathrm{average}}$ to the LC filter. In practical applications with a reasonable controller sample time and PWM frequency, the overall delay $\tau$ is not negligible and is no smaller than $1.5T_{\mathrm{s}}$ (see [8]). If the repetitiveness of the disturbance current is assumed, instead of relying solely on the measurement from the current pass, it is possible to make prediction using the increment from the previous pass

$$i_{\text{load}}^{\text{predicted}}(p,k) = i_{\text{load}}^{\text{m}}(p,k) + i_{\text{load}}^{\text{m}}(p+\tau,k-1) - i_{\text{load}}^{\text{m}}(p,k-1)\,. \tag{19}$$

If the load current is perfectly repetitive, the formula (19) simplifies to

$$i_{\text{load}}^{\text{predicted}}(p,k) = i_{\text{load}}^{\text{m}}(p+\tau,k-1)\,. \tag{20}$$

However, to reduce compensation errors during a non-repetitive disturbance, i.e. during transients caused by a change in a load type and/or its power, the formula (19) should be implemented. If $\tau$ is an integer multiple of the sample time $T_{\text{s}}$, the procedure reduces to reading relevant samples memorized in the previous pass. Otherwise, the linear interpolation can be applied to approximate predicted load current at any given point in time not covered by the sampling points. The proposed acronym for this kind of path is RDFF (repetitive disturbance feedforward).

**6. Numerical experiment results**

The following test has been performed to illustrate the benefits of the RDFF over the classical DFF:

a) the swarm is initialized with near zero $\boldsymbol{u}_{\text{PSO}}$ control vector (no pre-tuning is assumed),

b) the resistive load of ca. 4kW is applied for 750s,

c) the resistive load is switched off and the diode rectifier (ca. 6kW, current crest factor ca. 2.5) is switched on for 1500s,

d) the diode rectifier is switched off and the initial resistive load is applied once again.

The delay in the feedforward path is assumed to be $\tau = 2T_{\text{s}}$. The evolution of the root mean square error (RMSE) calculated for the single pass is compared in Fig. 4. The difference is clearly apparent – in this particular experiment the RMSE peaks have been reduced approximately twice.

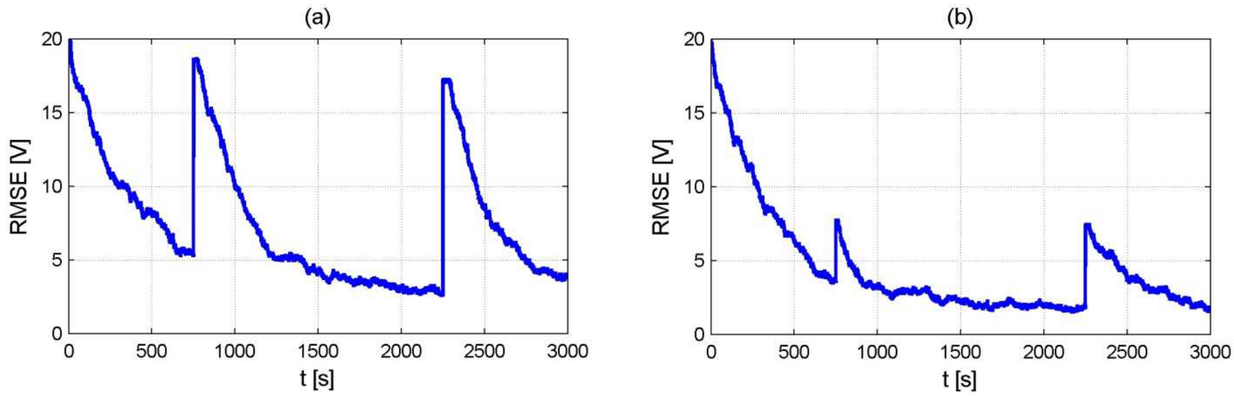


Fig. 4. Evolution of the root-mean-square error for the delay-non-compensated DFF (a) and the delay-compensated RDFF (b).

Unsurprisingly, the effective controller acting in the $p$-directions reduces time needed by the controller acting in the $k$-direction to force the control error down to the measurement accuracy. The two-dimensional (2D) behavior of the system is illustrated in Fig. 5. The quality of the output voltage along with the control signal are shown in Fig. 6. The observed instantaneous control error is near the boundary determined by the measurement noise.

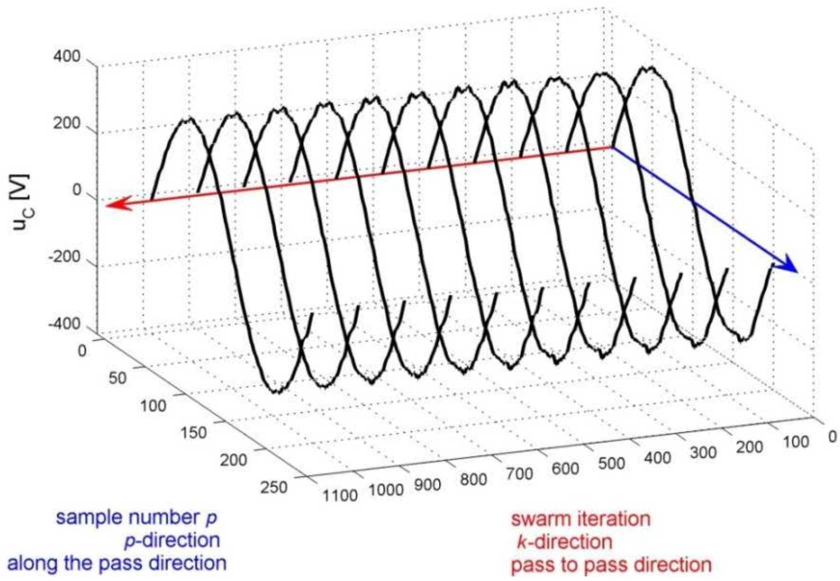


Fig. 5. Evolution of the output voltage in 2D after switching on the nonlinear load.

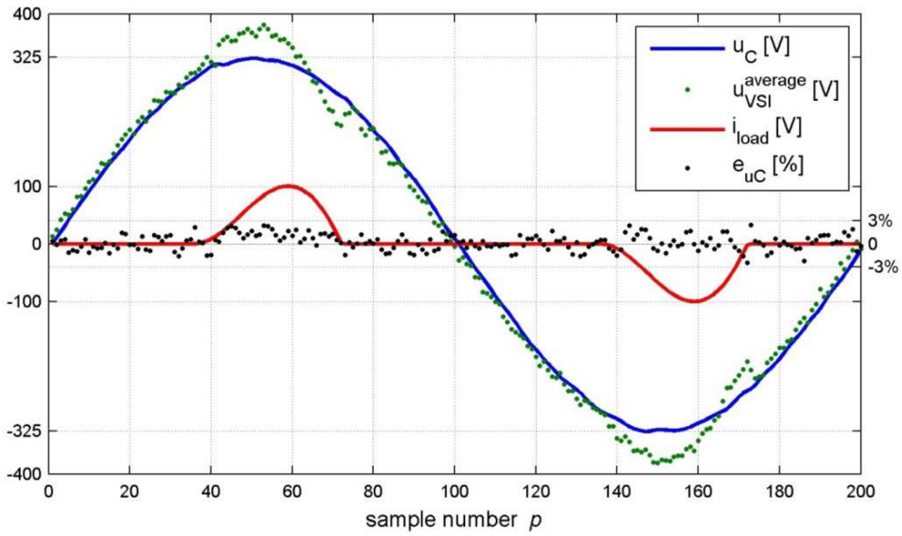


Fig. 6. Quality of the output voltage under repetitive load current drawn by the nonlinear load.

## 7. Conclusion

The previously developed repetitive controller for the constant-amplitude constant-frequency converter with the LC output filter has been enhanced by adding the predictive disturbance feedforward path. The prediction scheme allows to compensate the delays introduced by the digital control system and the modulator. This in turn makes the disturbance feedforward path more effective. The compensation algorithm exploits 2D features of the system. Samples of the load current from the previous pass are used to introduce delayless feedforward action. Proposed methodology is currently under the experimental verification.

**Acknowledgment**

The research work has been part-financed by the statutory grant for Electrical Drive Division (2013).

**References**

[1] Cai Z.: Iterative learning control: algorithm development and experimental benchmarking, Ph.D. thesis, University of Southampton, UK, 2009.

[2] Shi Y.: Robustification in repetitive and iterative learning control, Ph.D. thesis, Columbia University, USA, 2013.

[3] Longman R.W.: On the interaction between theory experiments and simulation in developing practical learning control algorithms, International Journal of Applied Mathematics and Computer Science, 13(1), pp. 101–111, 2003.

[4] Longman R.W.: Iterative/repetitive learning control: learning from theory, simulations, and experiments, Encyclopedia of the Sciences of Learning, Springer, pp. 1652–1657, 2012.

[5] Elci H., Longman R.W., Phan M.Q., Je-Nan Juang, 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, 49(6), pp. 753–767, Jun. 2002.

[6] Wang Y., Gao F., Doyle III F.J.: Survey on iterative learning control, repetitive control, and run-to-run control, Journal of Process Control, 19(10), pp. 1589–1600, Dec. 2009.

[7] Ufnalski B., Grzesiak L.M.: A plug-in direct particle swarm repetitive controller for a single-phase inverter, XI Conference on Sterowanie w Energoelektronice i Napedzie Elektrycznym (Control in Power Electronics and Electrical Drives, paper in English, *sene.p.lodz.pl*), SENE 2013, pp. 1-6, Nov. 2013 (preliminary accepted for publication).

[8] Nussbaumer T., Heldwein M.L., Guanghai Gong, Round S.D., Kolar J.W.: Comparison of prediction techniques to compensate time delays caused by digital control of a three-phase buck-type PWM rectifier system, IEEE Transactions on Industrial Electronics, 55(2), pp. 791–799, Feb. 2008.

[9] Eberhart R.C., Shi Y., Kennedy J.: Swarm Intelligence, 1st Edition, The Morgan Kaufmann Series in Evolutionary Computation, Morgan Kaufmann Publishers, 2001.

[10] Riget J., Vesterstrom J.S.: A diversity-guided particle swarm optimizer - the ARPSO, EVALife Technical Report no. 2002-02, Aarhus Universitet, Denmark, 2002.

[11] Cui X. , Charles J.St., Potok T.E.: A simple distributed particle swarm optimization for dynamic and noisy environments, Studies in Computational Intelligence: Nature Inspired Cooperative Strategies for Optimization (NICSO 2008), 236, pp. 89–102, 2009.

[12] Franklin G., Powell D., Workman M.: Digital control of dynamic systems, 3rd Edition, Prentice Hall, Dec. 1997.

**Authors**

**Dr.-Ing. Bartlomiej Ufnalski**
**Prof. Dr.-Ing. Lech M. Grzesiak**
Warsaw University of Technology
Faculty of Electrical Engineering
Institute of Control and Industrial Electronics
75 Koszykowa Str., Warsaw 00-662, Poland